



Fifth International Symposium
on Artificial Intelligence,
Robotics and
Automation in Space

iSAIRAS '99



ESTEC, Noordwijk, The Netherlands
1 – 3 June 1999

GEN92

esa SP-440
August 1999

iSAIRAS '99

Fifth International Symposium

on

**Artificial Intelligence,
Robotics and Automation
in Space**

**ESTEC, Noordwijk, The Netherlands
1 - 3 June 1999**

European Space Agency
Agence spatiale européenne

Title: Proceedings of the Fifth International Symposium on Artificial
Intelligence Robotics and Automation in Space,

Reference: ESA SP-440

Editor: Michael Perry

Published by: ESA Publications Division, Noordwijk, The Netherlands

ISBN: 92-9092-760-7

Copyright: © 1999 European Space Agency

Price: € 57.00

iSAIRAS '99

Objectives of the Symposium

Following the four previous meetings held in Kobe (Japan 1990), Toulouse (France 1992), Pasadena (USA 1994) and Tokyo (Japan 1997), i-SAIRAS '99 was the fifth in this series of international symposia. It was devoted to the technology of Artificial Intelligence (AI), Automation and Robotics (A&R) and its application in space. The Symposium took place on 1-3 June 1999 at ESTEC, the European Space Agency's Technology Centre located at Noordwijk in the Netherlands. The main topics covered by the Symposium were:

Artificial intelligence for space systems

- Spacecraft autonomy: Onboard software for mission planning and execution (resource management, fault protection, science data analysis, guidance, navigation and control), smart sensors, testing and validation, architectures;
- Mission operations automation: Decision support tools (for mission planning and scheduling, anomaly detection and fault analysis), innovative operations concepts, data visualisation;
- Design tools and optimisation methods, electronic documentation;
- Artificial intelligence methods (automated planning and scheduling, agents, model-based reasoning, machine learning and data mining).

Robotics and automation for space systems

- Application scenarios (e.g. space base assembly and servicing, external and internal payload tending, satellite inspection and servicing, planetary and cometary exploration, ground processing), programmatic and utilisation aspects;
- Robotics technologies for A&R systems, support equipment, ground segments, mobility, manipulation, end effectors and tools, sensing and robot vision, control, robot-friendly payload design, test and operations;
- Technology for (non-robotic) space laboratory automation, payload control systems, data communications, imaging, user interfaces and telepresence/telescience.

iSAIRAS '99

Sponsoring Organisations

Italian Space Agency, ASI (Italy)
French National Space Agency, CNES (France)
German Aerospace Centre, DLR, (Germany)
European Space Agency, ESA
Institute of Space and Astronautical Science, ISAS (Japan)
National Aerospace Laboratory, NAL (Japan)
National Aeronautics and Space Administration, NASA (USA)
National Space Development Agency, NASDA (Japan)
Netherlands Agency for Aerospace Programs, NIVR, (Netherlands)

Symposium Chair Panel

G. Hirzinger, DLR (Germany)
M. Montemerlo, NASA (USA)
K. Tsuchiya, Kyoto University (Japan)

Chairpersons of the Regional Programme Committees

R. Doyle, NASA (USA)
P. Putz, ESA
Y. Wakabayashi, NASDA (Japan)

European Organisation Committee

P. Putz, ESA (Chair)
N. Bataille, CNES
D. De Hoop, NIVR
S. Di Pippo, ASI
G. Gölz, DLR
M. Maurette, CNES

CONTENTS

Technology Surveys

Session Chairman: P. Putz, *European Space Agency*

Autonomous Rover Technology for Mars Sample Return, Weisbin C.R., Rodriguez G., Schenker P.S., Das H., Hayati S., Baumgartner E.T., Maimone M., Nesnas I, Volpe R.A., <i>Jet Propulsion Laboratory (USA)</i>	1
NASDA's Activities in Space Robotics, Ohkami Y., Oda M., <i>National Space Development Agency (Japan)</i> .	11
Autonomous Locomotion: CNES Technological Programme Lamboley M., Maurette M., <i>Centre National d'Etudes Spatiales (France)</i>	19
DLR's Robotics Lab - Recent Developments in Space Robotics, Hirzinger G., Landzettel K., Brunner B., Schaefer I., Fischer M., Grebenstein M., Sporer N., Schott J., Schedl M., Deutrich C., <i>German Aerospace Centre - DLR (Germany)</i> .	25

Space Robotics Applications on the ISS (1)

Session Chairman: J.C. Piedboef, *Canadian Space Agency (Canada)*

Overview of the Mobile Servicing System for the International Space Station, Stieber M.E., Hunter D.G., <i>Canadian Space Agency (Canada)</i> . Abramovici A., <i>M.D. Space and Advanced Robotics Ltd. (Canada)</i>	37
Special Purpose Dexterous Manipulator (SPDM) Requirements Verification, Bassett D., <i>Canadian Space Agency (Canada)</i> , Abramovici A., <i>M.D. Space and Advanced Robotics Ltd. (Canada)</i> .	43
ERA: The Flexible Robot Arm, Schoonejans P., <i>European Space Agency, ESTEC, Oort M., Fokker Space (Netherlands)</i>	49
Assembly of Large Spacecraft: The XEUS Mission, Didot F., Bavdaz M., Schiemann J., <i>European Space Agency</i> , Knoop U., <i>Daimler-Chrysler Aerospace (Germany)</i> Petersen H, <i>Fokker Space (The Netherlands)</i>	55

Rover Systems (1)

Session Chairman: R. Siegwart, *Ecole Polytechnique Fédérale de Lausanne (Switzerland)*

Designing of Lunar Rovers for High Work Performance, Aizawa J., Yoshioka N., Miyata M., Wakabayashi Y. <i>National Space Development Agency (Japan)</i> .	63
Nanokhod Microover Heading Towards Mars, Van Winnendael M., Visentin G., <i>European Space Agency</i> , Bertrand R., <i>Von Hoerner and Sulger GmbH (Germany)</i> , Rieder R., <i>Max-Planck Institut für Chemie (Germany)</i>	69
Low Power Mobility System for Micro Planetary Rover <i>Micro 5</i> , Kuroda Y., Kondo K., Nakamura K., <i>Meiji University (Japan)</i> , Kunii Y., <i>Chuo University (Japan)</i> , Kubota T., <i>Institute of Space and Astronautical Science (Japan)</i> .	77
Hopping Rover MINERVA for Asteroid Exploration, Yoshimitsu T., <i>University of Tokyo (Japan)</i> , Kubota T., Nakatani I., <i>Institute of Space and Astronautical Science (Japan)</i> , Adachi T., Saito H., <i>Nissan Motor Company (Japan)</i> .	83

Planning and Scheduling of Space Operations (1)

Session Chairman: K. Matsumoto, *National Aerospace Laboratory (Japan)*

Multi-Agent Planning and Scheduling Environment for Enhanced Spacecraft Autonomy, Das S.K., Gonsalves P., <i>Charles River Analytics Inc. (USA)</i> , Krikorian R., <i>MIT Media Laboratory (USA)</i> , Truszkowski W., <i>National Aeronautics and Space Administration (USA)</i> .	91
Iterative Repair Planning for Spacecraft Operations Using the ASPEN System, Rabideau G., Knight R., Chien S., Fukunaga A., Govindjee A., <i>Jet Propulsion Laboratory (USA)</i> .	99

Citizen Explorer- A Low-Cost Distributed and Incrementally Automated Mission Operations System, Willklow C., Antell J., AuCoin A., Doraisingam S., Faber J., Hansen E., , <i>Colorado Space Grant Consortium (USA)</i> .	107
The Real-Time Execution Performance Agent - An Approach for Balancing Hard and Soft Real-Time Execution for Space Applications, Siewert S., Nutt G., Hansen E., , <i>University of Colorado (USA)</i> .	111

Space Robotics Applications on the ISS (2)

Session Chairman: C. Heemskerck, *Fokker Space Systems (The Netherlands)*

EUROPA - External Use of Robotics for Payloads Automation, Mugnuolo R., Bracciaferri F., <i>Agenzia Spaziale Italiana (Italy)</i> . Didot F., <i>European Space Agency</i> , Colombina G., Pozzi E., <i>Tecnospazio (Italy)</i> .	121
Development of the European Technology Exposure Facility, Borghi G., <i>Carlo Gavazzi Space S.p.A. (Italy)</i> , Dettmann J., Visentin G., <i>European Space Agency</i> , .	129
Payload Tutor (PAT): A Relocatable Payload Robot for ISS Internal Automation System, Di Pippo S., Mugnuolo R., Bracciaferri F., <i>Agenzia Spaziale Italiana (Italy)</i> , Williams W.B., <i>National Aeronautics and Space Administration (USA)</i> , Visentin G., <i>European Space Agency</i> , Colombina G., Pozzi E., <i>Tecnospazio (Italy)</i> .	137
Wearable Exo-Skeletal Robot Skil Mate and its Application to EVA Suits, Umetani Y., Yamada Y., Morizono T., <i>Toyota Technological Institute (Japan)</i> , Yoshida T., Aoki S., <i>Shimizu Corporation (Japan)</i> .	145

Autonomy in Planetary Exploration

Session Chairman: R. Doyle, *Jet Propulsion Laboratory (USA)*

A Comparison of Three Ground-Based Path Planners for Planetary Rovers, Tarokh M., <i>San Diego State University (USA)</i> , Shiller Z., <i>University of California-Los Angeles (USA)</i> , Hayati S., <i>Jet Propulsion Laboratory (USA)</i> .	151
Traversability Index: A New Concept for Planetary Rovers, Seraji H., <i>Jet Propulsion Laboratory (USA)</i> .	159
Increased Flexibility and Robustness of Mars Rovers, Bresina J.L. Golden K., Smith D.E., Washington R., <i>National Aeronautics and Space Administration, (USA)</i> .	167
Autonomy Technology Challenges of Europa and Titan Exploration Missions, Atkinson D.J., <i>Jet Propulsion Laboratory (USA)</i> .	175

Automation of Spacecraft Mission Operations

Session Chairman: T. Iwata, *National Space Development Agency (Japan)*

Computer Intelligence in Integrated Satellite Design Support Infrastructure, Nakasuka S., Maeda K., <i>University of Tokyo (Japan)</i> , Sato M., Kiritani K., Sato K., Koda T., <i>Mitsubisi Electric Corporation (Japan)</i>	185
An AI Approach to Ground Station Autonomy for Deep Space Communications, Fisher F., Estlin T., Mutz D., Chien S., <i>Jet Propulsion Laboratory (USA)</i> .	191
Intelligent Optical Polarimetry Development for Space Surveillance Missions, McMackin L., Zetocha P., Sparkman C., <i>Air Force Research Laboratory (USA)</i> McIntire H., Fetrow M., Bishop K., <i>Applied Technology Associates (USA)</i>	199
Dealing with Uncertainty when Managing an Earth Observation Satellite, Bensana E., Verfaillie G., <i>ONERA-CERT (France)</i> , Michelon-Edery C., Bataille N., <i>CNES (France)</i> .	205

Robotic Servicing Demonstration Missions

Session Chairman: Y. Okhami, *National Space Development Agency (Japan)*

- Demonstration Mission of a Satellite Servicing System, 211
 Oda M., Inaba N., *National Space Development Agency (Japan)*.
- Vision and Interactive Autonomy Bi-Lateral Experiments on the Japanese Satellite ETS-VII, 217
 Galardini D., Kapellos K., Maesen E., *Trasys Space (Belgium)*. Visentin G., Didot F., *European Space Agency*
- The Ranger Telerobotic Shuttle Experiment: An On-Orbit Satellite Servicer, 225
 Parrish J.C., *National Aeronautics and Space Administration, (USA)*.
- The ISS Inspector Mission, 233
 Wilde D., Brüge U, *Daimler-Benz Aerospace Infrastructure (Germany)*, Sytin O., *RSC-Energia (Russia)*.

Rover Systems (2)

Session Chairman: S. Hayati, *Jet Propulsion Laboratory (USA)*

- Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica, 237
 Moorehead S., Simmons R.G., Apostolopolous D., Whittaker W., *Carnegie Mellon University (USA)*.
- Solar Power Expert for Remote Robotic Explorers, 243
 Shillcutt K., Whittaker W., *Carnegie Mellon University (USA)*.
- ROAMS: Rover Analysis Modeling and Simulation, 249
 Yen J., Jain A., Balaram J., *Jet Propulsion Laboratory (USA)*.
- An Integrated Architecture for Co-operating Rovers, 255
 Estlin T., Yen J., Petras R., Mutz D., Castaño R., Rabideau G., Steele R., Jain A., Chien S., Mjolsness E., Gray A., Mann T., Hayati S., Das H., *Jet Propulsion Laboratory (USA)*.

Spacecraft Autonomy

Session Chairman: F. Richard, *Alcatel (France)*

- Integrated Planning and Execution for Satellite Tele-Communications, 265
 Rajan K., Plaunt C., *Caelum Research Corporation (USA)*. Pell B., *Marketplace.Net (USA)*,
 Muscettola N., *RECOM Technologies (USA)*.
- Satellite Tele-Communications Scheduling as Dynamic Constraint Satisfaction, 277
 Plaunt C., Frank J., *Caelum Research Corporation (USA)*, Jonsson A.K., *RIACS, (USA)*.
- Adaptive Resource Profiling, 285
 Decoste D., *Jet Propulsion Laboratory (USA)*.
- Autonomy Architectures for a Constellation of Spacecraft, 291
 Barrett A., *Jet Propulsion Laboratory (USA)*.

Robotic Satellite Servicing Experiments on ETS-7

Session Chairman: G. Hirzinger, *German Aerospace Centre (Germany)*

- Results of the ETS-7 Mission - Rendezvous Docking and Space Robotics Experiments, 299
 Kasai T., Oda M., Suzuki T., *National Space Development Agency (Japan)*.
- Antenna Assembly Experiments Using ETS-VII, 307
 Kimura S., Tsuchiya S., *Communications Research Laboratory (Japan)*.
- Teleoperation Control of ETS-7 Robot Arm for On-Orbit Truss Construction, 313
 Matsumoto K., Wakabayashi S., Penin. L.F., Nohmi M., *National Aerospace Laboratory (Japan)*,
 Ueno H., Yoshida T., Fukase Y., *Shimizu Corp. (Japan)*.
- Performance Evaluation of Advanced Robotic Hand System in Space Experiment, 319
 Machida K., *Ministry of Trade and Industry (Japan)*, Akita K., *USEF (Japan)*,
 Ohno K., Moriya M., Nishida H., Ohsawa T., *Fujitsu Ltd. (Japan)*

Rover Control (1) : Control of Special Tasks

Session Chairman: J.L. Bresina, *National Aeronautics and Space Administration*

- Autonomous Rock Tracking and Acquisition from a Mars Rover, 329
 Maimone M., Nesnas I., Das H., *Jet Propulsion Laboratory (USA)*.
- Autonomous Sample Acquisition for Planetary and Small Body Explorations, 335
 Ghavimi A.R., Serricchio F., Dolgin B., Hadaegh F.Y., *Jet Propulsion Laboratory (USA)*.
- Evolution of Autonomous Self-Righting Behaviors for Articulated Nanorovers, 341
 Tunstel E., *Jet Propulsion Laboratory (USA)*.

Spacecraft Autonomy: Experiments on Deep Space One

Session chairman: P. Zetocho, *Air Force Research Laboratory (USA)*

- Validating the DS-1 Remote Agent Experiment, 349
 Nayak P., Kurien J., *National Aeronautics and Space Administration (USA)*. Dorais G., Millar W., Rajan K., Kanefsky R., *Caelum Research (USA)*, Bernard D.E., Gamble E.B. Jr., Rouquette N., Smith B.D., Tung Y-W., *Jet Propulsion Laboratory (USA)*, Muscoletta N., Taylor W., *Recom Technologies (USA)*
- Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment, 357
 Wyatt J., Sherwood R., Sue M., Szijarto J., *Jet Propulsion Laboratory (USA)*.
- Next Generation Remote Agent Planner, 363
 Jonsson A.K., *National Aeronautics and Space Administration, (USA)*. Muscettola N., *Recom Technologies (USA)*
 Morris P.H., Rajan K., *Caelum Research Corporation (USA)*

Demonstrations

Session Chairman: M. Van Winnendael, *European Space Agency*

- Micro Planetary Rover *Micro 5*, 373
 Kubota T., Nakatani I., *Institute of Space and Astronautical Science (Japan)*,
 Kuroda Y., *Meiji University (Japan)*, Kunii Y., *Chuo University (Japan)*.
- Active Surface Imaging System ASIS, 379
 Flatscher R., *Dornier Satellitensysteme (Germany)*, Ulbrich G-J., *European Space Agency*,
 Ulrich A., *Riegl Laser Measurement Systems (Austria)*, Paar G., *Joanneum Research (Austria)*.
- Demonstration of the Planetary Utilisation Testbed, 385
 Van Winnendael M., *European Space Agency*.

Control Experiments on ETS-7

Session Chairman: K. Tsuchiya, *Kyoto University (Japan)*

- ETS-7 Space Robot Teleoperation Through Virtual Force Reflection, 389
 Peñín L.F., Matsumoto K., Wakabayashi S., *National Aerospace Laboratory (Japan)*.
- A Wire Handling Experiment using a Teleoperated Advanced Robotic Hand on ETS-VII, 397
 Matsuhira N., Asakura M., Shinomiya Y., *Toshiba Corporation (Japan)*, Machida K., Tanie K. *Ministry of Trade and Industry (Japan)*, Nishida H., *Fujitsu Ltd. (Japan)*, Bamba H., *Aoba Sangyo Co. Akita K., IUSEF (Japan)*.
- Generalised Visual Aid for Direct Teleoperation Applied to ETS-7 Truss Manipulation Experiment, 403
 Wakabayashi S., Matsumoto K., *National Aerospace Laboratory (Japan)*.
- Vision-Based Robotics Control Experiment on ETS-VII, 409
 Vergauwen M., Koch R., Tuytelaars T., Van Gool L., *Catholic University of Leuven (Belgium)*.
- Reactionless Manipulations and Proposal to ETS-VII On-Board Experiments, 415
 Yoshida K., *Tohoku University (Japan)*, Nenchev D.N., *Hirosaki University (Japan)*.

Rover Control (2): Navigation and Piloting

Session Chairman: I. Nakatani, *ISAS (Japan)*

- Command Generation for Planetary Rovers using Virtual Reality, 423
 Blackmon T.T., Smith D., *National Aeronautics and Space Administration, (USA)*. Neveu C., Allport C., Anderson C, Gupta V., *Caelum Research Corp. (USA)*, Nguyen L., Kline A. *Recom Technologies (USA)*.
- Rover Self Localization in Planetary-Like Environments, 433
 Lacroix S., Mallet A., Chatila R., *LAAS-CNRS (France)*. Gallo L., *Aerospatiale (France)*.
- 3-D Localization for a Mars Rover Prototype, 441
 Roumeliotis S.I., Bekey A., *University of Southern California (USA)*.
- Stored Image-Based Map Building and Navigation for Planetary Rovers, 449
 Nakasuka S., Yamamoto H., Tanaka A., *University of Tokyo (Japan)*.
- Long Range Navigation for Mars Rovers Using Sensor-Based Path Planning and Visual Localisation, 455
 Laubach S.L., Olson C.F., Burdick J.W., Hayati S., *Jet Propulsion Laboratory (USA)*.

Intelligent Planning and Control of Spacecraft Operations

Session Chairman: N. Battaille, *French Space Agency (France)*

- Efficiency and Fairness when Sharing the Use of a Satellite, 465
 Bataille N., *CNES (France)*, Lemaitre M., Verfaillie G., *ONERA-CERT (France)*.
- Fuzzy Logic for Spacecraft Control: A European Approach, 471
 Ortega G., *European Space Agency*, Mulder J.A., Verbruggen H., *Technical University of Delft (NL)*.
- INTELMOD: Artificial Intelligence in Support of Mission Operation Tasks, 477
 Donati A., *European Space Agency* Romani E., *Dataspazio (Italy)*, Aynsley M., *Science Systems Ltd. (UK)*.
- Anomaly Detective Ground Support System for Mars Probe NOZOMI, 483
 Hashimoto M., *Institute of Space and Astronautical Science (Japan)*, Nishigori N., *Systems Engineering Group (Japan)*. Mizutani M., *Fujitsu Social Systems Engineering Ltd. (Japan)*.
- A New Design Approach of Software Architecture for an Autonomous Observation Satellite, 491
 Gout J., Fleury S., *LAAS-CNRS (France)*, Schindler H., *Matra Marconi Space (France)*.

Robot Operations Preparation and Commanding

Session Chairman: S. Nakasuka, *University of Tokyo (Japan)*

- Mission Preparation and Training Facility for the European Robotic Arm (ERA), 501
 Pronk Z., Schoonmade M., *National Aerospace Laboratory - NLR (Netherlands)*, Baig W., *Spacebel Informatique (Belgium)*.
- A Universal Task-Level Ground Control and Programming System for Space Robot Applications, 507
 Brunner B., Landzettel K., Schreiber G., Steinmetz B.M., Hirzinger G., *German Aerospace Centre - DLR (Germany)*.
- Projective Virtual Reality: A Novel Paradigm for the Commanding and Supervision of Robots and Automation Components in Space, 515
 Freund E., Rossmann J., *Institute of Robotics Research, IRF (Germany)*.
- A System Integrating High and Low Level Planning of Complex Tasks with a 3-D Visualiser, 521
 Finzi A., Pirri F., Schaerf M., *University of Rome 'La Sapienza' (Italy)*.

Systems Aspects of Space Manipulation

Session Chairman: K. Yoshida, *Tohoku University (Japan)*

- Safety Approach of Japanese Space Manipulation System, 531
 Tatsuo M., Satoh N., Satoh T., Hisatome Y., Doi S., *National Space Development Agency (Japan)*, Kuwao F., *Toshiba (Japan)*.
- Knowledge Representation and Reasoning for Fault Identification in a Space Robot Arm, 539
 Portinale L., *Università del Piemonte Orientale (Italy)*, Torasso P., Correndo G., *University of Turin (Italy)*.

- Research and Development of Reconfigurable Brachiating Space Robots, 547
 Ohkami Y., Hayashi R., Yamamoto H., Matunaga S., *Tokyo Institute of Technology (Japan)*.
- Tele-Science by Planetary Rover *Micro 5*, 553
 Kunii Y., Otsuka M., *Chuo University (Japan)*, Suhara M., Kuroda Y., *Meiji University (Japan)*,
 Kubota T., *Institute of Space and Astronomical Science (Japan)*.

Planning and Scheduling of Space Operations (2)

Session Chairman: K. Rajan, *National Aeronautics and Space Administration (USA)*

- Organisational Learning Agents for Task Scheduling in Space Crew and Robot Operations, 561
 Takadama K., Watabe M., Shimohara K., *ATR Human Information Processing Research Laboratories (Japan)*, Kasahara, H., *Nara Institute of Science and Technology (Japan)*
 Huang L., *Japan Institute of Science and Technology (Japan)*
 Ii H., Nakasuka S., *University of Tokyo (Japan)*.
- O-OSCAR: A Flexible Object-Oriented Architecture for Schedule Management in Space Applications, 569
 Cesta A., Oddi A., Susi A., *National Research Council CNR (Italy)*
- Distributed Planning in Constellations of Autonomous Spacecraft, 575
 Grant T.J., Broeils A., *Origin Nederland b.v. (Netherlands)*.

Space Robot System Verification

Session Chairman: Y. Wakabayashi, *National Space Development Agency (Japan)*

- Hardware-in-the-Loop Simulation, of Robots Performing Contact Tasks 583
 Aghili F., Dupuis E., Piedboeuf J.-C., Carufel J. de, *Canadian Space Agency (Canada)*.
- Experimental Validation of Contact Dynamics Simulation of Constrained Robotic Tasks, 589
 Van Vliet J., Sharf I., *University of Victoria (Canada)*,
 Ma O., *McDonlad Dettwiler Space and Advance Robotics Ltd. (Canada)*.
- An End-to-End Solution for Robot Workcell Calibration and Performance Assessment, 597
 Didot F., *European Space Agency, ESTEC*, Galardini D., *Trasys (Belgium)*,
 Geuens P., *Krypton (Belgium)*.
- Experiments of a Space Robot in the Free-Fall Environment. 601
 Watanabe Y., *National Space Development Agency (Japan)*,
 Nakamura Y., *University of Tokyo (Japan)*.

Robotics for Small Body Exploration

Session Chairman: S. Hayati, *Jet Propulsion Laboratory (USA)*

- Autonomous Landing and Smart Anchoring for In-Situ Exploration of Small Bodies, 609
 Ghavimi A.R., Serricchio F, Dolgin B., Hadaegh F.Y., *Jet Propulsion Laboratory (USA)*.
- Autonomous Landing System for MUSES-C Sample Return Mission, 615
 Kubota T., Sawai S., Misu T., Hashimoto T., Kawaguchi J., Fujiwara A.,
Institute of Space and Astronautical Science (Japan).
- Design of Contact Compliance and Simulation of Touch-Down Sequence of MUSES-C Satellite for Asteroid Sampling, 621
 Yoshida K., *Tohoku University (Japan)*.
- Precise Image-Based Motion Estimation for Autonomous Small Body Exploration, 627
 Johnson A.E., Mathies L.H., *Jet Propulsion Laboratory (USA)*.

Advanced Space Robot Technologies

Session Chairman: S. Matsunaga, *Tokyo Institute of Technology (Japan)*

- A Dexterous Gripper for Space Robotics, 637
 Bonivento C., Melchiorri C., Vassura G., *University of Bologna (Italy)*, Ferretti G., Maffezoni C., Magnani G.,
Polytechnic of Milan (Italy), G. Beccari., Caselli S., Zanichelli F., *University of Parma (Italy)*.
- On-Board Perception Processing for Space Robots, 643
 Wakabayashi Y., Miyata M., *National Space Development Agency (Japan)*,
 Nishimaki K., *AES Corporation (Japan)*.
- A Buyer's Guide to Forward Intersection for Binocular Robot Vision, 649
 Gruenfelder S., *Austrian Aerospace GmbH (Austria)*., Krickl R., *Vienna University of Technology (Austria)*.
- A Trajectory and Force Control of a Manipulator with Elastic Links, 655
 Tsujita K., Tsuchiya K., Kawano Y., *Kyoto University (Japan)*.
- Spacecraft 3-Axis Attitude Control by Space Robot Motion, 663
 Tsuda S-I., Aoki H., *Tokai University (Japan)*.

Exhibits

- Genetic Algorithms used to Determine WSB Trajectories for the Lunarsat Mission, 671
 Ockels W.J., *European Space Agency, ESTEC and Delft University of Technology (Netherlands)*,
 Biesbroek R., *GA Consultant (Netherlands)*.
- Force Simulation in Telerobotic System with Large Time Delay, 675
 Zhuang Jun., Sun Zenqi., Cheng Peng., *Tsinghua University (P.R. China)*.
- Limitations of Hardware-in-the-Loop Simulations of Space Robotics Dynamics using Industrial Robots, 681
 Krenn R., Schaefer B., *German Aerospace Centre, DLR (Germany)*.
- JERRY: A System for the Automatic Generation and Execution of Plans for Robotic Devices: 687
 The Case Study of the SPIDER Arm,
 Cesta A., Riccucci P., *National Research Council, CNR. (Italy)*, Daniele M., Traverso P., *IRST (Italy)*,
 Giunchiglia E., Piaggio M., *University of Genoa (Italy)*, Schaerf M. *University of Rome 'La Sapienza' (Italy)*.
- An Artificial High-Level Vision Agent for the Interpretation of the Operations of a Robotic Arm, 693
 Chella A., Gaglio S., Guarino D., Infantino I., *University of Palermo (Italy)*.
- The Jumping Tortoise: A Robot Design for Locomotion on Micro-Gravity Surface, 699
 Yoshida K., *Tohoku University (Japan)*.
- Smart, Simple and Low-Cost Control of Planetary Exploration Rovers 705
 Biesbroek R.G.J., Matthyssen A.Y.J., *JAQUAR Space Engineering (Netherlands)*
- SYMOFROS: A flexible Dynamics Modelling Software 709
 Piedboef J.C., Doyon M., Langlois P., L'Archevêque R. *Canadian Space Agency (Canada)*

List of Participants

719

Technology Surveys

AUTONOMOUS ROVER TECHNOLOGY FOR MARS SAMPLE RETURN

Charles R. Weisbin, Guillermo Rodriguez, Paul S. Schenker,
 Hari Das, Samad A. Hayati, Eric T. Baumgartner, Mark Maimone,
 Issa A. Nesnas, Richard A. Volpe
 Jet Propulsion Laboratory, California Institute of Technology
 4800 Oak Grove Drive, Pasadena, California

Planetary rovers enable good sample selection and retrieval for Mars sample return missions. After landing, the rovers search for the best possible scientific samples in the region around a lander, and they return these selected samples to an ascent vehicle that launches the samples into Mars orbit. To streamline the search for, the acquisition, and the retrieval of samples, rover autonomy is a critical technology. This paper summarizes a series of experimental results in the evaluation and demonstration of planetary rover autonomy, with a particular emphasis on rover system technology capabilities under development for a 2005 Mars sample return mission and its precursor missions.

1. INTRODUCTION

An autonomous system is defined here as one that can execute multiple-command sequences robustly at the remote rover site, without direct intervention by ground controllers while the sequence is being executed. The longer and more complex the task that a given rover can reliably execute by itself, the more autonomous the rover is. Execution of complex tasks with minimal ground control is essential to maximize science return for fixed mission duration, and to compensate for the long time that it takes for commands from Earth to reach their destination on the Martian surface. Four specific autonomous rover operational tasks are described in detail:

Mars Sample Return Rover Operations: Using a newly developed Field Integrated Design and Operations Rover (FIDO) rover, a complete "loop" was demonstrated of remote science panoramic imaging and target selection, autonomous navigation, *in situ* sample observation and analysis, and robotic sample coring and extraction functions. This work, under the leadership of P. S. Schenker and E. Baumgartner of JPL, was performed in the Mojave Desert at Silver Lake, CA, an ancient site replicating Mars-like geological features, mineralogy, and terrains. Field science operations were under direction of Mars'03/'05 Co-I Ray Arvidson (Washington Univ., St. Louis, MO) and PI Steve Squyres (Cornell Univ., Ithaca, NY), providing technology verification and Mission planning insight for the future Mars exploration.

Long Range Traverse and Science Acquisition: This task, under the leadership of R. Volpe and S. Hayati of JPL, involves long distance traverse from one site to another while operating in relatively rocky terrain, under

the constraint that the traverse must be executed autonomously with a single command cycle. The sequence includes the deployment of multiple instruments at various steps in the sequence, as well as the acquisition of multiple scientific samples. The experiments investigate the limits of what a planetary rover can do by itself, using technologies in on-board sensing, sequence planning, hazard avoidance, and goal confirmation.

Autonomous Small-Rock Pick-Up: This task, under the leadership of H. Das of JPL, illustrates the autonomous acquisition of small rocks, using visual and touch sensors and a rover-mounted micro-arm to achieve the rock pick-up operation. This type of autonomous operation is useful in streamlining the sequence of actions that the rover and its manipulator must take in response to detection of an interesting rock sample that needs to be picked up.

Two-Vehicle Surface Rendezvous & Sample Transfer: This sequence demonstrates, in a terrestrial analog experimental scenario, a terminal guidance, approach and rendezvous task that one autonomous rover executes as it comes close to another rover; and as a sample container is transferred from one vehicle to the other. This sequence illustrates techniques in visual target acquisition, on-board sequence planning, and terminal rendezvous operations. This work was conducted under the leadership of P. S. Schenker of JPL.

These four rover tasks illustrate experiments and tests with a variety of flight-like rover technology prototypes that are being developed at JPL as precursors to a sample return rover flight system. The autonomous system capabilities for each of these rover sequences are summarized in the Tables 1 and 2, together with the

major challenges that have been addressed in each of the sequences.

The terrestrial analog experimental scenarios illustrate the currently demonstrated autonomous rover technology, and the challenges in development and experimentation as outlined in more detail below.

2. FIELD INTEGRATED DESIGN AND OPERATIONS (FIDO) ROVER

Recently, the newest JPL rover vehicle was taken for a few practice trials around an ancient lake bed in the Mojave Desert. This next-generation Mars rover is helping NASA scientists and engineers learn more about driving the real thing on Mars. Future robotic rovers on Mars will need to find the best rocks to bring back to Earth, samples that are likely to contain the evidence scientists need to prove that life once existed on the red planet.

To find the best sample, scientists need a good retriever. The FIDO -- Field Integrated Design and Operations -- is helping them figure out how to use the kinds of instruments the next Mars rovers will need to fetch the most scientifically interesting rocks. FIDO [1,2] is designed to test the advanced technology of the Athena flight rover and science payload that will be launched as part of NASA's Mars Sample Return missions in 2003 and 2005. FIDO was recently tested in full-scale terrestrial field simulations of the planned Mars '03 Sample Return mission (ref: NASA Mars Surveyor Program, *Athena* science rover payload). It demonstrated a complete "loop" of remote science panoramic imaging and target selection, autonomous navigation, *in situ* sample observation and analysis, and robotic sample coring and extraction functions. This work was performed in the Mojave Desert at Silver Lake, CA, an ancient site replicating Mars-like geological features, mineralogy, and terrain. Field science operations were under direction of Mars'03/'05 Co-I Ray Arvidson (Washington Univ., St. Louis, MO) and PI Steve Squyres (Cornell Univ., Ithaca, NY), providing technology verification and Mission planning insight for the future Mars exploration.

No place on Earth is like Mars, but the field site on an ancient lake bed in the Mojave Desert comes close. The intent is to practice looking for rocks that contain carbonate minerals. If those kinds of rocks on Mars are found, it may tell us if the early planet had a carbon dioxide atmosphere. The rover has shown that it can find good rocks, drill samples out of them, and take the samples back to a lander.

FIDO's advanced technology includes the ability to navigate over distances on its own and avoid natural obstacles without receiving directions from a controller. The rover also uses a robot arm to manipulate science instruments and it has a new mini-corer or drill to extract and cache rock samples. There are also several camera systems onboard that allow the rover to collect science and navigation images by remote-control.

FIDO is about six times the size of Mars Pathfinder's Sojourner and is far more capable of performing its job without frequent human help. FIDO navigates continuously using on-board computer vision and autonomous control, and has similar capabilities for eye-to-hand coordination of its robotic science arm and mast. The rover has six wheels that are all independently steered and can drive forward or backward allowing FIDO to turn or back up with the use of its rear-mounted cameras.

In addition to testing FIDO, the scientists and engineers engaged students from four schools around the country in designing and carrying out their own mission with the rover. This is the first time students have been able to remotely operate a NASA/JPL rover. The students, from Los Angeles, Phoenix, Ithaca, NY, and St. Louis, (LAPIS), formed an integrated mission team responsible for planning, conducting and archiving a two-day mission using FIDO.

The FIDO rover shown in Figure 1 has a mass slightly greater than 60 kg. It has six wheels, and its dimensions are 100 cm in length, 80 cm in width, and 50 cm in height. It is a high mobility, multi-km range science vehicle, developed recently as an advance technology prototype for actual future NASA missions. It carries a mast-mounted multi-spectral stereo panoramic camera, a bore-sighted IR point spectrometer, a robot arm with an attached micro-camera and a Moessbauer Spectrometer, as well as a body-mounted rock sampling mini-corer.

The FIDO Mobility Sub-System consists of a 6-wheel rocker-bogie chassis. Each wheel is independently driven and steered with a 35 N-m torque/wheel at stall. Its flight-related actuator design provides a speed of < 9 cm/sec with 20 cm diameter wheels. Its ground clearance is 23 cm. The vehicle carries a 4 degrees-of-freedom mast with integral science instrumentation, and an instrumented science arm with four degrees of freedom and an actuated gripper/scoop.

Autonomous rover navigation and control are enabled by an on-board computational consisting of a PC104 80586, 133 MHz CPU. The vehicle has front/rear stereo camera pairs with a 120° field of view for hazard avoidance. An

inertial navigation system with a CCD-based sun sensor provides positioning information. Differential GPS is used for ground-truth in field trial applications. On-board science instrumentation includes a mast-mounted multi-spectral (650, 740, 855 nm) high resolution stereo camera pair with 10 degrees field of view. A colinear pair of B/W stereo cameras is used for long-range navigation. The vehicle also carries a mast-mounted point spectrometer operating at 1.25 - 2.50 microns. An arm-mounted color imager, and Raman and Mossbauer spectrometers are also part of the on-board instrumentation. A body-mounted mini-corer is used for sampling and a caching subsystem retrieves and stores samples. A belly camera is used for observation



Fig. 1: FIDO

Notable outcomes of the just-completed FIDO rover Desert included multiple autonomous traverse maneuvers to science targets, using new hazard detection and avoidance software over a distance of about 100 meters under continuous traverse. Increased dead-reckoning was achieved with new wheel-velocity synchronization techniques.

A "Presidential" multi-spectral panorama was constructed from approximately 1800 single images from a Panoramic Camera, totaling over 400MB in data return. Over 600 Navigation Camera images and measurements were taken with the near-IR point spectrometer. The Moessbauer spectrometer was utilized during overnight operations to analyze the field rock samples.

The Mini-Corer was deployed 3 times with successful cores acquired during 2 of the deployments. The only unsuccessful deployment occurred on a very hard rock, which the mini-corer drilled successfully. However, the core sample itself did not remain intact after it was broken off, and the core sample settled in pieces at the bottom of the hole.

Athena flight mission scientists directed the entire rover field mission, in cooperation with FIDO rover/instrument advanced technology engineers. FIDO operations lasted between 12 and 20 hours per day, weather permitting. On days when Moessbauer measurements were taken, the rover was on for about 20 hours, including the overnight hours. There were only a few hours of down time before start of the next day's activities.

3. LONG RANGE TRAVERSE AND SAMPLE ACQUISITION

Concurrently with the experimental trials conducted with the new FIDO vehicle, there are technology developments and experiments with a range of component and system technologies in robust navigation and position determination. These tests are being conducted with the Rocky 7 experimental vehicle shown in Fig. 2.



Fig. 2: Rocky 7

Experimental Demonstration Scenario

To provide the capability of single-day long-range traverses across Mars, the rover system must be capable of quickly and autonomously navigating through an obstacle laden terrain. While short range traverses can be planned by operators on the ground using panoramic stereo images, longer range traverses must be specified with incomplete knowledge of the terrain to be encountered by the rover. For instance, overhead imagery may not exist or be at too low of a resolution to identify obstacles, while ground-based panoramic imagery (beyond the stereo ranging capability) will not provide the distance to obstacles and is subject to terrain occlusions. Further, position error accumulated by the rover during the longer traverses must be minimized if the desired goal is to be attained. Finally, optimization of onboard processing, as well as processing of information in parallel with vehicle actions, is needed to increase the effective speed of the vehicle and thereby go longer distances between communication cycles with ground controllers.

Technical Issues Addressed in Demonstration

The nature of this scenario is such that advances in several technologies must be made and demonstrated at once. First, if the rover is to move beyond the range of stereo imagery where operators can safely specify paths, it is a logical step that new stereo images will be acquired at the border of the known and unknown regions and these images may be used for planning. Since the objective requires traversal of this new terrain without operator interaction, the analysis providing specification of safe routes must be done autonomously on-board. To this end, we have developed a new path planning algorithm which distills the natural terrain imagery into a map of geometric obstacles, and quickly plans a local optimal path through them [3]. To be consistent with ground controller specified paths, this path generation produces a similar set of waypoints which are passed to the system's lower level piloting algorithms for motion between them.

However, safe traversal is only part of the problem, since the rover must maintain accurate knowledge of its position relative to its start point, if it is to accurately achieve the global goal specified by ground controllers [4]. Two separate techniques are employed to add robustness and increase fidelity. First, proprioceptive sensory information processing has been developed to use onboard sensors to determine a continuous estimate of position and heading, as well as the error bounds on those estimates [5]. Second, correlation of the changing relative positions of fixed landmarks in the surrounding

terrain is used to visually provide an estimate of changes in the vehicles position and heading [6]. The two techniques are also complementary in their implementation, since the former is typically used while the vehicle is traversing between observation and planning locations along the full traverse, while the latter is used at these locations.

Finally, to increase the distance that may successfully be traversed between communication cycles with Earth, we have improve two capabilities which increase the average rover speed. First, on-board stereo processing has been optimized to run over five times faster (while using less memory). Second, we have implemented a continuous driving strategy, whereby, the rover does not need to stop while processing obstacle avoidance imagery [7]. Instead, continuous driving and steering are done in parallel with the sensing, until the intermediate goal locations are achieved and the rover must stop to plan a new path with a corresponding set of waypoints.

Pivotal Steps in Demonstration Sequence

STEP 1: The Rover Autonomously Plans a Path Through Local Terrain. In this step, the robot receives the goal location, raises the camera mast, and takes stereo images of terrain immediately before itself but in the general direction of the far away goal. Images are processed and path is planned as a series of intermediate waypoints out to the edge of the valid stereo data of about 10 meters. To do this the operator issues a single command with a distant goal point of about 100 meters. This is done in a completely autonomous mode.

STEP 2: Autonomous Sequential Drive to Intermediate Waypoints. Rover begins driving directly to the first intermediate waypoint. Along the way, stereo images are capture by body mounted wide-angle cameras, and analyzed for previously undetected obstacles on the path. If hazards are found, the direct path is abandoned, and a behavior control algorithm is employed to attempt to navigate to the next waypoint. If distance to the goal does not decline, then the sequence of waypoints is abandoned, the mast raised, new images are taken, and a new path is planned. The on-board executive provides the necessary sequences, in a completely autonomous mode.

STEP 3: Autonomous Position Determination. During driving, the rover monitors odometry, sun sensor, and accelerometers for tilt of the sun sensor, in order to determine position and heading. Separate tests have also included gyro data and a full kinematic model of the rover in an extended Kaman filter [5]. These techniques provide a position estimate that allows the rover to determine its progress in achieving the geometrically

specified waypoints. After the estimated position matches the desired position, the rover raises its mast to image the traversed terrain and compare the topography from this new vantage point with that obtained from the previous position. This comparison yields a more accurate estimate which replaces the prior one. The position estimation operation is fully autonomous, with the on-board sequences provided by the on-board executive.

While all of these actions are autonomous, there are continuing improvements in the robustness and precision of all actions. For instance, all path-planning for this demonstration was conducted in a single view provided by the mast stereo cameras. In denser terrain, a clear path may not exist in only one view. Therefore, this technique is being extended to use a map obtained from a mosaic of images.

Relationship to Other Work

Mars rover research is unique in its emphasis of small vehicles navigating through rough natural terrain. However, there are broad similarities with other mobile robot research. For instance, the path planner described here operates on a geometric map extracted from stereo imagery of the rough terrain. While the vehicle is still executing the traverse in the rough terrain, the path planner is applicable to a more structured environment problem such as indoor navigation [8]. Similarly, while the estimation work is made crucial by the natural terrain driving, position estimation is a problem common to all moving robots [9]. The crucial difference here is the lack of GPS or a planetary magnetic field to greatly aid the process.

4. ROVER- BASED SMALL ROCK ACQUISITION

Experimental Demonstration Scenario

The objective of this effort was to demonstrate, in a relevant ground environment, the acquisition of a small rock by a rover and manipulator arm from 1 meter away in a single operator command cycle. The task scenario was to have the operator select a sample to be acquired and indicate it with a mouse click on an image from the rover. The rover would then autonomously approach the specified target and deploy its sampling arm to pick up the target. The Rocky 7 rover [10] was used as the platform for this demonstration. Images from rover-mounted stereo black and white (B&W) wide-angle (120°) cameras was assumed to be available as was a sampling arm capable of picking up small rocks from the ground. The on-board computing for the demonstration was performed with a Motorola 68060 based VME

board. The reduced computational resources available on Mars rovers will result in longer task execution times. Vision from the two B&W cameras on-board the rover was the primary means of sensing. In addition, odometry from wheel encoders (six-wheel driving and two-wheel steering) and joint angle sensors on the two degrees-of-freedom arm were used to accomplish the demonstration. These conditions and assumptions reflect a realistic scenario for a rover on Mars. New technology elements from this demonstration are relevant to NASA's Mars Surveyor Program missions involving the use of rovers to perform science exploration and sample return.

Technical Issues Addressed in the Demonstration

A detailed description of the procedure used in this demonstration is reported in [11]. The key technology element that enabled this demonstration was the small rock tracking algorithm developed in this effort. Upon operator specification of the small rock to be acquired, the rover control software determined the target location for the rover in order to pick up the rock and drive towards that location. It periodically updated the target location with the tracking algorithm and re-planned its approach to the target. Once at the target location, the sampling arm was deployed to pick up the target.

The tracking algorithm assumed that the target was a local maximum in elevation around the specified target location. In each update to the vision sensing, an estimate of the new location of the target was found using the vehicle odometry. Stereo camera images were taken and a range map was built in a small window around the estimated target location. The local maximum within the window was found and refined using an intensity threshold. The new target location was used to re-plan the vehicle approach. The vehicle was then driven to the new target. After driving a pre-set distance (20 cm in the experiments we have conducted), the cycle was repeated. This procedure continued until the target was within 1 cm of the target location. When that condition was met, the sampling arm was deployed to pick up the rock.

Supporting technology elements that enabled the demonstration included:

- A graphics user interface (GUI) that displayed an image from the rover and accepted the user input for target specification.
- Bilateral communication software to transfer images, target designation and debugging data between the operator interface and the rover.
- Stereo processing algorithms to generate range maps from stereo images.

- Vehicle trajectory planning and driving towards a target.
- Sampling arm deployment, sensing ground elevation and rock grip sensing.

Pivotal Steps in Demonstration Sequence

STEP 1: Rover Sends Image to Operator Station. The rover is initialized to begin the small rock acquisition procedure. It acquires an image pair and sends the left image to the operator station. From the initialized configuration, the rover automatically sends the image to the operator station over an Ethernet communication link. The communication of the image to the operator station is first half of the single command cycle used in this operation.

STEP 2: The Operator Specifies the Target Small Rock and an Intensity Threshold. This is the only operator input for the entire procedure. The rover waits for a reply from the operator after acquiring the image. The image is displayed on the operator station. The operator clicks on the desired target in the image and types in the threshold intensity. The operator makes a decision and issues a command to the rover.

STEP 3: The Image Plane Target Location and Intensity are Sent to the Rover. The rover receives the command from the operator. The command is sent using a communication protocol that is designed to transfer data and images over an Ethernet link. The return communication is the second half of the command cycle in this operation. No further communication between the operator and the rover is needed. All subsequent processing occurs onboard the rover.

STEP 4: Determine the 3-D Location of the Rock. The rover processes images to determine the rock location. To do this, it uses stereo processing and calibrated camera models. This operation is fully autonomous.

STEP 5: The Rover Drives towards the Target Rock. This is done by computing a rover trajectory to the target and driving towards it. The operation is also fully autonomous.

STEP 6: Update the Target Location Estimate. The rover polls the target tracking software to get an update to the target location. This is done periodically, every 10 cm, to acquire a new set of stereo images and odometry and compute a new estimate of the target location. This is the most difficult and critical step in the procedure. Loss of tracking of the target results in failure of the procedure. Complete autonomy is used in this step in the operation.

STEP 7: Rover Re-directs Itself towards New Target Location Estimate. The rover stops its previous motion, changes steering wheel angles and starts driving again. A new trajectory towards the target is computed and the rover steering is corrected to drive towards the new target location. This operation is done in full autonomy. This operation is repeated every 10 cm until the rover is within 1 cm of the target location.

STEP 8: Deploy the Sampling Arm to Pick Up the Small Rock. The sampling arm is un-stowed and driven to pick up the rock. The scoops on the arm are opened and driven vertically down. A resistance (large difference between commanded and actual shoulder joint position) is used as an indication that the ground is touched. The scoops are lifted up slightly and then closed until resistance is again felt. This is an indication that either the ground or the rock is felt. The arm is lifted up slightly and then the scoops are closed again. This is repeated until the scoops do not close any more, indicating that the rock is grasped. The arm is lifted up indicating that the procedure is completed. The level of autonomy according to the following definition:

Metrics for the Degree of Autonomy

It is of interest to evaluate in a quantitative manner the degree of autonomy in the operational sequence just outlined. To this end, consider the following two possible metrics:

- **Autonomy Metric #1:** The degree of autonomy in the overall sequence equals one minus the ratio of the number of operator interventions to the total number of sequence steps. Under this definition, the degree of autonomy is $1 - 1/9 = 0.89$
- **Autonomy Metric #2:** The degree of autonomy equals one minus the ratio of operator interventions with this method to the number of operator interventions with Sojourner. Sojourner is the rover that was deployed on Mars in 1997. It is estimated that Sojourner with its associated operations architecture, under similar environmental (terrain, etc) conditions, would take 3-5 command cycles to perform a small rock pick-up (if it had an arm that would allow it to do so) while it took 1 command cycle with the demonstration we describe. If we use an average of 4 command cycles as the estimate for Sojourner, the degree of autonomy is $1 - 1/4 = 0.75$.

While such performance metrics are imperfect, and do not take into account many important issues, they

nonetheless provide a coarse indication of the degree of autonomy that was achieved in the described operation.

Relationship to Other Work

The autonomous small rock pick-up procedure described here is presented with greater detail in [11]. It builds on previous work done at JPL and elsewhere. The platform used in this demonstration is the Rocky 7 rover [10] – a prototype for Mars science exploration. The stereo image-processing algorithm used to compute a range map from a stereo pair of images was also developed at JPL [12,13]. Related earlier work at JPL was the demonstration of small rock identification and pick-up with off-board computation and imaging [2]. There has also been much work done elsewhere on related problems. Work at the NASA Ames Research Center [15] developed a visual servoing algorithm and applied it to the Marsokhod rover. Their approach relies on tracking texture in successive images and requires a continuous view of the target. A demonstration of autonomous grasping rocks within the workspace of a robot arm from a stationary platform has also been reported [14] using overhead gimbale cameras. In contrast, the development reported here uses realistic assumptions on the resources available and configurations to be used on future Mars rovers.

5. TWO-ROVER SURFACE RENDEZVOUS AND SAMPLE TRANSFER

The SAMPLE RETURN ROVER (SRR) is a novel 10 kg-class vehicle, 88 cm in length, 55 cm in width, and 36 cm in height. It has four wheels, and is a hybrid composite-metal vehicle for rapid (10-30 cm/sec) autonomous location, rendezvous, and retrieval of collected samples under integrated visual and beacon guidance. SRR collapses to less than one third its deployed field volume, and carries a powerful, visually-servoed all-composite manipulator. The rover was designed as an approach to the “sample cache grab” problem. For this problem, the function is to quickly, robustly, and autonomously, go from a landed spacecraft, find a nearby sample cache, and retrieve the cache to a Mars Ascent Vehicle containment. The operational horizon of the sample grab can in principle be small, given that techniques are currently under development to allow precision landing within as little as a hundred meters. Thus, SRR could possibly communicate remote via the lander link, versus a rover up-link to orbiter or direct-to-earth link, and be under visual observation and reference by the lander stereo cameras.

The operational model we have developed is broader. We assume that SRR may need to transit over the

horizon, start with approximate knowledge of the cache site (referenced to the lander surface coordinates), provide its own on-board visual sensing, and maintain an accurate state estimate of vehicle location. During its cross-terrain transits, SRR must in real time capture and analyze a visual terrain map, and must detect and avoid hazards. As it approaches the target cache site, SRR must detect the presence of the cache, accurately localize in relative distance and orientation, and take a suitable inbound heading. Finally, once this “terminal guidance” phase into the cache is completed, SRR must, in a cooperative robotic workspace, visually maneuver its arm to acquire and transfer the cache. See an example of this operation in the lower left inset in the figure. As a system design, SRR is optimized to the short-range sample cache grab operation. However, the technology functions it embeds are quite general and include: continuous, high speed autonomous driving; visual detection, tracking and precision terminal guidance; and accurate visual manipulation from a mobile platform.

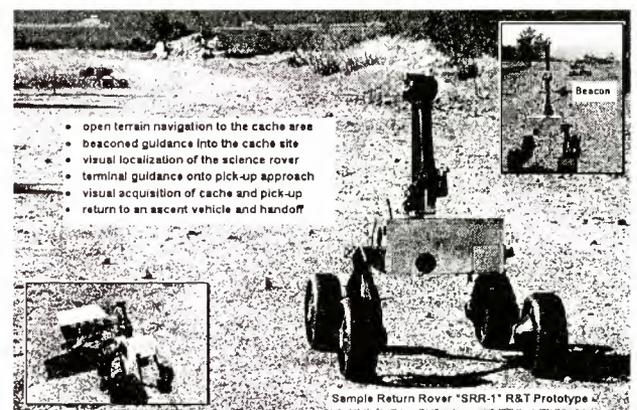


Fig. 3: Sample Return Rover “SRR-1”

The major characteristics of the vehicle are summarized below:

SRR Mobility: The rover has 4 wheels, all actuated with DC brushed/Maxon RE025 motors. It has skid steering on variable footprint and strut angles that is being upgraded to four-wheel steer-ability. The chassis is an articulated rocker/shoulder assembly with active control. It has 20 cm diameter deployable rigid wheels that are volume efficient, and it has self-deployable strut hinges with a ratio of 1:3 volume stowage.

Mass & Volume: The rover has a 7.2 kg mass that includes <5.0 kg for the baseline mechanism, 1.0 kg for the active rocker, 1.0+ kg for the wheels. When stowed, its dimensions are 44 cm in length, 55 cm in width, and 22 cm in height. Upon deployment, the vehicle has dimensions of 85 cm in length, 55 cm in width, and 36 cm in height.

Manipulation: The rover is equipped with a "micro-arm" consisting of 3 degrees-of-freedom with actuated gripping end-effector. The 2.0 kg arm has 0.7 meter total reach for cache acquisition and transfer functions.

Navigation Sensors: A forward-looking stereo camera pair (120 degree FOV) is used for obstacle detection and sample cache localization. A spot pushbroom laser/CCD active terrain sensor is used for fast terrain mapping and hazard detection. A manipulator-mounted goal camera (20 degree FOV) is used for long-range detection of science rover/lander. A tri-axial accelerometer package provides tilt sensing and a single-axis gyroscope for heading determination

Computing Platform: A PC104 80486 processor runs at 66 Mhz and is currently being upgraded to a 233-300MHz class Pentium class CPU. The operating system is VxWorks 5.3 (Tornado). There are 16-32 MB RAM onboard, with flash-drive enabling turn-key boot. Software modules are ANSI-C specification, in a layered environment that is readily modified in field.

The recent development and technology functions of SRR are described in [1,2], and references therein. The major advances are in the areas of autonomous visual detection of both natural and artifactual structures, visual tracking with respect to the same, and incorporation of this state knowledge in local sensor fused navigation estimates and robust visual localization and continuous terminal guidance to targets of interest. In particular, sample cache repositories or lander transfer sites. In summary, this progress includes:

- **Open terrain navigation:** high-speed reactive autonomous driving, utilizing integrated laser spot pushbroom and passive stereo (ref.: Rocky7, with occupancy grid analysis) 3D sensing modes; operated with pseudo-PM beacon bearing guidance (<1-2 m cross-track error over 100 m) for continuous motion up to 15 cm/sec.
- **Visual search-localization:** wavelet-based detection of science rover (or cache) at 1-to-20 m (<.15 - 5° error); visual search, range-pose estimation, and continuous guidance (VTG) on marked science rover in 10 -> 3 meter approach (< 2% average range/heading error; accommodates 3D rel. inclination and side views).
- **Cache recognition:** generalized image template matching technique ("eigen-face" model with K-L representation & Fourier domain implementation)

for rapid, accurate localization (1-2 cm) of target in well-calibrated stereo workspace; conception and initial simulations of hybrid image plane/stereo (HIPS) model to provide sub-centimeter accuracy for a priori poorly calibrated camera/manipulator model.

- **Visual registration/guidance:** real-time extraction and autonomous matching of 3D geometric features (multi-resolution; surface representations) to stored models; developed/applied for highly accurate registration of a sample return/transfer rover with cache site (~1 cm/1° error @1 m) and precision updates of SRR state (range, heading, slant/tilt) over variable course to cache/lander. Integrated/demonstrated latter as extensible Extended Kalman Filter (EKF) fusion framework (visual feature tracking and odometry as inputs, with <3% positioning error over 5-6 meters).

6. CONCLUDING REMARKS

This paper provides a brief overview of development in autonomous rovers at JPL. It emphasizes the detailed robotic tasks that have been made autonomous in terrestrial analog demonstrations. These tasks range from long traverse for exploration and survey, to autonomous acquisition of small rock samples, to the performance of autonomous surface rendezvous of two robotic vehicles followed by a sample transfer operation. These demonstrations constitute fundamental advances to the state-of-the-art in rover autonomy. They have also provided the technological underpinnings for much of the rover technology to be used in forthcoming sample return missions over the next decade.

7. REFERENCES

- S. Schenker, E. T. Baumgartner, R. A. Lindemann, H. Aghazarian, D. Q. Zhu, A. J. Ganino, L. F. Sword, M. S. Garrett, B. A. Kennedy, G. S. Hickey, A. S. Lai, L. H. Matthies; Jet Propulsion Lab.; B. D. Hoffman, Massachusetts Inst. Technology; T. L. Huntsberger, Univ. So. Carolina, "New planetary rovers for long range Mars science and sample return," in Intelligent Robotics and Computer Vision XVII, SPIE Proc. 3522, Boston, MA, Nov. 1-5, 1998 (*Invited*, 14 pages), and references therein.
- T. Baumgartner, P. S. Schenker, Jet Propulsion Lab.; B. D. Hoffman, Mass. Inst. Technology; T. E. Huntsberger; Univ. So. Carolina; P. C. Leger, Carnegie-Mellon Univ. "Sensor fused navigation and manipulation from a planetary rover," in Sensor Fusion and Decentralized Control in Robotic Systems (Eds., P. S. Schenker, G. T. McKee), SPIE Proc. 3523, November, 1998, Boston, MA; and B. Hoffman, E. Baumgartner, P. Schenker, and T. Huntsberger, "Improved Rover State Estimation in

- Challenging Terrain", in *Autonomous Robots*, February, 1999, and references therein. More information about FIDO is available at: <http://wundow.wustl.edu/rover>.
- Laubach and J. Burdick, "RoverBug: An Autonomous Path-Planner for Planetary Microrovers," Sixth International Symposium on Experimental Robotics (ISER'99), Sydney, Australia, March 1999.
 - Volpe, "Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype" *International Journal of Robotics Research*, Special Issue on Field and Service Robots. Accepted for publication.
 - Balaram, "Kinematic State Estimation for a Mars Rover." *Robotica*, Special Issue on Intelligent Autonomous Vehicles, Accepted for publication.
 - Olson, "Subpixel Localization and Uncertainty Estimation Using Occupancy Grids." *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit MI, 1999.
 - Martin-Alvarez, R. Volpe, S. Hayati, R. Petras, "Fuzzy Reactive Piloting for Continuous Driving of Long Range Autonomous Planetary Micro-Rovers." *Proceedings of the 1999 IEEE Aerospace Conference*, Aspen Colorado, March 6-13, 1999.
 - C. Latombe, "Robot Motion Planning." Kluwer Academic Press, 1991.
 - Durrant-Whyte, "Consistent Integration and Propagation of Disparate Sensor Observations." *Proceedings of the IEEE International Conference on Robotics and Automation*, April, 1986.
 - Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, R. Petras, S. Laubach, "The Rocky 7 rover: a Mars sciencecraft prototype" *Proc. International Conference on Robotics and Automation*, 1997. Volume: 3 , Page(s): 2458 –2464
 - Maimone, I. Nesnas, H. Das "Autonomous Rock Tracking and Acquisition from a Mars Rover," *Proc. 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space 1-3 June 1999 ESTEC*, Noordwijk, The Netherlands.
 - Matthies, "Stereo vision for planetary rovers: stochastic modeling to near real-time implementation," *IJCV*, 1992, Jul, Volume 8 1,pp 71-91, 1992.
 - Matthies, A. Kelly, T. Litwin and G. Tharp, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," *Robotics Research: the 7th International Symposium*, Springer-Verlag, 1996.
 - Theobald, W.J. Hong, A. Madhani, B. Hoffman, G. Niemeyer, L. Cadapan, J.J.-E. Slotine, J.K. Salisbury, "Autonomous Rock Acquisition," *Proc. AIAA Forum on Advanced Development in Space Robotics*, Madison, Wisconsin, August 1-2, 1996.
 - Wettergreen, H. Thomas, M. Bualat , "Initial Results from Vision-based Control of the Ames Marsokhod Rover," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Control of Wheeled Robots*, pp. 1377-1382, Grenoble, France, September 7-12, 1997.

ACKNOWLEDGEMENT

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA.

Rover Functions vs. Sequence Type	Long Range Traverse Sequence	Sample Return Sequence	Small Rock Pick Up Sequence	Two Vehicle Surface Rendezvous
Navigation	~100 meters per command	100 meters per command;cont	not critical; sequence starts ~1 meter away	not critical; sequence starts ~1 meter away
Image acquisition & object recognition	autonomous panoramic imaging in full autonomy	autonomous panoramic imaging in full autonomy	autonomous target imaging, localization and tracking every 10 cm	visual search & localize target rover at 1-20 m autonomous cache recognition
Sample selection	operator designated	scientist designated (03 athena simulations)	operator designated	autonomous cache localization
Sample acquisition & transfer	5 sample acquisition maneuvers per command	multi-instrument science; mini-corer; autonomous operations	1 command triggers autonomous vehicle driving to target, arm deployment and sample retrieval	autonomous inter-vehicle sample transfer ; move cache from one vehicle to another
Relative positioning, alignment & terminal guidance	autonomous visual localization; on-board, multisensor position estimation	see two-vehicle surface rendezvous sequence	continuous visual tracking toward selected rock target	visual registration and guidance (~1 cm/1 deg. error at 1 meter); relative tracking

Table 1: Rover Operations Enabled by Rover Autonomy

Rover Functions vs. Sequence Type	Long Range Traverse Sequence	Sample Return Sequence	Small Rock Pick Up Sequence	Two Vehicle Surface Rendezvous
Navigation	over the horizon navigation; multi-sensor fusion	build up of navigation errors; multi-sensor fusion	none; sequence starts at ~1m staging location	none; sequence starts at ~1m staging location
Image Acquisition & Object Recognition	deployment of mast camera; correlation of multiple images	deployment of mast camera; correlation of multiple images	getting accurate range maps & localizing rock target	real-time recognition of object features
Sample Selection	simple but effective operator interface	communications bandwidth and operator interface	high-resolution camera image displays	localization of cache from multiple images
Sample Acquisition & Transfer	concatenation of multiple rover commands	core from moving base; stability & robustness	sensing ground level, rock grasp and repeatable rock pick up	sample cache transfer maneuver
Relative Positioning, Alignment & Terminal Guidance	not applicable	see two vehicle surface rendezvous	continuity in tracking estimate between images	consistent relative position estimates

Table 2: Main Problems Overcome in Rover Autonomy

NASDA's activities in space robotics

Dr. Yoshiaki OHKAMI and Dr. Mitsushige ODA

National Space Development Agency of Japan (NASDA)

Abstract: Space robots are indispensable tools for future space missions such as building/operation of the international space station, realization of on-orbit satellite servicing (inspection, logistic support, repair, rescue from stranded orbit etc.), and lunar/planetary explorations. NASDA has various space robot related projects as follows.

- Retrieval of Japan's free flyer (SFU) by the shuttle's manipulator which was conducted by NASDA astronaut Wakata in January 1996 (STS-72).
- Manipulator Flight Demonstration (MFD) on the space shuttle which was conducted in August 1997 (STS-85).
- Space robot experiments which are being conducted on the Engineering Test Satellite7 (ETS-VII)
- A remote manipulator system (JEMRMS) for the Japanese experiment module of the international space station

This paper introduces these projects and shows the most up-to-date results of the ETS-VII robot experiments which are being conducted in space.

1. Retrieval of a free flyer (SFU) by the shuttle manipulator (Ref.1)

A team of Japan's national agencies (ISAS/NASDA/ MITI) developed and launched a free flyer named SFU in March, 1995. Mission of SFU was to conduct material and life science experiments in the micro-gravity environment. This free flyer was designed to be launched by NASDA's H-II rocket and to be retrieved by the space shuttle.

The retrieval mission was conducted in January 1996 and a NASDA astronaut, Mr. Wakata manipulated the shuttle manipulator in this mission. In this retrieval mission, SFU made a slight orbital descent from its mission orbit to the rendezvous orbit (472km alt.) and waited there. The space shuttle made the ground-up rendezvous and approached SFU from beneath of it (R-bar approach). When the shuttle's manipulator was positioned ahead of SFU, RCS of the shuttle was disabled and the shuttle manipulator grasped SFU. SFU maintained its attitude using reaction wheels while the space shuttle approached and grasped it by the manipulator. After the shuttle's manipulator grasped SFU at its grapple fixture,

the reaction wheels were run-downed and switched off. This mission gave us a lot of insight about the rendezvous docking and space robot operations.

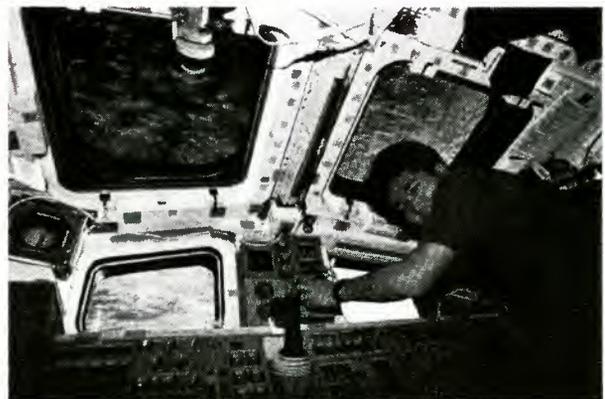
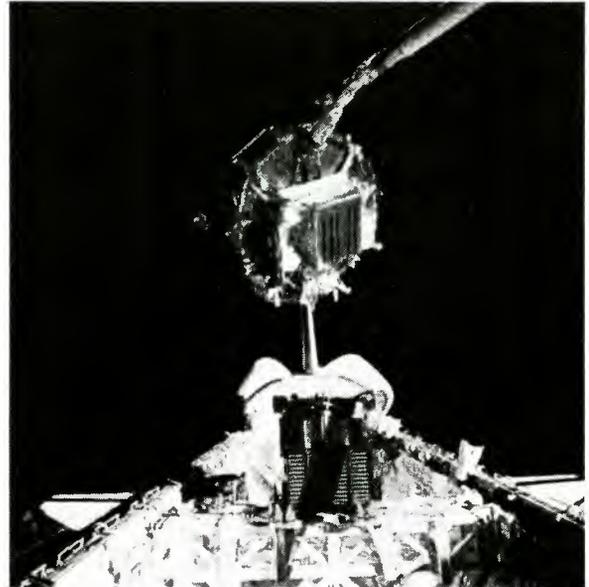


Fig.1 NASDA astronaut Wakata retrieved Japan's a free flyer(SFU) using the shuttle manipulator (STS-72 / 1996)

2. Manipulator Flight Demonstration (MFD) on the space shuttle (Ref.2)

NASDA is now developing the Japanese experiment module's manipulator system (JEMRMS) to be used on the international space station. In order to verify design of the JEM's manipulator and its operation system, NASDA conducted the manipulator flight demonstration (MFD) mission on the shuttle cargo bay in 1997 by STS-85.

A 6-dof robot arm of about 1.5m stretched length which is a replica of the JEMRMS small fine arm was mounted on the shuttle cargo bay using the Multi-Purpose Experiment Support Structure(MPESS) as shown in Fig.2a. The robot arm was operated from the control station in the aft flight deck of the shuttle as shown in Fig.2b. The control station includes a CCTV monitor, a pair of 3-dof joysticks, switch panel and other control electronics. Tasks conducted were handling (removal/install) of ORU(see Fig.2c), open/close of a hinged door. These contact operations were conducted using the compliance control capability of the MFD robot arm.

Beside these onboard operations by the shuttle astronauts, teleoperation experiments were conducted from the ground. Experiments were to control the robot arm using commands files which were generated on ground and sent to the shuttle during the mission. Visual inspection and other tasks which did not require the contact operations were conducted using this ground commanding.

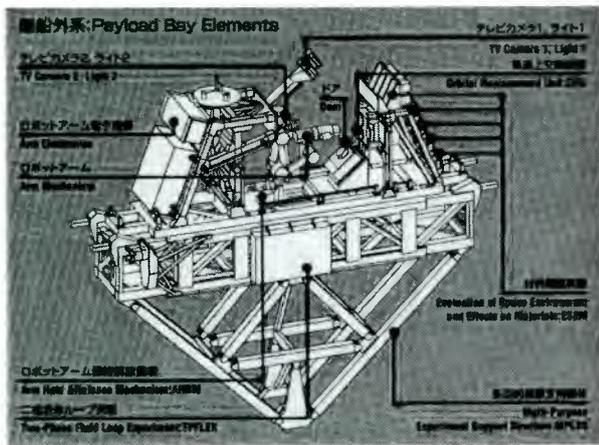


Fig2a MFD system

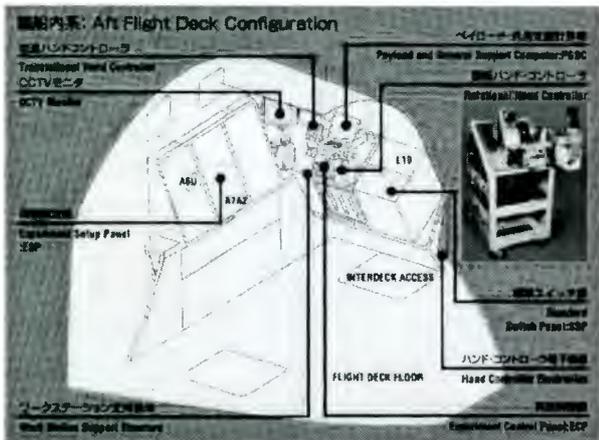


Fig.2b MFD operation system



Fig.2c MFD mission on STS-85 in 1997

3. Engineering Test Satellite 7 (ETS-VII)

The engineering test satellite No.7 (ETS-VII) was launched on Nov.28, 1997 to conduct the rendezvous docking and space robot technology experiments. ETS-VII consists of two satellites named the chaser and the target. Mass of the satellites are 2.5t and 0.4t respectively. Both satellites fly together for most of its mission life. The target satellite is released from the chaser satellite during the rendezvous docking experiments and is captured again at the end of each rendezvous docking experiments using the docking mechanism. The orbit of the satellites is 550km altitude and 35degrees inclination. Mission life of the satellite was 1.5 years after the launch. However, since the satellite is healthy after its original mission life, mission is extended to 2 years till end of Nov. 1999. Fig.3a shows in-orbit configuration of ETS-VII during the planned capture/berthing experiment which is to grasp the target satellite by the onboard robot arm. Fig.2b shows ETS-VII satellite on the H-II rocket. A folded down robot arm is mounted on the chaser satellite. The target satellite was mounted on top of the chaser satellite.



Fig.3a ETS-VII chaser(left) & target (right)

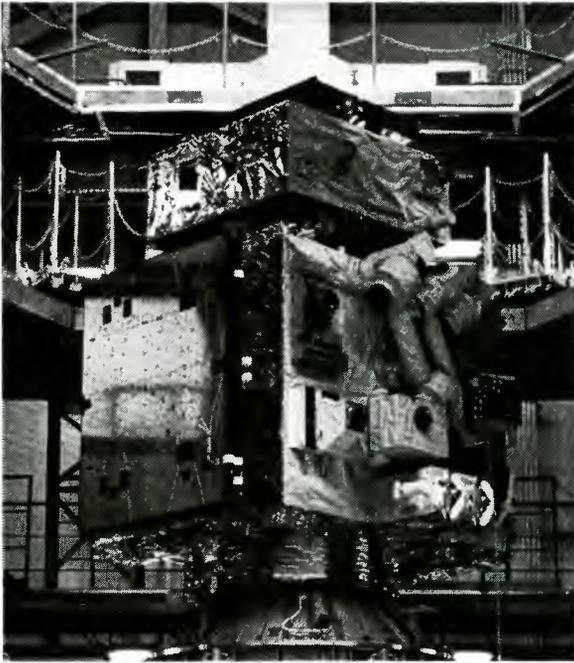


Fig.3b ETS-VII on top of the H-II rocket

3.1 Satellite operations

Satellite operations including experimental rendezvous docking and robot operations are conducted from NASDA's Tsukuba space center. Communication between the ETS-VII satellite and the on-ground control station is established using a data relay satellite (NASA's TDRS) in the geo-stationary orbit. The overall ETS-VII experiment system is shown in Fig.4. Since this communication network is realized by a lot of computers onboard the satellites and on-ground, a time delay of about 6 to 8 seconds in return existed in the robot control loop. Fluctuation of the time delay of up to 2 seconds is observed. This fluctuation is absorbed by the data buffer in the onboard and on-ground robot control computers.

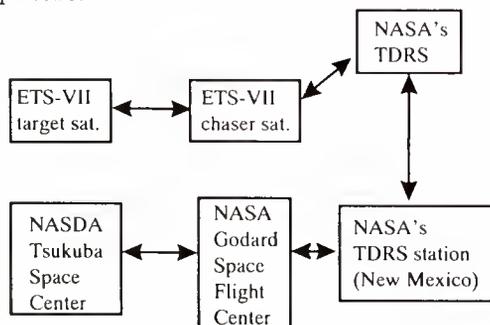


Fig.4 ETS-VII experiment system

Since not only NASDA but other agency (MITI: Ministry of International Trade and Industries) and national laboratories (NAL: National Aerospace Laboratory, CRL: Communication Research Laboratory) also conduct robot experiments on the ETS-VII

satellite, ETS-VII's teleoperation system was designed to allow other experiment users to use their own on-ground robot control facility. Those users brought their own on-ground robot operation facility to NASDA Tsukuba Space Center and their experimental commands are sent to the satellite through NASDA's on-ground robot operation facility. NASDA provided telemetry data and video data from satellite to these users' facility.

3.2 Onboard robot system of ETS-VII

ETS-VII's onboard robot experiment system consists of 6 dof(degrees-of-freedom) robot arm and many robot arm's experimental payloads which are shown in Fig.5. Besides NASDA's equipment (Robot arm, ORU, taskboard, target satellite handling tool), other agency(MITI, NAL,CRL)'s experimental equipment is also mounted on ETS-VII for their own robot experiments. (Ref.6)

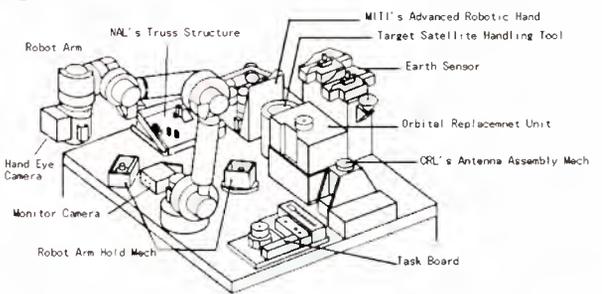


Fig.5 ETS-VII onboard robot equipment

The robot arm is about 2m stretched length and its joints are driven by combination of DC brush-less motor, harmonic drive gear and a resolver. An end-effector with three latching fingers and the force-torque sensor is mounted on the robot arm. A pair of hand eye cameras are also mounted on this end-effector. ETS-VII robot arm has following control modes;

- joint angle / velocity control mode
- arm tip position / attitude control mode
- compliance control mode (incl. force control, active limp and impedance control)

These control modes were tested in orbit and showed good performance. The positioning accuracy (repeatability) of robot arm was better than 1.5mm.

A hand eye camera is mounted on the end effector. Another monitor camera is mounted on the first joint. The first joint acts as camera's pan unit. Up to five video images per a second (5 f/s) are sent to ground using the JPEG compression format.

Add-on tools: ETS-VII robot arm's end-effector is designed to be most suitable for the

tasks like ORU handling. However it is not suitable to handle small equipment or to grasp a floating object like the target satellite. Therefore, ETS-VII robot arm uses additional tools to handle these payloads. A taskboard handling tool which has two fingers and a fixed peg is used to handle equipment on the task-board (slider, mechanical switches, surface for tracing). A target satellite handling tool which has large two fingers is used to grasp the target satellite. MITI's advanced robotic hand can also be attached to this robot arm.(Ref.5)

3.3 Coordinated robot and satellite attitude

The mass of the ETS-VII chaser satellite is about 2.5t. The ETS-VII's robot arm handles payloads of a few kg to 410 kg (target satellite). Attitude of the satellite platform is maintained within a few tenth degree by the reaction wheels and the gas jet thrusters against the robot arm's reaction. This is to maintain the communication link through the data relay satellite and to generate electrical power from its solar arrays. However, if the reaction of the robot arm motion is too large, the satellite attitude control system can not maintain the proper satellite attitude. Therefore, the coordinated control of the satellite attitude and the robot arm is realized through the coordination of the onboard satellite attitude control system, onboard robot control system, and the on-ground robot control system. Fig.6 shows this coordinated control system. (Ref.8)

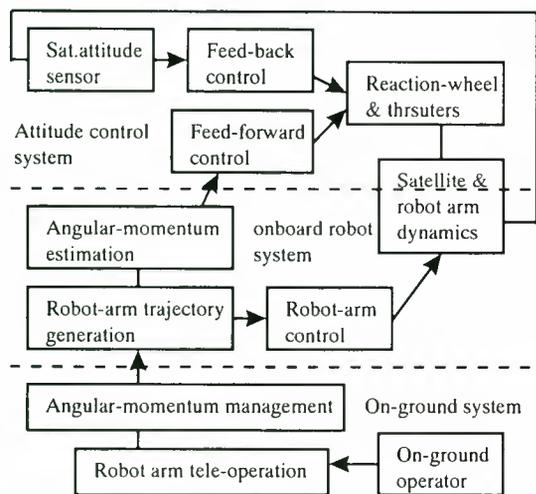


Fig.6 Coordinated satellite and robot control system

3.4 Teleoperation system

NASDA's robot teleoperation system has two operation modes, the "supervised control mode" and the "telemanipulation mode". In the

supervised mode, instructions to the onboard robot system can be sent in codes which mean like "Move from A to B at a speed of C, acceleration D, compliance parameters of E and, etc....." The onboard robot control system decodes this instruction and generates robot arm's tip trajectory. Then it calculates joint angles using the inverse kinematics, and controls individual joints. If the robot arm's working environment and the tasks to be conducted are well defined, automatic task execution is realized using this control mode. In this mode, the command sequences are written using the graphical user interface(GUI) into a flowchart. This commands sequence is verified using the built-in onboard robot simulator of the robot operation facility which simulates operations of the onboard robot system. In the actual operation, each command is automatically sent out each after the previous command is successfully sent and verified its execution. On-ground operator's task is to supervise this automatic task execution and if necessary to intervene and modify the process. This operation method is simple and safe, and is recommended for most of space robot's tasks which are well defined. Fig.7 ETS-VII shows robot teleoperation facility.(Ref.7)



Fig.7 ETS-VII robot teleoperation facility

(2) Telemanipulation of the robot arm

In the telemanipulation mode, instructions to the onboard robot system are sent in the form of the robot arm's tip position and pose at each 250msec. The onboard robot control system generates the robot arm's trajectory by interpolating these data. If one data is missing by communication error or other reasons, the onboard robot system will interpolate the missing command. If more than two commands are missing, the onboard system will stop robot arm's motion assuming that the commands from the on-ground station were stopped by the operator or by the communication error.

Telemanipulation under the time delay of 6 seconds is not easy. Therefore, ETS-VII's on-ground robot control system uses following operator's aids to assist telemanipulation.

- predictive computer graphics which shows how the robot arm will move if a command will be executed. This graphics also shows pose of the current robot arm using telemetry data from the satellite.
- shared control between the telemanipulation and the automatic control. Control of robot arm's individual coordinate can be selected between the automatic control and the telemanipulation
- imaginary guide plane to guide the robot arm motion to a desired position and to inhibit other motions.

Other functions such as the real-time health check of the onboard robot system which are used in the supervised control mode are also used in this telemanipulation mode to ease operator's workload.

3.5 Tasks conducted

Within NASDA's robot experiments, following robotic tasks were conducted.

(1) visual inspection of onboard equipment: Since ETS-VII's robot system is easy to teleoperate, satellite engineers who are not specialist of robotics can easily conduct inspection of onboard equipment using the hand eye camera of the robot arm.

(2) Handling of orbital replacement unit(ORU) and fuel supply experiment: ORU is widely used on the international space station to exchange equipment in orbit. Size and mass of ETS-VII's ORU are similar with those of a microwave oven. Fig.6 shows ORU handling by the onboard robot arm. This ORU houses fuel tank, valves, liquid connector and electrical connector to connect with other tank in the ORU port. A simulated fuel supply experiment was successfully conducted using this equipment.



Fig.8 ORU handling by the robot arm
 (3) handling of small equipment by the robot arm: There are many small experimental equipment on the taskboard such as a slider, mechanical switches, hole for peg-in-hole and

others. These equipment is handled by adding the taskboard handling tool which has two fingers and a fixed peg. Fig.9 shows handling a metal ball by the taskboard handling tool on the robot arm.



Fig.9 Handling of small parts using add-on tool (4) Telemanipulation experiments by shuttle astronaut: NASDA's shuttle astronaut, Mr. Wakata who conducted retrieval of free flyers using the shuttle manipulator on STS-72 (Jan.1996) conducted the telemanipulation experiments on ETS-VII. He also received through training of MFD robot arm's operation even though he was not assigned a crew of the mission. This experiments was to compare these robot arms. Tasks asked in this experiments were to trace a surface of the taskboard and to handle a truss structure by telemanipulating the robot arm. These tasks require control of robot arm's position/attitude and also force/torque between the objects. Mr.Wakata could satisfactory conduct these tasks even though he could spent only 2 days for training and there was a time delay of 6 to 8 seconds. This shows that ETS-VII robot system is easy to learn and to operate. Fig.10 shows the surface tracing experiment by astronaut Wakata which was conducted on March 16,1999.



Fig.10 Shuttle astronaut, Mr.Wakata(center) conducted surface tracing task by telemanipulation (Top left: hand eye camera's image, right: shoulder camera's image, Lower right: CG image of robot)
 (5) Handling of the target satellite by the robot arm: The target satellite whose mass is

410kg was handled by the onboard robot arm on the chaser satellite. This experiment was conducted as follows by the coordination of the on-board and on-ground robot arm rendezvous docking control systems. At first, a target satellite handling tool was attached to the robot arm and this tool grasped a handle on the target satellite. After releasing the target satellite by the docking mechanism, the robot arm move the target satellite up and down to give the maximum disturbance to check the coordinated satellite attitude control capability which is explained in next section. After these tasks, the target satellite was moved to the docking position by the robot arm and the docking mechanism grasped the target satellite. Chaser satellite's attitude was properly maintained throughout this experiment. Fig. 11 shows this experiment

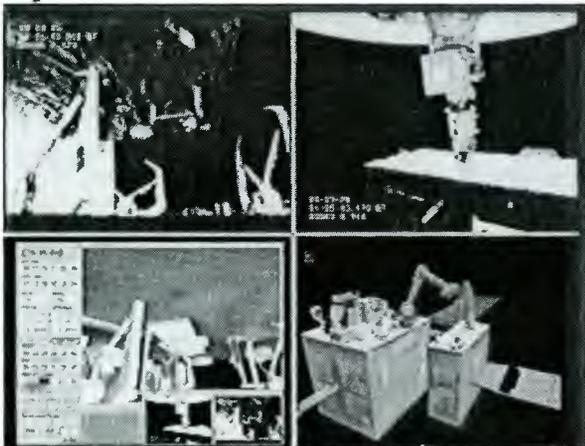


Fig.11 ETS-VII robot arm on the chaser satellite handles the target satellite

(Upper left: image from a docking camera (upper/target, lower/chaser), Upper right: image from a shoulder camera (target satellite handling tool is attached to the robot arm and grasps a handle on the target satellite)

3.6 Other agency's experiments

Since ETS-VII is a rare opportunity to conduct robot experiments in space, many institutions conducted space robot experiments.

- Ministry of the International Trade and Industries: They conducted the advanced robotics and experiments.
- National Aerospace Laboratory: They conducted handling of truss structures by the onboard robot arm
- Communication Research Laboratory: They conducted antenna assembling experiments using the onboard robot arm and the antenna assembling mechanism
- European Space Agency(ESA) and NASDA conducted the joint robot experiments using

NASDA's onboard / on-ground equipment and ESA's on-ground equipment.

- German Aerospace Center(DLR) and NASDA conducted the joint robot experiments using NASDA's onboard / on-ground equipment and DLR's on-ground equipment.

During these experiments, experimental commands were generated by user's own on-ground robot operation facility and were sent to NASDA's robot operation facility. NASDA's robot operation facility checked these commands in a real time whether the collision would not happen nor satellite attitude would not be disturbed. If the command was proper, then it was sent out to the satellite. Telemetry data and video data from the satellite were distributed to user's facility from NASDA's facility. This teleoperation system realized the user-friendly teleoperation system. Details of these experiments will be presented by these agencies. Fig.12 shows these users' experiments.

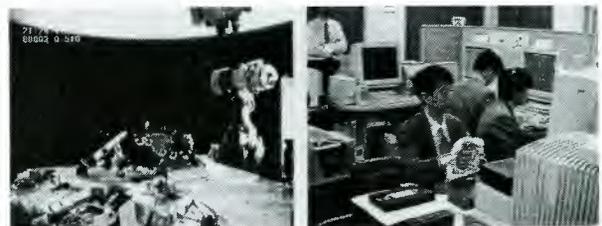


Fig.12a (left) Deployed truss structures of NAL and MITI's small robot arm with the advanced robotic hand. Upper is NASDA robot arm's end effector and the Earth. Fig.12b(right) MITI's teleoperation system



Fig.12c(left) NAL's teleoperation system

Fig.12d(right) CRL's teleoperation system



Fig.12e(left) Teleoperation from ESA's terminal

Fig.12f(right) Teleoperation from DLR's terminal

3.7 Rendezvous docking experiments

The rendezvous docking experiment is the another main mission of ETS-VII. It is characterized by autonomous rendezvous and soft docking. ETS-VII's chaser and the target satellite were launched together and were separated during the rendezvous docking

experiments. The chaser satellite measures the relative distance and relative rate of both satellite using the GPS receiver and the rendezvous radar. Then the chaser satellite autonomously generate trajectory to approach the target satellite and control the satellite to fly on that trajectory. At the docking, chaser's latching mechanism grasps the target's docking handle and connect the target satellite to the chaser. Fig.7 shows the docking of the chaser and the target satellite during the first rendezvous docking experiment. Details of the rendezvous docking experiments are shown in Ref.9.

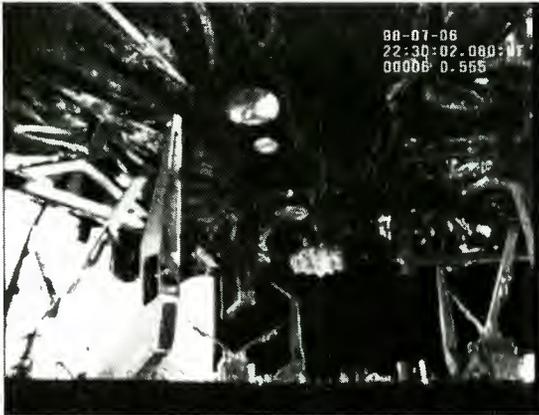


Fig.14 Docking of the ETS-VII chaser (bottom) and target satellite(upper)

4. JEMRMS (Japanese Experiment Module Remote manipulator System)

JEM(Japanese Experiment Module) including JEMRMS is the Japan's contribution to the international space station. Fig.15 shows the overall JEM including JEMRMS. JEM will be launched in three parts in Oct. 2001, Jan.2002, and June 2002. Development of JEM is in the final phase. Fig.16 shows JEMRMS engineering model(EM) under the JEM's system level test which was conducted in June 1998. (Ref.13)

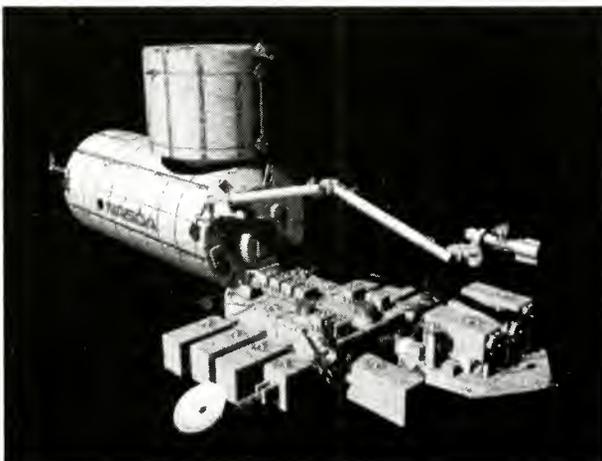


Fig.15 Japanese Experiment Module of the International Space Station

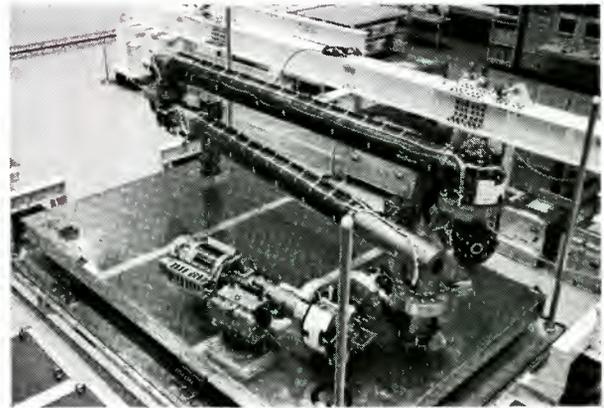


Fig.16 JEMRMS Engineering Model

5. Studies for future missions

Above mentioned are projects that their developments and operations are funded. NASDA is also conducting research activities for future possible robotic missions. Those include the ETS-VII follow-on mission which will demonstrate the in-orbit satellite servicing capability against troubled satellites. Service may include visual inspection, capture, attitude stabilization, orbital transfer, removal from orbit. Some preparatory experiments may be conducted using ETS-VII.

The Japanese Experiment Module (JEM) of the international space station is a platform for various scientific and engineering research. Robotic experiment is also under consideration and some preparatory work is being done by a team of NASDA, national laboratories, and universities. Details of these activities will be presented in other papers at i-SAIRAS'99.

References

More precise and up-to-date information on these projects can be got through the following web pages.

- http://www.nasda.go.jp/index_e.html
(NASDA web page)
 - http://oss1.tksc.nasda.go.jp/ets-7/index_e.html
(ETS-VII web page)
 - http://jem.tksc.nasda.go.jp/index_e.html
(JEM web page)
- (1) K.Ninomiya, M.Kawach, T.Yamaguchi, M.Sato, K.Tsukahara, "On-orbit operation of free flyer SFU's navigation, guidance and control system", Proc of AAS guidance and Control Conference, pp.1-15, AAS96-072, (1996)
 - (2) S.Wakabayashi, K.Matsumoto, Y.Horikawa, M.Nagatomo, "On the results of the manipulator flight demonstration for the JEM -Mission review and the results of the flight calibration experiment - IAF-98-T-2.03, 49th International Astronautical Congress (1998)
 - (3) M.Oda, et al.,"ETS-VII, the world first telerobotics satellite (Mission and its design

- concept)", in Proceedings of the 2nd International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'92), Toulouse, France, Sept.30-Oct.2,1992
- (4) M.Oda, I.Kawano and F.Yamagata; "ETS-VII (Engineering Test Satellite VII) - a Rendezvous Docking and Space Robot Technology Experiment Satellite", 46th International Astronautical Congress (IAF'95), Oslo, Norway, Oct.2-6,1995, IAF-95-U.2.01
 - (5) M.Oda, et al.; "Development of the EVA End Effector, Grapple Fixtures and Tools for the Satellite-Mounted Robot System", Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS'96), Nov.1996, Osaka, Japan, pp.1536-1543
 - (6) M.Nishida and M.Oda, "Space telerobot experiment system based on NASDA's ETS-VII satellite", Proc. of AIAA Guidance and Control Conference, Aug. 1997.
 - (7) M.Oda and T.Doi, "Teleoperation system of ETS-VII robot experiment system", Proc. of IEEE/RSJ International Conf. on Intelligent Robotics and Systems (IROS'97), pp.1644-1650, Sept.7-11, 1997, Grenoble, France
 - (8) M.Oda, "Motion control of the satellite mounted robot arm which assures satellite attitude stability", Acta Astronautica Vol.4, No.11, pp.739-750, 1997
 - (9) I. Kawano et.al., "First result of autonomous rendezvous docking experiments on NASDA's ETS-VII satellite", Proc. of 49th International Astronautical Congress, Sept.28-Oct.2,1998, Melbourne, Australia, IAF-98-A3.09
 - (10) K.Machida, H.Nishida and K.Akita, "Precise Telerobotic System for Space Experiment on ETS-VII," IAF-98-U.5.05, 49th Int. Astronautical Congress, pp.1-10, (1998)
 - (11) K.Matsumoto, et al., "Truss structure telemanipulation experiment using ETS-VII", Proc. of the 3rd int. symp. on artificial intelligence, robotics and automation in space, Oct.18-20,1994, Pasadena,CA,USA,pp.275-278
 - (12) S. Kimura, T. Okuyama, Y. Yamana, Y. Nagai, and H. Morikawa: Teleoperation System for Antenna Assembly by Space Robots, Telemanipulator and Telepresence Technologies V, ed. M. R. Stein, SPIE, 14-23 (1998)
 - (13) T.Matsueda, et al., "Development of Japanese Experiment Module Remote Manipulator System", Proc. of the 3rd International conference on artificial intelligence, robotics and automation in space (i-SAIRAS'94), Pasadena, CA., USA, Oct.18-20,1994, pp.183-186

Autonomous Locomotion : CNES Technological Program

M. Lamboley, M. Maurette

*Centre Spatial de Toulouse
18, avenue E. Belin 31401 Toulouse*

Tel. : (33)- 5-61-27-39-83 e-mail : marcel.lamboley@cnes.fr

ABSTRACT

Autonomy seems necessary to improve the range of operation of a planetary rover : this paper presents the developments (hardware and software) accomplished in C.N.E.S. in this topic and the results of experimentation done using different vehicles on the G.E.R.O.M.S. test site.

C.N.E.S. works have been focused for several years on stereo devices and processing software including picture processing to build a 3D model, path planning and rover control.

1.INTRODUCTION

The interest of mobile platform for planetary exploration is no more to be demonstrated : a set of scientific experiments on board of a rover is able to process different kind of samples encountered on its way and to select the ones which look particularly interesting for the scientists who examine the pictures taken from the rover.

But to do this on another planet, we have to cope with the delays required by data transmission, and especially pictures, from the rover to the Earth where they will be processed. These delays include :

- radio propagation delay, which can reach 20 minutes for a single way transmission using a satellite as transmission relay, we have to wait for the visibility of the relay, which happens not more than a few times a day
- using direct rover-Earth transmission allows only very low rates (for instance 1 kbit/s.) even with a radiated power of 20 W and a large antenna
- availability of the Deep Space Network
- processing time.
- In any case the delay between a snapshot from the rover and reception of command

from the Earth lasts at least several hours, that makes real time teleoperation unworkable.

Except the case where the rover is moving on a flat area without obstacles, it is unable to move beyond its perception range (i.e. in general several meters) between two command sessions if the path is defined from the Earth by an operator.

On this reason, a rover able to define its path and to move by itself on the soil of Mars will increase significantly its progression range. For a lunar rover, the transmission delays are much less even if a relay satellite is necessary, however the time necessary to process the data, define and validate the path can remain noticeable and a ground support must be available as long as the rover moves ; autonomy in this case is not mandatory, but can decrease a lot the operational charges.

2.REQUIREMENTS FOR AUTONOMY

The degree of autonomy developed by C.N.E.S. corresponds to a one-day mission during which the operator selects a distant goal without necessary knowing if there is a path to this point : the rover finds its way to this point by itself.

For this, the functions necessary on-board are :

- perception
- terrain analysis
- path planning
- path execution

Perception : perception of the environment shall be fine enough to detect all the obstacles to the progression of the robot. The notion of obstacle is related to the crossing capabilities of the rover ; we take into account two criteria for discriminating navigable zones in every cell of the digital terrain model (DTM):

- in an area of a size corresponding to the horizontal projection of a wheel, the difference of altitude between the lowest and the highest point shall not overpass a discontinuity threshold related to the crossing capability of the chassis ; this value must be chosen to fit with the rover features
- the slope of the rover in any direction shall not overpass a threshold related to the stability of the chassis.

Taking into account the perception range requirement (larger than 5 m.) and the features of the chassis ("light" rover whose length is between 50 and 100 cm, weight between 50 and 100 kg and wheel size between 10 and 20 cm) we need to build a digital terrain model with a vertical accuracy better than a few centimeters.

Two devices can be used :

- laser range finder : a laser beam is swept in site and azimuth in the whole field of view and we measure the propagation time of the beam reflected by the soil : the accuracy can be good enough, but this device is difficult to qualify in a stringent environment such as Mars or the moon due to the sweeping mechanisms. Furthermore, electrical consumption is high because of the mechanisms and wide-band electronics.
- stereovision : it consists of taking two pictures of the scene to analyze with two separate CCD cameras : in each camera, the pictures of a given object are at different places, according to the distance D between object and cameras. The difference of coordinates or parallax is proportional to $1/D$: stereo hardware is not too complex ; the drawback of this device is to require a big amount of calculations ; but, on one hand, processors are becoming faster and faster, on the other one, algorithms have been optimized to reduce computation time and memory volume to quite reasonable values.

We have based our development on stereovision ; the main features of our device are the following :

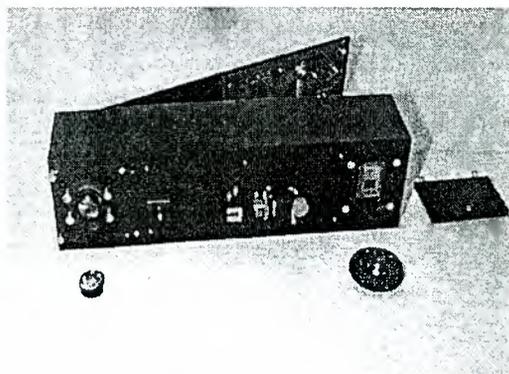
- stereo basis (distance between optical axes of the two cameras) : 200 mm
- CCD : 300x400 square pixels of 23 microns
- Focal length : 5.5 mm, giving as field of view +/- 40° in azimuth and +/-30° in site.

The grey levels of the pictures are digitized on 10 bits ; the pictures are sent to the processor

through a serial line at a rate of 10 Mbit/s : the two pictures are transmitted in less than 1 s. To reduce processing time for parallax calculation, it is mandatory that the images of the same object point in the two cameras are situated on the same line of the matrix, that requires either very harsh optical and mechanical specifications or an accurate characterization of the focal plane ; we choose the latter solution : in the whole field of view, we measure the position in the focal planes of image points given by a beam whose direction is very accurately known : the gaps between real and theoretical coordinates are put into a map used to correct the pictures. The one requirement of optics and mechanics is to remain stable.

Two prototypes of these stereo devices have been constructed using conventional technology for electronics, optics and mechanics ; their main features are following :

- Stereo basis : 200 mm
- mass : 1 kg
- size : 250x66x55 mm
- electrical consumption : standby 5 W
snapshot 30 W



CNES-LAS stereo device

For space application (Rosetta project) flight models are under development by CSEM using a very compact 3D assembly technology and a larger 1024x1024 points matrix : the size of the camera, optics included is 35x30x20 mm. With this technology, a stereo device using a 1024x1024 matrix would present following features :

- Stereo basis : 120 mm
- mass : 350 g
- size : 130x35x30 mm
- electrical consumption : < 4 W.

The stereo device is equipped with a triaxial accelerometer which can give roll and pitch angles allowing to build a digital terrain model with a vertical z axis ; it is connected to the

processor through a serial link whose throughput is 10 Mbit/s. The input link (processor to stereo device) is used to command the snapshots and define their parameters (exposure time and analog/digital converter to optimize the grey level histogram), and to require pictures and accelerometers data ; the output link delivers the data to the processor.

The use of active pixel sensors (APS) instead of CCD would still reduce mass, volume and consumption.

Terrain analysis : the pictures are first rectified (i.e. corrected from geometrical aberrations such as distortions or mechanical biases...) using the results of the calibration previously achieved.

The rectified pictures correspond to two cameras with optical axes parallel and distant between them of the stereo basis B , lines and columns of the two matrices parallel ; for both pictures, snapshot parameters are identical : focal length f , size of pixels p . Each object point seen by the two cameras gives one image point in each camera (on the same line in both after rectification): the difference d between the positions of the two corresponding points, called disparity or parallax gives the distance D between the projection of the object point on the optical axis of the cameras and the optical center of the lenses by the relation : $D = f.B/d$.

This allows to define the 3 coordinates of the object point when we have got the parallax d . To get the parallax, for each point of the left picture, we take a window centered on the point and we seek, in the same line of the right picture (in a range defined by maximal and minimal reconstruction limits) the position of the window giving a maximum of correlation between the textures of both windows according to our correlation criterion.

An interpolation gives a subpixellic value of the parallax to improve the reconstruction accuracy.

In return, taking the point found as maximum in the right picture, we look for its corresponding in the left picture : if we find the departure point, the correlation is validated ; if not, the point is labeled "unknown" : so, we eliminate most of the correlation errors.

When we have processed all the points, we have the parallax map from which a last filtering removes residual errors and we can build a cloud of 3D points used to build the digital terrain model.

Of course, this part of the algorithm requires a lot of processing time. This time has been reduced by different ways :

- simplification of the correlation criterion
- incremental calculation of the criterion by propagation along lines and columns
- calculation in two steps, the former one with reduced resolution to reduce the parallax interval.

This algorithm which represents the main part of the whole processing time requires less than 480 ms. on Ultraspac II at 248 MHz and less than 850 ms. on PowerPC604 at 150 MHz for pictures of 384x280 pixels, with a parallax interval of [4..60] and a base of 200 mm.

The digital terrain model is not complete : the correlation does not work on areas which are not seen by both cameras, and on very uniform places. These places are labeled "unknown" The other ones are labeled "navigable" or "not navigable" according to the criteria defined here above.

Path planning : we suppose that the goal of the robot is either a given point or a given direction. With a navigation map labeled as "navigable", "not navigable" or "unknown", it is possible or not to define a path leading to the goal through only navigable areas. If no path is available, the solution is to turn the cameras toward another direction to find a path. This method called "step by step planning" gives good results, but is not optimal for two reasons :

- if an accurate localization is not available, to take into account the uncertainty of the path execution it is necessary to enlarge the obstacles of a value equal to the maximum drift of the rover (proportional to the distance crossed by the rover), so that the planner would not pass between two obstacles separated by a gap wider than the rover if they are far from it.
- some navigable areas are labeled "unknown" when they are far from the rover because the density of 3D points decreases with the distance to the rover

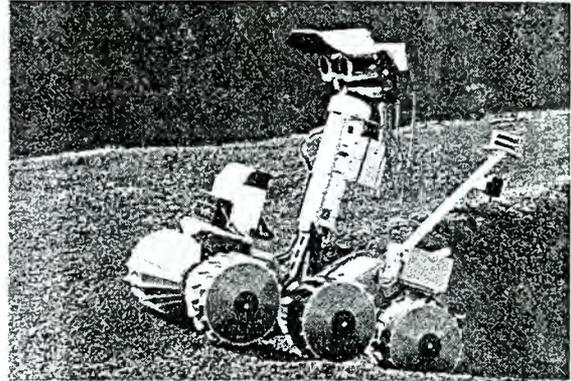
To overcome these drawbacks, another method called "continuous planning" has been developed [7] which consists of :

- merging the successive navigation maps to get larger planning areas : merging navigation maps whose content is the local slope and maximum discontinuity inside a cell solves the problem of elevation discontinuities due to localization and attitude uncertainty which occurs when we merge digital terrain models.
- The planning area around the rover is divided into two parts : one surrounding the rover can be crossed before a new perception and thus uses the same

algorithm than step by step planning ; but beyond, in the peripheral area, a new perception will be done before crossing it, and it is not necessary there to enlarge the obstacle or to forbid unknown areas which can become navigable by the next perception.

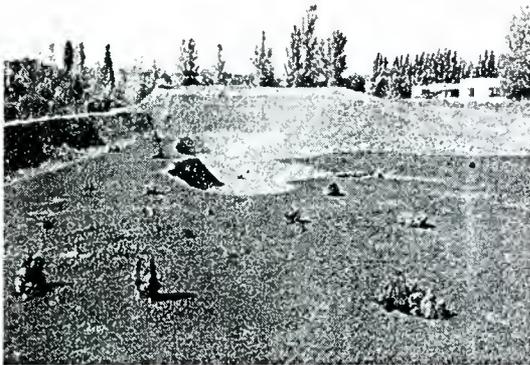
Path execution : the major problem in path execution is to get a good estimate of the motion ; on smooth soils, dead reckoning gives good results, but as far as soil becomes rocky or loose, the results become very poor. Inertial sensors can be used, but their mass and consumption are still very important. Motion estimation by vision has not these drawbacks and two different algorithms are presently under evaluation : accuracy of a few percent seems to be feasible.

considered for path planning. This method has already demonstrated its efficiency to drive the robot out of quite constrained rocky zones where the previous techniques [2], [7] met their limitations.



EVE vehicle

3 Results of experimentation



GEROMS test site

A first experimentation of these methods has been tested in CNES on the GEROMS site [1] using EVE vehicle adapted from a Marsokhod chassis [2]. In absence of gyros allowing measurement of instantaneous attitudes, image acquisitions are preferably performed with the chassis at stop. The experimentation validity is however satisfied since this mode could correspond to a non-nominal situation that would undervalue the method potentiality. Trajectories have been also adapted to the locomotion capabilities that can only execute linear movements and rotations on place. Localization errors up to 10% of the distance traversed and 2% in attitude have been

4. Conclusions

The major contribution of this method is the capability to improve the autonomy of a planetary rover by widening its knowledge of the environment already traversed. The robot can thus generate itself even better trajectories than what could be determined by an operator on the ground.

An interesting property of the navigation maps that we have proposed is the absence of hypothesis on the sensor type that produced the elevation data. This type of method will easily take into account data coming from other sensor devices (proximeters, radars for soft soil detection). Another advantage is the possibility to rely on localization sensors (position and attitude) with poor accuracy. This is particularly a critical issue for missions involving light vehicles (mass and power consumption). Finally the method is generic enough to be applied to small rovers (small CPU power and low memory). Perception data can be acquired either in motion or at stop. There is also no particular hypothesis made on the type of vehicle being used and the kind of trajectories executable by the locomotion system (gyrations, rotations on place, linear displacements...).

5. References

- [1] M. Delail. First campaigns on the GEROMS mobile robot test site. IARP., Montreal, June 1994.
- [2] M. Lamboley, C. Proy, L. Rastel, Marsokhod : Autonomous Navigation Tests on a Mars-Like Terrain, *Autonomous Robots*, 2, 1995.
- [3] J. Matijevic, D. Shirley, the mission and operation of the Mars Pathfinder microrover. IFAC 13th Triennial World Congress. San Francisco, USA, 1996.
- [4] R. Volpe, J. Balaram, T. Ohm, R. Ivlev, The Rocky 7 Mars Rover Prototype. 1996 IEEE Robotics and Automation Conference, Minneapolis April 22-28, 1996.
- [5] V. Michkiniouk, S. Medvedev, G. Koslov, Chassis of IARES-L planet rover demonstrator with a broad range of functional opportunities. i-SAIRAS'97, Tokyo, July 97.
- [6] L. H. Matthies, C. F. Olson, G. Tharp, S. Laubach, Visual Localization Methods for Mars Rovers using Lander, Rover, and descent Imagery. i-SAIRAS'97, Tokyo, July 97.
- [7] M. Delpech, C. Proy, L. Rastel, High Rate Autonomous Navigation For A Non Stop Traversing Rover. i-SAIRAS'97, Tokyo, July 97.
- [8] R. Volpe, Rocky 7 Lavic Lake Field Tests, IROS97 Rover Workshop, Grenoble, France, September 1997.
- [9] L. Rastel, M. Delpech, Rover Continuous Path Planning Using Merged Perceptions 1998 IEEE Robotics and Automation Conference.
- [10] M. Delpech, L. Rastel, M. Lamboley, Enhanced Path Planning and Localization Techniques for Autonomous Planetary Rovers ASTRA98 Advanced Space Technologies for Robotics and Automation ESTEC, Noordwijk December 98.

DLR's ROBOTICS LAB – RECENT DEVELOPMENTS IN SPACE ROBOTICS

G. Hirzinger, K. Landzettel, B. Brunner, I. Schaefer, M. Fischer,
M. Grebenstein, N. Sporer, J. Schott, M. Schedl, C. Deutrich

DLR Oberpfaffenhofen
German Aerospace Center,
Institute of Robotics and System Dynamics
D-82234 Wessling/Germany
Gerd.Hirzinger@dlr.de

ABSTRACT

The paper gives an overview of DLR's latest developments and project experience in space robotics. From the technology point of view, progress in the design and development of light weight robots and articulated multifinger-hands as well as in the refinement of DLR's sensor-based, task level teleprogramming system MARCO (and its virtual reality concept) is reported. In addition DLR's experiences with NASDA's free flying space robot ETS VII in terms of sensor-controlled ground programming and dynamic robot-satellite interaction are outlined. Ongoing laboratory experiments towards free flying space robots (ESS) are supposed to prepare the basis for a European or German free-flyer project. And the design of endeffector technologies and ground control concepts for the robotic part of EuTEF on the International Space Station are fully underway.

1 INTRODUCTION

After four decades of manned space flight, where many activities have become routine, one might forget that the space environment continues to be extremely hostile to human beings. They have to be encapsulated in vehicles (for intra-vehicular activities IVA) or special, extremely expensive suits, which protect them from the hazard of the space environment (for extra-vehicular activities EVA). When comparing human skills with those of present-day robots of course human beings in general are by far superior, but when comparing the skill of an astronaut in a clumsy space-suit with that of the best available robot technology, then the differences are already going to disappear, the more if there is a remote control and monitoring capability on ground with arbitrarily high computational and human brain power. For IVA activities a robot basically would have to compare with the full human skill and mobility; however to be honest, many of the manual operations to be done in a space-laboratory environment are fairly simple standard operations, like handling parts, opening and closing doors, pulling drawers, pushing buttons etc. which have to be done just by stepping through extensive, written procedures. Real intuition and manual skill is particularly requested in non-nominal situation, e.g. when a tape recorder has to be repaired. Although it is not clear today when a multi-fingered robot hand might be as skilled as the human hand and when (if ever) a robot might show up real intelligence and autonomy, it nevertheless is obvious that even with today's technology and the available telerobotic concepts based on close cooperation between man (e.g. the ground operator) and machine there are many tasks in space, where robots can replace or at least augment human activities with reduced cost at least from a long-term perspective.

Thus we are convinced that automation and robotics (A&R) will become one of the most attractive areas in space technology, it will allow for experiment-handling, inspection, maintenance, assembly and servicing with a very limited amount of highly expensive manned missions (especially reducing dangerous extravehicular activities). The expectation of an extensive technology transfer from space to earth seems to be more justified than in many other areas of space technology.

These are the reasons why DLR – after the big success of RO-TEX, the first remotely controlled space robot (Ref. 1) - has increased its efforts towards the development of a new, smart generation of light-weight-robots with articulated hands (robonauts) and convenient remote programmability from ground. The progress we have made, the technologies we prepare for the EuTEF robot on the ISS and our recent experiences with NASDA's freeflying space robot ETS VII are outlined in the sequel.

2 DLR's LIGHT WEIGHT ROBOT DESIGN

Space robotics is assumed to become a major drive for a new generation of light-weight robots, which will find numerous terrestrial applications, e.g. on mobile platforms, too.

The design-philosophy of DLR's light-weight-robots is to achieve a type of manipulator similar to the kinematic redundancy of the human arm, i.e. with seven degrees of freedom, a load to weight ratio of between 1:3 and 1:2 (industrial robots \approx 1:20), a total system-weight of less than 20 kg for arms with a reach space of up to 1,5 m, no bulky wiring on the robot (and no electronics cabinet as it comes with every industrial robot), and a high dynamic performance. As all modern robot control approaches are based on commanding joint torques, in the first carbon fibre type arm version (Fig. 1) showed up an inductive (13 bit, 1 KHz bandwidth) torque-measurement system that was an integral part of a double-planetary gearing system. A full inverse dynamics (joint torque) control system including a neural net learning system for compensating gravity modelling errors made use of it.

However the double-planetary gears (Fig. 1, right) with their extremely high reduction rate of 1 : 600 were very difficult to manufacture.

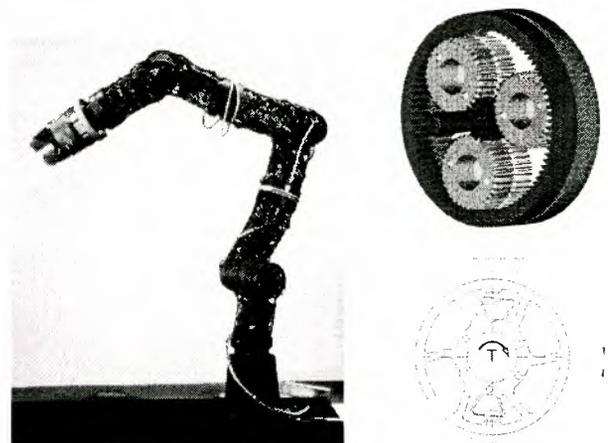


Fig. 1 DLR's first light-weight-robot with integrated electronics (left), double planetary gearing and inductive torque sensing (right)



Fig. 2 The CAD model of DLR's new 7-dof light weight robot arm (left) and the testbed setup of joint 2 (right)

Meanwhile a new light weight robot design (Fig. 2) is underway which tries to make optimal use of all the experience gained with the above „reference“ model. Its joints are based on special light-weight harmonic drives.

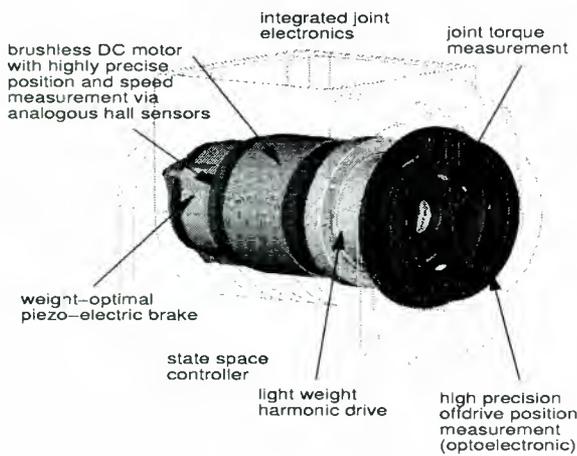


Fig. 3 Mechatronic components of DLR's new intelligent robot joint

In the drives we are measuring all relevant state variables, i.e. off-drive position, torque, motor position and speed (Fig. 3, 4a and 4b). For torque measurement we went back to strain gauge based systems. A first version of this new arm design uses so-called INLAND motors which were redesigned by us to provide hollow axes where all cabling is fed through.

A second version will use a new motor concept (Fig. 5) as developed in our lab, the optimized external rotor motor (OERM).

The electromagnetic torque generation to be delivered over a wide rotor speed range is realized by a multipole stator assembly interacting with rotor permanent magnet poles in a non-symmetrical configuration to virtually eliminate cogging effects. The dynamic performance is significantly enhanced by means of a special commutation control technique based on a single coil winding technique.

In view of the limited heat exchange to be realized with a compact design, the key design requirement is a large stall-torque-to-input-power-ratio. This number can be significantly enhanced as compared to conventional designs by careful tuning of geometrical dimensions and electromagnetic design parameters using magnetic field computations supporting a lumped parameter optimization process.

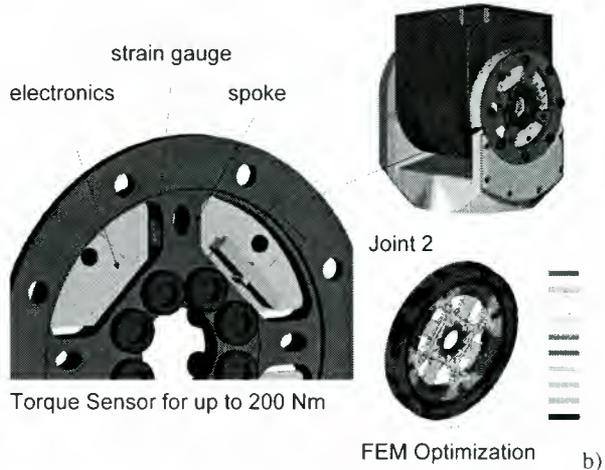


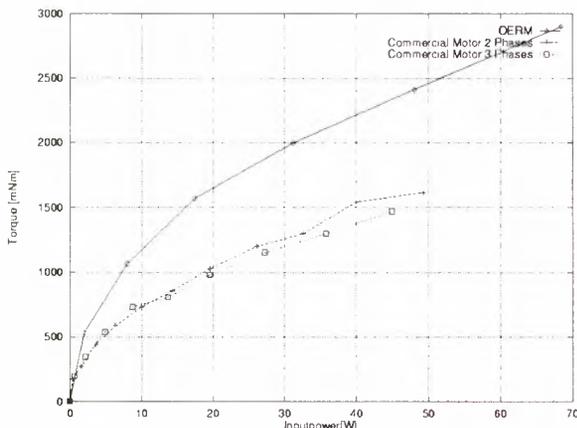
Fig. 4 Two sensors in DLR's new light weight robot (Fig. 2)
 a) Off-drive joint angle sensor (resolution 0,01 degrees)
 b) joint torque sensor
 Not shown here are the hall sensors used for motor position measurement.



a): Joint 2 with OERM and piezo controlled brake



b): OERM with Harmonic Drive



c) Stall torque OERM vs. the best commercial motors

Fig. 5 The Optimized External Rotor Motor (OERM) just needs about 38% of the stall torque input power which has been required by the best commercial motor used since, and moreover yields 50 % higher torques.

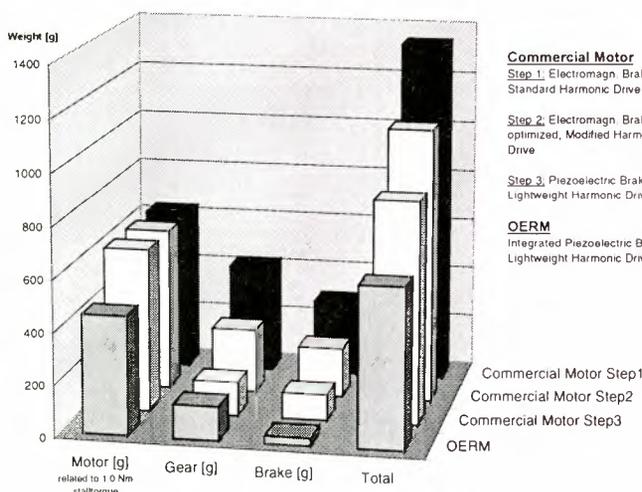


Fig. 6 The history of weight reduction in DLR's new LWR-Drive Units

The tedious history of weight reduction over the last two years is depicted in Fig. 6.

In the first step the (in our opinion) best commercially available high- end brushless DC- motor was combined with a slightly modified Harmonic Drive gear and a commercially available robot- safety-brake.

In the next steps the total weight was diminished by reducing weight in the Harmonic Drive's circular-spline and the development of a weight optimized, modified version of the original, commercially available electromagnetic brake, which has been replaced recently by DLR's new piezoelectric brake with a weight of less than half the original brake. Considerable further decrease of the drive-unit masses was reached by providing the Harmonic Drive with a new aluminum crafted wave generator and circular spline as developed in close cooperation with the company Harmonic Drive, so that it finally came out with only half the weight of the original part.

The biggest step towards an extremely lightweight construction was the development of a completely in-house designed Optimized External Rotor Motor (OERM) of high efficiency and

stall torque with a highly integrated piezoelectric safety- brake. The mass of the motor related to the stall torque at equal power consumption is less than 72% of the originally used high- end motor and the weight of the integrated brake (30 g) is just 1/10 th of the weight of the commercial brake used in the first step (300g).

The combination of the new Optimized External Rotor Motor (OERM), integrated safetybrake and lightweight Harmonic Drive gear yields an extremely powerful lightweight jointdrive with a related mass of just 55% of the weight of the original high- end drive unit and a joint quality measure of $J=250$, where we have defined this measure as

$$J = \frac{T}{W} \cdot \frac{v_{\max}}{(180^\circ/\text{sec})}$$

and where

$$T[\text{Nm}] = \text{Output torque (max)}$$

$$W[\text{kg}] = \text{Weight of joint}$$

$$v_{\max} [^\circ/\text{sec}] = \text{maximal speed}$$

Indeed it is not trivial to compare the performance of light weight joints, as output torque related to overall weight is meaningless if one does not take in account the joints maximum rotational speed, which we normalize via $180^\circ/\text{sec}$, a value which is e.g. a good standard for terrestrial robots.

In summary, we are convinced that the enormous efforts we made to arrive at joints with $\approx 210 \text{ Nm}$ output torque, $220^\circ/\text{sec}$ and $\approx 1,2 \text{ kg}$ weight including the brake system will pay out in the near future.

3 DLR's FOUR-FINGERED ARTICULATED HAND

For many space operations i.e. handling drawers, doors and bayonet closures (electric connectors) in an internal lab environment, two-finger grippers seem adequate and sufficient; the appropriate mechanical counterparts in the lab equipment are easily designed and realized even in a very late design stage.

For more complex manipulations future space robots (**robotonauts**) should use articulated multifingered hands.

In contrast to existing robot hand designs, it was our declared goal to build a **multisensory 4 finger hand** with in total twelve degrees of freedom (3 active dof in each finger), **where all actuators, uniformly based on the position-force-controlled artificial muscle® (see e.g. Ref. 2), are integrated in the hand's palm or in the fingers directly** (Fig. 7, Fig. 8). This means the hand is fully modular and may be mounted on any robot. Force transmission in the fingers is realized by special tendons (highly molecular polyethylene), which are optimal in terms of low weight and backlash despite of fairly linear behavior.

Each finger shows up a 2 dof base joint realized by two artificial muscles® and a third actuator of this type integrated into the bottom finger link (phalanx proximal), thus, actuating the second link (phalanx medial) actively and, by elaborate coupling via a spring, the third link (phalanx distal) passively. **Every finger unit with its 3 active degrees of freedom integrates 28 sensors(!).**

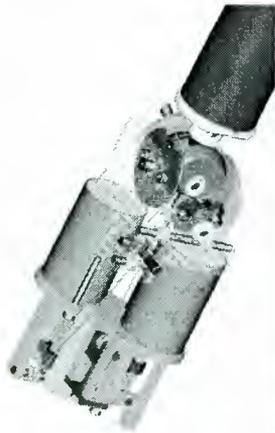


Fig. 7 The 2 degree of freedom base joint

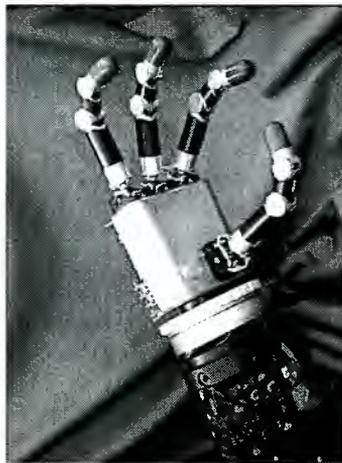


Fig. 8 Our 4 finger hand with its 12 actuators and 112 sensors integrates 1000 mechanical and 1500 electronic components

With 112 sensors, around 1000 mechanical and around 1500 electrical components the new hand is one of the most complex robot hands ever built. The fingers are position-force-controlled (impedance control), they are gravity compensated and they are prevented from colliding by appropriate collision avoidance algorithms. In addition recently a cartesian stiffness control scheme on hand level was implemented which turned out to be of crucial importance for all kinds of manipulation tasks. For more details see Ref. 3.

A number of telepresence demonstrations have meanwhile been performed using a dataglove, a polhemus tracker and on the „remote“ site the robotaut consisting of a 7-dof light-weight robot on a 3-axis rail system, and the four-fingerhand (Fig. 9).

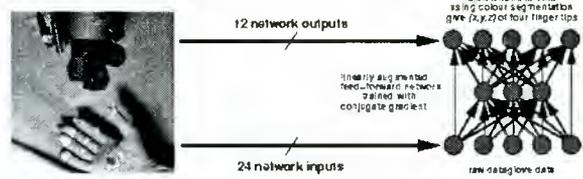


Fig. 9 Skill transfer from human hand to robot hand via data glove

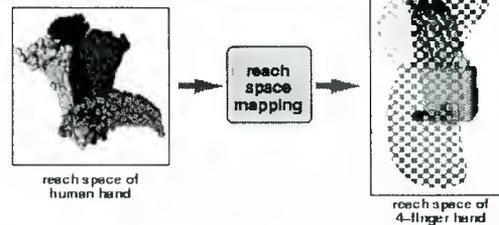
A spacelab mockup in our lab allows to remotely pull drawers, grasp objects in the most natural way etc. The robotaut concept of the „prolonged arm“ of man in space seems very realistic here (Fig. 2). Needless to say that in case of large delay all operations can be programmed and executed in a virtual environment using the MARCO telerobotic system (see section 4). Mapping the data glove signals into glove finger positions via neural nets (Fig. 10) as well as high and low level grasp planning modules for the position-force controlled fingers are meanwhile available for our hand (Fig. 11).

We have now started with the development of DLR hand II, which will show up an even higher degree of integration. As an example presently around 400 cables are coming out of the hand, they should be reduced down to less than 10 cables in DLR hand II.

1. Training phase.



2. Mapping of reach spaces.



3. Control phase.

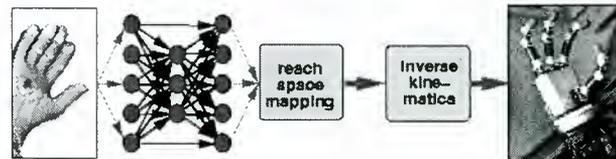


Fig. 10 Data glove control issues for 4 finger hand control

Model Based Manipulation

Object Motion Control with Spacemouse

Robustness by Stiffness Control



Grasp Planner

Online Planning (~ 10s)

Arbitrary 3D Objects



Fig. 11 High level manipulation and grasp planning skills are essential for efficient control of DLR's articulated hand

4 MARCO – DLR's TASK-DIRECTED SENSOR-BASED TELEPROGRAMMING SYSTEM

Following ROTEX we have focused our work in telerobotics on the design of a high-level task-directed robot programming system MARCO, which may be characterized as **learning by showing in a virtual environment** (Ref. 3) and which is applicable to the programming of terrestrial robots as well. The goal was to develop a unified concept for

- a flexible, highly interactive, **on-line programmable teleoperation station** as well as
- an **off-line programming environment**, which includes all the sensor-based control and local autonomy features as tested already in ROTEX, but in addition provides the possibility to program a robot system on an **implicit, task-oriented level**.

A non-specialist user - e.g. a payload expert - should be able to remotely control the robot system in case of internal servicing in

a space station (i.e. in a well-defined environment). However, for external servicing (e.g. the repair of a defect satellite) high interactivity between man and machine is requested.

To fulfill the requirements of both application fields, we have developed a 2in2-layer-model, which represents the programming hierarchy from the executive to the planning level.

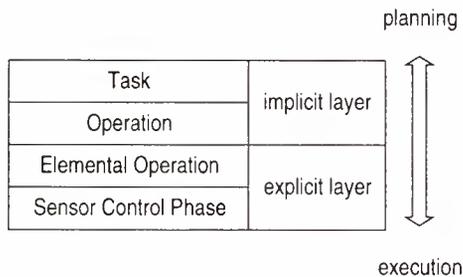


Fig. 12 2in2-layer-model

Based on this 4 level hierarchy (Ref. 4), an operator working on the (implicit) task level does no longer need real robotic expertise. With a 3D cursor (controlled by a Space Mouse) or with a human-hand-simulator (controlled by a data-glove) he picks up any desired object in the virtual world, releases it, moves it to a new location and fixes it there. Sequences of these kind of operations are easily tied together as complex tasks; and before they are executed remotely, the simulated robot engaging its path planner demonstrates how it intends to perform the task **implying automatic collision avoidance**. For having the real stereo-graphic imagination we use either shutter glasses with stereo-monitors or polarized glasses with large screens where many observers can watch at the same time. Stereo impression is perfect in both cases.



Fig. 13 DLR's universal telerobotic station MARCO (Modular A&R Controller) developed by contract with the German Space Agency

Nevertheless in the explicit layer (the learning phase) the robot expert has to show and demonstrate the elementary operations including the relevant sensory patterns and – if necessary – train the mapping between non-nominal sensory patterns and motion commands that servo into the nominal patterns later on in the real world. He performs these demonstrations by moving the robot's simulated gripper or hand (preferably without the arm) into the proximity of the objects to be handled (e.g. drawers, bajonet closures, doors in a lab environment), so that all sensory patterns are simulated correspondingly. The robot expert at this stage of course must have knowledge on position- and sensor-controlled subspaces (and must be able to define them, massively supported by MARCO functions), and he has to define

how operations (e.g. remove bajonet closure) are composed by elementary operations (approach, touch, grasp, turn etc.).

MARCO's two-handed VI interface concept

Thus as a general observation, on the implicit as well as on the explicit layer statement we have to move around 3D-pointers or grippers / hands in the virtual lab environment. Using classical "immersive" cyberspace techniques with data-glove and helmet was not adequate for our approach, as the human arm's reaching space is fairly small (e.g. in a lab environment) and with head motions only very limited translational shifts of the simulated world are feasible. As a general observation an alternative to the position control devices "data-glove and helmet" is the velocity control device "Space Mouse", particularly if the robot system to be programmed has no articulated hand. Velocity control here means we may easily steer around an object in VR over arbitrary distances and rotations via small deflections (which command velocities) of an elastic sensorized cap. The second important observation (confirmed by extensive tests of car manufacturers in the context of 3D CAD-design) is that just as in real life two-handed operations when interacting with 3D-graphics are the optimum. Indeed whenever humans can make use of both hands, they will do (e.g. when carving, modelling, cutting). In the northern hemisphere for around 90 % of the people the right hand is the working hand, while the left hand is the guidance and observation hand, which holds the object to be worked on (vice versa for left-handers).

This ideal situation for a human is easily transferred to the VR interface scenario. A right-hander preferably moves around the whole virtual world in 6 dof with a Space Mouse in his left hand (the guidance hand), while with his right hand he moves around the 3D cursor with a second Space Mouse (velocity control, Fig. 14) or a simulated hand with a data glove (position control, Fig. 15). One should note that now even for the glove the problem of limited workspace disappears, because with the left hand the operator is always able to move the virtual lab world around such that the objects to be grasped are very close so that even in position control mode with a data glove only small, convenient motions of the operator's hand are requested to reach them.

More details on MARCO's high level user interface as are Java/VRML client techniques are given in Ref. 4.

One of the key features of the MARCO system is the implication of sensor-based autonomy using the above mentioned storage of nominal sensory patterns.



Fig. 14 Two handed VR-interface using two Space Mice (ETS VII scenario as example)

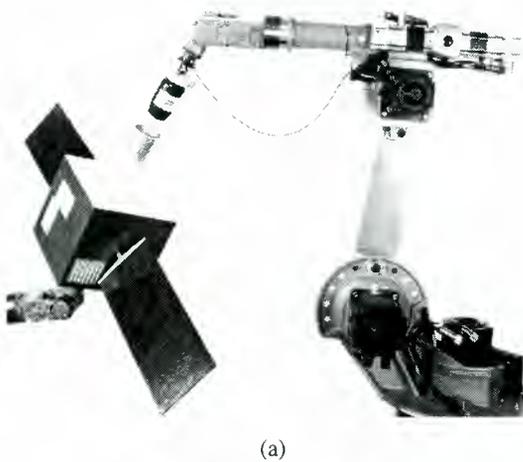


Fig. 15 Two handed VR-interface using Space Mouse and Data Glove (EuTEF scenario as example)

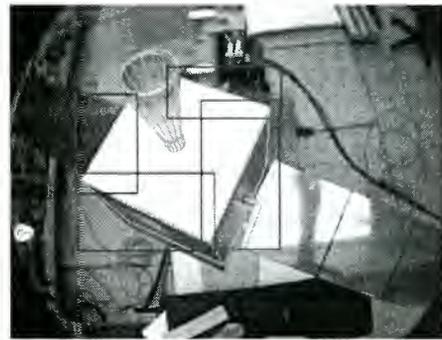
Indeed for comparing the real world with the virtual world, based on may be multisensory perception, and thus for either updating the world model or/and servoing into a nominal situation as learnt during the explicit layer training phase, MARCO provides several alternatives:

- if we have reliable CAD models of the objects and the environment and if we may assume that spatial 3D-contours are well detectable by a (mono or stereo) vision system, we prefer 3D-modelbased realtime tracking algorithms; in case of moving objects we additionally imply Kalman filters for estimating motions (Ref. 2). A typical example is the hardware simulated catching of free-floating satellite by a repair robot-satellite using two industrial robots in DLR's lab and a multisensory (vision, laser range, force-torque) capture tool; for more details see e.g. Ref. 2 and 4.

By the way if the robot is supposed to generate 3D models autonomously when CAD models are not available, one of our preferred technologies is 3D reconstruction from stereo images –using Radial-Basis-Function neural networks (Fig. 17), Ref. 5 and 6.



(a)

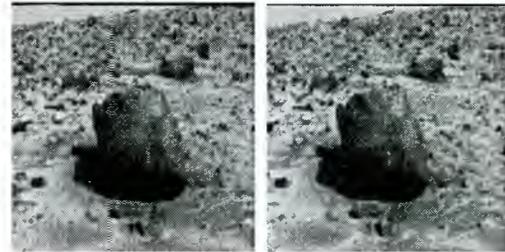


(b)

Fig. 16 A two robot system as testbed for a satellite repair project.

(a) The robot on the left carries a mockup of a satellite, that is tracked by a camera mounted in the tool of the repair robot (right).

(b) Satellite tracking as seen from the repair manipulator's hand camera. The wire frame model of the target is projected into the live video image at the currently estimated pose.



Stereo image

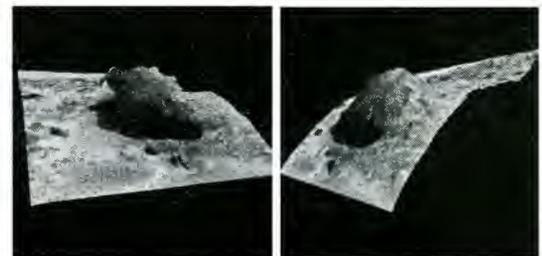
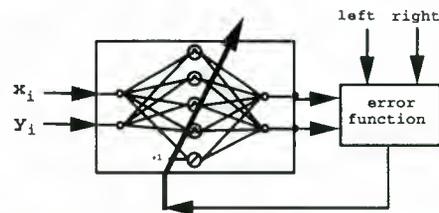


Fig. 17 3D-Reconstruction with neural networks
Example: stereo image of stone Yogi/Pathfinder Mars Mission

- if we have no 3D-CAD models or situations where they are not useful (e.g. if the camera does not see real 3D-contours) or where the sensor fusion aspect is prevailing (e.g. cameras combined with arrays of range finders), we prefer to train a linear mapping from sensory input patterns into corrective motions (that for example servo into some nominal relative position). Two alternatives have turned out to be very efficient here (Fig. 18).

- an analytical approach where the linear operator corresponds to the pseudoinverse of a Jacobian.
- a neural net approach using multilayer perceptrons.

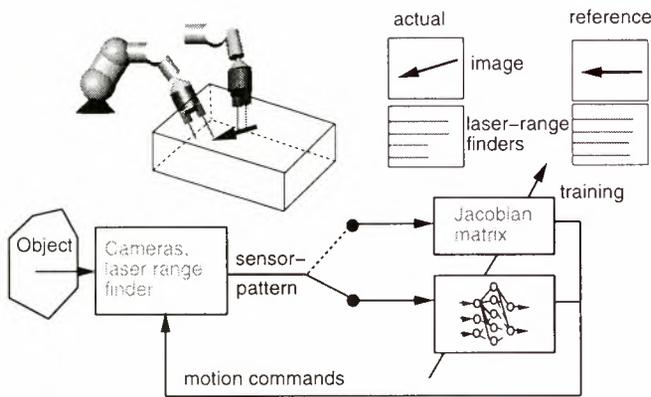


Fig. 18 Multisensory Servoing

In both cases the operator (or an automatic control scheme) has to move the robot gripper/hand slightly around some nominal situation in all six degrees of freedom.

Method a) has been used in the DLR-NASDA cooperation project GETEX with the ETS VII freeflying robot (see next section). In this case learning of the Jacobian was originally performed with simulated sensory patterns, and then repeated in Tsukuba/Japan with the real images yielding very similar results.

The sensorbased task-level-teleprogramming system MARCO, has reached meanwhile a high level of universality. It was not only used as ground control station for the ETS VII experiment (see section), but it is used also for technology studies of Germany's technology project Experimental Service Satellite ESS, as well as for remote ground control of EuTEF and for mobile terrestrial (fetch and bring services in hospitals) and planetary robot projects.

5 GETEX on ETS VII

From April 19 – 21 DLR (and the IRF Dortmund as subcontractor) had the opportunity (as offered by NASDA) to perform own experiments with NASDA's ETS VII free-flying space robot. Our goals were twofold:

- **To verify the performance of the MARCO telerobotic concept, in particular concerning the implicit task level programming capabilities as well as the sensor-based autonomy and world model update features.** A highlight was indeed the tele-programming of a peg-in-hole task, where in the virtual world we intentionally displaced the standby position of the peg from where the robot had to fetch it. Vision processing on ground using NASDA's tracking markers on the task board and the Jacobian matrix learning beforehand based on real images (as explained in the last chapter) caused the ETS VII robot to automatically and perfectly adapt to the unexpected situation. The peg-in-hole insertion as such (taking into account the fairly high tolerances) was less critical and of course made use of NASDA'S compliant motion commands.
- **To verify 6 dof dynamic models for the interaction between a robot and its free-flying carrier satellite.**

If a robot which is mounted on a spacecraft moves, it generates linear and angular momentum. In the case of an attitude and position controlled spacecraft, the attitude control system will permanently produce forces and torques compensating for the arm motion. The spacecraft may then be considered as inertial

in the co-ordinates of an orbit-fixed system, and the problem of robot motion planning can be solved using the same methods as for terrestrial manipulators. While for the control of the spacecraft attitude electrically powered momentum wheels can be used as well as thrusters, for control of the spacecraft translation fuel consuming thrusters are the only actuators currently in use. For this reason and because the position errors are generally negligible, most satellites are only attitude controlled. Due to the linear momentum conservation, which states that the center of mass of the system comprising the robot and the satellite is constant, the motion of a manipulator mounted on the satellite will lead to a compensating motion of the satellite. The amount of satellite translation produced depends on the masses of the bodies constituting the system. For space robotic systems which are neither position nor attitude controlled the angular momentum conservation law leads further to a rotation of the spacecraft, by an amount which results from the mass and inertia properties of the manipulator links and the spacecraft. It is generally assumed that no external forces act on such *free-floating robots* (Ref. 5 and Ref. 7). The free-floating mode of operation is of interest for space robots not only for the reason that attitude control fuel may be saved which augments the robot life-span, it will also be of importance during repair missions, when the servicing satellite is very close to or in contact with the target satellite: any action of the attitude control system of either of the two satellites during this phase would lead to a collision and thus to potential damage on the two spacecraft.

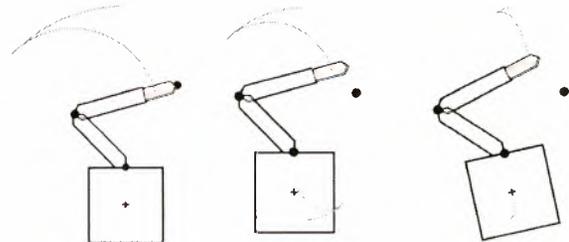


Fig. 19 The influence of the satellite attitude control mode on the path described by the robot end-effector - the same joint motion is carried out by a robot with a fixed base (left), an attitude controlled robot (middle) and a free-floating robot (right).

As long as the tasks performed with the robot are described in robot-fixed coordinates, the fact that the satellite position remains uncontrolled has no influence. If, however, the task is described with respect to an orbit-fixed co-ordinate system, as it would be the case for example for the capturing of a defect satellite, the satellite motion has to be taken into account (see Fig. 19). The equations relating the tool center point motion to the manipulator joint motion, which for robots with an inertially fixed base are purely kinematical equations, become thus dependent on dynamic parameters in the case of free-floating space robots. This influences the path planning methods which have to be applied. On one hand, singularities, that is joint configurations in which the robot is not controllable in cartesian co-ordinates, are no more a function of the robot kinematics only, but become dependent on the dynamic properties of the robot, too. Therefore, iterative methods based on the direct kinematics equations have to be used instead of the inverse kinematics equations. Moreover, the angular momentum equation makes the system nonholonomic (Ref. 9), which means that the satellite orientation is not a function of the current joint configuration only, but merely a function of the chosen path. Two different paths starting at the same initial configuration of the robot, and leading to the same final configuration, will therefore

result in different amounts of satellite rotation – and thus in different final inertial tool center point positions, too. As a consequence, nonholonomy offers the possibility to do a re-orientation of the satellite using manipulator motion only, by simply carrying out a closed-loop manoeuvre in joint space. This kind of manoeuvre can be employed to significantly augment the workspace of the robot, since it allows to turn the satellite into any desired orientation, bringing back the manipulator into its reference configuration. The maximum workspace of a free-floating space robot is thus described by a hollow sphere of which the inner and outer radius are given by the minimum and maximum possible distance between the tool center point and the system center of mass. Another possibility resulting from nonholonomy is that any point which is inside the fixed-base workspace of the robot may be attained with zero satellite attitude error. In the simplest case, this may be done by planning and executing the manoeuvre as for a robot with a fixed base and adding a closed-loop re-orientation manoeuvre to compensate for the produced attitude error. Path planning for nonholonomic system has been investigated in the context of cars and wheel-driven robots (Ref. 9). While those systems may generally be considered as planar, the case of free-floating robots demands spatial methods.

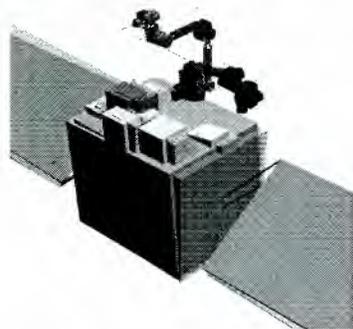


Fig. 20 Examples of *Dynamic Motion* manoeuvres carried out during the GETEX mission: simple point-to-point manoeuvre and re-orientation manoeuvre. The shaded robot indicates the reference position. The satellite reaction to the arm motion is scaled by a factor of 10 in this picture.

Whatever path planning method is applied to free-floating robots it is necessarily highly model-based. The parameters of the dynamic model have therefore to be known quite well. While this poses no problem for the geometric parameters and for the mass and inertia of the manipulator, the mass and the inertia of the spacecraft are subject to important changes during the lifetime of a servicing satellite. This is especially the case if the spacecraft is performing capturing or rendez-vous/docking like operations. One goal of GETEX has therefore been to identify the mass properties of the satellite after one year and a half of activity in orbit. Further objectives were the verification of the dynamic models and to obtain some insight into the nature and importance of the disturbances acting on a robotic satellite on low earth orbit. Additionally, the mission aimed at gathering data for the future design of controllers which combine the manipulator motion control with the satellite attitude control. To meet all these objectives, a variety of different manoeuvres were executed, which include simple point-to-point operations and closed-loop re-orientation manoeuvres (examples of which are given in

Fig. 20), sequences during which only one joint was active at a time as well as sequences during which all joints were moving simultaneously. The major constraints, due to mission security aspects, were the maximum satellite attitude error allowed by NASDA which was limited to $\pm 1.0^\circ$ around each axis and the

fact that the maximum tool center point velocity was limited, too. Furthermore, the reaction wheels were turning at a very low but non-zero constant velocity during the experiments, which introduced undesired torques into the system. Their effects will have to be considered during the evaluation of the mission results.

In total, over 110 minutes of dynamic motion experiments have been carried out, of which 52 minutes have been spent in free motion mode. The remaining time was used to repeat the experiments in reaction wheel attitude control mode for verification purposes. First evaluations of the measurement data confirm the need to account for external disturbance forces acting on the satellite, such as the gravity gradient torque and magnetic torque.

6 EuTEF on the International Space Station ISS

The European Technology Exposure Facility EuTEF, which basically consists of an Express Pallet on the outer truss structure of the ISS (Fig. 21), is supposed to be operated by a robot system.

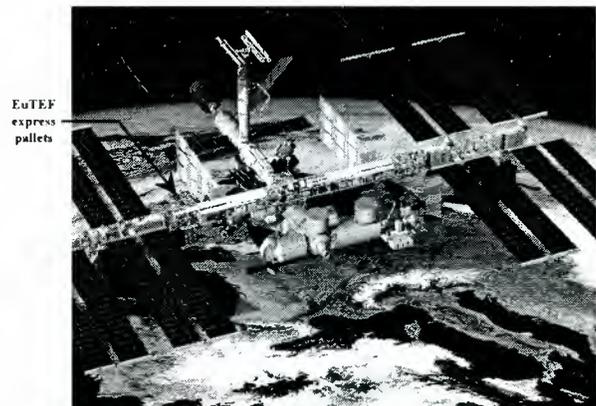


Fig. 21 International Space Station ISS (courtesy of ESA-ESTEC)

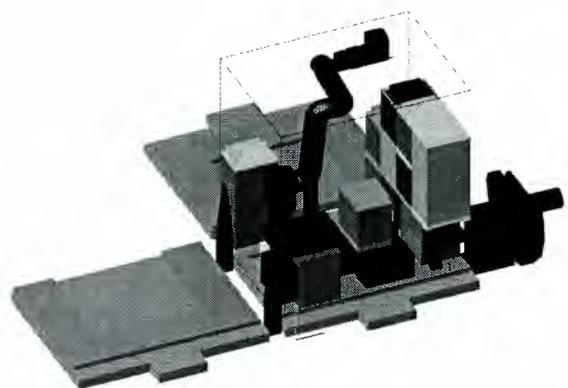


Fig. 22 EuTEF Scenario (courtesy of CGS)

Prime contractor of this ESA project is the Italian company CARLO GAVAZZI SPACE (CGS). The arm (a development of TecnoSpazio) presumably is provided by the Italian space agency ASI, while DLR intends to contribute the design of the endeffector, is counterpart (the Standard Grasping Unit) and the MARCO ground control system. It is our declared goal to really push space robot technology forward in the framework of EuTEF by

- refining the endeffector from a “basic” endeffector into a smart endeffector which allows for the onboard processing

of force **and** stereo vision, similar to concept we applied in the GETEX-ETS VII project.

- b) demonstrating by the powerful, high level ground control concept MARCO that fully remote control of **operational** space robot systems is feasible today.

The EuTEF robot is supposed to move around pallets and drawers, exposing them to sun, earth and stars, i.e. to perform operations which in the past needed e.g. complete reorientation of the shuttle. However safety is a key issue for this robot – not imaginable if it would for some reason (including programming errors) lose one of its loads. These necessities fully guide the endeffector's design and that of its complex counterpart, the *Standard Grasping Unit* (SGU).

During the past year DLR performed extensive studies on the design of the SGU to be used as the base for any *payload module* (PM) on the *Technology Exposure Facility* (TEF). This interface is robot operated via the *Basic End-Effector* (BEE) mounted to the robot's wrist.

Due to the mutual interface the SGU and the BEE are jointly designed by DLR. The finalizing and the manufacturing of the SGU will be done by HTS in Switzerland and CGS in Italy. The BEE is a national contribution and therefore the development will be performed by DLR exclusively.

Each PM consists of the *standard body structure* (SBS) mounted on the SGU. It is placed on the ExPA base by means of the *standard receptacle* (SR). Fig. 23 shows an exploded view of this arrangement.

A set of *standard receptacles* (SR) is fixed to the ExPA. This allows a payload module to be docked to certain predefined positions on the ExPA thanks to the SGU. Each PM is interfaced mechanically **and** electrically. Furthermore it incorporates on its top a mechanical interface similar to the SR. This allows payloads being stacked on top of each other.

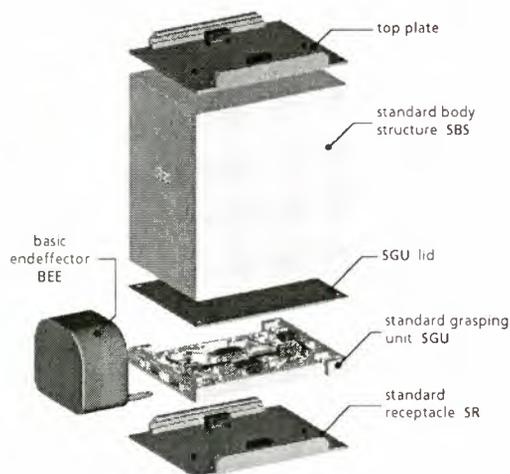


Fig. 23 Payload module PM (exploded view)

Fig. 24 takes a deeper look into the details of the SGU and the SR.

The payload docking interface has to withstand launch loads. The P-shaped receptacle clamps are prepared for this kind of load. They are slightly tapered to yield form closing contact with the blades of the SGU. To allow for easy insertion the receptacle clamps are chamfered noticeably thus yielding good guidance. To prevent jamming of the moving blades they are guided by a roller on each side (Fig. 24) thus reducing friction by a considerable amount. Two trigger pins on the SR are used

to block the locking mechanism of the SGU whenever the PM is lifted off the SR.

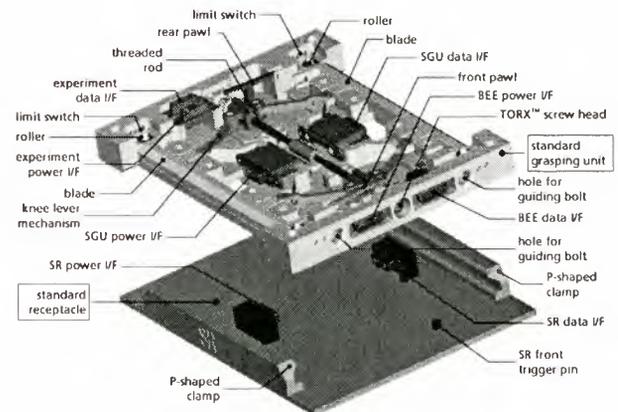


Fig. 24 SGU and SR (exploded view)

The SR routes power and data busses to the PM via two electrical connectors. These connectors are an integral part of the SR and thus aligned to its geometry.

The SGU incorporates three different mechanical/electrical interfaces: the *docking interface* towards the SR, the *grasping interface* for the BEE and the *experiment interface* towards the payload's experiment. Furthermore the SGU routes power and data busses to the PM's *top plate*, which in turn serves as SR for another PM (Fig. 23). This may be looked at as a fourth interface.

To initiate a *grasping action* the BEE is first inserted into the standard grasping interface. The alignment is supported by the guiding bolts. Being coupled the BEE drive operates the knee-lever mechanism by turning the threaded rod. It should be noted that this rod is locked by the front pawl until the BEE is inserted completely (Fig. 24). It should also be kept in mind, as indicated above, that this rod is locked by the SR's trigger pins *in either direction of motion* whenever the PM interface is lifted a small distance off it (front pawl and rear pawl). It is one of the benefits of the knee-lever arrangement that it yields **force balance within the rod**. Thus axial forces on the rod's bearings are zero under nominal conditions!

The threaded rod is linked to the BEE drive via a simple coupling element: It ends in a TORX™ screw head while the BEE drive is equipped with a matching screw driver shaft (compliant for proper insertion). Via the knee-lever mechanism the blades of the SGU may be opened thus releasing the lock of the SR. There is only a **relatively small torque** necessary for operation due to the gear ratio of the nuts on the threaded rod and the amplification effect of the knee-levers.

Following the very first movement of the blades the BEE is already mechanically latched to the SGU! This is done **well before** the payload module is unlocked. Together with the above mentioned pawls this prevents the PM from being lost due to erroneous commands.

During the undocking action the data and power busses are switched over to the robot in a make-before-break action. The docking action works inversely. The movement of the knee-lever mechanism is limited in either direction by a mechanical hard stop. Shortly before the hard stop positions limit switches are placed. This implies that a hard stop may be reached only, if the corresponding limit switch fails! The limit switch being used to detect the *closed position* will be reached shortly before final

closure. This allows to build up proper closure forces while making up for tolerances of different SRs.

The SGU is a very delicate and most sensitive part as far as robustness, safety and reliability is concerned. So there are used **mechanical latches only** for safety reasons. Although there are electrical sensors the whole mechanism can be operated without. These sensors are used to ease the handling of the whole apparatus and for redundancy purposes.

The BEE (Fig. 25) is attached on top of a separate force/torque-sensor mounted to the robot wrist flange. The BEE operates on the SGU as mentioned above. Since the BEE is a vital part of the TEF experiment it is most sensitive as far as robustness, safety and reliability is concerned. Thus it is built up making use of a sophisticated mechanical design in favor of smart electronic equipment. Generally speaking, sensors are used for backup or redundancy purposes only. This imposes constraints on the robot arm where accuracy and repeatability is asked for.

The BEE grasping interface consists of a double pronged fork similar to fork lifts. The cylindrical prongs (guiding bolts) are equipped with rounded tips for easy insertion into the SGU. The tips are grooved to complement the *interface latch*. There are no sensors integrated into the prongs themselves. Any torque due to a displacement from the nominal position during fit in must be either detected by the force/torque sensor and compensated by the calculated compliance of the robot or the fitting procedure must rely on the robot's mechanical compliance! The latter case is mandatory anyhow to serve as backup solution in case of sensor failure.

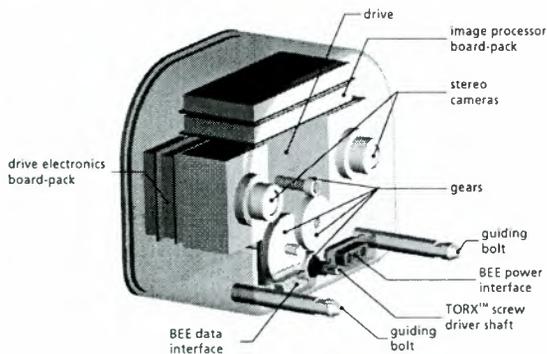


Fig. 25 BEE (semitransparent view)

The BEE's stereo cameras are instrumented for image processing. Compressed images are generated for transmission to the crew or the ground station via the MIL-STD bus. Furthermore adequate algorithms may be applied on the image data for control purposes **in the short local loop (local autonomy)**.

Prior to any mechanical contact the insertion process may be controlled by the above mentioned stereo cameras. The completion of the insertion is detected by a limit switch placed on the BEE's surface in the vicinity of the guiding bolts. The sensory aid is used to smooth the insertion process, but nevertheless it stays a backup solution, as told before.

As drive a hybrid stepper motor is an adequate solution. It combines high torque capabilities with a large number of steps per rotation. The electronic commutation logic allows to set the torque of the drive. It also allows to count the steps of the motor control and hence to measure incrementally the angle/turns of the spindle. With the help of the limit switches of the SGU there may be information retrieved on the absolute position of the blades. In case the limit switches fail, the drive is operated purely torque controlled. It is quite obvious that the drive elec-

tronics are on the 'critical path' as far as reliability is concerned. Therefore the drive electronics are built up in a redundant configuration.

CONCLUSION

Space robots in the future will take over more and more tasks from humans. Already at the space station – and even for its construction – a number of remarkable manipulator and robot systems will be active. However most of them will be more or less exclusively operated by astronauts, and this is one of our main concerns and disappointments. The real value of space robots lies in their remote programmability and controllability in combination with onboard autonomy, realizing the prolongation of human's arm into space. The relevant technologies including powerful and delay compensation 3D-graphics are available – it's our task to convince politicians and decision-makers in agencies that time is mature for the robotics age in space. As a consequent next step we try to help in making Eu-TEF the first fully remotely controlled **operational** space robot system. It is commonly accepted, that space robotics may become a major drive for many kinds of service robots – be it the light-weight aspect for mobile arms or the telepresence ideas in medical surgery of the future.

REFERENCES

- 1 G. Hirzinger, "Sensor-based space robotics – ROTEX and its telerobotic features", IEEE Transactions on Robotics and Automation, vol. 9, No. 5, Oct. 1993.
- 2 K. Landzettel, B. Brunner, G. Hirzinger, I. Schaefer, M. Fischer, M. Grebenstein, N. Sporer, J. Schott, „Space robotics – recent advances in DLR's Robotics Lab“, ASTRA 98
- 3 M. Fischer, P. van der Smagt, G. Hirzinger, Learning techniques in a dataglove based telemanipulation system for the DLR hand. IEEE International Conference on Robotics and Automation (ICRA), 1998
- 4 B. Brunner, K. Landzettel, G. Schreiber, B.M. Steinmetz, G. Hirzinger, "A universal task-level ground control and programming system for space robot applications, iSAIRAS 5th Int. Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 99, ESTEC, Noordwijk
- 5 Longman, R.W., Lindberg, R.E. and Zedd, M.F., "Satellite-mounted Robot Manipulators – New kinematic and Reaction Moment Compensation", International Journal of Robotics Research, 3, 1987.
- 6 G.Q. Wei, G. Hirzinger, „Intensity and feature based stereo matching“. Int. Conference Computer Vision, Bombay, India, Jan. 98.
- 7 Dubowsky, S. and Papadopoulos, E., "The Kinematics, Dynamics and Control of Free-Flying and Free-Floating Space Robotic Systems", IEEE Transactions on robotics and Automation, 5, 1993.
- 8 Nakamura, Y. and Mukherjee, R. : Nonholonomic Path Planning of Space Robots via a Bidirectional Approach, IEEE Transactions on Robotics and Automation, 4, 1991.
- 9 Li, Z. and Canny, J.F.: Nonholonomic Motion Planning, Kluwer Academic Publishers, 1993.

**Space Robotics Applications on the
International Space Station (1)**

Overview of the Mobile Servicing System for the International Space Station

Michael E. Stieber and David G. Hunter

Canadian Space Agency, Saint Hubert, Quebec Canada
 Adrian Abramovici
 MD Space and Advanced Robotics Ltd.

ABSTRACT

The Mobile Servicing System (MSS) will play a critical role in the on-orbit assembly, external maintenance and operations of the International Space Station. This paper reviews the mission and tasks to be performed by the MSS on the Station and provides an overview of the design of the MSS. Several examples illustrate how the mission requirements have shaped the design features of the MSS.

1. INTRODUCTION

The launch of the first module, "Zarya", of the International Space Station (ISS) in November 1998 marks the beginning of a period of unprecedented on-orbit construction to establish a permanent outpost for mankind in space. The Space Station as shown in Fig. 1 after completion of assembly will constitute a unique laboratory for research in material science, life sciences and other areas of scientific research.

Robotic systems provided by Canada will play a key role during the assembly of the Space Station in orbit and for the external maintenance of the station. This paper discusses the major mission requirements and corresponding design solution for the Mobile Servicing System (MSS) on the Space Station

2. MISSION AND ELEMENTS OF THE MOBILE SERVICING SYSTEM

2.1 Mission of the MSS

The MSS is being designed to perform the following functions on the Space Station [1]:

- Space Station Assembly including removal of Space Station elements and equipment from the shuttle cargo bay.
- Space Station external maintenance including changeout of Orbit Replaceable Units (ORU), actuation of mechanisms, mating/demating utilities, and visual inspection.
- Transportation on the Space Station of payloads such as Space Station elements, attached payloads and ORUs.
- Servicing of Space Station external payloads including ORU changeout, harvesting and replenishment, visual inspection, temporary provision of power, video and data connectivity, and visual inspection.
- Deployment and retrieval of free flyers by capturing and maneuvering to appropriate sites on the Space Station or deploying from the Station.
- EVA support including transporting or positioning of EVA crew, providing temporary storage, viewing, lighting.

2.2 Elements and Configuration of the MSS

The diversity of the tasks to be performed and the need to reach worksites all over the Space Station drive the design of the MSS towards a mobile system with multiple manipulators as shown in Figure 2. The large Space Station Remote Manipulator System (SSRMS) is designed to perform the capture, manipulation and berthing operations for large payloads. MSS functions requiring dexterous capabilities are satisfied by the smaller, dual-arm Special Purpose Dexterous Manipulator (SPDM).

The SSRMS, an operations platform called the MSC Base System (MBS), and a Mobile Transporter (MT) together form the Mobile Servicing Centre (MSC). The MT provided by the United States is designed to move along rails on the Space Station truss to transport the MBS, SSRMS, SPDM and other payloads. The SSRMS and SPDM feature standardized "LEE/PDGF" interfaces discussed below which allow the MSS to be configured in a number of different ways. The SSRMS and SPDM can "work together" as shown in Figure 2, or can attach themselves separately to the MBS or fixed points on the Station modules as required for different tasks.

The elements of the MSS are connected through a common computer and video network, and are controlled from one of two Robotic Work Stations (RWS) in the pressurized environment of the Station nodes. The MSS electronics equipment located outside the MSC and SPDM forms the MSS Control Equipment (MCE).

3. OPERATIONAL REQUIREMENTS AND DESIGN FEATURES OF ROBOTIC ELEMENTS

3.1 Space Station Remote Manipulator System

The SSRMS shown in Figure 3 is a 7-joint symmetrical manipulator approximately 17 metres in length when fully extended. A symmetrical arrangement of joints and a Latching End-Effector (LEE) at each end allows either end to attach to payloads, or to serve as a base for the SSRMS, providing that an appropriate Power Data Grapple Fixture (PDGF) interface is available. This design feature allows the SSRMS to "walk" on PDGFs and leave the MBS to use PDGFs on various Space Station modules as an operating base. This capability of peripeditation greatly increases the reach of the SSRMS on the Station as required during the assembly of the Station, e.g. as shown in Figure 5.

The LEE along with the PDGF interface are shown in Figure 4. The LEE incorporates a snare mechanism, a rigidizing carriage mechanism, a latching system and umbilical connection. The snare mechanism is designed to snare the protruding probe of the grapple fixture. After the snare is closed, the carriage containing the snare mechanism and the snared probe is drawn into the LEE until the grapple fixture base plate is in full contact with the face of the LEE with a specified preload. If a higher stiffness interface is required e.g. when serving as

the operating base of the SSRMS, or if power, data and video is to be transferred across the interface, a latching mechanism is activated, and an umbilical connector is engaged with its mating connector on the Grapple fixture. Power, data and video may be passed through the SSRMS to operate the SPDM while attached to the SSRMS LEE or to support the keep-alive power, telemetry and command requirements of payloads attached to the LEE. Each SSRMS LEE also incorporates a six-axis force-moment sensor which enables a force-moment control capability (FMA) of the SSRMS. Four video cameras are mounted on the SSRMS, one fixed camera at each end effector, and one, along with a pan and tilt unit on either side of the elbow joint on the main booms. A light is provided with each of the cameras.

The 7 joints of the SSRMS are arranged in clusters of three joints near each end of the manipulator to act as a "wrist" and "shoulder" respectively, with an additional joint at the midpoint "elbow" position. Starting from either end, the joint sequence is roll, yaw, pitch, pitch, pitch, yaw, roll. All joints are identical and have a range of travel of +/-270 degrees. Because the number of joints exceeds the 6 degrees-of-freedom in which the manipulator tip is being controlled, the manipulator is classified as kinematically redundant. The kinematic redundancy increases the operational flexibility by allowing to control the motion of the "elbow" independently of the payload motion in order to assist maneuvering the SSRMS around station hardware and helping to avoid kinematically singular configurations.

Operations using the SSRMS involve the handling and positioning of a wide range of payload shapes and mass properties. The mass range of payloads which can be handled by the SSRMS is from zero (i.e., no payload) up to 116,000 Kg, which is representative of a fully loaded shuttle orbiter. Assembly operations typically require the positioning of large payloads relative to berthing interfaces with an accuracy in the order of 5 to 10 cm. A "Space Vision System" (SVS) supports the precise positioning of payloads with the SSRMS in the absence of suitable direct vision/video reference from the berthing site. Figure 6 shows a typical module berthing scenario as seen from an SSRMS elbow camera. From such a video image, the SVS computes the relative position of the module berthing interfaces by photogrammetric analyses of the

location of "SVS Targets" on the modules. The criticality of the Space Station assembly operations and the need to reach into the Cargo Bay of the Space Shuttle drive the video, power and control systems of the SSRMS to full redundancy in order to satisfy the requirement to "fail operational" with full performance capability. In addition, the SSRMS design is "scarred" for a third "back-up" command string which would allow limited SSRMS operations after the occurrence of failures both in the primary and secondary string.

3.2 Special Purpose Dexterous Manipulator

The SPDM is shown in Figure 7. The robot is made up of a body and two manipulator arms attached to shoulder structures on the "upper" body. The upper body has a PDGF compatible with the SSRMS LEE and the "lower" body has a LEE. Both interfaces provide full power, video and data connectivity to the robot. Thus the SPDM can operate either while attached to the SSRMS, or when attached to a PDGF on the MBS, where it is also stored while not in use. The lower body features two outriggers carrying cameras and lights on pan/tilt units (CLPTA) and a tool holder with 4 tools for special operations. The SPDM lower body also provides a platform for the temporary attachment of an ORU carrier for several ORUs. A roll joint in the body allows the upper body including the arms to rotate relative to the LEE, cameras and tools.

The two SPDM arms are identical 7-joint manipulators with a straight-arm reach of about 3.3 m and a payload capacity of 600 kg. The arms have a clusters of 3 joints at the shoulder and near the tip, with a pitch joint at the elbow position near the midpoint of each arm. The arms have the same joint sequence as the SSRMS and are therefore described by similar kinematic equations. The tip of each arm is equipped with an ORU-Tool Changeout Mechanism (OTCM) and the wrist of each arm contains a six axis force-moment sensor. The OTCM incorporates a parallel jaw gripper compatible with standard H and micro fixtures, an extendable 7/16 inch socket drive, a camera with a two stop zoom lens and two lights, and an extendable umbilical mechanism. The umbilical can provide power, data and video connectivity to SPDM payloads.

The SPDM is employed for numerous dexterous operations in assembly and maintenance such as handling and replacing ORUs,

connecting/disconnecting utilities, attaching covers, actuating mechanisms with the socket drive, and performing operations with special tools. Figure 8 shows examples of dexterous tasks on the Space Station which require that the robot follows a constrained motion ("hinged motion"). The SPDM can also act as an extension of the SSRMS by means of the LEE on its lower body. About 250 ORUs of various designs are designated for robotic servicing on the Space Station. The dexterity of the tasks require a resolution of motion of the SPDM arms of 2 mm, and a high-performance programmable force-moment control capability (FMA) to avoid the possibility of jamming ORUs in their alignment guides.

Most operations are performed while the SPDM is attached via its PDGF interface to the SSRMS which positions the SPDM at the worksite. While performing ORU extraction or insertion with one arm, the SPDM is usually stabilized by attaching the OTCM of the other arm to a stabilization "H-fixture" near the worksite as shown in Figure 9. Once a failed ORU has been removed from its receptacle on the Station, the second arm removes the replacement ORU from the ORU carrier in order to free up a storage location for the failed unit. After the first arm has stored the failed ORU on the carrier, it is now used to stabilize the SPDM while the replacement ORU is installed on the Station by the second arm. The "Single ORU Storage Location" logistics concept for Space Station ORUs drives requirements for the SPDM Body Joint as well as for the dual-arm configuration of the SPDM.

The operations performed by the SPDM are of lower criticality to the Station in comparison to SSRMS operations since Extra-Vehicular Activity (EVA) by astronauts can be regarded a "second string" for performing maintenance on the Station. Safety requirements dictate that the SPDM fail "safe" which implies that after the occurrence of a failure the SPDM has to attain a state which is safe with respect to the SPDM element, its payload, and the Space Station. This usually requires that the SPDM has the capability to extract itself from the worksite. To this end redundancy is required for a number of functions of the SPDM such as payload release and bolt torquing.

3.3 Robotics Workstation

The MSS manipulators are controlled from one or two dedicated Robotic Work Stations (RWS) in the

pressurized, shirt-sleeve Space Station environment, one in the US Laboratory Module and one in the Cuplola as shown in Figure 10. The control station-to-operator interface includes a number of displays for video views of operations, a command and control display providing graphical and numerical information and soft keys activated by a pointing device, two hand controllers for manual manipulator commands, and a keyboard. The man-machine interface concept with two hand controllers to "fly" the manipulators using rate commands arose from a comprehensive trade-off study among 6 competing concepts conducted in the early 1990s, involving astronauts from different countries. This fairly conventional concept is also employed on the Space Shuttle and hence maximizes commonality between Shuttle and Station for Space Station assembly operations.

Concepts for an additional control station located in the Space Station Control Center on the ground are discussed in [3]. This control station represents a future enhancement of the MSS to provide the optional capability to operate the MSS from the ground. This control station would provide a similar operator interface as the on-orbit workstation, but would include some additional features to accommodate communications delays and other problems associated with remote operation.

3.4 Control Modes and Features

The control modes for the SSRMS and the SPDM are quite similar and include Human-in-the-Loop control modes and Automatic control modes [4]. Human-in-the-Loop control modes are generally tele-operator modes where the operator controls the manipulator Point-of-Resolution (POR) with the aid of resolved motion control algorithms in the manipulator control software (Manual Augmented Mode), or on a joint-by-joint basis (Single Joint Mode). In the Automatic modes, the operator takes on more of a monitoring role, initiating, observing, and perhaps modifying automated operations. Resolved motion or joint-by-joint operation is also available in the Automatic modes.

A number of control features may be selected by an operator to enhance manipulation capabilities and reduce operator workload. Some of the most important control features are:

- *Force-Moment Accommodation (FMA)*: Provides active backdriving/compliance to external forces and moments measured by force-moment sensors at the manipulator tip.
- *Position/Orientation Hold Selection (POHS)*: Automatically controls errors on uncommanded degrees-of-freedom of the manipulator in Manual Augmented mode.
- *Arm Pitch Plane Change (APPC)*: The kinematic redundancy of the manipulator is used to rotate the plane of the arm defined by the three pitch joints while the tip of the manipulator is held stationary. This allows repositioning of the arm to avoid collisions or provide better viewing or to avoid joint limits.

3.5 Space Vision System

An important operational component of the MCE is the Artificial Vision Unit (AVU) implementing the Space Vision System functionality discussed earlier. The AVU provides manipulator payload positioning data to the MSS operator in the form of graphical and textual displays. The AVU is capable of utilizing the output of any of the MSS or Space Station cameras for photogrammetric image processing in real-time at the video-frame rate of 30 Hz. The AVU development builds on the Space Vision System technology tested during Shuttle mission STS-52 [5].

4. SUMMARY AND CONCLUSIONS

The MSS will play a critical role in the assembly, maintenance and operation of the International Space Station. The robotic tasks on the Station are quite diverse and require a mobile robotic system with multiple manipulators in order to be able to perform the required robotic functions at various worksites across the station. The larger SSRMS manipulator is primarily used for the handling of large payloads while the SPDM, in conjunction with the SSRMS, is designed to perform dexterous maintenance operations involving smaller ORUs.

ACKNOWLEDGEMENTS

The authors would like to thank their colleagues at the Canadian Space Agency and the staff of Spar Aerospace for their efforts in the design of the MSS.

LIST OF ACRONYMS

REFERENCES

1. Werstiuk, H.L., Gossain, D.M., "The Role of the Mobile Servicing System on Space Station, IEEE Conf on Robotics and Automation, 1987.
2. Hunter, D., Darlington, T., Krukewich, K., Laurenzio, D., "Design and Operation of the Special Purpose Dexterous Manipulator (SPDM): Advancing the State of the Art in Space Manipulator Systems", 44th Congress of the International Astronautical Federation, October 1993, Graz, Austria.
3. Bassett, D.A., Wojcik, Z.A., Zaguli, R.J., Stieber, M.E., "Ground Based Control of Robots Aboard Space Station", 44th International Astronautical Congress, Graz, Austria, October, 1993.
4. Stieber, M.E., Trudel, C.P., "Advanced Control System Features of the Space Station Remote Manipulator System", 12th IFAC Symposium on Automatic Control in Aerospace", September 1992.
5. MacLean, S.G., Pinkney, H.F.L., "Machine Vision in Space", Canadian Aeronautics and Space Journal, Vol 39, No. 2, June 1993.

AVF	Artificial Vision Function
AVU	Artificial Vision Unit
EVA	Extra-Vehicular Activity
FMA	Force-Moment Accommodation
FMS	Force-Moment Sensor
HCA	Hand Controller Assembly
LEE	Latching End Effector
MBS	MRS Base System
MCE	MSS Control Equipment
MRS	Mobile Remote Servicer
MSC	Mobile Servicing Centre
MSS	Mobile Servicing System
MT	Mobile Transporter
ORU	Orbit Replaceable Unit
OTCM	ORU/Tool Changeout Unit
PDGF	Power Data Grapple Fixture
POR	Point-of-Resolution
RWS	Robotic Work Station
SPDM	Special Purpose Dexterous Manipulator
SSRMS	Space Station Remote Manipulator System

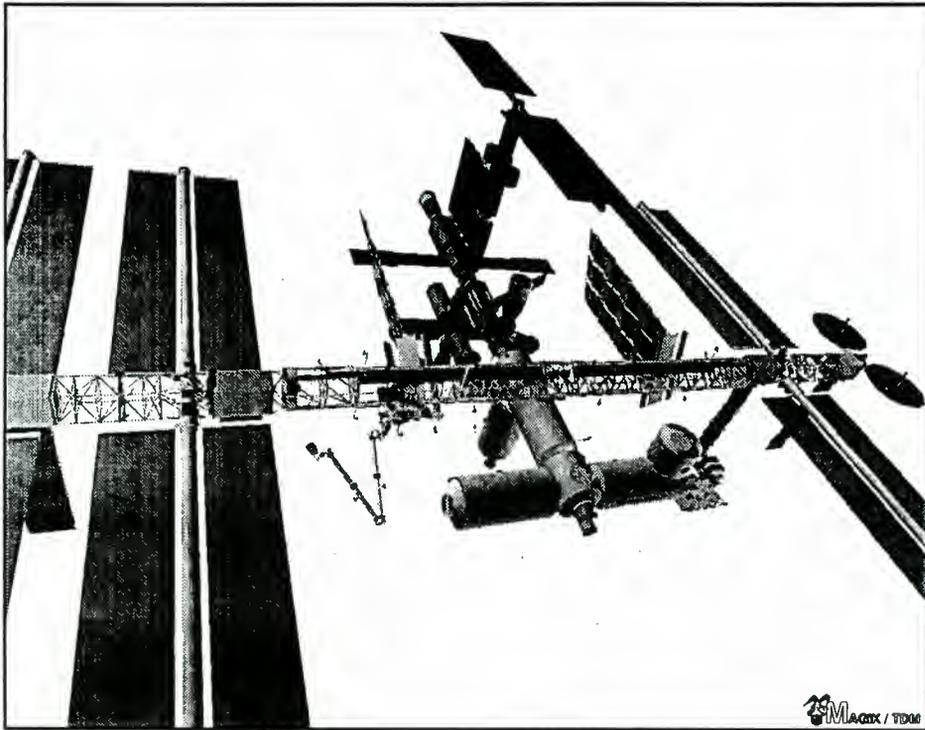


Figure 1: International Space Station

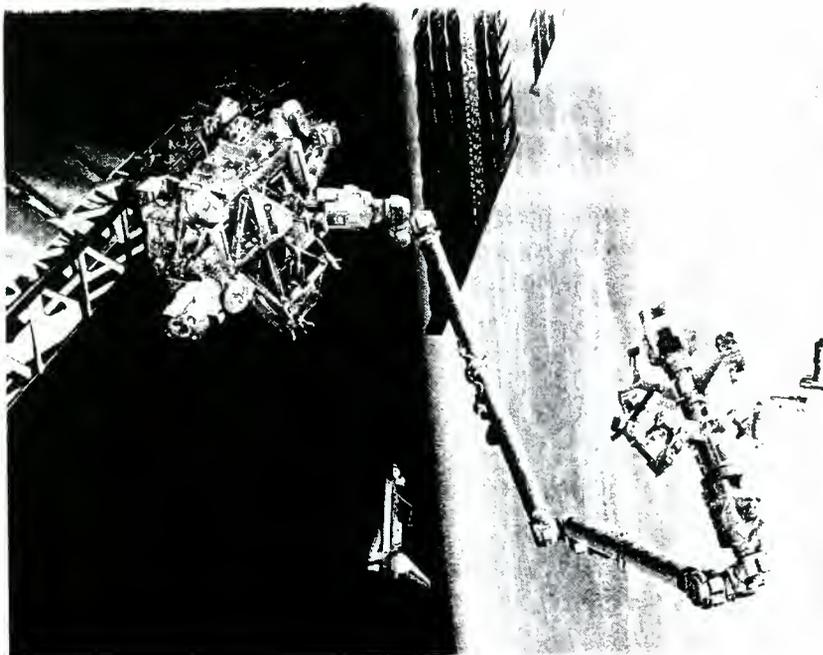


Figure 2: Mobile Servicing System

SPECIAL PURPOSE DEXTEROUS MANIPULATOR (SPDM) REQUIREMENTS VERIFICATION

D. A. (Doug) Bassett, M. Sc. - Deputy Program Manager Development
6767, route de l'Aéroport, Saint-Hubert, Quebec, Canada J3Y 8Y9
Tel. (450) 926-4477, Fax (450) 926-4576, doug.bassett@space.gc.ca
Adrian Abramovici, P. Eng., PMP – Director, Product Assurance,
MD Space and Advanced Robotics Ltd., 9445 Airport Rd. Brampton ON Canada L6S-4J3
Phone: (905) 790-2800 x4592, Fax: (905) 790-4506, aabramov@mdrobotics.ca

Abstract

The Special Purpose Dexterous Manipulator (SPDM) is the first space robot to be designed based on a large number of commonality and legacy components. Therefore the designers of the SPDM could not utilize the conventional approaches to the Verification Planning and Implementation of this complex system. This paper describes the approach taken by the SPDM designers and Customer in addressing this issue.

Introduction

The Special Purpose Dexterous Manipulator (SPDM) is the latest Space Robot being developed by McDonald Detwiller Space and Applied Robotics (MD Robotics, previously Spar Aerospace) for the Canadian Space Agency as part of the Canadian contribution to the International Space Station Program – the Mobile Servicing System (MSS). The SPDM is a dual-arm robot that responds to the ISS requirement for an external Dexterous Robotic alternative to Extra-Vehicular Activity. It is a complementary and integrated component of the MSS.

The design and development of the SPDM presents several challenges related to the nature of the Fixed Price program undertaken by MD Robotics Ltd.

A number of drivers affect the SPDM Verification Process:

- (1) The SPDM initial design was started and attained various levels of maturity for various components (PDR and CDR for some items) under the Space Station Freedom environment, then the Program was stopped and finally restarted under the harsher environment in which the International Space Station is being built,
- (2) The SPDM is the first space robot to utilize previously developed and space certified robotic components, such as Space Station Robotic

Manipulator System (SSRMS) Orbital Replaceable Units (ORU) and components,

- (3) Most of the electronics are based on previous designs developed for the SSRMS, with various degrees of modifications and component changes, and
- (4) Similarly, the SPDM S/W is composed of commonality (SSRMS) Software CSCIs, legacy (high percentage of commonality with previous designs) CSCIs and new Software.

MD Robotics has defined and is employing a unique, tailored Verification Program for the SPDM Systems and Components, including:

- (1) New Verification Planning and Tracking tools
- (2) An Incremental Buyoff process has been defined to facilitate full Customer visibility into the progress made during the verification process
- (3) Early testing of software components integrated in the overall SPDM Control model
- (4) The use of Flight Hardware for early software-to-hardware integration, will be described.

A brief description of the facilities created in support of the SPDM Verification effort will be provided, with emphasis on the common Test Host equipment developed to support both these environments.

The Special Purpose Dexterous Manipulator (SPDM)

The SPDM Purpose and Architecture

The Special Purpose Dexterous Manipulator (SPDM) is the latest Space Robot being developed by McDonald Detwiller Space and Applied Robotics (MD Robotics, previously Spar Aerospace) for the Canadian Space Agency as part of the Canadian contribution to the International Space Station Program – the Mobile Servicing System (MSS). The main elements of the MSS

are the SSRMS, the MBS, and the SPDM. The SSRMS, a seven-degree of freedom Robotic Arm, will be used for assembly maintenance servicing of the International Space Station and Transportation of its external payloads. The MBS forms the base for SSRMS and SPDM operations and will be transported along the Space Station truss to the work site by the Mobile Transporter. The SPDM is a dual-arm robot that responds to the ISS requirement for an external Dexterous Robotic alternative to Extra-Vehicular Activity. It is a complementary and integrated component of the MSS.

The SPDM is to be controlled by astronauts within the ISS using the MSS Control Equipment (MCE). The MCE is a set of specialized hardware and software which functions with the Space Station distributed systems to provide the capability to operate the MSS components. The MCE is comprised of the Robotic Work Station itself, the resident MSS Operations Control Software, the Artificial Vision Unit and the Graphical User Interface software which runs on the Space Station Portable Computer System.

The primary mission of the SPDM is the reduction of EVA hours by robotic execution of Space Station external maintenance tasks. These tasks consist of the change out and or replacement of failed externally mounted Orbit Replaceable Units (ORU). More than 200 ORUs at various locations on the Station will be "robotically compatible" and are hence designed for robotic removal and replacement by the SPDM. These ORUs vary considerably in size and configuration, including the design of their mounting alignment systems and of the visual targets provided. A typical ORU exchange scenario involves operation of the SPDM at the outer end of the SSRMS by an astronaut located within the Space Station. The SSRMS provides power, data and video connectivity to the SPDM in addition to performing large scale positioning to enable the SPDM to reach the work site.

The SPDM consists of a main body and ORUs and will be launched as nine (9) separate components mounted on special Flight Support Equipment (FSE) integrated to a Space Lab Pallet. NASA astronauts will perform the SPDM on-orbit assembly and deployment during a nominal 6 hour EVA.

In the event of failure, the SPDM design permits faults to be isolated to individual ORUs. Faulty SPDM ORUs will be replaced on-orbit via EVA with IVA and Flight Controller support. SPDM ORUs will be available on the ground for delivery to orbit when required.

The SPDM is to be controlled by astronauts within the ISS using the MSS Control Equipment (MCE). The MCE is a set of specialized hardware and software which functions with the Space Station distributed systems to provide the capability to operate the MSS components. The MCE is comprised of the Robotic Work Station itself, the resident MSS Operations Control Software, the Artificial Vision Unit and the Graphical User Interface software which runs on the Space Station Portable Computer System.

The SPDM History

The last of the three Robotic Components of the Canadian Space Station Program, the SPDM has been developed in the shadow of the Space Station Remote Manipulator System (the Canadarm's bigger, newer cousin), and the Mobile Base System. Station redesign and delays, and budgetary constraints driven by the work on the first two Robotic Components drove the Canadian Space Agency to stop work on the SPDM in the summer of 1995. The Project was shut down for a period of 20 months. At the time of shutdown, the overall system had attained PDR level of maturity and the various components had reached different developmental milestones, such as PDR or CDR.

The need for an SPDM-like robot to ensure assembly and maintenance of the Space Station has not disappeared, however, so the CSA renegotiated the Project with MD Robotics on a Firm Fixed Price basis, and re-started the SPDM development in August 1997.

Impacts on SPDM Verification Planning

The SPDM Verification Plan had to accommodate a mix of components at various levels of maturity, with each type of component requiring a slightly different verification approach:

- a) Commonality Components - Components developed, tested and Certified for flight under the MSSP Program and reused as off-the-shelf items for the SPDM. SPDM Items common with the MSS were fully Qualified and Certified under earlier phases of the Canadian Space Station Program.

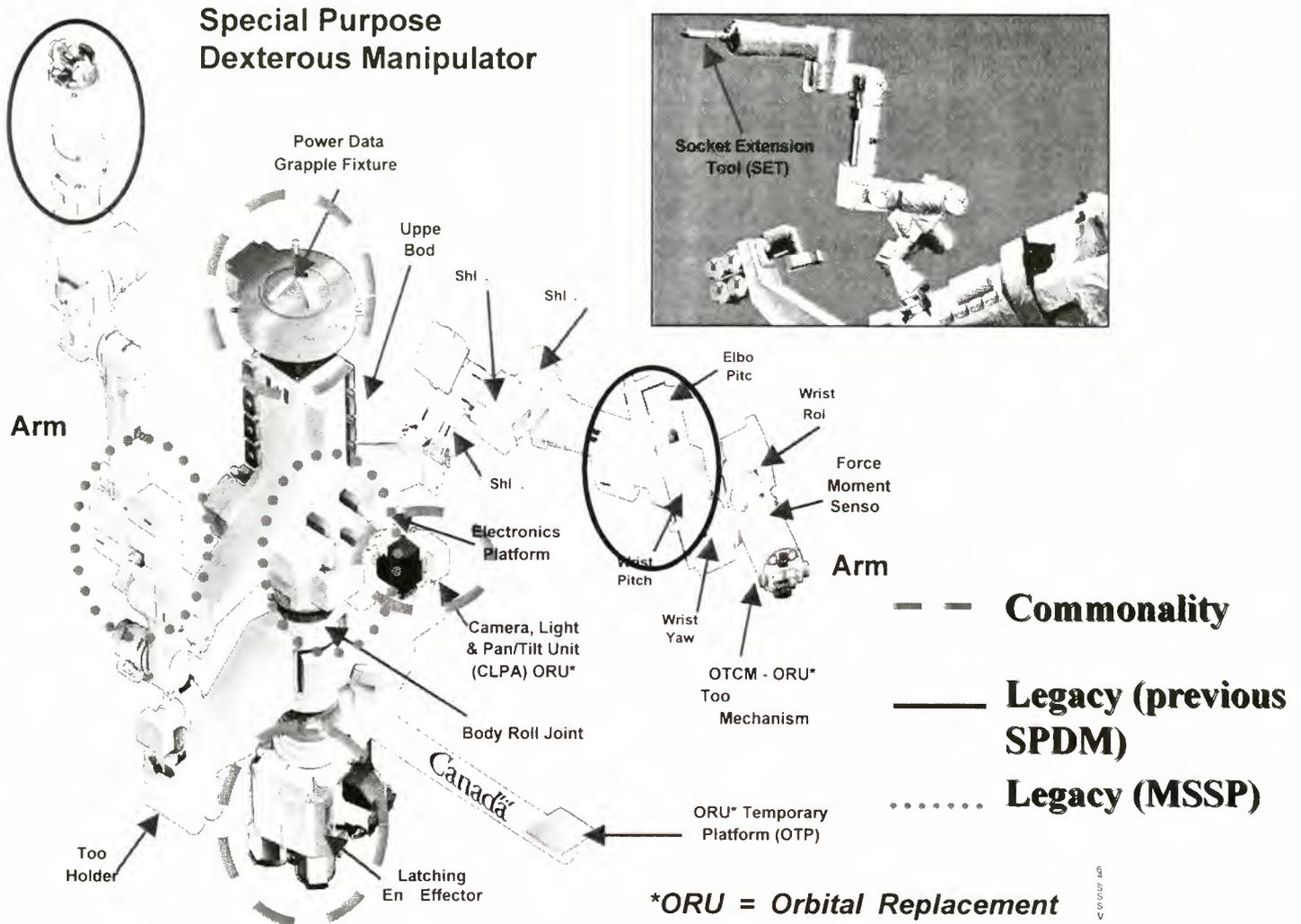


Figure 1 – The SPDM Architecture

The SPDM Program will compare by analysis the original (MSS) Certification Environment of such items and the environment that the Item will be exposed to when used in the SPDM. The Program must ensure that the SPDM environment is equal to or less severe than the one for which the MSS Item has been certified or, if the latter cannot be achieved, MD Robotics will perform re-qualification as required. This Analysis will be the only Verification Document addressing Commonality Equipment, and will be submitted in support of the Certification request for the SPDM Element. No separate Certification will be requested for the MSS Commonality equipment used on SPDM if the Analysis shows they are useable as-is.

The Verification activities for such Items consist of the Flight Item manufacturing inspection and in-process tests, and complete Acceptance Test. These

Verification activities will be performed using the MSS Program test requirements and documentation.

- b) Legacy Components - Items that were developed to CDR level under previous MSS Contracts. These Items are assumed to have successfully completed their Design Verification Phase, but the SPDM Program must confirm this. The SPDM Program will be responsible for the overall Verification Completion for the Item, and will have to complete any outstanding Design Phase Verification activities that might be found during the attempt to close the Design phase verification

MD Robotics is responsible to compare every Commonality Item's original design Environment and Requirements and the one the Item will be used in on the SPDM, and to ensure that the SPDM environment is equal to or less severe than the one for which the Item has been originally designed. The

SPDM Program is also responsible for any re-verification and re-certification of the design if any major redesign of the item is required.

- c) New Components - New designs, to be developed for the SPDM under the present Contract will undergo a full-fledged Verification Program based on Protoflight and Qualification Testing.

The SPDM Verification Planning and Tracking Tool

The Unit Specifications for the SPDM reflect its legacy. Some are 10 or more years old, created for a totally different system, with features not used on the SPDM, and therefore in most cases could not fully trace to the SPDM System Specification requirements. Others are 4-6 year old, created under the previous incarnation of the SPDM program, and some are new. The MD Robotics designers decided against re-writing all these Specifications for the SPDM.

Most off-the-shelf Verification planning tools surveyed were deemed to be too cumbersome for use on the SPDM, requiring reloading of old Specifications and of full (old) Verification Matrices and would complete traceability of these disjointedly created Specifications just to be able to start the planning effort. In order to minimize the effort, the MD Robotics Systems Team developed it's own tools in an MS Access Database, the Verification Allocation Database (VAD).

The VAD is different in it's approach to traceability and mandatory roll-up of Verification. All the SPDM Specifications produced (new, legacy and commonality) were loaded into the VAD, but traceability has been established only against the functional requirements of the SPDM System Specification, tracing them down to the lowest unit where they are actually implemented and verified. Derived "support requirements" (which complete the Unit Specifications at lower level) are not traced, nor is their Verification "rolled up" to the System level for Certification purposes. This allowed MD Robotics to trace only those requirements that are significant to the verification of the SPDM System, and to sidestep irrelevant "commonality" driven requirements from the older Specifications.

The VAD is used at all levels of the SPDM, with Subsystems or ORUs at each level of decomposition having their own VAD, and applying the same overall traceability approach below it. For the SPDM System Certification, the roll-up of Verification Completion activities is performed against the System Specification

requirements, using the traceability feature of the database.

The VAD allocates the Verification activities for every requirement not just by method (Analysis, Test, Demo etc.), but also defines additional specifics for each method, such as types of analysis, facilities for test, organization or group responsible for performing the task, etc. These allocations are then considered when defining the task load for the various engineering groups working on the Program. The VAD collects the Verification Completion Documents (e.g. Test Reports) and automatically generates the various forma Verification Closure forms required by CSA for the SPDM System Certification.

Approach to SPDM Verification

A number of new approaches to integrated Verification and Risk mitigation, such as early testing of software components integrated in the overall SPDM Control model and the use of Flight Hardware for early software-to-hardware integration have been implemented on the SPDM.

Use of Flight Hardware for initial software testing and integration

The SPDM Contract requires the manufacture of 21 Joints (three arms including one spare), as well as a Flight Equivalent Engineering Model OTCM. In order to mitigate the risk of problems during software and hardware integration at the system level MD Robotics decided to use a Joint cluster (3 joints arranged as the arm's elbow or shoulder) and the EM OTCM as an early-integration software Test Bed. The Joints will undergo only a set of Functional Tests prior to their allocation to this Quick Test Bed, and will be fully tested after the software testing is completed, and will be used for the MD Robotics SPDM Arm.

Use of the SPDM Simulation Model as a software test bed

An SPDM Simulation Model has been built with the intent to allow not only Simulation Verification of the System, but also to act as an early test bed for software modules. The functionality of the main SPDM CSCI has been modeled in a manner that allows incrementally developed individual CSCs to be "dropped" into the model and run against the previous results of the model without the real code, thus acting as an initial test bed for the code. Ultimately the flight code will replace their

respective modeled parts in the System Simulation, to create a very flight-like Simulation Model that can be used for training purposes in the future.

Environmental Testing

Since some of the major ORUs of the SPDM are tested as Protoflight components, it was extremely important to coordinate the “vertical” accumulation of environmental testing cycles up to the System level. This was accomplished in the early planning of the Verification approach and submitted to Customer (CSA) approval as part of the Program baselining.

Neutral Buoyancy and EVR Tool Testing

A full Neutral Buoyancy test was performed early on into the program at the NASA NBL facility (Figure 2), to ensure that astronauts can deploy and maintain the SPDM. The comments and inputs received from this test were incorporated into the baseline design.

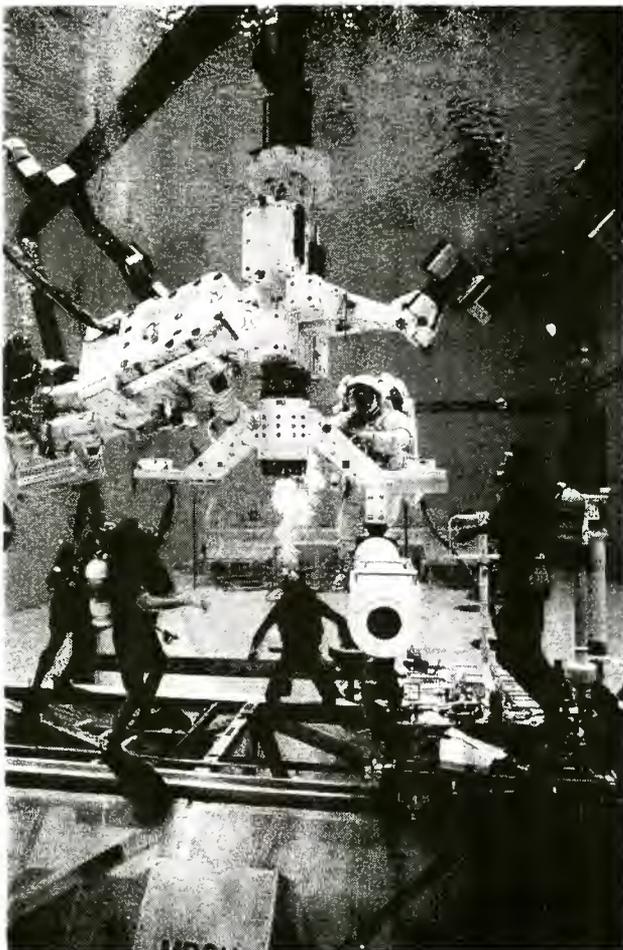


Figure 2 – The SPDM Neutral Buoyancy Lab (NBL) tests

Since the NBL test does not cover SPDM utilization, additional tests were performed with the SPDM Socket Extension Tool (SET) and with the SPDM Ground Testbed (GT). The SET compliance tests (Figure 3) concentrated on the ability of the SET to be used at marginal angles of approach to bare bolts, while the GT tests (Figure 4) are demonstrating the ability of the SPDM to operate the OTCM alone and with the SET.



Figure 3 – The Socket Extension Tool (SET) tests

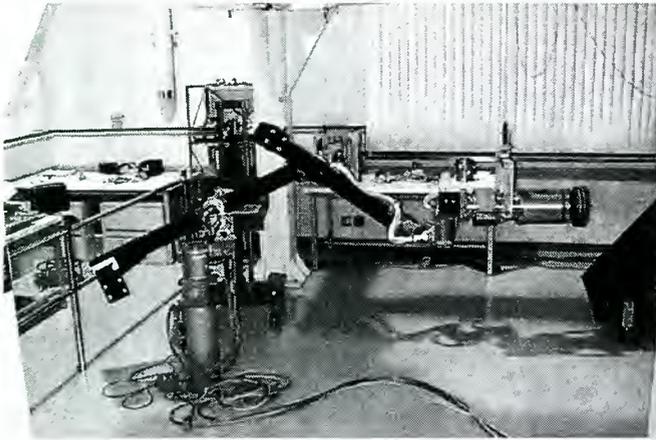


Figure 4 – The SPDM GT test setup

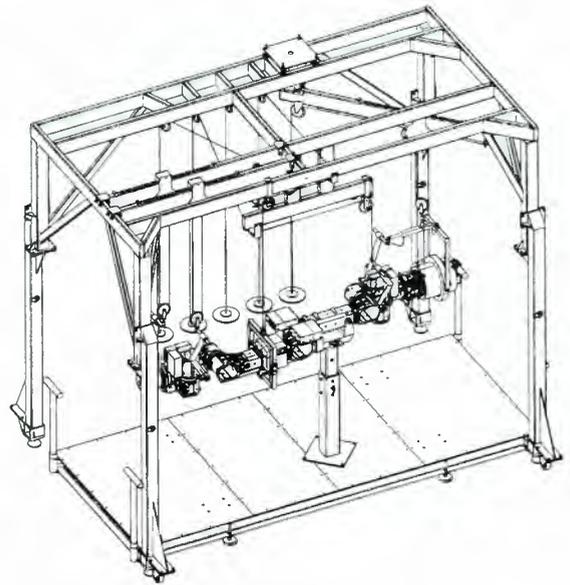


Figure 5 – The SPDM SITF test setup

SPDM Verification Facilities

The SPDM Program will use some the test beds and equipment previously developed for the MSSP, but some critical System level facilities will not be available to it due to the ongoing test activities on the other Program. Therefore it was decided that a new set of test beds would be developed:

- a) The SPDM Avionics Integration Facility (SAIF) – allowing multi-user testing of CSCIs during the development stage, as well as CSCI integration with it's hardware platform and testing of the SPDM avionics at increased levels of integration up to and including End-to-end integration with Space Station Control software components
- b) The SPDM Integration Test Facility (SITF) – allowing integration and testing of the SPDM Flight Hardware as well as SPDM hardware with Space Station hardware (the Robotic Work Station, from which the SPDM will be controlled on-orbit by the Astronauts). The SITF allows offloading of the arm weight by counter balance weights (Figure 5).

These two test beds share a common design for the Test Equipment Host, allowing the same test control software and equipment to run tests both on the software and the hardware test beds. The benefits of this approach are much lower development and maintenance costs, test script re-use, easier test equipment troubleshooting, capability to use one test crew on all System levels tests, etc.

The Incremental Verification Process

The SPDM is a Fixed Price Program, normally allowing both sides, if they wanted, to minimize the Customer's

involvement in the Verification completion and buy-off until the Acceptance Review. Since this could create unnecessary risk to the Program, the CSA and MD Robotics have agreed upon an Incremental Verification approach coupled with clear rules for re-verification and re-certification of components or the SPDM System.

In order to obtain the System Certification and Acceptance, MD Robotics must show that each requirement in the System Specification has been fully verified at all the levels to which it is applicable. The VAD accumulates the various documents containing these "proofs of verification" and automatically "rolls them up" to the System level to provide the Customer required data. The CSA is being provided with and approves Verification Status Reports at critical Program Milestones. For example at the SPDM CDR all of the stand-alone Design Verification tasks, such as Analysis, were reviewed and closed as completed, barring any unforeseen events that might require re-verification in the future (e.g. due to redesign driven by failures in test).

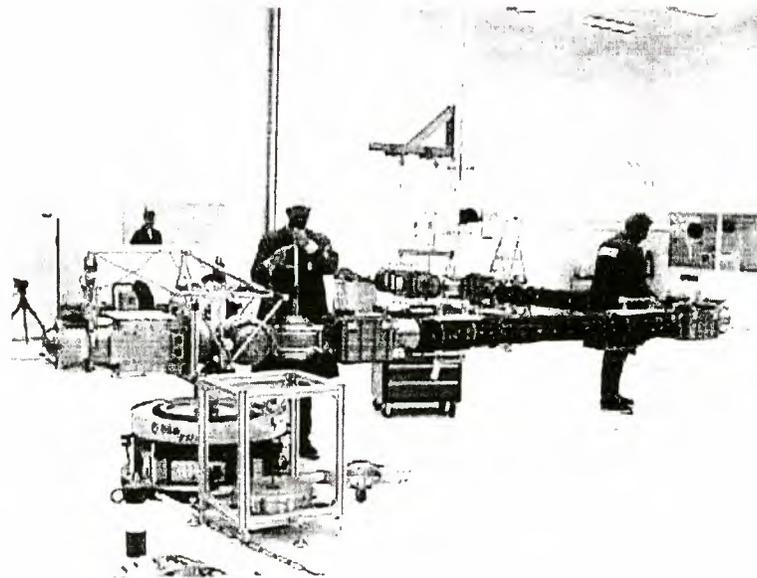
Conclusion

The SPDM Program has poses some interesting Verification challenges to both the Canadian Space Agency and to its Prime Contractor, MD Robotics, driven by the mixture of old and new components. Careful planning and good interaction between the CSA and MD Robotics teams has resulted in a viable approach to the SPDM System verification, one that ensures good verification coverage and early risk mitigation with a minimum of overhead and unnecessary cost.

ERA, the Flexible Robot Arm

Phillippe Schoonejans, ESA,
Keplerlaan 1, 2201 AZ Noordwijk, NL. pschoone@estec.esa.nl, phone +31715655026, fax +31715654437

Marc Oort, Fokker Space
Newtonweg 1, 2303 DB Leiden, NL. m.oort@fokkerspace.nl, phone +31715245448, fax +31715245499



Abstract

The **European Robotic Arm [ERA]** is being built for use on the Russian Segment of the International Space Station. The project is commissioned by ESA as part of their manned-space program, with Fokker Space as Prime Contractor, and 23 companies from 7 European countries participating in the development of the arm. The ERA is scheduled to be launched by Space Shuttle to the ISS early in the next millennium. The system CDR was held at the same time as the I-SAIRAS conference. This paper focuses on the operational capabilities of ERA, ranging from fully automatic (monitored by a cosmonaut) to fully manual.

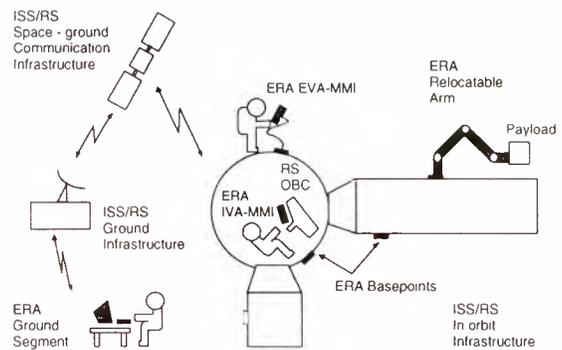


Fig 1: The ERA system

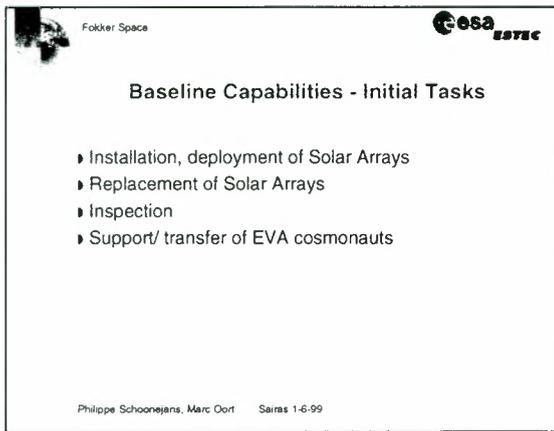
1. Introduction

The ERA system (Figure 1) consists of an arm, an EVA Man Machine interface, an IVA Man Machine Interface, a Refresher Trainer [RTR] and a Mission Preparation and Training Equipment [MPTE]

The ERA arm is a 11 meter, 6 Degree-of-freedom arm, whose most striking feature is the ability to cover large distance on the ISS by “hopping” from on basepoint (which supplies the power and communication interface) to another. Table 1 shows the ERA key performance parameters

Parameter	Required performance
positioning open loop	< 40mm, 1° [all axes]
positioning closed loop	< 5 mm, 1° [all axes]
maximum speed	0.2 m/s
maximum transportable mass	8000 kg
clearance	0.9 m
stopping distance	0.15 m
outer loop control frequency	20 Hz
inner loop control frequency	300 Hz

Table 1: ERA required performance

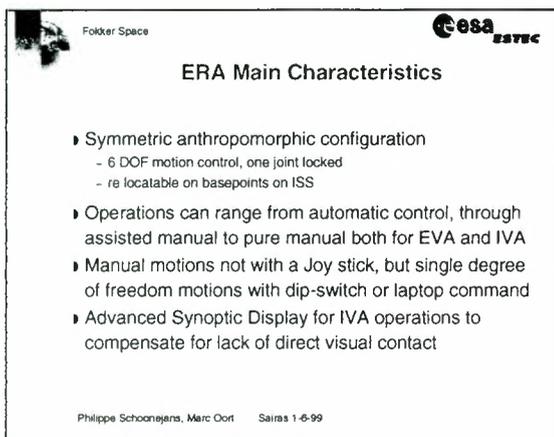


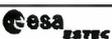
Fokker Space 

Baseline Capabilities - Initial Tasks

- ▶ Installation, deployment of Solar Arrays
- ▶ Replacement of Solar Arrays
- ▶ Inspection
- ▶ Support/ transfer of EVA cosmonauts

Philippe Schoonejans, Marc Oort Sairas 1-6-99

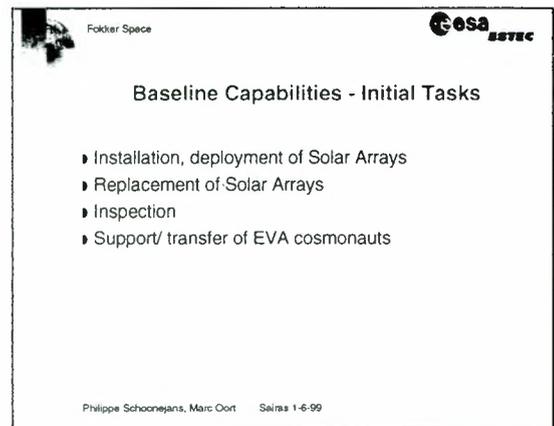


Fokker Space 

ERA Main Characteristics

- ▶ Symmetric anthropomorphic configuration
 - 6 DOF motion control, one joint locked
 - re locatable on basepoints on ISS
- ▶ Operations can range from automatic control, through assisted manual to pure manual both for EVA and IVA
- ▶ Manual motions not with a Joy stick, but single degree of freedom motions with dip-switch or laptop command
- ▶ Advanced Synoptic Display for IVA operations to compensate for lack of direct visual contact

Philippe Schoonejans, Marc Oort Sairas 1-6-99



Fokker Space 

Baseline Capabilities - Initial Tasks

- ▶ Installation, deployment of Solar Arrays
- ▶ Replacement of Solar Arrays
- ▶ Inspection
- ▶ Support/ transfer of EVA cosmonauts

Philippe Schoonejans, Marc Oort Sairas 1-6-99

2. Operational design drivers

One of the important design drivers for any space-based robot arm is that it should significantly reduce the strains associated with operations during EVA, or even to do away with EVA operations altogether. This has as a consequence that the active operator involvement in operating the arm should be minimized as much as possible, while leaving a maximum of monitoring and intervention capability. The operational flexibility, coupled with the limitations of the processing and memory resources which are common in space applications due to the severe environmental conditions, pose a great challenge to the functional design of the arm itself and the controlling Extra Vehicular Man-Machine Interface (EVA MMI).

Additionally, it is also possible to operate ERA from inside the ISS, using a dedicated IBM 760 laptop, the IVA MMI. This laptop has more monitoring and commanding capabilities than the EVA MMI, but has been designed to resemble the latter as much as possible, this to insure commonality in operations from the IVA MMI and EVA MMI.

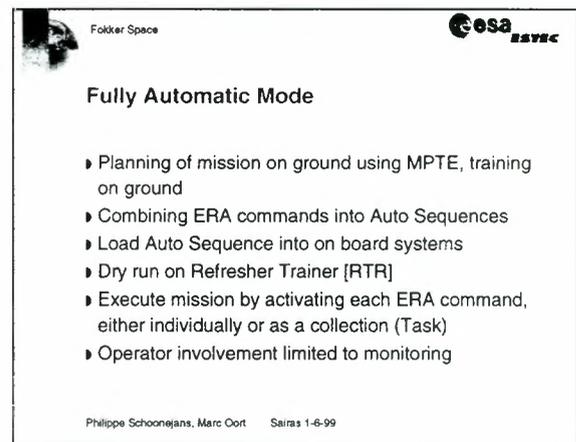
A layout of both the EVA MMI and IVA MMI can be found in the appendix of the paper

It is possible to operate ERA in three operational modes:

- Fully Automatic mode, using Auto Sequences
- Partially Manual mode, supported by Mini Auto Sequences [MAS]
- Fully Manual mode

Each of these three will be addressed below.

3. A flexible arm: Pre-planned Operations in Fully Automatic Mode



Fokker Space 

Fully Automatic Mode

- ▶ Planning of mission on ground using MPTE, training on ground
- ▶ Combining ERA commands into Auto Sequences
- ▶ Load Auto Sequence into on board systems
- ▶ Dry run on Refresher Trainer [RTR]
- ▶ Execute mission by activating each ERA command, either individually or as a collection (Task)
- ▶ Operator involvement limited to monitoring

Philippe Schoonejans, Marc Oort Sairas 1-6-99

As part of planning an ERA mission, the ground segment can combine all commands to the ERA Control Computer, and the characteristics of up to four payloads into one or more "files", and load these into the on-board computer as dedicated *Auto Sequences*. The limitations in CPU capabilities prevent the Auto Sequences from acting as a pseudo programming language, with conditional branching or looping). The correctness of these sequences has been verified on the ground, and the supervisory task of the operator is thus limited to starting actions and to monitor the general progress of the mission. To compensate for the lack of direct visibility, the IVA operator has TV cameras and an (IVA MMI generated) geometrical world model available for monitoring. The operator has the choice to either explicitly activate each single command in an Auto Sequence, or the command the ERA to automatically a set of commands in sequence (a Task). At any time, the operator can take over command and continue manually, after which he can resume the Auto Sequence.

Below is an example of a Task in an Auto Sequence (Table 2). The first task could typically be executed at task level, the second one at action level. A snapshot of

the procedure for the EVA MMI cosmonaut, in the ISS Standard Ops Data Format is given in Table 3.

Task 13 Preparatory

113 Set Max Speed Discrete
 325 Check Active Basepoint
 328 Check Initial Pose
 114 Set Payload Class
 331 Download Frames
 200 Go to Controlled Hold
 304 Load Working Data Base
 305 Enable New Data Base
 323 Select FOR
 300 Select TAF (Target BP, or PMU)
 322 Direct Bus Command (Wrist TFS ON)
 4000 Select Viewpoint

Task 12 Transfer:

107 Free Move
 108 Single Joint Move
 126 Pitch Move
 203 Go to Standby
 206 Go to Shoulder Yaw Joint Hold
 108 Single Joint Move
 203 Go to Standby
 200 Go to Controlled Hold

Table 2: An example of a part of an Auto Sequence

ER - EMMI 'MENU'
 Scroll 'SEL AS' to Center Window, then to Left Window
 ✓ '01' in Center Window
 Scroll '01' TO Left Window
 Verify 'START' LED flashing
 Press 'START' Switch for 2+ seconds
 ' ERA STATE'
 ✓ Verify 'INFO DUMP'
 After ~ 40 seconds, ✓ 'PL move' in Left Window (Name of AS)
 ✓ 'TOP' in Top Window
 ✓ 'T013 01' in Center Window
 ✓ 'T018 02' in Bottom Window

Table 3: Example of an operational procedure

4. A flexible arm: Operations in Partially Manual Mode

It is also possible to operate the ERA without the help of pre-planning by the ground. These operations use hard-coded Auto Sequences inside the ERA ECC, each of which are designed to complete a partial mission objective (see Table 4.

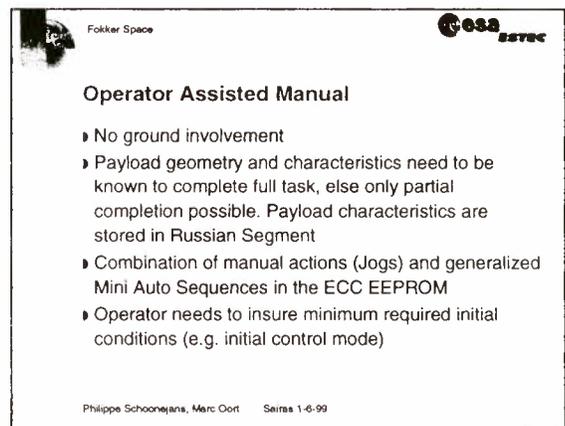
6	ATTACH_BP	attach to basepoint
7	ATTACH_GF	attach to grapple fixture
8	DETACH_BP	detach from basepoint
9	DETACH_GF	detach from grapple fixture
10	INSTALL	install PL on PMU
11	REMOVE	remove PL from PMU
12	SCLU OFF	emergency CLU off
13	ESCLUOFF	emergency CLU off
14	INSPECT	inspect surface

15	POPIN	popin screwdriver
16	FIRMFLEX	firm/flex TFS

Table 4: Available Mini Auto Sequences in ERA

The commands inside these Mini Auto Sequences are designed to be independent of the desired position for the arm and/or its payload. A selectable list of hard-coded coordinates of basepoints and payload mounting units, allows the operator to specify the final position without having to determine or type in coordinates. If a position is not in the list, the ERA can "build" a target position from the end-effector camera's image of a standard reflective pattern which is located next to each Grapple Fixture.

The ERA Control Computer, and its counterpart in the IVA MMI, also keep track of the position of payloads moved within its model of the space station, thus keeping the on-board collision avoidance routines up to date, provided the objects are moved using ERA.



5. A flexible arm: Unplanned Operations in Fully Manual Mode

If even the hard-coded sequences are not sufficient or usable, it is possible to move the arm manually using keys and/or switches (no joy stick). Both long motions (up to 10 meters) or short steps (10 mm, 1 degree) are possible. Motions can be commanded both in a frame of reference coupled to the arm, and one fixed to the ISS. As the motions of the arm happen under full control/responsibility of the operator, the rotations have been limited to one degree of freedom at a time. The arm can even be operated in this manner when the Russian Segment Central Post Computer (which is in charge of the communications between the MMIs and the ERA) has failed. The EVA MMI can be connected directly to the ERA bus, and commanded from there. If all else fails, mechanical overrides (using EVA) are possible as well, requiring no SW at all.

Fokker Space 

Pure Manual

- ▶ Approach, transfer and retract by means of single degree-of-freedom jogs
- ▶ Custom-made coordinate frame definition using ECC functionality
- ▶ Insertion of End Effector or Latch interface by manual action with ERA in Yield mode
- ▶ Manual (un)grasping using dedicated tools

Philippe Schoonejans, Marc Oort Sairas 1-6-99

6. Conclusions

The ERA has been designed to cope with a wide range of operational conditions, from fully planned to unplanned manual contingency operations, thus making a truly flexible robot arm. Astronaut reviews of the MMI designs have shown that the ERA is operable and user friendly, at least as seen in simulated conditions. Man-in-the-loop testing planned next year will show to what extent this is true for the full system when a complete mission has to be executed.

Fokker Space 

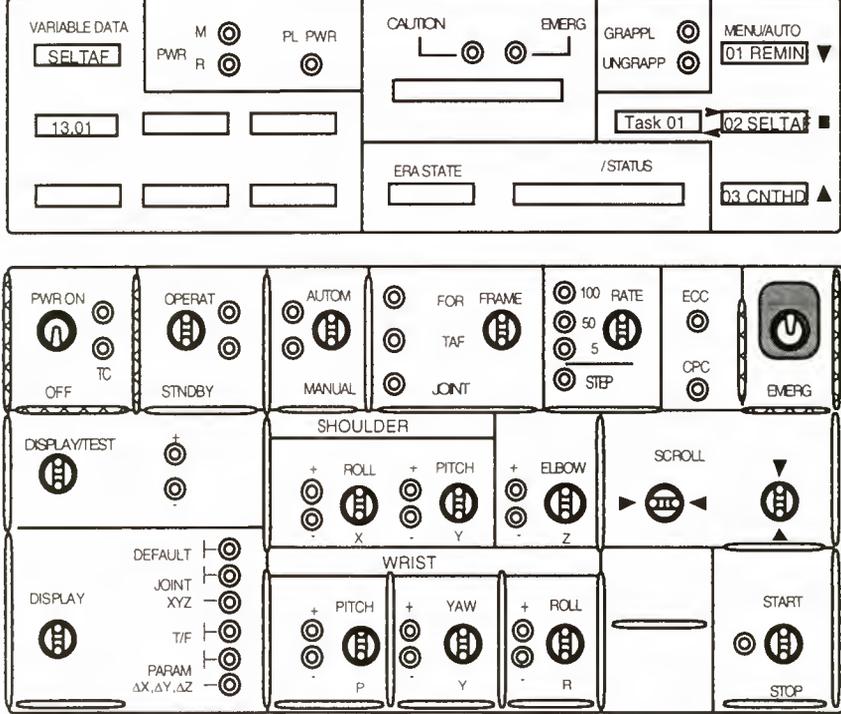
ERA, the flexible arm

- ▶ Operations can range from automatic control, through assisted manual to pure manual both for EVA and IVA
- ▶ Advanced Synoptic Display for IVA operations to compensate for direct visual contact
- ▶ Sufficient commonality with other ISS arms to allow easy familiarization (e.g. coordinate frames, terminology,...)
- ▶ The generic capabilities provide the opportunity for additional future tasks on the Russian Segment of the ISS

Philippe Schoonejans, Marc Oort Sairas 1-6-99

Annex: Man Machine Interface Layout

Fokker Space **ERA : EVA MMI Layout** 

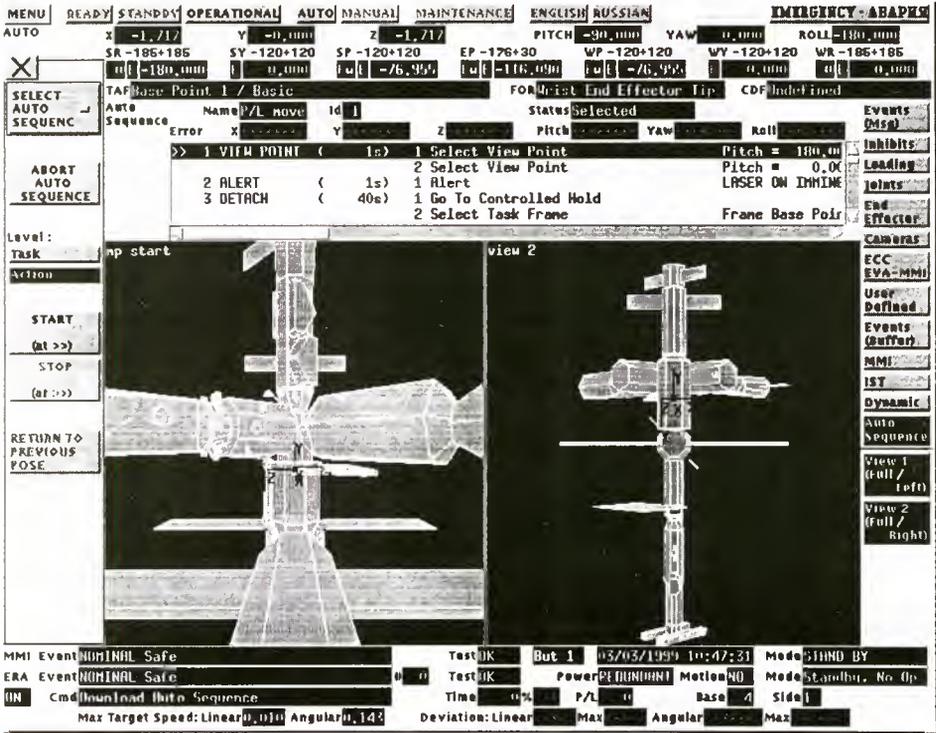


The diagram illustrates the ERA EVA MMI layout, organized into several functional sections:

- Top Panel:** Includes 'VARIABLE DATA' (SEL TAF), power controls (M, R, PL PWR), status indicators (CAUTION, EMERG), and menu options (GRAPPL, UNGRAPP, MENU/AUTO 01 REMIN, Task 01, 02 SEL TAF, 03 CNTHD).
- Control Row:** Features power (PWR ON/OFF, TC), operation (OPERAT, STNDBY), automation (AUTOM, MANUAL), frame (FOR, FRAME, TAF, JOINT), rate (100, 50, 5, STEP), emergency (ECC, CPC, EMERG), and a power button.
- Display/Shoulder Section:** Contains DISPLAY/TEST, SHOULDER (ROLL, PITCH, ELBOW), and SCROLL controls.
- Wrist/Display Section:** Includes DISPLAY (DEFAULT, JOINT XYZ, T/F, PARAM ΔX, ΔY, ΔZ), WRIST (PITCH, YAW, ROLL), and START/STOP buttons.

Philippe Schoonejans, Marc Oort Sairas 1-6-99

Fokker Space **ERA: IVA MMI layout** 



The ERA IVA MMI layout displays a comprehensive interface for the IVA (Intra-Vehicular Activity) task. It features a data table, a task list, and two 3D views of the robotic arm.

Menu	READY	STANDBY	OPERATIONAL	AUTO	MANUAL	MAINTENANCE	ENGLISH	RUSSIAN	EMERGENCY - ABAPK
AUTO	X -1,717	Y -0,000	Z -1,717	PITCH -90,000	YAW 0,000	ROLL -180,000			
	SR -185+185	SY -120+120	SP -120+120	EP -176+30	WP -120+120	WY -120+120	WR -185+185		

Task List:

Sequence	Name	P/L	Mode	Id	1	2	3	4	5	6	7	8	9	10
1	VIEW POINT	(1s)	1	Select View Point	Pitch = 180,00								
2	ALERT	(1s)	1	Alert	Pitch = 0,00								
3	DETACH	(40s)	1	Go To Controlled Hold	LASER ON INTIME								
				2	Select Task Frame	Frame Base Pair								

3D Views: 'view start' and 'view 2' show the robotic arm in different orientations.

MMI Event Log:

MMI Event	NOMINAL Safe	Test OK	But 1	03/03/1999 10:47:31	Mode	STNDBY
ERA Event	NOMINAL Safe	Test OK	Power	RETURN	Motion	NO
MMI	Cmd	Unload Auto Sequence	Time	00:00	P/L	0
			Base	4	Side	1

Max Target Speed: Linear 0,00 Angular 0,142 Deviation: Linear Max Angular Max

ASSEMBLY OF LARGE SPACECRAFT: THE XEUS MISSION

F. Didot¹, M. Bavdaz¹, J. Schiemann¹
 U. Knoop²
 H. Petersen³

¹ ESA, ESTEC, Postbus 299, 2200AG Noordwijk, Netherlands
 Tel. : +31 71 565 4403 Fax : +31 71 565 5419

² Daimler Chrysler Aerospace, Bremen, Germany

³ Fokker Space, Leiden, The Netherlands
fdidot@estec.esa.nl

Abstract

XEUS represents the next logical step forward in X-ray astrophysics after the current set of missions have been launched and completed their operational lives. The scientific objectives of XEUS are so demanding that the mission probably represents significant technological challenges compared to past astrophysics missions. The development and ultimate success relies heavily on the capability of the International Space Station (ISS) to enable large spacecraft assembly. In this paper we describe the key characteristics of the mission, and the XEUS assembly sequence at ISS is described and finally robotic technology development needs are identified for enabling the XEUS mission.

1. Introduction

XEUS : The **X**-ray **E**volving **U**niverse **S**pectroscopy mission represents a potential follow-on mission to the ESA XMM cornerstone currently nearing completion. The XEUS mission was considered as part of ESA's Horizon 2000 plus program within the context of the International Space Station (ISS). The original mission concept has arisen through extensive discussions by the European Scientific Community particularly at the Workshop held at Leicester University UK in July 1996. At this international workshop the foundations for the "Next Generation of X-ray Observatories" was laid. With XMM due for launch in early 2000, with a mission duration of 5-10 years, it is not too early to consider the post XMM era. At the turn of the century two great X-ray observatories will have embarked on their astrophysics programs – XMM and NASA's AXAF and thus the requirements of XEUS must take account of the key thrusts and potential discoveries from both these powerful missions.

The XEUS mission aims to place a permanent X-ray observatory in space with a telescope aperture equivalent to the largest ground based optical telescope currently built to date – essentially the equivalent of the

Keck Observatory for X-ray astronomy in space. By making full use of the facilities available at the ISS in the next century and by ensuring in the XEUS design a significant growth and evolution potential the overall mission lifetime of XEUS could be well over a quarter of a century. The power of this observatory will be such that for the first time detailed imaging spectroscopy studies in high energy astrophysics of objects associated with the evolution of the early universe will be undertaken.

2. The Scientific Rationale for XEUS

The aim of XEUS is to study the astrophysics of some of the most distant and hence youngest known discrete objects in the universe. The specific scientific issues, which XEUS aims to address, can be summarized as follows :

- To measure the spectra of objects with a redshift $z > 4$ at flux levels below 10^{-17} erg cm^{-2} s^{-1} . Note this is at least a 100 times fainter than XMM.
- To determine from the X-ray spectral lines the redshift and thus age of these objects
- To thereby establish the cosmological evolution of matter in the early universe

To achieve these demanding aims a large X-ray telescope will need to be developed. An X-ray mirror with an effective collecting area at 1 keV (~ 1 nm) of 30 m^2 . This calls for a large collecting area, in effect a large mirror of the order of 10 meters in diameter. On one hand such mirror dimension can not be accommodated within actual launch fairing. On the other in-orbit deployment techniques can not be used to obtain such mirror: it would be too complex and the deployment itself too inaccurate. To solve this dilemma, it was proposed to make use of the International Space Station as an assembly base for the XEUS Mirror completion.

Thus the XEUS mission is conceived in a two essential mission build-up phases:

- A first phase during which the two XEUS constituent spacecraft are launched: the mirror spacecraft and the detector spacecraft.
- A second phase during which the XEUS mirror spacecraft visits the International Space Station for growth.

3. The XEUS Mission Profile

The XEUS spacecraft consists of two free flying spacecraft: a detector spacecraft (DSC) and a mirror spacecraft (MSC) separated by ~ 50 m and aligned in the low earth orbit by an active orbital control and alignment system. In the current baseline scenario it is envisaged to launch the “zero growth” XEUS mated pair (MSC1+DSC1) directly into a Fellow Traveler Orbit (FTO) to the ISS using an Ariane 5 or similar launcher. The FTO is a low earth orbit, altitude ~ 600 km with an inclination similar to the ISS. The mated pair will decouple in FTO and the DSC1 will take up station 50 m from the MSC1. After alignment validation and normal spacecraft/payload checkout the “zero growth” astrophysics observation program can begin. The MSC will point at a given target field and maintain a stable attitude while the DSC1 will maintain the focal distance and alignment with respect to the MSC1 so that the field image as measured by the DSC1 detectors remains stable.

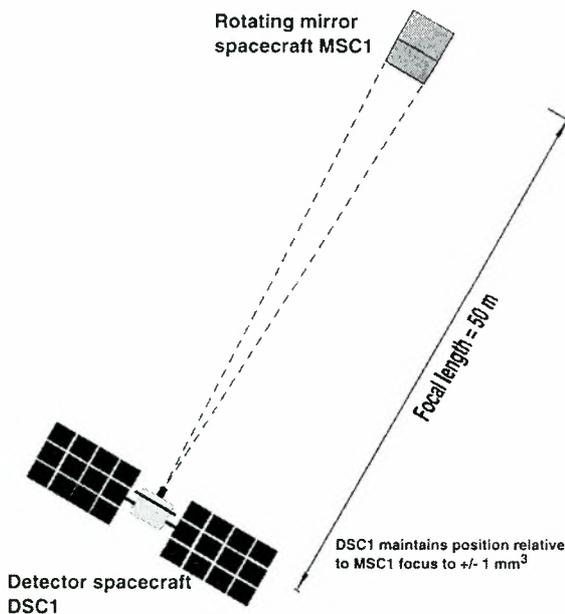


Figure 1; Detector Spacecraft – Mirror Spacecraft relative orbit station keeping

It must be stressed that XEUS is in this scenario completely autonomous from the ISS. In the initial launch “zero growth” configuration the MSC1 will contain only the two inner annuli of the telescope filled with 32 petals. Despite the XEUS initial collecting area is huge (~6 m² at 1 keV) a growth capability is crucial for achieving the ultimate scientific goals of the mission.

The pair of XEUS spacecraft can dock in orbit and, through the use of the orbit control system (OCS) on DSC1, the pair can perform an orbit change and come to the vicinity of the ISS. At this point the two docked S/C pair are able to wait for up to one year, following the ISS from a safe distance, until the Mirror sectors are uploaded by Shuttle to the ISS. Prior to the docking to the ISS, the DSC1 will separate from the MSC1, and then the MSC1 will approach and dock to the ISS. As far as possible, docking technology developed for the Automated Transfer Vehicle (ATV) will be used. The DSC1, after de-docking from the MSC1 the DSC1 will undergo a controlled de-orbit.

4. The Mirror Spacecraft (MSC1)

A key characteristic of XEUS is the large X-ray mirror aperture. The MSC capitalizes on the successful XMM mirror technology and the industrial foundations, which have been already laid in Europe for this program. Unlike XMM however, where a heavily nested mirror was fabricated from closed shells, the XEUS mirror is divided into annuli, with each annulus sub divided into sectors. The initial mirror aperture is dictated by the fairing diameter, that is 4.4 meters in outer diameter. At the center of MSC, a diameter of 1.2m is reserved for the ISS docking port accommodation and S/C avionics. To reach the science goals, additional mirror surface needs to be added. The ISS is used as an assembly base to complete on-orbit the mirror spacecraft by adding new sectors to it. That growth is essential to push the spectroscopic limits to the highest redshifts and therefore the youngest objects.

Because mirror petals are very sensitive to contamination each petal will be protected from contamination during launch, maneuvering and docking with the DSC1 and while in the vicinity of the ISS by hermetically sealed doors. The operating temperature of each petal must be maintained constant so as to prevent deformations of the highly accurate mirror plate surfaces. While each petal contains an integrated stray light and thermal baffle as part of the unit, the MSC1 will operate as a spinning spacecraft rotating about its major axis at 1 degree/second. This will ensure a uniform temperature around the circumference of the spacecraft. In addition the MSC1/2 contain large thermal baffles to shield the mirror from direct sunlight.

The MSC will be flying in low earth orbit but will not actively control its orbit during observations. The MSC1 will have a complete AOCS, compatible with the requirements for docking to the ISS. Major orbit changes, e.g. for visiting the ISS, will be performed using the DSC1 OCS after docking to the DSC1. The pointing direction of the MSC1 will be restricted due to stray light, thermal and power reasons. The angle between the telescope pointing and the sun-vector will always be in the range $90 - 120^\circ$ during the observation phase of the mission.

Figure 2 shows the MSC1 in cross section.

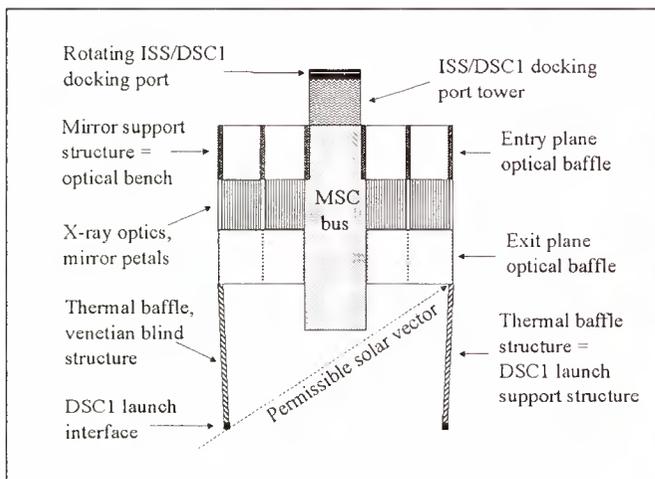


Figure 2: MSC1, cross section

5. The Detector Spacecraft (DSC1)

The DSC1 will be tracking the focus of the X-ray telescope on-board the MSC1 to within ± 2 mm. This implies that the DSC1 will be flying in a non-Keplerian orbit. The orbital characteristics of this MSC1-DSC1 tandem pair is summarized in Table I.

Table I :Orbital Characteristics of the DSC/MS

Parameter	Specification
Altitude	600 km
Eccentricity	0
Inclination	51.6 deg
Period	97 min
Maximum eclipse	35.5 min
Node spacing	24 deg
Node precession	-4.5 deg/day
De-orbit dv	256 m/s
Altitude change dv	0.54 m/s/km
Plane change dV	132 m/s/deg
Earth angular radius	66 deg
Range to horizon	2831 km

Figure 3 shows a perspective view of DSC1 looking at the anti-solar side, which shows the instrument thermal radiators.

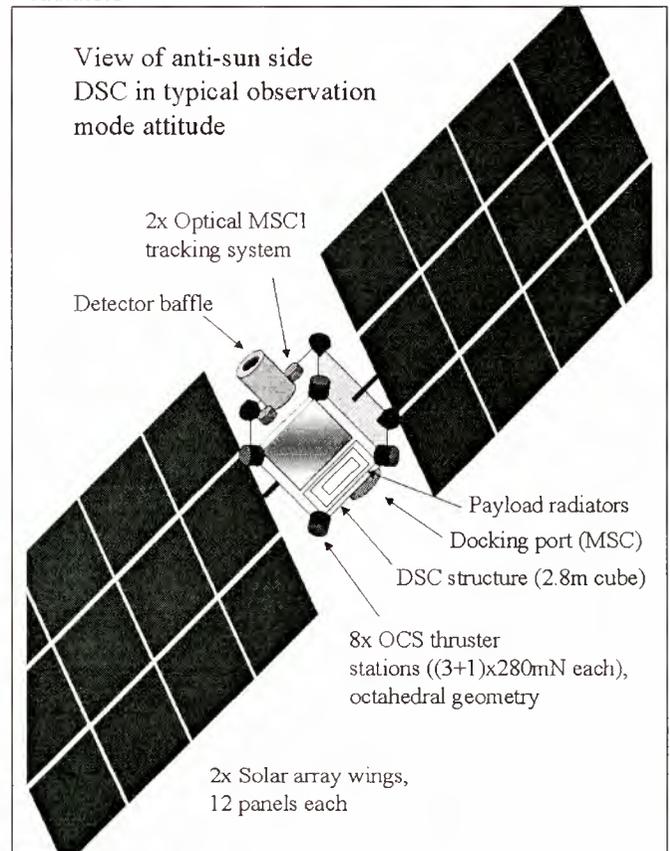


Figure 3: DSC1, perspective view

The payload of DSC1 will include three X-ray imaging spectrometers. This payload has not been subject to a detail study.

6. The Activities at the ISS

In the current baseline scenario it is envisaged that the mated XEUS pair (MSC1 +DSC1) will arrive in the vicinity of the station from FTO and the MSC1 will then dock at the ISS using the same docking port as ATV on the Russian segment. This rendezvous with the ISS, which will take advantage of the evolution of the XEUS orbit with respect to the ISS, will occur on a timescale of $\sim 4-5$ years after initial launch of MSC1 and DSC1. At the ISS the MSC1 is grown to MSC2.

The top level activity envisaged at the ISS after the MSC1 has docked with the ISS can be summarized as follows:

- Insert the 8 sectors into MSC1 mirror support structure. Note that each mirror sector is ~ 1700 kg. Figure 4 shows a mirror sector extracted from the transport container.

- Deploy the thermal baffle elements to the MSC2 as indicated in figure 7.
- Perform checkout of MSC2.
- De-dock MSC2 and move it to the ISS safety perimeter
- Transfer MSC2 orbit to FTO.

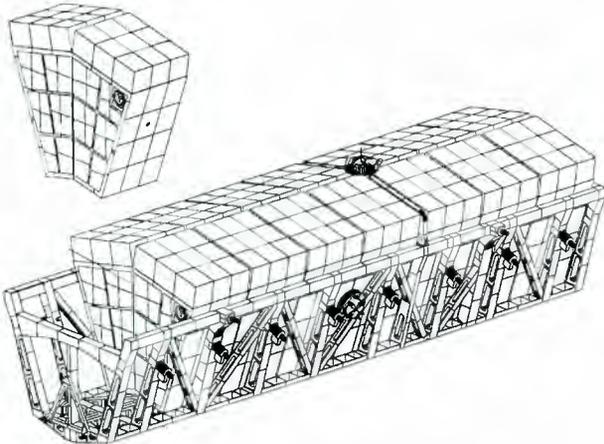


Figure 4: Mirror Sector Transport Container for eight mirror sector

6.1 Preparatory Activities

Likewise the European ATV, the MSC1 will dock to the Russian service module. The service module will be, at the time of MSC1 visit, the only site at ISS that can accommodate a large and heavy S/C such as MSC1 for a period of up to 2 months and offering sufficient free area around the S/C for its assembly. It is also the only module equipped with reflective targets used for ATV docking.

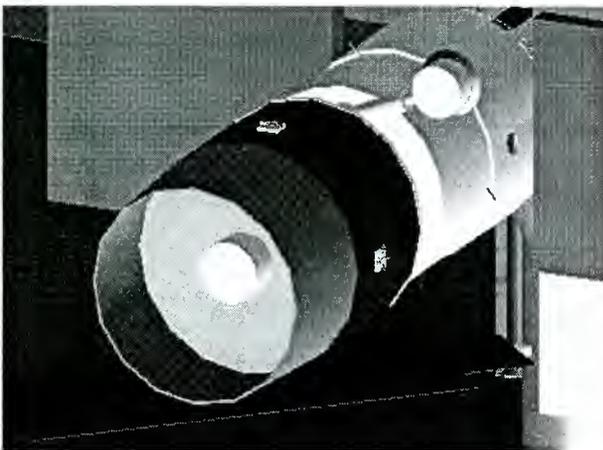


Figure 5: MSC 1 docked at the Service Module

The European Robot Arm (ERA) will be used to support the mirror sector assembly activities that will take place on the Russian Segment side. ERA is stored on the science power platform. To support the assembly task, an ERA basepoint on the service module will be installed. Fortunately the necessary fixation interface is

available on the external diameter of the service module, and can be used to connect an ERA basepoint

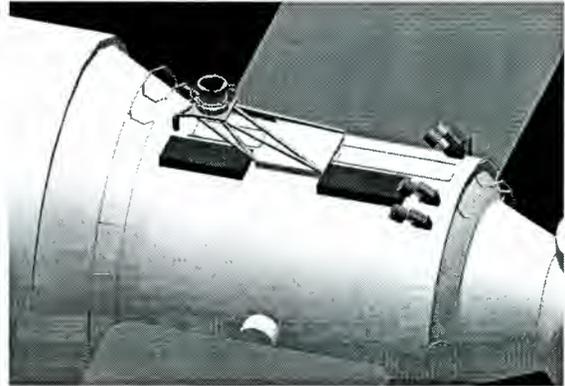


Figure 6: ERA basepoint attachment concept on the Service Module

Once the MSC1 has docked to the service module, the ERA can relocate onto the service module basepoint and a visual inspection of the MSC1 can be done to assess its general status.

Closely after the MSC1 docking to the ISS, the 8 mirror sectors will be brought to the ISS by a single Space Shuttle launch. The required 8 mirror sectors will be accommodated on one dedicated transport container (TC). That transport container will be handed over by the Shuttle RMS to the Space Station RMS using the nominal payload hand-off procedure.



Figure 7: Transport Container handed over by the Shuttle RMS to the SSRMS

A power and data grapple fixture (PDGF) will be mounted on the TC such the SSRMS can deliver survival power to the mirror sectors. The SSRMS will transfer the transport container to the Z1 truss. Z1 truss is the only area in ISS where to a container of such size (nearly 12 meters in length) and mass (nearly 15 tons) can be stored. Once the transport container is mechanically and electrically mounted on Z1 structure, the SSRMS will relocate to the Zarya module (ex FGB).

At this stage, all preparatory tasks are completed and the MSC1 upgrade to a MSC2 can start.

6.2 The Mirror Sector Assembly Activities

From its basepoint on Zarya, the SSRMS will get the transport container stored on Z1 and transfer it to near the service module, at a hand-over pose.

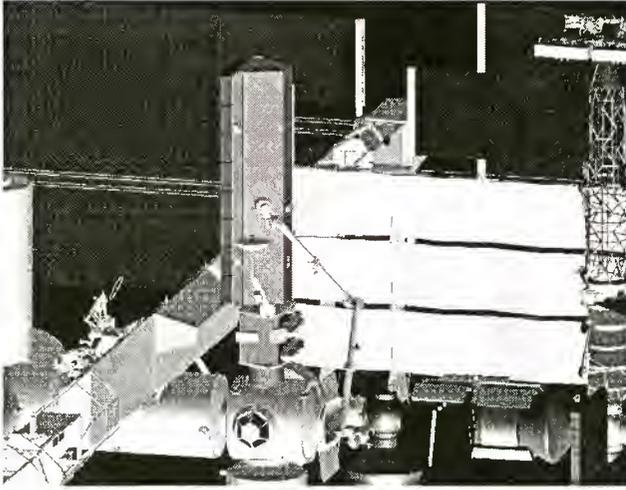


Figure 8: SSRMS relocated on ZARYA PDGF gets the TC for transfer to the Service Module vicinity

Once arrived at its hand-over pose, the SSRMS drives will be disabled, and its brakes applied. ERA will then transfer to the hand-over pose, and get from the transport container held by the SSRMS a mirror sector.

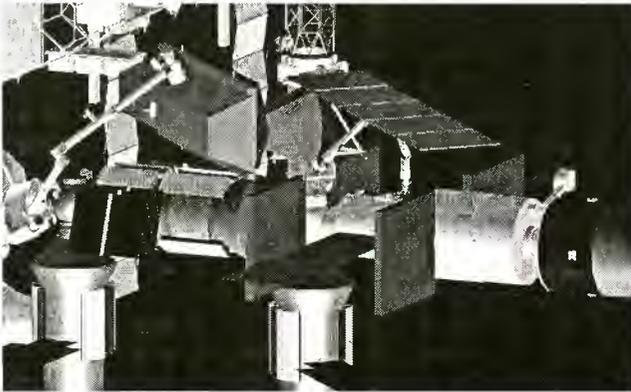


Figure 9: ERA extracts the first mirror sector from the transport container

The mirror sector will be extracted by ERA from the transport container, transferred to MSC1 and installed on the MSC1.

ERA will actuate thermal “knives” releasing a split bolt mechanism that will mechanically attached the mirror sector to the MSC. Once the safety of the mechanical fixation of the mirror sector has been verified, ERA will release the mirror sector. The mirror sector, which is mounted on a rotary structure, will then be rotated by 180 degrees. A new mirror sector slot is presented, for

the next mirror sector engagement. That rotary structure is essential to install all mirror sectors by ERA based on one basepoint only.

Once the last mirror sector is installed on the S/C, the transport container is stored back on the Shuttle for return to ground.

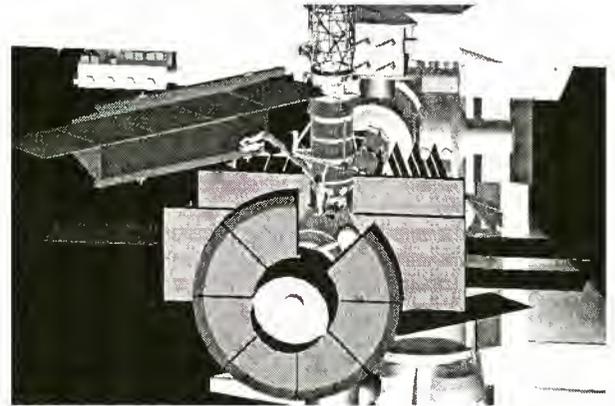
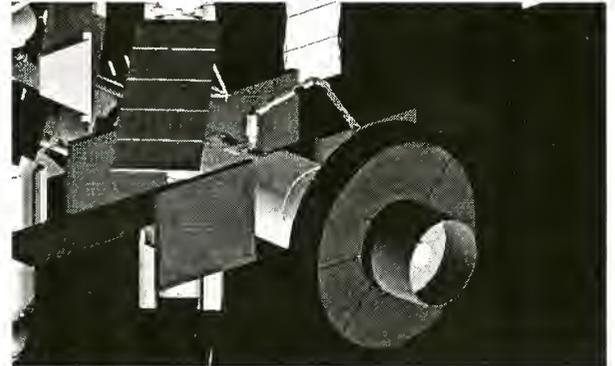


Figure 10&11: Extracting the last mirror sector



from the TC and placing it on the MSC2

The MSC2 assembly is near complete. Electrical connections of the mirror sectors to the S/C still need to be performed by EVA, with the support of ERA.

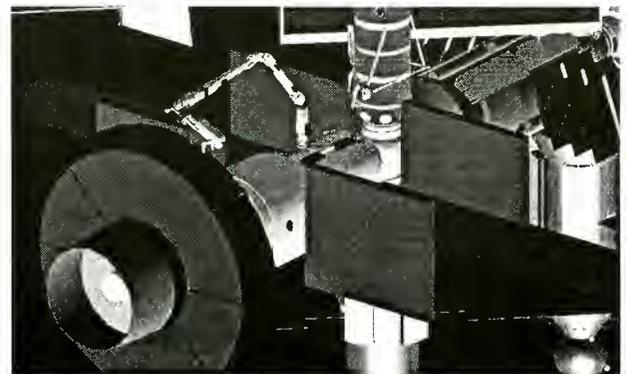


Figure 12: Electrical connection by EVA with support from ERA

Finally the thermal baffles will be deployed, and interconnected at their end by EVA to rigidify the overall structure. Once completed, the MSC2 status is overall checked. This terminates all ISS related support

activity. The MSC2 is ready for un-docking from ISS and rendez-vous with the new DSC.

7. Conclusion

The Xeus mission baseline assembly scenario was presented. This baseline strategy allows Xeus to take advantage of the ISS as an in-orbit assembly facility, while minimizing the complexity of the tasks and resource demands required of the ISS. The Xeus mirror spacecraft completion is feasible within one shuttle upload to ISS, and the required assembly time will fit within a Shuttle visit to ISS. This enables to minimize the mission build-up impact on the required ISS resources as assembly can take place in less than 11 days.

The XEUS mission represents an ambitious project full of new approaches and technologies. At the spacecraft and mission level a number of technical and logistical issues need to be addressed: For the robotics part, the following technology area needs to be investigated in near future:

- Dynamic & control interaction between the SSRMS and the ERA
- Approach and insertion strategy for the mirror sector onto the MSC2
- Launch attachment mechanism for the mirror sectors in the transport container
- In-orbit fixation mechanism for the mirror sector on the MSC2

After this initial analysis of the in-orbit assembly of the Xeus mirror spacecraft, no serious technology showstopper could be found. The ISS proves to be an enabling infrastructure to support ambitious new space science missions.

Rover Systems (1)

DESIGNING OF LUNAR ROVERS FOR HIGH WORK PERFORMANCE

Junji Aizawa, Nobuto Yoshioka, Makoto Miyata, Yasufumi Wakabayashi

National Space Development Agency of Japan (NASDA)

2-1-1, Sengen, Tsukuba, Ibaraki, Japan

E-mail: Wakabayashi.Yasufumi@nasda.go.jp

Phone: +81-298-52-2369 Fax : +81-298-52-2415

ABSTRACT

NASDA, office of R&D, has been studying key technologies of planetary rover for years in the fields of a study of remote driving methods by the 6-wheel chassis and mobility performance evaluation by the 3-wheel chassis. Since these studies were related only to the moving capability on the surface, study has begun in the useful rovers that has high work performance in various exploration missions, as candidates for the coming lunar surface initiative mission of Japan. This paper describes some study results of elemental exploration tasks and designing of small lunar rovers. As for elemental studies, specimen-observation technique was tested with computer aiding analysis, and a compact laser range finder was developed and tested for terrain recognition. As for designing of small rovers, two concepts are studied; ground test model of mutually co-operating 2-rover-system has been designed, and capability of re-fuelable flying-rover-system were analyzed preliminary.

1. INTRODUCTION

We developed the 6-wheeled rover (Fig.1) and the 3-wheeled rover, the TRISTAR-II (Fig.2) for the studies of remote driving methods and mobility performance [1, 2]. Since these studies were related only to the moving capability on the surface, designing has been begun in the useful rovers that has high work performance in various exploration missions, as candidates for the coming lunar surface initiative mission of Japan.



FIG. 1 6-wheeled rover



FIG. 2 the TRISTAR-II

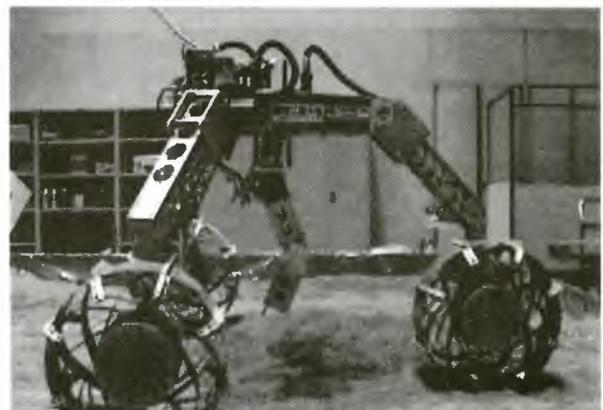


FIG. 3 the TRISTAR-II with a 4-DOF manipulator

And the focus was in the study of how to execute basic exploration tasks effectively. In section 2, observation technique with computer aiding analysis, and a compact laser range finder for terrain recognition are described. In the exploration mission, the allowable total mass to be shared for rovers is considered to be about 100kg, and more effective system as a total exploration device is required. The multi-rover-system is to be more robust and attractive than single system with redundancy in the unknown environment. A mutually co-operating 2-rover-system was proposed, and designed: 40kg rover; 20kg for the bus system, 20kg for the mission equipment. As for night hibernation on the moon, there are three kinds of system concept; (1) system with RTGs (Radio-isotope Thermal Generator), (2) system with big thermal assistant device like a water tank or fuel cells, (3) landing vehicle base system with no self-dedicated thermal assistant device. In the study landing vehicle base system was assumed, in which rovers are activated in noon and return to the landing vehicle at evening for night hibernation. Under these conditions, two types of rover were designed for high work performance. In section 3, the concept of the systems and some results of the design are described.



FIG.4 Manipulator with camera



FIG.5 Manipulator with scoop



FIG.6 Visibility in the forward lighting



FIG.7 Visibility in the inverse lighting

2. ENHANCEMENT OF WORK PERFORMANCE

2.1 COMPUTER AIDED OBSERVATION

Observation of surroundings and rock specimen is one of the basic tasks of planetary rovers, and computer aided analysis can enhance the performance. Tests were conducted using the modified the TRISTAR- II . Fig.3~ Fig.5 show the rover with a 4-DOF manipulator, cameras and a scoop. The followings are added functions to the TRISTAR- II .

- a) 3-CCD camera for observation of surroundings (Horizontal resolution: 750TV lines, SN: 62dB)
- b) High magnifying power camera for rock observation (Horizontal resolution: 560TV lines, SN: 54dB)
- c) 4-DOF (Degree Of Freedom) manipulator
 - 3 Ultrasonic motor -joints, Linear-actuator on the top
- d) Scoop for digging and sampling

1) Wide-viewed display of the surrounding

It is considered very difficult to operate a rover in unknown environments, and the high contrast on the lunar surface will make the task even difficult [3]. Fig.6 and Fig.7 show the visibility in forward and inverse lighting on the pseudo-lunar surface. The results were examined using wide-viewed display which was generated by connecting multi-images by pattern matching. Fig.8 shows the wide-viewed image consisted of 8 images. The field of view is equivalent to 240 degrees, and the pixel resolutions are 2488 horizontally and 480 vertically.

2) High resolution observation of a rock specimen

In the high-resolution observation of uneven surface of rock specimen by a high magnifying power camera, there exist focused regions and unfocused regions within an image. The field of view of the camera is narrow and its depth of focused field in optical axis



FIG.: 8 Wide-viewed image consisted of 8 images

(Horizontal Pixels: 2488, Vertical Pixels: 480)

direction is very shallow. Fig.9 and Fig.10 show the images of the Fassaita pyroxene by high magnifying power camera. The former is the image of concave surface and the latter is the image of convex surface. We examined the method in which the focused regions from each image are extracted, and connected to get a fully focused wide image data.

Fig.11 shows the connection of 2 images, and Fig.12 shows that of 6 images, which resolutions are 1495 horizontally and 480 vertically. By enabling onboard processing, the specimen observation tasks would be executed effectively even in the low communication rate to the earth station [4].

2.2 A COMPACT LASER RANGE FINDER

A laser range finder (LRF) is considered key sensor for navigation and environment recognition. A small-size scanning Laser Range Finders was studied for space applications for years, and a new compact model developed for small rovers [5]. Fig.13 shows the photo of the laser range finder, and Table.1 shows its main characteristics. Fig.14 shows the schematic figure of optical and scanning module in which the optical system consists of a polygon mirror and two parabolic reflectors.

A direct vertical driving of polygon's mount, without belt/gear, is adopted and the upper flat mirrors are eliminated for further compactness, which enabled to minimize the optical system as a designed feature.

Fig.15 shows sand field scenes of 1~5m range. It shows that the LRF can clearly measure the terrain for the lunar/planetary rover application. Because the mounting height of sensor head will be restricted to be less than ~1m for small rover and the sensor spatial resolution is limited,

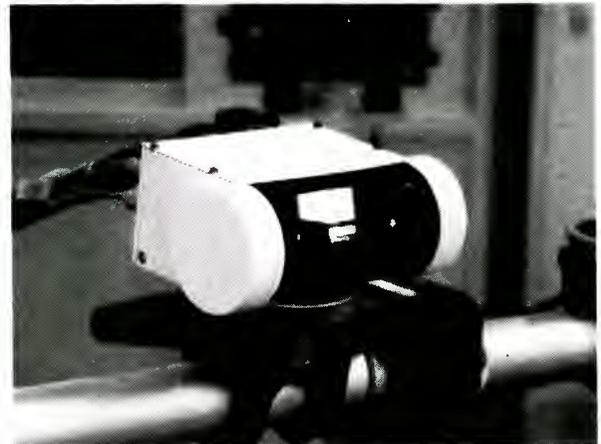


FIG. 13 Photo of the Laser Range Finder



FIG. 9 focused on the left



FIG.10 focused on right



FIG. 11 focused on both

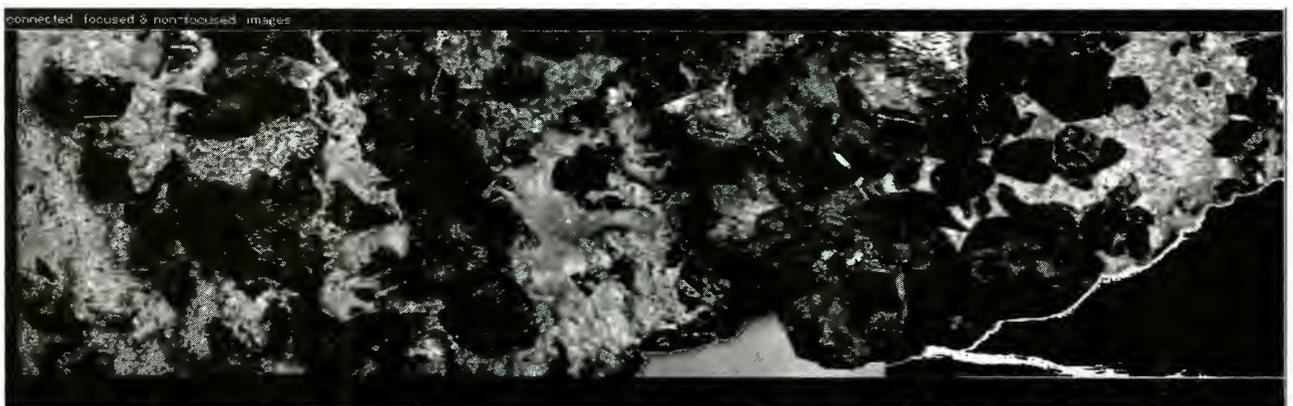


FIG.: 12 focused on 6-images

(Horizontal Pixels: 1495 pixels, Vertical Pixels: 480)

TABLE.: 1 Main Characteristics of the LRF

Items	Specification	Unit
Technical method ranging	Phase detection on AM modulation CW	
H.scanning	Mechanical scanning by a 4-facet polygon	
V.scanning	Mechanical scanning of horizontal scanning module	
total field of view(H×V)	90×45	deg
instantaneous FOV emitted beam	5	mrاد
incident beam	10	mrاد
spatial resolution	(256, 128, 32, 8)×64	pixel
data frame rate	4	Hz
operational range objects	0.5~5	m
CCR	>100	m
resolution range	16	bit
radiance	8	bit
Peak LD power($\lambda = 780\text{mm}$)	150	mW
CW modulation frequency (90%)	10 and 1.5	MHz
effective optical aperture	15	mm ϕ
polygon mirror rev. speed	4800	rpm
driving mechanism of vertical scan	direct drive	
range measurement circuit	full digital	
temperature compensation	direct coupling or optical fiber	
size(W×H×L) scanner	100×81×42	mm
electronics	150×80×120	mm
weight scanner	0.18	kg
electronics	1	kg
power consumption	15	W

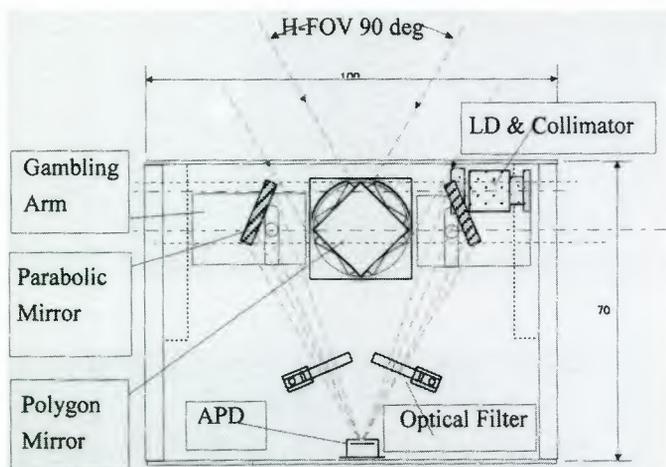


FIG. 14 Schematic figure of Optical and Scanning Module

the measurement range area of 1~5m is adequate for its autonomous navigation and/or map registration. And the high frame rate of this LRF enables the on-board supervision during high speed remote driving. This compact sensor is applied to a small rover described in the next section.



FIG. 15 Range Image Data of a sand field scene

3. DESIGN OF SMALL EXPLORATION ROVERS 3.1 MULTI-ROVER-SYSTEM

The multi-rover-system is to be more robust and attractive than single system with redundancy in the unknown environment. In the study a mutually co-operating 2-rover-system was proposed, and a concept of the small exploration robot is shown below. The chassis concept is 3-wheeled type derived from the TRISTAR-II. The wheels can be folded under the rover's body by the wheel rotation and be expanded to gain high mobility performance.

- a) The improved chassis has low center of gravity of mass and shelter covers for the night hibernation.
- b) The rover has a 4 degree-of-freedom manipulator for exploration tasks such as observation and sampling of specimens. And it also can be used for reconfigurations of rover itself.
- c) The system consists of 2 rovers, each has same configuration with different mission equipment. It is possible to connect the rovers for long distance traverse over the relatively moderate terrain and to operate separately for cooperative works, such as investigation of a small deep crater, recovering from stacked condition, and measurements that need active-roll and passive-roll.



FIG. 16 Overview of Small Exploration Rover

A ground test model was designed with following characteristics from the viewpoint of evaluation on its

working performance and its operability.

- Dust-proof and waterproof feature for experiments on various areas
- Variable communication capacity within the range of 32Kbps-10Mbps for advanced operability
- On-board intelligence of perception for evaluation of its effectiveness

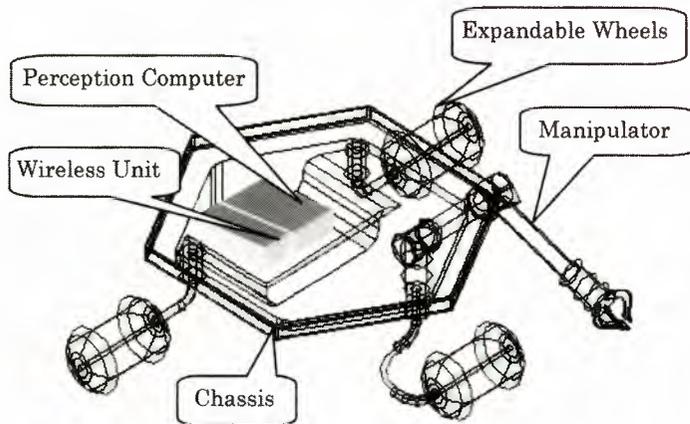


FIG.17 Small Exploration Rover for Cooperative Works

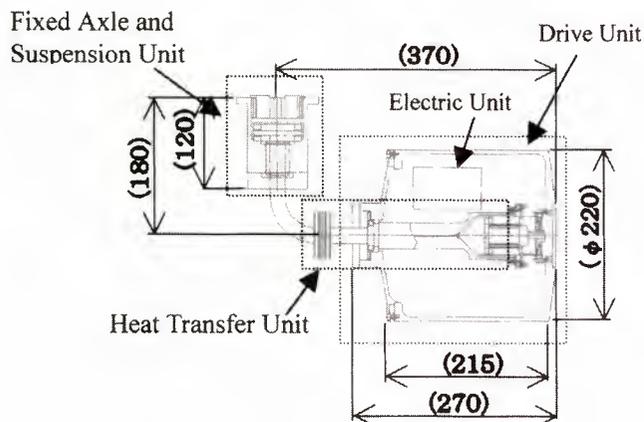


FIG. 18 Wheel System [mm]

Fig.16, Fig.17, Fig.18 and Fig.19 show the ground test model. The model is 1.0m(W) \times 1.0m(L) \times 0.8m(H) size and 40kg weight for the bus system for 1G environment. The distributed control systems are implemented, which use the RS-485 (5Mbps) as the internal communications for command/event data, and the Ethernet (10/100Mbps) as the internal and the external communications of the heavy sensory data. All data are collected at the onboard note-PC to interface the operational station. In a wheel hub system, there are

- Motor control unit
- High power DC-motor and HD-gear
- Heat transfer unit by passive heat pipe
- Perception processing unit

The shaft mounting system with suspension and rotary mechanism is located at body interface. A 4-DOF manipulator by step-motors is located on the top of rover.

As for cooperative tasks, such as pulling the mission equipment (Fig20), pulling-up the other rover (Fig.21), and tasks which need separated transmitter and receiver in particular observations, mutual force and various methods in the viewpoint of the operational technique and driving technique must be considered.

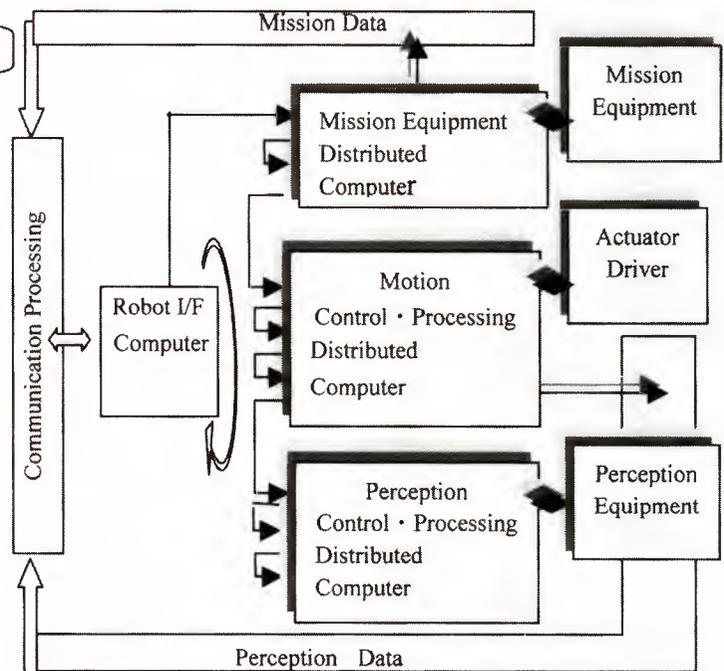


FIG. 19 Distributed Control Systems

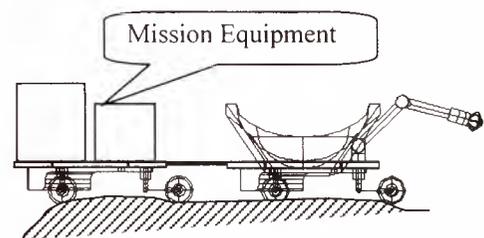


FIG. 20 Works in Connected mode

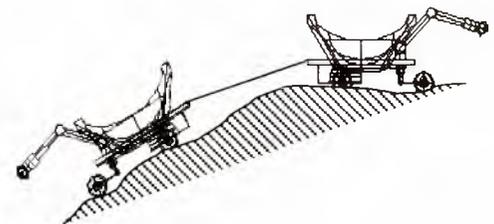


FIG. 21 an Example of cooperative Works

3.2 THE LUNAR FLYING ROVER

A rover with appropriate system could fly around on the lunar surface relatively easily because of its small gravity. There have been many studies related to such systems, but it has shown that self-contained or stand-alone system can not be realized because of weight restriction. Since the mobility performance of rovers is originally limited, a flying probe is remarkably useful in the exploration system that consists of mother base system with small probes. In the study, the flying rover system was designed, which returns to the landing vehicle at night for hibernation and re-fueling, and flies again for several times during noon. The system is designed to fly to ~ 25km in horizontal one way or to ~ 1km in vertical one way during a single day mission, when (amount of propellant) / (total mass of the rover) = 2 / 5; for example, 40kg rover with 16kg propellant. Fig.22 shows the horizontal flight time vs. the ratio of (amount of propellant) / (total mass of the rover). Fig.23 shows the flight control simulation, the results suggested the possibility to operate from the ground with 7sec delay supported by the ground computer.

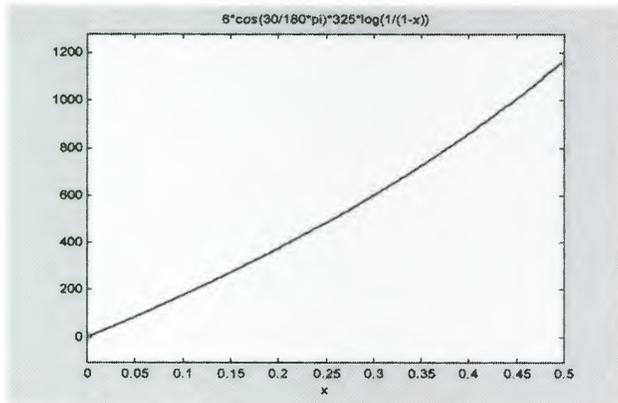


FIG. 22 Horizontal flight time[sec] and the ratio of (Amount of propellant) / (Total mass of the rover)

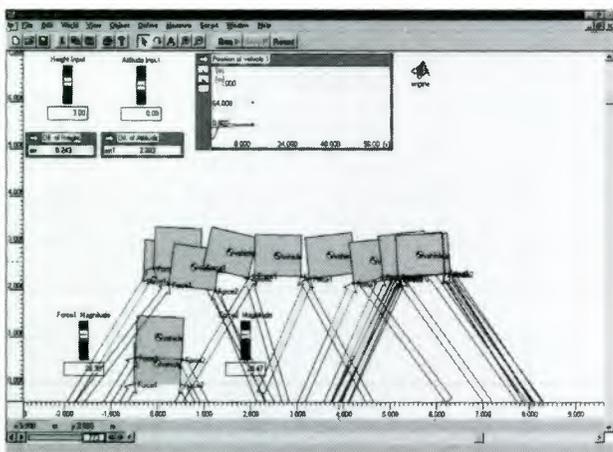


FIG.23 The flight simulation of the flight exploration robot

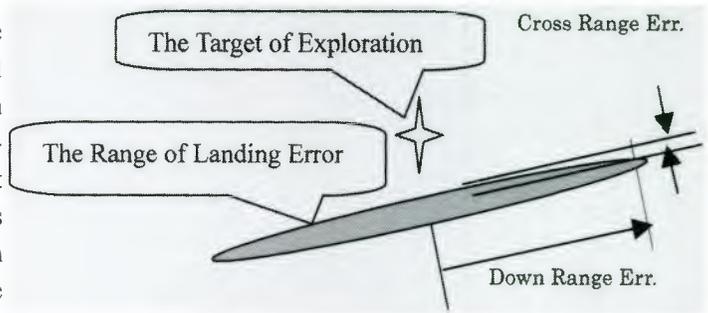


FIG.24 The target of exploration and the landing point error

Such a flying rover is suitable for the mission to the specific target of exploration (Fig.24), and exploration of more dangerous, vital regions such as inside of deep craters, far-side of the moon. The system also has an advantage that it can be operated at the target area in long duration since it requires very short flight time.

4. CONCLUSION AND ACKNOWLEDGEMENTS

For the effective lunar surface mission, NASDA, office of R&D, has been studying basic technologies and system concepts, which is useful for the exploration tasks, and attractive as technical targets. The paper described the research feature of basic computer aided observation, a new laser range finder, the small exploration rovers, which are operated in coordination, and the lunar flying rover system.

REFERENCES

- [1] N. YOSHIOKA et al, "Driving Technology and Preliminary Tests of a Lunar Rover", Proceedings of 13th International Federation of Automatic Control, Vol. P, Session 8a-015, pp.23-28. San Francisco, 1996
- [2] S.HIROSE et al, "The Mobility Design Concepts / Characteristics & Ground Testing of an Offset-Wheel Rover Vehicle", Proc. of the International Conference on Mobile Planetary Robots & Rover Roundup (Planetary Society, Ed.), Santa Monica, 1997
- [3] NASA TR R-401, "Mobility performance of the Lunar Roving Vehicle Terrestrial studies APOLLO 15 results", 1972
- [4] Y.WAKABAYASHI, et al, "Onboard Perception Processing for Space Robots", i-SAIRAS'99
- [5] Y.OHKAMI, et al, "A Compact Laser Range Finder for Space Applications", SPIE99, Conference on AeroSense, 1999

NANOKHOD MICROROVER HEADING TOWARDS MARS

Michel Van Winnendael, Gianfranco Visentin

European Space Agency, ESTEC, Automation and Robotics Section (TOS-MMA)
PO Box 299, NL-2200 AG Noordwijk, The Netherlands
mvwinnen@estec.esa.nl, phone +31-71-5653510
gvisenti@estec.esa.nl, phone +31-71-5654835

Reinhold Bertrand

Von Hoerner & Sulger GmbH, Schlossplatz 8, D-68723 Schwetzingen, Germany
bertrand@vh-s.de, phone +49-6202-21091, fax +49-6202-24303

Rudolf Rieder

Max-Planck-Institut für Chemie, Postfach 3060, D-55020 Mainz, Germany
rieder@mpch-mainz.mpg.de, phone +49-6131-305265, fax +49-6131-371290

ABSTRACT

The European Space Agency has commissioned a number of development activities on micro-rovers for planetary surface missions, aiming primarily at geochemistry and exobiology science applications on Mars. This paper addresses first the background and development logic of these activities and recalls part of the results of the recently completed contract "Micro-Robots for Scientific Applications" which has produced a conceptual design of a flight model as well as an advanced breadboard model of a 'Nanokhod' tracked microrover. Furthermore it describes the complementary near-term development activities.

1. INTRODUCTION

There is growing interest in Europe and the world in exploration of celestial bodies by means of instruments deployed on their surface. An essential element of unmanned missions to the surface of planets such as Mars and Mercury will be robotic mobile devices to deploy instruments in a certain range around a lander spacecraft, to provide visual observation, to sample surface material, and to feed analysis instruments. Such robotic devices will have to function reliably and be versatile in poorly known and often difficult terrain, with very restricted means of intervention from Earth because of extreme remoteness.

To serve the current trend to do smaller, cheaper, more frequent missions, micro-rovers are considered a critical enabling technology. Here, the terms "micro-rovers" and "micro-robots" refer to roving vehicles of the class (significantly) below 10 kg, in keeping with an established terminology of large / mini / micro-rovers for space exploration. Since their primary purpose is to deploy scientific payloads, they have to be optimized to perform this task making as little use of resources as possible.

The European Space Agency (ESA) has funded the development of two micro-rover concepts, specifically designed to perform scientific analysis in-situ, in the frame of its Technology Research Programme activity

"Micro Robots for Scientific Applications", abbreviated "Micro-Rosa", see [1],[2]. This work was performed by *Von Hoerner & Sulger GmbH* (vH&S, Schwetzingen, Germany) as prime contractor, with the *Max-Planck-Institut für Chemie* (MPIC, Mainz, Germany), the *DLR-Institut für Raumsimulation* (DLR, Köln, Germany), the *DLR-Institut für Robotik und Systemdynamik* (DLR, Oberpfaffenhofen, Germany), *Mecanex S.A.* (Nyon, Switzerland) and the *Ecole Polytechnique Fédérale de Lausanne* (EPFL, Switzerland). For the primary concept, named "Nanokhod", which is based on previous work performed at the Max-Planck Institute for Chemistry, a conceptual design of a flight model has been made which offers a very high instrument-mass/rover-mass ratio.

Furthermore ESA has initiated some complementary activities aiming at the delivery of a Nanokhod microrover system which is sufficiently ready to be used for a mission. These activities are related to the development of an end-to-end control system [3], the development of a robotic sampling system [4] and the rover design optimization and qualification of an Nanokhod engineering model.

2. NANOKHOD DESIGN SUMMARY

An overview of the requirements and the design are given in [1],[2]. The main features will briefly be recalled below.

Figure 1 shows the Micro-Rosa system architecture. The Nanokhod "rover segment", with a total mass of 2550 g, including 1100 g of payload, is shown in Figure 2. It is a rugged, simple, reliable yet effective microrover, to carry a set of instruments in the immediate surroundings of a lander (i.e. at least 20 m away from it). In order to maximize locomotion efficiency, it carries around only what is strictly needed for moving and deploying the instruments. Its scientific sensor instruments are accommodated in the central payload cab as shown in Figure 3. Two rotation axes at the ends of the payload cab levers provide 2 degrees of freedom for positioning one of its two viewing

windows w.r.t. a rock or soil spot of interest. Locomotion is performed by means of tracks. As a consequence of the very limited mass and volume budgets no batteries or other power supply devices are on board of the rover. Instead it is equipped with a thin tether consisting of two wires, providing a power and data connection to the lander. Semi-autonomous control is performed using a 3D digital elevation model

of the terrain acquired by means of a panoramic camera on the lander. Thermal control is entirely passive. All micro-rover components shall withstand, under non-operating conditions, temperatures of -140°C to +70°C. Operation will be limited to time slots during which drives and electronics are within their operating range of -80°C to +50°C.

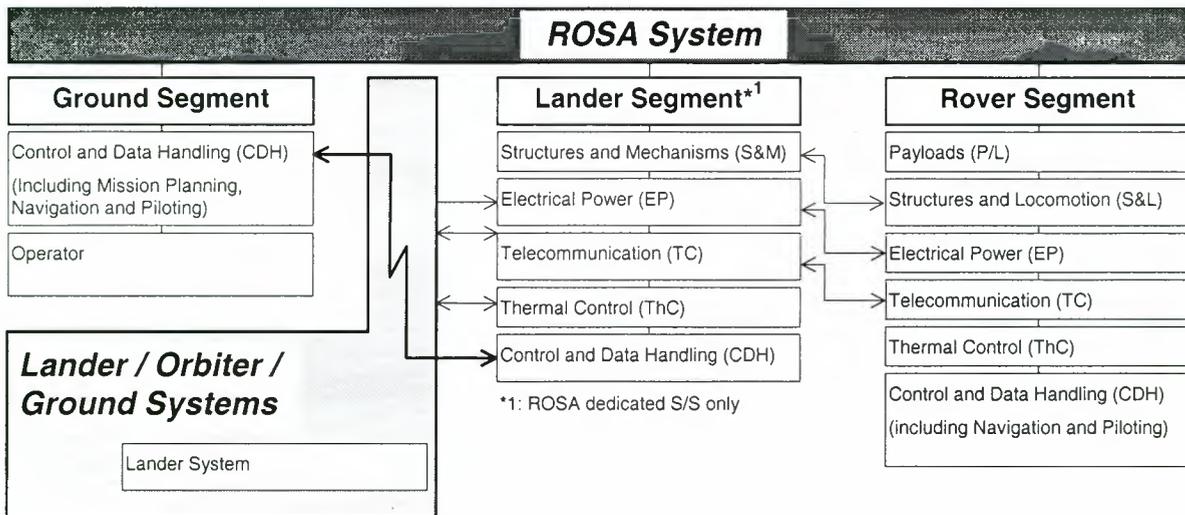


Figure 1: Micro-ROSA System Breakdown

3. BREADBOARD MODEL DEVELOPMENT AND TESTING

As part of the Micro-RoSA activity an advanced breadboard model of the Nanokhod micro-rover has been developed and produced [1]. The main objectives for this breadboarding activity had been to demonstrate locomotion and payload positioning performance in a

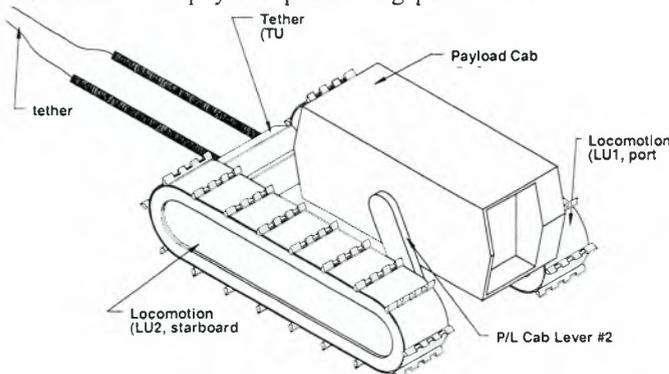


Figure 2: Isometric Sketch of the Nanokhod Rover Segment

Figure 4 shows the hardware items of the laboratory model. The rover is controlled with a standard notebook computer which communicates with the on-rover microprocessor. The rover can be operated by manual piloting using joysticks, or it can execute automated motion sequences in a macro mode including simple sensor feedback.

With 1376.5 g, the breadboard model in its current state is already within the mass specification of the

configuration that is representative in mass and size of the actual rover flight model. Dry lubrication, wear and friction performance, track guidance as well as dust sealing were addressed in detail. The Power and Telecommunication subsystem was designed following the conceptual design baseline for the flight model with respect to the integration of power conversion and telecommunication on the tether wire.

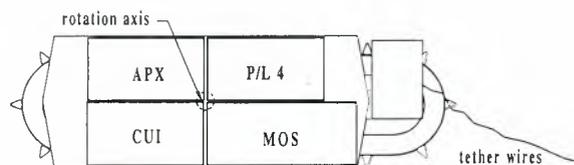


Figure 3: Typical Payload Accommodation (Stowage Configuration)

flight model (1450 g), confirming the feasibility of the concept within the given mass requirements.

A variety of tests have been undertaken not only to verify the performance of the Nanokhod laboratory model with respect to the specified requirements, but also to show that the key design concepts chosen in the conceptual design baseline can actually be realised for a flight mission.



Figure 4: Model A Breadboard Model Hardware

Proof of Concept Testing

Considering the extreme mass restrictions and the resulting passive thermal control approach, environmental compatibility of components and subsystems is a critical point. Therefore, a number of assessment tests have been carried out in order to verify the feasibility of the chosen technical solutions. These tests included:

Deep temperature behaviour of commercial off-the-shelf DC-brush-motors: After removing the lubricant from the motor, it was possible to operate it down to temperatures of -170°C . A careful analysis and adaptation of the motor needs to be carried out prior to flight model development, but the early tests confirmed the feasibility in principle.

Deep temperature behaviour of the track sealing: In order to protect subsystems accommodated in the locomotion units, sealing of the track bodies is a critical issue. With a special brush sealing, this problem has been solved. The mechanical characteristics of the sealing were tested well under -120°C .

Tether Mechanical and Electrical Behaviour at Deep Temperatures: Multiple deployment tests of the tether cable including thermal cycling between ambient temperatures and -195°C have been carried out, confirming mechanical and electrical robustness of the chosen tether under worst case mechanical loads.

Tether Low Pressure Glow Discharge Test: This test was conceived to check the electrical behaviour of a pair of tether wires under simulated Martian atmospheric conditions. In a vacuum tank with a CO_2 -atmosphere between 0,15 and 14,2 hPa, no plasma phenomena could be triggered assessing different tether configurations (parallel, divergent, twisted) and voltage levels up to 500 V.

Operation of electronic subsystems at low temperatures is also an important issue. However, earlier studies [5] have shown that the operating temperatures targeted

are inside the actually attainable limits of space-qualified components.

The tests allowed to confirm the feasibility of technical concepts and component issues, thus supporting the conceptual baseline definition by verification of key design issues. A more detailed and complete verification will be performed in the frame of complementary technology studies and the detailed design and development of the rover flight model.

Breadboard Verification Tests

Within the Micro-RoSA activity, the Nanokhod breadboard model has been submitted to overall system tests in order to verify traction capabilities, tether unrolling on flat terrain, overcoming of obstacles, slope climbing, tip-over recovery, payload cab positioning, as well as use of control and sensors. In particular the locomotion tests have proven the outstanding performance of the tracked micro-rover concept.

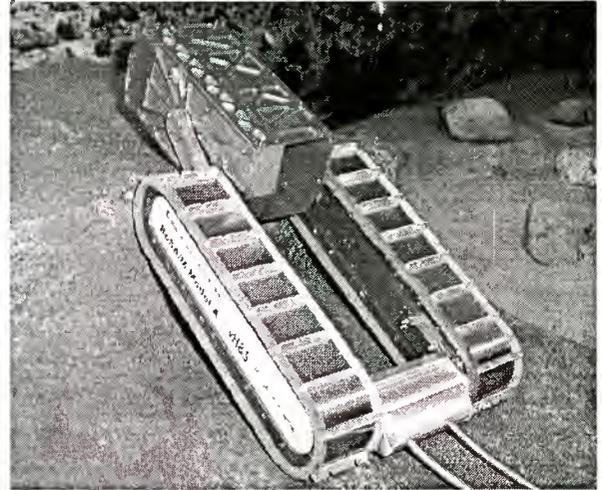


Figure 5: Obstacle Climbing

Although having a total height of only 60 mm in stowed configuration, the Nanokhod was able to overcome 0.1 m step-shaped obstacle.

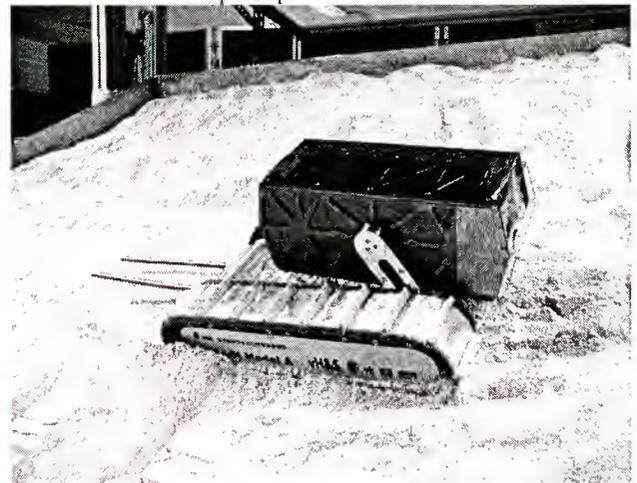


Figure 6: Slope Climbing

Climbing obstacles is facilitated by adjusting the center of mass by movements of the payload cab (Fig. 5). A

similar procedure can be applied for trench shaped obstacles. In Fig. 6, Nanokhod climbs a slope on Mars soil simulant (particles with sub-micron size) at a slope angle of 24 °. On the slope, the vehicle was also able to turn on the spot and to drive sideways.

Another test addressed the tether mechanical behaviour while driving on flat terrain. 50 m of tether cable had been accommodated in the tether unit on the rover. In this test, the tether was deployed while driving forward over a distance of 20 m. The deployment was successful and had no impact of the operational behaviour of the mechanical, electrical and telecommunication subsystems.

The Micro-RoSA test campaign confirmed that all breadboard requirements as specified in the test plan have been met. Nanokhod in particular showed a very robust and reliable performance with respect to overcoming of obstacles and recovery from tip-over.

4. END-TO-END CONTROL

To support the control and navigation the following items are planned to be used:

- an imaging head placed on top of the lander and fitted with optical means to allow modelling of the environment and the localisation of the Nanokhod (part of 'Lander System' in Figure 1)
- a lander-mounted computer that runs the navigation software and controls the rover through the tether (also part of 'Lander system')
- a ground control station (as part of 'Ground Segment').

ESA has recently initiated an industrial activity with Space Applications Services (B) as prime Contractor, together with DLR-Oberpfaffenhofen (D), KU Leuven (B), and von Hoerner & Sulger (D), the overall objective of which is to produce all the remaining elements of the end-to-end control system, to enable quick completion of the Nanokhod in an upcoming mission. In particular, this activity shall study, define, design and produce:

- the ground control station (GCS)
- the on-lander control system (LCS)
- the imaging head of the localisation subsystem (optics, cameras, and pointing)
- the on-rover control system (RCS)

This activity will not produce flight-ready hardware, however it shall use standards, processes, components and materials, which enable a relatively low cost qualification for the Martian environment. The development of all the control system will be based on the MORCA architecture [6], following the 'Interactive Autonomy' concept originally applied for robot arms working in a structured environment.

The end-to-end elements will be designed to support three phases of operation:

- *Pre-preparation*: this phase starts immediately after the lander settles on the planet surface. The

relevant elements of the control system are used to acquire the characteristics of the environment in which the rover will operate.

- *Preparation*: in this phase rover operations are programmed. The rover motion and interaction with the environment is planned, programmed in form of rover programs and verified.
- *Utilisation*: In this phase the rover programs are downloaded to the LCS and executed. This phase includes also the analysis of the telemetry produced during the execution.

The principle on which the control of the Nanokhod is based is that everything is as much as possible rigidly programmed. Therefore the pre-preparation and preparation phases features enable the creation of robot programs the execution results of which can be predicted with high accuracy. In Pre-preparation, great care is taken to create an extremely precise model of the environment. The geometry of the terrain is acquired with the use of the imaging head and high precision computer vision algorithms. Computer vision is also used for interpreting the soil characteristics.

In Preparation, the rover motion is planned using a high performance path planner which produces paths that optimise rover motion and minimise risk of deviation. In Utilisation, the rover motion is continuously measured by the imaging head, together with the localisation software. The control system is therefore able to correct deviations before they become significant.

5. ROBOTIC SAMPLE ACQUISITION

Functions and Requirements

A clear extension of the capabilities of the Nanokhod is in the direction of acquiring samples and not just analyzing them in-situ.

The task of collecting samples for scientific analysis presents several aspects with related requirements. The first aspect is related to accessing the samples. Specifically scientists demand two types of samples:

- Surface samples: these are extracted from surface stones/rocks by coring at the depth of few centimetres
- Deep soil samples: these are extracted vertically from a depth of >1 meters

Beside its use in Earth crust investigations, drilling has been already addressed for space exploration missions and seems to be the most acceptable technical means for accessing both surface and deep samples.

The requirements related to drilling can be resolved into the tasks to:

- penetrate deep (>1 or even 2 meters)
- penetrate non-homogeneous soil of unknown hardness (soft to very hard)
- allow multiple drilling (the research nature of the deep sampling does not guarantee anything interesting is found in the first drill hole)

- operate in unmanned and automatic mode (deployment, drilling and sampling)
- operate in low gravity: the system cannot rely on weight to generate drill thrust

The second step of sample acquisition is sampling. The requirements relative to sampling are:

- acquire a pristine sample of unknown hardness (soft to very hard) and consistency (loose to compact)
- sample at a certain depth, material of that specific layer (not material carried through from upper layers)
- allow investigation of several layers
- preserve morphology of the sample

The third step consists of delivering the collected material. Some analysis instruments, due to their large dimensions, cannot be transported by the rover. They are mounted on the lander and the collected soil/rock material needs to be brought there. The requirements for delivering are:

- transport the sample to the instrument
- do not alter sample morphology
- do not pollute sample with surface material

In view of these requirements, ESA has initiated a first technology research activity (see [4]) to analyse, design and develop a Robotic Sampling System based on the Nanokhod micro-rover (RSS/N). The RSS/N shall be capable of deep drilling, sampling and delivering samples to a lander.

This activity has been recently started by an industrial team composed by Space Systems Finland Ltd. (FIN) with VTT Automation (FIN) and Helsinki University of Technology (FIN). The following description shows only the preliminary concept of such system.

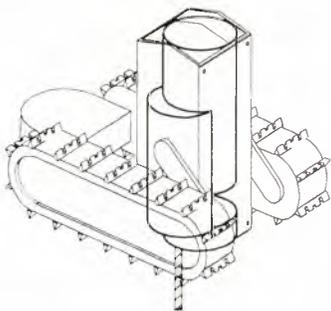


Figure 7: The Drilling and Sampling Subsystem inside the Payload Cab

The RSS/N will be composed of two main items:

- the Drilling and Sampling Subsystem: this element is housed inside the Nanokhod Payload Cab (see Figure 7) and allows the coring of surface rocks as well as sampling in depth.
- the Docking and Sample Delivery Port: this element is accommodated on the lander (see Figure 8) and allows the handing over of samples from the rover to a Sample Processing and Distribution System.

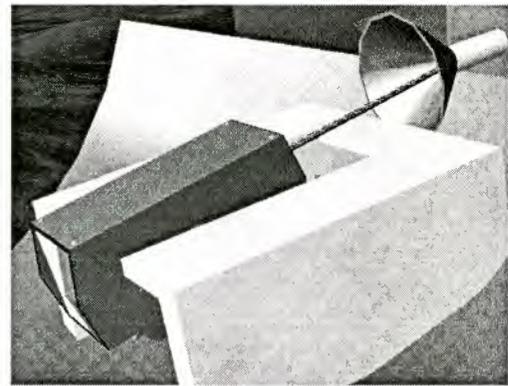


Figure 8: Concept of the Docking and Sample Delivery Port

The Drilling and Sampling Subsystem

This subsystem implements the functions of drilling and coring. For achieving different drilling inclinations the articulation of the payload cab is used. The requirement of drilling at different depths is accommodated by the capability of the subsystem of assembling a *drilling string* of variable length. The *drilling string* is composed by one *tool bit* and up to ten *drill pipes*. A typical assembly sequence is shown in Figure 10.

The *drill pipes* and *tool bits* are connected via socket-plug joints. The coupling and de-coupling is performed by a *drill string assembly unit*. The joints provide a stiff connection thanks to the use of an elastic o-ring, to the matching tetrahedral shape of the socket and plug and to magnetic/thermally actuated clips (see Figure 11).

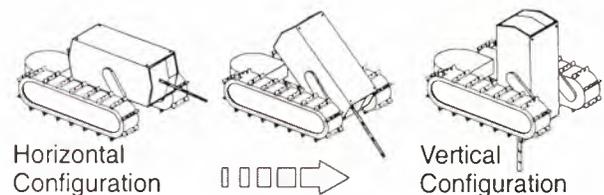


Figure 9: Different Possible Drilling Configurations

The requirement to be able to sample at different depths is accommodated by a simple drill operating procedure and by an innovative tool bit design. Basically the tool bit performs three functions at the same time: drill bit, coring tool and sample storage. The switch between different functions is achieved through different combinations of thrust and rotation. The combined action of drilling and coring (see Figure 13) is achieved when the drill string is pushed downwards and rotates in clockwise direction. Once the desired depth of sampling is reached, the core, which has been developed into the tool bit, can be cut by operating at zero thrust speed and anticlockwise rotation. This combination of thrust/rotation activates the core cutter, which slowly separates the core from its originating material. At this point the drill string

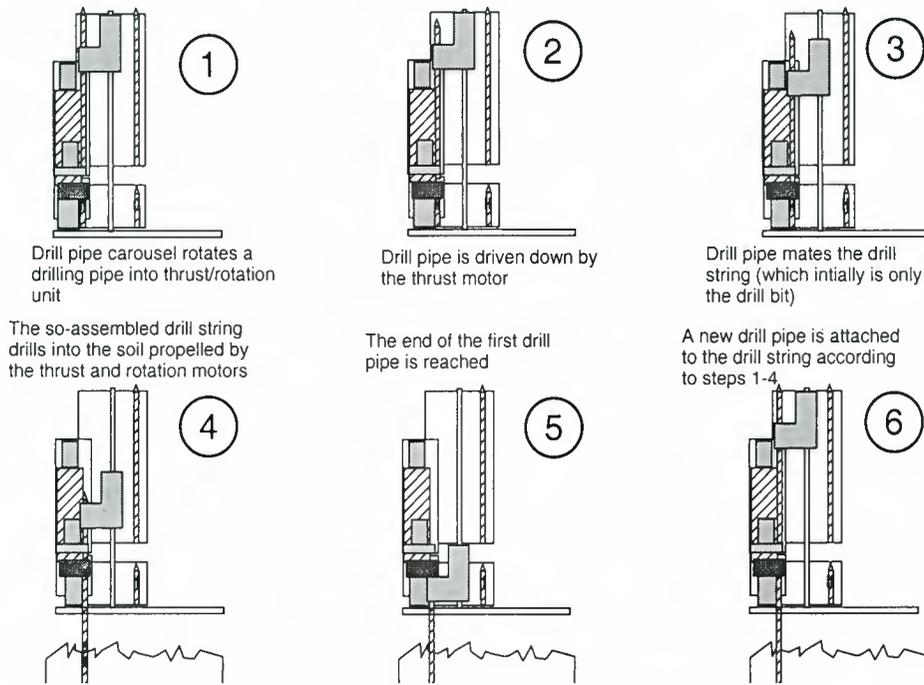


Figure 10: Drill String Assembly Sequence

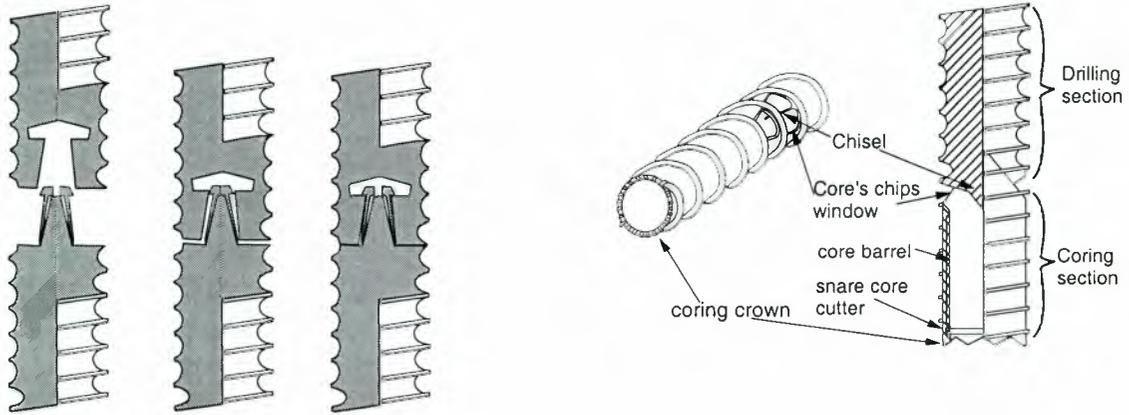


Figure 11: Sequence showing the Connection between two Drill String Elements

Figure 12: The Tool Bit

may be pulled out of the hole by imposing an upward thrust and an anticlockwise rotation.

When the drill string is disassembled, the tool bit holds the core. It is therefore used as sample storage until it is transferred to the Sample Delivery Port. Since the Drilling and Sampling Subsystem holds up to 10 tool-bits, the Nanokhod can collect a total of 10 samples before getting back to the lander.

The Docking and Sample Delivery Port

This subsystem implements the interface between the rover, its Drilling and Sampling Subsystem and the lander. Its is conceived to enable:

- an easy deployment of the rover on the planet surface
- some passive guidance of the rover when it returns to deliver samples

- easy delivery of samples (held into *tool bits*) to a Sample Processing and Distribution System

The first two functions may be realised with rather conventional means (foldable ramps with suitable shapes).

For the last function a first simplistic implementation has been already shown in Figure 8.

The delivery operation uses the Nanokhod payload cab and the Drilling and Sampling Subsystem as a 3 degrees of freedom robot. First, the two tilt-axis of the payload cab point the Drilling and Sampling Subsystem towards the Sample Delivery port. The tool bit is then inserted into the latter by the thrust action of former. The tool bit is then detached from the drill string and its content passed to a Sample Processing and Distribution System.

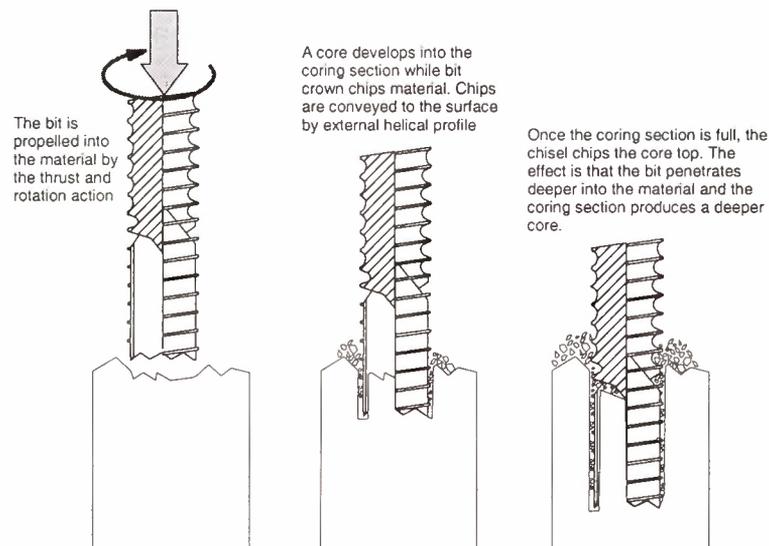


Figure 13: Combined Drilling and Coring

6. TOWARDS AN ENGINEERING MODEL

Complementary to the development steps mentioned in section 4 (End-to-End Control) and section 5 (Robotic Sample Acquisition), the further development of the micro-rover can now address overall system aspects of environmental compatibility including system testing. Within Micro-RoSA, technical concepts for all subsystems have been established and basic technical and operational principles have been verified. While the breadboarding activities have been focussed on locomotion and payload accommodation, the rover as an integrated system now needs to be further advanced in order to resist the environmental conditions on Mars. These activities must include in particular the detailed definition of:

- the low temperature concepts for motors, gears and bearings
- sealing concepts for bearings, tracks, payload cab and tether unit

These can then be implemented in an engineering model which allows for testing of the complete rover in thermal vacuum and dust chambers. The objectives of these tests are to verify the overall system behaviour under realistic environmental conditions, thus establishing the basis for flight model development.

Another issue to be addressed is the system integration of electrical, control and data handling subsystems, as well as payloads. This would include a detailed analysis and design for distribution and optimisation of subsystem and payload elements (P/L sensor parts, analogue and digital electronics, power conversion and distribution, interfaces for power, data, and control), as well as reliability, redundancy, and risk considerations on a subsystem and overall system level.

Applicability on possible other target planets is being considered too, e.g. for a Mercury Sample Return Mission.

7. CONCLUSIONS

Feasibility of a micro-rover for scientific applications with a system mass below 3.3 kg, a peak power need of less than 3 W and a payload to total rover mass fraction between 40 and 50 % has been demonstrated.

Complementary development work has started and has been planned to produce all elements needed for a flight model development for a Mars mission, although there is no confirmed mission opportunity so far.

8. ABBREVIATIONS

APX	Alpha-Proton-X-ray-Spectrometer
C&DH	Control and Data Handling
CUI	Close-Up Imager
DLR	Deutsches Zentrum für Luft- und Raumfahrt
ESA	European Space Agency
I/F	Interface
LU	Locomotion Unit
Micro-ROSA	Micro-Robots for Scientific Applications
MOS	Moessbauer-Spectrometer
MPICH	Max-Planck-Institut für Chemie
P/L	Payload
PLC	Payload Cab
PLCL	Payload Cab Lever
vH&S	Von Hoerner & Sulger GmbH

9. REFERENCES

- [1] R. Bertrand et al: *Micro-Robots for Scientific Applications – Summary Report*. ESTEC Contract No. 12052/96/NL/JG(SC), von Hoerner & Sulger, March 1999.
- [2] R. Bertrand et al: *European Tracked Micro-Rover for Planetary Surface Exploration*. 5th ESA Workshop on Advanced Space Technologies for

Robotics and Automation (ASTRA'98), Dec. 1-3, 1998, ESA WPP-154.

- [3] *Payload Support for Planetary Exploration*, ESA Statement of Work, ITT3445
- [4] *Micro Robots for Scientific Applications 2*, ESA Statement of Work ITT3447
- [5] *Extreme-Temperature Low-Power Electronics*. ESA / TRP No. 11233/94/NL/FM(SC), ESA-ESTEC, 1997.
- [6] A. Martin Alvarez, P.Putz "A Unified Control Architecture for Planetary Rovers", *Artificial Intelligence, Robotics and Automation for Space "i-SAIRAS 94" Jet Propulsion Laboratory, Pasadena, CA USA October 18-20 1994*

Low Power Mobility System for Micro Planetary Rover “Micro5”

Yoji KURODA*, Koji KONDO*, Kazuaki NAKAMURA*,
Yasuharu KUNII**, and Takashi KUBOTA***

*Dept. of Mechanical Engineering, Meiji University

**Dept. of Electrical and Electronic Engineering, Chuo University

***The Institute of Space and Astronautical Science

1-1-1 Higashi-mita, Tama-ku, Kawasaki 214-8571, JAPAN
Phone/Fax: +81-44-934-7183, e-mail: ykuroda@isc.meiji.ac.jp

Abstract

In this paper, we propose a newly developed suspension system called “Pentad Grade Assist Suspension (PEGASUS)”. PEGASUS has high mobility as well as rocker-bogie suspension, with very simple mechanism called only-one-joint architecture. PEGASUS is able to successfully climb up on the step-like terrain with low energy consumption by scale model test. The prototype of micro planetary rover Micro5 which has been developed by the rover R&D group of ISAS, Meiji university, and Chuo university, adopts PEGASUS as its mobility system. The Micro5 is under testing on the variety types of land environment.

1. Introduction

A mission which is an investigation of the Moon or planets is one of the most effective methodology to know how the Solar system was born. As the result of those many missions, e.g. sensing their surfaces with orbiter or lander, had been carried out, we were able to gain wonderful knowledge. Now, it can be said that more detailed investigation should be started to solve more wonder. [1] On the other hand, almost all countries had to cut down the budget for the development of the space because of their economical problem. As a total cost of missions has been reduced, mission concepts have to be changed to “Smaller, Faster, Better” from large scale projects like Apollo missions. In order to carry out the missions with low-cost, using a small unmanned roving vehicle (shown as ‘rover’ hereafter) is one of the fascinating ideas.

On July 4, 1997, NASA succeeded to carry out the mission Mars Pathfinder which the rover called Sojourner [2][3][4] was launched to the surface of

Mars. Since the mission was the first challenge to explore the surface of Mars by using a rover, Sojourner was just tested its technical capability to rove near around a lander station. In spite of that, this success was good news because it was proved that the rover was very useful for the surface investigation. Thus, the rover has been required to explore wider in area during longer term for more scientifically-valued missions. In order to explore wider in area, the rover should have a high degree of mobility to traverse rough terrain.

In this paper, we propose a new mobility system called “Pentad Grade Assist Suspension (PEGASUS)” especially for the small long-range rover.

2. Micro Planetary Rover

The next generation micro planetary rover is required to carry out scientific measurement at many points on wide planetary surface during its limited lifetime. The requirements are not clearly defined, but it could be said that total mass of the rover would approximately be less than 30kg because of capacity limitation of launch vehicles.

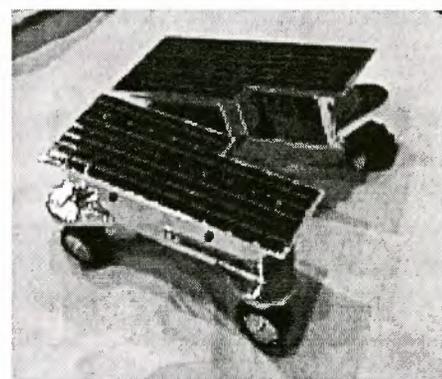
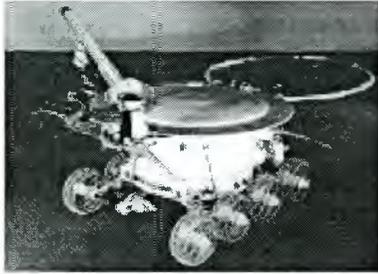


Fig.1 Micro5: prototype micro planetary rover

(a) *Nomad: 4 wheeled rover*(b) *Lunokhod: 8 wheeled Luna rover*Fig.2 *Variety of mobility systems*

In this paper, we discuss the rover system on the premise that the power is supplied from solar cells and batteries. In order to achieve above requirements, the rover *must* have a mobility system which has (1) mechanism to improve its degree of mobility, (2) low energy consumption, and (3) simplicity so as to be embedded in a small body.

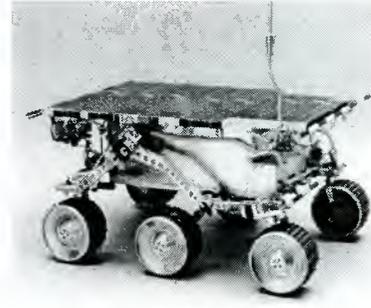
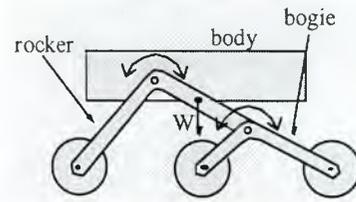
These things listed above are showing a tendency to exaggerate when a rover becomes smaller, because a small rover has to pass over relatively big rocks within relatively low power. In order to examine micro planetary rover in a total system, the prototype rover called *Micro5* [14] has been developed in March 1999 by the R&D group consists of the Institute of Space and Astronomical Science (ISAS), Meiji university, and Chuo university. *Micro5*, is shown in Fig.1, was designed to have small body sized as a microwave oven, and extremely light in weight as 5 kg excluding scientific instruments. PEGASUS was adopted as a new mobility system for *Micro5* rover.

3. Mobility

Because the suspension system is the key issue of degree of mobility, a variety of mobility systems for cruising rough terrain have been proposed. In this chapter, we'll see some typical systems to know what essential points are for improving mobility.

3.1 Conventional 4-wheel drive system

As you know well, a 4-wheel drive (4WD) shown in Fig.2 (a) is the most popular system for an auto-

(a) *Sojourner: the Mars rover with Rocker-bogie suspension*(b) *Mechanisms of a rocker and a bogie*Fig.3 *Rocker-bogie suspension system*

mobile to traverse rough terrain. We call this conventional 4WD here after. There are many rovers adopted this system for its locomotion, e.g., ISAS/Nissan rover [5][6], Nomad [7], AMSL Minirover [8], etc. Conventional 4WD has no mechanisms to improve its mobility, but has just one provides torque to each wheel distributively. It's simple and light in weight, however, it doesn't have high degree of mobility compared with following systems.

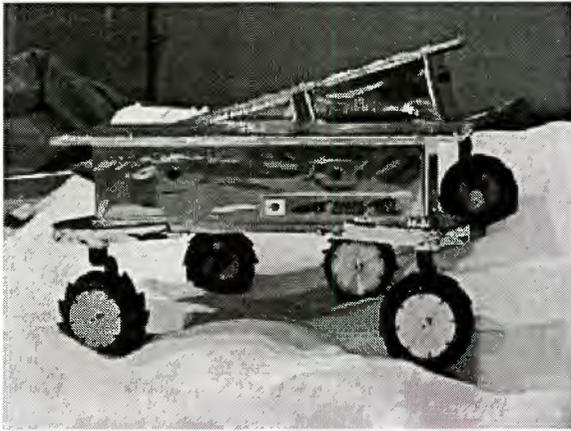
3.2 Multi-wheel (6 or more) drive system

Some Russian rovers [9], e.g. Lunokhod [10], Marsokhod [11], etc., have 6 or more wheels as shown in Fig.2 (b). We classified them as a multi-wheel drive (MWD) system here after. As same as the 4WD system, the MWD system has only the mechanism to distribute the load of weight and torque to each wheel. Though the system has higher degree of mobility because of its load distribution capability, the system is intended to be heavier. Therefore, it is not appropriate to apply for a small rover.

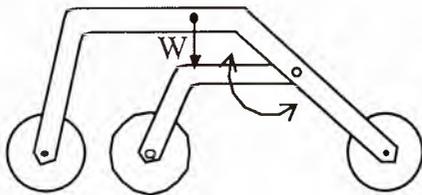
3.3 Rocker-bogie suspension system

Rocker-bogie suspension [2]-[4][12][13] is the system to rove on Martian surface in the mission Mars Pathfinder. NASA has developed this suspension system in a series of the project called 'Rocky'.

As shown in Fig.3, the system consists of a pair of two links called the rocker and the bogie which are attached to each other by a passive rotary joint. The rocker, therefore a set of three wheels, is attached to each side of the chassis by another rotary



(a) The Micro5 rover prototype with PEGASUS system in a mobility test on silica sand



(b) Mechanism of PEGASUS system

Fig.4 "Micro5" the prototype micro planetary rover with PEGASUS mobility system

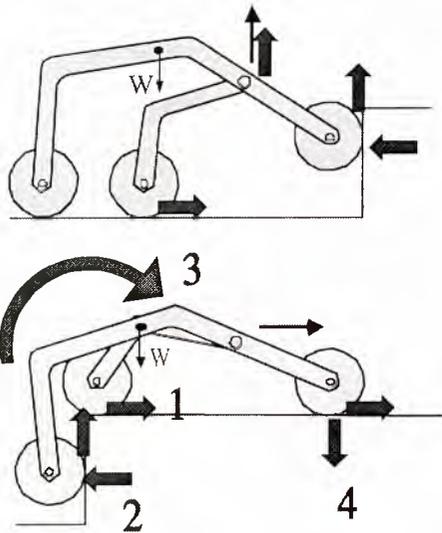


Fig.5 Kinematics of PEGASUS to climb a step

joint. As the wheels are free to move up and down with rotary joints, the system is able to distribute the load of weight to six wheels equality. By this combination of the rocker and the bogie, the rover can climb rocks 1.5 times its wheel diameter in height smoothly.

Though the system provides extremely high degree of mobility for the rover as mention before, this is not a perfect system for smaller rover. One is that

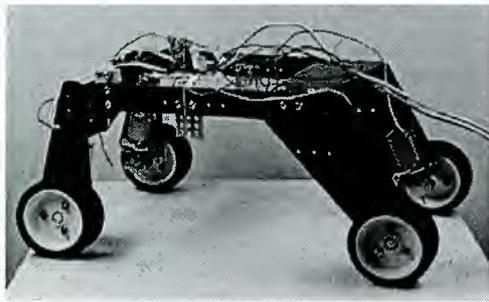
the rocker-bogie has six wheels. Many wheels it has means many motors and gears are contained. So, many-wheeled system, in general, has easily increased the weight. Another problem comes from its complex structure that wheels are attached on the end of the long links and the links are connected by rotary joints as a chain. A pair of lockers and the body are connected via a differential bar in order to stay pitching angle of the body at the middle of two lockers. Therefore, If small force affected on the wheels, very strong stress would be loaded on the links and the joints because of the lever-like structure. Since the links and the joints have to endure high stress, the structure must be made stronger. This leads the system to be heavier.

3.4 PEGASUS system

As above, the small long-range rover is required to have both a simple and light weight mechanism like 4-wheel drive system and a high degree of mobility like rocker-bogie suspension system. In order to achieve these opposed requirements, we propose a new suspension system named "Pentad Grade Assist Suspension (PEGASUS)". A newly developed rover prototype *Micro5* equipped with PEGASUS system is shown in Fig. 4 (a). PEGASUS consists of a conventional four-wheel drive system and a fifth active wheel. As shown in Fig. 4 (b), the fifth wheel is attached to the end of a link, and the other end of the link is attached to the body with a passive rotary joint. PEGASUS needs only one joint rather than the rocker-bogie which needs 4 joints. In general, joints are heavy parts and easily lead to trouble in space environment. So, the architecture called "Only-One-Joint" would be one of advantages.

The system is designed to distribute the load of weight equally to all five wheels when the rover climb up on the step-alike terrain. It means that the fifth wheel supports the load taken to the front wheels when the front wheels climb up rocks, and it also supports that taken to the rear wheels when the rear wheels climb up the rocks. As shown in Fig.5, when the rear wheel climb a step, forward force generated by the traction of the fifth wheel (shown #1 in Fig.5) pushes the rear wheel backward as (#2). These forces produce nose-dive moment (#3), then the moment turns to a vertical force of the front wheel (#4) to support traction. This is the reason why PEGASUS has extremely high mobility. This system realizes such high mobility in simple and light mechanism.

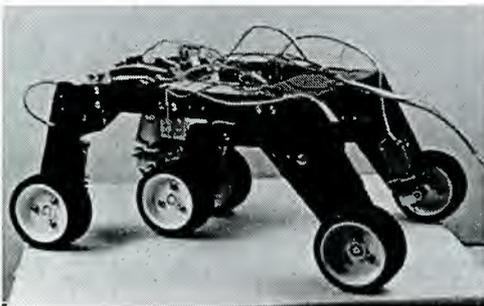
In the following chapters, we discuss how the system is appropriate to the small long-range rover compared with some other typical systems through the



(a) Conventional 4-wheel drive (4WD)



(b) Rocker-bogie suspension



(c) PEGASUS system (consists of the 4WD model with the fifth wheel)

Fig.6 Rover models for energy consumption test

performance evaluation which is carried out by experimental analysis.

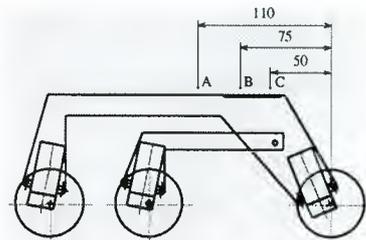
4. Energy Consumption Evaluation

In order to compare energy consumption performance with those of typical mobility systems, we made three types of 1/2 scaled rover models which have conventional four-wheel drive, rocker-bogie suspension, and PEGASUS, respectively. The model are shown in Fig.6. An example view of energy consumption test is shown in Fig.7. Energy consumption of each models are defined that each models consume electricity while they are climbing over the step. Torque generated by each tire can be calculated from current measured in each motor. For the equality in conditions for each suspension systems, size and weight of models are set to be same. The test is carried out in three state of center of gravity as shown in Fig.8 to examine the position effect of center of gravity.

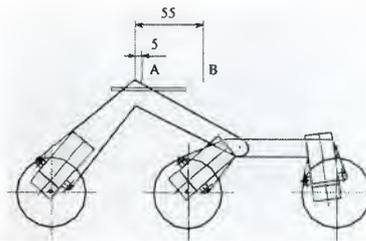
Figure 9, 10 and 11 show the results of model test while climbing over the step. In Fig.10 (a), the



Fig.7 An example view of energy consumption test



(a) Model of 4WD and PEGASUS system



(b) Model of Rocker-bogie suspension

Fig.8 Variety of the position of center of gravity

rear wheel of 4WD model slipped and couldn't climb the step because the center of gravity is located too behind. But PEGASUS could climb over the step smoothly in the case of (a) as shown in Fig.11 (a). Total energy consumption calculated by the summation of current of whole tires while each model is climbing the step is shown in Fig.12. By the result, PEGASUS and rocker-bogie suspension was better than 4WD because their load distribution capability. Furthermore, PEGASUS consumes 20% lower than rocker-bogie suspension. The reason can be considered that PEGASUS has five wheels rather than six wheels.

5. Conclusion

Micro planetary rover needs to be equipped with the mechanism which actively or passively improve its mobility because of overcoming its physical limitation. In this paper, we proposed a newly developed suspension system called PEGASUS. PEGASUS has high mobility as well as rocker-bogie suspension, with very simple mechanism called Only-One-Joint architecture. PEGASUS was able to successfully move within very low energy consumption

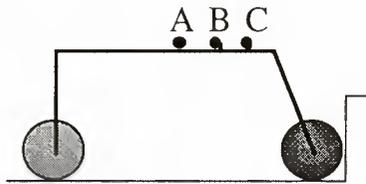
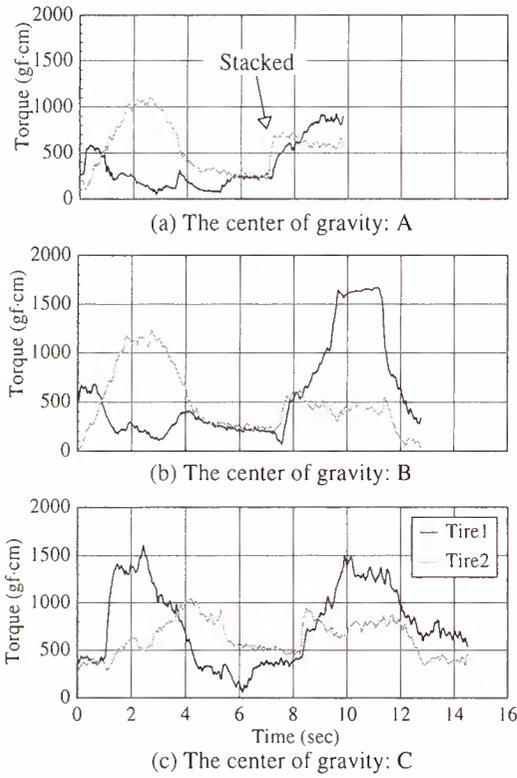


Fig.9 Results of energy consumption experiment
Conventional 4-wheel drive

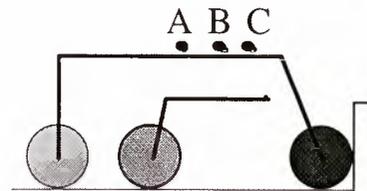
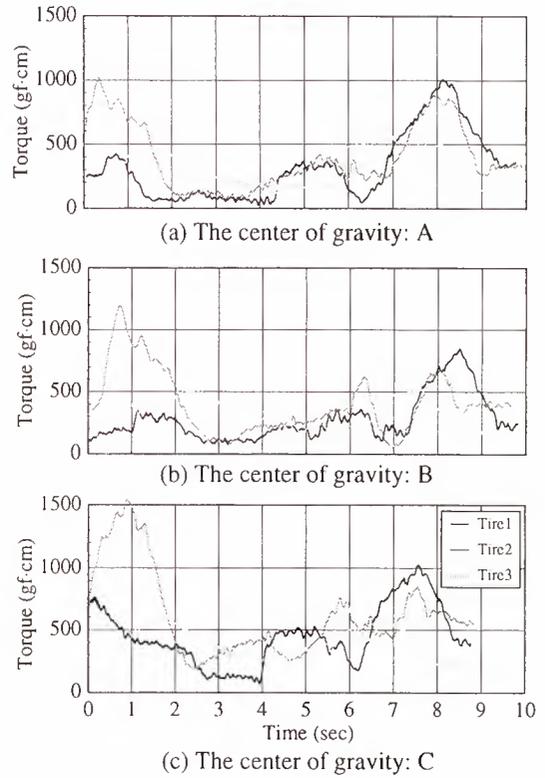


Fig.11 Results of energy consumption experiment
PEGASUS system

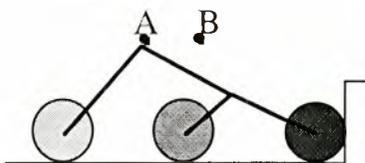
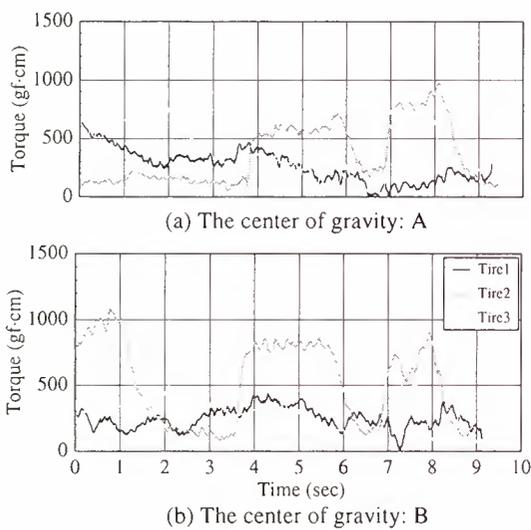


Fig.10 Results of energy consumption experiment
Rocker-bogie suspension

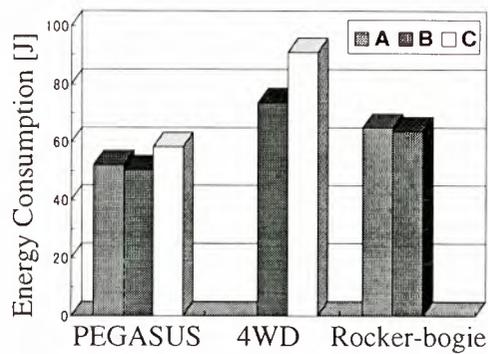


Fig.12 Results of energy consumption experiment
comparison of total performance

by scale model test. The prototype of micro planetary rover Micro5 which has been developed by the rover R&D group of ISAS, Meiji university, and Chuo university, adopted with the PEGASUS as its mobility system. The Micro5 is under testing on the variety types of land environment.

Acknowledgments

The *Micro5* prototype planetary rover was developed by collaboration with *GH-Craft ltd.* and *Advanced Circuit Technologies (ACT) Inc.*

Reference

- [1] C. R. Weisbin, D. Lavery, G. Rodriguez: "Robotics Technology for Planetary Missions into 21st Century", *Proc. of Artificial Intelligence, Robotics and Automation for Space (i-SAIRAS'97)*, July, 1997.
- [2] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin: "Mars Microrover Navigation: Performance Evaluation and Enhancement", *Autonomous Robotics Journal*, vol.2, no.4, 1995.
- [3] Henry W. Stone: "Mars Pathfinder Microrover-a Small, Low-Cost, Low-Power Spacecraft", *Proc. of AIAA Forum on Advanced Developments in Space Robotics*, 1996.
- [4] T. Kubota: "Mars Pathfinder", *Proc. of Journal of the Robotics Society of Japan (JRSJ)*, vol.15, no.7, pp.986-992, 1997. (in Japanese)
- [5] I. Nakatani, T. Kubota, H. Katoh, Y. Kuroda, T. Adachi, H. Saito, T. Iijima, T. Takano: "A Long Distance Moving Test for Planetary Rover", *Proc. of IROS'97, September*, 1997.
- [6] T. Kubota, I. Nakatani: "Autonomous Behavior Control Scheme for Lunar or Planetary Rover", *Proc. of 5th Workshop on Astrodynamics and Flight Mechanics*, pp.334-339, 1995.
- [7] W. Whittaker, D. Bapna, Mark W. Maimone, E. Rollins: "Atacama Desert Trek: A Planetary Analog Field Experiment", *Proc. of i-SAIRAS'97*, July, 1997.
- [8] Y. Kuroda, K. Kondo, K. Miyazawa, T. Kubota, I. Nakatani: "Development of a Combination of Mother/Daughter Planetary Rovers", *Proc. of ROBOMECH'97*, pp.357-358, 1997. (in Japanese)
- [9] A. L. Kemurdjian et al.: "Soviet Developments of Planet Rovers in Period of 1964-1990", *Proc. of JRSJ*, vol.12, no.7, pp.993-1001, 1994. (in Japanese)
- [10] "Lunokhod-1", FTD-MT-24-1022-71, pp.66-77.
- [11] A. Eremenko et al.: "Rover in The Mars 96 MISSION", *Missions, Technologies and Design of Planetary Mobile Vehicles*, pp.277-300, 1992.
- [12] Richard Volpe, J. Balaram, Timothy Ohm, and Robert Ivlev: "The Rocky 7 Mars Rover Prototype", *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, November, 1996.
- [13] Samad Hayati, Richard Volpe, Paul Backes, J. Balaram, Richard Welch, Robert Ivlev, Gregory Sharp, Steve Peters, Tim Ohm, Richard Petras, and Sharon Laubach: "The Rocky 7 Rover: A Mars Sciencecraft Prototype", *Proc. of IEEE ICRA'97, THAI2-2*, 1997
- [14] T. Kubota, Y. Kuroda, Y. Kunii, I. Nakatani: "Micro Planetary Rover 'Micro5'", *Proc. of i-SAIRAS'99*, to be appeared.

HOPPING ROVER "MINERVA" FOR ASTEROID EXPLORATION

Tetsuo Yoshimitsu

Dept. of Electrical Engineering, The University of Tokyo,
3-1-1 Yoshinodai, Sagamihara-shi, Kanagawa, 229-8510 Japan,
phone: +81 42 759 8311, fax: +81 42 759 8304, e-mail: kikko@nsl.isas.ac.jp

Takashi Kubota, Ichiro Nakatani

The Institute of Space and Astronautical Science,
3-1-1 Yoshinodai, Sagamihara-shi, Kanagawa, 229-8510 Japan,
phone: +81 42 759 8305, fax: +81 42 759 8304, e-mail: {kubota, nakatani}@nsl.isas.ac.jp

Tadashi Adachi, Hiroaki Saito,

Aerospace Division, Nissan Motor Co., Ltd.,
21-1 Matoba-Shinmachi, Kawagoe-shi, Saitama 350-1107 Japan,
phone: +81 492 31 1113, fax: +81 492 31 1116, email: {a-tadashi, hiro-saito}@mail.nissan.co.jp

ABSTRACT

The Institute of Space and Astronautical Science (ISAS) of Japan will launch an engineering test spacecraft, MUSES-C to a near Earth asteroid NEREUS (4660) in 2002. The MUSES-C spacecraft will perform the world first sample and return attempt from the asteroid NEREUS. A science-equipped robot which moves on the surface of the asteroid would provide an in-situ scientific observation. So the authors have proposed a small robotic lander for the MUSES-C mission. This robot is called "MINERVA" (MICRO/NANO Experimental Robot Vehicle for Asteroid), which has the mobility by hopping and can take images on the asteroid surface. Currently MINERVA remains in an optional payload of the MUSES-C spacecraft. The final decision will be made just before the launch of the spacecraft whether it is included in the spacecraft. So we are developing MINERVA according with the MUSES-C milestones. In this paper, the mission scenario, the mobility mechanism, the microgravity experiments by the test model and the prototype model of MINERVA are presented.

1 INTRODUCTION

In recent years, missions for exploring small bodies such as asteroids, comets, and meteorites have received significant attention across the world. It is believed that the earth-type planets (Mercury, Venus, Earth, Mars etc.) were formed by small bodies. Hence the studies on these small bodies would throw a light upon the origin and evolution of the earth-type planets. Rendezvous and sample return missions for asteroids, specially, would be expected to provide extensive rewards from both technological and scientific points of views. Especially, in-situ surface observations of an asteroid have been of great interest to planetary science.

The Institute of Space and Astronautical Science (ISAS) of Japan will launch an engineering test spacecraft,

MUSES-C to a near Earth asteroid NEREUS (4660) in 2002[1][2]. The MUSES-C spacecraft will perform the world first sample and return attempt from the asteroid NEREUS.

A science-equipped robot which moves on the surface of the asteroid would provide an in-situ observation [3][4][5]. So the authors have proposed a small robotic lander for MUSES-C mission[6]. This robot is called "MINERVA" (MICRO/NANO Experimental Robot Vehicle for Asteroid), which has mobility system by hopping [7][8]. Currently MINERVA is an optional payload of the MUSES-C spacecraft. Its realization is dependent on the total mass of the MUSES-C spacecraft. The final decision will be made just before the launch of MUSES-C spacecraft whether it is included in the spacecraft. So we are developing the robot according with the MUSES-C milestones.

In section 2 of this paper, the mission scenario of MINERVA is presented. Section 3 describes the mobility system which is newly proposed for explorations on the surface of small planetary bodies. In section 4, the microgravity experiments by the test model is detailed. Section 5 shows the developing prototype model of MINERVA which is to be manufactured and tested in the near future.

2 MINERVA MISSION

2.1 ASTEROID ENVIRONMENT

The asteroid NEREUS or 1989MI, are the candidate target for MUSES-C mission to explore. The orbit of these asteroids are known by the observation from the Earth. But the parameters of the asteroid itself (size, shape, density etc.) are not decided so far. TABLE 1 shows our estimation for the asteroids.

The gravity on the surface of the targeted asteroid is expected to be very weak from 1 to 100[μ G] compared

with the Earth. The robot on the asteroid has to be movable to any arbitrary direction in such a low gravity environment

The traction force that drives the robot horizontally are obtained from the friction between the robot and the asteroid surface. Not-friction-based mobility, such as thrusting a gas backward, can not be used because of the contamination.

Another requirement for the robot comes from the payload weight and size. The robot has to be small and light-weighted. The allowed total mass is 1[kg] including the OME (on-board mount equipment). The simple mechanism is essential in order to meet this weight limitation.

We have developed a new mobile mechanism which is fitted in such low gravity environment and can meet the above requirements.

TABLE 1 : Estimated environment of the asteroid

diameter	a few 100[m]~ a few [km]
shape	ugly (major to minor axis length ratio : 1 ~ 2.5)
density	1 ~ 4[g/cm ³]
rotation axis	unknown
rotation period	4 ~ 30[hours]
temperature	-100 ~ +140[°C]
material	unknown
surface gravity	10 ⁻⁵ ~ 10 ⁻³ [m/s ²] (1~100[μG])
escape velocity	0.02 ~ 2[m/s]

2.2 MISSION SCENARIO

The MUSES-C spacecraft remains at the distance of 20[km] away from the targeted asteroid when the parameters of the asteroid, such as its shape, its size and its rotating rate, are investigated by remote sensing. The parameters are initialized to the robot because they are required for autonomous navigation. Also the global path-planing or strategy will be made on the ground.

When the MUSES-C spacecraft descends to the asteroid for acquiring the surface fragments, MINERVA is released from the OME(on-board mount equipment) which keeps the robot to the spacecraft. Then MINERVA falls onto the asteroid surface and starts the exploration. Small cameras and a sun sensor are used for the autonomous navigation.

The mobile robot provides an in-situ observation of the asteroid surface. The science mission candidates are shown below.

- (1) view the asteroid surface crossly by small cameras. (including the observation of the MUSES-C sampling point)

- (2) measure the surface gravity.

- (3) measure the surface temperature.

Tiny instruments that enable the above missions are now under developing. Because the allowed total mass for MINERVA is limited in 1[kg], not all the listed instruments are possible. For reducing the mass, the installed navigation cameras are also used as the scientific observation.

3 PROPOSED MOBILITY SYSTEM

It is necessary to develop a high-mobility robot configuration which can meet the mission and science requirements. Under the micro gravity environment on the surface of small planetary bodies, robots with traditional wheeled mechanism would not work well because of two reasons stated below.

- (1) The contact force between the robot and the surface is very weak and so the robot-surface friction is very low. If the traction is larger than the maximum friction, the robot slips. So the traction has to be so small, which makes the horizontal speed extremely slow.
- (2) Small disturbance on the surface of rough terrain makes the robot away from the surface, which robs of the traction.

For the above latter reason, we have concluded that the mobility which is specialized in hop has the advantage for the surface of the asteroid. If the robot hops with horizontal force, it can move with no contact with the surface.

There are several ways that make the robot hop, but our proposed hopping mechanism is an innovative one that the robot includes a torquer inside. By rotating the torquer, a reaction force against the surface makes the robot hop at a significant horizontal velocity. After hopping into the air, the robot moves ballistically (FIG. 1).

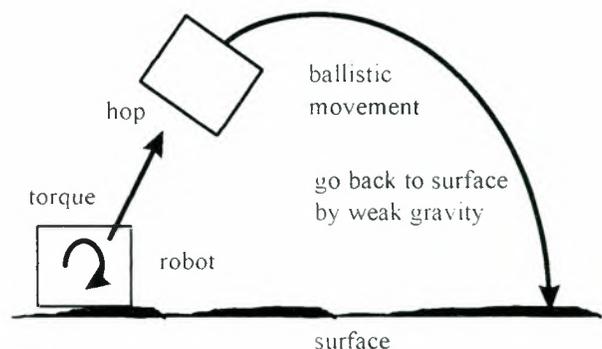


FIG. 1 : Proposed hopping robot

The proposed mechanism has several significant advantages :

- (1) No actuators, which have direct interactions with the asteroid surface, are necessary outside the robot body. Actually the torquers inside the body can even be sealed to prevent contamination by dusts.
- (2) The torquer can also be used for attitude control during hopping.
- (3) The contact force between the surface and the body is increased with the help of the artificial pushing force made by the torquer, which makes the friction larger and provides the mobility at larger horizontal speed.
- (4) DC motors can be used as a torquer, the control of which is easy. The imposed torque has to be adjustable in order to move in any situations of the gravity from 1 to 100 μ G. A DC motor driven by PWM is used to provide the torque adjustability.
- (5) On the microgravity environment of the asteroid surface, the required torque is as small as the small DC motors, which makes the robot light-weighted.

4 EXPERIMENTAL TEST MODEL

The test model for the experiments was developed to confirm the proposed mobility. The configurations of the test robot are shown in TABLE 2(a). A DC motor is used as the torquer whose specification is shown in TABLE 2(b). The motor is driven by PWM of 8.1[V] pulses, the pattern of which can be programmed on the following two modes.

- (a) step drive: the constant duty ratio pulses are imposed on the DC motor. The duty ratio d_m can be programmed to an arbitrary value.
- (b) ramp drive: At $t = 0$, the duty ratio of the pulse is 0%, increased proportionally to the passed time, consequently comes to 100%. Time t_m when the duty ratio arrives at 100% can be programmed to an arbitrary value.

A 30[g] aluminum flywheel is attached to the rotor of the DC motor for increasing the inertia of the rotor.

The micro gravity of 10[sec] is obtained by the free fall of 490[m] with help of jets thrusting upward in order to compensate for the resistance of the air. With this facility, three experiments were conducted under the microgravity environment. The conditions in three experiments varied in (1) the friction between the robot and the surface and (2) the motor driver method. TABLE 2(c) shows the simulation conditions in each experiment.

In experiment #2 and #3, the surface is made up of a flat piece of wood, where the coefficient of static friction is 1.0. In experiment #1, a board is inserted between the robot and the surface. Also a needle sticks out from the

bottom of the robot. With these items, the robot never slides leftward, making the friction ∞ .

The video images during the experiment #1 are shown in FIG. 2. These figures show the validity of the proposed mechanism visibly because the robot hops with a significant horizontal velocity.

All the experiments are summarized in TABLE 3. The hop time in TABLE 3 shows the passed time since the DC motor started. The hop velocity v_x and v_y denote the horizontal and vertical speed at the instant of hop. The hop speed is calculated by $v = \sqrt{v_x^2 + v_y^2}$. The hop angle θ is measured from the normal line to the flat surface which is calculated by $\theta = \tan^{-1} v_x/v_y$.

The numerical simulations under the zero gravity are also conducted, which are summarized in TABLE 4. The simulation results and the experimental results are consistent except for the hop angle and hop time of the first experiment.

The hop angle and hop time of the first experiment have differences from those in the simulation. In the first experiment, compared with the corresponding simulation, the contact duration time between the robot and the surface in the experiment(0.53[sec]) is longer than in the simulation(0.26[sec]). This makes the actual robot's hop angle larger. In experiment #1, to prevent the robot slide leftward making the friction ∞ , a board is inserted. There should be no force from outside after the ascending speed of the robot surpasses the descending speed of the contact point with the surface. But in the case of experiment #1, the needle of the robot contacts the side of the inserted board after the detachment from the surface. This contact force from the side of the board increases the robot's horizontal speed and makes the contact duration time longer and the hop angle larger.

With these experiments, the proposed hopping mechanism has been verified to work well through the experiments. Also the simulations reflect the actual movement. In the future design of MINERVA, we are able to make the most use of the simulations.

TABLE 2 : Parameters in microgravity experiments

(a) robot testbed spec.

shape	$120 \times 96.5 \times 61$ [mm]
total mass	0.55 [kg]
robot inertia	1.0×10^{-3} [kg m ²]
DC motor rotor inertia	2.3×10^{-5} [kg m ²]

(b) DC motor spec. (in 9[V] measuring voltage)

Stall torque	500 [gf cm]
No-load speed	7650 [rpm]
Torque constant	11.2 [mNm/A]

(c) different parameters in each experiment

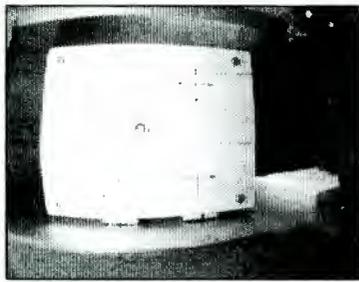
experiment	friction	motor drive method
#1	$\mu = \infty$	step
#2	$\mu = 1$	
#3		ramp(max at 1.71[sec])

TABLE 3 : Hop angle, speed and time of three experiments

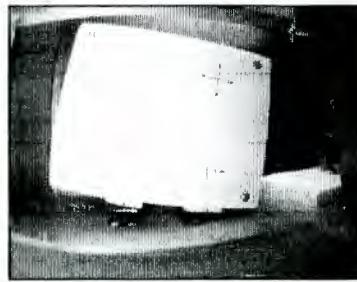
experiment	hop angle [deg]	hop speed [mm/s]			hop time [sec]
		v_x	v_y	v	
#1	73	144	44	151	0.53
#2	43	83	91	123	0.43
#3	46	57	58	82	1.10

TABLE 4 : Simulation result (hop angle, speed and time)

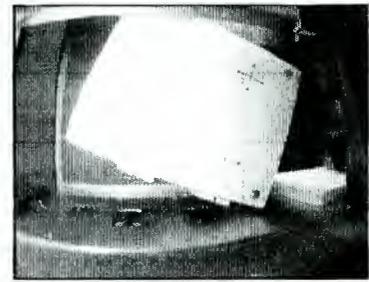
experiment	hop angle [deg]	hop speed [mm/s]			hop time [sec]
		v_x	v_y	v	
#1	55	120	86	148	0.26
#2	40	90	109	141	0.30
#3	39	48	59	76	0.87



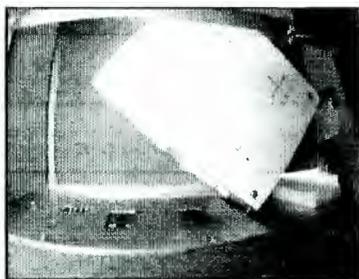
(a) free fall starts



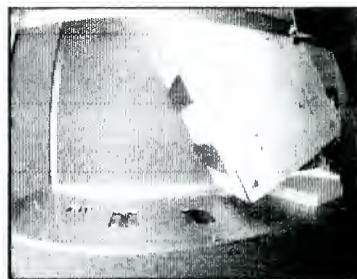
(b) 0.167 sec passed since DC motor started



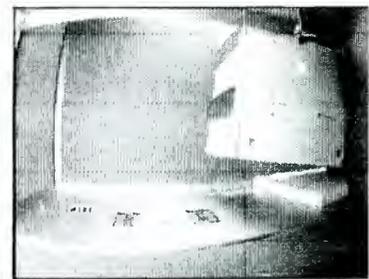
(c) 0.333 sec passed



(d) 0.500 sec passed



(e) 0.667 sec passed



(f) 0.833 sec passed

FIG. 2 : Video images of the experiment #1

5 PROTOTYPE MODEL

We are now developing the prototype model (PM) of MINERVA according with the MUSES-C milestones. The prototype model will be in the process of mechanical and thermal test in the near future.

The concept of PM is shown in FIG. 3. It has a cylindrical shape with two actuators. It includes a big turn table rotator on which is placed a torquer for hop. The adopted mobility is the hopping mechanism proposed in section 3, but has the ability to control the hop direction. It turns the table to set the torquer for an arbitrary direction, which decides the hop direction.

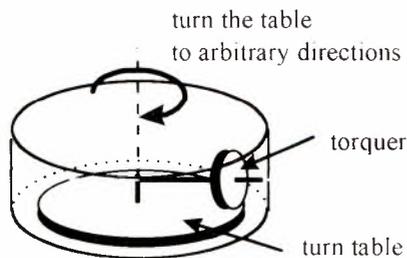


FIG. 3 : MINERVA PM concept

The precise design figure at present is shown in FIG. 4 and the specifications are summarized in TABLE 5. The height and the diameter are 100[mm] and 120[mm] respectively. But the actual shape is an octangle pole, all faces of which are put many solar panels on.

The prototype model has two condensers, where the generated solar energy is charged. The maximum generated power is estimated to be 2.2[W]. The required power of the selected torquer is 2.5[W], which is supplied from the charged condenser. Two flywheels are attached to the torquer for extending the accelerating duration of the torquer.

The possible payloads of the robot are a camera, a sun sensor, a G-sensor and so on. The camera is placed on

the turn table so as to change pointed direction by rotating the table.

There are at least 16 pins sticking out from the vertices of the robotic lander. The purposes of pins are (1) to protect the solar panels from contacting against the asteroid surface and (2) to make intentional hooks to increase the friction.

The mass of the robotic lander is 550[g]. The on-board mount equipment (OME) will be made in 450[g], which keeps the robotic lander while the spacecraft cruises to the asteroid and releases it to the asteroid surface. The total mass of MINERVA and OME is 1[kg] and the requirement for the mass will be satisfied.

TABLE 5 : MINERVA PM specification

Body Size	ϕ : 120[mm], height: 100[mm]
Weight	550[g]
Mobility system	turn table type (two actuators) two flywheels (15[g] each)
Power Supply	solar panels : 2.2[W] (peak) condenser: 5V 50F
On-board computer	32bit CPU
Communication	RF link with MUSES-C S/C (max range : 20[km])
Payload	Camera (CMOS or CCD) Sun sensor G sensor (if possible)
Power consumption	2.5[W] for actuators 1.0[W] for telecommunication 0.5[W] for on-board computer 0.5[W] for camera
OME	450[g]

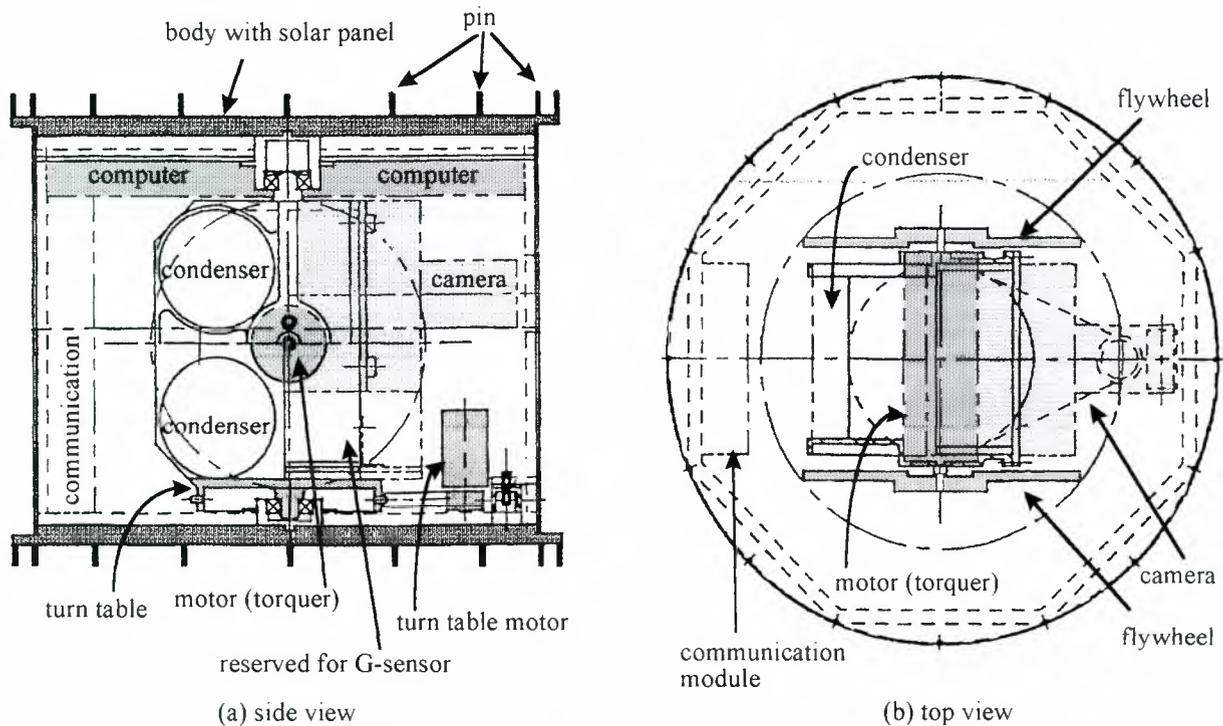


FIG. 4 : MINERVA PM layout

6 CONCLUSIONS AND FUTURE WORKS

This paper describes the proposed MINERVA asteroid surface exploration project for the MUSES-C mission. The scenario, the proposed mobility system, the microgravity experiments by the test model and the development of the prototype model are presented.

A novel mechanism that drives a robot by hopping was proposed for the asteroid exploration. The validity and effectiveness of the proposed mobility system have been verified by microgravity experiments, which is consistent with the numerical simulations. With these validations, the prototype model of MINERVA is being designed using the proposed mobility system. The prototype model will be tested mechanically and thermally. We are also planning to do another microgravity experiments using the developed prototype model.

For robust asteroid explorations, the navigation strategy is also needed to be established. Also the tiny scientific instruments for MINERVA have to be developed.

REFERENCE

- [1] J.Kawaguchi, K.Uesugi, A.Fujiwara, H.Matsuo, "The MUSES-C, World's First Sample and Return Mission from Near Earth Asteroid: NEREUS", 2nd IAA Int. Con. on Low-Cost Planetary Missions, IAA-L-0202, 1996.
- [2] J.Kawaguchi, K.Uesugi, A.Fujiwara and H.Saitoh, "The MUSES-C, Mission Description and its Status", 3rd IAA Int. Conf. on Low-Cost Planetary Missions, IAA-L98-0505, 1998.
- [3] T.Kubota, B.Wilcox, H.Saito, J.Kawaguchi, R.Jones, A.Fujiwara and J.Veverka, "A Collaborative Micro-Rover Exploration Plan on the Asteroid Nereus in MUSES-C Mission", 48th Int. Astronautical Congress, IAF-Q.5.06, 1997.
- [4] B.Wilcox, S.Weinstein and R.Jones, "Nanorover Technology and the MUSES-CN Mission", Proc. of i-SAIRAS '97, pp.445-450, 1997.
- [5] R.Jones, S.Weinstein, B.Wilcox and D.Yeomans, "NASA ISAS Collaboration on the ISAS MUSES C Asteroid Sample Return Mission", 3rd IAA Int. Conf. on Low-Cost Planetary Missions, IAA-L98-0506, 1998.
- [6] T.Yoshimitsu, I.Nakatani, T.Kubota, Y.Kuroda and T.Adachi, "Hopping Rover for MUSES-C Asteroid Exploration Mission", 1998 ISAS 8th Workshop on Astrodynamics and Flight Mechanics, 1998.
- [7] T.Yoshimitsu, I.Nakatani and T.Kubota, "New Mobility System for Small Planetary Body Exploration", Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.1404-1409, 1999.
- [8] T.Yoshimitsu, T.Kubota, I.Nakatani, T.Adachi and H.Saito, "Microgravity Experiment of Hopping Rover", Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.2692-2697, 1999.

Planning and Scheduling of Space Operations

MULTI-AGENT PLANNING AND SCHEDULING ENVIRONMENT FOR ENHANCED SPACECRAFT AUTONOMY

Subrata Das and Paul Gonsalves

Charles River Analytics, Inc.

725 Concord Ave., Cambridge, MA 02138, USA

phone: +1 617 491 3474, fax: +1 617 868 0780, e-mail: {sdas,pgonsalves}@cra.com

Raffi Krikorian

MIT Media Laboratory

20 Ames Street, Cambridge, MA 02139, USA

raffik@mit.edu

Walt Truskowski

NASA Goddard Space Flight Center

Code 588, Greenbelt, MD 20771, USA

Walt.Truskowski@gsfc.nasa.gov

ABSTRACT

Spacecraft autonomy has the potential for effecting significant cost savings in mission operations by reducing the need for dedicated ground staff. In an autonomous operating mode, operators will communicate only high-level goals and deadlines directly to the spacecraft. The spacecraft will then perform its own planning and scheduling, decomposing a goal into a set of sub-goals to be achieved with onboard subsystems and/or in cooperation with other spacecraft in the environment. In this paper, we present this distributed (or equivalently, multi-agent) approach to onboard planning and scheduling that helps a spacecraft function as an autonomous agent. Such an agent's domain knowledge of tasks and their components is manifested through a hierarchical language taking into account spacecraft operational aspects and resource constraints. The task decentralization problem is solved by the use of the hierarchical knowledge structures, and the resource optimization problem is addressed by its explicit representation within the model. The reasoning performed by an agent for the required planning and scheduling tasks is based on a constraint propagation paradigm. Schedule quality is enhanced by the introduction of agent cooperation. A limited-scope Java prototype is developed and demonstrated using space-based scenarios involving onboard sensors and a satellite constellation. We are specifically targeting our effort to enhance the planning and scheduling capability of NASA's proposed Remote Agent architecture.

1 INTRODUCTION

Spacecraft autonomy has the potential for effecting significant cost savings in mission operations by reducing the need for dedicated ground staff. In an autonomous operating mode, operators will communicate only high-level goals and deadlines directly to the spacecraft. The spacecraft will then perform its own planning and scheduling, decomposing a goal into a set of sub-goals to

be achieved in cooperation with other spacecraft in the environment. In this paper, we present this distributed approach to onboard planning and scheduling that helps to function a spacecraft as an autonomous agent.

The term 'planning' refers to the generation of activities that satisfy a current set of goals. For example, a planning process to satisfy the request for an image generates activities such as rolling the camera to the correct position, activating the camera shutter, and transmitting the captured image. The term 'schedule' is an association of these specific activities with particular times by satisfying constraints: for example, rolling should be performed *before* the shutter action. The onboard spacecraft subsystems must execute these time-sensitive activities autonomously to achieve the goals. If none of the subsystems of the spacecraft is capable of executing an activity then a cooperation from another spacecraft in the environment is required to get the activity executed to achieve the overall goal. For example, if a spacecraft is incapable of taking an infrared imagery of a certain swath of the planet then it has to seek cooperation from another spacecraft in the environment that can do so. In addition to serving these payload-oriented functions, planning and scheduling are also necessary to achieve goals generated to ensure safe spacecraft on-orbit operations. As described in (Pell, 1997), the onboard planner assumes a domain model containing an explicit representation of spacecraft subsystems, tasks, goals, and the norms, under which they operate. These norms are a set of flight rules and constraints that are represented in a high-level syntax.

Two major trends for task representation in the history of AI planning have been observed (Georgeff, 1987): goal achievement (GA) and hierarchical task network (HTN). The origin of GA-based planning is in STRIPS (Fikes, 1971). In this model of representation, an initial situation, a set of possible actions, and a goal that is to be achieved are given. Planning consists of finding a sequence of actions that would lead from the initial

situation to the final one. Several planners were subsequently built on the GA model including TWEAK (Chapman, 1987), and SNLP (McAllester, 1994). On the other hand, the HTN representation has its origin in NOAH (Sacerdoti, 1974). A planner based on the HTN model is presented with a *task* or *activity network*, which might contain several *non-primitive* tasks. Planning proceeds by selecting a non-primitive task, decomposing it into *subtasks* using a library of available *decomposition methods* and then detecting and resolving conflicts with other tasks. This process is repeated until no non-primitive tasks remain and all the conflicts have been resolved. Typical examples of HTN planners are FORBIN (Dean, 1988), and NONLIN (Tate, 1977). There are also planners combining features from these two such as O-Plan (Currie, 1991) and SIPE (Wilkins, 1988).

Given a representation in either GA or HTN, solving a planning problem can be viewed as a straightforward search problem, that is, find some or all possible orderings of the actions that would result in achieving the specified goal, given the rules and constraints of the environment. In general, the HTN paradigm can lead to more efficient planners because it allows the user to limit the search space by guiding the planner towards exploring only acceptable solutions. A typical implementation of the search engine of a planner operates on a temporal database such as the HSTS system (Muscettola, 1994) and Time Map Manager (Boddy, 1994). The search engine posts constraints to the database. The temporal database then constructs a constraint network and provides a *constraint propagation* (Le Pape, 1990) service to verify the global consistency of the posted constraints with the goals, rules and constraints of the spacecraft. This global consistency guarantees the existence of a schedule satisfying the constraints. Both the consistency checking and search for an optimal solution in cooperation with other agents in the environment are computationally intractable, that is, NP-hard. A distributed approach to planning and scheduling allows cooperation among agents in the environment and increases efficiency in the search for an optimal solution by partitioning the whole search space.

In recent years, there has been a growing interest in agent-oriented problem solving (CACM, 1994), which provides the basis of our proposed distributed solution (Chaib-draa, 1992) to planning and scheduling. The agent-oriented problem-solving environment increases efficiency and capability (Rosenschein, 1982) by employing a set of agents, communicating and cooperating with each other to achieve their goals, that is, to find a local solution that satisfies both its hard and soft constraints. By an *agent* we mean, an entity which operates in an environment either autonomously or semi-autonomously interacting with other agents in the environment by means of communication. Agents are sometimes software agent (Genesereth, 1994) implementing the behavior of humans, machines or hardware, etc. Agents can also be mechanical or

electronic robots (Simmons, 1991) with the capability of perceiving or sensing the environment and capable of executing appropriate actions. Our assumption is that even if an environment consists of such heterogeneous agents there will be a well-defined means of communication between these agents. In other words, every agent has an interface, which understands a common communication language.

In our envisioned distributed (or equivalently, multi-agent) environment (Conry, 1988; Georgeff 1983), a set of problem solving autonomous agents (an agent is either an onboard subsystem of a spacecraft or the spacecraft itself) communicate and co-operate to achieve high-level goals through planning and scheduling. This distributed planning and scheduling emphasizes a decentralized organization, plans are generated and executed cooperatively and concurrently by the subsystem agents and spacecraft agents, taking into account system flight rules and resource constraints. In a *centralized* planning environment, goals, rules, constraints, and resources from individual agents are accumulated at a central place and a centralized planner is used to generate a global schedule. An individual agent is then provided its relevant portion of the schedule of tasks. The agent then informs the centralized planner the progress on the schedule. This centralized approach is particularly unsuitable when the problem is inherently distributed such as in a spacecraft environment where each subsystem and spacecraft functions autonomously. A centralized planner is unable to exploit fully the expertise and knowledge of each individual agent, and makes the search space unnecessarily larger. In a *distributed* environment, in contrast, each agent (i.e., an onboard subsystem or a spacecraft) generates and maintains its own plan and schedule, and therefore the whole search space is divided into a number of smaller ones to be managed by individual agents. The overall plan and schedule is obtained by combining or synchronizing plans from individual agents, resolving any conflicts that arise from the constraints on the resources.

In our envisioned distributed environment, an agent's model of the environment and tasks is manifested through a hierarchical knowledge representation language taking into account spacecraft operational aspects and resource constraints. The task decentralization problem is solved by the use of the hierarchical knowledge structures, and the resource optimization problem is addressed by its explicit representation within the model. The reasoning performed by an agent for the required planning and scheduling tasks is based on a constraint propagation paradigm. Schedule quality is enhanced by the introduction of agent cooperation. A limited-scope Java prototype is developed and demonstrated using scenarios involving onboard sensors and satellite constellations.

The rest of the paper is organized as follows. First we describe two space-based scenarios to illustrate the envisioned operating mode of a spacecraft agent, that is, to achieve high level goals through distributed planning

and scheduling. Then we present a generic architecture in section 3 that can be instantiated appropriately to implement an agent in the environment. The hierarchical syntax for modeling an agent's domain knowledge of tasks is presented in Section 4. Section 5 describes the protocol for inter-agent communication. Section 6 contains our approach to decentralization and coordination of tasks among agents. The functionality of the current Java prototype is described in section 7. Finally, we summarize our work in section 8 and lay out our future plan for extending the work.

2 EXAMPLE SCENARIOS

We present two space-based scenarios in this section to illustrate our envisioned distributed planning and scheduling. Onboard subsystems of a spacecraft are considered as agents in the first scenario, whereas individual spacecraft themselves are agents in the second scenario.

The first scenario is a modified version of the 'Spaceworld' model scenario presented in (Vere, 1983), where the goals are to send pictures of objects in deep space from the spacecraft to Earth. In the current New Millennium Remote Agent (NMRA) architecture (Pell, 1997; Chien, 1997), the executive will pass these goals to the planning and scheduling component, which recursively selects and schedules appropriate activities to achieve the goals. The component also synchronizes activities and allocates global resources over time such as power and data storage capacity. Thus, the planning and scheduling component of NMRA maintains a dynamic model for each of the subsystems to carry out its task.

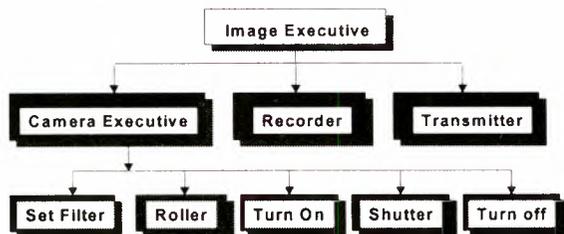


FIG 1: A Multi-Agent View of Intra-Satellite Image Request Processing Activities

In our envisioned distributed environment, the executive will delegate the task to an image executive (illustrated in figure 1), dedicated to managing goals related to obtaining, recording, and transmitting a picture to earth. This executive is only a high-level planning and scheduling agent, and it does not deal with resource allocation. The executive's plan can take one of the following two directions: 1) if the earth is in view (can be verified with the help of the camera executive) then it will send a request to the camera executive to take the picture followed by a request to the transmitter to transmit the picture; 2) if the earth is not in view then it will send a request to the camera executive followed by a request to the recorder to record for a subsequent transmission. The scheduling part of the image executive consists of

specifying time intervals along with the requests to the camera executive, recorder and transmitter agent.

Upon receiving a request from the image executive, the camera executive schedules activities such as filter setting and turning on and off the camera, by taking into account its prior commitment of its own resources to other agents. Additionally, it will contact the roller agent to roll the spacecraft to position the camera for the desired picture and requests to maintain that position for a certain amount of time. A roller agent may just only be a resource manager. It will meet the request from other agents on a first-come-first-serve basis, and thus no serious scheduling activity is involved. If the camera agent fails to meet the request of the image executive then it will inform the executive with possible alternative slots for cooperation. The executive will coordinate with the camera agent to come up with an agreeable time slot. In a similar manner, the recorder and the transmitter agents will perform their own local scheduling and resource allocation in cooperation with the image executive.

To illustrate further our distributed environment where agents are individual satellites themselves, consider the scenario consisting of a constellation of satellites with different viewing capabilities (infra-red (IR), visible, or ultra-violet (UV)) orbiting a planet – and the goal is a full spectrum sweep of a certain swath of the planet. Traditionally, a human operator in the ground station would need to lookup to see which satellites with a given capability will be making a pass over the section of the planet indicated. Following this, the operator will need to address each satellite to request and organize the sweep with all relevant details down to the transmission of the data back to earth. Ideally, the operator should only need to transmit a high-level goal similar to "Take a full spectrum imagery of the area bounded by <latitude and longitude data> and transmit the picture back in two days". An executive level satellite can receive this command, decompose it, and then negotiate with the agent community (where each satellite in orbit is part of the community) to attempt to schedule a plan (as illustrated in figure 2). From there, each satellite can respond with information such as "will be passing over the site in 36 hours, I can generate in IR" or "will not be passing over the site for another 96 hours, I can not generate the image." Certain constraints may come into play also – UV and visible light sensors are only useful when that side of the planet is facing the sun. Responses to this situation may be similar to "will be passing over the site in 4 hours, but the site is currently on the dark side of the planet" or "will be passing over in 4 hours when site is on dark side, but will pass over again in 20 hours when it is local noon." Of course, there are certain requests which just cannot be fulfilled, it is the executive's job to notice these, come up with the "closest fit" to the request issued from the human operator and report back with the closest fit to ask for a go-ahead on that schedule.

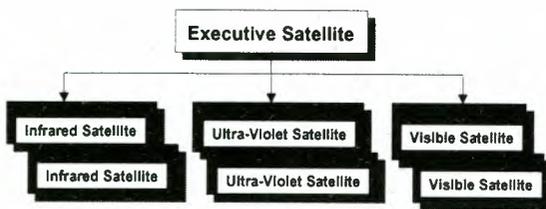


FIG 2: A Multi-Agent View of Inter-Satellite Image Request Processing Activities

When all of the planning has been performed through the negotiation, the executive satellite can issue the plan to all image gathering satellites (perhaps via the Tracking Data Retrieval Satellite System (TDRSS)). The satellites will receive their plan, and internally they will schedule their own control (perhaps via an internal agent network for subsystem control as described in the Spaceworld scenario) for setting up their imaging systems, recording the image, and then transmitting it. The satellite will pass over the section of the planet when the time is right, record the images, and transmit their image back to TDRSS. TDRSS will assemble the images when the whole spectrum has been covered, and transmit that back to the human controller.

3 AGENT ARCHITECTURE

Our architecture of an agent is essentially *deliberative*, i.e., there is an explicit symbolic representation of the model of the dynamic environment; agents make decisions via logical reasoning based on pattern matching and symbolic manipulation. Several different deliberative agent architectures have been proposed in the literature and two of them are most prominent: *horizontally layered architecture* (Ferguson, 1992) and *vertically layered architecture* (Muller, 1994). A layered approach models an agent as consisting of several hierarchical functional modules representing different requirements of an agent. Possible layers incorporate communication, reaction, inference for planning or scheduling, perception, knowledge maintenance, etc. Each layer in a horizontally layered architecture has access to both the perception and the action components whereas, in a vertical approach, only one layer has direct interface to the perception and action.

The architecture we have adopted is displayed in figure 3 and it fits into the vertically layered category. The three layers are world interface layer, inference layer, and network management layer. An agent's knowledge base is also split into three types corresponding to the three layers.

The *world interface layer* contains the agent's facilities for perception, action and communication. These activities require a detailed knowledge about the environment. An agent's world model contains information about the environment, for example, information about other agents such as their locations and capabilities. The world interface layer enables an agent to communicate with other agents in the environment to

perform activities related to planning and scheduling such as sending and receiving requests, responding to a request, etc.

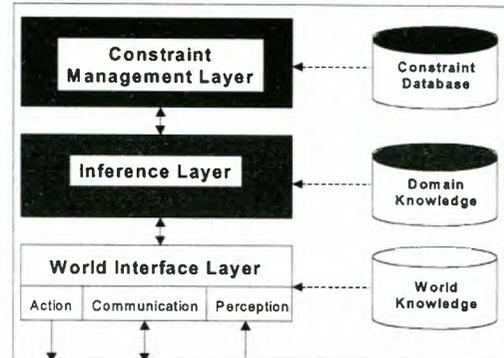


FIG 3: Vertically Layered Agent Architecture

Upon receiving a request from another agent through the world interface layer, the *inference layer* does planning or scheduling or resource allocation, depending on the type of the agent, using the available domain knowledge. The domain knowledge consists of the knowledge of the application, for example, description of different task abstractions and resources, effects of a task when it is carried out, and so on. Most part of the domain knowledge is static in nature in the sense that it remains the same for a particular application.

The job of the *network management layer* (also called the *temporal database layer*) is to manage the temporal constraint network generated during the planning and scheduling process by the inference layer. The constraint database is a persistent store for the constraint network. The layer provides the consistency checking service for the inference layer upon receiving a propagation of constraint from the inference layer.

To illustrate the interactions among the layers, we provide a small example of resource allocation activity of the recording agent described in the previous section. Suppose the maximum recording capacity at any time is 1GB (this is a resource constraint) and 700 MB of it has been scheduled for the time interval [800, 900]. This information along with the constraint is appropriately stored in the constraint database as a temporal network. The current state of the database is consistent. Now, if a request for 400MB for the interval [850, 950] arrives from the image executive, then the world interface layer will pass this request to the inference layer. The inference layer posts this request as a constraint to the network management layer. The network management layer tries to construct a consistent schedule combining the existing network with the incoming request. Upon failing, the layer informs the inference layer, which in turn informs the requesting agent through the world interface layer.

4 HIERARCHICAL MODELING

As mentioned in the introduction, a planning process based on a HTN representation first constructs a plan containing abstract high-level activities and then refines

these components in more detail. This process of refinement continues until these high-level activities themselves correspond to the physical actions in the real world. The advantage of this approach is that the feasibility of a plan can be studied incrementally. If an autonomous software agent is implementing the above refinement process then domain knowledge of the tasks and their components have to be codified in some language. We provide here a flavor of how our proposed HTN representations (similar to (Das, Fox et al., 1997)) look like. The syntactical details and expressiveness of this language are not important at this stage, as our objective is mainly to explain the concepts.

A *compound task* specification has three components: 1) a set of sub-components which specify the *subtasks* and *atomic actions* from which this compound task is built; 2) a set of *constraints* including constraints on ordering between subtasks; and 3) a set of effects when the task is carried out successfully. The two compound task specification for the image executive (figure 1) in the 'Spaceworld' example presented in section 2 is provided below:

compound-task decomposition:	send-picture-to-earth(Object, Filter)@[S, E] picture-object(Object, Filter)@[t1, t2]; transmit-picture(Object, Filter)@[t3, t4]
constraints:	S = t1; E = t4; t2 =< t3
effect:	received-on-earth(picture, Object, Color, S, E).
compound-task decomposition:	send-picture-to-earth(Object, Filter)@[S, E] picture-object(Object, Filter)@[t1, t2]; record-picture(Object, Filter)@[t3, t4]; transmit-picture(Object, Filter)@[t5, t6]
constraints:	S = t1; E = t6; t2 =< t3; t4 =< t5; t5 - t4 =< 100
effect:	received-on-earth(picture, Object, Color, S, E).

The first specification for the compound task send-picture-to-earth contains two subtasks: picture-object and transmit-picture. This task will be normally followed by the image executive, if the earth is in view; otherwise, the second alternative is pursued. This kind of options provides non-determinism of the unfolding process in hierarchical planning. The efficiency of the planner and the schedule quality depends on which option is chosen. If either one of these two tasks is carried out successfully then the earth will receive the picture as its effect.

The expression $t2 \leq t3$ constrains the fact that a transmission cannot be started before taking the picture task is finished. Various types of constraints will be considered and propagated from agents to agents in a distributed planning process. *Hard constraints* represent those objective requirements and procedures that must be met to ensure a correct solution by an agent. The constraint just stated is an example of a hard constraint. On the other hand, *soft constraints* represent criteria that can be relaxed and are not essential for achieving a correct solution. For example, $t5 - t4 \leq 100$ constrains that the time between the recording and transmitting should be less than 100. This can always be considered as a preference. The compound task specification for the camera executive is specified in a similar manner:

compound-task decomposition:	picture-object(Object, Filter)@[S, E] set-filter(Filter)@[t1, t2]; roll-camera(Object)@[t3, t4]; turn-on-camera@[t5, t6]; shutter-camera(Object)@[t7, t8]; turn-off-camera@[t9, t10]
constraints:	S = t1; E = t10; t2 =< t3; t4 =< t5; t6 =< t7; t8 =< t9
effect:	in-camera(picture, Object, Color, S, E).

The compound task specification for the executive satellite in our satellite constellation scenario (figure 2) is specified as follows, where an individual satellite is responsible for transmitting to the earth the image that it captures:

compound-task decomposition:	full-spectrum-imagery(Object)@[S, E] infra-red(Object)@[t1, t2]; ultra-violet(Object)@[t3, t4]; visible(Object)@[t5, t6]
constraints:	S <= t1; t2 <= E; S <= t3; t4 <= E; S <= t5; t6 <= E
effect:	received-on-earth(image, Object, S, E).

Each *primitive task* (or atomic action) in the Spaceworld scenario is specified along with its *precondition* and *effect*. The precondition of a primitive task must be satisfied before the action can be executed. The effect is the effect on the environment after the task has been executed successfully. An example representation corresponding to the shutter-camera primitive task is provided below:

primitive-task precondition:	shutter-camera(Object)@[S, E] locked-onto(Object); shutter-speed(Speed); camera(on)@[t1, t2]; platform(still)@[t3, t4]
constraint:	[S, E] \subseteq [t1, t2]; [S, E] \subseteq [t3, t4]; Speed = E - S
effect:	in-camera(picture, Object, S, E).

The preconditions for the primitive task shutter-camera are as follows: the camera is locked onto the desired object, it is on, and the platform is still. The constraint $[S, E] \subseteq [t1, t2]$ states that the camera is locked on at least during the interval [S, E]. Once the action is performed, the picture of the object for the interval [S, E] is in the camera. The primitive task specifications corresponding to the turning on action of the camera is simpler:

primitive-task precondition:	turn-on@[S, E] camera(off)
constraint:	E - S = 30
effect:	camera(on).

Our explicit representation and handling with resources usage is evident in the following example specification corresponding to the recording action:

primitive-task precondition:	record-picture(Object, Color)@[S, E] in-camera(picture, Object, Color, t1, t2); tape-recorder(on)@[t3, t4]; data-mode(im2); tape-position(Start-Position); tape-empty(Start-Position, End-Position)
constraint:	[S, E] \subseteq [t3, t4]; E - S = 48; End-Position = Start-Position + 336;
effect:	tape-full(Start-Position, End-Position); recorded(Start-Position, End-Position, picture, Object, Color, t1, t2).

When a picture is recorded on a tape, the recorded portion of the tape resource becomes unavailable. This kind of effect on resources can be taken into account during the unfolding process of a plan construction to improve the efficiency in search for a solution. If an effect violates the

resource optimization function then this branch in the search space will not generate a potential solution.

5 DECENTRALIZATION AND COORDINATION

Decentralization or decomposition is the process of breaking down a problem into a set of subtasks. Since we have adopted a hierarchical modeling environment, there will be a natural way of decentralizing a task by an agent. We illustrate this process by considering the following specification of the compound task “send-picture-to-earth” already described in section 4. Suppose the image executive would like to produce a schedule for a picture. By looking at the above task decomposition it can decide to get this job done by the camera executive and the transmitter. Upon receiving a request from the image executive, the camera executive will follow the same decentralization process by using the composite task specification for picture-object.

When an agent decentralizes a task, it sends several requests to other agents. Correspondingly, the agent receives a set of replies according to the requests. It is not necessary that the agent receive the messages in the order they were sent. This is due to the fact that some agents are more efficient and some requests are harder to serve than others. Thus, every agent requires some amount of *coordination* of requests and answers. For example, messages may be tagged with priorities and an agent responds according to the priorities.

When the image executive agent sends the top-level task to its two subordinate agents then it expects two successful schedules, and their combination is the schedule of the whole task. It may so happen that one of the subordinate agents is not able to satisfy the constraint sent with the request. So the image executive may relax the constraint (e.g., by extending the interval) for that agent while imposing a constraint to another agent to compensate this. This process of relaxation and imposition of constraints is part of the coordination process.

In the above example specification, the image executive can achieve the goal in various ways. For example, it divides the interval [S, E] into two and asks two agents to plan in these two intervals so the constraint is automatically satisfied (Georgeff, 1983). The constraint need not be sent along with the request and the two agents can work simultaneously. Alternatively, it can ask the camera executive to work within the first half of the interval. If it fails then it can relax this constraint by stretching the interval. Another alternative approach is to let each agent cooperate with other agents to resolve any conflict. If an agent fails to provide a solution to a request, be it a planning or a resource request, an answer should provide reasons for failure in the constraint field.

We have argued that task decentralization by an agent and coordination is natural in a hierarchical modeling environment, which we have adopted for our distributed environment.

6 AGENT COMMUNICATION

Coherence, cooperation and conflict resolution can be improved by carefully designing the amount and type of communications among agents in the form of messages (Patil, et al., 1992). The information communicated should be relevant, timely and complete (Durfee, 1985). A message in our framework is composed of the following fields: 1) *sender*: sender of the message; 2) *receiver*: receiver of the message; 3) *identifier*: This is a unique identifier generated by the sender of the message; 4) *type*: a type describing whether a message is either a request or an answer to a request or an acknowledgement, etc.; 5) *task*: a task describes what the message is about, that is, whether it is planning (p), scheduling (s), resource allocation (r), or their combinations (p/s/r) or database related transaction such as insert, delete, update, lock and unlock; 6) *description*: in the case of a request this field describes the task requested, for example, description of a planning tasks. Similarly, in the case of an answer this field provides the answer of an earlier request or informing failure with explanations; 7) *constraint*: a constraint along with a request from a sender means that receiver meets the request by satisfying the constraint.

Following is an example of a p/s/r request message sent by the image executive agent to the camera executive agent, and its answer from the camera executive agent:

```
{m1, 'image executive', 'camera executive', request, p/s/r,
image(star, color)@[800, 1100], exclude([900, 950])}
{m1, 'camera executive', 'image executive', answer, p/s/r,
image(star, color)@[950, 1050]}
```

The request is for an image of any time in the interval [800, 1100] subject to the constraint that no image in [900, 950]. The answer from the camera executive is that the task will be carried out in [950, 1050].

7 PROTOTYPE IMPLEMENTATION

The software platform used for the prototype development is Java and we used multicast sockets for inter-agent communication.

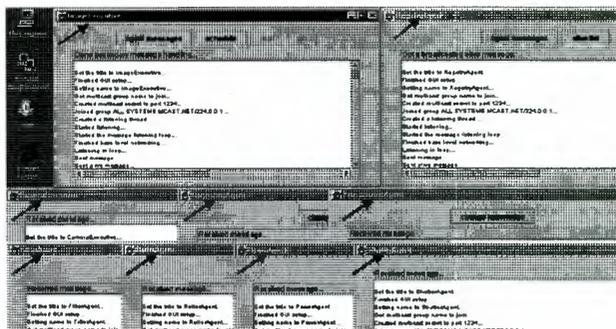


FIG 4: Prototype Implementation of the Spaceworld Multi-Agent Environment

Figure 4 shows the state prior to the start of scheduling of the implementation of the Spaceworld multi-agent environment. Each window is a separate process representing the agent pointed by a red arrow. All the agents are separate pieces of software, but we have

written a single unified program to launch the Java applications in different threads so only one Java virtual machine needs to be started up. Once all the agents are started up, they all negotiate with the Registry Agent to get confirmation that they are allowed to come online, and they negotiate with each other to allow to report ready to their executives. A newly registered agent obtains information about other agents in the environment from the Registry Agent. As shown in figure 5, the planning and scheduling process is initiated by pressing the 'schedule' button and a pop-up window will appear asking for the total time to be permitted (100 units) for the schedule. Using the hierarchical knowledge structure for 'send-picture-to-earth' specified in section 4, the executive agent then produces a schedule in cooperation with other agents.

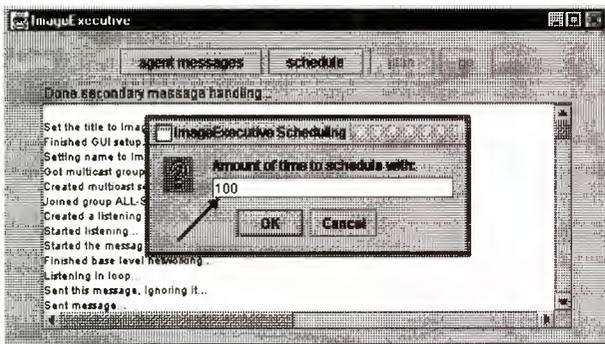


FIG 5: Schedule Request to the Image Executive Agent by Specifying the Allowed Time

The schedule produced can be viewed by pressing the 'plan' button as shown in figure 6.

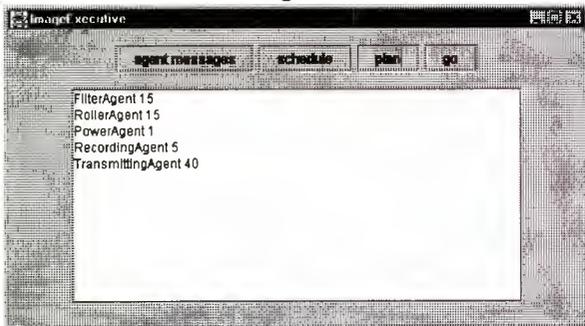


FIG 6: Schedule Produced by Image Executive Agent in Cooperation with Other Agents

Once scheduling has been finished, the actual execution begins by pressing the 'go' button. The Image Executive agent has the schedule already, so it simply sends the plan down the tree and each agent executes as needed and when needed. If a problem occurs somewhere down the line, then re-planning will be necessary. Within our Satellite Constellation scenario, there are many ways a schedule can be produced for a full spectrum sweep of a certain swath of the planet. Figure 7 is one such schedule produced by the prototype.

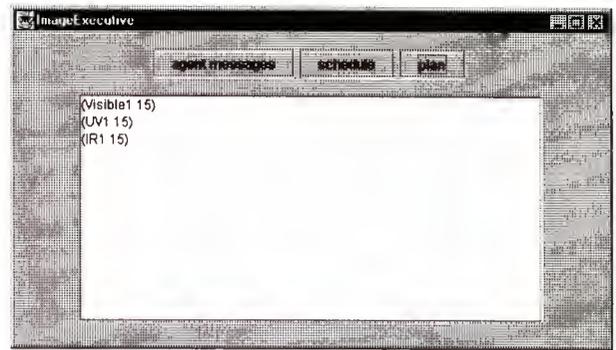


FIG 7: A Schedule Produced by the Prototype for a Full Spectrum Sweep

Our initial strategy was to accept the schedule that is found first during the search. For example, if the first satellite UV1 with ultra-violet imaging capability was not capable of carrying out the required task then the executive contacts the second satellite UV2 with the same capability and a schedule is constructed as shown in figure 8.

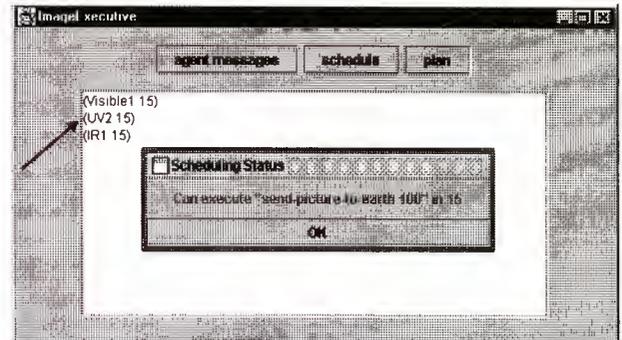


FIG 8: Revised Schedule of Figure 7 Using an Alternative for Ultra-Violet Spectrum

If both UV1 and UV2 are available then ideally the most optimized schedule among the two from the point of view of time and onboard resources should be produced. This kind of resource optimization issue will be addressed during our follow-on study. A constraint propagation paradigm usually allows encoding cost functions to produce the cost associated with each schedule.

8 CONCLUSION

In this paper, we have demonstrated how our distributed approach to planning and scheduling helps to achieve high-level goals and thereby enhances spacecraft autonomy. A hierarchical syntax has been adopted for representing domain knowledge of tasks by taking into account spacecraft operational aspects and resource constraints. The task decentralization problem has been solved by the use of the hierarchical knowledge structures. A constraint propagation paradigm has been employed for the required planning and scheduling tasks performed by an agent. The resource optimization problem has been addressed by its explicit representation within the problem domain. We have shown that a schedule can be generated (if it exists) and its quality can

be enhanced by the introduction of agent cooperation. A limited-scope prototype has been developed and demonstrated to assess overall feasibility.

This phase of the work has been carried out as part of NASA's effort on a program to develop and Remote Agents for flight software development. Various enhancements of our proposed distributed approach are planned during our follow-on effort.

Constraint propagation: Our current implementation of the constraint propagation activity is ad-hoc. For our follow-on development, we plan to use an off-the-shelf constraint-based temporal reasoning engine such as Honeywell's TMM (Time Map Manager), NASA's HSTS problem solving framework, and Prolog II software system. The advantage with a Prolog II type of declarative system is that it will allow us to perform high-level symbolic reasoning required as part of the planning and scheduling process, thus reducing the burden from the tedious development process in an imperative environment such as Java.

Resource optimization: The hierarchical representation of compound and primitive tasks of the application domain incorporates information about their resource consumption, and a database containing up-to-date resource status will be maintained. Therefore, during the hierarchical planning process, which unfolds a compound task into a set of subtasks and resolves task preconditions using the information in the resource database, the system can choose an unfolding path that consumes the least amount of resource. This process which we plan to implement guarantees an optimized plan to achieve the goal from the point of view of resource usage.

Inter-agent communication: We plan to take advantage of CORBA or KQML or ISP for enhanced cross-platform communication.

Inter-agent negotiation: Currently, we assume friendly relationship among agents and therefore no negotiations occurred between two agents. Although this is appropriate in an environment where agents represent onboard subsystems (e.g., the Spaceworld scenario), it may not be the case in a scenario involving a constellation of satellites owned by various companies, agencies, and countries. In the future, we will assume one of various types of relationships between two agents including friendly, subservient, and bargain. An agent is awarded or penalized according to its use of resources. The existence of bargain type relationship therefore introduces the possibility of negotiations between two agents without sacrificing their own interests.

Acknowledgements: This work was performed under contract NAS5-98120 with NASA's Goddard Space Flight Center.

9 REFERENCES

- Boddy, M. (1994). "Temporal reasoning for planning and scheduling." SIGART Bull. 4(3).
- Burke, P., and Prosser, P. (1991). "A distributed asynchronous system for predictive and reactive scheduling." Artificial Intelligence in Engineering 6: 106-124.
- CACM (1994). Intelligent Agents - Communication of the ACM, ACM Press.
- Chaib-draa, B. M., Mandiau, R., and Millot, P. (1992). "Trends in distributed artificial intelligence." AI Review 6: 35-66.
- Chapman, D. (1987). "Planning for Conjunctive Goals." Artificial Intelligence 32: 333-378.
- Chien, S., DeCosta, D., Doyle, R. and Stolorz, P. (1997). Making and impact: artificial intelligence at the Jet Propulsion Laboratory. AI Magazine. 18.
- Conry, S. M., R. and Lesser, V.R. (1988). Multiagent negotiation in distributed planning. Reading in Distributed Artificial Intelligence. A. a. G. Bond, L., Morgan Kaufmann: 367-384.
- Cooper, R. G. (1993). Winning at New Products: Accelerating the Process from Idea to Launch. New York, Addison Wesley.
- Currie, K. a. T., A. (1991). "The Open Planning Architecture." Artificial Intelligence 52(1).
- Das, S. K., J. Fox, et al. (1997). Decision Making and Plan Management By Autonomous Agents: Theory, Implementation, and Applications. First International Conference on Autonomous Agents, California.
- Dean, T., Firby, R. J., and Miller, D. (1988). "Hierarchical planning involving deadlines, travel time, and resources." Computational Intelligence 4: 381-398.
- Durfee, E. H., Lesser, V.R., and Corkill, D.D. (1985). Increasing coherence in a distributed problem solving network. 8th International Joint Conference on Artificial Intelligence.
- Ferguson, I. A. (1992). Touring Machines: AN Architecture for Dynamic, Rational. Mobile Agents, Computer Laboratory, University of Cambridge.
- Fikes, R. a. N., N. (1971). "STRIPS: A new approach to the application of theorem proving to problem solving." Artificial Intelligence 2: 189-208.
- Genesereth, M. R., and Ketchpel, S.P. (1994). "Software agents." C ACM 37: 48-53.
- Georgeff, M. (1983). Communication and interaction in multi-agent planning. AAAI.
- Georgeff, M. P. (1987). "Planning." Annual Review in Computer Science 2: 359-400.
- Le Pape, C. (1990). Constraint propagation in planning and scheduling.
- McAllester, D. a. R., D. (1994). Systematic nonlinear planning. AAAI-94.
- Muller, J. P. a. Pischel, M. (1994). Modeling interacting agents in dynamic environments. 11th European Conference on AI.
- Muscettola, N. (1994). Integrating planning and scheduling. Intelligent Scheduling. M. a. Z. Fox, M., Morgan Kaufmann.
- Patil, R. S., R. E. Fikes, et al. (1992). The DARPA Knowledge Sharing Effort: Proc. of Knowledge Representation and Reasoning (KR&R-92): 777-788.
- Pell, B., Bernard, D.E., Chien, S.A., Gat, E, Muscettola, N., Pandurang Nayak, P., Wagner, M.D., and Williams, B.C. (1997). An autonomous spacecraft agent prototype. 1st International Conference on Autonomous Agents.
- Rosenschein, J. S. (1982). Synchronization of multi-agent plans. AAAI, pp.115-119.
- Sacerdoti, E. (1974). "Planning in a hierarchy of abstraction spaces." Artificial Intelligence 5.
- Simmons, R. (1991). "Coordinating planning, perception, and action for mobile robots." SIGART Bulletin 2: 156-159.
- Tate, A. (1977). Generating project networks. 5th Int. Joint Conf. on Artificial Intelligence.
- Truszkowski, W., Odubiyi, J., and Ruberton, E. (1995). "An agent model in a multi-agent system architecture for automating distributed system." *Proceedings of the 1st ICMAS*, pp. 463.
- Vere, S. A. (1983). "Planning in Time: Windows and Duration for Activities and Goals." IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5(No. 3).
- Wilkins, D. E. (1988). Practical Planning: Extending the Classical AI Planning Paradigm, Morgan Kaufmann.

ITERATIVE REPAIR PLANNING FOR SPACECRAFT OPERATIONS USING THE ASPEN SYSTEM

Gregg Rabideau, Russell Knight, Steve Chien, Alex Fukunaga, Anita Govindjee

Jet Propulsion Laboratory, California Institute of Technology
 4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
 phone: +1 818 393-5364, fax: +1 818 393-5244, email: {firstname.lastname}@jpl.nasa.gov

ABSTRACT

This paper describes the Automated Scheduling and Planning Environment (ASPEN). ASPEN encodes complex spacecraft knowledge of operability constraints, flight rules, spacecraft hardware, science experiments and operations procedures to allow for automated generation of low level spacecraft sequences. Using a technique called *iterative repair*, ASPEN classifies constraint violations (i.e., *conflicts*) and attempts to repair each by performing a planning or scheduling operation. It must reason about which conflict to resolve first and what repair method to try for the given conflict. ASPEN is currently being utilized in the development of automated planner/scheduler systems for several spacecraft, including the UFO-1 naval communications satellite and the Citizen Explorer (CX1) satellite, as well as for planetary rover operations and antenna ground systems automation. This paper focuses on the algorithm and search strategies employed by ASPEN to resolve spacecraft operations constraints, as well as the data structures for representing these constraints.

1. INTRODUCTION

Planning and scheduling technology offers considerable promise in automating spacecraft operations. Planning and scheduling spacecraft operations involves generating a sequence of low-level spacecraft commands from a set of high-level science and engineering goals (see [Chien et al., 1998b] for an overview). We discuss ASPEN and its use of an *iterative repair* algorithm for planning and scheduling as well as for replanning and rescheduling.

ASPEN is a reconfigurable planning and scheduling software framework [Fukunaga et al., 1997]. Spacecraft knowledge is encoded in ASPEN under seven core model classes: activities, parameters, parameter dependencies, temporal constraints, reservations, resources and state variables. An activity is an occurrence over a time interval that in some way affects the spacecraft. It can represent anything from a high-level goal or request to a low-level event or command. Activities are the central

structures in ASPEN, and also the most complicated. A more detailed definition is given in a later section. Together, these constructs can be used to define spacecraft components, procedures, rules and constraints in order to allow manual or automatic generation of valid sequences of activities, also called *plans* or *schedules*.

Once the types of activities are defined, specific instances can be created from the types. Multiple activity instances created from the same type might have different parameter values, including the start time. Many camera imaging activities, for example, can be created from the same type but with different image targets and at different start times. The sequence of activity instances is what defines the plan.

The job of a planner/scheduler, whether manual or automated, is to accept high-level goals and generate a set of low-level activities that satisfy the goals and do not violate any of the spacecraft flight rules or constraints. ASPEN provides a Graphical User Interface (GUI) for manual generation and/or manipulation of activity sequences. However, the automated planner/scheduler will be the focus of the remainder of this paper.

In ASPEN, the main algorithm for automated planning and scheduling is based on a technique called *iterative repair* [Zweben et al., 1994]. During iterative repair, the conflicts in the schedule are detected and addressed one at a time until no conflicts exist, or a user-defined time limit has been exceeded. A conflict is a violation of a reservation, parameter dependency or temporal constraint. Conflicts can be repaired by means of several predefined methods. The repair methods are: moving an activity, adding a new instance of an activity, deleting an activity, detailing an activity, abstracting an activity, making a reservation of an activity, canceling a reservation, connecting a temporal constraint, disconnecting a constraint, and changing a parameter value. The repair algorithm may use any of these methods in an attempt to resolve a conflict. How the algorithm works is largely dependent on the type of conflict being resolved.

Specifically, we haven't taken a most-committed, local, heuristic, iterative repair approach. This approach has many desirable properties for spacecraft operations planning. First, using a repair algorithm allows automated planning to be utilized at any time and on any given initial plan. The initial plan may be as incomplete as a set of goals, or it may be a previously produced plan with only a few flaws. Repairing an existing plan enables fast replanning when necessary from manual plan modifications or from unexpected differences detected during execution. Second, heuristics allow the search to be pruned, ruling out less promising planning choices. In addition, heuristics may also suggest particular choices that may lead to a solution in less time, or to a higher quality solution. Third, a local iterative algorithm does not incur the overhead of maintaining intermediate plans or past repair attempts. This allows the planner to quickly try many plan modifications for repairing the conflicts. However, unlike systematic search algorithms, it cannot be guaranteed that our iterative repair algorithm will explore all possible combinations of plan modifications or that it will not retry unhelpful modifications. In our experience, these guarantees are not worth the required overhead. Finally, by committing to values for parameters, such as activity start times and resource usages, the effects of a resource usage and the corresponding resource profiles can be efficiently computed. Least-commitment techniques retain plan flexibility, but can be computationally expensive for large applications. Further discussions on this topic can be found in [Chien et al., 1998b].

The full paper will describe the ASPEN search structure in greater detail. We will describe the constraints that can be modeled in ASPEN, as well as the conflicts for each type of constraint violation. We will also describe how the search can be influenced using heuristics. Finally, we will discuss current, future and related work.

2. MODEL COMPONENTS AND CONSTRAINTS

Spacecraft models are developed in the ASPEN Modeling Language (AML) [B. Smith et al., 1998; Sherwood et al., 1998]. These models are parsed into data structures that provide efficient reasoning capabilities for planning and scheduling. There are seven basic components to an ASPEN model: activities, parameters, parameter dependencies, temporal constraints, resources, state variables, and reservations. Together, they describe what the spacecraft can and cannot do during operations.

A *parameter* is simply a variable with a restricted domain. One parameter, for example, can be the range of integers between ten and twenty. Other parameter types

include floating point numbers, booleans and strings. A *parameter dependency* is a functional relationship between two parameters. An activity end time, for example, is a function (the sum) of the start time and the duration. A more complicated dependency might compute the duration of a spacecraft slew from the initial and final orientation.

In the model, relative ordering constraints can be specified for pairs of activities. A *temporal constraint* is a relationship between the start or end time of one activity with the start or end time of another activity (see Figure 1). One might specify, for example, that an instrument warming activity must end before the start of an activity that uses the instrument. Minimum and maximum separation distances can be specified in a temporal constraint. The warming activity for example, might be required to end at least one second but at most five minutes before using the instrument. Temporal constraints can be combined with conjunctive or disjunctive operators to form more complicated expressions.

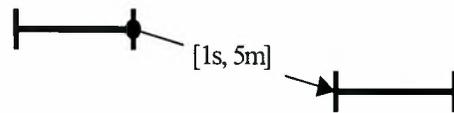


Figure 1: A temporal constraint with a required separation of at least 1 second and at most 5 minutes.

A *resource* represents the profile of a physical resource or system variable over time (see Figure 2), as well as the upper and lower bounds of the profile. In ASPEN, a resource can either be depletable or non-depletable. A depletable resource is used by a reservation and remains

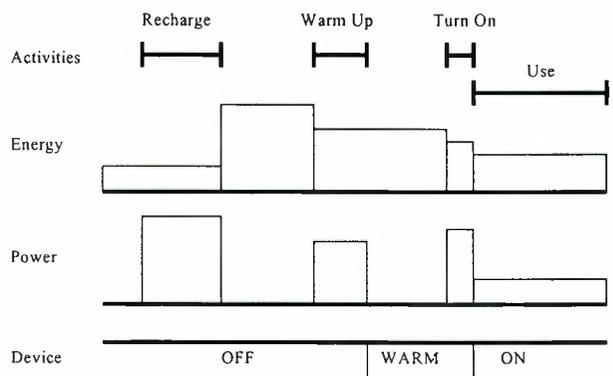


Figure 2: Timelines for activities, a depletable resource (energy), a non-depletable resource (power), and a state variable (device).

used even after the end of the activity making the reservation. Examples of depletable resources on spacecraft include memory, fuel and energy. A non-depletable resource is used only for the duration of the activity making the reservation. Power is an example of a non-depletable resource. A resources can be assigned a capacity, restricting its value at any given time. A *state variable* represents the value of a discrete system variable over time. The set of possible states and the set of allowable transitions between states are both defined with the state variable. An example of a state variable is an instrument switch that may be either ON, WARMING, or OFF. This state variable may be restricted to transitions from OFF to WARMING and not directly to ON. *Reservations* are requirements of activities on resources or state variables. For example, an activity can have a reservation for ten watts of power. Some reservations are modeled as instantaneous effects (e.g., reservations that change the state on a state variable). The user can specify whether this effect occurs at the start or end of the activity.

Activity hierarchies can be specified in the model using decompositions (see Figure 3). A decomposition is a set of sub-activities along with temporal constraints between them. In this way, one can define a high-level activity that decomposes into a set of lower-level activities that may be required to occur in some relative order. These activities in turn may have their own decompositions. In addition, an activity may have multiple decompositions to choose from. Thus, allowing an activity to be expanded in different ways.

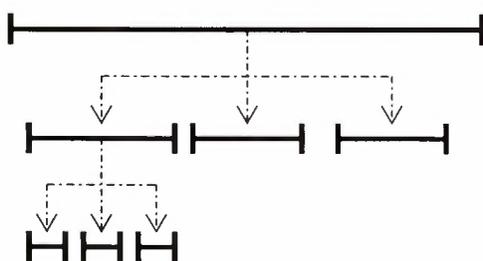


Figure 3: An activity hierarchy.

An *activity* has a set of parameters, parameter dependencies, temporal constraints, reservations and decompositions. All activities have at least three parameters: a start time, an end time and a duration. There is also at least one parameter dependency, relating these three parameters. In addition, all activities have at least one temporal constraint that prevents the activity from occurring outside of the planning horizon. Any additional components are optional.

3. CONFLICTS

A complete plan may not always be consistent with the constraints in the model. A conflict is a violation of one of the model constraints. We define ten basic types of conflicts in ASPEN:

- Abstract activity conflicts
- Unassigned parameter conflicts
- Violated parameter dependency conflicts
- Unassigned temporal constraint conflicts
- Violated temporal constraint conflicts
- Unassigned reservation conflicts
- Depletable resource conflicts
- Non-depletable resource conflicts
- State requirement conflicts
- State transition conflicts.

Each conflict provides information about what objects are involved and how to repair the conflict.

An *abstract activity conflict* is simply an activity that has not yet been decomposed into its sub-activities. All activities must be expanded to their most detailed level. If an activity has more than one decomposition, the planning algorithm must decide which decomposition to use when detailing the activity. Detailing an activity involves creating instances of the activities specified in the decomposition. In addition, all temporal constraints and parameter dependencies must be connected among the new sub-activities and the parent activity.

An *unassigned parameter conflict* is a parameter that does not have a unit value. A parameter can be a range or a set of values. However, this is a conflict in the plan until a value, chosen from the range or set, has been assigned to the parameter.

A *violated parameter dependency conflict* is a violation of a functional relationship between two parameters. In other words, the value of a parameter is not equal to the result of a function that constrains that parameter value. For example, a parameter p may be required to be the square of another parameter q . If q is assigned to 5 and p is assigned any value other than 25, this will be a parameter dependency conflict. This conflict can be resolved by assigning a different value to either p or q .

An *unassigned temporal constraint conflict* occurs when a temporal constraint exists for an activity, but an activity instance has not been selected to satisfy the constraint (see Figure 4). A temporal constraint is defined in one activity type A and specifies the requirement for another activity B within some temporal relationship. When an instance of A is created, the temporal constraint is created and is not initially assigned an instance of B . The conflict

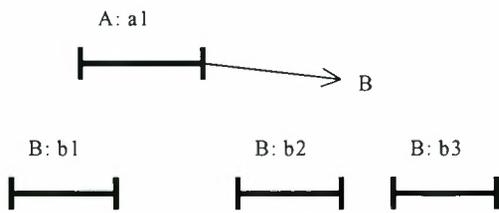


Figure 4: An unassigned temporal constraint conflict requiring an activity of type B. Any of b1, b2 or b3 can be used, or a new instance of type B can be added.

computes all activity instances that can repair this conflict (basically, all instances of type B).

A *violated temporal constraint conflict* occurs when a temporal constraint has been assigned, but the relationship (specified in the model) does not hold for the two participating activities (see Figure 5). For example, consider an activity instance A that must end before the start of activity instance B by at least 10 seconds but at most 1 minute. If A ends at time t , then there is a conflict if B does not start between time $t+10$ and $t+60$. The conflict keeps track of the contributing activities, which in this example includes activities A and B. In addition, the conflict computes the start time intervals for moving an activity that would repair the conflict. Continuing with the example, the repair interval for B would be from $t+10$ to $t+60$. Activity A could also be moved to a different repair interval.

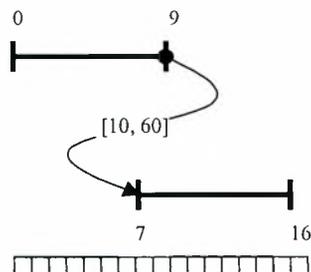


Figure 5: A violated temporal constraint conflict.

An *unassigned reservation conflict* is a reservation in an activity that has not been assigned to a resource or state variable of the required type. Resource and state variable types are defined in the model, and the plan can have multiple instances of the same type (e.g., multiple power sources). The plan keeps a timeline for each instance representing the value of the resource or state variable value. An unassigned reservation conflict is repaired by selecting a resource or state variable instance and making

the reservation (i.e., propagating the effects of the reservation on the timeline).

The most complicated types of conflicts are *violated timeline conflicts*. A conflict can occur on a depletable resource, a non-depletable resource, or a state variable. For state variables, there are two types of conflicts: state usage and state transition conflicts.

When a resource value at a particular time exceeds the minimum or maximum bounds of the resource, a conflict is generated. The contributing activities are the activities with reservations that use the resource during the time of the conflict (see Figures 6 and 7). For non-depletables, these are the reservations that overlap, exceeding the resource bounds. For depletables, these are all reservations on the timeline that occur at or before the conflict. If the value is above the resource maximum (i.e., *overuse*), then contributors are only those activities with reservations that reserve a positive value. Those with negative values are contributors when the resource value is below the minimum (i.e., *underuse*). The conflict also knows which activity types would repair the conflict if a

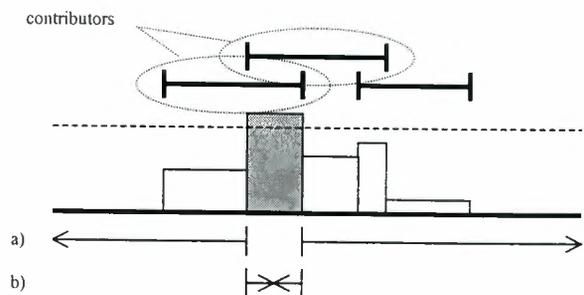


Figure 6: Time intervals that resolve a non-depletable resource conflict by a) moving a positive contributor or b) adding a negative contributor.

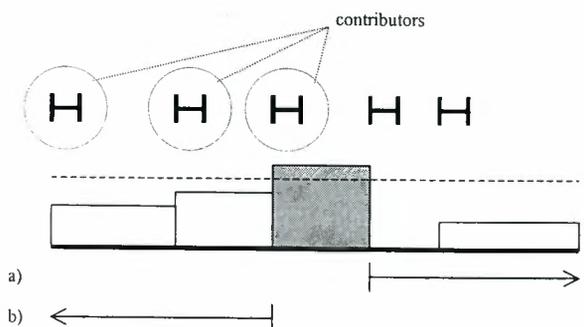


Figure 7: Time intervals that resolve a depletable resource conflict by a) moving a positive contributor or b) adding a negative contributor.

new instance were created. This includes activity types with negative usage for overuse conflicts and types with positive usage for underuse conflicts. The conflict also computes the start times indicating where to move or add activities in order to repair the conflict (see Figures 6 and 7). For moving existing activities, repair start times are all times except during the conflict. For adding new activities, repair start times are just the opposite—times during the conflict.

A state variable can have a conflict in two ways: when a reservation requires a state that is not available for the duration of the reservation (i.e., state requirement conflict), or when a reservation makes a transition that is not allowed by the state variable (i.e., state transition conflict). The contributors of a state usage conflict include the activity that changes the state (called a *changer*) and all activities that use a state (called *users*) that is different from the state during the time of the conflict (see Figure 8). In order to fix this conflict, the users might be moved anywhere but over the state in conflict. Otherwise, if we decide to move the changer, it must be moved to a time later than the state in conflict or earlier than the previous state so that this changer no longer affects the state required by the conflicting users. For state transition conflicts, the contributor is only the activity that changes the state (i.e., makes the illegal transition). Again, the changer must be moved to a time later than the state in conflict or earlier than the previous state. As with resource conflicts, new activities can be created to repair state variable conflicts. For a state usage conflict, we can add activities that can change to the desired state. These activities must be added at a time before the conflicting user, but after the conflicting changer. For state transition conflicts, we can add

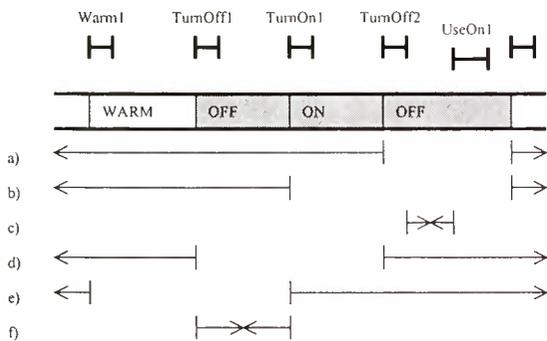


Figure 8: Time intervals that resolve a state variable usage conflict by a) moving UseOn1 b) moving TurnOff2 or c) adding TurnOn; and time intervals that resolve a state variable transition conflict by d) moving TurnOn1 e) moving TurnOff1 or f) adding Warm.

activities that can change to a state that makes a legal transition. These activities must be added between the two conflicting changers.

4. ITERATIVE REPAIR SEARCH

ASPEN organizes its search around several types of constraints that must hold over valid plans. ASPEN then has organized around each constraint type, a classification of the ways in which the constraint may be violated. These violations are called conflicts. Organized around each conflict type, there is a set of repair methods. The search space consists of all possible repair methods applied to all possible conflicts in all possible orders. We describe one tractable approach to searching this space.

The iterative repair algorithm searches the space of possible schedules in ASPEN by making decisions at certain choice points, and modifying the schedule based on these decisions. The choice points are:

- Selecting a conflict
- Selecting a repair method
- Selecting an activity for the chosen repair method
- Selecting a start time for the chosen activity
- Selecting a duration for the chosen activity
- Selecting timelines for reservations
- Selecting a decomposition for detailing
- Selecting parameters to change
- Selecting values for parameters

Given a schedule with a set of conflicts of all types, the first step in the iterative repair algorithm is to select one of the conflicts to be attacked. Next, a method is selected for repairing the conflict. We define the possible repair methods as:

- Moving an existing activity to a new location
- Creating a new activity and insert at a location
- Deleting an existing activity
- Connecting a temporal constraint between two activities
- Disconnecting a temporal constraint between two activities
- Detailing an activity
- Abstracting an activity
- Making reservations of an activity
- Canceling reservations of an activity
- Grounding a parameter in an activity
- Applying a dependency function between two parameters

As described in the previous section, the type of conflict will determine the set of possible repair methods for any given conflict. If it was decided to try to move or delete an activity, the algorithm must decide which activity to move or delete. The type of conflict and the location of the conflict will determine the set of possible activities

that, if moved or deleted, may resolve the conflict. In addition, a new start time and duration must be assigned to the activity. If it was decided to try to add a new activity, the activity type must be chosen from the list of possible types determined by the conflict. For abstract activity conflicts, the repair algorithm will most likely choose to detail the activity. If it has multiple decompositions, one of them must be chosen. Deciding to abstract an activity requires choosing which activity to abstract. When making a reservation in an attempt to resolve a conflict, a resource or state variable must be chosen for the set of possible resources or state variables. Also, if the reservation has an unspecified value, one must be chosen for it. Canceling reservations only requires choosing which reservation to cancel. If the repair algorithm has decided to connect a temporal constraint, the specific activity for the constraint must be selected. When disconnecting, only the constraint to be disconnected must be chosen. Finally, changing a parameter value requires choosing a new value for the parameter. After all decisions are made and the repair method is performed, the effects are propagated and the new conflicts are computed. This process repeats until no conflicts exist or a time limit has been exceeded.

5. SEARCH HEURISTICS

All throughout the iterative repair algorithm, many decisions must be made. In other words, there are many ways in which a conflict may be resolved. Some ways ultimately work better than others do. For example, deleting an activity may resolve a resource conflict caused by that activity. However, that activity may have been required by other activities. Or, if the activity was a high-level goal, the user might prefer to have as many goals satisfied as possible. Another typical example involves choosing a location to move an activity. Many locations may resolve the conflict being addressed, but many locations may also create additional conflicts. In order to guide the search toward more fruitful decisions, the user can define a set of search heuristics.

In ASPEN, a heuristic is a function that orders and prunes a list of choices for a particular decision in the search. Heuristics can be defined at each of the choice points in the algorithm. For example, one heuristic might sort the list of conflicts, indicating which conflicts to address first. In addition, each heuristic can use the knowledge of all previous decisions made. For example, the heuristic for deciding which method to use to resolve the conflict can (and should) be dependant on which conflict was chosen. Each heuristic can be assigned a confidence level that indicates how often the heuristic should be used. When

the heuristic is not used, other heuristics can be specified, otherwise the decision will be made randomly.

ASPEN currently has some built-in domain-independent heuristics that can be used for repairing conflicts. First, a heuristic exists for sorting conflicts by their type. This heuristic prefers conflicts that require new activities (i.e., planning type conflicts) and then considers conflicts on timelines (i.e., scheduling type conflicts). This heuristic seems to work well and therefore has a high level of confidence for most of our models.

There is also a heuristic for selecting the repair method for a given conflict type. This heuristic prefers moving activities for repairing most types of conflicts. If move is not selected, the next preferred method is adding new activities. Finally, a small percentage of the time, it will choose to delete an activity. Obviously, these methods are only chosen for those conflicts for which they make sense (e.g., timeline conflicts). Some conflicts have only two possible repair methods, one of which is to delete, therefore making the decision much easier (e.g., undetailed activity conflicts can only be resolved by detailing or deleting the activity).

Another significant heuristic available in ASPEN is a heuristic for selecting start time intervals for activities being moved or created. This heuristic first tries selecting start time intervals that not only resolve the current conflict but also do not create any new conflicts¹. If there are no such start times, the heuristic may try selecting times that create only a few conflicts. If this list is also empty, then it may select start times that simply resolve the current conflict. Sometimes, however, it may decide to return an empty list, indicating that this particular activity should not be moved or added.

A few other heuristics are currently being used in some of the domains modeled in ASPEN. All of them, however, are relatively simple and work well for the wide range of ASPEN models.

6. RELATED WORK

This work builds on considerable previous work in iterative repair problem solving. The high-speed local search techniques used in ASPEN are an evolution of those developed for the DCAPS system [Chien et al.,

¹ In general, ASPEN provides functions for querying the current plan about operations that can be performed or values that can be assigned without creating new violations. These algorithms are interesting in their own right, and will be discussed in future work.

1998a] that has proven robust in actual applications. In terms of related work, iterative algorithms have been applied to a wide range of computer science problems such as traveling salesman [Lin & Kernighan, 1973] as well as Artificial Intelligence Planning [Chien & DeJong, 1994; Hammond, 1989; Simmons, 1988; Sussman, 1973]. Iterative repair algorithms have also been used for a number of scheduling systems. The GERRY/GPSS system [Zweben et al., 1994; Deale et al., 1994] uses iterative repair with a global evaluation function and simulated annealing to schedule space shuttle ground processing activities. The Operations Mission Planner (OMP) [Biefeld & Cooper, 1991] system used iterative repair in combination with a historical model of the scheduler actions (called chronologies) to avoid cycling and getting caught in local minima. Work by [Johnston & Minton, 1994] shows how the min-conflicts heuristic can be used not only for scheduling but also for a wide range of constraint satisfaction problems. The OPIS system [S. Smith, 1994] can also be viewed as performing iterative repair. However, OPIS is more informed in the application of its repair methods in that it applies a set of analysis measures to classify the bottleneck before selecting a repair method. With iterative repair and local search techniques, we are exploring approaches complementary to backtracking refinement search approach used in the New Millennium Deep Space One Remote Agent Experiment Planner [Muscettola et al., 1997].

7. CONCLUSIONS AND FUTURE WORK

Planning and scheduling technology offers considerable promise in automating spacecraft operations. Planning and scheduling spacecraft operations involves generating a sequence of low-level spacecraft commands from a set of high-level science and engineering goals. We have extended and implemented a technique called *iterative repair* for automatically resolving conflicts in a plan/schedule. In addition, we have isolated a set of conflict types that identify plan violations as well as suggest ways in which to repair the violation.

Current and future work includes integrating repair planning with execution [Chien et al., 1999]. Here, the idea is to continuously replan around updated information coming from execution monitoring. As an embedded system, ASPEN would enable fast response to unforeseen events (e.g., faults or science opportunities) with little or no human interaction. In addition, we are also working on a framework for plan optimization. In this case, the objective is to find plans with high quality in addition to being conflict-free. We take an approach that parallels iterative repair called *iterative optimization*. Here, we

classify a set of user preferences for certain plan characteristics. These preferences are used to calculate a score for the plan. The iterative optimization algorithm makes plan modifications suggested by the preferences in order to increase the overall score.

ACKNOWLEDGEMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

- Biefeld, E., and Cooper, L., "Bottleneck Identification Using Process Chronologies," *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.
- Chien, S., and DeJong, G., "Constructing Simplified Plans via Truth Criteria Approximation," *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, Chicago, IL, June 1994, pp. 19-24.
- Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Integrated Planning and Execution for Autonomous Spacecraft," *Proceedings of the 1999 IEEE Aerospace Conference*, Aspen, CO, March, 1999.
- Chien, S., Rabideau, G., Willis, J., and Mann, T., "Automating Planning and Scheduling of Shuttle Payload Operations," *Artificial Intelligence Journal Special Issue on Applications*, 1998.
- Chien, S., Smith, B., Rabideau, G., Muscettola, N., Rajan, K., "Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft," *IEEE Intelligent Systems*, September/October, 1998, pp. 50-55.
- Deale, M., Yvanovich, M., Schnitzius, D., Kautz, D., Carpenter, M., Zweben, M., Davis, G., and Daun, B., "The Space Shuttle Ground Processing System," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.
- Fukunaga, A., Rabideau, G., Chien, S., and Yan, D., "Toward an Application Framework for Automated Planning and Scheduling," *Proceedings of the 1997 International Symposium of Artificial Intelligence, Robotics and Automation for Space (iSAIRAS-97)*, Tokyo, Japan, July 1997.

Hammond, K., "Case-based Planning: Viewing Planning as a Memory Task," Academic Press, San Diego, 1989.

Johnston, M., and Minton, S., "Analyzing a Heuristic Strategy for Constraint Satisfaction and Scheduling," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

Lin, S., and Kernighan, B., "An Effective Heuristic for the Traveling Salesman Problem," *Operations Research* Vol. 21, 1973.

Minton, S., and Johnston, M. "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems," *Artificial Intelligence*, 58:161-205, 1988.

Muscettola, N., Smith, B., Chien, S., Fry, C., Rajan, K., Mohan, S., Rabideau, G., Yan, D., "On-board Planning for the New Millennium Deep Space One Spacecraft," *Proceedings of the 1997 IEEE Aerospace Conference*, Aspen, CO, February, 1997, v. 1, pp. 303-318.

Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., Fukunaga, A., "Using ASPEN to Automate EO-1 Activity Planning," *Proceedings of the 1998 IEEE Aerospace Conference*, Aspen, CO, April, 1998.

Simmons, R., "Combining Associational and Causal Reasoning to Solve Interpretation and Planning Problems," Technical Report, MIT Artificial Intelligence Laboratory, 1988.

Smith, B., Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., Fukunaga, A., "Representing Spacecraft Mission Planning Knowledge in ASPEN," *Artificial Intelligence Planning Systems Workshop on Knowledge Acquisition*, Pittsburgh, PA, 1998.

Smith, S., "OPIS: An Architecture and Methodology for Reactive Scheduling," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

Sussman, G., "A Computational Model of Skill Acquisition," Technical Report, MIT Artificial Intelligence Laboratory, 1973.

Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," *Intelligent Scheduling*, Morgan Kaufmann, San Francisco, 1994, pp. 241-256.

CITIZEN EXPLORER – A LOW-COST, DISTRIBUTED
AND INCREMENTALLY AUTOMATED MISSION OPERATIONS SYSTEM

Colette Wilklow
Joseph Antell
Adrienne AuCoin
Shankini Doraisingam
Jack Faber
Elaine Hansen

Colorado Space Grant Consortium
University of Colorado
Campus Box 520
Boulder, Colorado 80309-0520
Phone: (303) 492-3141, Fax: (303) 492-
Email: (wilklow, antell, adrienne, doraisin, faberj, elaine)@greco.colorado.edu

ABSTRACT

The Colorado Space Grant Consortium (CSGC) has developed a low-cost, distributed and incrementally automated mission operations system. Automation technologies are being applied to support mission operations flexibility, user interaction, beacon assessment, autonomy migration, failure mode and effects analysis, and the use of integrated planning and scheduling capabilities. The application of such technologies becomes critical for low-budget spacecraft missions. Our goal is to implement a system that is not cumbersome, that can be learned quickly, that can automatically monitor the health and status of the spacecraft and respond to any anomalies. Automating components of the operations system will result in more cost-effective operations due to a significant reduction in operations staffing. In addition, automation allows us to focus on complex fault handling, opportunity management and the demonstration of new technologies.

The forum we will use to demonstrate our mission operations system is the Citizen Explorer-I (CX-I) spacecraft currently being designed and built by a team of undergraduate and graduate students at the University of Colorado, Boulder. CX-I is designed for a sun-synchronous circular orbit with a 10:15AM/10:15PM equator crossing and an altitude of approximately 700 kilometers. This orbit will provide limited access time with our ground stations in Boulder, Colorado and Fairbanks, Alaska. Due to the limited access time, it is imperative that we implement a robust planning and scheduling scheme. To meet this need, CSGC has developed the Citizen Explorer Design and Operations Planning System (CXDOPS). CXDOPS (Figure 1) was designed to support mission design, systems modeling and mission operations. Several components make up this

system. The first is a Commercial Off-The-Shelf (COTS) tool called the Satellite Tool Kit (STK). STK was developed and donated to CSGC by analytical Graphics Incorporated (AGI). STK is a satellite system modeling, analysis and visualization tool. STK is used to generate reports to be used by CXDOPS. These reports include ground track information, ground station access times, terminator crossing times and other data relevant to the spacecraft model.

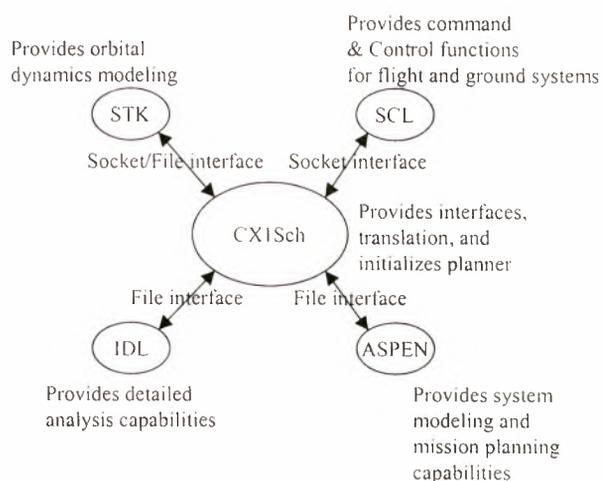


Figure 1: Citizen Explorer Design and Operations Planning System

The next component is another Off-The-Shelf (OTS) tool called ASPEN (Automated Scheduling and Planning Environment). ASPEN is a suite of planning and scheduling software developed by and provided to CSGC by the Jet Propulsion Laboratory (JPL) for use with spacecraft missions. This software generates sequences of low-level spacecraft commands from a specified set of high level science

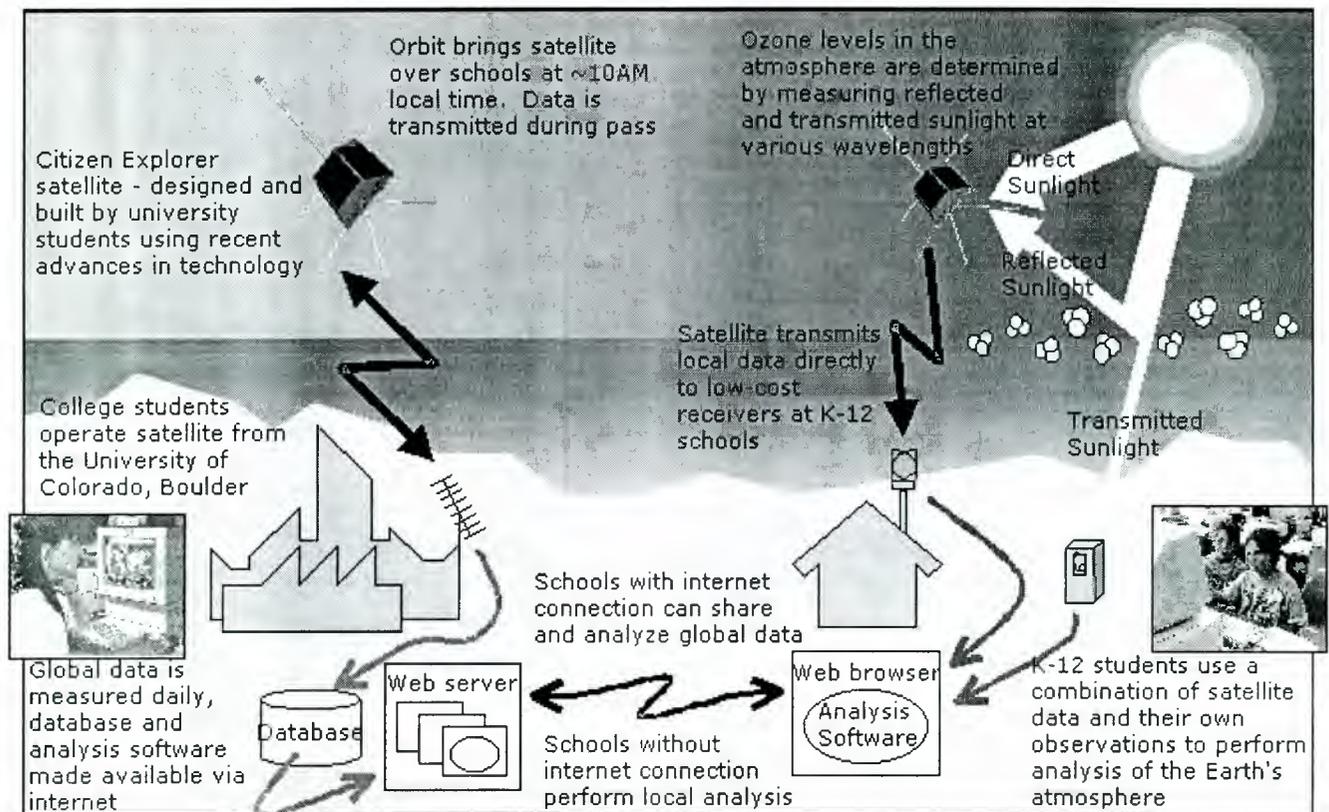


Figure 2: CX Concept Architecture

and engineering goals. ASPEN encodes complex spacecraft operability constraints, flight rules, spacecraft hardware models, science experiment goals and operations procedures. From these inputs, ASPEN automatically generates low-level spacecraft sequences. ASPEN is used in CXDOPS for system design visualization and mission planning.

The heart of CXDOPS is CX1Sch (Citizen Explorer – I Scheduler). CX1Sch is a Java program developed at CSGC, which provides transparent communication interfaces between each component of CXDOPS. CX1Sch also provides the Graphical User Interface (GUI) to CXDOPS. The GUI allows the operator to adjust various parameters, which enables them to perform several tasks using the CXDOPS system, including running various simulations which assist in the design process and system modeling. Planning and scheduling scenarios can be run for various portions of the mission. During the mission, CXDOPS will be used to perform orbit prediction and analysis, determine access times, perform spacecraft analysis, create goals and activities for the planning system and optimize schedules.

Failure Mode and Effects Analysis (FMEA) is being used to make decisions prior to launch that are reflected in flight and ground software in the form of scripts, rules and constraints implemented in

Spacecraft Command Language (SCL). SCL, a product of Interface and Control Systems, Inc., is a portable, distributed, intelligent command and control system. Decisions are based on the results of the FMEA process where first, the subsystem teams identify potential failures. Those who designed the system and are familiar with it most easily predict failures at this level. The criticality of these failures is preliminarily ranked by the subsystems. The main measures that are taken into account are impact of occurrence (what effect this failure would have on the mission if it were to occur), probability of failure (the likelihood of this failure occurring) and detection (the ease and promptness of detection). A secondary measure is also used to determine the chance of controlled recovery. This information is used both within the subsystem and at an overall systems level to prioritize and concentrate mitigation efforts. In addition to analyzing the criticality of each failure, the cause of failure and symptoms are documented. From this point both mitigation and contingency plans are formulated and the flight software is updated to reflect these plans. Using SCL, we will migrate knowledge gained to the flight system both during system integration and throughout the mission. SCL will be interfaced to the CXDOPS system, specifically, the low-level spacecraft sequences generated by ASPEN will be mapped to sequences in SCL. Sequences generated through CXDOPS can

then be uplinked and executed on the spacecraft.

Another OTS tool we will be using for the CX-I mission is SELMON (Selective Monitoring). SELMON was also developed and provided by JPL. SELMON and SCL together perform realtime monitoring and evaluation of engineering health and status data. Using SCL and SELMON data, beacon summaries of overall system performance are produced. These beacon summaries will be sent as spacecraft telemetry, which will be received by mission operators at the beginning of each pass, allowing the operators the maximum amount of time to react to any changes in system performance. To further enhance the usefulness of the beacon assessment technology, we will downlink the beacon summaries to K-12 schools participating the CX-I program. Schools across the state of Colorado, the United States, and eventually throughout the world will be equipped with an EduStation (Educational Ground Station) which consists of a receiver and a personal computer running a variety of applications software. The EduStation will enable K-12 students to monitor the spacecraft as it passes overhead. In the event that a school detects a change in the spacecraft performance, they can alert the CX-I mission operations team, giving them time to prepare the necessary plan of action for the next Boulder or Fairbanks pass.

The high level information displayed to students using the EduStation is the end product of the entire EEMOS (End-to-End Mission Operations System) software system. EEMOS consists of embedded flight software running aboard the spacecraft as well as a wide array of proprietary and COTS applications running on the ground, which produce downlink, parse, process and store CX-I data. These applications will produce secondary data products that will be sent to and displayed at the EduStation while all of the underlying distributed software will remain transparent at the EduStation level.

The front-end software (GUIs) for the mission operations system provides users with a high level interface to a complex and distributed software system by making extensive use of Web applications written using a COTS software package called Sammi (Kinesix Corporation), as well as HTML, Java and CGI. This approach enables a powerful, distributed, cross-platform, easy-to-use interface for all users by making extensive use of the well established web based protocols and software. The EduStation interface running at participating K-12 schools will present informative displays that clearly represent both engineering and science data. Software

applications written in Java will enable K-12 students to view 2D graphical displays that reflect the current health and status of the satellite. Schools will receive broadcast packets during a satellite pass that will contain valuable science and engineering data. Java applications will parse the raw data that is received and create a series of images that will include thermometers and power gauges representing current voltages and temperatures of the satellite's subsystems. Beacon summaries that are included in the downlink will also be graphically displayed. Science data taken from the satellite and K-12 handheld devices will be calibrated to determine more accurate UV and ozone data values. Science data will be represented by a series of Java applets and IDL images that will include line, bar and contour graphs. Both science and engineering data will be displayed through informative and interactive graphical user interfaces at participating K-12 schools that will help the CX-I operations team monitor the spacecraft.

The object-oriented design methods used to implement the Java applications and applets will encourage reuse and maintainability. Software applications are being designed with abstract interfaces and minimal dependencies between classes. This object-oriented implementation will reduce future costs that are required to reuse and maintain the software applications. The CX-I mission and future missions will benefit from Java's cross-platform capabilities, minimizing the need to port to different platforms used by the K-12 schools. Java distributed nature also encourages high-level support for networking. Java applets will be used to frequently download and display historical science data across the Internet. The Graphical User Interfaces running on the schools' Web browsers will benefit from Java's built-in multithreaded environment by improving the interactive performance. Java's object-oriented, interpreted, distributed, and robust characteristics make it an ideal choice for the CX-I ground-level software development.

The mission operations interface, also sitting atop the aforementioned distributed EEMOS system framework, will provide both scientists and engineers with powerful command and monitoring abilities. Scientists will have access to both the ground and flight science data archives as well as realtime control of the science instrument aboard the spacecraft. Operators will have the ability to issue commands, monitor sensors, run diagnostics, access both ground and flight databases, update the SCL project and database and perform code updates. Performing such updates allows for a high level of flexibility in the mission operations system. CSGC heritage has

proven this flexibility is vital for a robust and optimized system as it allows the mission operations team to tailor the system to best meet the needs of the spacecraft. At the start of the mission, before spacecraft flight behavior is well characterized, operations personnel monitor the system very closely. This requires a larger operations team to handle any anomalies that might occur or to make changes to the flight and/or ground system. As the mission progresses and spacecraft performance is characterized, operators introduce increased autonomy via additions and updates to existing scripts, rules, constraints, sequences, and operational procedures. This capability enables the system to be incrementally upgraded based on the needs of the spacecraft. Once these updates have been performed and the system has been sufficiently tested with the new updates, the cost of operations is decreased by allowing a significant reduction in operations staffing.

The demonstration of the CX-I EEMOS will result in a mission operations system that can be applied to a variety of spacecraft missions. EEMOS demonstrates that robust, automated operations systems can indeed be developed for missions with a modest budget.

REFERENCES

Willis, Jason R., G. Rabideau, C. Wilklow, "The Citizen Explorer Design and Operations Planning System," IEEE Aerospace Conference 1999.

Grusin, M. Citizen Explorer Critical Design Review (CDR) Document 1998. Colorado Space Grant College, Boulder, CO

Riddle, E and J. Owens. Citizen Explorer: A Small Satellite for the Promotion of Education. 1997. Colorado Space Grant College, Boulder CO

The Real-Time Execution Performance Agent An Approach for Balancing Hard and Soft Real-Time Execution for Space Applications

Sam Siewert, Gary Nutt, and Elaine Hansen
Department of Computer Science
University of Colorado, Boulder, CO 80309-0430
siewerts@rodin.colorado.edu

ABSTRACT

Use of AI (Artificial Intelligence) algorithms such as adaptive planners, intelligent monitors, and data miners can help optimize overall return from space systems by providing adaptive operations that can exploit opportunities. Typically, space systems involve many hard real-time functions including: attitude, thermal, propulsion, and mechanism control; detector/sensor data stream processing; telemetry gathering and packetization; command handling; and many other periodic tasks which must be executed such that processing is completed by a periodic deadline. While there has been a concerted effort to design AI algorithms to have predictable execution requirements (e.g. anytime algorithms), most of these applications are relegated to running in a best effort fashion using slack time left over from the hard real-time periodic tasks which must be given higher priority to ensure safety and control. The problem with executing the AI algorithms in slack time is that this makes their execution performance impossible to predict. The alternative of requiring AI algorithms to be anytime algorithms so that they can be treated like a hard real-time task with a deterministic minimum response time can be prohibitive since such algorithms are hard to design and the minimum response may not provide much of an optimization. This paper describes a third alternative which provides an intelligent execution control mechanism, the EPA (Execution Performance Agent), that ensures execution of algorithms based on required reliability and confidence in meeting deadlines rather than priorities. The EPA provides predictable and

safe execution of hard real-time safety critical and soft real-time mission optimizing tasks. By analogy, the EPA provides a balancing capability much like the everyday ability people have to walk without tripping while contemplating how to build a better career. It does this by executing tasks in specific execution reliability and confidence space and monitoring actual execution times to determine when resources must be adjusted. The EPA is currently being evaluated in a digital control and continuous video media testbed at the University of Colorado. Based upon testbed results, the EPA is also being considered for execution control of real-time operating system tasks including AI and digital control applications on a small spacecraft, Citizen Explorer, being built by the Colorado Space Grant College. The EPA was inspired by experience with a Space Grant Space Shuttle small payload which included control of three instruments and optimization of their operations using an adaptive planner and an intelligent monitoring system from the NASA Jet Propulsion Laboratory. The requirements for both hard real-time tasks and the use of AI applications on Citizen Explorer will be more demanding, and it is hoped the EPA can be shown to increase reliability and predictability of such systems. Details of the EPA mathematical formulation, the testbed implementation, performance results, and results of the analysis to determine if the EPA meets the Citizen Explorer requirements will be discussed in the paper.

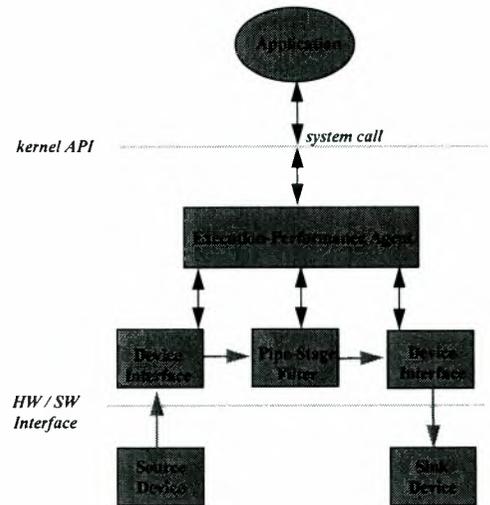
1.0 Introduction

The RTEPA (“Real-Time Execution Performance Agent”) mechanism introduced in this paper is intended to provide time-critical applications with quantifiable assurance of system response using a simple EPA (“Execution-Performance Agent”) interface to the deadline monotonic scheduling algorithm. In addition, the RTEPA provides a system call and signal interface which allows applications to monitor and control pipeline real-time performance on-line, and therefore significantly extends existing work on “in-kernel” pipelines. The set of applications requiring this type of performance negotiation support from an operating system is increasing with the emergence of virtual reality environments [Nu95], continuous media [Co94], multimedia [Ste95], digital control, and “shared-control” automation [Bru93][SiNu96]. The RTEPA mechanism is being implemented in the VxWorks microkernel, and is being tested in a rail-guided air-powered vehicle testbed incorporating continuous media, digital control, and “shared-control” pipelines. Likewise, the RTEPA is being tested with a 5 DOF robot arm that provides basic pick-and-place capabilities.

Traditionally, if an application requires service time assurances, there are three approaches: best-effort systems, hard real-time systems, and application specific embedded systems. Best-effort systems rely upon adequate resources always being available whenever an arbitrary task requests service, and can make no guarantees when they are even temporarily overloaded. Hard real-time systems require that the application provide resource bounds (e.g., the “Worst-Case Execution Time” or WCET) so that the operating system can mathematically check schedulability and admit only tasks whose complete execution can be guaranteed by hard deadlines. Embedded systems typically include cooperative tasks implemented in a single protection domain. Each task is designed with full knowledge of all other tasks and resource demands; it is difficult to change or scale embedded software. These three

approaches do not provide controllable real-time reliability or ability to make on-line tradeoffs.

Figure 1: In-Kernel Pipe with Filter Stage and Device Interface Modules



In contrast, the RTEPA mechanism supports a broad spectrum of contemporary applications ranging from virtual environments to semi-autonomous systems [Si96]. The RTEPA facility allows an application developer to construct a set of real-time kernel modules that manage an input (source) device; apply simple processing stages on the input stream (pipeline stage filters); control individual processing stage behavior through parameters obtained from a user-space application; provide performance feedback to the controlling application; and manage the output (sink) device. This basic “in-kernel” pipeline design is very similar to the *splice* mechanism [Fal94], but the EPA and scheduling control are much different. Each RTEPA module, shown in Figure 1, is implemented as a kernel thread configured and controlled through the EPA and scheduled by the DM (“Deadline Monotonic”) algorithm. The controlling application executes as a normal user thread. The RTEPA mechanism is efficient due to removal of overhead associated with protection domain crossings between device and

processing buffers, and reliable due to kernel thread scheduling (compared to split-level scheduling of user threads). The RTEPA interface provides configuration and execution flexibility on-line, with performance-oriented “reliable” execution (in terms of expected number of missed soft deadlines and missed termination deadlines).

The EPA interface is intended to allow an application to specify desired service and adjust performance for both periodic pipelines requiring isochrony and aperiodic pipeline execution. Many scenarios exist for on-line RTEPA service renegotiation for continuous media, digital control, etc. [Si96]. For example, a continuous media application might initially negotiate reliable service for a video pipeline with a frame-rate of 30 fps, and later renegotiate on-line for 15 fps so that an audio pipeline may also be executed. An application loading pipeline stages must specify the following parameters for a service epoch:

- 1) Service type common to all modules in a single pipeline; *<guaranteed, reliable, or best-effort>*
 - i) Computation time type; *<C_{worst-case} for guaranteed, C_{expected} for reliable, or none for best-effort>*
 - ii) Off-line execution samples for *C_{expected}; <{Sample-array}, [distribution-free or (normal, σ , C_{expected})]>*
- 2) Input source or device interface designation (source must exist as stage or device interface); *<source>*
- 3) Input and output block sizes; *<S_{in}, S_{out}>*

The application must also provide and can control these additional parameters on-line during a service epoch:

- 5) Desired termination and soft deadlines with confidence for *reliable*; *<D_{term} D_{soft} term-conf, soft-conf>*
- 6) Minimum and optimal time for output response (earlier responses are held by EPA); *<R_{min} R_{opt}>*

- 7) Release period (expected minimum interarrival time for aperiodics) and I/O periods; *<T, Tin, Tout>*

The approach for scheduling RTEPA thread execution is based on the EPA interface to the fixed priority DM scheduling policy and admission test called the EPA-DM approach here. The EPA-DM approach supports reliable soft deadlines given pipeline stage execution times in terms of an execution time confidence interval instead of deterministic WCET. Also noteworthy, the RTEPA facility uses two protection domains; one for user code and one for operating systems code. However, the RTEPA facility allows “untrusted” code to be executed in the kernel protection domain. We have focused on the functionality of architecture, relying on the existence of other technology such as that used in the “SPIN” operating system [Be95] to provide compile time safety checking. The negotiative control provided by RTEPA is envisioned to support isochronous and event-driven applications which can employ and control these pipelines for guaranteed or reliable execution performance.

2.0 EPA-DM Approach to Thread Scheduling

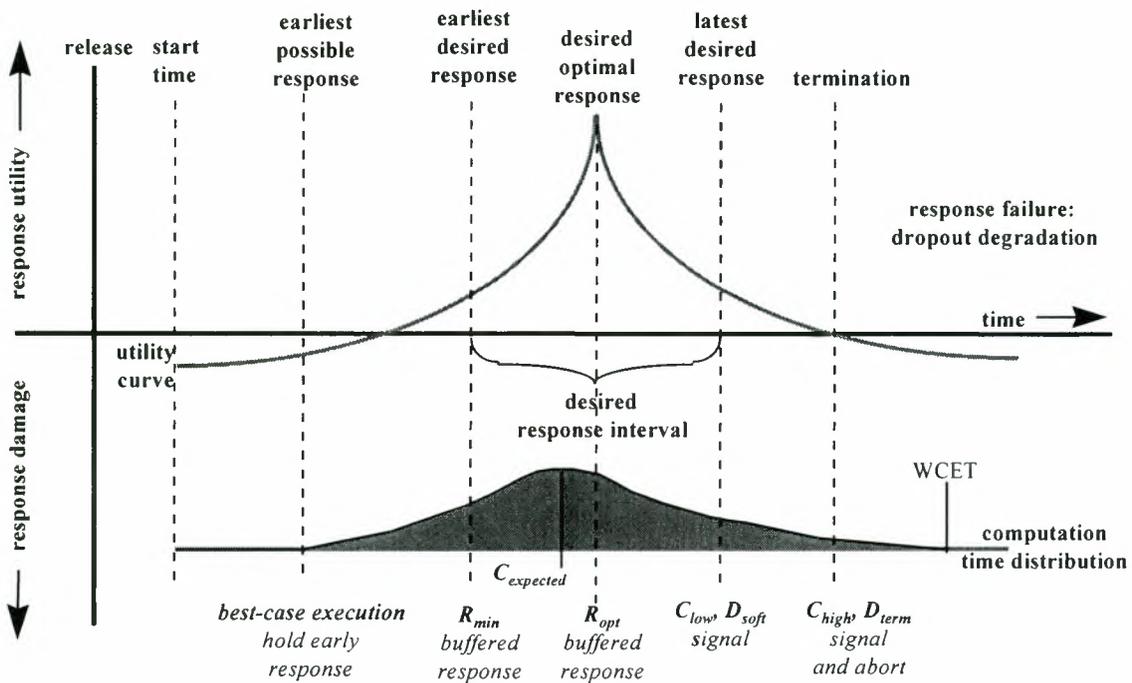
The concept of EPA-DM thread scheduling for pipeline stages is based upon a definition of soft and termination deadlines in terms of utility and potential damage to the system controlled by the application [Bu91]. The concept is best understood by examining Figure 2, which shows response time utility and damage in relation to soft and termination deadlines as well as early responses. In this design, the EPA will signal the controlling application when either deadline is missed, and specifically will abort any thread not completed by its termination deadline. Likewise, the EPA will buffer early responses for later release at R_{opt} or at R_{min} worst case. The EPA allows execution beyond the soft deadline. Signaled controlling applications can handle deadline misses according to specific performance goals, using the

EPA interface for renegotiation of service. For applications where missed termination deadline damage is catastrophic (i.e. termination deadline is a “hard deadline”), the pipeline must be configured for guaranteed service rather than reliable service.

The well established DM scheduling policy and schedulability test are used due to their ability to handle execution where deadline does not equal period [Au93]. This may often be true for the applications to be supported. One major drawback of the DM scheduling policy is that to provide a guarantee, the WCET of each pipeline stage thread must be known along with the release period.

Otherwise, for performance-oriented applications -- where occasional soft and termination deadline failures are not catastrophic, but simply result in degraded performance -- the “reliable” option with quantifiable assurance is provided, given expected execution time. Despite the ability to opt for no guarantee, this mechanism does not just provide “best effort” execution. Instead, a compromise is provided based on the concept of execution time confidence intervals and the EPA interface to the DM scheduler. An example of the EPA-DM approach is given here with a simple two-thread scenario preceded by a review of the goals for the EPA-DM approach.

Figure 2: Execution Events and Desired Response Showing Utility



The EPA-DM schedulability test eases restriction on the DM admission requirements to allow threads to be admitted with only expected execution times (in terms of an execution confidence interval), rather than requiring deterministic WCET. The expected

time is based on off-line determination of the execution time confidence interval. Knowledge of expected time can be refined on-line by the EPA each time a thread is run. By easing restriction on the WCET admission requirement, more complex processing can be incorporated, and pessimistic

WCET with conservative assumptions (e.g. cache misses and pipeline stalls) need not reduce utility of performance-oriented pipelines which can tolerate occasional missed deadlines (especially with probability of misses).

With this approach, the DM schedulability tests, which consider computation time and interference for a thread set, can still be used by the EPA as stages are loaded. Basic DM scheduling formulas are extended to return expected number of missed soft and termination deadlines to the controlling application. For this capability, when a module is loaded, the computation time must be provided with a sufficient sample set for distribution-free confidence estimates, or an assumed distribution and a smaller sample set of execution times measured off-line. From this, the computation time used in the schedulability tests is computed based upon desired confidence for meeting soft and termination deadlines. All interfering threads are pessimistically assumed to run to their termination deadline where they either will have completed or are aborted. For example, for thread i , let $C(i)$ = expected execution time; $D_{\text{soft}}(i)$ = soft deadline; $D_{\text{term}}(i)$ = termination deadline; and $T(i)$ = period; with the DM condition that $C(i) \leq D_{\text{soft}}(i) \leq D_{\text{term}}(i) \leq T(i)$. The worst-case confidence interval execution times $C(i)_{\text{low}}$ and $C(i)_{\text{high}}$ used in the extended DM schedulability tests below are based on desired confidence in execution time and probability of late response. In cases where the actual execution time is greater than the worst-case confidence interval execution time, deadlines will be missed. The expected number of missed deadlines will be less-than or equal to expected execution times outside the confidence interval resulting in response beyond a given deadline. So, if a thread has an execution time confidence of 0.999 and passes the admission test, then it is expected to miss its associated deadline 0.1% of the time or less.

For example, consider two threads that have a normal distribution of execution times (the normal distribution assumption is not required, but greatly reduces the number of off-line samples needed

compared to assuming no distribution), so that unit normal distribution quantiles $Z_{p_{\text{low}}}$ and $Z_{p_{\text{high}}}$ can be used, and assume that $WCET(i)$ is known for comparison, so that we have:

thread $i=1$: $C_{\text{expected}}(1)=40$, $\sigma(1)=15$, $N_{\text{trials}}(1)=32$,
 $Z_{p_{\text{low}}}(1)=3.29$ for soft-conf=99.9%,
 $Z_{p_{\text{high}}}(1)=3.72$ for term-conf=99.98%,
 $WCET(1)=58$, $D_{\text{soft}}(1)=50$, $D_{\text{term}}(1)=60$,
and $T(1)=250$

thread $i=2$: $C_{\text{expected}}(2)=230$, $\sigma(2)=50$, $N_{\text{trials}}(2)=32$,
 $Z_{p_{\text{low}}}(2)=1.96$ for soft-conf=95%,
 $Z_{p_{\text{high}}}(2)=3.72$ for term-conf=99.98%,
 $WCET(2)=310$, $D_{\text{soft}}(2)=400$,
 $D_{\text{term}}(2)=420$, and $T(2)=500$

If these threads can be scheduled based on the EPA inputs to the admission test, then thread one has a probability of completing execution before D_{soft} of at least 99.9% expressed $P(C_{\text{low}} < D_{\text{soft}}) \geq 0.999$. Similarly, probability $P(C_{\text{high}} < D_{\text{term}}) \geq 0.9998$. Likewise thread two has respective deadline confidences $P(C_{\text{low}} < D_{\text{soft}}) \geq 0.95$ and $P(C_{\text{high}} < D_{\text{term}}) \geq 0.9998$. Based on sufficient, but not necessary schedulability tests for DM [Au93] with EPA execution time confidence intervals inputs rather than just worst-case execution time, the schedulability with desired confidence in deadlines can be derived from the execution time confidence intervals, as shown below.

From execution time confidence intervals and sufficient (but not necessary) DM schedulability test:

eq 1: From probability theory for a normal distribution, $C_{\text{low or high}}(i) = C_{\text{expected}}(i) + Z_{p_{\text{low or high}}}(i) \left(\frac{\sigma(i)}{\sqrt{N_{\text{trials}}(i)}} \right)$

eq 2: EPA-DM admission test: $\forall i: 1 \leq i \leq n$:
 $\left(\frac{C_{\text{low or high}}(i)}{D_{\text{soft or term}}(i)} \right) + \left(\frac{I_{\text{max}}(i)}{D_{\text{soft or term}}(i)} \right) \leq 1.0$?

eq 3: $I_{\text{max}}(i) = \sum_{j=1}^{i-1} \text{ceiling} \left(\frac{D_{\text{term}}(i)}{T(j)} \right) D_{\text{term}}(j)$; where

$I_{\text{max}}(i)$ is the interference time by higher priority

threads $j=1$ to $i-1$ which preempt and run up to the “ceiling term” number of times during the period in which thread i runs.

Can thread $i=1$ be scheduled given execution time confidence and desired D_{soft} and D_{term} confidence? Yes

$$\text{using eq 1: } C_{\text{high}}(1) = 40 + Z_{p_{\text{high}}}(1) \left(\frac{15}{\sqrt{32}} \right) =$$

49.86; and likewise $C_{\text{low}}(1) = 48.72$

$$\text{using eq 2\&3: } \left(\frac{48.72}{50} \right) \leq 1.0 \text{ and } \left(\frac{49.86}{60} \right) \leq 1.0$$

for $C_{\text{low}}(1)$ and $C_{\text{high}}(1)$; likewise $\left(\frac{58}{60} \right) \leq 1.0$ for

WCET

C_{low} , C_{high} can be scheduled. (note: highest priority thread has no interference, so $I_{\text{max}}(i)=0$)

Can thread $i=2$ be scheduled given execution time confidence and desired D_{soft} and D_{term} confidence? Yes

$$\text{using eq 1: } C(2)_{\text{high}} = 230 + 3.72 \left(\frac{50}{\sqrt{32}} \right) =$$

262.88; and likewise $C(2)_{\text{low}} = 247.32$

$$\text{using eq 2\&3: } \left(\frac{C_{\text{low or high}}(2)}{D_{\text{soft or hard}}(2)} \right) + \left(\frac{I_{\text{max}}(2)}{D_{\text{soft or hard}}(2)} \right)$$

≤ 1.0 ?; $I_{\text{max}}(2) = \text{ceiling} \left(\frac{D_{\text{term}}(2)}{T(1)} \right) D_{\text{term}}(1)$

In the worst case, given the abort policy for incomplete threads reaching their termination deadline, maximum interference occurs when all higher priority threads execute until they are aborted by the EPA.

$$\text{simplifying eq 2\&3: } \left(\frac{247.32}{400} \right) + 2 \left(\frac{60}{400} \right) \leq 1.0$$

$$\text{and } \left(\frac{262.88}{420} \right) + 2 \left(\frac{60}{420} \right) \leq 1.0;$$

$$\text{simplifying eq 2\&3: } \left(\frac{310}{420} \right) + 2 \left(\frac{60}{420} \right) \leq 1.0 ? \text{ is}$$

FALSE ; WCET can not be scheduled

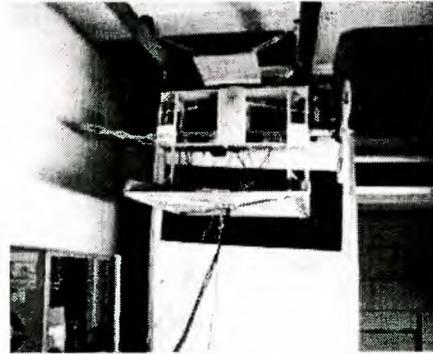
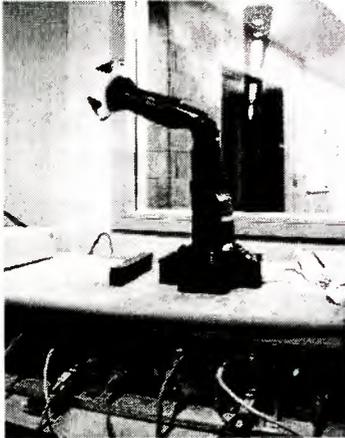
C_{low} , C_{high} can be scheduled. (note: thread 1 interferes up to its termination deadline twice in this example)

These formulas show that the two threads can be scheduled using non-WCET execution time such that desired performance is achieved. Note that the basic DM formulas show that the thread set is not considered schedulable if only WCET is considered. In this case, WCET, which is a statistical extreme, lead to rejection of a thread set which can be scheduled with $\geq 99.98\%$ probability of successfully meeting termination deadlines.

3.0 In-Work Implementation, Experimentation and Evaluation

The mechanism is being implemented VxWorks with modifications to rate monotonic scheduling of real-time kernel threads to implement the EPA-DM approach. The kernel is also being modified to incorporate the pipeline EPA system call and signal interface with functionality for loading and controlling pipe stage modules and device interface modules. The RACE (Rail-Guided Air-Powered Control Experiment) testbed (Figure 3) has been built using off-the-shelf “68HC11” microcontrollers for sensor and actuator control, with a serial interface to an Intel x86 computer for implementation of the digital control, continuous media, and “shared-control” pipelines. The RACE testbed experiments with the RTEPA mechanism on-board the RACE vehicle use a basic set of device commands (*safe*, *pitch-motors <angle>*, *thrust <left/right> <level> <duration>*, *read compass*, *read vertical range*, *read forward range*). These commands can be used in digital control pipelines to implement ramp station keeping and yaw control. The ramp ranging is provide with continuous media video-based ranging from an on-board “QuickCam” output piped to ranging estimation and control algorithm. Likewise, the digital compass output is piped to a yaw estimation and control algorithm.

Figure 3: 5 DOF Robotic Testbed (left); RACE Digital Control Testbed (right)



4.0 Related Work

A number of pipeline mechanisms for continuous media have been developed [Gov91], [Co94], [Fal94]. However, most common implementations include application-level processing with device buffers mapped from kernel space into user-space rather than an “in-kernel” mechanism for executing user code loaded into kernel space. Likewise, these memory-mapped implementations also employ user-level threads with split-level scheduling or bindings of user threads onto kernel threads. The *splice* mechanism is most relevant since it operates “in-kernel” using loadable modules or simple streaming as the RTEPA will, and was shown to have up to a 55% performance improvement [Fal94]. However, to our knowledge, *splice* does not provide a configuration and on-line control interface like the EPA.

Many examples of periodic hard real-time digital control streams exist [K194], but no general mechanism for “reliable” real-time control of pipelines is known to exist. Research on process control requirements for digital control indicate that parametric control of a number of kernel pipes within a general operating system environment would be useful for sophisticated industrial applications. Finally, many real-time semi-autonomous and “shared control” projects are in progress [Bru93]

[Fle95], including applications where occasional missed deadlines would not be catastrophic [Pa96] [Bro95].

5.0 Conclusion

Experiments will be implemented using both the RTEPA and user-level applications to compare performance. However, the RTEPA is not just expected to improve throughput compared to application-level processing, but is more significantly expected to provide reliable configuration, monitoring, and control of this type of efficient mechanism through its EPA interface to the DM scheduler. A fundamental aspect of the EPA performance control is based on the EPA-DM confidence interval approach for reliable execution. Thus, the EPA will be evaluated in terms of how well pipelines are able to meet expected and desired performance in terms of missed deadlines. Finally, experiments are being evaluated in terms of real-time parameters such as video stream dropouts, latency variation, overshoot and drift to evaluate the reliability afforded by the EPA to applications. These experiments will be run individually and simultaneously to evaluate use of the RTEPA mechanism for complex real-time applications involving multimedia and interaction between users for complex applications such as “shared” control.

6.0 References

- [Au93] Audsley, N., Burns, A., and Wellings, A., "Deadline Monotonic Scheduling Theory and Application", *Control Engineering Practice*, Vol. 1, pp 71-8, 1993.
- [Be95] Bershad, B., Fiuczynski, M., Savage, S., Becker, D., et al., "Extensibility, Safety and Performance in the SPIN Operating System", *Association for Computing Machinery, SIGOPS '95*, Colorado, December 1995.
- [Bu91] Burns, A., "Scheduling Hard Real-Time Systems: A Review", *Software Engineering Journal*, May 1991.
- [Bro95] Brooks, R., "Intelligence Without Reason", In Steels, L. and Brooks, R., eds., The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1995.
- [Bru93] Brunner, B., Hirzinger, G., Landzettel, K., and Heindl, J., "Multisensory shared autonomy and tele-sensor-programming - key issues in the space robot technology experiment ROTEX", *IROS '93 International Conference on Intelligent Robots and Systems*, Yokohama, Japan, July, 1993.
- [Co94] Coulson, G., Blair, G., and Robin, P., "Micro-kernel Support for Continuous Media in Distributed Systems", *Computer Networks and ISDN Systems*, pp. 1323-1341, Number 26, 1994.
- [Fal94] Fall, K., and Pasquale, J., "Improving Continuous-Media Playback Performance With In-Kernel Data Paths", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 100-109, Boston, MA, June 1994.
- [Fle95] Fleischer, S., Rock, S., Lee, M., "Underwater Vehicle Control from a Virtual Environment Interface", *Association for Computing Machinery, 1995 Symposium on Interactive 3D Graphics*, Monterey CA, 1995.
- [Gov91] Govindan, R., and Anderson, D., "Scheduling and IPC Mechanisms for Continuous Media", *13th ACM Symposium on Operating Systems Principles*, 1991.
- [Kl94] Klein, M., Lehoczky, J., and Rajkumar, R., "Rate-Monotonic Analysis for Real-Time Industrial Computing", *IEEE Computer*, January 1994.
- [Nu95] Nutt, G., Antell, J., Brandt, S., Gantz, C., Griff, A., Mankovich, J., "Software Support for a Virtual Planning Room", *Technical Report CU-CS-800-95*, Dept. of Computer Science, University of Colorado, Boulder, December 1995.
- [Pa96] Paulos, E., and Canny, J., "Delivering Real Reality to the World Wide Web via Telerobotics", *IEEE International Conference on Robotics and Automation*.
- [Si96] Siewert, S., "Operating System Support for Parametric Control of Isochronous and Sporadic Execution in Multiple Time Frames", *Ph.D. dissertation proposal*, Univ. of Colorado Boulder, 1996.
- [SiNu96] Siewert, S., and Nutt, G., "A Space Systems Testbed for Situated Agent Observability and Interaction", In the *Second ASCE Specialty Conf. on Robotics for Challenging Environments*, Albuquerque, New Mexico, June 1996.
- [Ste95] Steinmetz, R., and Wolf, L., "Evaluation of a CPU Scheduling Mechanism for Synchronized Multimedia Streams", in Quantitative Evaluation of Computing and Communication Systems, Beilner, H. and Bause, F. eds., *Lecture Notes in Computer Science*, No. 977, Springer-Verlag, Berlin, 1995.
- [To90] Tokuda, H., Nakajima, T., and Rao P., "Real-Time Mach: Towards a Predictable Real-Time System", *Proceedings of USENIX Mach Workshop*, October 1990.

**Space Robotics Applications on the
International Space station (2)**

EUROPA (External Use of Robotics for Payloads Automation)

R. Mugnuolo, F. Bracciaferri (Agenzia Spaziale Italiana, Italy)

F. Didot (ESA - Automation and Robotics Section, The Netherlands)

G. Colombina, E. Pozzi (Tecnospazio, Italy)

To demonstrate the potentiality of in orbit use of robotics for external applications, ESA and ASI decided to fly a technology demonstration mission, named JERICO (Joint European Robotic Interactive and Calibrated Operations).

Different mission scenarios were considered: a cooperation with NASDA, a mission on the Space Shuttle, on MIR and on Russian segment of ISS. Unfortunately, none of them went on. In the mean time, it has grown up the general opinion that robotics is a valid tool for in orbit operations and therefore a mere demonstration mission can be considered as no longer necessary.

As consequence, ASI is now proposing to NASA a payload called EUROPA (External Use of Robotics for Payloads Automation) to be flown on ISS late 2002. The preferred ASI approach is to keep the same JERICO elements and the involvement of the same partners. This possibility is now under discussion and will be finalised in the next future.

1. EUROPA OBJECTIVES

The EUROPA payload is intended to perform, as first objective, a realistic end-to-end robotic technology demonstration to show the advantages and the feasibility of a versatile robotically tended exposed payload infrastructure.

Following this, the infrastructure shall allow exposure payloads or payload units to be installed, pointed, serviced/manipulated, inspected, analysed and retrieved in a flexible way without the need for human EVA.

This would constitute a unique service to the world-wide user community of relatively low-

cost, rapid and reliable logistics support for scientific experimentation on the ISS.

The payload is built around the SPIDER medium-sized dexterous robot arm.

It can perform the following tasks:

- installation/removal of small payload containers on exposure attachment ports;
- handling of payload units (experiment samples or sample cassettes) for the purpose of scientific/technological investigations;
- close-up visual inspection of payload units by means of a camera.

All of the above tasks can be pre-programmed and checked on ground and then performed automatically on orbit with ground monitoring and possibility to intervene and correct the situation in case anomalies are detected.

2. EUROPA DESCRIPTION

2.1. EUROPA flight segment

The flight segment of EUROPA is the part aimed to the execution of the robotic capability demonstration and the payloads handling.

It will be accommodated on one of the six Adapters of the EXPRESS Pallet (ExP) (fig. 2.1-1).

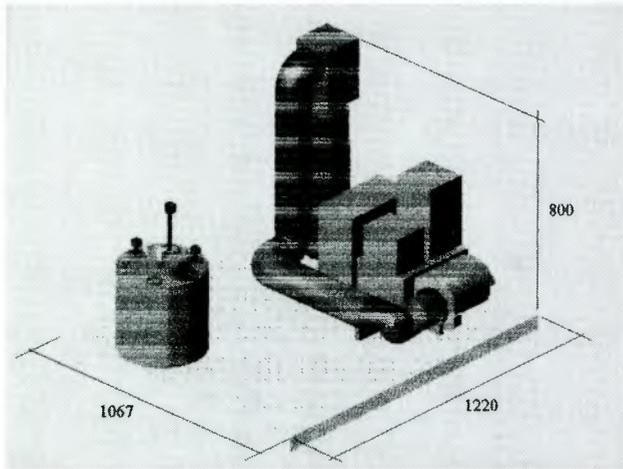


Fig. 2.1-1 : EUROPA flight configuration.

The EUROPA consists of the following subsystems:

- manipulator composed of:
 - hold down - to fix the arm during launch and re-entry phases;
 - arm - to perform, when integrated with its avionics, all the required activities (such as open drawer, close door, install peg, etc. ...);
 - Force/Torque (F/T)- to allow force and torque control during the execution of operations;
 - End Effector (EE) - to grasp the items to be handled;
 - RObot CALibration Tool (ROCAT) - to calibrate the system;
- avionics composed of:
 - controller - to send the operating sequence in a pre-programmed automatic way, to receive the resolver signals and to provide the interpretation and execution of commanding and the collection and transmission of engineering telemetry data. For local commanding, a man machine interface is provided running on one or more crew computers;
 - driver - to provide the signals to the motors;

- Power Distribution Unit (PDU) - to distribute power from EXPRESS Pallet Adapter (ExPA) power bus and EUROPA items;
- harness - to electrical connect all EUROPA items;
- emergency unit - to allow arm stowing in emergency conditions;
- supervisor camera - to provide an overview of the overall scene while robot is operating;
- taskboard - to provide all the required in-orbit infrastructure to demonstrate the technology and evaluate/measure the performance capabilities of the arm and its avionics (such as compliant motion capabilities, accuracy, repeatability).

Fig. 2.1-2 shows the EUROPA breakdown.

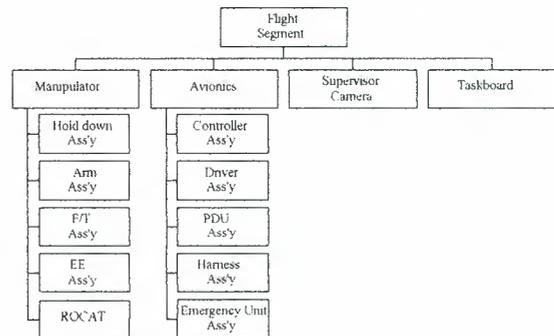


Fig. 2.1-2.: EUROPA Flight Segment breakdown.

2.2. EUROPA Ground segment

The ground segment of EUROPA is the part aimed to:

- ground calibration;
- program preparation and verification on the Ground Reference Model (GRM);
- flight robot monitoring and command;
- on-ground data handling.

The EUROPA ground segment consists of the following subsystems:

- Ground Support checkout Equipment (GSE) - to provide the necessary equipment to support the complex

integration test and the acceptance testing before launch;

- engineering support equipment - to provide all the necessary HW and SW to execute support activity;
- preparation and verification work station - to provide the necessary HW and SW environment to prepare and verify all the activities of EUROPA and the Payloads;
- GRM - to provide a ground replica of the flight segment;
- Robot Monitoring and Command work Station (RMCS) - to provide the necessary HW and SW environment to execute and monitor a complete EUROPA activity plan. The monitoring will be achieved both using graphic simulation based on the flight segment telemetry data and possibly compressed image(s). The RMCS will be connected to the GRM in order to have a real emulation of flight segment operations;
- Payload Monitoring and Command work Station (PMCS) - to provide the necessary HW and SW environment for a scientist, in a user home base, to plan, execute and monitor pre-defined activities on his payload. The monitoring will be done both using high level graphic simulation based on the telemetry data and possibly compressed image(s).

Fig. 2.2-1 shows the EUROPA ground segment breakdown.

All the ground segment items will be located at ASI-Matera.

The GSE will, on the contrary, follow the Flight Segment helping during integration phase.



Fig. 2.2-1.: EUROPA Ground Segment breakdown.

2.3. EUROPA Operations

Two payloads will be used for the demonstrations::

- the taskboard assembly, with representative elements to systematically exercise the typical expected payload tending tasks (open/close sample drawers, extract/insert sample containers, point sample containers, ..) with well defined degree of difficulties;
- one dummy payload, to make the demonstration of operations on a 'real' payload.

3. EUROPA KEY ELEMENTS

The two key elements around which EUROPA payload is built are:

- the SPIDER robotic arm;
- the CESAR robot controller.

3.1. SPIDER Robotic Arm

The Arm Ass'y is the one developed in the frame of SPIDER project an ASI contract (see fig. 3.1-1) with the following modifications:

- mounting of ROCAT system between the wrist and the Force/Torque,
- addition of external end strokes (to apply only if needed) for joint 1, 2 and 7;
- addition of heaters;
- change of the external connectors and fixation on a dedicated flange (shoulder side).

The **Arm manipulator** is anthropomorphic and features 7 rotational degrees of freedom (d.o.f.).

The Arm kinematic structure of the arm is schematically represented in fig. 3.1-2.

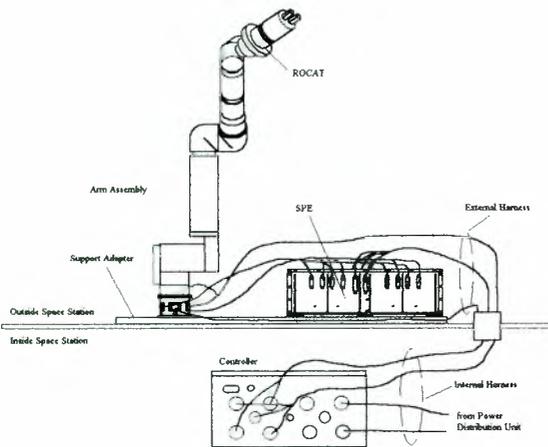


Fig. 3.1-1: SPIDER Arm.

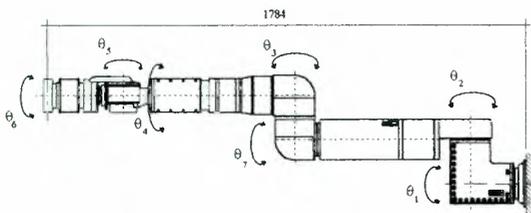


Fig. 3.1-2: Arm kinematic structure.

Each joint (numbered from 1 to 7) corresponds to a degree of freedom, in sequential order from shoulder to wrist, apart from joint 7 which is located between joint 2 and 3.

The rotation angle of each joint is shown in table 3.1-1.

Joint 1	Joint 2	Joint 7	Joint 3	Joint 4	Joint 5	Joint 6
$\pm 180^\circ$	$\pm 120^\circ$	$\pm 180^\circ$				

Table 3.1-1: Maximum joints stroke.

Joints 1 and 2 are in the shoulder assembly, joints 3 and 7 are in the elbow assembly and joints 4, 5 and 6 are in the wrist assembly.

Each joint of the arm is powered by an electromechanical actuation group composed of motor, gearbox, input and output shaft sensors and brake.

Table 3.1-2 summarises SPIDER arm key characteristics at 0g conditions, valid in all points of the operational envelope at the environmental conditions shown in table 5.1-3. The Arm can also be operated at 1-g conditions, without any support equipment.

Parameters	Characteristics
Max load carrying (COG at 500 mm from Arm end flange)	up to 250 kg
Continuous actuation force - isotropic	25 N (100 N short period < 10 s)
Continuous actuation torque:	
• shoulder joints	40 Nm (200 Nm short period < 10 s)
• elbow joints	30 Nm (100 Nm short period < 10 s)
• wrist joints	30 Nm (50 Nm short period < 10 s)
Position repeatability	1 mm
Orientation repeatability	0.05 °
Position accuracy (before cal.)	3 mm
Orientation accuracy	0.1 °
Max linear speed	0.1 m/s
Max rotational speed	0.1 rad/s
Mass	65 kg
Stowing volume	510x1040x310 mm
Power consumption (estimated)	30 W to 45 W (90 W at 1g)

Table 3.1-2: SPIDER arm characteristics in 0g condition.

Thermal	<ul style="list-style-type: none"> survival operational 	-60 °C to +150 °C -40 °C to +80 °C
Pressure in orbit	<ul style="list-style-type: none"> survival-operational 	10 ⁻⁴ to 10 ⁻⁶ Pa
Launch vibration	<ul style="list-style-type: none"> first frequency sinusoidal random 	> 35 Hz Max level 6 g; sweep rate 3 oct/min up and down Max load 13.8 gr.m.s; time duration 150 s

Table 3.1-3: SPIDER arm and End Effector environmental requirements.

A **Force/Torque Sensor**, mounted on the wrist end, has been developed to measure and control the force and the torque exerted by the arm during operation.

Its main characteristics are:

F/T Characteristics	
Sensor type	strain gauges
Measuring range	± 200 N, ± 20 Nm
Measuring accuracy	$\pm 3\%$ (temperature range -40 °C +80 °C)
Measuring resolution	0.1 N, 0.01 Nm
Overload protection	± 2000 N, ± 200 Nm
Mass	1.15 kg
Power consumption	< 2 W

The F/TS is shown in fig. 3.1-4.

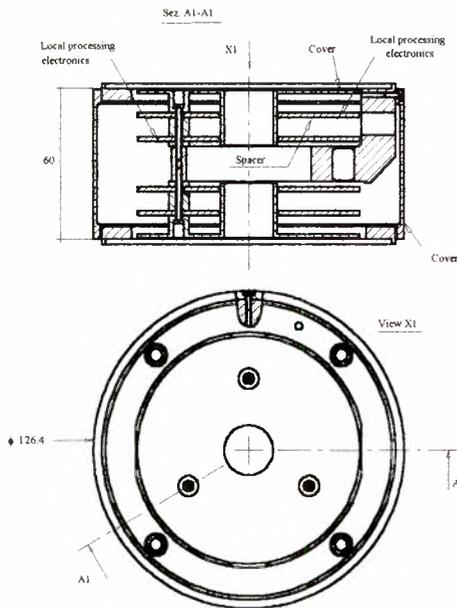


Fig. 3.1-4: Force/Torque Sensor drawing.

The main characteristics of the **End Effector** (see fig. 3.1-5) are shown in the following table.

<i>Parameters</i>	<i>Characteristics</i>
Max opening width	76 mm
Min opening width	0 mm
Max gripping force	300 N
Max dimension of grasping object	70 mm
Mass	3.995 kg
Power consumption	6 W

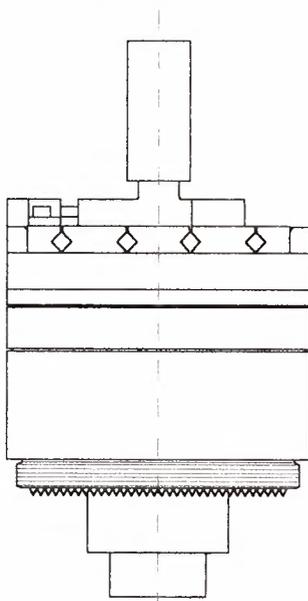


Fig. 3.1-5: End Effector schematic drawing.

The End Effector is equipped with two Tactile Sensors (located on the End Effector jaws) in order to monitor the force exerted during gripping.

The characteristics of the Tactile Sensors are:

<i>Parameters</i>	<i>Tactile sensors Characteristics</i>
Sensor type	strain gauges
Max force	200 N
Accuracy	$\pm 3\%$ (temperature range $-40\text{ }^{\circ}\text{C}$ $+80\text{ }^{\circ}\text{C}$)
Resolution	0.1 N

On the End Effector a latching/delatching mechanism has been implemented to easily disconnect the End Effector from SPIDER arm.

4. THE CESAR ROBOT CONTROLLER

Considering that the development of a mature robot controller is a multi-million, multi-year effort, ESA has embarked on the development of a common Controller for European Space Automation and Robotics (CESAR) in hardware and software by starting from a mature, well-proven industrial product: the COMAU C3G controller.

The architecture of CESAR (Fig. 4-1) is composed of a Robot Control Unit (RCU), which performs the most computation intensive high level tasks, and a set of more or less intelligent slave modules, named Servo Control Units (SCU), which control the robotic hardware (servo drives, sensors). Due to this strict master-slave structure CESAR does not need any sophisticated multi-processor bus (such as VME). Instead a multi-drop master-slave serial bus is adopted to allow for communication between RBU and SCUs. This serial bus enables both the concentrated and distributed control.

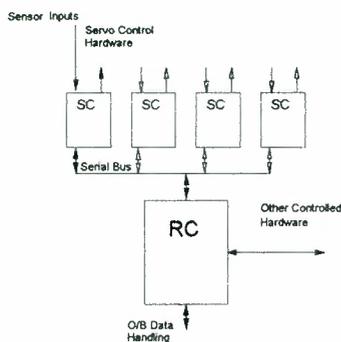


Fig. 4-1: CESAR general architecture.

The software architecture (fig. 4-2) features three types of tasks:

- system tasks (implementing the interface to Telemetry/Telecommands, a monitor shell and some built-in test logic)
- robotic tasks (robot program interpretation, motion control)
- user tasks (to interface to external auxiliary hardware)

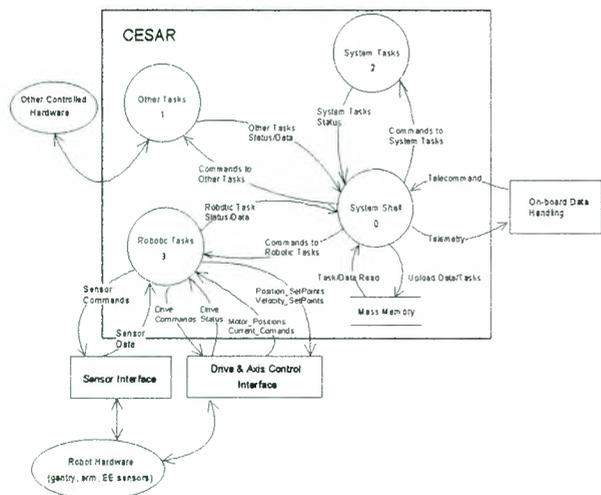


Fig. 4-2: CESAR software architecture

The RCU software architecture allows for the easy replacement or addition of tasks to modify/augment the CESAR functionalities.

The real-time operating system chosen for the CESAR SW is vxWorks, which supports many microprocessors, including the newly available radiation tolerant microprocessor

ERC32 and Digital Signal Processors (TSC21020E) developed on behalf of ESA by European industry.

The software modularity and the wide micro-processor support for the operating System, enable the adoption of HW architectures even different from the CESAR general one.

Hardware Implementation

The CESAR-HW uses two different types of electronic boards to implement the RCU and the SCUs. The RCU uses a Standard payload Computer (SPLC) CPU module fitted with a mezzanine SPLC LAN Adapter and a MIL-Bus Adapter, while the SCUs are developed specifically for the CESAR-HW.

The SPLC uses a common mezzanine bus for their mezzanine slots (MIL-Bus, LAN). The same bus concept is also used for the SCU Boards.

A common mezzanine bus concept over the whole CESAR-HW reduces costs since all the elements can make use of existing SPLC Adapters.

The SCUs are designed in a modular manner. Each SCU consists of a Base-Board, a Core-Board and one or two Mezzanine-Boards.

A Base Board carries Core- and Mezzanine-Boards and provides the required interfaces to the robot servo amplifiers.

Core-Boards include a DSP CPU, drivers, non-volatile memory and program/data RAM.

Mezzanine-Boards are used to interface to the serial bus. They feature a micro-controller, which performs data-communication tasks up to the Application Layer of the ISO/OSI model. With this arrangement the Core-Board is not affected by the specifics of the serial bus used.

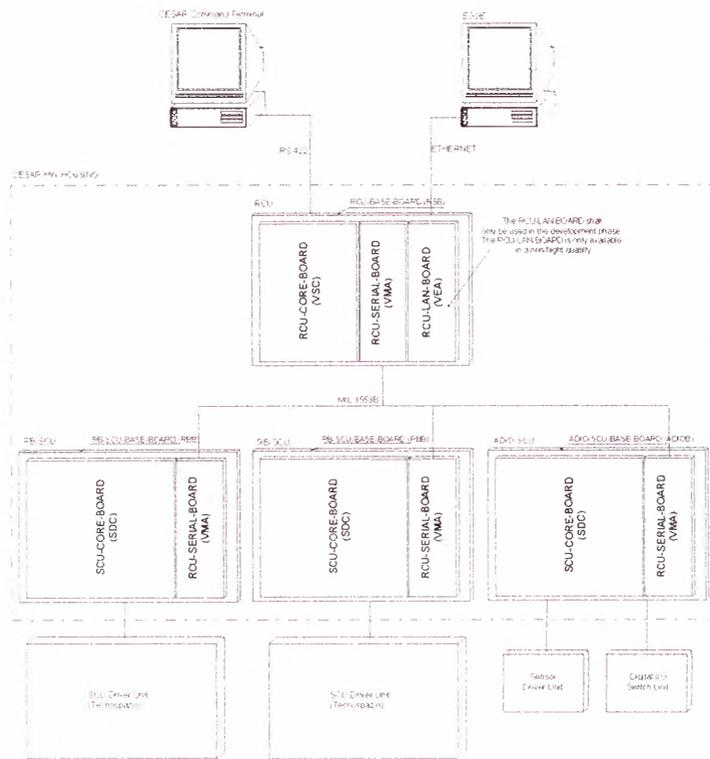


Fig. 4-3: A CESAR hardware implementation.

The RCU and the Command terminal communicate via a RS422 interface with a baud rate of up to 19200 baud. The communication between the RCU and the SCU's will be realised via the MIL-Bus 1553B.

For development and debugging of the system software an Ethernet interface is provided. This interface is implemented through a S PLC Ethernet Mezzanine card. **SW Tests**

For what regards the software, the porting work has been finished. CESAR-SW now runs on a ERC32-compatible platform using the VxWorks real-time operating system.

An extensive test campaign was performed to validate the software using:

- a standard test suite (normally used in industrial robot applications)
- a series of space related tests involving normal and contingency

cases (e.g. loss of communication with remote interface).

For the purpose of demonstrating the software, a series of graphical man machine interfaces (MMI) has been produced. These allow to command and monitor robot operations through a CESAR command terminal. Figure 4-5 shows one of these MMIs, which mimics a teach-pendant appearance.

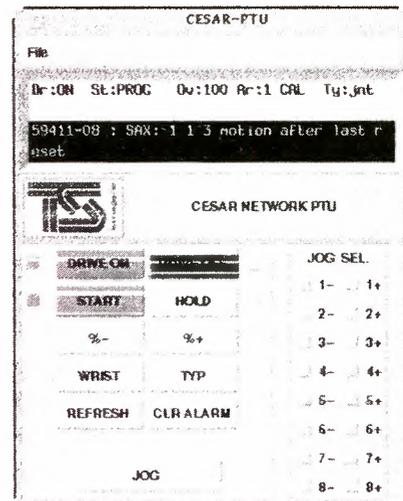


Figure 4-4 Teach Pendant MMI.

5. CONCLUSIONS

The payload is aimed to perform a realistic end-to-end robotic technology demonstration to show the advantages and the feasibility of a versatile robotically tended exposure payload infrastructure. This infrastructure shall allow exposure payloads or payload units to be installed, pointed, serviced/manipulated, inspected, analyses and retrieved in a flexible way without need of human EVA.

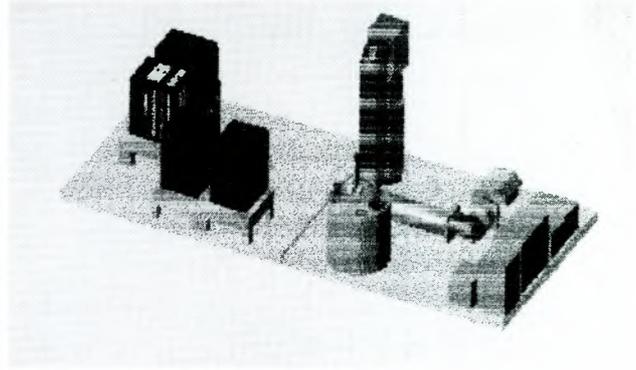
It could be seen as a unique service to the world-wide user community of relatively low-cost, rapid and reliable logistics support for scientific experimentation on the ISS, able to perform all the manipulation activities for payload operations.

The payload is built around a medium-size dexterous robot arm. It can perform the following tasks:

- installation/removal of small payload containers on exposure attachment ports;
- handling of payload units (experiment samples or sample cassettes) for the purpose of scientific/technological investigations;
- close-up visual inspection of payload units by means of a camera.

All of the above tasks can be pre-programmed and performed automatically, with ground monitoring and the possibility to interfere and correct in the case of anomalies ("interactive autonomy").

The first possibility to show the capability of this Robotic Adapter in operating real payloads is to place it close to the European Technology Exposure Facility (EuTEF), in order to provide the manipulation services to this facility.



6. REFERENCES

- [1] P. Putz, "SPARCO: an Advanced Space Robot Controller", Preparing for the Future Vol. 5 No 4, Dec. 1995

Development of the European Technology Exposure Facility

Giuseppe Borghi
Carlo Gavazzi Space
Milano, Italy
gborghi@cgspace.it

Jan Dettmann
ESA-ESTEC
Noordwijk, The Netherlands
jdettman@estec.esa.nl

Gianfranco Visentin
ESA-ESTEC
Noordwijk, The Netherlands
gvisenti@estec.esa.nl

ABSTRACT - The European Technology Exposure Facility (EuTEF) is a novel multi-user facility for technology research and assessment in Low-Earth Orbit. It will be developed by the Carlo Gavazzi Space under contract with ESA with strong contributions from the Italian and German national space agencies in the area of the robotics sub-system.

EuTEF will be launched in late 2002 and installed externally on an Express Pallet of the ISS by exploiting the robotic systems of the ISS (MSS). After installation it shall provide a total of three years of in-orbit experimentation time.

One of the innovative features of EuTEF is the presence of a small, dexterous robot arm with pre-programmed automatic operation. The selected operation mode for the experiment execution is called Interactive Autonomy being a mid point between full automation and teloperation. By Interactive Autonomy a set of complex sequence can be safely executed by the robotic system with the minimum involvement of the operator at the ground station.

The robotic arm acts, via an end effector, on a set of standard boxes called Payload Modules (PMs), containing the experiments. The end effector provides both mechanical and electrical interfaces to the PMs. The PMs have standard dimensions and are characterized by standard mechanical and electrical interfaces. The PMs can be relocated and stacked allowing to place them in the most beneficial locations in orbit, according to an experimentation schedule. The continuity of the electrical connection (power and data) to the PMs during the relocation constitutes a chief features of EuTEF system, allowing the experiment to be prosecuted during relocation. A PM contains trays filled by specimens to be exposed to the space environment. The robotic arm can open and close experiment trays in the PMs in order to provide pre-programmed and controlled exposure profiles. The PMs can be visually inspected via a stereo camera and a lighting unit mounted on the end effector. It can provide visual inspection of the contents of experiment trays and PMs. A PM or a tray can be pointed into desired directions for prescribed periods

compensating the ISS orbiting, for example for sustained solar exposure.

EuTEF is endowed with an Environment Monitoring Station providing centralized source of environmental data such as, e.g., radiation, pressure, contamination, and oxygen flux. These data can be used to correlate experiment results with the in orbit environment condition. A Material Property Laboratory is installed on EuTEF and it composed by a spectroreflectometer and a microscope for superficial inspection of the specimens. The robotic arm can bring a drawer to the analysis instruments for in-situ materials property investigations.

The arm enables the logistics re-supply by exchanging "old" and "new" Payload Modules in the case of a new upload. To investigators from many space technology domains, the EuTEF provides the advantages of low cost access to space exposure, short experiment lead time, high operational flexibility, rapid (quasi-online) availability of experiment results, and confidentiality of the contents and results of the investigation.

The EuTEF development started in the early 1999. The main areas of development accomplished in the first phase of the program are the design of a suitable end effector endowed with a stereo camera, lighting unit and on-board electronics, the design of the robotic arm based on the previous ASI SPIDER program results, the definition of the ground segment and flight segment S/W-H/W architecture. The present paper describes the current development status corresponding to the assessed review of the system requirements.

1 Introduction

As part of a bartering agreement between ESA and NASA, ESA negotiated access to 3 Express Pallet Adapters (ExPA) locations during the early utilisation phase of the International Space Station (ISS), for a period of three years.

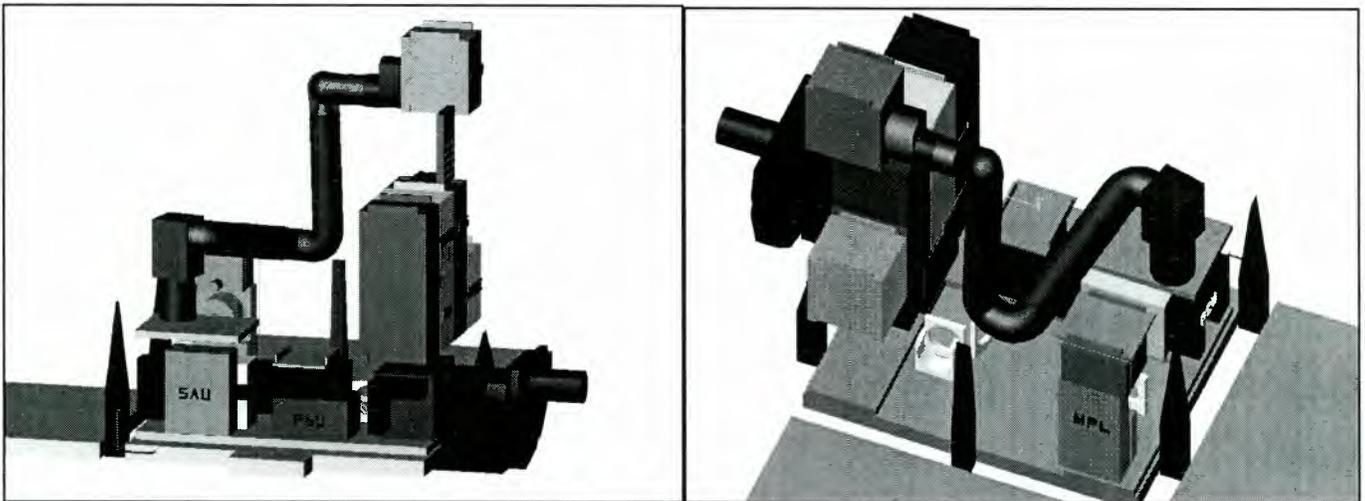


Figure 1 EuTEF in orbit operation (Some structural part has been blanked for clarity)

One of the ExPA has been allocated to the EuTEF mission that will be installed with the Utilisation Flights number 4 (UF4) on ISS external sites on the external structure of the ISS, the so called Express Pallet (ExP) at the S3 truss attached site onto the outboard zenith pointing ExP (see Figure 2).

Carlo Gavazzi Space (CGS) was awarded by ESA the role of integrating the European Technology Exposure Facility and as such CGS is in charge of defining the EuTEF facility in all its aspects. The activity comprises the definition of the mission phases, of the programmatic and technical interfaces with the experimenters, the design, developing, manufacturing of the EuTEF facility, the analytical and physical integration of the payloads, up to achieving safety certification of the integrated EuTEF in NASA and the on-orbit commissioning.

The EuTEF project is currently carrying out the Phase B activity aimed to achieve the design specification at system and sub-systems level ready for the subsequent manufacturing activity (Phase C/D). The System Requirement Review is planned for middle of June and the Preliminary Design Review will be on October 7th, 1999. The Phase C/D will start presumably in November and will last two years ending with the EuTEF delivery at NASA KSC and on-orbit installation with the UF4 launch.

Three key aspects characterizes EuTEF driving its design:

- the presence of large set of Investigations to be carried out in a 3 year mission duration in external space environment;
- the presence of a small dexterous robotic arm allowing to change on-orbit facility configuration;
- the requirements to provide the Experimenters with a quasi on-line access to their scientific data.

The EuTEF activity will be performed in an external space environment with limited control from ground

segment implying a high degree of autonomy of the facility. For example, the robotics operations are based on a set of primitive function that can be executed autonomously by the robotic subsystem allowing the relocation of payloads, the tracking of external target. This concept has been called Interactive Autonomy and is described in Section 1.5.5.

The EuTEF thermal environment allows only limited power dissipation capability and impose to EuTEF a wide range of temperature leading to a challenging mechanical and thermal design. This has also strong impact on the payloads operations that have to be scheduled based on these constraints. A novel thermal design approach is under study in order to cope with a multi-configuration facility.

The Ground Segment in conjunction with the functionality provided by the Flight Segment will allow the Investigator to have quasi on-online access to their scientific data. In obtaining that CGS is exploiting the ultimate software technology in term of networked system and of space system autonomy able to react to asynchronous request coming from the users.

In the following Sections the EuTEF system is described pointing out the key aspect of current development.

1.2 Mission Phases

During its lifetime EuTEF is operated according to the following mission phases:

Integration and Test Phase - The integration and test phase encompasses all processing activities from equipment assembly up to acceptance testing of EuTEF.

Launch Phase - The launch phase begins at the instant of NSTS lift-off and ends when the NSTS is docked to the ISS

STS docked Phase - This phase starts when the NSTS is docked to the ISS and ends when EuTEF or ExP is handed over by the SSRMS robotic arm.

Transfer NSTS /ISS - This phase starts when EuTEF or ExP is handed over by the SSRMS and ends when

EuTEF or ExP is installed at the S3 attached site. During this phase the power to EuTEF is not continuously available.

In Orbit Commissioning Phase - This phase starts when EuTEF is installed at the S3 attached site and ends after initialization and in orbit testing. It will be dedicated to initiate Instruments and support equipment, to perform initial check-out and to tests the different operational modes.

Operational Phase - This phase starts at the end of the in orbit commissioning phase and covers the time when EuTEF is in operational or stand by mode under control of the ground segment.

Return Flight - This phase starts when the NSTS is de-docked from the ISS and ends with the NSTS landing.

Typical phase duration are given in Table 1

1.3 EuTEF Overall Block Diagram

The System General Block diagram is shown in Figure 2 where the following major blocks are identified.

- EuTEF Flight Segment
- Express Pallet System (ExPS)
- International Space Station (ISS) Flight Segment
- ISS Ground Segment

PHASE	DURATION
Storage	up to 1 year
Transportation/Launch Site Check-out	4 months
Launch & Flights Phase	48 hours
STS docked Phase	Up to 11 days
Transfer NSTS/ISS	Several hours
In orbit Commissioning Phase	2 to 3 weeks
Operational Phase	18 months to 36 months
Reboost Phase	3 days , every 90 days
Transfer ISS/NSTS	Several hours
STS Docked Phase	Several days
Return Flight	6 to 12 hours

Table 1 Typical Mission Phases Duration

□ EuTEF Ground Segment

ISS Flight and Ground Segments architecture is hidden in the block diagram, being these details outside the scope of this preliminary description, concentrated on EuTEF interface aspects.

The purpose of the diagram is in fact to provide an overview of the EuTEF Context with special regard to Data and Control Flow aspects.

The **EuTEF Flight Segment** interfaces the **ISS Flight Segment** via the **Express Pallet System**. EuTEF is accommodated on a platform called Express Pallet Adapter (ExPA) providing the mechanical and electrical I/Fs (See Figure 4).

The major electrical interfaces provided via the Express Pallet Adapter (ExPA) are:

Power Interface, via which the electrical power is provided to the EuTEF Flight Segment for its operation. ExPA interface makes available to EuTEF 120 V and 28 V power outlets.

EuTEF provides ISS with CCSDS Telemetry (TM) Packets via the MIL STD 1553B Interface. These Packets are then downlinked by ISS to Ground.

- Analog Input Lines to ExPA, used to monitor directly EuTEF analog parameters, like temperature
- Digital Input / Output Lines, used to control EuTEF via discrete lines and to read its status

Data Handling Interface, via which Data and Commands are exchanged between EuTEF and ISS. Data Interface is implemented via a dual redundant MIL STD 1553B Interface via which are exchanged CCSDS Telemetry / Telecommand (TM/TC) Packets. Additional Data Handling interfaces include an Ethernet link, used for the transmission to Ground of Scientific Data.

The ISS Flight Segment downlinks Telemetry data to its Ground Segment while the ISS Ground Segment uplinks Telecommands to its Flight Segment.

The **ISS Ground Segment** provides EuTEF with the following major services in a transparent way:

- delivers EuTEF downlinked TM Data to EuTEF Ground Segment
- receives from EuTEF Ground Segment requests of Telecommands issue to its Flight Segment and takes care to uplink the requested Telecommands to ISS Flight Segment for eventual delivery to EuTEF

The EuTEF Ground segment is described in the following Section.

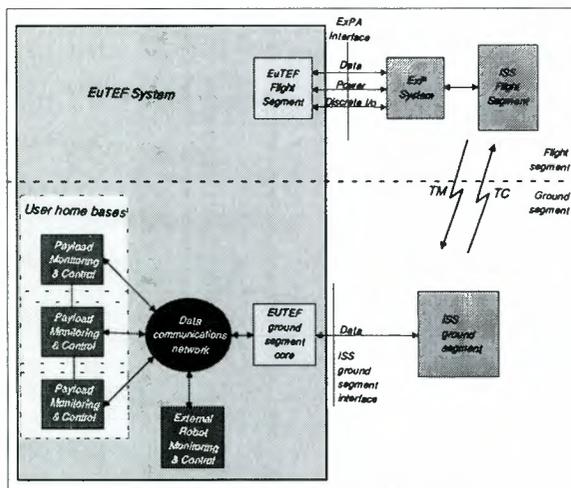
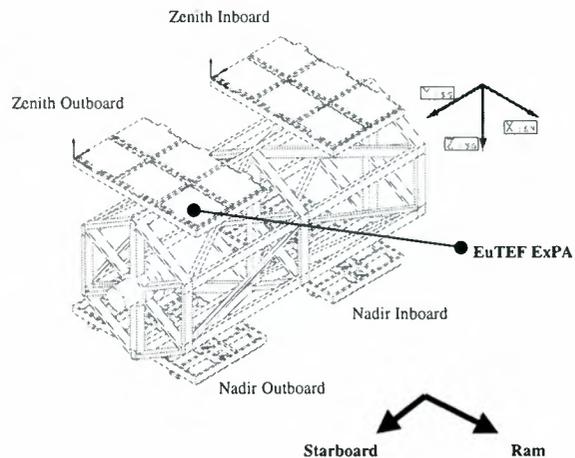


Figure 2 EuTEF System level block diagram and the ExP system



1.4 EuTEF Ground Segment

The **EuTEF Ground Segment** is a distributed system consisting of the following major blocks:

- A centralized block conventionally called as the **EuTEF Ground Segment CORE**, which actually interfaces the ISS Ground Segment. This block implements also centralized functions like EuTEF Monitoring and Control, Data Archiving, Data Presentation, Operations Preparation and Validation. The Monitoring and Control Tasks include also the ones related to Robot Operations. A further task of the Core Block is to handle the interfaces with the other blocks of the EuTEF Ground Segment distributed architecture, like Payload Monitoring / Control and External Robot Monitoring / Control.
- The **Payload Monitoring and Control** blocks, located at the Investigators User Home Bases. The tasks allocated to these blocks include:
 - Specification of the Experiments to be run on the Payloads during the EuTEF experimental sessions. The Experiments Specifications prepared by the Investigators are then subjected to a centralized Scheduling Process (running in CORE block, see Section 1.4.1) for a compatibility check with the requirements of the other Investigators and with the EuTEF and ISS constraints.
 - Monitoring of the Payload TM Data generated on board and actually delivered to the Payload Monitoring & Control block from the EuTEF Ground Segment Core block.
 - Issue of Control Command to the Payload. These commands are actually forwarded to the Core Block that takes care of verifying their compatibility with the experimental

session in progress and with Facility status and constraints.

- The **External Robot Monitoring and Control** block. This block provides the same Robot Monitoring and Control functions built-in in the Core block but located remotely.

The EuTEF Ground Segment provides I/Fs to these external S/Ss allowing remote connection to the Ground Segment Core. A Data Communications Network linking the Core block to the distributed EuTEF Ground Segment elements like the External Robot Monitoring and Control Block and the Payload Monitoring and Control Blocks located at the User Home Bases based on a TCP/IP connection.

1.4.1 Investigation timeline definition

In this Section a brief description of the process aimed to obtain the facility operation timeline is provided. The necessity to schedule the activity of EuTEF comes from the limited resource in term of power, data rate, thermal dissipation, etc.. available at system level. These resources are below of the summation of the Experimenters needs preventing the possibility to have simultaneous operations of all the payloads.

Each Experimenter submits its specific requests to EuTEF in terms of:

- investigation operation program to be executed;
- resource usage;
- execution time line;
- investigation specific information.

We call this specification *EuTEF Investigation Specifications* (EIS). The EIS are coded using a specific high level language called EISL (EuTEF Investigation Specification Language). The set of EISs has to be merged by the EuTEF operation control center in a unique sequence of EuTEF operations to be uploaded on the Flight Segment and executed (see Figure 3).

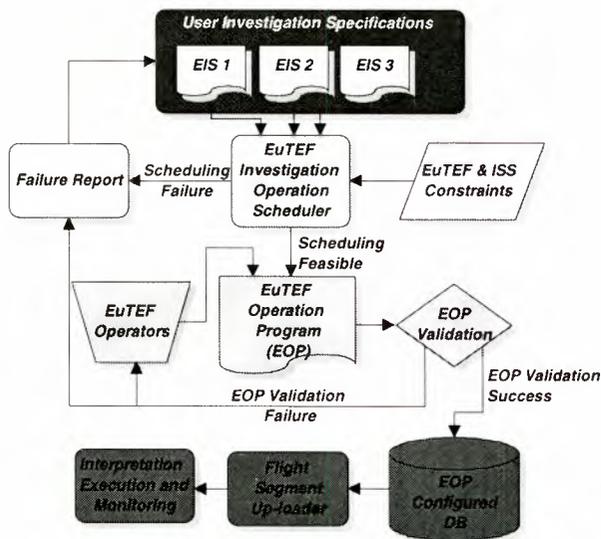


Figure 3 EuTEF Operation Program Generation - The data flow diagram shows the validation loop necessary to obtain an EOP

The scheduling activity define the time-line for every EuTEF operation taking into account the following:

- ❑ the investigation specifications (EISs);
- ❑ the cost (time and resource usage) of every EuTEF elementary operation;
- ❑ the EuTEF system constraints;
- ❑ the ISS constraints.

This computer aided process allows to optimize the usage of resources that are critical for EuTEF activity, such as, e.g., power, data link, thermal dissipation and to synchronize EuTEF with ISS modes and ISS time windows for EuTEF operations. We take advantage, during the EuTEF Operations Scheduling Process of:

- ❑ time dependent resource usage;
- ❑ the possibility to set experiment in survival mode;
- ❑ the possibility to delay investigation start time;

to find a EuTEF Operation Program that minimize the execution time and that does not violate the system constraints.

1.5 EuTEF Flight Segment

EuTEF will be accommodated in zenith outboard pallet and from this position will have Ram, Starboard and Zenith open field view (see Figure 1 and Figure 2). On Ram direction side there will be some ExP structures (a Scuff-Plate and a Robot-Guide) that can partially shade objects located at a height less then about 300mm. EuTEF envelop (available volume) on ExPA measures 1168 x 863 x 1244 mm (See Figure 4)

In Figure 1 is depicted the operational configuration of EuTEF. The following major EuTEF functional blocks are identified:

- ❑ Facility Infrastructure including:

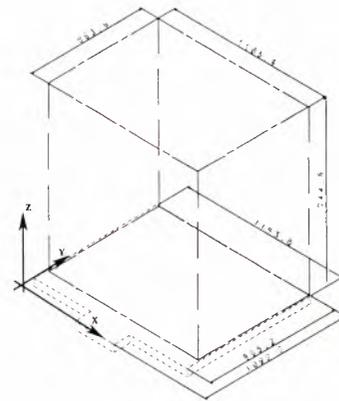


Figure 4 The standard ExPA with the EuTEF available envelope

- ❑ EuTEF Support Structure
- ❑ EuTEF Avionics
- ❑ Environment Monitoring Station (EMS)
- ❑ Materials Properties Laboratory (MPL)
- ❑ Robotic Subsystem
- ❑ Payload Modules Set including:
 - ❑ Payload Modules
 - ❑ Trays

The **EuTEF Support Structure**, mounted on the Express Pallet Adapter Plate, provides the physical support onto which all the EuTEF parts are integrated. In Figure 1 it has been blanked allowing to see the EuTEF avionics.

The **EuTEF Avionics** (see Figure 5) is in charge of providing EuTEF elements with all the required services in terms of Electrical Power supply and of centralized Supervision, Control and Monitoring.

A further task of Avionics is to handle the electrical EuTEF interfaces to ExPA i.e.:

- ❑ Power Interface, to be further conditioned and distributed to the EuTEF elements
- ❑ Data Interface, via which EuTEF delivers to ISS TM / Video data and receives Commands and Ancillary Data.

Among the Supervision and Control tasks it is worth to mention the execution of the EuTEF Operations Programs. The Avionics is also in charge of performing the following further major functions:

- ❑ ISS Commands Reception, Interpretation and Delivery to the actual EuTEF element in charge of their eventual execution
- ❑ TM Data Acquisition (HK, Science and Video) from the various EuTEF elements, Formatting in accordance with the applicable CCSDS standards and delivery to ISS
- ❑ Monitoring of the various EuTEF elements on the base of the acquired TM HK data.

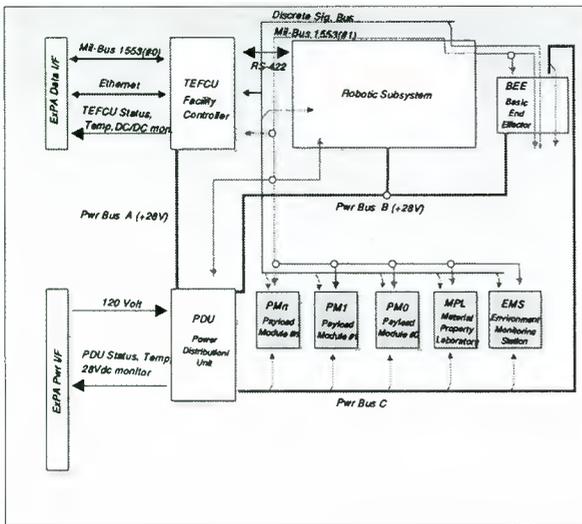


Figure 5 EuTEF Avionics Architecture

The Monitoring process produces TM reports to Ground in case of limit exceeding on parameters under surveillance. In case of critical situations automatic recovery actions (i.e. put EuTEF in a safe configuration) are implemented as well.

A **Material Properties Laboratory (MPL)** is part of the Facility Infrastructure with the purpose of measuring the thermo-optical properties of the material samples exposed to the Space Environment. The MPL is endowed with a spectroreflectometer and a microscope.



Figure 6 PM architecture

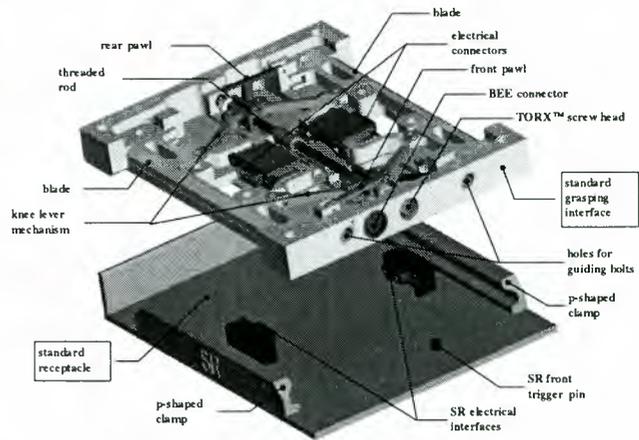


Figure 7 EuTEF Standard Grasping Unit

At this purpose it is equipped with one slot where Trays can be temporarily inserted by means of the Robot S/S. MPL allows measuring properties like emissivity, solar absorptivity and reflectivity. The microscope provides visual inspection of the superficial characteristics of the samples

An **Environment Monitoring Station (EMS)** provides EuTEF with centralized measurement capabilities for Pressure, Atomic Oxygen and Contamination.

1.5.3 Investigation Interfaces

EuTEF foresees the following three kind of investigations:

- 1 Investigation to be carried out with a dedicated hardware (Instrument)
- 2 Robotics Investigation requiring or not dedicated hardware
- 3 Set of sample to be exposed at space environment

The investigation 1) and 2) will be accommodated inside Payload Modules, while the 3) in the Tray system.



Figure 8 The Tray System

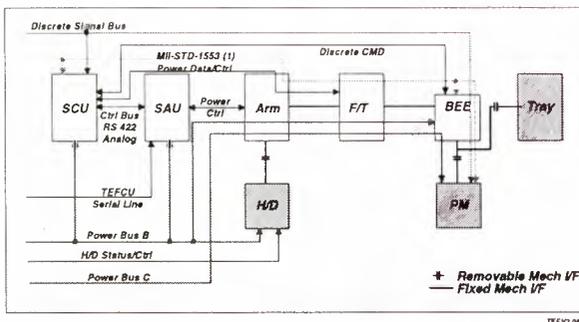


Figure 9 Robotic S/S architecture

The **Payload Modules (PMs)** are exploited for the accommodation of the various Instruments onto which it is required to perform Investigations. All PMs have the same base dimensions and identical Mechanical/Thermal/Electrical interfaces. The external envelope, as first design iteration, foresees two different PM dimensions (single and double) : 210 x 270 x 310 mm and 210 x 270 x 600 mm. The PM can be relocated in different exposure configurations by means of the Robot S/S. In order to allow that, a dedicated mechanism, called Standard Grasping Unit (see Figure 7), is mounted below each PM allowing them to be removed and safely latched in a new position inside receptacle attached to the EuTEF support structure.

The PM electrical I/F provides a 210 W @ 28V power interface, a MIL-STD-1553B serial bus data interface, a set of digital command to be used for instrument low level control, such as thermal control command, and an analog signal to measure the Instrument inside temperature. All these I/Fs are continuously connected to the facility avionics also during relocation.

Trays are exploited for providing time-limited exposure of small subjects pasted on their surface. As in the case of PMs, Trays are manipulated by means of the Robot S/S that can slide in and out the Tray. Trays are mounted inside dedicated PMs (see Figure 8) from which they can be fully extracted and moved independently to inserted in the MPL for in situ the measurement of superficial properties.

1.5.4 Robotic Subsystem

The **Robotic Subsystem** is in charge of performing all the EuTEF Operations that require physical movement of items like Payload Modules and Trays in order to implement the Investigation required by the Experimenter. Also Robotics investigation will be performed.

These required operations include actions like Open and Close Trays, Install and Remove Payload Modules into / from EuTEF receptacles, Point Trays / Payload Modules to a defined direction (see Section 1.5.5).

The Robotic Subsystem includes the following main items (see Figure 9):

- Robot Avionics, constituted by the Robot Control Unit (RCU) and the Servo Amplifier Unit (SAU).

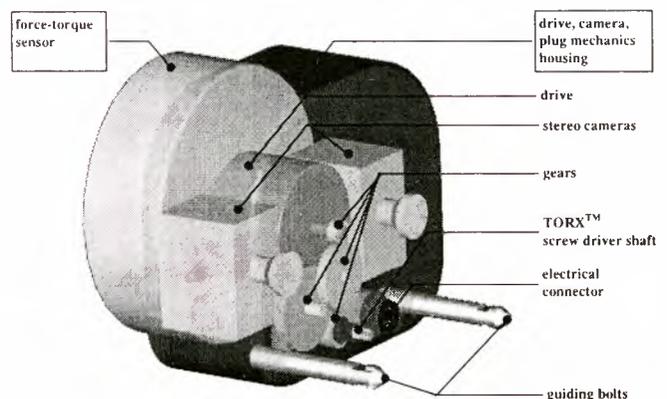


Figure 10 EuTEF Basic End Effector

- Robot Arm
- Force Torque sensor
- Basic End Effector (BEE)
- Hold Down Mechanism

The robot arm will be provided by the Italian Space Agency ASI and is constituted by a 7 d.o.f. anthropomorphic arm with endowed force torque sensor and hold down mechanism. The robot avionics will be developed by Carlo Gavazzi Space. The Basic End Effector will be developed by the German space agency DLR.

The **BEE**, permanently attached to the Robot Arm, is provided with a Standard Grasping Interface (SGI), featuring mechanical and electrical interfaces, to grasp all EuTEF items provided with SGI (Payload Modules, Trays, EMS). The design concept of the BEE is presented in Figure 10. The BEE is endowed with a screw driver shaft to actuate the SGU mechanism, with to guiding bolts and a connector providing power and data connection to the grasped payload during relocation. BEE includes a Stereo Camera and a lighting unit allowing scene and close-up visual inspection.

The **Hold Down Mechanism** is exploited for blocking the Robot Arm to the ExPA during launch/reentry phases. This active mechanism will be based on paraffin actuators and is endowed with redundant mechanisms preventing a catastrophic hazard represented by failures during the robot arm stowing operation.

1.5.5 Interactive Autonomy

The Interactive autonomy concept has been developed in order to reduce the necessity of continuous control from ground or on-board crew member of the robotics operations. As an example in Table 2 are presented a subset of the tasks that the robotic subsystem can be performed autonomously with ideally no control from ground. This idea has to be supported by a careful design in order to cope with the performance and safety requirements of the robotic operations.

The robotic arm due to its mechanical design aimed in reducing its mass and to the thermal deformation

an expected total error of ± 5 mm, ± 0.1 degrees. A sensorial feedback is required also for the simpler contact operation, and therefore the Robotic Arm is endowed with a force/torque sensor. The stereo camera mounted in the Basic End Effector is foreseen to be used during non-nominal operation in case of F/T sensor failure. The design of the mechanical I/Fs has to tackle this expected error and shall provide guiding surfaces allowing a smooth insertion of the BEE in the SGU and the SGU in the receptacle. Also the on-board computer has to provide force/torque algorithm to be tailored for the provided mechanical I/Fs.

The Interactive Autonomy operations have to be provided as a service to the payload and not only as a technology demonstration and due to the inherent risk of failure, are a big issue from the safety point of view. In term of safety theory they are classified as catastrophic hazards due to the risk of accidental release of the grasped object, such as the PM. In that case the PM will become a free fly object with obvious risks for the ISS.

The design of all the involved mechanism shall comply to a double fault tolerant design, meaning that shall be three independent inhibit/path that control failures classified as catastrophic hazard. As an example the mutual design of the BEE and of the SGU foresees that, when the PM is grasped and outside of the receptacle:

- ❑ the BEE motor cannot be powered also if a power on command is issued;
- ❑ the guiding bolts are both independently latched inside the SGU;
- ❑ the rotation of the BEE motor shaft is mechanically inhibited.

This inhibits are released if and only if the SGU is safely engaged inside the receptacles.

A redundant set of switch and current loop are provided in each moving subject allowing the TEFCU and the RCU to monitor the status of the facility also after a non-nominal power down/power up phase of EuTEF.

From ground any of the Interactive Autonomy tasks depicted in Table 2 can be issued to the facility using an object oriented MMI allowing to select graphically object and target position. The tasks are executed under complete control of the RCU, while the TEFCU provide monitoring and inhibit function of the robot activity in order to reduce the risk of facility damage in case of RCU failure. From ground the robot activity is continuously monitored based on the RCU TM and in case of unexpected behavior the task can be aborted.

2 Conclusions

This paper has presented the current status of the development of the EuTEF facility corresponding to the System Requirement Review pending on middle of

June. Some of the key features of the EuTEF design has been highlighted although, for sake of brevity, some other important aspects, such as e.g. thermal design, structural design, are here missing. The manufacturing, integration and testing phase will last for two years ending with the launch of the facility in late 2002 by UF4.

3 Reference

J. Dettmann, G. Visentin, R. Aceti, D. Andresen, "European Technology Facility" *Proc. of the 2nd European Symposium on the Utilization of the ISS, ESA-ESTEC, November 1998.*

Robotic Tasks
<p>RT1 : ActuateObject</p> <p>Approach and grasp a SGI, then exert a prescribed force/torque until termination condition is met. Ungrasp and retract.</p>
<p>RT2 : CloseTray</p> <p>Approach, grasp the Tray, insert it to the fully closed position. Ungrasp and retract with the arm.</p>
<p>RT4 : InstallPM</p> <p>Approach the grasped PM to a docking location. Install it in the receptacle. Ungrasp and retract with the arm.</p>
<p>RT8 : PointPM/PointTray</p> <p>Orient the PM/Tray attached to the arm in a prescribed direction</p>
<p>RT11 : TrackObject</p> <p>Point the object attached to the arm to a moving external object given the trajectory in EuTEF coordinates</p>
<p>RT12 : TransportPM/TransportTray</p> <p>Move the PM/Tray attached to the arm to a prescribed position or along a trajectory without contact to the environment</p>

Table 2 Interactive Autonomy Robot Tasks

PAYLOAD TUTOR (PAT)

A RELOCATABLE PAYLOAD ROBOT FOR ISS INTERNAL AUTOMATION SYSTEM

S. Di Pippo, R. Mugnuolo, F. Bracciaferri (Agenzia Spaziale Italiana)
Willie B. Williams (ISS Payload Integration Manager NASA-JSC)
G. Visentin (ESA - Automation and Robotics Section)
G. Colombina, E. Pozzi (Tecnospazio)

1. INTRODUCTION

In 1996, a call for proposal was issued by NASA (NRA 15-OG3-6-16P), with the aim to collect ideas to enhance operations on Space Station.

Among more than 120 proposals to NASA, one of the three selected was a concept proposed by ASI for Internal Automation, called **Payload Tutor (PAT)** and consisting in a small relocatable robotics system which can be mounted by the crew close to the rack to be serviced.

A bilateral co-operation ASI/NASA is now in progress, with the aim to prove the validity of the concept on an American payload rack: once validated, the concept could be applied extensively to American payloads.

A draft memorandum of understanding between ASI and NASA on PAT development is now ready for its finalisation and signature.

This foresees the following main steps:

- in orbit demonstration by end of 2003;
- use of PAT on American payloads in 2004.

For the first demonstration NASA will provide Shuttle launch services plus crew time and ISS infrastructure, while ASI will provide the system with a dedicated task

board to demonstrate PAT capabilities. The demo phase will end with the operation on ASI and/or NASA payload.

Particular care will be given to MMI aspects, in order to allow monitoring and control from ground and/or from the on board crew.

It is worth mentioning that ASI is now planning the development of a ground prototype, for which a co-operation with ESA for the robot controller is under definition.

This paper presents the concept and the programmatic aspects of the development.

2. TECHNICAL DESCRIPTION

2.1. The Express Rack

The International Space Station (ISS) is an international, Earth orbiting, research facility. Its mission is to conduct scientific, technological and commercial application research in a microgravity environment, with emphasis on long duration activities.

In order to ease and widen the utilization of ISS, NASA has created the Expedite the Processing of Experiments to Space Station (EXPRESS) program.

EXPRESS provides payload accommodations that will allow quick, simple integration with the use of standardised hardware interfaces and a streamlined integration approach.

The EXPRESS Rack utilizes (see fig 2.1-1) an International Standard Payload Rack (ISPR) in conjunction with secondary structure and avionics hardware, resulting in a simple payload interface.

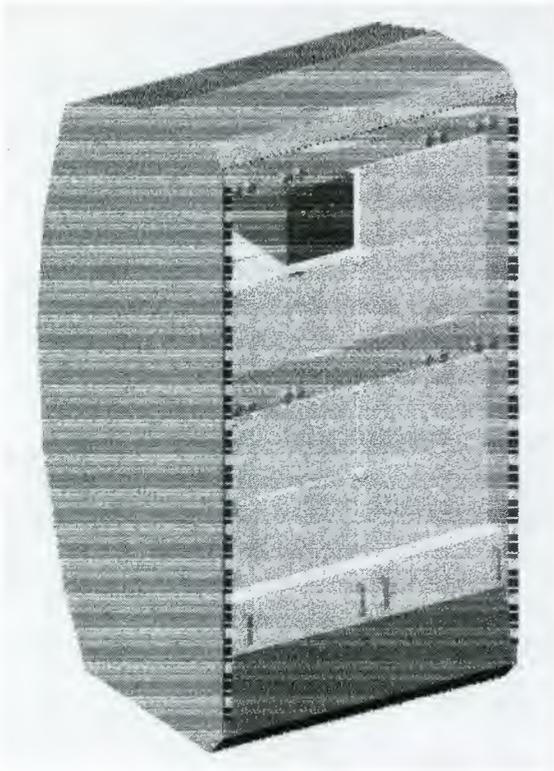


Fig. 2.1-1: EXPRESS Rack concept.

EXPRESS racks are used to provide payload services to facilitate payload operations for all types of payload users. The rack provides accommodations for eight middeck lockers and two Standard Interface Rack (SIR) drawers. Each middeck location can accommodate 72 pounds of experiment equipment, including container weight. A total of 2 kW of 28 Vdc power is provided to the payloads in the rack, with each payload position receiving up to 500 W. Interfaces for EXPRESS Rack payloads on the ISS will include RS422, ethernet, analog, discrete and video. In addition, air cooling will be available at each payload location.

Payload may be arranged in (see fig. 2.1-1):

- lockers;
- large or small drawers;
- mounting plates, these may occupy 1 or 2 (contiguous) middeck locations.

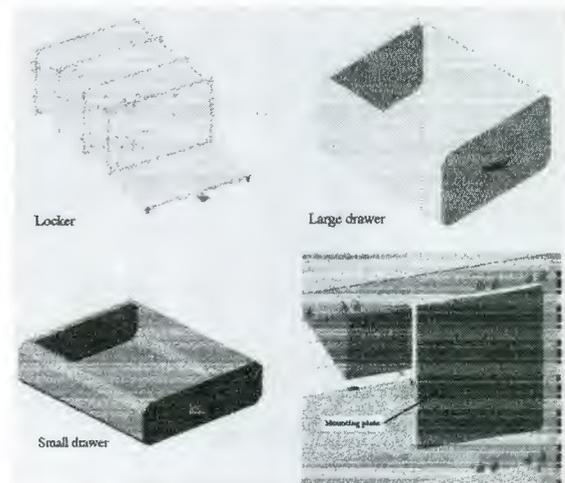


Fig. 2.1-1: Payload accommodation possibilities.

The EXPRESS Rack eases the work of payload developers, offering a set of standard hardware and interfaces. The concept, however, still requires crew for operations. To save crew time for payload operations, the relocatable robot is planned to be used.

2.2. PAT Concept

The basic idea is to perform the most common operations on payloads in a similar way an astronaut would do it. This has a double effect:

- first it minimise the changes in classical payload developments;
- second, it allows the users to think at operations in the same way as they were supported by an astronaut.

The robotics concept is shown in fig. 2.2-1.

The robot is mounted at the rack on which it has to operate. It is a short manipulator arm with 6 d.o.f., mounted on a vertical rail.

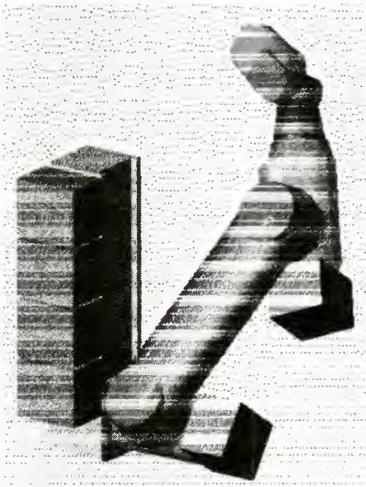


Fig. 2.2-1: ISS Relocatable Payload Robot Concept.

The Arm is mounted on the rail in order to provide full dexterity in performing rack operations. At the tip of the Arm an End Effector is mounted. The electronics is mounted partially on the rail and partially on the arm.

The Robot kinematics provides a very good dexterity to the system. End Effector orientation is provided with axes 4, 5 and 6 of the Arm, while its position is mainly provided by the vertical rail and the axes 1,2 and 3 of the Arm. The End effector is sensed and a camera is mounted on it for inspection purposes.

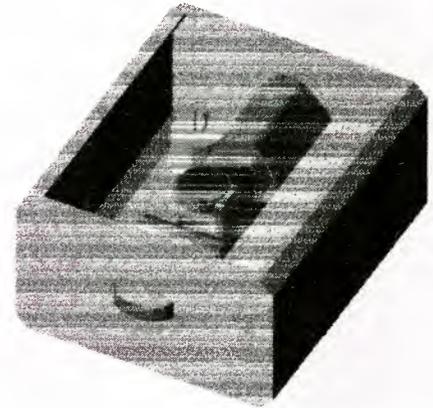
The whole robot can be removed, by the astronaut, from the rack and placed on another rack to be serviced. For operating in parallel different racks, more than one robot can be used. During initialisation activity, an automatic Robot calibration procedure is foreseen, to calibrate the Robot w.r.t. the rack it has to operate.

It has to be highlighted that, once the Robot has been 'plugged' on the rack to be serviced, no astronaut intervention is necessary, as the robot can be controlled directly from ground. The control from ground consists of simple commands to start robot program tasks already present inside the robot controller, according to a user defined sequence and with user defined parameters.

Required crew activities for PAT are summarised the following:

Installation

- PAT retrieval from the Drawer (see fig. 2.2-2)



- Mechanical connection in front of the rack
- Electrical connection
- Switch on

During nominal operations

NONE.

During non-nominal operations

- Switch off power
- Backdrive the Arm

The PAT system is being designed to fulfill the following requirements:

- PAT operates on payloads hosted in the EXPRESS Rack;
- PAT has no impact on EXPRESS rack design;
- the Crew deploys PAT wherever is needed. That means that PAT may be mounted in different locations on the same EXPRESS rack as well as on different racks;
- PAT operates on payload subjects (e.g. samples, drawers, trays, doors) accessible on/beneath the aisle-facing surface of an EXPRESS rack;
- the basic PAT operations are the classical sample manipulation (extract, transfer, insert),

opening/closing of lockers/drawers, plus some actuation (e.g. flipping switches, rotating dials). The subjects for manipulation and open/close must be equipped with appropriate grasping interfaces;

- PAT operations can be started and monitored from ground, as well as from a Crew PC.

3. PAT SERVICING OPERATIONS

3.1 PAT Servicing Operations on EXPRESS Rack Standard Items

Among the possible operations PAT needs to perform, there are some which are related to EXPRESS Rack items. In fact payloads sometimes make use of standard middeck lockers or drawers to store samples or consumables.

PAT is designed to be able to handle such items. Examples are reported in fig. 3.1-1 (PAT opening a middeck locker by hinging its door. The door launch locks have been previously released by the crew. The door is kept closed by the magnetic latch. PAT grasps a flap and hinges out the door.) and fig. 3.1-2 (PAT sliding out a large middeck drawer. The drawer is pulled out by its handle).

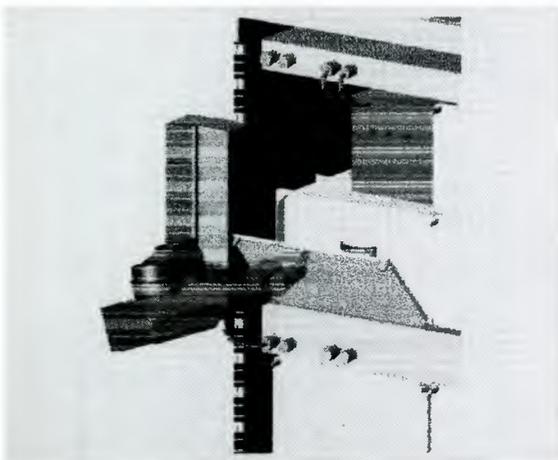


Fig. 3.1-1: PAT opening a middeck locker.

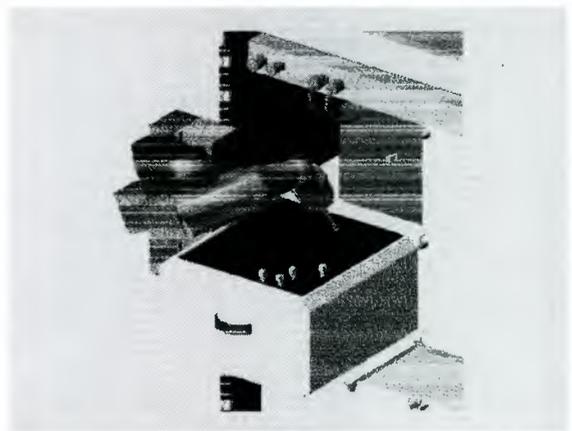


Fig. 3.1-2: PAT sliding out a large middeck drawer.

3.2 PAT Servicing Operations on Payload Samples

The main use of PAT is, of course, the handling of samples. The operation is logically divided into three steps: picking, transporting and placing.

Fig. 3.2-1 shows PAT feeding three samples (in this case vials of liquid) into a facility. The vials are picked-up from their storage position (in a foam mat inside a standard middeck drawer) and inserted into three holes of the facility.



3.3 PAT Servicing Operations on Payload Mechanisms

PAT is also able to actuate payload mechanisms. Some actuation tasks are shown in fig. 3.3-1 (rotation of a knob), fig. 3.3-2 (actuation of a slider), fig. 3.3-3

(open/close of shutters) and fig. 3.3-4 (actuation of flip switch).



Fig. 3.3-1: Rotation of a knob.



Fig. 3.3-2: Actuation of a slider.



Fig. 3.3-3: Open/close of shutters.



Fig. 3.3-4: Actuation of flip switch.

4. PAT MAIN ELEMENTS

4.1 PAT Arm

The PAT Arm can be considered 'classic' for what concerns the architecture.

It is a machine with double offset and six plus one degree of freedom.



Fig. 4.1-1: PAT Arm.

PAT design is done in order to:

- capitalise on the well known performance in real environment, of certain well known components, such as ETEL actuators, in order to benefit the reliability of the system;
- attempt a standardization of these components even in cases where the specific performance does not necessitate it, to benefit the imminent industrialization of the product;
- use of proven rather than experimental technology and components.

4.2 PAT Electronics (CIRCUS)

The CIRCUS system shall be implemented in a modular way (fig. 4.2-1) and shall consist of one Electrical Distribution/Power Supply Unit, one RCU board and four SCU boards. RCU and SCU shall be based on the same type of board, differently configured. RCU, SCUs and the Power Supply Board shall be ORU,

replaceable with a simple exchange of boards.

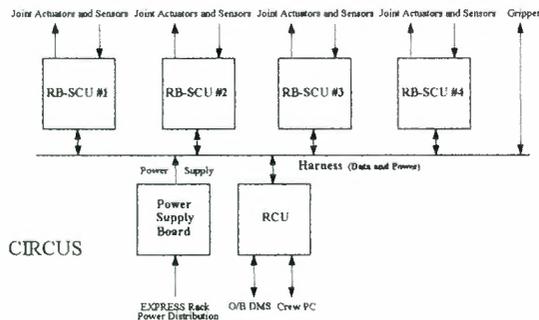


Fig. 4.2-1: CIRCUS module overview.

RCU represents the interface between the EXPRESS Rack or the Crew PC and SCU's. RCU and SCUs will make use for the computational part of the Epson Card-PC, in order to minimise volume and mass. The Epson Card-PC will contain the required amount of RAM and the resident code on non-volatile memory support.

The communication between RCU and SCU's will be realised via a dedicated bus implemented as a part of the Electrical Distribution/Power Supply Unit.

A modular Electrical Distribution/Power Supply Unit, allowing for data exchange and power distribution, will be developed. The Electrical Distribution/Power Supply Unit will be composed by:

- a common, stand-alone Power Supply Board;
- a Bus Interface Card for each of the SCU units.

The Electrical Distribution/Power Supply Unit will be implemented in a distributed layout, thanks to the Bus Interface Cards. In this way, each of the SCU units will be able to be physically detached from the others, except for the required cabling.

The mass memory storage device will provide a non-volatile support for data to be preserved when the system is in hibernation mode.

RB-SCU Concept

The Resolver Brushless - Servo Control Unit (RB-SCU) shall consist of one board (Base-Board), supporting, in a modular way:

- 1) the CPU Card EPSON Card-PC;
- 2) the bus interface;
- 3) one Motor/Resolver Interface Unit;
- 4) one R/D and S&H Unit;
- 5) two servo amplifiers;
- 6) solid state current limiters.

The EPSON Card-PC shall provide the computational power required to receive position and velocity set points and operation commands from RCU and perform the functions of μ -interpolator, position/velocity/current control, motor commutation (based on resolver reading) and on/off brake control. It shall also implement watchdog functions with status/error reporting to RCU.

The bus interface, which shall be part of the Electrical Distribution/Power Supply Unit, shall connect to the Card-PC either via the card ISA bus or the card parallel port and shall allow data exchange with RCU.

The Motor/Resolver Interface Unit shall interface the Card-PC with the resolvers and the servo amplifiers. The unit shall be able to drive up to two servo amplifiers and acquire the input signals of up to two Motor Shaft Resolvers, for which it shall also provide the resolver excitation signal.

The R/D and S&H Unit shall contain the Resolver to Digital converters and the Sample & Hold function to hold the current reference vector values for the servo units.

The servo amplifiers shall be a re-use of the SPEAR servo amplifies whose implementation shall make use of

custom-designed hybrid circuits. The servo amplifiers shall therefore implement thermal and current monitoring/protection and interface to robotic joints equipped with brushless DC motors (2 or 3 phase) and failsafe brake. The servo amplifiers shall also implement active current limitation by means of a solid state current limiter.

4.2.1 RCU/SCU Base-Board Standardization

In order to allow for minimal base-board differentiation between RCU and SCUs, a modular approach shall be followed on these units. They shall be based on the same, standardised base-board and configured via software and additional plug-in boards to behave like a RCU or a SCU. This approach also eases the implementation of a redundancy concept.

The Carden, the EPSON Card-PC and the DiskOnChip2000 are COTS products. The Servo Amplifiers shall be a miniaturisation of the SPEAR ones. The internal Bus Interface shall be a part of the Electrical Distribution/Power Supply Unit. The special functions of Motor/Resolver, RS422 and I/R interface units are implemented on the base-board itself.

The configuration requirements needed to make the standard board behave like a RCU or a SCU are shown in fig. 4.2.1-1. The dashed rectangles imply the absence (logical or physical) of the corresponding part. Weight saving can be obtained by not mounting unnecessary components and/or connectors on the boards which shall perform only as RCU or SCU.

The operating system delivered with CIRCUS RCU shall consist of a VxWorks operating system including a Board Support Package for CIRCUS constituents.

Both are foreseen to be executed on RCU processor.

The VxWorks operating system as well as the Board Support Package are commercial off the shelf products which are both

configured to meet the needs of CIRCUS project.

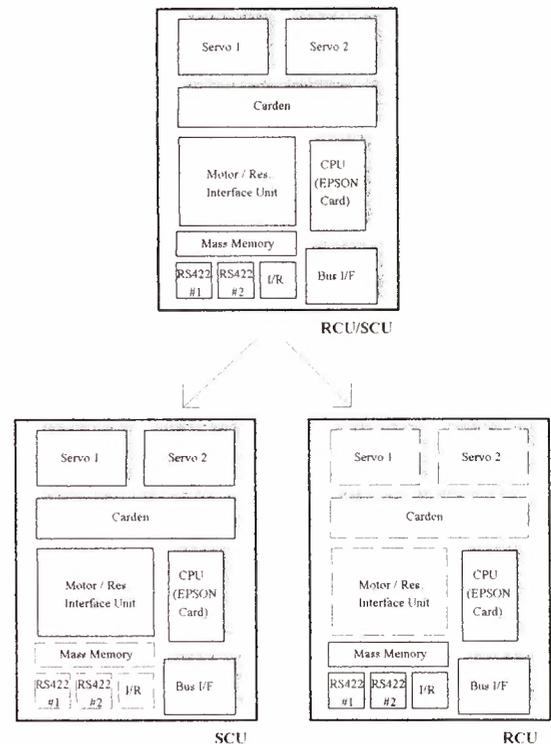


Fig. 4.2.1-1: RCU/SCU configuration.

5. CONCLUSION AND EVOLUTION

PAT can be considered as a general purpose tool, flexible enough to support the crew for payload servicing operations.

PAT, in fact, would allow to free the crew from tedious, repetitive tasks (such as exchanging experiment samples) and, as consequence, to increase the efficiency, throughput, cost and accuracy of standard payload operations.

In addition, PAT offers the possibility that the "real" investigator, namely the scientist on Earth, monitors and interacts with her/his experiment through the telepresence link and the robot.

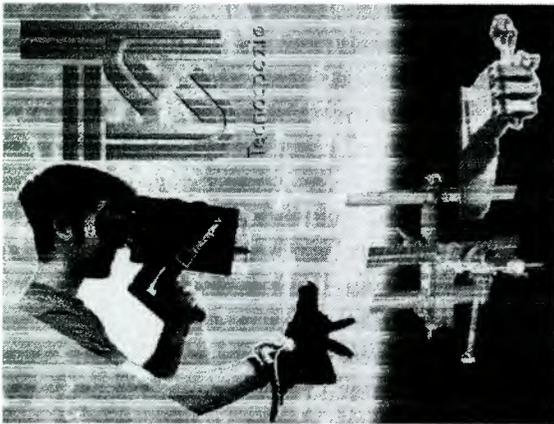
We estimate the following mission advantages for replacing crew members in the different activities:

- *Enhance crew time efficiency more than 25%*
- *Reduce routine task time and complexity.*
- *Advance autonomous capabilities in all station systems*
- *Enhance microgravity environment.*

It is planned to have by the end of the year 2000 a ground prototype, which will operate on EXPRESS Rack payloads on ground. This will be used to validate the concept and to consolidate interfaces to payload developers and users.

The concept will evolve towards ground programming and control based on virtual reality (i.e. to give the ground operator the impression to operate directly himself on the payloads like he was close to them) especially for telepresence operations.

The operator interacts with the scene and the robot motion is controlled with a glove and 'head' which allows the operator to modify and correct the planned sequence.



The operator has full control on the manipulator and the relevant anthropomorphic hand.

It is possible to execute on the payloads either automatic operations (defined during the programming and verification phase) or in a direct teleoperated mode.

Also during the execution of automatic sequence, the operator has the possibility to stop and modify the sequence if necessary.

WEARABLE EXO-SKELETAL ROBOT, SKILMATE, AND ITS APPLICATION TO EVA SUITS

Yoji UMETANI, Yoji YAMADA, Tetsuya MORIZONO
Toyota Technological Institute
2-12-1, Hisakata, Tempaku-ku, Nagoya 468-8511, JAPAN
phone: +81-52-809-1790, fax: +81-52-809-1794, e-mail: umetani@toyota-ti.ac.jp

Tetsuji YOSHIDA, Shigeru AOKI
Shimizu Corporation
Seavans South 1-2-3 Shibaura, Minato-ku, Tokyo 105-8007, JAPAN
Phone: +81-3-5441-0910, fax.: +81-3-5441-0439, e-mail: ty@spo.shimz.co.jp

Abstract

The authors propose a new concept "SkilMate", a kind of wearable robot, in order to reduce the fatigue of astronauts and to demonstrate their skill in EVA. This paper describes an outline of the SkilMate project. They also discuss the concept in view of the man-machine system, the configuration design, the control system, and the final target of the project.

1. Problems on Workability of Spacesuit.

It has been already reported that the operation performance of EVA in space is very bad and that the fatigue of the astronaut is intense. It is because that the comfort of the spacesuit is very terrible. This fact becomes a large problem which should reach solution for future space development.

There, we have the following three problems which must be cleared in order to improve the operation performance in this spacesuit wear and in order to demonstrate skill of the EVA work acquired by intense training of the astronaut.

- (1) Attaching servo-actuators on each joint of the spacesuit to make it actively controlled.
- (2) Controlling the position, the posture and the impedance of the whole movement system as the astronaut efficiently carries out the

mission.

- (3) Applying the compound haptic device on the finger palm to detect force, pressure, and slip senses.

Hence we denote that such machinery as satisfied with these three items should be a robotized machine. The spacesuit mounted with such three kinds of functional ability described above should work for the skilled astronaut to accomplish EVA activity skillfully.

2. Concept of "SkilMate".

The authors propose a new concept named SkilMate. SkilMate is defined as a generic name of machines or devices characterized as below.

- (1) It is to be worn by a man of skill in case that he should unavoidably work to accomplish tasks as an expert under a hostile environment.
- (2) It is preferably exo-skeletal structured to envelop the man's body, and it is movable enough flexibly as the man is able to move as his will. Therefore, it should be mounted with haptic receptors or sensors in order to enhance intelligent workability. Consequently it looks move like an anthropomorphic robot when it is worn.

SkilMate is nothing but a wearable exo-skeletal

machine for the skilled worker to move maintaining his skill. It assists him in power as well as in skill due to the servo mechanisms of SkilMate.

The benefit of SkilMate will arise in the case where the worker should need to slip on some clothes to protect himself from a hazardous environment.

3. Features of “the SkilMate” as New Concept.

By inventing various kinds of machine, the human, being were able to expand the functional ability. Especially, the human is able to convert the intelligent function into automatic machines or robots recently. However, in spite of the effort of the conversion, human skill is still difficult to convert even if we use the most advanced intelligent robot technology. The advanced skill of the worker remains, therefore, very important and the conversion of such talent as veteran engineer executes the mission under the ultimate environment is also important.

In this research project, we assume that the astronaut wears a spacesuit in which a servo-actuated skeletal structure mounted with haptic sensors is attached, as if the handicapped wears artificial limb or hand. Thus we propose a new concept called SkilMate. Therefore, “the SkilMate” is a generic name of the machine with such effect, as described above. It can be sufficiently demonstrated that a skilled worker is able to perform his tasks most efficiently, when the worker wears it. He is also amplify the range and speed at work. In addition, the fatigue is drastically reduced by the wear.

The SkilMate is also defined as a wearable robot, because it shows its functional ability in case that it is worn by a skilled worker.

The machine of the resemblance to the SkilMate was proposed in 1960 generation.[1] The foreseeability must be evaluated. However, it was not successful by lack of the generic technology and the uncertainty of its concept.

The SkilMate defined in this paper is a well-defined machine, employing the most advanced robotic technology. It never moves as far as it is driven by the wearer even if the power is turned on. It moves, only if the human wears and drives it. The SkilMate is just an instrument in a sense. It is neither a powered machine nor automatic machine. And we say, it is not a robot standing alone. However, it moves on as the wearer’s intention, if he once fixes it and manages the body. It demonstrates the function which dynamically exceeds the human ability. It is able to attain controllability for stabilizing the movement and the posture of its own body. The skill in manual working is as efficient as that in bare hand. Therefore, it accomplishes the work as an intelligent machine. However, it has nothing on the autonomous function. It is just a passive machine having non-autonomous intelligent response incorporated in the human intellectual operation system. It is a uniquely defined machine that we have never experienced until now.

4. Compensation of Ballooning Caused by Spacesuit.

When an astronaut wearing a spacesuit bends the joint (elbow joint, for example) during performing a task in the extravehicular activity (EVA), the vacant volume contained inside between the arm and the spacesuit decreases, then the air in this vacancy is compressed. The compressed air causes a spring effect on the spacesuit which extends the bent astronaut’s joint back to its extended posture.[2] This spring-like

effect is known as “ballooning”.[3] The ballooning fatigues the astronaut in a short term and reduces efficiency of the task during the EVA, because he has to keep exerting the bending moment on the joint in order to hold it in the bending posture.

It is considered that the above disadvantages of the current spacesuit can be improved by compensating the ballooning with the SkilMate. However, the following problems are anticipated for the compensation:

- (1) The spring effect caused by the ballooning will be nonlinear. Moreover, it will be difficult to identify the model of this effect accurately.
- (2) The ballooning effect will vary according to the astronauts, because the shape and volume of the air space is different among them. This anticipation can be supported by an investigation on the EVA glove which reported that the hand of the astronauts is, the more the grip drops compared to the bare hand.[4]

5. Tuning of Control Parameters.

Not only compensating the ballooning, but also an ability to help the astronaut to show the skill is necessary for the SkilMate in order to improve the performance of the EVA. Therefore, the SkilMate is needed to determine its own control parameters so as to realize this ability. This is the second control problem of the SkilMate. In this paper, we consider this problem as a parameter tuning one on mechanical impedance of the whole system including the spacesuit, the SkilMate and the object handled by the astronauts. According to this consideration, the problem can be specified as follows:

- (1) How does the SkilMate determine the ideal inertia and viscous coefficients?

- (2) Does the SkilMate fix these parameters all over a task, or set them as time is varying?
- (3) Whether does the SkilMate append the spring effect or not? If the spring effect is preferred, where should its equilibrium point be located?

In order to solve this problem, we need an experimental study to find out the ideal impedance for tasks performed in the EVA. Since it is desired that the SkilMate works well under variation of conditions such as a motion velocity, and an initial and terminal postures of the task, this variation should be taken into account in our study. It is finally needed to express the ideal impedance quantitatively so that they can be referred by the controller of the SkilMate.

If we consider that the SkilMate is a robot cooperating with a human, a previous study on a cooperative task to carry an object can be referenced.[5] However, this study focused only the viscous coefficient, and has not dealt with the variation of the task conditions.

6. Goal of SkilMate Project.

SkilMate project was put into practice in August 1998. This project aims to manufacture mainly an exo-skeletal structure assembly to be worn by the astronauts for EVA (Extra-Vehicular Activity).

This structure assembly will consist of three parts, the upper torso, the lower torso assembly, and the pair of arms with gloves. Similar to SSA (Space Suit Assembly), these parts are able to be assembled when it is put on. The movement of every joint (shoulder, elbow, wrist, and so on) is controlled with servo-actuators which assist in the intentional movement of the astronaut.

The structure of gloves are most carefully designed, and the haptic devices such as tactile or slipping-off sensors/displays are indispensable to

maintain the skillness of the astronauts.

Reference Literature

- [1] Makinson, B.J., Research and Development Prototype for Machines Augmentation of Human Strength and Endurance, Hardiman I Project, GE, 1971
- [2] B.R.Prouty, R.C.Wilde and M. Gan, "Evolution of the Extravehicular Mobility Unit for Future Missions", SAE paper, 911349, 1991.
- [3] J.A. Main, S.W. Peterson and A. M. Strauss, "A Prototype Power Assist EVA Glove", Publ. Soc. Automot. Eng., SP-872, 85-93, 1991.
- [4] J.R.Roesch, "Handgrip Strength with the Bare and in the NASA Spacesuit Glove", Proc. the Human Factors Society, 786-790, 1987.
- [5] R. Ikeura and H. Inooka, "Variable Impedance Control of a Robot for Cooperation with a Human", Proc. IEEE Int. Conf. On Robotics and Automation, 3097-3102, 1995.

Acknowledgement

SkilMate project is financially supported by NEDO, New Energy and Industrial Technology Development Organization.

Autonomy in Planetary Exploration

A Comparison of Two Traversability Based Path Planners for Planetary Rovers

M. Tarokh¹, Z. Shiller² and S. Hayati³

¹ Robotics and Intelligent Systems Laboratory
San Diego State University, San Diego, CA 92182-7720

² Dept of Mechanical and Aerospace Engineering
University of California, Los Angeles, CA 90095

³ Jet Propulsion Laboratory
California Institute of Technology, Pasadena, CA 91109

Abstract

The paper presents two path planners suitable for planetary rovers. The first is based on fuzzy description of the terrain, and genetic algorithm to find a traversable path in a rugged terrain. The second planner uses a global optimization method with a cost function that is the path distance divided by the velocity limit obtained from the consideration of the rover static and dynamic stability. A description of both methods is provided, and the results of paths produced are given which show the effectiveness of the path planners in finding near optimal paths. The features of the methods and their suitability and application for rover path planning are compared.

1 Introduction

Following the successful launch and deployment of Mars Sojourner rover, NASA has planned further rover missions to Mars starting in 2001 with Marie Curie, a rover similar to the Sojourner. Two additional rover missions in 2003 and 2005 have been planned for in-situ experiments, and another in 2007 for sample return to Earth. An important element for the success of these missions is incorporating a reasonably high level of autonomy in the rover so that it can traverse distances of 100 meters or more per communication cycle. In order to traverse these distances, it is necessary to delegate the motion planning task to the rover using the image obtained from mast mounted cameras. The challenge is then to use these images to perform on-board path planning.

The existing path planners focus almost exclusively on obstacle avoidance, treating obstacles as forbidden regions and the rest of the terrain as free spaces [1]. This binary environment is not appropriate for the Martian terrain and a rover that can climb over some rocks [2] if such traversals result in more optimal routes. In fact NASA's experience with Sojourner has revealed

many cases where a binary obstacle model has resulted in halted motions, often leaving the rover in an undesirable situation [3]. Recently several path planners have been developed that consider the traversability of the terrain [4]-[7]. Terrain topology and simple vehicle dynamics are considered in [4] to generate global optimal paths on general terrain. In [5] the shortest feasible path for off-road vehicles is computed. A genetic algorithm is used in [6] to synthesize path from segments, each evaluated for its static stability and for satisfying certain mission tasks. A recently developed planner [7] uses fuzzy logic to characterize the terrain traversability, and then finds traversable paths in a rocky terrain.

The purpose of this paper is to discuss two path planners for possible Mars rover applications. The first algorithm is based on fuzzy characterization of the terrain roughness, and the use of a genetic planner to optimize a fitness function. The second algorithm considers constraints imposed by certain vehicle dynamics and terrain topology to come up with an optimal path. The common feature of both planners is finding paths that are optimal in the sense of both distances and traversability, where the latter quantifies the ease of traversal of the terrain. These two algorithms find paths that result in reduced rover energy consumption and enable exploring larger regions of the Martian terrain.

2 Genetic Path Planner

The path planner starts by creating several random paths between start and goal points on the terrain. These initial paths in general go through rough or impassable regions on the terrain, and must be improved. This improvement is achieved by applying certain genetic operators to a randomly selected path from the population. Each genetic operator has a particular role in bringing about a change in the path. For example,

replace operator replaces an undesirable way-point (a way-point on a rough region), with a random and potentially better way-point. The selection of particular operator is based on the probability assigned to it. After a genetic operation is performed, the quality of all paths are compared, and the worst path is eliminated from the population. The process of applying a genetic operator to create a new path, and eliminating the worst path, is referred to as a generation. The population goes through generations and is thus evolved. After each generation, the quality of the paths is either improved or in the worst case remain unchanged. The evolution is continued until an acceptable path is found, or until a preset number of generations are performed.

2.1 Terrain Roughness

Consider a terrain divided into a grid of regular square cells whose size depends on the dimension of the rover, and the desired resolution of surface description. The roughness of a flat obstacle free cell is assigned a value of 0, and that of a rugged cell with large obstacles is assigned a value of 1. The measure of roughness depends on a number of parameters as follows:

- Height of the tallest obstacle in the cell - The roughness becomes smaller with a decrease in the rock height.
- Size or surface area of the cell occupied by obstacles or rocks - If two cells have rocks of the same height, the region with less rock occupied area is smoother and thus has a lower roughness value.

In addition to roughness, two path dependent quantities, namely path slope and curvature, affect the difficulty of the traversal by a rover. These will be considered in Section 2.2.

The most commonly used sensors for mobile robots are cameras and their associated image processing hardware and software. Despite the availability of vision processing software, exact determination of the heights and sizes of rocks affecting roughness is not possible. These parameters can be found, at best, approximately due to errors, misinterpretations and ambiguity involved in extracting information from images. It is therefore essential to set the problem in a fuzzy and approximate reasoning framework.

The height of the tallest rock in the cell under consideration, h , and the size or surface area occupied by rocks in this cell, s , are used to find the cell roughness ρ . The crisp values of h , s and ρ are fuzzified to obtain the linguistic variables \tilde{h} , \tilde{s} and $\tilde{\rho}$, respectively. The "if-then rule" of the following form is employed to obtain the fuzzy roughness,

$$\text{if } \tilde{h} \text{ is } \tilde{H}_k \text{ and } \tilde{s} \text{ is } \tilde{S}_k \text{ then } \tilde{\rho} \text{ is } \tilde{\rho}_k \quad (1)$$

where \tilde{H}_k , \tilde{S}_k and $\tilde{\rho}_k$, $k = 1, 2, \dots, \nu$ are the linguistic values associated with \tilde{h} , \tilde{s} and $\tilde{\rho}$, respectively, and ν

is the number of linguistic values. The fuzzy sets H_k , S_k and ρ_k are used to quantify the linguistic statements " \tilde{h} is \tilde{H}_k ", " \tilde{s} is \tilde{S}_k " and " $\tilde{\rho}$ is $\tilde{\rho}_k$ ", respectively. The fuzzy sets H_k for the height are chosen as very low ($H_1 \equiv VL$), low ($H_2 \equiv LO$), medium ($H_3 \equiv ME$), high ($H_4 \equiv HI$) and very high ($H_5 \equiv VH$). The membership functions μ_{H_k} for these fuzzy sets are standard triangular and have equal base width with a 25% overlap. The fuzzy sets associated with the rock size are tiny ($S_1 \equiv TI$), small ($S_2 \equiv SM$), medium ($S_3 \equiv ME$), large ($S_4 \equiv LG$) and extra large ($S_5 \equiv XL$), and are also triangular with 25% overlap. The fuzzy sets for roughness are very low ($\rho_1 \equiv VL$), low ($\rho_2 \equiv LO$), medium ($\rho_3 \equiv ME$), high ($\rho_4 \equiv HI$) and very high ($\rho_5 = VH$). The membership functions μ_{ρ_k} for the roughness are designed to be triangular with different base widths to give more weighting to rougher terrains.

The rule matrix implementing (1) is given in Figure 1, and consists of 25 rules which are self-explanatory. Zadeh's compositional rule of inference, and center of height defuzzification method is used to obtain the crisp value of the cell roughness ρ .

2.2 Path Representation

A path is represented by a sequence of way-points connecting the start to the goal. The way-points W_k , $k = 1, 2, \dots, m$ are specified by their (x_k, y_k) coordinates on the terrain. The generation and evolution of a path refers to the creation and modification of the way-points. These way-points in turn specify the terrain cells that the path traverses over. A cell that is located on a path, will be referred to as a *path cell*, and has two main attributes as follows:

- The roughness ρ_i of the cell, which provides information on the heights, sizes and concentration of rocks on a cell, as described in Section 2.1.
- The curvature or jaggedness of a path cell is obtained using the information about the way-points. Specifically, the curvature $\hat{\zeta}_k$ of the way-point W_k is defined as

$$\hat{\zeta}_k = \frac{d_k}{D_k} \quad k = 1, 2, 3 \dots, m \quad (2)$$

where d_k is the perpendicular distance of W_k to the line segment joining the previous way-point W_{k-1} to the next way-point W_{k+1} , and D_k is the distance between W_{k-1} and W_{k+1} . Note that $\hat{\zeta}_k$ is a dimensionless quantity, and that $0 \leq \hat{\zeta}_k < \infty$. Furthermore, (2) also gives the curvature of the path cell that contains a way-point.

It is noted from Section 2.1 that roughness is normalized and varies between 0 and 1. However, curvature can have large values. In order to enable easy comparison between the two cell attributes, we normalize

curvature as follows:

$$\zeta_i = 1 - e^{-a\hat{\zeta}_i} \quad (3)$$

where a is a constant whose role will be explained shortly. Note that $0 \leq \zeta_i \leq 1$ for all values of a .

The above two quantities, namely roughness and curvature, which are attributes of path cells, are combined to define a *cell impedance* η_i as follows

$$\eta_i = \frac{1}{2}(\rho_i + \zeta_i) \quad (4)$$

The cell impedance varies between 0 and 1 and quantifies the difficulty of the path cell traversal by a rover. Consequently, a path cell containing no rocks that is located on a straight path segment will have a minimum impedance of 0. On the other hand a very rough cell on a jagged path segment will have a maximum impedance of 1. The constant a in (3) determines the weight given to curvature relative to the roughness. Lower values of a reduce the contribution of curvature to the overall cell impedance. It is noted that other path attributes such as slope can easily be included in the above formulation of the path impedance.

A cell with an impedance of more than a threshold becomes *intraversable*. The value of the threshold is chosen based on the mobility characteristics of the particular rover being used. We identify a path as being traversable if every cells on the path is traversable, otherwise the whole path becomes intraversable. In the genetic evolutionary process, these two type of paths are treated separately. Although, traversable paths have priority over intraversable paths, the latter are not automatically discarded since they may prove to produce good offsprings later on during the evolutionary process. The *path impedance* is defined as the sum of impedances of all cells on the path, that is

$$\eta = \sum_{k=i}^n \eta_i \quad (5)$$

When a population of paths consisting of both traversable and intraversable paths are compared for selection, any traversable path is given preference over best (lowest η) intraversable path. However, when the population consists of only traversable paths or only intraversable paths, then the selection is based on lower values of η .

2.3 Genetic Operators

In order to evolve paths from one generation to the next, several operators have been devised. Two of these operators, namely cross over and mutation, are commonly used in genetic algorithms. Others are specifically designed for the path planner. Operators are

applied to way-points, and as a results of changes in way-points, the path cells are also changed. Note that each time an operator is applied, a new path is generated. If this new path produces a path impedance that is lower than the impedance of any path in the population, it is accepted as a new member of population, and the path with highest impedance is discarded.

Cross-Over

This operator randomly selects two paths from the population, say P_1 and P_2 , and divides each path into two path segments about a randomly elected way-point. Denoting these paths by $P_1 = (P_{11}, P_{12})$ and $P_2 = (P_{21}, P_{22})$, where P_{ij} is the j -th segment of path i , then two new paths are formed as $\hat{P}_1 = (P_{11}, P_{22})$ and $\hat{P}_2 = (P_{21}, P_{12})$.

Mutate

This operator randomly selects a path and a way-point in this path. It then changes the x, y coordinates of the selected way-point with random values. Mutate operator can produce a significant change in the path.

Replace

This operator is applied to an intraversable path. It replaces an intraversable way-point with one or more way-points whose location and number are random. If there are more than one intraversable way-points, one of them is selected randomly for replacement.

Swap

The operator interchanges the locations of two randomly selected way-points on a randomly selected path. The swap operator can be applied to both traversable and intraversable paths. It has the possibility of removing or introducing a "zig-zag".

Smooth

The role of this operator is to reduce sharp turns. The way-point with the highest curvature, say W_k , is selected and two new way-points are inserted, one on a randomly selected cell between the way-points W_{k-1} and W_k and the other on a cell between W_k and W_{k+1} . After this insertion, the way-point W_k is removed. The effect of this operation is the smoothing of a sharp turn. This operator is only applied to traversable paths.

Pull-out

This operator is intended to pull out a path segment from inside an intraversable region to its surrounding traversable region. Pull-out is more elaborate than the other operators, and details of its implementation is omitted here for the sake of brevity.

The probability of occurrence of an operator depends on the role played by it in the evolution of paths. An

adaptation scheme is devised to modify the probabilities based on the population diversity, and traversability. For example, if most paths in the population are similar and have high impedances, mutation is given higher probability and cross over is assigned a smaller probability. This is due to the fact that in this situation cross over of intraversable paths also produce other intraversable paths and a substantial change is needed which is achieved by mutation.

3 The Global Optimization Planner

This planner formulates the motion planning problem as a three stage optimization. At the lowest level, a given path is evaluated for its traversability by computing the maximum speeds along the path at which the vehicle is dynamically stable. The second level consists of a parameter optimization that selects a locally optimal path in the neighborhood of an initial guess. The third and highest level of the optimization selects the initial guesses for the local optimization. The global optimization is based on a branch and bound search that prunes the initial set of all paths between the end points to a small number of candidates for the local optimization [4]. These candidates represent the most promising regions, one of which contains the global optimal path. Optimizing these paths with the local optimization yields the best path, in addition to a number of good alternatives. These paths are not necessarily the shortest, but they are traversable at the widest speed range of all paths with similar or shorter lengths, as is demonstrated in several examples in this paper.

3.1 Terrain and Path Representation

The terrain is represented by a cubic B patch, which is a parametric surface made of a mesh of cubic splines. A typical point p on a single patch in three dimensional space is a function of two parameters, v and w ,

$$p = VMRM^TW^T \quad (6)$$

where $V = [v^3, v^2, v, 1]$, $v = [0, 1]$, $W = [w^3, w^2, w, 1]$, $w = [0, 1]$ M is the 4×4 matrix specifying the type of spline used to construct the patch, and R is a 4×4 matrix of 16 control points.

The control points of the patch are generated by placing a uniform grid on the map-range data generated from stereo images taken by the on-board mast camera. The resolution of this grid is chosen economically at about half the rover size: roughly 20cm between neighboring points. This ensures that obstacles the size of the rover and larger are depicted by the B-patch. Smaller obstacles may be filtered out.

The path is represented by a smooth curve on the surface, obtained by parameterizing v and w by a single parameter u :

$$c(u) = p(v(u), w(u)) = V(u)MRM^TW^T(u) \quad (7)$$

Reducing the $v - w$ space to a line reduces the B patch to a continuous curve that is guaranteed to stay on the surface.

3.2 Vehicle Model

At top speeds of 10 - 20 cm/s, the motion planning problem for Mars Rover can be considered a kinematic problem. However, we do account for certain rover dynamics for the purpose of quantifying traversability and dynamic stability, with the premise that paths that are traversable at a wide speed range are safer than those that are not.

The vehicle is modeled as a point mass, suspended above ground at the location of the vehicle's center of mass. The height of the center of mass above ground and the width between the wheels are used to evaluate stability with respect to lateral tip over.

The external forces acting on the vehicle consist of the friction force F (the sum of all the horizontal tire forces), the normal force R (the sum of all normal tire forces) applied by ground on the vehicle in the r direction, and the gravity force.

The equation of motion of the vehicle are written in the vehicle fixed frame in terms of the tangential speed \dot{s} and the tangential acceleration \ddot{s} [4]

$$f_t = mgk_t + m\ddot{s} \quad (8)$$

$$f_q = mgk_q + m\kappa n_q \dot{s}^2 \quad (9)$$

$$R = mgk_r + m\kappa n_r \dot{s}^2 \quad (10)$$

where f_t and f_q are the components of the friction force tangent and normal to the path, k_t , k_q and k_r are the projection of the vertical unit vector, k , on the respective axis of the vehicle fixed coordinate frame, and $1/\kappa$ is the path curvature, \cdot . The moment of the friction force around the center of mass is considered later when we account for the tip over constraint.

Equations (8) to (10) are used to determine the feasible speed and acceleration for given limits on the friction and normal forces.

3.3 Dynamic Constraints

Constraints between the vehicle and ground are considered to ensure vehicle dynamic stability along the path.

Sliding Constraint

The maximum friction force is a function of the normal force and the coefficient of friction between the wheels and ground:

$$|F| \leq \mu R \quad (11)$$

Substituting (8)- (10) in (11), then solving for \ddot{s} yields constraints of the form [4]

$$-gk_t + \sqrt{\Delta} \leq \ddot{s} \leq -gk_t - \sqrt{\Delta} \quad (12)$$

where

$$\Delta = as^4 + 2bs^2 + c \geq 0 \quad (13)$$

yields constraints on the feasible vehicle speed along the path. The feasible speed range is determined by the roots of (13). Only the positive roots are of interest.

Contact Constraint

To ensure that the vehicle does not lose contact with ground on rough terrain, the normal force R applied on the vehicle should be positive. Setting $R = 0$ in (10), we obtain the maximum speed allowed by the contact constraint:

$$\dot{s} \leq \sqrt{-\frac{gk_r}{\kappa n_r}}. \quad (14)$$

where n_r is the projection of the path normal, n , on the surface normal, r . Equation (14) applies only for the cases where path curvature points opposite to the direction of the surface normal. Note that the velocity limit is infinite for a flat terrain ($n_r = 0$), and zero for a sharp vertical bump ($\kappa n_r = \infty$), as expected.

Tip-Over Constraint

The tip-over constraint is obtained by expressing the limiting condition before the vehicle is about to tip-over in terms of \dot{s} , \ddot{s} . The vehicle will not tip-over if the reaction force and the lateral friction force satisfy [6]

$$f_q^2 \leq \left(R \frac{b}{h}\right)^2 \quad (15)$$

Substituting (8) and (9) into (15) yields a constraint on \dot{s} similar to (13).

Velocity Limit Curve

Plotting the velocity limits due to the dynamic constraints along the path forms the *velocity limit curve* in the phase plane $s - \dot{s}$. It represents the upper bound for vehicle speeds for which the dynamic constraints discussed earlier are satisfied. The height of the velocity limit represents a measure of safety and traversability: a zero velocity limit implies static instability, whereas a nonzero but low velocity limit implies a stable but dangerous position along the path. Obviously, the higher the velocity limit, the wider the speed range that the vehicle can move along the path without sliding, tipping over, or flying off the ground.

3.4 Global Search and Local Optimization

The search for the optimal path follows the method presented in [4]. It combines a grid search in the position space with a local optimization to yield the global optimal path for a variety of static and dynamic cost functions, such as distance and motion time. This approach eliminates the search in the $2n$ dimensional state-space without sacrificing global optimality.

The cost function for Mars rover is computed by dividing the path length by the maximum constant speed that does not cross the velocity limits for that path. This cost function is the minimum motion time at the constant speed along the path. It quantifies the cumulative effects of path distance, terrain topography, and vehicle dynamics. It also favors regions with high velocity limits, which are traversable at the widest speed range.

The optimization starts by searching for a set of best paths along a uniform grid over the terrain, using the Dryfus algorithm. These paths are pruned by retaining the best path in each neighborhood, each representing the neighborhood of a potential local minimum. Submitting these paths to a local optimization that further minimizes the cost function yields the global optimal path in addition to a set of good alternatives. This optimization, admits paths that might go over obstacles if such a path is dynamically feasible and it is less costly than going around.

4 Comparison of Results

The two planners were tested on images obtained from the JPL Mars Yard. The images were electronically manipulated to make the terrain more challenging by adding large rocks in the central region. A monochrome version of the color image used for path planning is shown in Fig. 2.

In the absence of stereo images, the apparent rock height and size were determined from a single image based on several assumptions on camera location and geometry. The height is estimated by multiplying the apparent height by a correction factor derived from perspective transformation. Similarly the size of a rock is estimated from its apparent boundary by subjecting it to perspective transformation. The the number of pixels within the perspective corrected boundary is then found, giving the size (area) of the rock. A contour map is then constructed on the basis of location, height and size of each obstacle. The contour map of the Mars terrain (Fig. 2) is shown in Fig. 3, where darker areas correspond to higher elevations. This contour map was used by both path planner.

For the genetic planner, the 512×512 pixel image representing a 10 square meter region was divided into 32×32 cells. The number of cells can be increased for higher resolution, if required. The impedance of each cell was determined using the method described in Section 2.1. A population size of five paths was chosen, and these paths went through the genetic evolution described in Section 2. The initial intraversable paths were quickly evolved into traversable paths, and as the evolution continued these paths in turn changed into shorter ones passing through less rock concentrated

		Height				
		VL	LO	ME	HI	VH
Size	TI	VL	VL	LO	ME	VH
	SM	VL	VL	LO	ME	VH
	ME	VL	LO	LO	LO	VH
	LG	VL	LO	ME	LO	VH
	XL	VL	LO	ME	LO	VH

Fig. 1. The fuzzy rule matrix. The entries are terrain roughness

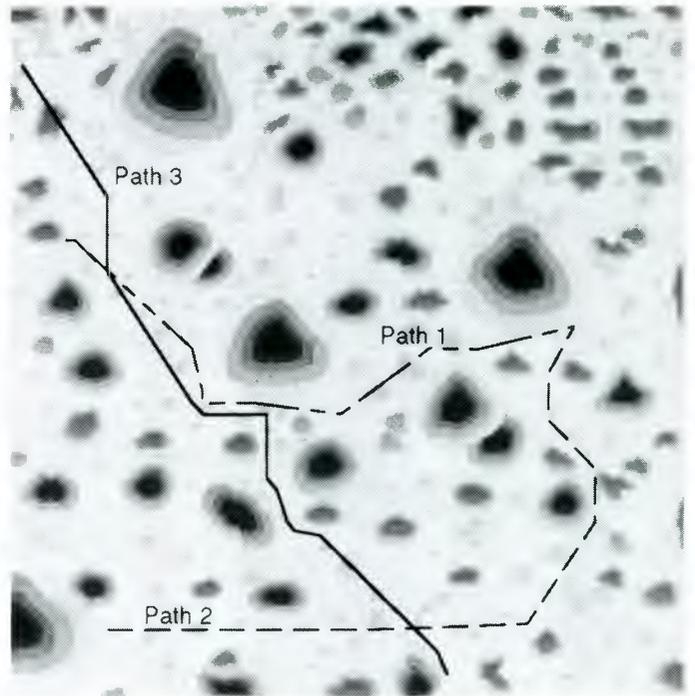


Fig. 3. Three paths found by the genetic path planner shown on the contour map (Smoothing not applied to the paths)



Fig. 2. A reconstructed Mars image

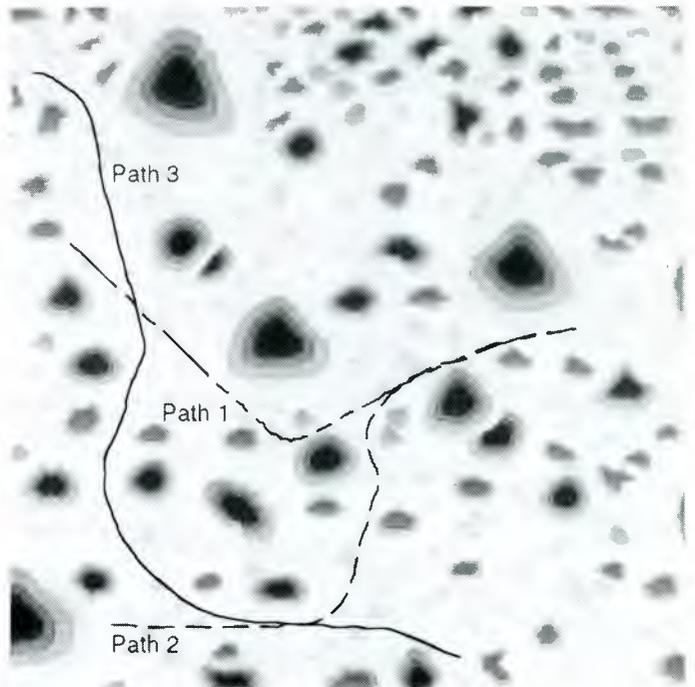


Fig. 4. Three paths found by the global optimization planner shown on contour map

areas and avoiding larger rocks. Near optimal paths were usually found after 200 to 400 iterations (generations), thus good paths were found very quickly. Figure 3 shows three paths generated by the genetic algorithm for several start and goal points located in difficult regions.

The global planner uses the contour map directly, and performs the optimization method described in Section 2. Figure 4 shows the paths found by the global planner for the same start and goal locations as those used for the genetic planner.

Several observations are now made regarding the generated paths. First, the genetic planner produces the waypoints, and in Figure 3 these waypoints are connected by straight line segments. To obtain smoother paths, these waypoints can be connected by cubic polynomials or any other suitable interpolations. It is also noted that in these runs a low weighting (α in (3)) was assigned to curvature relative to the cell impedance to obtain shorter paths. As a result a path sometimes traverses over small rocks to achieve shorter path lengths (and path impedance). However, a closer examination shows that all paths are in fact traversable by the rover (in this case NASA's Rocky 7 rover [2]). The global optimization planner produces smoother path due to using a finer grid resolution.

Even though both planners attempt to optimize their respective performance indices, they have different conceptual basis. The genetic planner employs a fuzzy description of the terrain, and attempts to come up with a path that is short and passes over reasonably smooth parts of the terrain. It delegates the local maneuvering of the rover along the planned path to the rover navigation system. Thus the rover kinematics and dynamics are only considered indirectly through terrain topology during the path planning phase. The global planner uses both terrain topology information and a simplified kinematic/dynamic rover model to achieve both path planning and navigation. As a result of the added task of taking kinematic/dynamic constraints into considerations, it is generally more complex and requires more computation compared to the genetic planner. This added complexity is justified provided that a reasonably accurate terrain topology can be constructed from the images of the terrain, and that the simplified kinematic/dynamic model can adequately represent the actual rover behavior. On the other hand, the genetic planner requires only imprecise information about the terrain but relies upon on-line hazard detection for possible local adjustments to the path. The paths produced by both planners are generally longer than the shortest paths between respective end points (Fig. 3 and 4) but they seem to pass mostly through wider corridors and hence are safer.

5 Conclusions

The path planners described in this paper share the common attribute of attempting to optimize certain performance indices. It has been shown through planning of paths for a simulated Mars terrain that both are capable of producing short paths that traverse over smooth parts of the terrain and avoid areas with large rocks. While both planners perform some form of optimization, they are conceptually different. The genetic planner requires only an approximate description of the terrain and operates on the basis of evolutionary process and stochastic search to generate a near optimal path. The global planner incorporates certain kinematics and dynamics into the planning phase, and require more knowledge about the environment and the rover. The relative simplicity of the genetic planner and the benefit of incorporating kinematic/dynamic constraints of the global planner can be combined to achieve better results. For example, the genetic planner can quickly produce a number of paths based on imprecise terrain description and the global planner can then evaluate or modify these paths to take into consideration the rover kinematic/dynamic constraints.

6 Acknowledgments

The work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

7 References

- [1] J.-C Latomb, *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [2] S. Hayati et al, "The Rocky 7 rover: A Mars science-craft prototype," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2458-2464, Albuquerque, NM, April 1997.
- [3] S. Laubach, J. Burdick and L. Matthies, "A practical autonomous path planner implemented on the Rocky 7 prototype microrover," *IEEE Int. Conf. Robotics and Automation*, 1998.
- [4] Z. Shiller and Y-R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Trans. Automation and Robotics*, pp. 241-249, vol. 2, 1991.
- [5] T. Simeon and B. Darce-Wright, "A practical motion planner for all-terrain mobile robots," *Proc. Int. Conf. Intelligent Robots and Systems*, 1993.
- [6] S. Farritor, H. Hacot, and S. Dubowsky, "Physics-based planning for planetary exploration," *IEEE Int. Conf. on Robotics and Automation*, 1998.
- [7] M. Tarokh, R. Chan and C. Song, "Path planning of rovers in rugged terrain using fuzzy logic and genetic algorithm," *Technical Report, Robotics and Intelligent Systems Laboratory, San Diego State University*, 1999.

TRAVERSABILITY INDEX: A NEW CONCEPT FOR PLANETARY ROVERS

Homayoun Seraji

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109, USA

Abstract

Traversability Index is introduced in this paper as a new and simple measure for traversability of natural terrains by mobile robots. This index is developed using the framework of fuzzy logic, and is expressed by linguistic fuzzy sets that quantify the suitability of the terrain for traversal based on its physical properties, such as slope and roughness. The Traversability Index is used for classifying natural terrains, and provides a simple means for incorporating the terrain quality data (out to about 10 meters) into the rover navigation strategy. A set of fuzzy navigation rules is developed using the Traversability Index to guide the rover toward the safest and the most traversable terrain. In addition, another set of fuzzy rules is developed to drive the rover from its initial position to a user-specified goal position. These two rule sets are integrated in a two-stage procedure for autonomous rover navigation without a priori knowledge about the environment. A computer simulation study is presented to demonstrate the capability of the rover to reach the goal safely while avoiding impassable terrains.

1 Introduction

Although considerable research has been conducted on mobile robots in recent years, the bulk of this research is focused on in-door robots operating in highly-structured, human-made environments. Typically, the environment consists of a flat, smooth, horizontal floor on which the robot moves. *Field* mobile robots, on the other hand, must traverse harsh *natural* terrains that are uneven, rough, and have

slopes. These physical properties of the terrain add a new dimension to the complexity of the robot navigation problem. Rover navigation on Martian and Lunar surfaces has been an active area of research at JPL and CMU, respectively, [see, e.g., 1-2]. The photograph of the Martian terrain shown in Figure 1 indicates that each region of the terrain offers different traversability characteristics to the Sojourner rover in the Mars Pathfinder Mission.

In this paper, a new concept called *Traversability Index* is introduced for the first time for mobile robots (rovers) operating on natural terrains. This index is expressed by linguistic fuzzy sets that represent the suitability of the terrain for traverse based on its physical properties, such as slope and roughness. The index also gives a basis for classifying natural terrains according to their ease of traverse, ranging from "highly-impassable" to "highly-passable" terrains. Using the Traversability Index, a set of fuzzy navigation rules is developed to guide the rover toward the safest and the most traversable terrain. This rule set is integrated with fuzzy rules for goal seeking to obtain an autonomous navigation strategy for a mobile robot that requires no *a priori* knowledge about the environment.

The paper is structured as follows. In Section 2, the Traversability Index is defined using the fuzzy logic framework. A set of fuzzy navigation rules based on this index is presented in Section 3. Section 4 discusses fuzzy logic rules for the rover goal seeking. The integration of the terrain traversing and goal seeking fuzzy rule sets is described in Section 5. An illustrative example is presented in Section 6 for proof-of-concept and demonstration. The paper is concluded in Section 7 with a brief review and future plans.

2 Traversability Index for Field Robots

This section establishes *Traversability Index* as a new and simple measure for traversability of natural terrains by mobile robots. This index is developed using the framework of fuzzy logic, which has been used extensively for navigation of in-door mobile robots operating in structured environments [see, e.g., 3-11]. The Traversability Index τ is expressed by linguistic fuzzy sets quantifying how traversable a particular terrain is for a given rover. Several options are available for defining the Traversability Index as a function of the terrain physical properties. In this paper, the Traversability Index τ is defined by fuzzy relations in terms of two physical variables: the terrain slope α and the terrain roughness β , where α and β are both expressed by linguistic fuzzy sets as described below.

2.1 Terrain Slope α

The terrain slope α can be measured by a stereo vision system mounted on the rover [1]. The slope α is represented by the four linguistic fuzzy sets { LOW, MEDIUM, HIGH, VERY HIGH }. The membership functions of these sets are shown in Figure 2a, where the abscissa α is the magnitude of the terrain slope and the ordinate $\mu(\alpha)$ is the degree-of-membership. Note that the slope can be either a positive quantity representing a mound or a hill, or a negative quantity representing a crater or a downward surface. Observe that precise measurement of the terrain slope is *not* needed using the fuzzy logic framework.

2.2 Terrain Roughness β

Several methods can be adopted to assess the terrain roughness β . For instance, the roughness β can be computed using a least-squares fitting algorithm based on the range map obtained by the on-board stereo vision system [1]. Alternatively, the terrain roughness β can be computed from two measurements supplied by the vision system: the rock sizes and the rock concentration (density) on the terrain. Let δ denote the "average" rock size (\approx height \times cross-sectional area) on the terrain. Then δ can be represented by the two linguistic fuzzy sets { SMALL, LARGE }. Similarly, the rock concen-

tration on the terrain is denoted by ω , and is represented by the two linguistic fuzzy sets { LOW, HIGH }. Then, the terrain roughness β can be expressed in terms of the rock size δ and the rock concentration ω using a set of simple fuzzy relations. Let β be represented by the four linguistic fuzzy sets { SMOOTH, ROUGH, BUMPY, ROCKY }, where the membership functions are shown in Figure 2b. The dependence of β on δ and ω can then be expressed intuitively by a set of four simple fuzzy rules summarized in Table 1. Notice that precise measurements of the average rock size δ and the rock concentration ω are *not* needed, because of the multi-valued nature of the linguistic fuzzy sets used to describe them.

2.3 Traversability Index τ

The Traversability Index τ is defined by a set of fuzzy relations in terms of the slope α and the roughness β of the terrain. In the framework of fuzzy logic, the *Cartesian product* is used to represent fuzzy functional relations [12]. Let $A = \{A_1, A_2, A_3, A_4\}$ and $B = \{B_1, B_2, B_3, B_4\}$ represent, respectively, the fuzzy sets defined on the input variables α and β . The Cartesian product of these input fuzzy sets is the output fuzzy set $T = A \times B$ with the membership function defined by $\mu(\tau) = \mu(\alpha) * \mu(\beta)$, where $*$ denotes one form of the fuzzy set union ("and") operation and T is the fuzzy set of the output variable τ . The Traversability Index τ is represented by the four linguistic fuzzy sets $T = \{ \text{POOR, LOW, MEDIUM, HIGH} \}$, with the membership functions shown in Figure 2c. In the context of the Traversability Index τ , the Cartesian product functional relation can be represented by a set of sixteen simple fuzzy rules summarized in Table 2. Based on these rules, it is seen that the Traversability Index of the terrain τ is defined to be POOR when the terrain slope α is VERY HIGH or the terrain roughness β is ROCKY (see fourth row and column in Table 2). This implies that terrains with very high slope or with rocky surfaces are considered to be highly impassable and must be avoided. When these two extreme cases are excluded, the Traversability Index τ falls in the range of possible values spanned by the four fuzzy sets POOR through HIGH, depending on the slope and roughness of the terrain (see rows 1-3 and columns 1-3 in Table 2). Note that the Traversability Index varies with the size, drive mechanism, and rock climbing capability of the rover, and therefore the

above definitions apply to the *particular* rover under consideration with the given mechanical design.

The fuzzy logic process for computation of the Traversability Index τ consists of the following stages. The terrain roughness β is first obtained by fuzzy inference from the on-board measurements of the terrain rock size and concentration δ and ω . The crisp values of the terrain slope α and the terrain roughness β are then passed through the “fuzzification” stage to find the degrees-of-membership in their corresponding fuzzy sets. This data is then used to evaluate the Traversability Index based on the fuzzy rules given in Table 2. This stage, which is referred to as “inference” in fuzzy logic, produces the activation levels or strengths of the rules that are “fired” using the max-min fuzzy inference method [12]. This information is then passed to the “defuzzification” stage where the crisp value of the Traversability Index τ is computed using the centroid defuzzification method [12]. Note that the fuzzy logic framework used for computation of τ only requires reasonable estimates of the terrain quality data α and β obtainable from inexpensive sensors that are expected to be imprecise. This method does *not* need expensive precision sensors that also require extensive processing of sensory data for precise interpretations.

2.4 Terrain Classification Based on τ

The Traversability Index provides a basis for classifying natural terrains according to their ease of traverse by the rover. Using the fuzzy linguistic description of the Traversability Index τ , different regions of the terrain can be classified into four categories based on their value of τ . The four linguistic fuzzy sets for τ can be interpreted as follows:

- POOR $\tau \rightarrow$ HIGHLY-IMPASSABLE TERRAIN.
- LOW $\tau \rightarrow$ IMPASSABLE TERRAIN.
- MEDIUM $\tau \rightarrow$ PASSABLE TERRAIN.
- HIGH $\tau \rightarrow$ HIGHLY-PASSABLE TERRAIN.

3 Navigation Rules Based on Traversability Index

In this section, the Traversability Index defined in Section 2 is used to develop simple rules for determination of the rover heading and speed on a planetary surface. In other words, the Traversability Index is used to navigate the rover toward the safest and the most traversable terrain. This index provides a simple means for incorporating the terrain quality data (out to about 10 meters) into the rover navigation strategy. The control variables of the rover are the translational speed v and the heading angle change $\Delta\theta$ per control cycle. We shall now discuss the fuzzy rules for determination of the rover heading change and the rover speed based on the Traversability Index.

3.1 Turn Rules

It is assumed that the rover can only move in the forward direction (i.e., reverse motion is not allowed). The terrain in front of the rover is partitioned into five regions, namely: front, front-right, front-left, right, and left of the rover at a distance up to r from the rover, where r defines the radius of the sensing envelope and is typically 10 meters [1]. The “front” refers to the region the rover is heading toward at present, “front-right” and “front-left” regions are sectors at $\pm 45^\circ$ relative to the rover heading, and “right” and “left” regions are sectors between $\pm 45^\circ$ and $\pm 90^\circ$ relative to the heading. The terrain traversability data is assumed to be available for the five regions. Therefore, at any instant, five crisp Traversability Indices are computed for the five possible traversable regions described above, namely: τ_f , τ_{fr} , τ_{fl} , τ_r and τ_l . At this stage, the on-board software compares these five quantities and selects the one with the highest value τ_{max} , that is, the most traversable region is chosen. When the situation has a non-unique solution, i.e., there is more than one region with the highest τ , then the one which is closest to the front region is chosen so that unnecessary rotations are avoided. The five turn rules are as follows:

- IF $\tau_{max} = \tau_l$, THEN $\Delta\theta$ is HARD-LEFT.
- IF $\tau_{max} = \tau_{fl}$, THEN $\Delta\theta$ is LEFT.
- IF $\tau_{max} = \tau_f$, THEN $\Delta\theta$ is ON-COURSE.

- IF $\tau_{max} = \tau_{fr}$, THEN $\Delta\theta$ is RIGHT.
- IF $\tau_{max} = \tau_r$, THEN $\Delta\theta$ is HARD-RIGHT.

where { HARD-LEFT, LEFT, ON-COURSE, RIGHT, HARD-RIGHT } represent the five linguistic fuzzy sets of the rover heading change $\Delta\theta$, with the membership functions shown in Figure 3a.

3.2 Move Rules

Once the region to be traversed is chosen based on the relative values of τ , the rover speed v can be determined based on the value τ_{max} of the Traversability Index τ in the chosen region. This determination is formulated as a set of four simple fuzzy move rules as follows:

- IF τ_{max} is POOR, THEN v is STOP.
- IF τ_{max} is LOW, THEN v is SLOW.
- IF τ_{max} is MEDIUM, THEN v is MODERATE.
- IF τ_{max} is HIGH, THEN v is FAST.

where {STOP, SLOW, MODERATE, FAST} represent the four linguistic fuzzy sets associated with the rover speed v , with the membership functions shown in Figure 3b.

4 Fuzzy Rules for Goal Seeking

In this section, we present fuzzy rules for navigation of the rover from its current position to the desired goal position. Two sets of rules are developed for the rover speed v and the rover heading change $\Delta\theta$. The basic idea behind the navigation rules is that the rover tries to: (1) approach the goal with a speed proportional to the distance between the current position and the goal position, defined as the "position error" d , (2) rotate toward the goal position by nullifying the "heading error" ϕ , which is the angle by which the rover needs to turn to face the goal directly.

We shall now present the fuzzy navigation rules for goal seeking in the following subsections.

4.1 Turn Rules

The rover heading change $\Delta\theta$ depends on the heading error ϕ , where the angles are defined to be positive in the clockwise direction. The heading error ϕ has the linguistic fuzzy sets { GOAL-FAR LEFT, GOAL-LEFT, HEAD-ON, GOAL-RIGHT, GOAL-FAR RIGHT }, with the membership functions depicted in Figure 4a. The fuzzy rules for the rover turn are as follows:

- IF ϕ is GOAL-FAR LEFT, THEN $\Delta\theta$ is HARD-LEFT.
- IF ϕ is GOAL-LEFT, THEN $\Delta\theta$ is LEFT.
- IF ϕ is HEAD-ON, THEN $\Delta\theta$ is ON-COURSE.
- IF ϕ is GOAL-RIGHT, THEN $\Delta\theta$ is RIGHT.
- IF ϕ is GOAL-FAR RIGHT, THEN $\Delta\theta$ is HARD-RIGHT.

It is seen that the rover heading change $\Delta\theta$ is only a function of the heading error ϕ , and is independent of the rover speed v .

4.2 Move Rules

The rover speed v is generated by the position error d . The goal distance or position error d has the linguistic fuzzy sets { VERY NEAR, NEAR, FAR, VERY FAR }, with the membership functions depicted in Figure 4b. The fuzzy rules for the rover speed are as follows:

- IF d is VERY NEAR, THEN v is STOP.
- IF d is NEAR, THEN v is SLOW.
- IF d is FAR, THEN v is MODERATE.
- IF d is VERY FAR, THEN v is FAST.

It is seen that the rover speed v is only a function of the goal distance d , and is independent of the heading error ϕ .

5 Integration of Traverse and Seek Behaviors

In the preceding two sections, fuzzy rule sets are given for the two *independent* behaviors of terrain

traversing and goal seeking. The rule set for each behavior is concerned solely with achieving its particular objectives, disregarding the constraints imposed by the other behavior. In this section, we discuss the integration of these two behaviors to obtain an autonomous navigation strategy for the rover. A two-stage procedure is proposed for autonomous rover navigation without *a priori* knowledge about the environment. In the first stage, the traverse-terrain and seek-goal rule sets make their individual, independent recommendations for rover speed and heading change commands. In the second stage, these recommendations are integrated by using appropriate weighting factors to generate the combined, coordinated recommendation for the rover navigation based on the rover status.

Consider the rover navigation procedure shown in the block diagram of Figure 5. Each of the two behaviors, traverse-terrain and seek-goal, generates a set of independent recommendations for v and $\Delta\theta$ based on its own objectives. These sets of recommendations $\{v^t\}, \{\Delta\theta^t\}$ and $\{v^s\}, \{\Delta\theta^s\}$ are then “weighted” by the crisp weighting factors t_w and s_w assigned to the outputs of the traverse-terrain and seek-goal behaviors, respectively. In other words, the final recommendations \bar{v} and $\bar{\Delta\theta}$ result from defuzzification of the weighted aggregated outputs of the traverse-terrain and seek-goal rule sets. The weighting factors t_w and s_w represent the strengths by which the traverse-terrain and seek-goal recommendations are taken into account. These factors are represented by the linguistic fuzzy sets {NOMINAL, HIGH }, whose triangular membership functions have the central values of 1 and 10, respectively. Within this context, the traverse and seek weighting factors are assumed to have the fuzzy NOMINAL value except in the following extreme cases:

- IF τ is POOR OR τ is LOW, THEN t_w is HIGH.
- IF d is VERY NEAR, THEN s_w is HIGH.

The first rule implies that when the terrain is not easily passable by the rover, the recommendation of the traverse-terrain rule set is assigned a HIGH weighting factor with the central value 10 relative to the seek-goal recommendation which has the NOMINAL weighting factor with the central value 1. The second rule suggests that when the goal position is almost reached, the seek-goal recommendation takes

on the HIGH weighting factor relative to the NOMINAL weighting factor for the traverse-terrain recommendation. Excluding these two extreme cases, the traverse-terrain and seek-goal recommendations for v and $\Delta\theta$ are combined using equal weightings of unity to obtain the final recommendations for the rover speed and heading change \bar{v} and $\bar{\Delta\theta}$ that are passed to the rover for execution.

6 Illustrative Example

In this section, a computer graphical simulation study is presented to demonstrate fuzzy logic-based rover navigation using the traverse-terrain and seek-goal rule sets developed in this paper. The simulations are performed using the Rover Graphical Simulator (RGS) developed at JPL. This simulator is written in Java and is platform-independent, running on both PC and Unix machines. The RGS provides an essential tool for visualization of the rover reasoning and decision-making capabilities using the fuzzy logic navigation rule sets. It depicts a terrain composed of regions with different grades of traversability, together with the initial and goal rover positions. The rule sets for the two behaviors, namely, traverse-terrain and seek-goal, are integrated in the RGS. A simple Graphical User Interface (GUI) is provided to issue rover motion commands and display the rover movements graphically under the fuzzy navigation rules.

In this study, there are three impassable regions between the initial and the goal positions of the rover as depicted by dark circles in Figure 6. The rover is required to drive to the goal position while avoiding the three regions. These regions are a crater with POOR Traversability Index, a high-slope region with POOR Traversability Index, and an area of high rock density with LOW Traversability Index. The remaining regions of the terrain have HIGH Traversability Index. The path traversed by the rover under the fuzzy traverse-terrain and seek-goal rule sets is shown by the dotted line in Figure 6. It is seen that the test is successfully completed with the rover reaching the goal safely while avoiding the three impassable terrains.

7 Conclusions

The new concept of Traversability Index is introduced in this paper for mobile robots operating on natural terrains. Fuzzy logic framework is used to define the Traversability Index in terms of the physical properties of the terrain, such as slope and roughness. This index is used to classify natural terrains according to their suitability for traverse by the rover. A set of fuzzy navigation rules based on this concept is developed to guide the robot toward the most traversable terrain. These rules are then integrated with another set of fuzzy rules for goal seeking to obtain an autonomous navigation strategy for a field rover.

Fuzzy logic provides a natural framework for formulating and expressing the attributes of the human navigation expertise and for emulating this expertise for field mobile robots. The use of linguistic fuzzy sets is simple, intuitive, and akin to the human reasoning and decision-making processes. A novel feature of the proposed approach is the utilization of the *regional* traversability information obtained from the terrain data for rover navigation. This information augments the *local* information obtained from en-route obstacles to provide a comprehensive approach for autonomous rover navigation that requires no *a priori* knowledge about the environment. Future research is focused on implementation and verification of the proposed approach on a commercial mobile robot designed for out-door operations.

8 Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The development of the Rover Graphical Simulator used in Section 6 by Bruce Bon is gratefully acknowledged.

9 References

1. D. B. Gennery: "Traversability analysis and path planning for a planetary rover", To appear in *Journal of Autonomous Robots*, 1999.
2. R. Simmons, L. Henriksen, L. Chrisman, and G. Whelan: "Obstacle avoidance and safeguarding for a Lunar rover", *Proc. AIAA Forum on Advanced Developments in Space Robotics*, Madison, 1996.
3. A. Saffiotti: "The uses of fuzzy logic in autonomous robot navigation", *Journal of Soft Computing*, vol. 1, no. 4, 1997.
4. R. A. Hogle and P. P. Bonissone: "A fuzzy algorithm for path selection in autonomous vehicle navigation", *Proc. IEEE Conf. on Decision and Control (CDC)*, pp. 898-900, Las Vegas, 1994.
5. R. N. Lea, Y. Jani, M. G. Murphy, and M. Togai: "Design and performance of a fuzzy logic based vehicle controller for autonomous collision avoidance", *Proc. Fuzzy Neural Systems and Vehicle Applications Conf.*, Tokyo (Japan), 1991.
6. W. Li: "Fuzzy logic-based perception-action behavior control of a mobile robot in uncertain environments", *Proc. IEEE Intern. Conf. on Fuzzy Systems*, pp. 1626-1631, Orlando, 1994.
7. W. Blochl: "Fuzzy control in real-time for vision guided autonomous mobile robots", *Proc. Austrian Conf. on Fuzzy Logic in AI*, pp. 114-125, 1993.
8. S. G. Goodridge, M. G. Kay, and R. C. Luo: "Multi-layered fuzzy behavior fusion for reactive control of an autonomous mobile robot", *Proc. IEEE Intern. Conf. on Fuzzy Systems*, pp. 579-584, Barcelona (Spain), 1997.
9. P. Reignier: "Fuzzy logic techniques for mobile robot obstacle avoidance", *Journal of Robotics and Autonomous Systems*, pp. 143-153, 1994.
10. H. R. Beom and H. S. Cho: "A sensor-based obstacle avoidance controller for a mobile robot using fuzzy logic and neural network", *Proc. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1470-1475, 1992.
11. E. W. Tunstel, Jr.: "Adaptive Hierarchy of Distributed Fuzzy Control: Application to behavior control of rovers", Ph. D. Thesis, Department of Electrical and Computer Engineering, University of New Mexico, 1996.
12. K. M. Passino and S. Yurkovich: *Fuzzy Control*, Addison-Wesley Publishing Company, 1997.

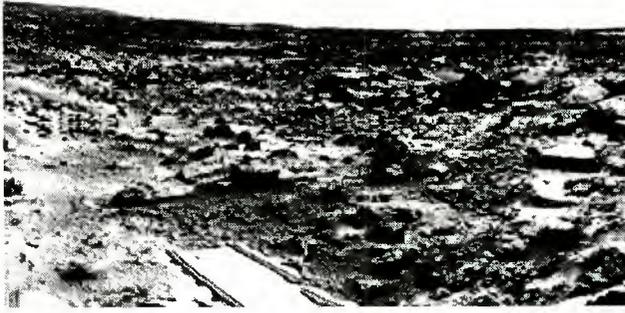


Figure 1. Photograph of the Martian terrain

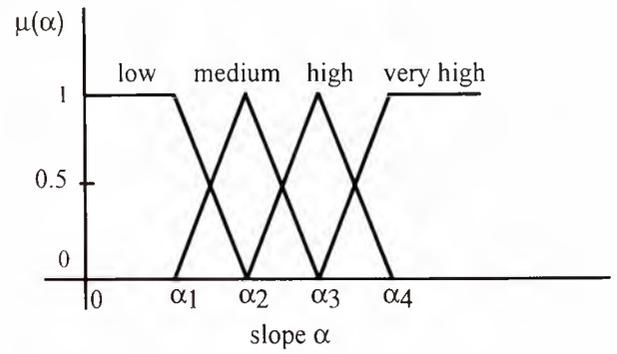


Figure 2a. Membership functions for terrain slope α

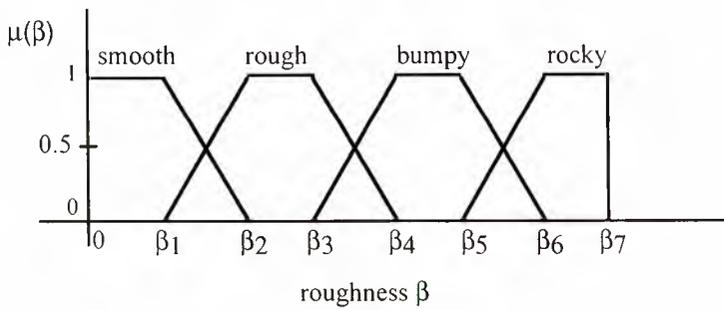


Figure 2b. Membership functions for terrain roughness β

		rock concentration ω	
		low	high
rock size δ	small	smooth	rough
	large	bumpy	rocky

Table 1. Rule set for terrain roughness β

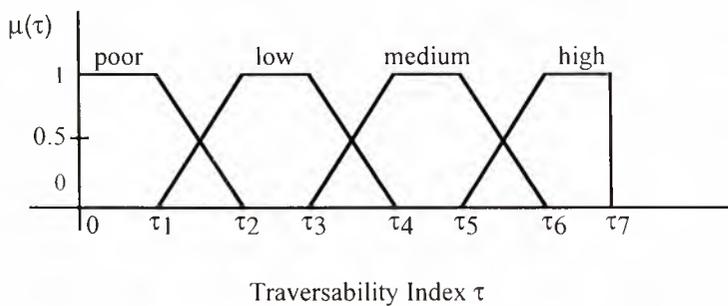


Figure 2c. Membership functions for Traversability Index τ

		terrain roughness β			
		smooth	rough	bumpy	rocky
terrain slope α	low	high	medium	low	poor
	medium	medium	medium	low	poor
	high	low	low	poor	poor
	very high	poor	poor	poor	poor

Table 2. Rule set for Traversability Index τ

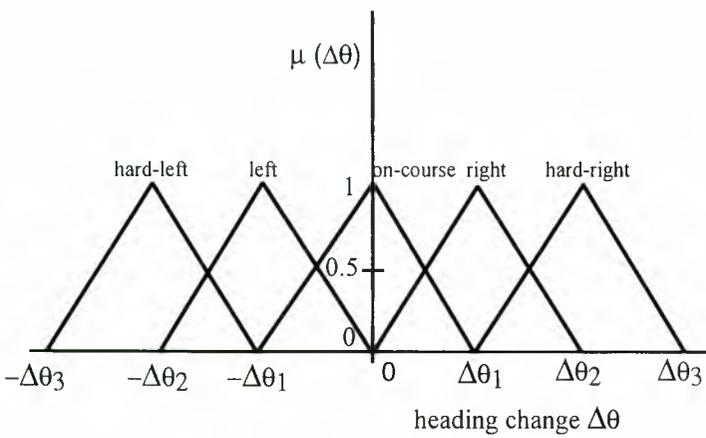


Figure 3a. Membership functions for heading change $\Delta\theta$

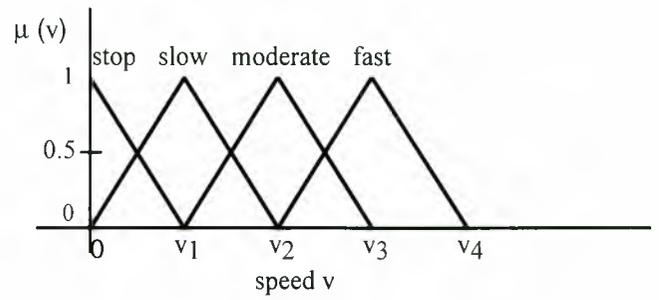


Figure 3b. Membership functions for speed v

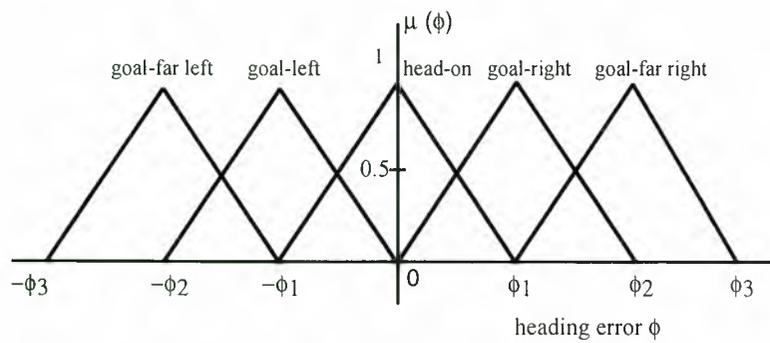


Figure 4a. Membership functions for heading error ϕ

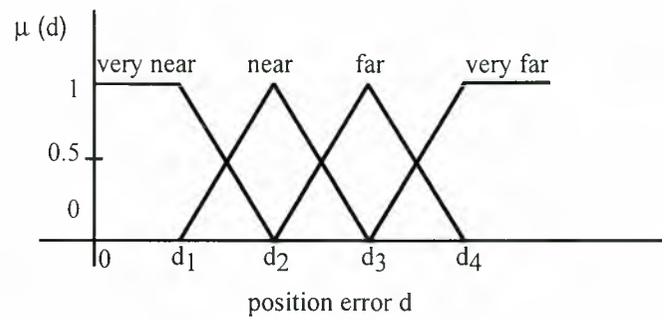


Figure 4b. Membership functions for position error d

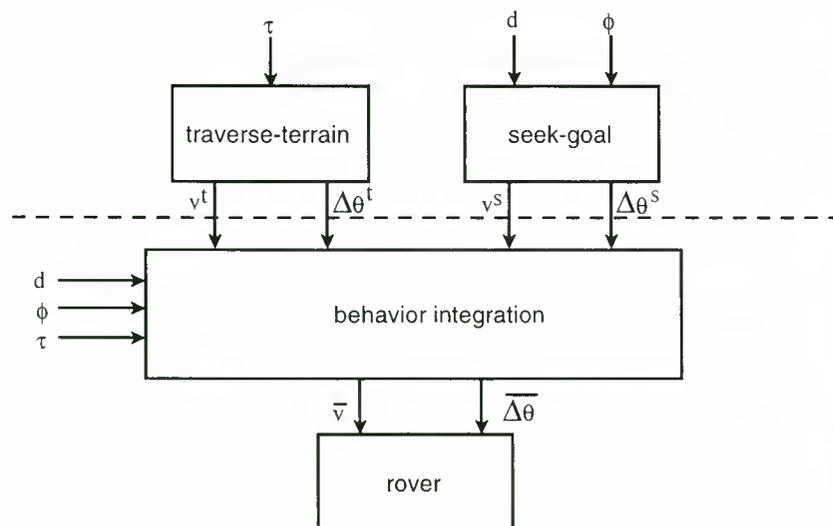


Figure 5. Two-stage rover navigation procedure

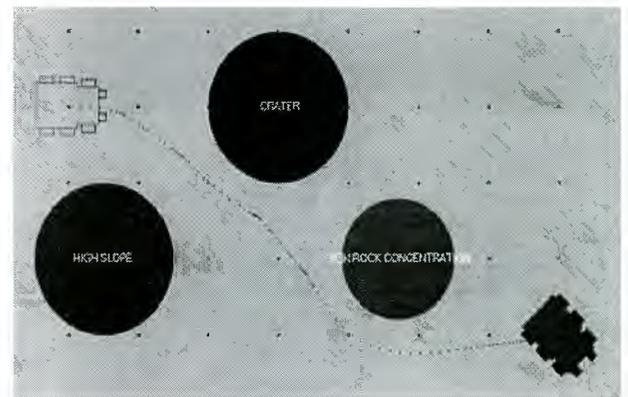


Figure 6. Simulation of the fuzzy navigation rules

Increased Flexibility and Robustness of Mars Rovers

John L. Bresina

Phone: 650.604.3365

bresina@ptolemy.arc.nasa.gov

Keith Golden

Phone: 650.604.3585

kgolden@ptolemy.arc.nasa.gov

David E. Smith

Phone: 650.604.4383

de2smith@ptolemy.arc.nasa.gov

Rich Washington*

Phone: 650.604.1140

richw@ptolemy.arc.nasa.gov

Fax: 650.604.3594

NASA Ames Research Center

Mail Stop: 269-2

Moffett Field, CA 94035-1000 USA

ABSTRACT

Our overall objective is to improve the productivity of Mars rovers by increasing the flexibility and robustness of their autonomous behavior. To achieve this objective, we set out to increase the on-board autonomy of rovers and enable commanding at a higher level with a more flexible command language. In February, 1999, we demonstrated some of our rover autonomy technologies as part of a Marsokhod rover field test that simulated aspects of the Mars '01-'05 missions. In this paper, we present the commanding language employed in this field test, called the *Contingent Rover Language (CRL)*, and describe the ground tools and on-board executive capabilities that were developed to generate and execute CRL plans. A key feature of CRL is that it enables the encoding of contingent plans specifying what to do if a failure occurs, as well as what to do if a serendipitous science opportunity arises.

1. INTRODUCTION

Traditionally, spacecraft commanding is accomplished *via* rigid time-stamped sequences of primitive operations. If anything goes wrong during execution, built-in routines attempt to safe the spacecraft and await further instructions from Earth. As NASA missions become more challenging, more sophisticated spacecraft are required, as are more advanced means of commanding them. As a case in point, the Mars Pathfinder's Microrover Flight Experiment made significant advances over previous robotic missions. Sojourner had to operate in an uncertain environment and respond more autonomously to sensor input.

With respect to the Sojourner microrover, for the purposes of this paper, we focus on the issues of commanding and contingency; for more details, see Mishkin, *et al.*, 1998. Like traditional spacecraft, Sojourner was commanded with time-stamped sequences,

and the commands tended to be primitive operations. However, there were operations that were specified at a higher level; the primary example is the "Go to Waypoint" command, which implemented autonomous navigation to a specified coordinate.

A command sequence typically specified the activities for one sol (Martian day) plus "runout" commands in case the next sol's sequence was delayed. These sequences contained no explicit contingencies; however, contingency responses to certain drastic scenarios were pre-loaded on both the Pathfinder lander and rover. The "Backup Mission Load" was to be used in the event of a communication loss from Earth to the lander, and the "Contingency Mission Load" was to be used in the event of a communication loss from the lander to the rover.

Our aim is to continue in the technology direction set by the Pathfinder mission and increase the robustness of autonomous rovers by enabling a higher level of commanding with a more flexible and contingent language. The intended benefit is to increase rover productivity without a decrease in safety. Our strategy is to make incremental advancements in this direction so as to maintain relevance to currently planned Mars rover missions and to eventually enable missions beyond the current capabilities of flight rovers.

With planetary rovers, there is uncertainty about many aspects of sequence execution: exactly how long operations will take, how much power will be consumed, and how much data storage will be needed. Furthermore, there is uncertainty about environmental factors that influence such things as rate of battery charging or which scientific tasks are possible. In order to allow for this uncertainty, sequences are typically based on worst-case estimates and contain fail-safe checks. If an operation takes less time than expected, the rover waits for the next time-stamped operation. If operations take longer than expected, they may be terminated before completion. In some cases,

*NASA contractor with Caelum Research Corporation.

all non-essential operations may be halted until a new command plan is received. These situations result in unnecessary delays and lost science opportunities.

Our first steps in this effort involved designing a new commanding language, called the *Contingent Rover Language (CRL)*, described in the next section. A key feature of CRL is that it enables the encoding of contingent plans specifying what to do if a failure occurs, as well as what to do if a serendipitous science opportunity arises. For example, a CRL plan could specify the following contingent rover behavior: when a failure occurs, execute a contingency plan to recover from the failure; if none is available, then execute a contingency plan to acquire additional data to support failure diagnosis and recovery by the ground operations team. We also implemented the ground tools and on-board executive capabilities needed to generate and execute CRL plans, described in the following sections. For further discussion of the ground and on-board techniques, see [Washington, *et al.*, 1999].

In February, 1999, we had an opportunity to demonstrate some of these rover autonomy technologies as part of a field test that was meant to simulate the main objectives of the Mars '01-'05 missions. During this exercise, both advanced rover technologies and science investigation strategies for planetary surface operations were demonstrated. In this paper, we primarily report the aspects of this field test relevant to rover commanding *via* CRL plans.

2. CONTINGENT ROVER LANGUAGE

In this section, we describe a new commanding language, called the *Contingent Rover Language (CRL)*. CRL was designed to serve as the communication medium between the ground operations team and a planetary rover, under the following design criteria.

- **Contingency and Flexibility.** The language should express the constructs that are necessary to achieve scientific goals. In particular, the language should express a variety of temporal and state constraints, and it should support conditional execution of contingency plans based on the execution context.
- **Simplicity.** The language should be simple enough that an automatic, mixed-initiative planning system can provide effective support for plan generation. The intended benefit is to reduce effort on operations staff and to improve the quality of the command plans. Similarly, the language should not be so complex that verification of command plans is impractical. Safety is of paramount importance in space missions, given the high cost of mission failure, so guarantees on execution correctness are critical for any deployed system.
- **Compatibility.** The language should be compatible with existing command languages; *i.e.*, it should allow ground operators to control a rover in the same way that they do now. In particular, it should be possible to easily specify a time-stamped command sequence. The additional capabilities should be available for incremental incorporation as needed to achieve mission goals.

A CRL command plan contains a nominal sequence (possibly) with a set of contingent branches, as well as a library of *alternate plans*. The alternate plans can be thought of as *global contingencies*, whereas the contingent branches are *local contingencies* at specific points in the command plan.

If there are no deviations from the *a priori* execution expectations, then the rover's behavior is governed by the nominal sequence. The contingent branches specify alternative courses of action in response to expectation deviations. Within any contingent branch there may be further contingent branches; hence, the plan is a tree of alternative courses of action.

The alternate plans are not attached to particular points in the command plan; rather, they can be used throughout plan execution, whenever their eligibility conditions are satisfied. When eligible, each alternate plan can either replace the rest of the current plan or be inserted before the rest of the current plan.

Consistent with our *compatibility criterion*, CRL can be used to encode the type of sequences used in the Mars Pathfinder mission, including both the daily uplink sequences as well as the Backup Mission Load and Contingency Mission Load; these loads would be encoded as alternate plans.

Due to our *simplicity criterion*, CRL does not include any control constructs for looping. The design decision we made is that when control loops are needed for execution robustness, they should be embedded within a high-level CRL command. An example of a high-level, robust command with embedded control loops is the "Visual Servo" command, which is somewhat similar to Sojourner's "Go to Waypoint" command. The Visual Servo command, which was used in the 1999 Marsokhod Field Test, implemented autonomous navigation to a specified coordinate *via* visual tracking of a target at that coordinate [Wettergreen, Thomas, and Bualat 1997].

Next, we describe the representations used in CRL. The basic data type in CRL is a *node*. Each node has associated with it a set of conditions that must be satisfied for successful execution; the following are the condition types.

- *start-conditions*: The set of conditions that must be true for the node to begin execution. Conditions can include information about the internal state of the rover (*e.g.*, wheel current), external state (*e.g.*, location), and time windows.
- *wait-for-conditions*: A subset of start-conditions for which the rover will wait until they become true. Other conditions will fail without waiting. Some conditions are automatically waited for whether or not that is specified explicitly; *e.g.*, a constraint on when an action can start executing.
- *maintain-conditions*: A list of conditions that must be true throughout node execution
- *end-conditions*: A list of conditions that must be true at the end of node execution, to verify that an action had the intended effects. Constraints on action duration can be included here.

The conditions can contain variables to be bound during constraint checking; these bindings are used to spe-

cialize the plan according to the execution-time context. The rich expressiveness of temporal and other state constraints on the plan supports effective specification of science goals and safety policies, as well as providing increased flexibility during execution. For example, rather than time-stamps, each action can have a start time interval (and an end time interval).

A node also includes information regarding the expected utility of executing the rest of the plan, as well as information regarding how to react to execution failures: execution may continue to the next node or abort.

CRL has three node subtypes: *block*, *task*, and *branch*; a command plan is defined to be a node, typically of subtype *block*. A *block* represents a sequence of nodes over which there may be shared state conditions. A *task* represents an action to execute. A task also specifies what action to perform if the task is interrupted due to execution failure. In addition, a task specifies a relative priority and expectations about resource and time usage. A *branch* represents a mutually exclusive choice point in the command plan. Each of the alternative execution paths is represented by an *option*.

An *option* is not a node subtype but a separate data type that has one subtype: *alternate plan*. Options and alternate plans specify the conditions under which they are eligible for execution and the node (typically of subtype *block*) to execute. In addition to the eligibility conditions, an alternate plan specifies when to check its eligibility: (i) whenever a failure occurs, (ii) whenever a node finishes execution, or (iii) periodically throughout plan execution. As mentioned earlier, when an alternate plan is selected for execution, it can either be inserted before the command plan suffix or it can replace the suffix.

3. CONDITIONAL PLAN EXECUTION

In this section, we describe the version of the on-board executive that was employed in the 1999 Marsokhod Field Test. The conditional executive (CX) is responsible for interpreting the command plan uplinked from ground control, monitoring plan execution, and selecting contingency plans when warranted. CX interacts with the rover control system (RC) and with the Mode Identification system (MI), which performs monitoring and fault diagnosis (described in the next section).

CX starts by executing the nominal sequence of the command plan. At each point in time, CX may have to choose among different courses of action defined by the eligible alternate plans and, if at a branch point, the eligible branch options. CX chooses the course of action with the highest estimated expected utility.

CX receives state information from the Mode Identification system (MI). It uses this information to check the various types of state conditions (in nodes), as well as to check the eligibility conditions of the alternate plans. The ability to branch on any state condition provides the plan writer with a powerful language for specifying rover behavior.

When a failure occurs, CX responds as dictated by the node, either continuing to the next node or aborting the executing plan and checking for eligible alternate

plans. In the case that no alternate plans apply, CX aborts the plan and awaits new instructions.

CX communicates with the rover control system (RC) using a datagram model of communication. This communication model allows RC to execute its real-time control loops without blocking on communication, but it carries with it a risk of lost packets. Hence, the communication protocol between CX and RC must be robust to this possibility.

RC broadcasts state and command status information on a continual, periodic basis (currently 10 times/sec). The command status information indicates whether a command is currently executing or terminated; for the latter, success or failure is also indicated.

CX sends out a single packet to initiate action along with a unique command identification. CX then waits for confirmation that RC has received the packet, indicated by seeing a command status (associated with the ID) of executing or terminated. If no such message is received within the time limit, CX will resend the packet. There is a maximum number of command resends that are allowed before causing execution failure. RC ignores the receipt of duplicate command IDs that might arise from the asynchronous communication.

4. MODE IDENTIFICATION

Health maintenance is an important issue for rovers; additionally, in order to support the execution of contingent plans, the executive must have an assessment of the current rover state. The traditional approach for fault detection is to monitor the values of particular sensors and trigger an alarm if a sensor value ever exceeds a given threshold. For example, if the product of current and time ever gets too large (*i.e.*, there is a high current over an extended interval of time), that may indicate a motor stall or other malfunction.

Such a simple mechanism can be useful, but does not easily scale when faults cannot be determined by looking at one or two sensors, or when multiple faults can occur simultaneously. For example, if an ammeter in a motor is failed, then wheel current cannot be used to determine whether the motor has stalled. However, if the encoder (which measures motor position) indicates that the motor is not turning when it should be, that *could* indicate a motor stall. It could also indicate an encoder failure. If other sensors are available, such as accelerometers, cameras, compass or GPS, these could then be used to disambiguate between the two possible failures. Such reasoning is very difficult using the approach discussed above.

Qualitative model-based diagnosis has been successfully applied in such domains, using a model of the system's normal behavior, and optional models of faulty behavior, to produce robust, reliable diagnoses based on all the sensor data, even in the presence of multiple failures. This approach is used in the MIR (Mode-Identification and Reconfiguration) component of the Remote Agent, which flew on board the Deep Space 1 spacecraft [Bernard *et al.*, 1998]. Thus, we decided to use the same system to do mode identification in our architecture. There are many advantages to this approach, which we outline below. However, we also found that due to differences between spacecraft and

rovers, some of the assumptions and design decisions used in MIR are inappropriate for rovers. In the section on the field test experience, we discuss these problems and propose some solutions for them.

The Mode Identification (MI) component of the on-board architecture eavesdrops on commands sent by CX to the rover. As each command is executed, MI receives observations from low-level monitors, which extract qualitative information from the rover sensors. For example, a current monitor may map the continuous-valued current into the set of qualitative values {low, nominal, high}. MI is informed whenever the qualitative value returned by a monitor changes. Based on monitor inputs, the commands executed on the rover, and a declarative model of the rover, MI infers the most likely current state. MI also provides a layer of abstraction to the executive, allowing plans to be specified in terms of component modes, rather than in terms of low-level sensor values.

The behavior of each state of a component is expressed using qualitative, abstract, modular models [Weld and de Kleer, 1990; Williams and de Kleer, 1991], which describe qualities of the rover's structure or behavior without the detail needed for precise numerical prediction. Such models are much easier to acquire and verify than quantitative engineering models, and are easier to reuse. For example, although the Marsokhod has six wheels, each containing a motor, only one wheel module is needed.

While such models cannot specify how far to the left the rover will drift if the motor has failed in one of its left wheels, they can be used to identify the source of failure, given the available sensor data. Such inferences are robust, since small changes in the underlying parameters do not generally affect the high-level behavior of the rover. In addition, abstract models can be reduced to a set of clauses in propositional logic, allowing behavior prediction to use unit propagation, a restricted and very efficient inference procedure.

5. COMMAND PLAN GENERATION

In this section, we discuss the ground tools developed to support the generation of CRL command plans. The process begins with the specification of science goals. CRL was designed to encode not only command plans but also goals. For the 1999 Marsokhod Field Test, a powerful set of intelligent user interface tools was used to support science planning and goal specification. The capabilities provided include generation, display, and manipulation of 3D photorealistic VR models of the rover and its environment; this VR user interface could be used to generate science goals. A separate form-based user interface could also be used to generate and edit CRL goals as well as CRL command plans. The user interface tools also provided the capability to generate CRL command plans with the support of a mixed-initiative, contingent planner/scheduler, which we refer to as *CPS*. For more details on these user interface tools, see Blackmon, *et al.*, 1999 (in this volume).

A typical field test planning cycle proceeded as follows.

1. The scientists provided a set of high-level tasks to be performed on the next simulated sol.
2. Based on this information, we developed a set of high-level CRL tasks using the VR environment and the form-based interface. The VR environment was used for the following tasks: (i) to select the best route for drive operations; (ii) to help compute angles and distances to targets; and (iii) to envision possible obstructions and illumination for image and spectrometer commands.
3. The resulting set of high-level CRL tasks was then passed from the form-based interface to CPS to be recursively decomposed into lower-level tasks and sequences of rover operations. Some decompositions included checks and contingent branches to deal with common faults. In some cases, the decompositions resulted in hundreds of individual rover commands (*e.g.*, panoramic image). If the resulting tasks were unordered, CPS would determine an ordering that satisfied the given time and power constraints.
4. The resulting schedule was passed back to the form-based interface, where it could be displayed and edited. Using the editor, individual steps, groups of steps, or whole branches could be removed or replaced. The resulting schedule fragment was fed back through CPS for any necessary decomposition and completion.
5. Finally, the schedule would be run through a simple syntax checker and uplinked to the on-board rover executive.

In order to allow the kind of mixed-initiative scheduling outlined above, CPS uses a greedy local search strategy. It accepts a seed schedule (possibly empty) and recursively attempts to improve it by fitting additional tasks into gaps in the schedule. When a plateau is reached, tasks already present in the schedule can be exchanged, removed, or shifted. Random walk and restarts further help CPS escape from local minima.

CPS also has the ability to automatically add contingent branches to schedules where appropriate. Building contingency plans is, in general, intractable, and so contingency planners tend to be slow [Draper, Hanks, and Weld, 1994; Pryor and Collins, 1996; Weld, Anderson, and Smith, 1998]. To overcome this problem, CPS employs the *Just-in-Case (JIC)* approach [Drummond, Bresina, and Swanson, 1994], originally developed to handle action duration uncertainty in telescope observation schedules. For the rover domain, we extended the JIC approach as follows.

- To consider uncertainty in power consumption and data production (as well as in task duration).
- To choose among potential contingency branch points based on an assessment of expected utility rather than just probability of failure.
- To allow insertion of setup steps for a contingent branch prior to the actual branch point.

6. THE FIELD TEST EXPERIENCE

In this section, we describe results and lessons learned from our Marsokhod field test experience. The 1999 field test was meant to simulate the main objectives of the Mars '01-'05 missions; the field test employed the

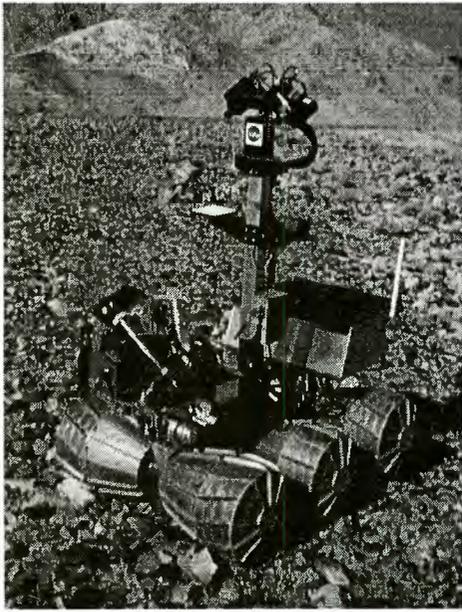


Figure 1: Marsokhod at the 1999 Mojave Field Test.

Ames Marsokhod rover (Figure 1) and took place during February. The remote site was at Silver Lake dry lake bed in California's Mojave desert, and the operations center was at NASA Ames. The field test team consisted of computer scientists and engineers from the NASA Ames Computational Sciences Division, scientists from NASA Ames Space Sciences Division, and planetary scientists from around the world; there were about seventy people who participated.

The Marsokhod platform has been demonstrated at field tests starting with Russian tests in 1993, followed by tests in the Mojave desert in 1994, at Kilauea in Hawaii in 1995, and in the Arizona desert in 1996. Marsokhod is a medium-sized planetary rover built on a Russian chassis. The rover has six wheels, independently driven, with three chassis segments that articulate independently. It is currently configured with imaging cameras that correspond to those planned for use in near-term missions, a spectrometer, and an arm equipped with cameras. The on-board computing environment is a Pentium-based Linux system, for ease of research software integration.

In the rest of this section, we describe the field test results and lessons learned for each of the major autonomy architecture modules: plan execution, mode identification, and plan generation.

6.1. PLAN EXECUTION RESULTS

This was the first Ames field test during which the rover was commanded by uplinking sequences, which were automatically executed on-board, rather than by "joysticking" with the Ames *Virtual Dashboard* interface [Wettergreen, *et al.*, 1997]. A major result of the field test was to build confidence in sequence-based commanding using the CRL language. Although, as expected, complex positioning tasks remain easier through real-time feedback and "joystick" controls, many tasks that involve repetitive activities or precise orientations can be more easily specified and more ef-

ficiently executed using CRL.

The following are some examples of how contingency plans were used in the 1999 Marsokhod Field Test and the preparatory readiness tests.

- If a visual-servo command terminates with failure, then acquire an image mosaic to enable re-localization by the operations team.
- If a wheel failure is detected, then acquire images of the failed wheel to support diagnosis.
- If orientation (taco angles) limits are exceeded, then stop and acquire images around all six wheels to support recovery planning.
- During a dead-reckoning traversal, if time (and data storage) allows, then take additional images, to support science and future operations, whenever the rover turns.

Another use of contingent plans is to support on-board, automated science analysis techniques, such as those being developed within the "Graduate Student on Mars (GSOM)" project [Gulick, *et al.*, 1999]. One of the GSOM suite of tools identifies rocks in an image. The following is an example employing this rock-finding algorithm within a contingent command plan. The rover drives a pre-set pattern (*e.g.*, a rectangular circuit) while scanning the environment for rocks. When a rock is found, the rover takes a high-resolution image of the region where GSOM indicates, and it stores this image for later downlink. Other tests, such as spectrometer readings, could be performed on the target location as well, potentially leading to other opportunities for on-board science analysis, *e.g.*, automatically identify carbonates from spectrometer readings. The rover is given a time limit to drive the search pattern, so if it spends too much time analyzing the images and performing tests, it skips some analyses in favor of reaching its way points on schedule.

An important part of robust, autonomous execution is to handle and react to failures that are not within the plan but throughout the system. We have taken steps in that direction with our explicit communication protocol to handle lost packets; however, other challenges remain, such as software failures within real-time controllers or hardware failures in the rover itself. Some of these are handled *via* fault identification by MI and recovery by contingency plans. Some system failures need to be handled in a more comprehensive manner to ensure the rover performs as desired. In particular, approaches ranging from simple heartbeat monitoring and pstate parameter recording to system reconfiguration need to be considered.

The plans constructed by CPS can include contingent branches to handle deviations from expected resource usage. The resources currently considered, in addition to time, are power and data storage. CX could make use of a resource manager to track resource usage and availability, as well as to signal resource conflicts or opportunities. We have developed a prototype resource manager and are integrating it into the on-board executive architecture. The resource manager will ensure that the rover executes its plans within the limits of the available resources and will support branching on a richer set of resource availability conditions.

6.2. MODE IDENTIFICATION RESULTS

Despite the advantages of our approach to state assessment and fault diagnosis, discussed above, MI does have some representation limitations, with respect to modeling rovers. These limitations can be classified as quantitative, probabilistic, and temporal.

Quantitative: There are many advantages to using qualitative models, as outlined above, but many of the more complex aspects of a rover that we would like to model, such as motor behavior and kinematics, are inherently quantitative. Consider again the simple threshold test discussed at the beginning of Section 4: a motor stall is indicated when the current-time product is too high. But how do we determine what is too high? “Normal” wheel currents depend on whether the rover is turning, driving uphill, or going over rocky terrain. The expected current, thus, is a quantitative function of factors such as pitch, turning, and bumpiness (as measured, perhaps, by accelerometers).

The approach we have taken is to use a purely qualitative model, abstracting away quantitative details using monitors. However, using this approach, we either end up with most of the complexity hidden in the monitors, or we are forced to discretize the values in question into many intervals and rely on qualitative arithmetic to do the math in MI, which can be computationally expensive. It would be much simpler and more efficient to work with the numbers directly. This is possible by incorporating quantitative models, using hybrid continuous/discrete systems, such as HCC [Carlson and Gupta, 1998]. HCC is already used for a simulation of the Marsokhod [Sweet, Blackmon, and Gupta, 1999], and work is underway to combine it with MI, for use in diagnosis. We are also considering the use of Kalman filters, which are ideal for combining numerical data from multiple noisy sensors, and which have successfully been used in MIR monitors.

Probabilistic: In MI, transitions to particular states can be conditional or probabilistic, but not both. That is, they are either deterministic, commanded transitions into “okay” modes, or unconditional random transitions into fault modes. Many aspects of the rover behavior involve conditional probabilistic transitions. For example, going up a steep hill results in high torque on the rear wheels, which leads to an increased probability that the wheel motors will stall.

With the current representation, we cannot express the fact that motor stalls are more likely to occur in the presence of high torques. To do so, we need conditional probabilities. Effectively using conditional probabilities requires tracking multiple trajectories, which is not currently done in MI for efficiency reasons; thus, entailing a larger computational burden. We are also considering other representations, including Markov decision processes.

Temporal: One of the assumptions underlying MI is that the system being monitored is synchronos, spending most of its time in a steady state (at least at the qualitative level reflected by the models) and that transitions between states are rapid enough that by simply waiting for quiescence, MI can treat them as instantaneous. However, on the rover, this assumption is violated. State transitions are sufficiently fre-

quent and transitions between states are sufficiently slow that there is no guarantee that the rover will reach a steady state. This is due in part to a high degree of uncertainty in the time that will be required for a transition to occur.

6.3. PLAN GENERATION RESULTS

During the field test, we learned a number of lessons about generating command plans. Some parts of the process worked very well, but there were places where we clearly needed additional software tools, or needed to improve the capabilities of our existing tools.

Probably the most glaring omission was the lack of adequate tools to allow the scientists to generate high-level CRL tasks directly. Although a web interface was developed for this purpose, it did not cover the full spectrum of possible scientific experiments and objectives. In addition, the interface did not allow them to specify temporal constraints and did not provide adequate feedback concerning resource requirements or expected data production for proposed experiments. As a result, the interface received little use by the scientists and, instead, the scientific goals were relayed verbally. As a result, significant manual labor was involved in turning the scientists requests into a fleshed out set of high-level CRL tasks.

In contrast, we made extensive use of automated decomposition of high-level science tasks into detailed sequences of rover commands. This capability was essential for efficient development of command plans. In some cases, the command plans contained hundreds of commands and we simply could not have generated these by hand in the time allotted.

We did not make significant use of the automated scheduling capabilities. The primary reason for this is that for each sol the scientists were providing a specific ordered set of tasks to be performed. They did not provide a larger set of prioritized tasks from which choices could be made, based on time, power, and data considerations. This was due, at least in part, to the relatively short duration of the field test, which did not allow the scientists to develop a set of longer-term objectives. Additionally, the scientists were not made aware of how to take full advantage of the capabilities CPS could provide. For an extended mission with a larger number of distributed scientists submitting requests, we believe that the scheduling capability would become more important, especially if employed to generate multi-sol command plans.

We also did not make significant use of automatically generated contingency branches. Without a larger set of tasks to choose from, CPS cannot build useful alternative branches. However, even with a larger set of tasks, CPS would not have been able to anticipate many of the failures that occurred during the field test.

Currently, CPS only develops contingent branches for failures that result from time and resource conflicts. During the field test, most of the plan failures were due to other things, such as losing visual targets during traverses and motor current anomalies. In these cases, useful alternative plans could have been developed automatically, but to do so, we need to enrich the set of potential failures considered by CPS.

7. CONCLUDING REMARKS

In this paper, we presented the *Contingent Rover Language (CRL)* for commanding planetary rovers, and we described the ground-based and on-board systems that were demonstrated in the 1999 Marsokhod Mojave Field Test. Our overall objective is to increase the flexibility and robustness of autonomous rover behavior in order to improve science productivity. The initial efforts towards this objective (reported here) focused on the concept of "contingency". CRL allows the specification of contingent courses of action for the purposes of recovering from expectation failures or taking advantage of serendipitous science opportunity. Our mixed-initiative planner/scheduler (CPS) supports the generation of contingent CRL command plans and our on-board executive systems (CX and MI) enable robust plan execution that is responsive to the runtime, dynamic environment.

In the previous section, we mentioned future work directions for each of the three component technologies. In addition, we intend to pursue command plan verification. In order to support verification, as well as plan generation, we plan to integrate rover simulation with constraint reasoning and planning techniques. In the future, we would also like to migrate some of the planning activities on-board the rover as appropriate; for example, the ability to replan science activities in response to on-board science analysis and runtime conditions (*e.g.*, resource availability).

ACKNOWLEDGMENTS

We'd like to acknowledge the following collaborators who helped with the work reported here: Corin Anderson for his help with CPS and user interface; Katherine Smith for her help with the MI system and Marsokhod models; Trey Smith for his help with plan execution; and Corin Anderson, Ted Blackmon, David Miller, Barney Pell, and Trey Smith for their help in CRL development. The implementation of CX uses constructs from ESL [Gat, 1996].

We'd also like to acknowledge the following collaborators that worked on Marsokhod's hardware, on-board software, user interfaces, stereo modeling, simulation, and other ground tools: Ted Blackmon, Maria Bualat, Vineet Gupta, Gary Haith, Aaron Kline, Linda Kobayashi, Cesar Mina, Charles Neveu, Laurent Nguyen, Sergey Sokolov, Hans Thomas, Anne Wright, and Eric Zbinden.

We'd like to thank all the field test collaborators: John Schreiner (Field Test Lead), Michael Sims (Technical Lead), Hans Thomas (Engineering Lead), Carol Stoker (Science Lead), and Nathalie Cabrol (Science Deputy), the NIR and MIR instrument teams, the GSOM team, and the other 50+ participants. We'd also like to thank NASA Ames management, in particular within the Computational Sciences Division, for supporting our research and the field test activity.

REFERENCES

- Bernard, D.E., Dorais, G.A., Fry, C., Gamble Jr., E.B., Kanefsky, R., Kurien, J., Millar, W., Muscettola, N., Nayak, P.P., Pell, B., Rajan, K., Rouquette, N., Smith, B., and Williams, B.C. 1998. Design of the remote agent experiment for spacecraft autonomy. In *Proc. of the IEEE Aerospace Conference*.
- Blackmon, T., *et al.*, 1999. Command generation for planetary rovers using virtual reality. In *Proc. of iSAIRAS*.
- Carlson, B., and Gupta, V. 1998. Hybrid CC and interval constraints. In *Proc. of the International Workshop on Hybrid Systems: Computation and Control*.
- Draper, D., Hanks, S., and Weld, D. 1994. Probabilistic planning with information gathering and contingent execution. *Proc. of the Second International Conference on AI Planning Systems*.
- Drummond, M., Bresina, J., and Swanson, K. 1994. Just-in-case scheduling. *Proc. of the Twelfth National Conference on AI*.
- Gat, E. 1996. ESL: A language for supporting robust plan execution in embedded autonomous agents. *Proc. of the AAAI Fall Symposium on Plan Execution*.
- Gulick, V.C., Morris, R.L., Ruzon, M., and Roush, T.L. 1999. Autonomous science analysis of digital images for Mars sample return and beyond. *The 30th Lunar Planetary Science Conference*.
- Mishkin, A.H., Morrison, J.C., Nguyen, T.T., Stone, H.W., Cooper, B.K., and Wilcox, B.H. 1998. Experiences with Operations and Autonomy of the Mars Pathfinder Microrover. *Proc. of the IEEE Aerospace Conference*.
- Pryor, L., and Collins, G. 1996. Planning for contingencies: A decision-based approach. *Journal of AI Research*.
- Sweet, A., Blackmon, T., and Gupta, V. 1999. Simulation of a Rover and Display in a Virtual Environment. In *Proc. of the American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems*.
- Washington, R., Golden, K., Bresina, J., Smith, D.E., Anderson, C., and Smith, T. 1999. Autonomous rovers for Mars exploration. In *Proc. of the 1999 IEEE Aerospace Conference*.
- Weld, D.S. and de Kleer, J. 1990. *Readings in Qualitative Reasoning About Physical Systems*, Morgan Kaufmann Publishers, Inc., San Mateo, California.
- Weld, D.S., Anderson, C.R., and Smith, D.E. 1998. Extending Graphplan to handle uncertainty and sensing actions. *Proc. of the Fifteenth National Conference on AI*.
- Wettergreen, D., Bualat, M., Christian, D., Schwehr, K., Thomas, H., Tucker, D., and Zbinden, E. 1997. Operating Nomad during the Atacama Desert Trek. In *Proc. Field and Service Robotics Conference*.
- Wettergreen, D., Thomas, H., Bualat, M. 1997. Initial Results from Vision-based Control of the Ames Marsokhod Rover. In *Proc. of the International Conference on Intelligent Robots and Systems, Control of Wheeled Robots*.
- Williams, B. C. and de Kleer, J. 1991. Qualitative reasoning about physical systems: A return to roots. *Artificial Intelligence*, 51:1-10.

AUTONOMY TECHNOLOGY CHALLENGES OF EUROPA AND TITAN EXPLORATION MISSIONS

David J. Atkinson

Jet Propulsion Laboratory, California Institute of Technology
MS 126-221, 4800 Oak Grove Dr., Pasadena, CA 91109
Phone: 818 393 2769, Fax: 818 394 2454, Email: djatk@jpl.nasa.gov

ABSTRACT

This paper discusses requirements for autonomy technology that arise from the unique attributes of proposed exploration missions to Titan, a moon of Saturn, and Europa, an ice-encrusted moon of Jupiter. Recently, the Project Design Center¹ at NASA Jet Propulsion Laboratory was the focal point for an intensive study of these missions. The mission to Europa tentatively includes a communications station on the surface of the ice, a "cryobot" which will melt through the ice to the ice/water interface, and a "hydrobot" which would free-swim under the water in a scientific search for hydrothermal vents. Autonomous commanding and fault protection technologies are key requirements of this mission, as well as the ability to conduct a science mission with very limited communication to other spacecraft or Earth. The proposed Titan mission includes an "Aerobot", a robotically controlled lighter-than-air vehicle. Part of the mission for the Titan Aerobot includes sampling and scientific analysis of surface materials. Some of the significant drivers of autonomy requirements on a Titan mission include the difficulty in selecting sampling sites, the consequences of long round trip light time delays for commanding, and exogenous events such as weather. Autonomous site selection, commanding, science operations, and robust fault detection, isolation and recovery are a few of the mission critical areas that are discussed in the paper.

INTRODUCTION

Europa is one of the highest priority targets in outer Solar System exploration. Liquid water is believed to exist beneath its highly fractured icy crust, perhaps forming a global ocean. At the bottom of this ocean there may be active volcanoes just as there are today on Europa's neighboring satellite Io. Most intriguing, life may exist near those volcanic vents, just as it is found on Earth: at great depths in the ocean, beyond the penetration of sunlight, thriving on upwelling chemical nutrients from the interior of the planet.² The driver for exploration of Europa is the discovery and description of its life.

Titan, the largest satellite of Saturn, is the only moon in the solar system with a substantial atmosphere. A dense nitrogen atmosphere, a haze of organic photochemical aerosols, liquid methane oceans, and potential volcanic activity make Titan a cauldron of activity. In remarkable environments like this, complex organic molecules are known to have formed, and these are the precursors of life. Among other goals, the search for these pre-biotic molecules is a priority for a Titan Aerobot mission.

These two missions present some of the most challenging requirements for autonomy technology in future space exploration. They represent several major shifts in what will be required by numerous future missions: Reactive planning in complex, dynamic environments, and; closed-loop interaction and decision-making in science data analysis.

EUROPA MISSION

The major goals of exploring Europa are:

1. Locate and describe the life forms in the Europa Ocean and ice crust
2. Evaluate the Europa Ocean, including its water, bottom, and ice cover, as a habitat for life. It is usually true that life is found at interfaces, and the bottom of the ice as well as the top of the sediment will both be key areas to explore.
3. Determine the long-term history of the European habitats and niches for unusual life forms. The discovery of so-called extremophile bacteria (bacteria that live where there are few of the normal food and other requirements for life) lead us to believe that truly novel life can evolve to be successful in a wide variety of niches.

The task of conducting exploration under the ice of Europa is vastly complex in every aspect including scientific strategy, space component, radiation environment, robotic systems for ice and ocean transportation, planetary protection, instrumentation, autonomy, and communications.

The mission scenario developed in JPL's Project Design Center includes major components that would be landed on Europa: The *lander* which would deliver the science payload to the surface and serve as a communication relay (direct to Earth or to a relay satellite in Europa or Jupiter orbit), a penetrator called a "*cryobot*" which would melt through the European ice sheet and deliver the third component, a submarine "*hydrobot*" to the European ocean for exploration.



Figure 1: Europa Submarine, Artist's Conception³

The European environment itself embodies many challenges to exploration. High radiation at the surface, about 14 Mrads per year, indicates that a lander must not only be highly radiation resistant, but should burrow under the surface for additional shielding by the ice if it is to survive the prospective two year duration of the mission. According to Galileo gravity data, the cryobot delivery system must penetrate approximately 10 kilometers of ice. The hydrobot must travel to the bottom of the ocean to a potential depth of 200 kilometers (the upper limit on depth). The pressure at the bottom is around 3 kilobars, roughly three times the pressure at the bottom of Earth's Mariana Trench. The ambient water temperature is around 260 Kelvin, although temperatures will be significantly higher near hydrothermal vents. It will be very dark, so lights for imaging are required. Strong tidal or other currents may be present.

The most interesting, and challenging, aspect of the mission involves exploring the sediment in search of volcanic activity and life.

EUROPA: AUTONOMY SCENARIOS

There are significant autonomy challenges for all three vehicles, the lander, the cryobot, and the hydrobot. In this section we identify these challenges in the context of likely mission scenarios.

Arrival and Landing at Europa

Our data on Europa will be more complete by 2015 when this mission is to launch. Nevertheless, some key decisions may need to be made upon arrival at Europa. Choosing a landing target area will be accomplished before launch. However, as the lander approaches the surface, it will be important to choose a location on solid and level ice from which to launch the cryobot into the ice. With the highly variable Europa ice terrain, this is likely to be a significant challenge. The significant light-time delay dictates that this operation must be accomplished using on-board autonomous capabilities.

The Descent of the cryobot

Upon release from the lander, the cryobot will be propelled downward by gravity as it melts through the ice (see Figure 2). The cryobot must penetrate approximately 10 km of ice. The speed of descent is on the order of 1 km per month. There are likely to be many hazards in the ice to be avoided. Large rocks or heavy concentrations of smaller rocks and dust can block the path of the cryobot. Pockets of water embedded in the crust could lead to the cryobot being stopped or to the hydrobot being released into a closed bubble. The path of the cryobot can be altered slightly using differential heating of the skin of the cryobot. Sonar and other sensors can be used to gather information about the surrounding ice and to aid in navigation of the cryobot.

The cryobot will trail a communication link to the lander, either fiber optic micro cable or multiple deployed relay repeaters. With its slow descent, it may be possible to teleoperate the cryobot to a certain degree from Earth even with significant light-time delay. However, once the cryobot nears the water-ice boundary, it must quickly detect and anchor itself in competent ice a few meters above the water-ice boundary in order to provide a base of operation for the hydrobot. This operation must be largely autonomous. Accidentally bypassing the competent ice into the water could mean a catastrophic loss of the mission.

The mission gets really interesting when we examine three key operational scenarios involving the hydrobot: Exploring the ice/water interface, descending to the sediment, and exploring the ocean floor.

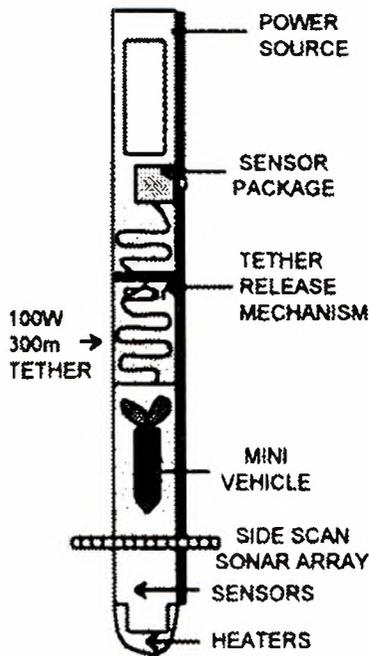


Figure 2: Early Cryobot Concept⁴

Exploring the Ice/Water Interface

Life on Earth is often found at the boundaries of different environments. Exploring the immediate area around the cryobot when it first encounters the liquid European ocean is therefore an important scientific goal of the mission. The cryobot may use sonar to characterize the general shape of the ice/water interface. The hydrobot will then explore nooks and crannies in search of interesting chemistry and biosignatures. High bandwidth communication between cryobot and hydrobot in this realm are possible, which will facilitate coordination between cryobot and hydrobot as well as reduce the immediate requirements for hydrobot autonomy.

Descending to the Sediment

The descent to the Europa ocean floor is more than just a drop through the water. The mission study anticipated that no more than 72 bps may be available to the hydrobot for communication with the cryobot. These communications limitations will initially force the hydrobot to stay directly underneath the cryobot until reaching the sediment. Scientists would also like to study the environment at different depths in the water column below the cryobot.

These requirements indicate that the hydrobot cannot afford to send engineering data after starting its descent to

the ocean floor. The hydrobot must reason about its own internal state, including analysis of its own engineering data to detect faults and appropriately modify its behavior to maximize science return.

Safely navigating a direct descent with uncertain tides and currents is an autonomy challenge. While descending, the hydrobot needs to reason about descent and lateral velocities in order to stay in acoustic contact with the cryobot and avoid crashing into the bottom.

Exploring the European Ocean Floor

In its travels through the European ocean, the hydrobot will use a combination of dead reckoning, inertial sensing, and the analysis of its surroundings to navigate. These, and imagery from side-scan sonar and other sensors, can be used to allow the hydrobot to maintain an estimate of its location relative to the cryobot.

Given the low communications bandwidth, the hydrobot will have to detect and pursue science opportunities with minimal interaction from Earth. This involves generating its own map of the bottom while out of communications, identifying potential targets for further study, performing the studies, and then returning to a place below the cryobot where it can uplink results.

A number of exploration patterns have been suggested. Owing to the limitations of telecommunications through seawater, the hydrobot must return to the vicinity beneath the cryobot in order to uplink data collected on its surveys. Even directly beneath the cryobot, communications capabilities are likely to be very limited. The most likely exploration patterns include "out and back" features, with multiple lobes in different directions centered on the spot immediately below the cryobot.

These mission attributes indicate that the hydrobot must have the ability to reason about when it can communicate, carefully select what to say, and maintain a coherent dialog with the cryobot over the course of the mission.

As the hydrobot explores the ocean floor it will search for hydrothermal vents using side-scan sonar, flash photography, and chemical and thermal sensors. The chemical and thermal sensors will most likely be used to simply follow gradients to their source. When in the vicinity of a vent, side-scan sonar maps of the area will allow the hydrobot to pin-point the vent. Once the vent has been located precisely, flash photographs of the vent will then be used to guide the taking of samples from the vent and its environs. This will include scraping the vent and bringing the sampled material closer to on-board sensors.

It will not be possible for the hydrobot to transmit all the collected scientific data to Earth due to both restricted communication opportunities and on-board data storage capacity. Priorities concerning data taken to support the science objectives will be established before launch. However, the data collected in each phase of the mission will need to be prioritized autonomously by the hydrobot for relay to Earth based on quality, information content, and relevance to mission objectives.

CORE AUTONOMY FOR THE EUROPA MISSION

There are clearly many technical challenges present in a mission to Europa. A few of the most central core autonomy technologies required for the Europa mission vehicles include reactive planning, data fusion and interpretation, and scalable computing.

Since the hydrobot will be out of touch with human controllers for extended periods, the effects of uncertainty and incomplete knowledge about the environment will make it infeasible to execute detailed plans generated on Earth. Such detailed plans might work reasonably well for short range missions at the ice/water interface, but hydrobot missions any significant distance away from the cryobot will need a planner to continuously adapt an abstract mission plan to the current context as it unfolds. Unexpected events are likely in a mission of this complexity in an uncertain, dynamic environment. Reactivity will also help make the most of scientific opportunities, such as the detection of a hydrothermal vent. Survivability is enhanced by reactivity and continuous planning as well. There is no static "safe" mode in which the hydrobot can stop and wait for instructions from Earth. The hydrobot must return to a point below the cryobot before it can communicate with Earth-based controllers. An appropriate response to engineering anomalies will require the vehicle to remain "fail operational" so that it can contact Earth if necessary and continue the mission. An approach to these continuous reactive planning requirements called "Iterative Repair Planning" is currently being pursued at JPL.⁵

In order to allow the cryobot and hydrobot to navigate, a number of different sensor modalities will have to be combined through data fusion. Using input from the sonar and chemical and thermal sensors, both the cryobot and hydrobot will have to form models of the world around them. The fusion of this data must result in a consistent stable model of the world which can be used both for navigation over thousand kilometer traverses and planning to achieve mission goals.

With a largely unknown environment, there will be much uncertainty in the vehicles' reasoning about its location in the world and in the data that it receive from the sensors.

Missing and anomalous data will occur in many circumstances. For example, an area may be shadowed by an underwater obstacle, a fault may have caused a loss or corruption of data, or a damaged or faulty sensor may leave the robot partially blinded. This missing or anomalous data must be accommodated seamlessly without paralyzing the vehicles' ability to navigate autonomously.

In the absence of oversight from Earth, the lander, cryobot, and hydrobot must self-calibrate their instruments so that meaningful analysis can be performed on-board. As the robotic vehicles move through the European environment, conditions will change and the sensors and algorithms using sensor outputs will have to recalibrate and track the drift. Furthermore, although we generally assume that the environment will be stable in the short term, we must be prepared for it to change over the course of the mission (the Cryobot will descend through the ice for ten months and the hydrobot will explore the ocean for up to twelve months). Both the ice/water boundary and the ocean floor may evolve with time (e.g., hydro thermal vents often appear suddenly) and the model of the world maintained by the robots will have to adapt accordingly.

Underlying many of the autonomy technologies required for the Europa mission is the ability to process images and other sensor data to recognize and classify patterns of interest. Classification is carried out in the presence of noise which is inherent in the environment and the sensor modalities. Robust noise-tolerant algorithms for classification have yet to be developed. Unique classification methods that operate over multiple dynamically-evolving data sets must also be developed. These will be key in the search for underwater hydrothermal vents. For example, combinations of water temperature and concentrations of dissolved gases may be used to help identify the direction and location of underwater vents. Some of the most important classification algorithms that must be developed are in the area of biosignature recognition.

The Europa hydrobot promises to have the most computationally intensive operations of any future mission, and much of this derives from the requirements for autonomy. With many different semi-independent computational subsystems such as the planner, navigator, world modeler, data acquisition, and data analysis all vying for computational resources, parallel processing and intelligent scheduling of tasks will be necessary to get everything done in an efficient and robust way.

TITAN MISSION

Titan, the largest satellite of Saturn, is the only moon in the solar system with a substantial atmosphere. The dense nitrogen atmosphere has twice the surface pressure of that of Earth. This makes it practically ideal for exploration with Aerobots.⁶ A ubiquitous haze layer of organic photochemical aerosols obscures the surface from observation from space except with radar. (In the Los Angeles area, we would refer to this as a “class one million smog alert”!) An in situ vehicle penetrating beneath the haze layer may find a remarkable low temperature world in which familiar features of Earth such as oceans, rainfall and volcanic activity appear. The surface may include liquid oceans, solid features, and slush. The oceans may be composed of liquid methane, the rain made up of drops of methane and liquid nitrogen and the lavas pouring on to the surface formed of liquid water and ammonia. In this remarkable cauldron of activity, complex organic molecules are known to have formed and prebiotic molecules may exist. The highest scientific priority at Titan is the chemical analysis of surface materials.

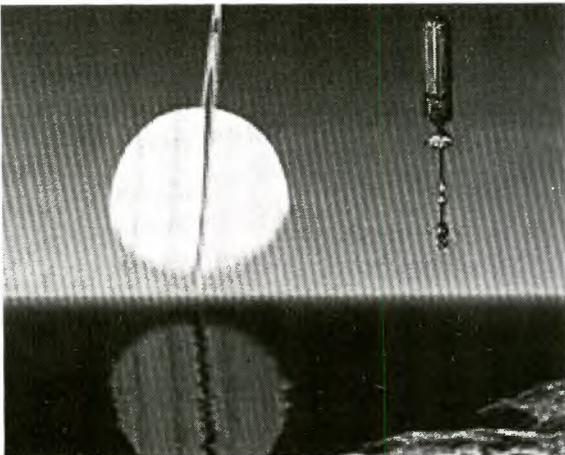


Figure 3: Titan Aerobot, Artists Conception

Some of the high level objectives for a Titan Aerobot mission would include:

- Characterize surface morphology below haze layer.
- Make low atmosphere chemical composition measurements.
- Sample surface (liquid and solid) organic chemistry and "mineralogy" at designated sites.
- Contribute to understanding of global atmosphere dynamics and winds.
- Perform global inventory of surface volatiles.

The mission concept includes a Titan orbiter and a planetary aerobot. The orbiter would be used both as a

science platform and for data relay, either direct to Earth or possible relay via the Cassini spacecraft in orbit around Saturn. The planetary aerobot would descend into Titan's atmosphere for direct in-situ measurements and exploration. Planetary aerobots are robotically controlled lighter-than-air vehicles. Titan's dense atmosphere, extreme but uniform temperature environments, and challenging surface environment make it a good candidate for exploration with aerobots.

TITAN: AUTONOMY SCENARIOS

After aerocapture and insertion into the Titan atmosphere, the Titan aerobot would drift longitudinally with the Titan winds. Periodically, the aerobot would descend to a targeted point on the surface for sampling and other tests. The concept for the primary mission requires sampling from a minimum of ten separate sites around Titan.

There are several driving challenges for autonomy in the mission concept: The aerobot must select target sites for sampling as it floats over the world. The aerobot must navigate from high altitude to the targeted site, and conduct the sampling operation. The aerobot must respond safely to exogenous events, such as weather. And the aerobot must make the best use of the available bandwidth of the communications relay to send the highest priority data back to Earth. We will discuss each of these in turn.

The selection of sites for surface sampling and analysis poses one of the first challenges. Due to the layers of atmospheric haze, it is impossible for the sampling sites to be pre-selected from orbiter data. Similarly, round trip light time delays of three hours or more prohibit real-time selection of sampling sites by scientists on Earth: By the time a site was selected, the aerobot is likely to have drifted past a point where it can descend from altitude to the target. By integrating data from a variety of instruments onboard the aerobot gondola, the aerobot must be able to autonomously identify a desirable site for sampling when the opportunity arises.

Before the mission, we can provide the aerobot with several heuristics to aid in site selection. For example, the first two or three sampling sites are likely to be “safety” driven, that is, we want to sample from areas where the relevant systems can be exercised and samples collected without substantial risk to the mission. Risks might arise for example, from snagging on rough terrain and this would suggest that a safe site is one with a relatively smooth surface. Other heuristics might include: sample from areas with whose albedo contrasts with nearby areas, sample from a variety of topographic surface types (e.g., shorelines, valleys, cliffs), and sample a certain distance between sites. From aerobot imagery, we may be lucky to see and extract scientific features that could be useful

sampling targets, such as geysering. While helpful, these heuristics are not sufficient to help select desirable sampling sites. Although we may have a limited amount of surface topography information from the Huygens probe on a coarse level, the bottom line is that we won't know the precise characteristics of scientifically desirable sample sites until the aerobot has arrived at the planet and captured and analysed several samples. This consideration suggests that a trainable recognizer for science site selection will be required.

Once a sampling site has been selected, the next challenge is to reach it so that sampling mechanisms can be deployed. This need not occur directly at the surface; a variety of sample collection mechanisms from low altitude are under consideration including deployable instrumented snakes, sondes, and other tethered sampling paraphernalia.⁷

Like most balloons, the aerobot has only vertical control for ascent and descent. Otherwise, it is at the mercy of the Titan winds. These winds are estimated to be tens of kilometers per hour at five thousand meters ("cruise") altitude, and about 1 kilometer per hour at the surface. With three hour round trip light time delay from Earth, commanding the descent to a target sampling area must be conducted autonomously on-board the aerobot. As the aerobot descends, it may easily drift laterally and thereby bypass the targeted site. A better approach would be for the aerobot to select several target sites, and then plan a descent trajectory that will give it the highest potential of reaching the highest value sites. The planned trajectory would be monitored and altered as necessary to achieve the goal. To do this, the planning system must have an understanding of the scientific value of the target sites, and use this information to set appropriate goals.

The success of the aerobot's mission is dependent on its ability to plan and execute effective operations in the context of what is likely to be a very dynamic weather environment. Methane rain, storms, winds, lightning and other meteorological phenomena will affect the aerobot's plans on a continuous basis. The aerobot planner will be able to make better plans if it can adapt and correct its predefined model of Titan weather effects using experiential data. Similarly, over the course of an extended mission lasting several Titan seasons, it is reasonable to expect on-board anomalies and other contingencies to arise, some due to weather effects (e.g., charge build-up, corrosion) and some due to equipment or software malfunctions. The aerobot should have the ability to plan flexibly in the context of a degradation of its capabilities to continue the mission.

The vast majority of the volume of Titan aerobot scientific data will consist of imagery. Other instruments in the baseline mission concept include a gas

chromatograph mass spectrometer (GCMS), an x-ray fluorescence instrument (XRF), an infra-red spectrometer, specialized instruments for study of pre-biotic chemistry, a complete wet chemistry lab, and radar. Although onboard data storage will be prodigious relative to current missions, it will nevertheless be oversubscribed. Scientists will undoubtedly demand certain types of data be returned, and this will pose an additional constraint on the resources. Furthermore, data relay to the orbiter will be intermittent and have limited bandwidth. The aerobot therefore must have the ability to decide what data to keep and what data to transmit back first. For example, imagery (wide angle or narrow-angle) may reveal important scientific features that cannot be reached by the current balloon pass. These features can be flagged and prioritized for later download to Earth.

CORE AUTONOMY FOR THE TITAN MISSION

The Titan aerobot mission requires an unprecedented degree of autonomous decision-making and commanding. Many of the technology needs are shared with other in-situ explorations, such as the Europa Ocean exploration. A few of the most central core autonomy technologies required for the Titan mission include:

- closed-loop sensing, planning, and execution;
- goal-based commanding, resource management, fault detection, fault isolation and fault recovery;
- contingency planning;
- adaptive planning;
- adaptive modeling;
- autonomous science image feature detection;
- on-board science data processing.

Technology development for these capabilities is challenging and a very active area of research. Current research at JPL is pursuing variants on an architecture consisting of four fundamental components: a mission manager, a planner scheduler, a diagnostics executive, and a real-time controller.^{8,9} While the real-time controller implements activities by managing feedback control loops, the diagnostics executive determines the internal state and external surroundings by monitoring (and possibly aborting) the feedback loops. Given the context determined by the executive, the planner scheduler reasons about desired future activities and instructs the executive what to do next. Finally, the mission manager determines mission and context dependent goals to motivate future desired activities. These activities are computed and maintained by the planner.

Different technology alternatives implement these components in different ways. In some cases components are even merged into a single rule-based expert system on top of a real-time control system. The main development

issue involves how much reasoning is performed at each level, and whether the levels interact continuously or intermittently. For instance, the DSI remote agent has an executive that continuously interacts with the real-time system, but the planner scheduler only wakes up intermittently to interact with the executive. The component technologies are described more fully elsewhere.¹⁰

A key research topic is how to deal with uncertainty in both the world model and the results of the actions of the aerobot robot on the world. For instance, in an aerobot descent, reducing the buoyancy a certain amount for a given time will not necessarily result in the predicted vertical movement unless the atmospheric pressure and winds aloft are taken into account during plan execution. Ideally, the world modeling system would be able to use experiential data to reduce the uncertainty for future descent plans.

One of the key attributes of the algorithms used to provide the autonomy capabilities is that they must produce results in an incremental fashion such that they can be stopped any time and produce the current best answer. Anytime algorithms are needed so that resources can be redistributed quickly if necessary instead of being tied up with lengthy calculations before producing a high quality answer. Also, if more resources are available, the algorithms can be run longer in order to achieve a more precise answer. The algorithms also need to scale with the resources available to them. The resources available may change due to usage by higher priority tasks or partial failure of the computing hardware.

SUMMARY

In this paper, we have outlined the autonomy challenges for two of the next millenia's most exciting and challenging missions: Exploration of the oceans of Europa and the atmosphere and surface of Titan. Some of the component autonomy requirements are common to both missions, such as the ability to make and execute plans in a highly uncertain and dynamic environment, with limited ability to interact with Earth-based mission controllers. Another common attribute of these missions is the need to include science planning, data collection, and data interpretation in a closed-loop with autonomous mission planning.

Current autonomy technology research programs have been occupied for years with developing robust component systems, such as planners, diagnostic systems, and science data analysis systems. Considerable progress has been made and these systems are now entering routine use in ground applications. Only recently have some of these components come together in technology flight experiments, such as the DSI Remote Executive

Experiment (RAX). To achieve the level of readiness required for the Europa and Titan missions, considerable new research effort needs to be made to bring scientific judgement into the autonomous control loop of these systems. Despite the fact that these missions are at least a decade away, the time seems short.

ACKNOWLEDGEMENTS

The Europa and Titan mission studies were conducted by JPL's "Team X" in the Project Design Center, a joint activity of JPL Space Exploration and Science Directorate and Technology and Applications Program Directorate. This paper draws from the comments, discussion and data produced by many individuals who participated in the study. The author especially wishes to acknowledge the valuable contributions of Drs. Anthony Barrett, Jonathan Cameron, Anita Govindjee, Benjamin Smith, Paul Stolorz and Timothy Stough. Copyright © 1998, California Institute of Technology. ALL RIGHTS RESERVED. U. S. Government Sponsorship Acknowledged under NAS7-1407

NOTES

¹ The Project Design Center, <http://pdc.jpl.nasa.gov/>, Jet Propulsion Laboratory, Pasadena, CA, 1999.

² "Europa/Vostok Initiative", <http://spacepc.jpl.nasa.gov/europa/>, Jet Propulsion Laboratory, Pasadena, CA, 1998

³ Artists concepts courtesy of Jet Propulsion Laboratory, California Institute of Technology, 1998.

⁴ Europa Vostok Initiative Implementation Plan, Space and Earth Science Programs Directorate, Jet Propulsion Laboratory, Pasadena, CA, 1998

⁵ "Iterative Repair Planning for Spacecraft Operations Using the ASPEN System.", Rabideau G., Knight R., Chien S., Fukunaga A., Govindjee A., Proceedings of I-SAIRAS'99, Noordwijk, NL, 1999.

⁶ "Titan Exploration with Aerobots", <http://telerobotics.jpl.nasa.gov/aerobot/studies/titan.html>, Jet Propulsion Laboratory, Pasadena, 1998.

⁷ "Titan Aerobot Missions", Team X-T Study, Cutts J., Cameron J., unpublished presentation, Jet Propulsion Laboratory, Pasadena, CA, 10 November 1998.

⁸ Artificial Intelligence Group, <http://www-aig.jpl.nasa.gov/>, Jet Propulsion Laboratory, Pasadena, CA, 1999

⁹ Machine Learning Group, <http://www-aig.jpl.nasa.gov/public/mls/>, Jet Propulsion Laboratory, Pasadena, CA, 1999

¹⁰ Op. Cit., Rabideau G.

Automation of Spacecraft Mission Operations

COMPUTER INTELLIGENCE IN INTEGRATED SATELLITE DESIGN SUPPORT INFRASTRUCTURE

Shinichi Nakasuka and Koichiro Maeda
Department of Aeronautics and Astronautics, University of Tokyo,
Hongo 7, Bunkyo-ku, Tokyo, 113-8656, JAPAN
TEL:+81-3-5841-6590, FAX:+81-3-5841-8560, nakasuka@space.t.u-tokyo.ac.jp

Masao Sato, Kotaro Kiritani, Kenichiro Sato and Toshio Koda
Kamakura Works, Mitsubishi Electric Corporation
Kamimachiya 325, Kamakura, 247-8520, JAPAN
TEL: +81-467-41-6155, FAX: +81-467-41-6951

ABSTRACT

An intelligent computer system to support satellite design is proposed. To support human designer in time consuming trial-and-error design process, the system has the capability 1) to integrate various local design support tools to obtain one feasible solution as quickly as possible, 2) to give intelligent advise to the designers on how to modify the current design in order to improve the design solution in a certain direction, and 3) to enable human designers to customize and implement a certain design sequence so that the computer can perform parts of design process autonomously. Gradient search technique and machine learning-based production system is employed for the second function, and a concept of design process editor is proposed for the third function. The way to implement this computer intelligence into the system is discussed. Basic concept of the system architecture as well as a prototype model is described and discussed.

1. INTRODUCTION

Satellite design is a highly complicated and time consuming task because many different areas must be considered concurrently to reach a consistent as well as satisfactory design solution. In AI terms, this process can be interpreted as searching for a set of many design parameters which yield a design solution satisfying various given requirements under various constraints. In satellite design case, the number of design parameters, and consequently, the search space is huge. Moreover, the search process becomes even more difficult because of the complicated interactions between design parameters. For example, a modification of a certain parameter affects, in most cases, not only the quality of the design solution but also the characteristics of the design problem, such as the sensitivity of the design quality to the change of other parameters. As a result, the search process should become highly trial-and-error fashion, including many backtracking and iterations.

In the current satellite design, this search is made in

most cases by human designers. Computer support is partly incorporated, but is limited to the local analysis of the effects of design parameters within each design area, such as within thermal design, communication link design or control system design. The most difficult task of trading-off between the requirements given from the different areas or obtaining a consistent design solution is dependent on human designers, which results in the long time required even to obtain one feasible solution. From the needs' side, however, it is essential to obtain one feasible design solution quickly especially in the conceptual design phase, and it is highly desirable if the computer system could advise how to change certain parameters to improve the design quality. These capabilities have not been provided in the current computer aided satellite design system.

In order to respond to these needs, we have been developing an integrated computer support infrastructure for satellite design. The key objectives of this system are as follows:

- 1) Integration of the various local design support tools in order to obtain one feasible solution as quickly as possible.
- 2) Intelligent support by the computer to advise the designers how to modify the current design in order to improve the design solution in a certain way.
- 3) Enabling human designer to customize and implement a certain design sequence so that the computer can perform parts of design process autonomously.

In section 2, the concept as to how the computer supports human in design process is proposed, and section 3 gives the detail of the system architecture. Modification of design parameters plays the key roles in the proposed design framework, and computer intelligence is highly required to support this function, which is discussed in section 4. Current status of actual implementation is briefly given in section 5, and conclusions and future works are summarized in section

6.

2. BASIC CONCEPT OF COMPUTER SUPPORT IN DESIGN PROCESS

2.1 Difficulties of Design Problem

Design problem can be interpreted as searching for a set of usually many design parameters which yield a design solution satisfying given requirements under given constraints. In case of satellite design, the difficulty comes from not only its large number of design parameters and evaluation items but also from the mutual interactions between them. The whole design problem can seemingly be separated into several design sub-problems such as orbital design, structural design, thermal design, control system design, etc, but actually these sub-problems have strong mutual interactions, and one good solution of a certain design problem often is a bad solution for another problem. At that time, human designers often negotiate with designers of the conflicting design fields, trading-off the requirements given to them, finally to reach the acceptable, not optimal, solution. This process is quite time consuming and even stressful, and therefore we need an intelligent support from computer not only in easy analysis or drawing of the designed subsystems but also in trading-off the various requirements in the different design fields.

This kind of "Divide-and-govern" approach has been studied extensively especially in the field of distributed problem solving or multi-agent systems (MAS). For example, Yoshida et al. proposed a new MAS architecture in which each agent performing different field of design communicates with the other agents by way of evaluating the design results of the other agents [1]. By learning appropriate balance between the desire of each agent to pursue their own benefits and the desire to cooperatively achieve the goal, the total system gradually acquires expertise to reach acceptable solutions more and more efficiently. Many researches have been performed in this line, but the common limitation of this approach is the difficulty to specify the way or expertise to mitigate conflicts between agents in the different design fields.

2.2 Sequential Operator Application Approach

Another approach to this design problem is "sequential operator application" framework such as in Fig.1. Starting from the initial design (top node), "design operators" are applied sequentially until the goal condition (i.e., sufficiently good design) is achieved. Here, "design operators" mean such operators as to design a certain part which has not been designed yet, to change the current design in a certain way, or sometimes

to modify the given requirements and constraints. The initial design may be a design of the previous satellite with similar missions, or sometimes nothing. The search strategy may be depth-first type, breadth-first type or best-first type. This formulation is quite straightforward and suited for applying various AI techniques, but its difficulty resides in the combinatorial explosion of the search space because there are usually many applicable operators at each node.

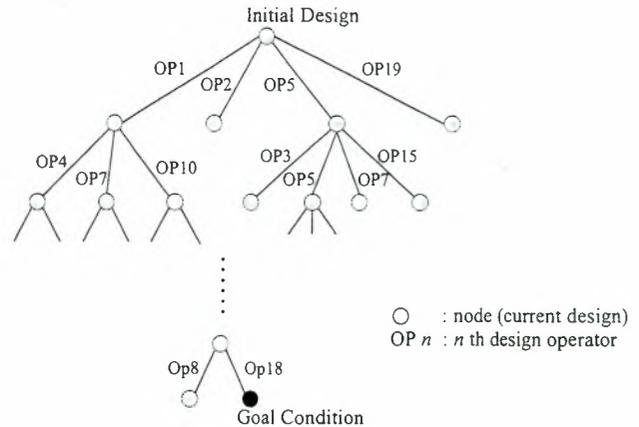


FIG.1 Sequential Operator Application Framework for Design Problem

One way to solve this problem is that the system is implemented with knowledge to specify what type of operator should be applied at what situation. Nakasuka et al. suggest in [2] and [3] that machine learning can play the essential role in obtaining such knowledge, in case that human designer cannot provide enough knowledge for this objective. In [2] and [3], knowledge as to the relationships between the attribute values describing the current design status and the desirable design operators are acquired by machine learning in the course of initially trial-and-error style problem solving process. This methodology has been found quite effective and applicable to various problems, and successful applications have been made to design of control system [2] and scheduling problems [3][4]. The architecture proposed in this paper also employs this machine learning based methodology in principle. The detail will be described in section 4.

2.3 Design Process Editor

Another important knowledge often very useful for design efficiency is "sequence of design." For example, we had better define the equipment list before designing the size of solar panel, and we frequently had better design satellite orbit before designing communication system (Of course, there is some problem where the reverse is true.) This appropriate sequence of design sometimes comes naturally from the causal relationships of parameters (such as the former example above) or sometimes comes from more

complicated efficiency reasons (such as the latter example). The desirable design sequence of the former type can easily be deduced from the causal network of design parameters. The latter type, on the other hand, is itself a certain expertise and cannot easily be specified. Computer should support human designers also in these respects. In [5], Obata et al. proposed a satellite design support system which can provide the capability of generating several "design processes," which specify the sequence of designing multiple parameters, from which users can select the most appropriate one according to their intention. Obata uses model based method which does not require any experiential knowledge to generate the candidates of design process.

In this paper, this idea is further enhanced to propose an idea of "design process edit," which enables users to "design the design process" freely, in addition to the above capability of model based generation of the sequence of designing parameters based on the causal relationships of parameters. Figure 2 and 3 give one example of this capability; Fig. 2 is a causal relationships of design parameters and Fig.3 shows a generated design process partly based on Fig.2 and partly on the user edition.

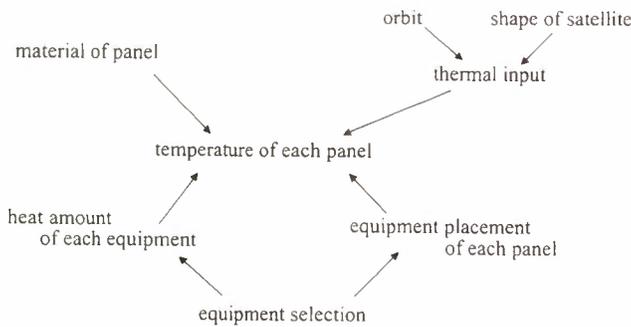


FIG.2 Example of Causal Relationships of Parameters

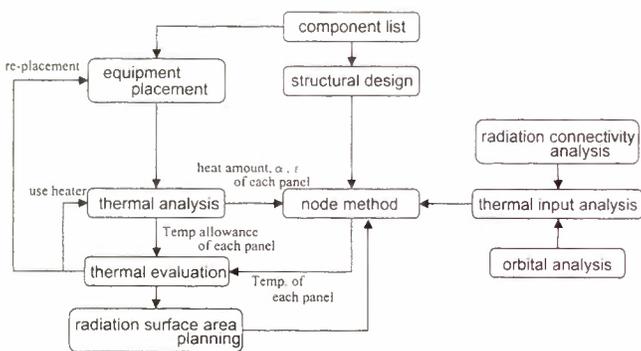


FIG.3 Example of Generated Design Process

It should be noted that not only a sequential parameter setting but also an iteration loop can be incorporated, if needed. Such loops are needed where parameters are dependent on each other. The design process can also reflect the priority between different requirements.

The design process is in a sense a "script" in AI terminology; it describes how the parameters are sequentially got set or tuned. Once this design process is completed, the computer automatically chooses how to set a certain parameter from among the options of simply calculating it from the already set parameters, doing some iterations until converged, or consulting human designers for the value. The system should be made so that once a certain parameter has been modified, then all the parameters affected by this change are re-calculated automatically. With these capabilities, the system can quickly generate one solution while maintaining the consistency of the parameter values, even though of course the solution does not always satisfy all the requirements and constraints.

3. PROTOTYPE SYSTEM ARCHITECTURE

A prototype system is now being developed based on this concept. The system consists of several modules each of which corresponds to a local design tool such as link design tool or solar paddle design tool, and is coded in object oriented fashion (Fig.4). Each module has input parameters and output parameters and knows which modules can possibly change the values of its input parameters. Some modules have a user interface to enable human designer to modify some parameters directly and others have just the function of calculating the output parameter (design results) from the input parameters (assumptions.) In the design phase, when a certain module is triggered to change some parameter values, the downstream modules (modules which are affected by this change) are also triggered automatically to maintain consistency of the parameter values.

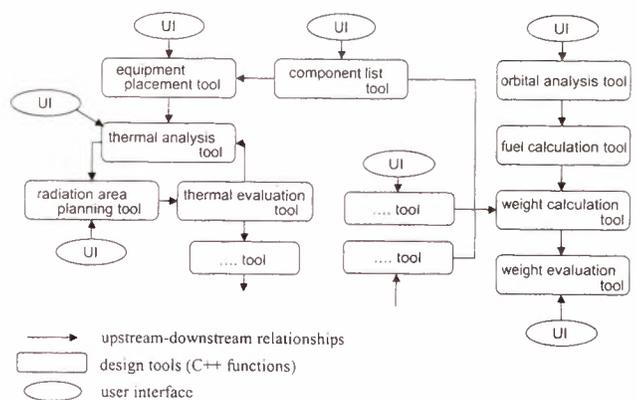


FIG.4 Overall Architecture of Prototype System

Some iterative design modules are also implemented in the same manner, which performs a certain design process requiring an iterative calculation which does not require any expertise such as which parameter to change, etc., until a certain local constraint(s) is satisfied. Examples of such modules already implemented in the prototype system are as follows:

- Tuning of RF power to satisfy link margin requirement and PFD constraint
- Tuning of altitude to satisfy link margin requirement and PFD constraint
- Tuning of antenna gain to satisfy link margin requirement and PFD constraint
- Tuning of area of surface radiation and heater electricity power to satisfy temperature requirement
- Tuning of required fuel to keep the desired altitude
- etc.

As you can note from these examples, there are several ways to modify parameters to satisfy a certain currently not-satisfied requirement (such as the first three in the above example for link margin and PFD requirements.) These modules do not provide support in terms of which modification way is the best in the current situation, but provide just an automatic calculation function which requires, if a human designer does it, lots of time because of iterative nature. A human designer, therefore, must specify which to use to modify the parameters.

The key characteristics of this architecture are that :

- (a) each module is coded in declarative form and the sequence of its activation is not hard coded, which enables human designers to pursue their own design process and to modify any design parameters at any time of the design phase,
- (b) the consistency of the design parameters is maintained at any moment, which enables designers to change parameters without paying much attention to the effects of this change,
- (c) local experiential design rules can easily be implemented in the form of design modules which can also be triggered at any moment, and
- (d) the system does not change parameter values by itself unless the changes are necessary to keep consistency or a certain iterative design module is triggered by a human designer.

With this capability, the system can quickly obtain one design solution. However, it is not guaranteed that the given requirements and constraints are satisfied in the obtained solution. The next question is how the system can support the efficient modification of parameters to more satisfy the requirements and constraints, and computer intelligence is, of course, required in this respect. We will discuss it in the next section.

4. INTELLIGENT SUPPORT FOR PARAMETER MODIFICATIONS

4.1 Sum of penalty functions

In order to systemize the satellite design, it would be required to provide some way to evaluate the overall goodness of the design in an objective way. In AI field, "sum of penalty functions each of which represents how each requirement or constraint is satisfied" is frequently used for this objective. We would like to follow this trend. In this method, hard constraints can be represented by very sharp valley of penalty function as in the right figure of Fig.5, while standard requirement will be like the left figure. The merit of using this strategy is that the intention of the human designer can very easily be represented, such as which requirement has higher priority or how strict the constraint is, etc.

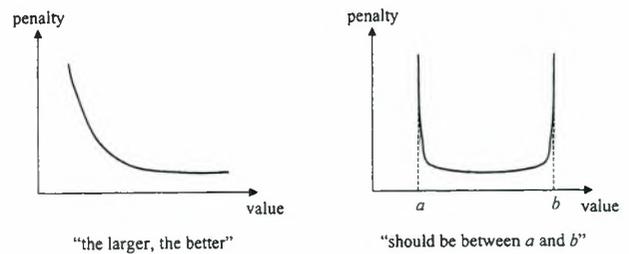


FIG.5 Penalty Functions to Represent Requirements (left) and Constraints (right)

Using the term of "Sum of Penalty Functions" which is referred to as "SPF" hereafter, the ideal design process can be stated as "the search for the global minimum of SPF". This is, of course, the ideal design, and usually we cannot obtain the global minimum, but just local minimums or a certain point whose SPF is below a certain threshold. In our concept of computer intelligence to help design, computer should provide some guidance to lead the human designer to the global optimum or the design point whose SPF is as near the global minimum as possible. We identify two mechanisms needed for this objective.

4.2 Local modification based on gradient search

Locally, we can tell, based on the sensitivity of SPF to each parameter, modifications of which parameter to which direction will reduce the SPF. This type gradient descent algorithm is rather easy to implement, such as in the following algorithm:

- (i) find the penalty function which has the worst value in the current design
- (ii) find the design parameters which have large effects

- on the selected penalty function
- (iii) search for the changes of these parameters which result in the largest descent of the SPF
- (iv) if any changes will not reduce SPF any more, then quit. Otherwise go back to (i).

The most important expertise in this algorithm is (ii), that is, the knowledge as to which parameters should be modified in order to improve a certain penalty function. In the prototype system, this expertise is obtained from the human expert designers in the satellite design area. We are now studying application of machine learning technology in order to more precisely indicate which parameters to modify.

4.3 Global modification based on design knowledge

When the above algorithm stops, the design solution is at “a local minimum point.” If SPF of this point is below the threshold, then the design is completed, i.e., we have obtained one feasible solution which satisfy the requirements and constraints. But, if not, the local search alone cannot lead the design to the good direction any more. Or, if we want more than one design alternatives which all satisfy the requirements and constraints to some extent, this local search will not be of any help. In these cases, we need more global or “jump type” modifications such as in Fig.6.

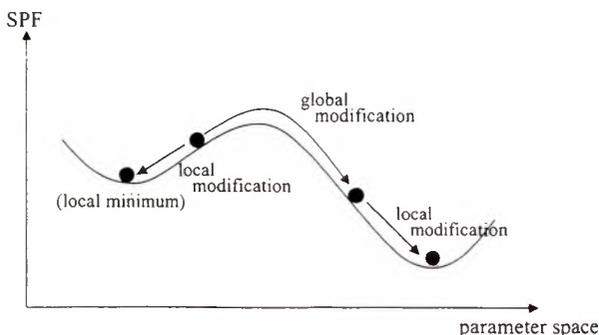


FIG.6 Local and Global Parameter Modifications

The simple and effective way to realize this function is to prepare several modification rules in the form of IF-THEN production rule. IF part represents the current situation of the design, which should include:

- which penalty function(s) has bad value now
- which parameters still have room for increase (or decrease)
- relative hardness of requirements/constraints (such as “weight constraint is very hard in this design problem as compared with electricity constraint”, etc.)
- etc.

THEN part shows some modification operation such as:

- modify parameter A to reduce a certain penalty
- modify parameters A and B so that a certain penalty function(s) can be as small as possible
- modify parameter A until a certain penalty function becomes less than a certain value and then modify parameter B until this penalty function become sufficiently small
- increase (or decrease) a certain parameter by a certain amount
- etc.

The attributes in IF part and the modification strategies in THEN part as well as some combinations of IF and THEN parts can be specified beforehand using expertise in satellite design. Examples of such knowledge are as follows.

- IF the total weight is too large, THEN change the equipment to be used.
- IF the total weight is too large, THEN replace the current equipment with one requiring less power.
- IF the total weight is too large, THEN change the initial altitude.
- IF the required heater power is too large, THEN change the position of the equipment requiring largest power.
- etc.

However, as you can easily imagine, this a priori knowledge alone is not sufficient to effectively lead the design process to the global optimum point. In order to deal with this “knowledge bottleneck” problem, we have been applying machine learning technology in the same way as employed in [2] -[4]. In these papers, the relationships between the attributes describing the current status and the effective operators are learned using training data obtained during the exhaustive search phase. Neural network [2], state-space representation [3], or decision tree [4] is utilized as a learning schema. In our prototype system, neural network is utilized because of its flexibility in representing the relationships between IF and THEN parts.

5. CURRENT STATUS OF ACTUAL IMPLEMENTATION

University of Tokyo and MELCO are now cooperatively implementing the actual intelligent design support system based on this concept. This is the first step activity in MELCO towards the ultimate goal of building an integrated satellite design infrastructure including mission design, early conceptual design, detailed design and analysis [7]. The target will include wide variety of satellites from small to large as well as

LEO to GEO. The key requirements for the system are that:

- 1) it can quickly generate many alternative designs, even if it is not optimum solutions, assuming its usage in the early conceptual design phase,
- 2) it can analyze the current design and make suggestions as to the inconsistency and parameter modifications, and
- 3) it can incorporate various in-house design routines already coded or will be coded by the satellite design experts.

The system consists of C++ functions of autonomous design routines, design process editors and evaluation modules as well as spread sheet type human interfaces.

The current status of development is that the first version of the system which uses only human expertise for IF-THEN rules of parameter modifications are completed, and the machine learning capability is now being implemented. The "design process edit" function is now implemented in the form of modification of code level, not on the graphical user interface.

6. CONCLUSIONS AND FUTURE WORKS

Integrated design support infrastructure and computer intelligence can be said essential to accelerate and improve the otherwise quite complicated and time consuming satellite design process, especially in the near future when more and more varied satellites should be designed in much shorter time. The operations in the design process for which computer intelligent support is the most indispensable are deciding the sequence of subsystem designs and the parameter modification strategy when the current design should be modified. The proposed concept provides one approach to how these kinds of intelligence is obtained and in what way it is implemented.

Much work should be done towards the really useful design support system: in the research level, the incorporation of machine learning is the main research item. Especially how to represent the attributes to describe the current status and how to obtain sufficient training data for learning will be the main issues. In technical level, graphical user interface for design process editor as well as IF-THEN rule editors should be implemented. We are continuing research in these directions.

REFERENCES

- [1] T.Yoshida, K.Hori and S.Nakasuka, A Cooperation Method via Metaphor of Explanation, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E81-A, No.4, (1998)
- [2] S.Nakasuka and T.Ninomiya, Applications of Dynamic Scheduling Technique to Space Related Problems - Some Case Studies, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.423-426, (1994)
- [3] S.Nakasuka, Machine Intelligence in Control System Design - Its Possibility and Methodology, Transactions of the Japan Society for Aeronautical and Space Sciences, Vol.32, No.98, pp.206-216, (1990)
- [4] S.Nakasuka and T.Yoshida, Dynamic Scheduling System Utilizing Machine Learning as a Knowledge Acquisition Tool, International Journal of Production Research, Vol.30, No.2, pp.411-431, (1992)
- [5] T.Obata, T.Watanabe, R.Tachibana, K.Hori and S.Nakasuka, Concept of Intelligent Design Support System for Satellites, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.343-346, (1997)
- [6] D.Goldin, S.Venneri and A.Noor, New Frontiers in Design Synthesis, IAF-98-U.2.01, 49th IAF Congress, (1998)
- [7] K.Ijichi, T.Fuji, T.Anzai, M.Sato, H.Hasegawa and T.Yamaguchi, The Advanced Small Satellite Bus System with Integrated Spacecraft Controller, IAF-98-U.1.05, 49th IAF Congress, (1998)

An AI Approach to Ground Station Autonomy for Deep Space Communications

Forest Fisher, Tara Estlin, Darren Mutz, and Steve Chien

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
phone: +1 818 393-5368, fax: +1 818 393-5244, email: {firstname.lastname}@jpl.nasa.gov

ABSTRACT

This paper describes components of a system for an autonomous deep space tracking station. The system enables fully automated routine operations encompassing scheduling and resource allocation, antenna and receiver predict generation, track procedure generation from service requests, and closed loop control and error recovery for the station subsystems. This system has been validated by the construction of a prototype Deep Space Terminal (DS-T) tracking station, which has performed a series of demonstrations of autonomous ground station control for downlink services with NASA's Mars Global Surveyor (MGS).

INTRODUCTION

The Deep Space Network (DSN) [8] was established in 1958 and has since evolved into the largest and most sensitive scientific telecommunications and radio navigation network in the world. The purpose of the DSN is to support unmanned interplanetary spacecraft missions and to support radio and radar astronomy observations taken in the exploration of space. The DSN currently consists of three deep-space communications facilities placed approximately 120 degrees apart around the world: at Goldstone, in California's Mojave Desert; near Madrid, Spain; and near Canberra, Australia. This strategic placement permits constant observation of spacecraft as the Earth rotates, and helps to make the DSN the largest and most sensitive scientific telecommunications system in the world. Each DSN complex operates a set of deep space stations consisting of 70-meter, 34-meter, and 26-meter antennas. The function of the DSN is to receive telemetry signals from spacecraft, transmit commands that control spacecraft operating modes, generate the radio navigation data used to locate and guide a spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry (VLBI), and geodynamics measurements.

From its inception the DSN has been driven by the need to create increasingly more sensitive telecommunications

devices and better techniques for navigation. The operation of the DSN communications complexes requires a high level of manual interaction with the devices in the communications link with the spacecraft. In more recent times NASA has added some new drivers to the development of the DSN:

1. reduce the cost of operating the DSN,
2. improve the operability, reliability, and maintainability of the DSN, and
3. prepare for a new era of space exploration with the New Millennium program: support small, intelligent spacecraft requiring very few mission operations personnel [14].

In the past, the process of operating such stations has been labor and knowledge intensive. Recently, efforts have been made to reduce the cost of operations. One such effort has been in the area of antenna station automation. Many approaches have been applied to automation control/commanding of different types of antenna systems. In the AI group at JPL, we have worked on automating the scheduling of communications antennas and the generation of antenna command sequences. The scheduling of communications antennas consist of allocating an oversubscribed resource, the antenna, to a flight project in order to provide communication services, while antenna command sequences set up and perform a particular communications link with a spacecraft [2]. These sequences can be run as control scripts to operate the station and all of its relevant subsystems [14]. This work was demonstrated as a component of the Deep Space Terminal (DS-T) during a series of demonstrations from April to September of 1998. Through the use of these technologies a high level goal-oriented interface is provided to the system. This interface enables users to specify *what* they want done and does not require that they specify or even know *how* it should be done.

The rest of this paper is organized in the following manner. We first offer a brief background on how the DSN operates. Next we provide an introductory explanation of the DS-T functionality. From here we

discuss two of the underlying technologies providing much of the DS-T's autonomy: automated scheduling and automated planning. We then conclude the paper with results from our demonstrations and talk about future work.

How the DSN Works

The DSN track process occurs daily for dozens of different NASA spacecraft and projects, which use the DSN to capture spacecraft data. Though the process of sending signals from a spacecraft to Earth is conceptually simple, in reality there are many earthside challenges that must be addressed before a spacecraft's signal is acquired and successfully transformed into useful information. In the remainder of this section, we outline some of the steps involved in providing tracking services and in particular discuss the problem of track plan generation.

The first step in performing a DSN track is called *network preparation*. Here, a project sends a request for the DSN to track a spacecraft involving specific tracking services (e.g., downlink, uplink). The DSN responds to the request by attempting to schedule the necessary resources (i.e. an antenna and other shared equipment) needed for the track. Once an equipment schedule and other necessary information has been determined, the next step is the *data capture process*, which is performed by operations personnel at the deep space station. During this process, operators determine the correct steps to perform the following tasks: configure the equipment for the track, perform the actual establishment of the communications link, and then perform the actual track by issuing control commands to the various subsystems comprising the link.

Throughout the track the operators continually monitor the status of the link and handle exceptions (e.g., the receiver breaks lock with the spacecraft) as they occur. All of these actions are currently performed by human operators, who manually issue hundreds of commands via a computer keyboard to the link subsystems.

This paper discusses the application of two AI systems for automated antenna operations. These systems are an AI scheduling system for allocating communications resources, and an AI planning system to generate deep space communication antenna control scripts. These two components are intended to dramatically reduce the need for many manual steps.

Deep Space Terminal

The components discussed in this paper were demonstrated as part of the Deep Space Terminal (DS-T),

a prototype 34-meter deep space communications station developed [9][10][11] as a technology demonstration of fully autonomous *lights-out* operations. In the DS-T concept, a global DSN schedule is disseminated to a set of autonomous DS-T stations, where each DS-T station operates autonomously, performing tracks in a largely independent fashion. When requested to perform a track, the DS-T station performs a number of tasks (at appropriate times) required to execute the track. First, the DS-T station uses appropriate spacecraft navigation ephemeris and predict generation software in order to produce necessary antenna and receiver predict information required to perform the track. Next, the DS-T station executes the pre-calibration process, in which the antenna and appropriate subsystems (e.g., receiver, exciter, telemetry processor, etc.) are configured in anticipation of the track. During the actual track, the signal from the spacecraft must be acquired and the antenna and subsystems must be commanded to retain the signal, adjust for changes in the signal (such as changes in bit rate or modulation index as transmitted by the spacecraft), and perform error recovery. Finally, at the completion of the track, the station must be returned to an appropriate standby state in preparation for the next track. All of these activities require significant automation and robust execution including closed loop control, retries and contingency handling.

In order to provide this autonomous operation capability, the DS-T station employs tightly coupled state of the art hardware and software. At the core of the autonomy are two areas of artificial intelligence (AI) technology, AI scheduling and AI planning. We will offer a brief example of each and a brief context for how they apply to the DS-T.

The original goal of the DS-T task was to build an autonomous control system for a deep space communications station. This system had to meet the following criteria: schedule driven with a high level service request interface; an automated scheduling component for initial scheduling and rescheduling; provide script guided control; ability to generate predicts or use provided predicts; automatically configure pre-track; utilization of COTS (Commercial Off The Shelf) components wherever feasible; operations based on defined but expandable set of services; autonomous error recovery for a defined class of problems; post pass data delivery; and treat ground terminal as a network computer with an RF peripheral.

One of the most important points was the idea of a ground station looking just like a network computer to a user, operator, or mission. This is best demonstrated by an

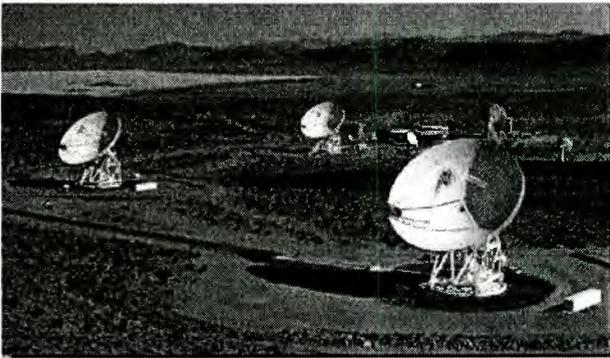


Figure 1: 34m BWG Antennas at Goldstone

operational scenario. To provide service a user need only login to the DS-T work-station and submit a service request to the scheduling system, or FTP a schedule and service request to a particular file system location. From either of these inputs DS-T would detect the existence of a track/service schedule, proceed to schedule station specific tasks, configure the station to provide the service, and finally when the time comes, the track would begin without further user interaction.

As mentioned above, the station reacts to a service request derived schedule generated by an automated scheduling system. It is through the reaction to this schedule that the dynamic track-specific control scripts are generated. Autonomous operations of the station takes place through the execution of these control scripts.

In Figure 1, we show a picture of the three 34-meter Beam Wave Guide antennas at Goldstone, CA. In the foreground is DSS-26, which was the station selected for prototyping the DS-T.

In April 1998, the DS-T prototype first demonstrated automated downlink capability of single isolated tracks for the Mars Global Surveyor (MGS) spacecraft. Between April and September 1998, many multi-day demonstrations took place including a six day unattended demonstration. During these demonstrations, a service request for downlink services, a track sequence of events, and spacecraft ephemeris data were used to automatically downlink data from the MGS spacecraft.

Scheduling for DS-T

When the decision is made to fly a mission, a very knowledge-intensive process begins that will ensure the necessary DSN antenna coverage. First, a forecast is made of the DSN resources that the spacecraft will require. In the Resource Allocation Process (RAP), the types of services, frequency, and duration of the required tracks are determined as well as high-level re-source

requirements (e.g., antenna). While the exact timing of the tracks is not known, a set of automated forecasting tools are used to estimate network load and to assist in ensuring that adequate network resources will be available. One part of the network architecture is a unified tool suite that has been developed called TMOD Integrated Ground Resource Allocation System (TIGRAS), which uses operations research and probabilistic reasoning techniques to allow forecasting and capacity planning for DSN resources [1].

As the time of the actual tracks approaches, this estimate of resource loading is converted to an actual schedule, which becomes more concrete as time progresses. In this process, specific project service requests and priorities are matched up with available resources in order to meet communications needs for earth-orbiting and deep space spacecraft. This scheduling process involves considerations of thousands of possible tracks, tens of projects, tens of antenna resources and considerations of hundreds of subsystem configurations. In addition to adding the detail of antenna subsystem allocation, the initial schedule undergoes continual modification due to changing project needs, equipment availability, and weather considerations. Responding to changing context and minimizing disruption while rescheduling is a key issue.

At the high level of resource allocation, schedule execution does not involve execution monitoring. However, rescheduling is often necessary due to: equipment outages, last minute track requests, last minute changes to scheduled tracks, and changing atmospheric conditions. Rescheduling can occur in two ways: (1) it can be initiated top-down due to a change to a previously scheduled track or the addition of another request; and (2) it can occur bottom-up in that equipment outages can occur or tracks can fail necessitating rescheduling. From the standpoint of the scheduler the important feature is the degree of change required to make the schedule consistent.

Because of the size and complexity of the rescheduling task, manual scheduling is prohibitively expensive. Automation of these scheduling functions is projected to save millions of dollars per year in DSN operations costs. Based on these motivating factors, the Demand Access Network Scheduler (DANS), which was designed to deal with the complex subsystem and priority schemes required to schedule the 34 and 70 meter antennas, was used as one of the scheduling components of the DS-T.

DANS: Automated Scheduling

The Demand Access Network Scheduler (DANS) [3] system, designed to deal with the complex subsystem and priority schemes required to schedule the larger 34 and 70 meter antennas, uses the forecasted antenna schedule produced by the RAP process and supports rescheduling as required by changing tracking requirements and equipment availability. The main inputs to DANS are the current schedule and a set of new tracking requests or changes to current tracks and/or equipment that must be handled in the final schedule. A tracking request usually specifies information such as the spacecraft or project name (e.g., DS1, Voyager), the type of antenna requested (e.g., 70M, 34M), the number of individual tracks requested (e.g., 4 tracks per week), the start time and end time for each track, priority of each track, etc.

DANS uses priority-driven, best-first, constraint-based search and iterative optimization techniques to perform priority-based rescheduling in response to changing network demand. In this approach, DANS first considers the antenna allocation process, since antennas are the central focus of resource contention. After establishing a range of antenna options, DANS then considers allocation of the 5-13 subsystems per track (out of the tens of shared subsystems at each antenna complex) used by each track. DANS uses constraint-driven, branch and bound, best-first search to efficiently consider the large set of possible subsystems schedules. The DANS objective is to satisfy as many activity requests as possible while maintaining a conflict-free status (i.e. no hard constraints violated) with minimal disruption to the existing schedule.

The DSN scheduling problem is complicated by three factors: (1) context-dependent priority; (2) subsystem allocation; and (3) the possibility of reducing the length of the tracks. DSN track priorities are context dependent in that they are often contingent on the amount of tracking the project has received so far in the week. For example, a project might have priority 3 to get 5 tracks, priority 4 to get 7 tracks and priority 6 to get 9 tracks (where lower priority numbers represent more important tracks). This reflects that 5 tracks are necessary to maintain spacecraft health and get critical science data to ground stations; 7 tracks will allow a nominal amount of science data to be downlinked; and 9 tracks will allow for downlinking of all science data (e.g., beyond this level additional tracks have little utility). An important point is that specific tracks are not labeled with these priorities (e.g., the project is allowed to submit 5 tracks at priority 3, 2 at priority 4 and so on). Rather, when considering adding, deleting, or

moving tracks the scheduler must consider the overall priority of the project in the current allocation context.

Finally, the DSN scheduling problem is complicated by the fact that the track duration can be relaxed. For example, a project may request a 3 hour track but specify a minimum track time of 2 hours. When evaluating potential resource conflicts the scheduler must consider the option of shortening tracks to remove resource conflicts. DANS also uses a linear weighting scheme in conjunction with a modified SIMPLEX algorithm to trim tracks in accordance with prioritizations.

Once generated, a schedule is first used at a network wide level designating what resources (primarily the antennas) shall be used to provide what services (primarily communications tracks). In the DS-T architecture the schedule is then disseminated to each DS-T station to designate when and what type of service is to be performed by that station. From this high level description of the service, each DS-T station proceeds to schedule station specific activities in order to provide the desired services. This secondary station specific scheduling component utilized a simple macro expansion scheduling algorithm and controls the execution of the schedule. These activities consist of track script generation and execution of the track script for each track.

Planning for DS-T

Once a DS-T station has been allocated to provide the communications service for a particular mission (i.e. has been scheduled) and the station specific activities have been scheduled, the DS-T script generator is invoked just prior to the track beginning in order to generate the antenna control script. The final result is the set of antenna commands necessary to setup and perform the request track (communication service).

The DS-T script generator (SG) is where the majority of the control autonomy is provided. The SG uses Artificial Intelligence planning techniques to perform a complex software module reconfiguration process [5]. This process consists of piecing together numerous highly interdependent smaller control scripts in order to produce a single script to control the operations of the DS-T station.

The core engine used in the SG is the Automated Scheduling and Planning ENVIRONMENT (ASPEN) [13]. The ASPEN system is a reusable, configurable, generic planning/scheduling application framework that can be tailored to specific domains to create conflict-free plans or schedules. It has a number of useful features including an

expressive modeling language, a constraint management system for representing and maintaining antenna operability and/or resource constraints, a temporal reasoning system and a graphical interface for visualizing plans and states. ASPEN has been adapted to input antenna-tracking goals and automatically produce the required command sequence necessary to create the requested link [12].

The control script produced by the SG:

- sets up the track by configuring the station during pre-track;
- provides the track service requested by commanding the antenna and sub-systems to acquire and maintain lock on the signal throughout mode changes; and
- cleans up and shuts down the station at the completion of the track.

TRACK PLAN GENERATION: THE PROBLEM

Generating an antenna track plan involves taking a general service request (such as telemetry - the downlink of data from a spacecraft), an antenna knowledge-base (which provides the information on the requirements of antenna operation actions), and other project specific information (such as the spacecraft sequence of events), and then generating a partially-ordered sequence of commands. This command sequence will properly configure a communications link that enables the appropriate interaction with the spacecraft. To automate this task, the ASPEN planning and scheduling system has been applied to generate antenna operation procedures on demand.

ASPEN has been adapted to use high-level antenna track information to determine the appropriate steps, parameters on these steps and ordering constraints on these steps that will achieve the input track goals. In generating the antenna track plan, the planner uses information from several sources. In Figure 2 we show the inputs and output of the DS-T script Generator.

Equipment Configuration - This configuration details the types of equipment available and includes items such as the antenna, antenna controller, the receiver, etc.

Project Service Request - The service request specifies the DSN services (e.g., downlink, uplink) requested by the project and corresponds to the goals or purpose of the track.

Project SOE - The project sequence of events (SOE) details spacecraft events occurring during the track - including the timing of the beginning and ending of the

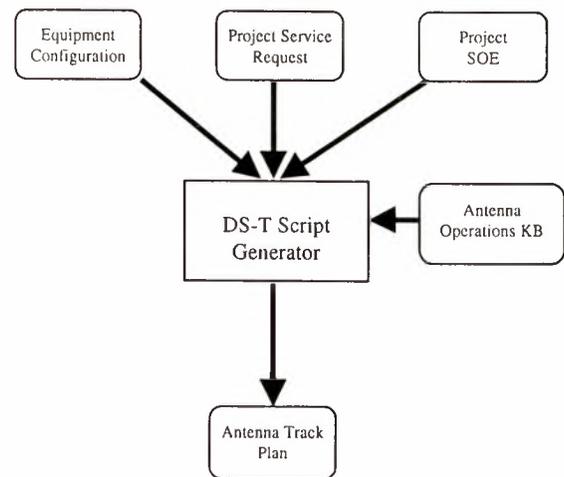


Figure 2: DS-T Script Generator Inputs and Outputs

track and spacecraft data transmission bit rate changes, modulation index changes, and carrier and subcarrier frequency changes.

Antenna Operations KB - The Antenna Operations Knowledge Base (KB) stores information on available antenna operations actions/commands. This KB dictates how actions can be combined to provide essential communication services. Specifically, this includes information such as action preconditions, postconditions, and command directives and also includes any other relevant information such as resource and state descriptions.

Antenna Track Plan - The Antenna Track Plan is the output of the ASPEN/DS-T Script Generator. The track plan is the dynamically produced command control script. When executed these scripts issue all of the necessary subsystem command directives to configure, control, and perform the communications track.

Through the use of the ASPEN/DS-T Script Generator these high level inputs provide the goal-oriented interface enabling the system to be used by specifying *what* is to be done instead of *how* it should be done.

DS-T Demonstrations

The Deep Space Terminal (DS-T) [10][11] concept was validated through a number of demonstrations. These began with the automation of partial tracks in April 1998, continued with 1-day unattended operations in May, and concluded with a 6-day autonomous "lights-out" demonstration in September 1998. Throughout these demonstrations ASPEN was used to automatically generate the necessary command sequences for a series of

Mars Global Surveyor (MGS) downlink tracks using the equipment configuration at Deep Space Station 26 (DSS26), a 34-meter antenna located in Goldstone, CA. These command sequences were produced and executed in a fully autonomous fashion with no human intervention. During the September demonstration, DS-T performed all Mars Global Surveyor coverage scheduled for the Goldstone antenna complex. This corresponded to roughly 13 hours of continuous track coverage per day.

While the overall DS-T effort consisted of a large team and a project duration of approximately 1.5 years, the DS-T automation team consisted of three team members. Of this team's work, approximately one work year was spent on the script generation effort. This effort primarily consisted of knowledge acquisition and model development, while a small effort was made in the integration of the script generator. A key factor in the quick development was the ability to adapt a general purpose planning and scheduling system. As the domain of ground communication-station commanding shared many similarities to spacecraft commanding, ASPEN seemed like a logical choice. This was confirmed by the ease of knowledge base development and integration. Spacecraft commanding also consists of generating a sequence of commands, however it is predominately a resource-scheduling problem, whereas ground-station commanding is predominately a sequencing problem.

Results

In order to provide qualitative results, we present statistical data from September 16, 1998, a representative day during our 6-day autonomous unattended demonstration, during which we collected above 90% of the transmitted frames. This performance is on par with the operator-controlled stations, however required no support personnel (i.e. reduced operations cost).

In Figure 3, the graph represents when MGS was in view of the ground stations at each of the three complexes (Madrid, Goldstone, and Canberra). DS-T, which is located at Goldstone, tracked MGS through the five track segments indicated in Figure 3.

Before continuing with the analysis of the results, let us explain the different modes indicated in Figure 3 for each of the different track segments. When a spacecraft is downlinking data it is said to be in 1way mode. When an uplink and a downlink are taking place simultaneously the spacecraft is said to be in 2way mode. If a station is communicating in 2way mode with a spacecraft, and another station is listening in on the downlink of the spacecraft, the second station is said to be in 3way with

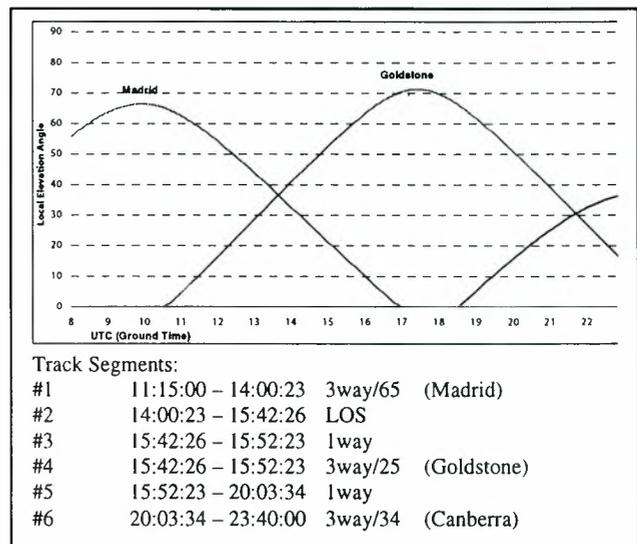


Figure 3: September 16, 1998 MGS Track

the 2way station. Because DS-T is not equipped for uplink, DS-T operates in either 1way or 3way mode. In this example, during segment 4 dss25 (deep space station) was in 2way and DS-T was in 3way with 25 (3way/25).

Track segment 2, which is labeled LOS, indicates that there was a scheduled loss of signal (LOS) so during this segment no frames were collected. During each of the other respective track segment DS-T collected 75%, 91%, 96%, 90%, 23% of the broadcasted frames. As shown by the graph, during segment 1 and 6 the elevation of the dish is low in the sky. Under these circumstances there is considerably more atmospheric interference which explains the lower percent of frame collection. On the other hand, if you look at segment 4, where there is a long segment with the spacecraft high in the sky, the data collection is quite high. In segment 3 and 5 the values are a little lower due to the shortness of the segments. This is explained by the fact that some data is lost during a change in mode, as in the transition from LOS to 1way and 3way/25 to 1way.

As a component of the DS-T, the DANS automated scheduler enabled us to demonstrated how a network of DS-T like terminals would perform in a schedule driven environment. It is partially through this functionality of starting from a high-level service request and producing a resource allocation schedule that the DS-T concept is able to provide communications service through a high level interface. In conjunction with the scheduling system the DS-T Script Generator performed flawlessly, producing dynamically instantiated control scripts based on the desired service goals for the communications pass as

specified in the service request. The use of such technology resulted in a three primary benefits:

- Autonomous operations enabled by eliminating the need for hundreds of manual inputs in the form of control directives. Currently the task of creating the communications link is a manual and time-consuming process which requires operator input of approximately 700 control directives and the constant monitoring of several dozen displays to determine the exact execution status of the system.
- Reduced the level of expertise of an operator required to perform a communication track. Currently this complex process requires a high level of expertise from the operator, but through the development of the knowledge base by a domain expert this expertise is captured within the system itself.
- The knowledge base provides a declarative representation of operation procedures. Through the capture of this expertise the knowledge base documents the procedural steps of performing antenna communication services.

Related Work

There are a number of existing systems built to solve real-world planning or scheduling problems [15][16][17]. The problem of track plan generation combines elements from both these fields and thus traditional planners and schedulers cannot be directly applied. First, many classical planning elements must be addressed in this application such as subgoaling to achieve activity preconditions (e.g., the antenna must be "on_point" to lock up the receiver) and decomposing higher-level (abstract) activities into more detailed sub-activities. In addition, many scheduling elements are presents such as handling metric time and temporal constraints, and representing and reasoning about resources (e.g., receiver, antenna controller) and states (e.g., antenna position, subcarrier frequency, etc.) over time.

One other system has been designed to generate antenna track plans, the Deep Space Network Antenna Operations Planner (DPLAN) [4]. DPLAN utilizes a combination of AI hierarchical-task network (HTN) and operator-based planning techniques. Unlike DPLAN, ASPEN has a temporal reasoning system for expressing and maintaining temporal constraints and also has the capability for representing and reasoning about different types of resources and states. ASPEN can utilize different search algorithms such as constructive and repair-based algorithms, where DPLAN uses a best-first search. And, as described in the next section, ASPEN is currently being extended to perform dynamic planning for closed-loop

error recovery, where DPLAN has only limited replanning capabilities.

As for the resource allocation type of scheduling performed by DANS, traditional scheduling system are not sufficient because of the unique type of constraints that the DSN scheduling problem poses.

A previous DSN scheduling system, OMP-26, was designed to perform the scheduling process for the smaller 9, 11, and 26 meter antennas. While the use of OMP-26 resulted in a five-fold reduction in scheduling labor and a doubling of network usage, OMP-26 does not have the ability to deal with the longer term forecasting required in the larger antenna network.

Another system developed for the resource allocation process in the DSN is the previously mentioned TIGRAS system. While TIGRAS has powerful tools for the visualization of network load and tools to assist with network forecasting, TIGRAS was not designed to perform automated rescheduling as was DANS nor demand access scheduling.

Future Work Providing Closed-Loop Control through Dynamic Planning

Currently, we are working on modifying and extending the current ASPEN Track Plan Generator to provide Closed Loop Execution and Recovery (CLEaR) for DSN track automation. CLEaR is built on top of CASPER [7], a real-time planning system built as an extension to ASPEN. The approach taken is to dynamically feed monitor data (sensor updates) back into the planning system as state updates. As these dynamic updates come in, the planning system verifies the validity of the current plan. If a violation is found in the plan, the system will perform local modification to construct a new valid plan. Through this continual planning approach [6], the plan is disrupted as little as possible and the system is much more responsive and reactive to changes in the real (dynamic) world.

As part of the CLEaR effort further research is being done in the area *mixed-initiative control*. This addresses the interaction of an operator with, for all intensive purposes, an autonomous system. In these circumstances a planning and execution engine must maintain consistency with in the engine if an operator overrides the system so that once the operator returns the system to nominal operations the system is able to resume control without missing a heartbeat.

This CLear effort is also being integrated with a Fault Detection, Isolation and Recovery (FDIR) system. FDIR is an expert system providing monitor data analysis. As is often the case with large complex systems, monitor (sensor) data is often related in different ways that becomes difficult for a human to detect. The advantage of combining these two systems is that FDIR can first interpret the vast amount of data and summarize it into a set of meaningful values for a planning system to react to. We think of this union as intelligent analysis and intelligent response, much like a careful design and implementation; one without the other is of little use.

Conclusion

This paper has described the concept of the Deep Space Terminal (DS-T) and two of the key enable components in the DS-T autonomous operations capabilities. We first introduced the DSN problem domain and the DS-T 34-meter prototype antenna communications station. Next we described in detail the Demand Access Network Scheduling (DANS) system used to perform resource allocation/scheduling and the DS-T/ASPEN Script Generator used for antenna control script generation. We then concluded with results of the DS-T autonomous "lights out" operations demonstrations, discussion on related work, and presented some insight to future work being done in the area of DSN automation.

REFERENCES

- [1] Borden, C., Wang, Y-F., and Fox, G., "Planning and Scheduling User Services for NASA's Deep Space Network," *Working Notes of the 1997 International Workshop on Planning and Scheduling for Space Exploration and Science*, Oxnard, CA, 1997.
- [2] Chien, S., Decoste, D., Doyle, R., and Stolorz, P., "Making an Impact: Artificial Intelligence at the Jet Propulsion Laboratory," *AI Magazine*, 18(1), 103-122, 1997.
- [3] (Chien et al. 1997b) Chien, S., Lam, R., and Vu, Q., "Resource Scheduling for a Network of Communications Antennas," *Proceedings of the 1997 IEEE Aerospace Conference*, Aspen, CO, February, 1997.
- [4] Chien, S., Hill Jr., R., Govindjee, A., Wang, X., Estlin, T., A. Griesel, R. Lam and K. Fayyad, "A Hierarchical Architecture for Resource Allocation, Plan Execution, and Revision for Operation of a Network of Communication Antennas," *Proceedings of the 1997 IEEE Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- [5] Chien, S., Fisher, F., Mortensen, H., Lo, E., Greeley, R., Govindjee, A., Estlin, T., and Wang, X., "Using Artificial Intelligence Planning Techniques to Automatically Reconfigure Software Modules," in *Tasks and Methods in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol. 1416, Angel Pasqual del Pobil, Jose Mira, and Moonis Ali (Eds.), Springer-Verlag, 1998, pp. 415-426.
- [6] Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," *IJCAI99 Workshop on Scheduling and Planning meet Real-time Monitoring in a Dynamic and Uncertain World*, Stockholm, Sweden, August 1999.
- [7] Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Integrated Planning and Execution for Autonomous Spacecraft," *Proceedings of the 1999 IEEE Aerospace Conference*, Aspen, CO, March 1999.
- [8] Deep Space Network, *Jet Propulsion Laboratory Publication 400-517*, April 1994.
- [9] Estlin, T., Fisher, F., Mutz, D., Chien, S., "Automated Planning for a Deep Space Communications Station", *Proceedings of the IEEE Conference on Robotics and Automation*, Detroit, Michigan, May 1999.
- [10] Fisher, F., Chien, S., L. Paal, E. Law, N. Golshan, and M. Stockett, "An Automated Deep Space Communications Station," *Proceedings of the IEEE Aerospace Conference*, Aspen, CO, March 1998.
- [11] Fisher, F., Mutz, D., Estlin, T., L. Paal, and Chien, S., "The Past, Present and Future of Ground Station Automation with in the DSN," *Proceedings of the IEEE Aerospace Conference*, Aspen, CO, March 1999.
- [12] Fisher, F., Estlin, T., Mutz, D., and Chien, S., "Using Artificial Intelligence Planning to Generate Antenna Tracking Plans," *Eleventh Conference on Innovative Applications of Artificial Intelligence*, Orlando, Florida, July 1999.
- [13] Fukunaga, A., Rabideau, G., Chien, S., and Yan, D., "Toward an Application Framework for Automated Planning and Scheduling," *Proceedings of the 1997 International Symposium of Artificial Intelligence, Robotics and Automation for Space (ISAIRAS-97)*, Tokyo, Japan, July 1997
- [14] Hill, Jr., R., Chien, S., Fayyad, K., Smyth, C., Santos, T., and Bevan, R., "Sequence of Events Driven Automation of the Deep Space Network," *Telecommunications and Data Acquisition 42-124*, October-December 1995.
- [15] Tate, A., Drabble, B., and Kirby, R., "O-Plan2: An Open Architecture for Command Planning and Control," *Intelligent Scheduling* (Eds. M. Fox & M. Zweben), Morgan Kaufmann, 1994.
- [16] Wilkins, D., "Extending the Classical AI Planning Paradigm," *Practical Planning*, Morgan Kaufmann, 1994.
- [17] Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," *Intelligent Scheduling*, Zweben, M., and Fox, M., eds., Morgan Kaufmann, 1994, pp.241-256.

INTELLIGENT OPTICAL POLARIMETRY DEVELOPMENT FOR SPACE SURVEILLANCE MISSIONS

Lenore McMackin, Paul Zetocha, Clint Sparkman

Air Force Research Laboratory, Space Sensing and Vehicle Control Branch
AFRL/VSSS, Kirtland AFB, NM 87117 USA
Phone: (505) 853-4114 Fax: (505) 846-6053 Email: zetocha@plk.af.mil

Harold McIntire, Matthew Fetrow, Kenneth Bishop

Applied Technology Associates, Inc
1900 Randolph Rd., Albuquerque, NM 87106
Phone: (505) 846-6110 Fax: (505) 846-6132 Email: fetrow@ataway.aptec.com

ABSTRACT

The rapidly increasing numbers and complexity of earth-orbiting satellites in recent decades has placed heavy demands upon telemetry and ground support equipment and personnel to maintain and control the systems. A major thrust of current space program developments is the reduction of dependence upon ground control for normal satellite operations. This paper describes one such experiment currently under development at the Air Force Research Laboratory (AFRL) which addresses these needs. The experiment combines an optical polarimeter for measurement of multi-spectral polarization signals of orbiting objects and a system of intelligent software agents which will provide automated payload and bus control. We discuss the development of the optical system hardware, software agent development, and aspects of the processing and control of information from on-board data.

I. INTRODUCTION

Intelligent satellite systems have the potential to transform space-based surveillance and scientific missions from the current ground-based, manpower-intensive approach to fully autonomous satellite systems capable of independent data acquisition, system control, and data processing, on-board cueing and mission reconfiguration, selective dissemination of information to users, and anomaly detection and correction. These systems will be able to acquire and process data and use information extracted from that data to control on-board systems such as attitude and orbital position, or to cue other space or ground systems automatically.

Autonomous systems would find immediate use in a multitude of applications involving space-based surveillance of both space and ground objects.

This paper addresses technological challenges of constructing and deploying an intelligent space-based target cueing device that operates on principles of optical polarimetry, processes data on board using smart sensing technology, and is guided autonomously by intelligent software agents. Specifically, we aim to apply previous and current basic research in material and shape characterization based on low spatial resolution polarization signals typical of small polarization imaging systems to the design of an intelligent space polarimeter. We will do this by adapting a four-channel polarimeter already in use at the Air Force Research Laboratory (AFRL) to a space platform such as the MightySat II.2 satellite while controlling data acquisition, smart polarization state processing, target detection and cueing signal generation tasks using intelligent agent software architectures.

The ultimate aim of the proposed work is to produce a small, lightweight, inexpensive cueing device that can narrow the potential field for more data intensive surveillance systems saving time, unnecessary computation and bandwidth. This paper will discuss automated control systems for a prototype space polarimetry experiment and how an optical payload interfaces with other pertinent satellite subsystems. Subsystem models used in the prototype include components of the attitude control, propulsion, and sensor payload will then be described along with their interrelationship and a description of the collaboration mechanisms used. Various simulation scenarios are

currently being devised to exercise the prototype system. Descriptions of these simulations will then be given. This will include descriptions of the various AI modules used and the benefits to the overall reasoning process that resulted. Our results will be discussed in the context of the most likely operational role for polarimetry in space.

We will begin with a discussion of the fundamentals of optical polarimetry. An overview of the intelligent agent architecture and the framework used for intelligent polarimetric data processing is then given. The autonomous planning capability will then be described and results from its incorporation into the agent architecture will be highlighted. The paper will then summarize the strengths and weaknesses of the agent approach to the enhancement of on-board spacecraft autonomy. Lastly the paper will conclude with a summary of our future plans for this prototype system.

2. POLARIMETRY

The polarization of reflected and emitted optical radiation is highly dependent on material properties and is also influenced by orientation and surface roughness. Measurement of polarization can therefore be expected to yield additional information about target surfaces features, shape and configuration without necessarily requiring high spatial resolution, greatly simplifying space optical systems.

The polarization state of an electromagnetic wave is fully described by a set of 4 parameters known as the Stokes parameters. Collectively, these parameters comprise the Stokes vector.

$$\mathbf{S} = \{S_0, S_1, S_2, S_3\} \quad (1)$$

Measurement of the polarization is typically performed by measuring four basic intensity values: I_0 , the intensity of all polarization states; I_1 , horizontal linear polarization, I_2 , polarization at $+45^\circ$, and I_3 , the right circular polarization component. Each of these is measured by filtering the incident light. These intensity values are converted to the Stoke's vector parameters by a simple linear combination of these intensity measurements,

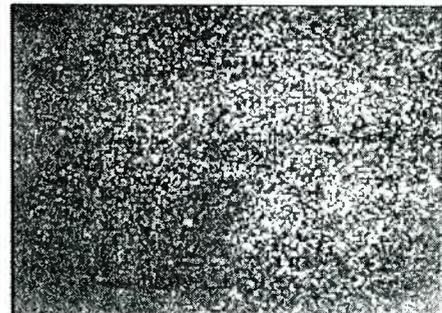
$$\begin{aligned} S_0 &= 2I_0 \\ S_1 &= 2I_1 - 2I_0 \\ S_2 &= 2I_2 - 2I_0 \\ S_3 &= 2I_3 - I_0. \end{aligned} \quad (2)$$

From the Stokes components, the degree of polarization may be computed using simple math,

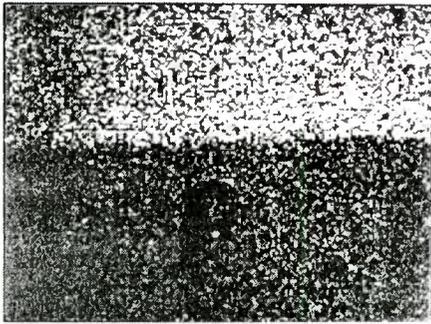
$$DoP = \frac{\sqrt{S_1^2 + S_2^2 + S_3^2}}{S_0} \quad (3)$$

Visible and IR polarimetry may provide improvements in the ability to distinguish targets from natural backgrounds due to sharply contrasting polarization signatures of man-made and natural materials. The figure below illustrates the ability of polarization to distinguish between visually similar materials. When imaged in linearly polarized light white-painted aluminum is virtually indistinguishable from white paper. However, when illuminated with circularly polarized light the distinction between the materials becomes clear.

In addition, the retardance, which is a measure of the phase difference between two orthogonal polarization components of the light, has been shown to discriminate between metallic and dielectric materials in machine vision applications [1]. Figure 1 is not an image in the traditional sense of intensity but is a "pseudo image" visualization of the calculated polarization states of each pixel. It should be noted that the quality of polarization pseudo images is not highly dependent on the contrast ratio of the intensity measurement and much information can be obtained even from low signal to noise measurements.



(a)



(b)

Fig. 1. (a) In linear polarization it is visually difficult to discriminate white paint from paper. (b) Active illumination with and detection of circular polarization component, S_3 , easily reveals the different materials. The top half of each image is painted, the bottom half is paper.

The primary objective of our polarimeter will be to collect polarization signals from targets that are not necessarily spatially resolved. Analysis of the signal data will be used to evaluate the use of polarization for detecting, classifying and identifying optically unresolved objects from autonomously operating space platforms. While the polarimetry data and analysis results are expected to be of scientific interest in their own right, the principal purpose of the experiment is to demonstrate the use of intelligent agents for mission autonomy in space. For mission autonomy to be feasible optical systems must operate autonomously, acquiring data, calculating polarization vectors for each pixel, detecting possible targets based on the signals and utilizing target information to reconfigure or cue other satellite subsystems. Therefore, optical systems must be interfaced with other satellite functions such as attitude control and orbit control systems to achieve mission autonomy. The use of intelligent agents to control optical data acquisition and processing will be demonstrated.

3. INTELLIGENT AGENTS

The majority of approaches to satellite autonomy are piecemeal with specific techniques applied to given activities. Decision making is generally based on information from a subset of the total information available and generally does not take into account the status of other components and mission objectives. In reality much of the functionality on board a satellite has a high degree of interdependence and truly intelligent decision making should account for all of these aspects. Intelligent agents offer a mechanism to integrate these various components. Agent-based systems are goal-oriented systems in which individual agents are assigned specific tasks and in which larger problems are

solved by having a suite of agents operating in cooperation with each other. Agents collaborate through what are known as *blackboards* or *message centers* [2][3].

Consider how an agent-based system might function for a surveillance satellite mission. With regards to the payload itself, there might be agents which control the mirror positioning, control the taking of images, and agents which perform a pattern recognition function. Mirror control is a function of the output of the pattern recognition agents. Based on the results of these agents it may be desirable to reorient or maneuver the spacecraft. If a maneuver is desired several additional agents come into play. For a maneuver to occur several tasks need to happen, which includes determination of the desired end orbital elements of the spacecraft, maintenance of attitude during maneuver, appropriate heating and temperature maintenance of the catalyst bed thruster heaters, and thruster firing. Within each of these areas any number of subtasks need to happen. These tasks can be represented as agents and collaboration between agents can take place through agent message centers. A system could be designed so that all agent communication takes place through a single agent message center, however it is easy to see that for reasons involving complexity and speed this is less than desirable for large systems. A better approach would be to have a hierarchy of agents where communication between agents can be kept local when necessary while still allowing for communication between any two agents when appropriate. For instance, the picture below depicts a hierarchical system where communication between agents in the ACS, propulsion, thermal, or Command and Data handling subsystems are through localized agent message centers. When any of these agents needs to collaborate amongst each other this is done through a top level message center. The figure below depicts a setup which may exist on-board one satellite. This naturally extends to a constellation of satellites. Similarly equipped satellites might communicate with each other through an additional higher level agent message center. This is also how ground-based agents would communicate with on-board agents. This idea of a hierarchy of agents has some analogy with object oriented systems. Agents can also have the ability to inherit skills from parent agents in their hierarchy.

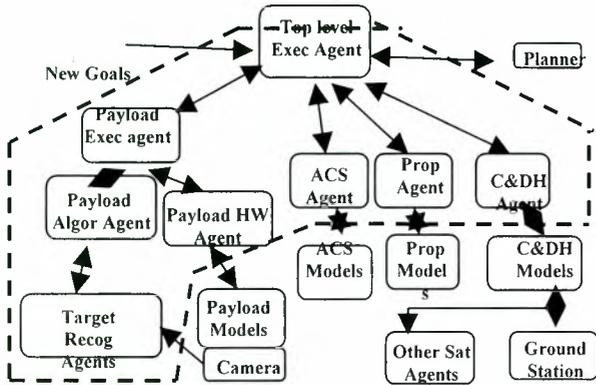


Figure 2: Agent Hierarchy

We are developing an on-board agent based architecture for intelligent satellite processing and control. The prototype agent architecture is being developed in MATLAB under Windows NT, with appropriate subsystem models also developed in that language. Subsequent development will port this architecture to C++ and a real-time flight processor. To assist in the reasoning process the architecture is equipped with a number of Artificial Intelligence (AI) modules which include neural networks, an expert system, and a model-based mechanism to perform satellite fault detection, isolation, and resolution (FDIR). Figure 3 depicts our agent architecture. The basic element in our architecture is the *skill*. A *skill* is basically any task or set of tasks which would need to be accomplished. Agents are made up of skills encapsulated in a natural language. These agents then communicate with each other through an agent message center by registering themselves and their skills (i.e., capabilities).

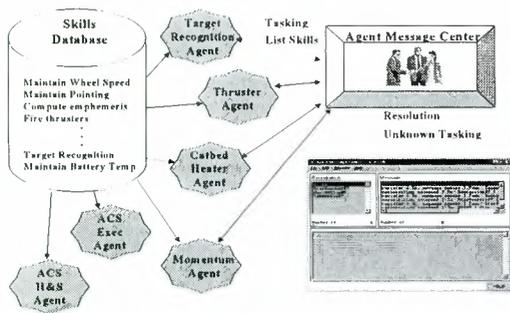


Figure 3: Agent Architecture

As a first prototype of our architecture we are applying intelligent agents to the problem of autonomous control of a space-based polarimeter. The prototype is being developed and will be simulated in an in-house testbed

which is currently under development. In our prototype environment all polarimetric processing, including stokes vector and degree of polarization calculations, is done within Matlab. The majority of the processing will be treated as a set of skills for use by the agents. Related to the payload, agents are being developed which will request an image to be taken, calculate the stokes vector, calculate and retrieve the degree of polarization, determine the polarization signature of an image on a pixel-by-pixel basis, generate a cueing signal, and control the pointing of a gimbaled mirror. The cueing signal is basically target/background information and contains information regarding the pixels in the image which contain target information. Mirror control tasking by the agents is a function of the cueing signal.

External to the payload agents we are developing agents in other subsystems for which the polarimeter has dependence upon. For instance it may be desirable to reorient the spacecraft due to a request to take a picture of an area which is currently not viewable within the current constraints of the gimbaled mirror. Agents are being developed which maintain, control and reorient spacecraft attitude along with all spacecraft activities associated with performing those functions.

Our initial prototype will make only limited use of the AI tools available within the architecture. identification. For target detection The initial focus is on target detection as opposed to the much more difficult problem of target we will make limited use of the expert system. Subsequent development will utilize more fully our AI tools.

3. CONCLUSION

Our research to date in the use of intelligent agents for on-board processing and control is still very preliminary, however the technology appears very promising. Much more work still needs to be done in order to access the true viability of the use of this technology in order to enhance spacecraft autonomy. Our architecture offers the potential for greater autonomy, with its integrated AI tools, modular and extensible design, and natural language capability. In addition we are developing a backend to our Matlab environment which will map our architecture to C++ in a real-time flight environment. We are also currently developing an autonomous planner which will take high level goals and perform system reconfiguration in response to changing mission requirements or contingencies.

4. REFERENCES

1. H. Chen and L. B. Wolff, "Polarization phase-based method for material classification in computer vision," *Int'l. Journal of Computer Vision* 28(1), 73-83 (1998).
2. Dan R. Ballard, "Intelligent Agents and Agent Communication Languages for Mission Operations", Final Report for NASA contract NAS5-33264, June 1996.
3. Barney Pell, Erann Gat, Ron Keesing, Nicola Muscettola, Ben Smith, "Robust Periodic Planning and Execution for Autonomous Spacecraft", *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.

DEALING WITH UNCERTAINTY WHEN MANAGING AN EARTH OBSERVATION SATELLITE

Eric Bensana, Gérard Verfaillie*, Claire Michelon-Edery, Nicolas Bataille**

*Office National d'Etudes et de Recherches Aérospatiales (ONERA - CERT)
2, avenue Edouard Belin, BP 4025, 31055 Toulouse Cedex 4, France
E-mail: {Eric.Bensana ,Gerard.Verfaillie}@cert.fr

**Centre National d'Etudes Spatiales (CNES)
18, avenue Edouard Belin, 31401 Toulouse Cedex 4, France
E-mail: {Claire.Michelon-Edery,Nicolas.Bataille@cnes.fr

Abstract

The possible presence of clouds is the main origin of uncertainty when managing earth optical observation satellites. Forgetting it can lead to poor results in terms of really achieved photographs. In this paper, we show how a mathematical approach, drawn from the *Markov Decision Process* framework, allows us to define a rational way of taking in account this uncertainty in the daily optimization process.

Keywords : planning, uncertainty, markov decision process

1. Context

At the highest level, managing an earth observation satellite, like *Spot*, consists in choosing the sequence of photographs to be taken. Typically, this choice is made each day for the next day. The set *SA* of the photographs that can be taken the next day (according to the satellite trajectory and the instrument maneuvering ability) is extracted from the current order book. From this set *SA*, one tries to extract a subset *SE* that is feasible (there is no conflict between photographs in *SE*; all the physical satellite constraints are satisfied) and optimal (a gain, usually equal to the sum of the gains associated with each selected photograph, is maximum).

Due to the nature of the problem (a multi-knapsack problem with a large number of capacity constraints, each of them involving only a small number of 0/1

variables^{*}) and to the size of the instances to solve (until several hundreds of 0/1 variables), efficient algorithms¹⁻² are needed. Whereas optimal algorithms³, using a *Branch and Bound* schema, can solve small and medium size instances, only sub-optimal algorithms, using a *Iterative Local Search* schema, can deal with large size instances.

Unfortunately, this daily optimization approach does not take into account the fact that most of the photographs have several other feasibility opportunities after the next day and that the number of remaining feasibility opportunities is highly variable, depending on the deadline associated with each photograph.

Moreover, it takes into account, neither the uncertainty about the realization the next day of the selected photographs (due, with optical instruments, to the possible presence of clouds), nor the uncertainty about the number and the nature of the photographs that will be concurrently added to the order book.

To face this problem, gains associated with each photograph are usually modified in order to favor photographs, that have the smallest number of remaining feasibility opportunities and the highest realization likelihood (good meteorological forecast for the next day). But the way of combining the three criteria (gain, number of remaining feasibility opportunities, meteorological forecast) is not obvious and is generally empirically achieved, without any clear idea of the consequences in terms of really achieved photographs.

* A 0/1 variable is usually associated with each photograph. The value 1 (resp. 0) means that this photograph is selected (resp. not selected).

2. A rational approach

However, a mathematical approach, drawn from the *Markov Decision Process (MDP)* framework⁴, currently used in *Decision Theory*, can help us to define a rational way of aggregating those three criteria.

To take into account the presence of several feasibility opportunities for a photograph, it is necessary to consider a global gain criterion over a given horizon rather than a daily gain criterion. A sensible choice consists in considering an horizon that covers all the feasibility opportunities of all the photographs belonging to the current order book. To take into account the presence of uncertainty, it is necessary to consider an expected global gain criterion rather than a global gain criterion.

Using this expected global gain criterion, the strict *MDP* approach leads us to intractable problems, because of the lack of knowledge, even in terms of probability, about the photographs that will be added to the order book and, above all, because of the huge number of states that should be explored by the *Dynamic Programming* algorithm, currently used in the *MDP* framework to compute optimal policies.

Fortunately, thanks to some simplifying assumptions (essentially, no influence of the current decision upon the future expected gains associated with the photographs that either belong to the current order book, or will be added to it), one can establish that the optimal policy (the one that maximizes the expected global gain) consists in selecting each day a set *SE* of photographs that is feasible and maximizes the sum of the weights of the photographs in *SE*, with weights set according the following formula:

$$w(p, d_c, \pi^*) = g(p) \times p_r(p, d_c) \times P_{ef}(p, d_c, \pi^*) \quad (1)$$

$P_{ef}(p, d_c, \pi^*)$ being computed by the rule :

$$\begin{aligned} &\text{if } RFO(p, d_c) = \emptyset && (2) \\ &\text{then } P_{ef}(p, d_c, \pi^*) = 1 \\ &\text{else } P_{ef}(p, d_c, \pi^*) = \\ &\quad \prod_{d \in RFO(p, d_c)} [1 - p_r(p, d) \times p_s(p, d, \pi^*)] \end{aligned}$$

where:

- p is a photograph;
- d_c is the current day;

- π^* is the optimal policy;
- $w(p, d, \pi)$ is the weight to be associated to the photograph p , the day d , according to the policy π ;
- $g(p)$ is the gain associated with the actual realization of the photograph p ;
- $p_r(p, d)$ is the realization probability of the photograph p the day d ;
- $P_{ef}(p, d, \pi)$ is the non-realization probability for the photograph p on the days after d , using the policy π ;
- $RFO(p, d)$ is the set of feasibility opportunities of the photograph p , remaining after the day d ;
- $p_s(p, d, \pi)$ is the selection probability of the photograph p the day d , using the policy π .

The realization probability of a photograph p the day d can be easily obtained, either from short term meteorological forecasts, or from long term climate statistics.

As for the selection probability of a photograph p the day d , using the policy π , if one assumes that the order book keeps globally stable, at least over a large period, one can consider that it is a function f of p 's weight, localization and type, *i.e.*

$$p_s(p, d, \pi) = f[w(p, d, \pi), l(p), t(p)]$$

where $l(p)$ is p 's localization and $t(p)$ is p 's type. Indeed:

- the higher p 's weight is, the higher p 's selection probability is;
- the higher the demand in p 's area is, the higher the likelihood of conflict with other photographs is and the lower p 's selection probability is;
- the more resource consuming p is (example: stereo demands), the lower p 's selection probability is.

But, how to fix function f ? It seems that a sensible option consists in learning it, in fact in approximating it, either off-line from simulations, or on-line from the observation of the system behavior⁵⁻⁶. For example, a

multi-layer neural network could be used for that. Whatever the technical option is, note that on-line learning has the advantage to allow the system to adapt itself to mid and long-term changes in the size or the nature of the order book.

As soon as the selection and realization probabilities (p_s and p_r) have been fixed, the recurrent equations 1 and 2 can be used to compute the weight to be associated with a possible photograph: the process starts with the last opportunity ($RFO(p,d) = \emptyset$) and ends with the current one, alternating computations of weights and of selection probabilities.

Note that, as it was foreseeable, equations 1 and 2 favor photographs that:

- have a high associated gain;
- are subject to a good meteorological forecast for the next day;
- have a small number of remaining feasibility opportunities;
- are subject to bad meteorological forecasts for the days associated with these remaining feasibility opportunities;
- are localized in areas in great demand;
- are very resource consuming.

3. Conclusion

As a conclusion, thanks to some simplifications, an *MDP*-like approach provides us with a rational way of dealing with uncertainty when managing an earth observation satellite. The next step of this work would consist in fixing the learning process of the selection probability and in carrying out simulations in order to measure the actual gain in terms of achieved photographs.

References

- [1] J.C. Agnès, N. Bataille, E. Bensana, D. Blumstein and G. Verfaillie. *Exact and Approximate Methods for the daily management of an Earth Observation Satellite*. In Proc of the 5th ESA Workshop on Artificial Intelligence and Knowledge Based Systemes for Space, Noordwijk, The Netherlands, 1995. <ftp://ftp.cert.fr/pub/verfaillie/estec95.ps>
- [2] E. Bensana, G. Verfaillie, J.C. Agnès, N. Bataille, D. Blumstein. *Exact and Approximate Methods for the Daily Management of an Earth Observation Satellite*. SpaceOps 96, Munich, Germany. <ftp://ftp.cert.fr/pub/verfaillie/spaceops96.ps>
- [3] G. Verfaillie, M. Lemaître, T. Schiex. *Russian Doll Search for Solving Constraint Optimization Problems*. AAAI 96. <ftp://ftp.cert.fr/pub/verfaillie/rds-aaai96.ps>
- [4] M. Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
- [5] D. Berstekas, J. Tsitsiklis. *Neuro Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.
- [6] R. Sutton and A. Barto. *Reinforcement learning*. MIT Press, Cambridge, Massachusetts, 1998.

Robotic Servicing Demonstration Missions



DEMONSTRATION MISSION OF A SATELLITE SERVICING SYSTEM

Mitsushige Oda and Noriyasu Inaba

Engineering Test Satellite VII Project Team
National Space Development Agency of Japan
2-1-1, Sengen, Tsukuba, Ibaraki, 305-8505 Japan
phone:+81-298-52-2855, fax:+81-298-60-6718
e-mail: Oda.Mitsushige@nasda.go.jp

Abstract

On-orbit service such as refueling, repairing and re-orbiting is highly required for space activities not only from economical view point but also from environmental view point, since recycle type space activity saves launch cost in a long term and bears fewer space debris. The Japanese Engineering Test Satellite VII (ETS-VII) was launched in 1997 to conduct on-orbit servicing experiments. Based on the results of the experiments using ETS-VII, the over all scenario to provide on-orbit services to spacecraft on orbit, including non-cooperative target like a failed satellite is presented. Technologies to be studied and actions to be taken are also reviewed not only from technical view points but also from political view points to enhance contents and quality of on-orbit servicing provided in accordance with the scenario above.

Key words: On-orbit servicing, ETS-VII, Refueling, Space Debris

1. Background

Number of satellite and spacecraft which will be launched into space is increasing year by year. Most of them will conduct their mission and some of them are not because of anomalies of the system. As missions go on, some spacecraft requires logistic support during their mission lives. Every satellite will end their mission after or before conducting their mission. Most of these satellite and spacecraft will require at least one of the following services.

- (1) Logistic support such as fuel supply and/or equipment exchange
- (2) Recover of mission either by transporting the spacecraft from a current orbit to the desired one, acquiring satellite attitude stability and/or fixing a failed part of the spacecraft
- (3) Removing spacecraft from the occupied orbital position or orbital altitude after the mission

In a case of launcher's miss insertion to an orbit, it is also welcomed to recover a mission by re-boot the satellite from the unexpected orbit to the goal orbit. This potential needs for the re-boost is quite large especially in a case that the satellite is very expensive. Although a low cost commercial satellite is not cost worth to conduct recover works, it is quite important to remove them from the occupied orbit because of the following reason. Development of constellation satellite for pan-earth communication will bring many satellite in orbit. Those satellite will finish their mission in some years. If those unused satellite stayed in orbit, risk of colliding them will become higher and higher. Therefore removal of those satellites become important.

NASDA developed and launched Engineering Test Satellite VII (ETS-VII) to test and demonstrate primarily on-orbit servicing technologies which are essential to provide services mentioned above⁽¹⁾.

2. Satellite servicing system

Rendezvous and docking technique used in ETS-VII requires a user satellite to have special sensors (GPS receiver and large markers for rendezvous sensors) and equipments. (docking mechanism, etc) In addition to this, components can be replasable only by an unit of ORU which is conveniently designed for manipulator handling.

These request for the user is a strong "user penalty" in terms of cost, size and weight. Moreover, this technique can not be used for a target which lost it's controll like a failed satellite. This penalty still seems to be a strong barrier to promote the concept of on-orbit servicing. Therefore, a technique to approach, capture a non-cooperative target is highly required to promote the concept of on-orbit servicing. A strategy to provide an on-orbit services to non-cooperative target is presented in this chapter.

2.1 Overall strategy for Satellite Servicing

An overall strategy to approach and capture a non-cooperative target in a future mission is illustrated in Fig. 1. The detail of each step in the strategy is discussed in the following sections.

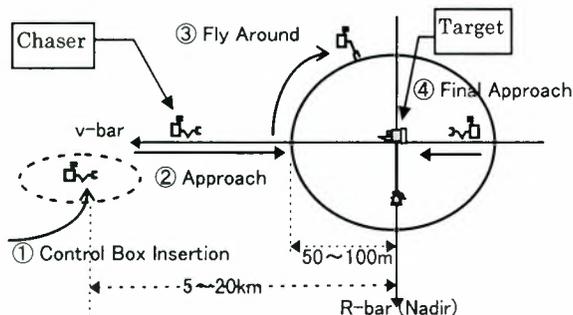


Fig.1 A Strategy to approach and Capture a Target

2.2 Ground Based Observation

A servicing vehicle (chaser) is launched and inserted into an orbit guided by on ground radar observations. Fig. 2 shows a sample of ground based observation of ETS-VII orbiting 550km in height. The image was taken by a German ground radar and provided to NASDA under a collaboration between German Aerospace Center (DLR) and NASDA. The attitude motion of a satellite as well as orbital elements can be estimated from a series of ground based observations. Since ground based orbit determination includes an error, an insertion point of the chaser is generally set 5 to 20 kilometer behind

or ahead of a target satellite in the Euler-Hill's frame to ensure that the target is inside of the navigation sensor's field of view at the time of the control box insertion.

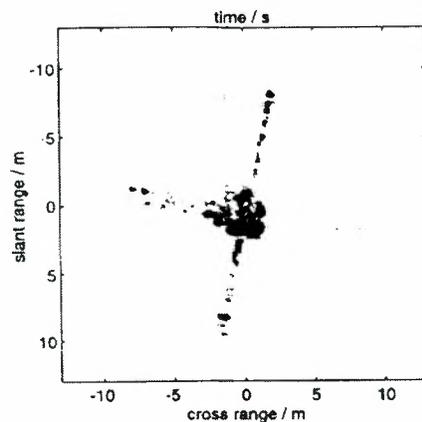


Fig.2 A Ground Radar Image of ETS-VII

2.3 Rendezvous to a Target

The chaser approaches to the target based on the navigation sensor's information. Well known CW guidance technique⁽²⁾ is a convenient way to guide a chaser to the target with a small consumption of fuel. A radio radar is a conservative choice as a navigation sensor in this approaching phase, since it is usable all around the orbit including a solar eclipse period. But it raise the cost of the system. Therefore, it is preferable that necessary measurements are conducted from general CCD camera images of a target satellite to reduce the total cost of the system.

A fly around maneuver is conducted to observe the condition of the target and also to get information necessary for a final approach. Final approach is made to capture the target. The approaching direction and maneuvering mode are depend on the motion pattern of the target.

2.4 Capturing

After the final approaching phase, the chaser tries to keep the constant distance and attitude relative to the target unless the motion of the target is too fast. By doing so, the capture by a manipulator will be easy. It is not so unrealistic to assume that the motion of the target is not so rapid. Because an rigid body object on a low earth orbit will orient it's minimum axis of inertia toward nadir by the influence of the gravity gradient torque in accordance with energy dissipation regardless of the initial condition.

An action to dump the excessive momentum will be

conducted using a flexible mechanism or other equipment attached to the tip of a manipulator in a case that the target has too large momentum to be captured immediately. An Extra Vehicular Activity to dump a excessive momentum of NASA’s SMM satellite gives us a good hint to design a same kind of rescue mission conducted by a robot satellite.⁽³⁾ Appropriate part of a target satellite should be chosen as a capturing point which can endure the stress in a capturing process and ensure the collision free path planning of the endeffector of the manipulator. Under the assumption that the chaser is keeping constant distance and orientation by station keeping maneuver, the collision free path plan of the manipulator is relatively easy. Because the environment around the manipulator is static. An off line path planning can be done based on the environment model which is stored on ground data base and updated by in-situ observations. Visual servoing technique is also important to compensate model error and to guide the endeffector finally to the grasping point. An strategy of final approach and capturing is illustrated in Fig.3.

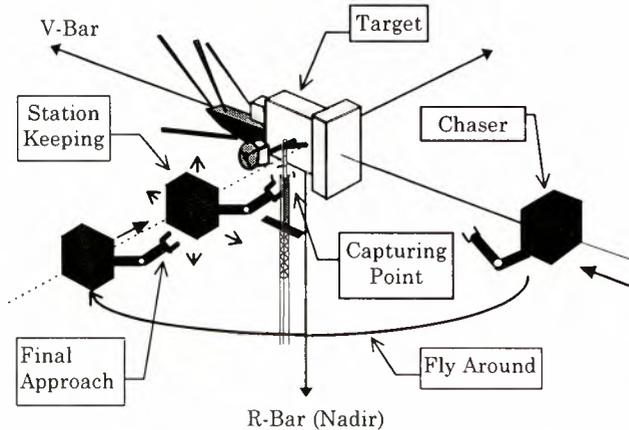


Fig. 3 A Strategy of Final Approach and Capture

An important point is that a small “user penalty” like a handle with vision marker installed on the ETS-VII target (Fig. 4) will make this grasping process easy and sure to a large extent.

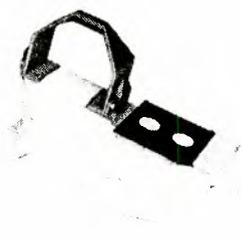


Fig. X ETS-VII’s Capturing Handle

2.5 Servicing

Following on-orbit services can be supplied to the target:

- Inspection
- Changing ORUs and components
- Refueling
- Re-orbiting or De-orbiting

Since current ORU design requirements give not a small impact to satellite system’s design in terms of mechanical size and clearances between components, an effort to mitigate these system impact is necessary by increasing a dexterity of a manipulator and setting an appropriate robotics work-site interface NASA’s standard of extra vehicular activity (EVA) interface⁽⁴⁾ is a good example to think about a world wide robotics interface standard for on-orbit servicing.

3. precursor experiment on ETS-VII

On orbit servicing experiments using NASDA’s ETS-VII are reviewed to confirm demonstrated technology and also to clarify the direction to be followed in this area.

3.1 Rendezvous to cooperative target

Unmanned and autonomous rendezvous docking technology was developed and verified in ETS-VII’s RVD experiments. The chaser approached and finally docked to the target from maximum 10km separated point. Using three kinds of navigation sensors, GPS receivers, laser radar and proximity sensor. A typical flight path plan of a RVD experiment is shown in Fig. 5.

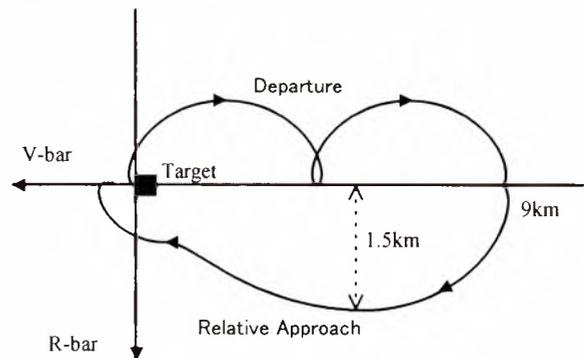


Fig 5 Rendezvous Flight Profile

Visual inspections of the target were also conducted during RVD experiments. Fig.6 shows a image of the target separated from the chaser by 13[m] .

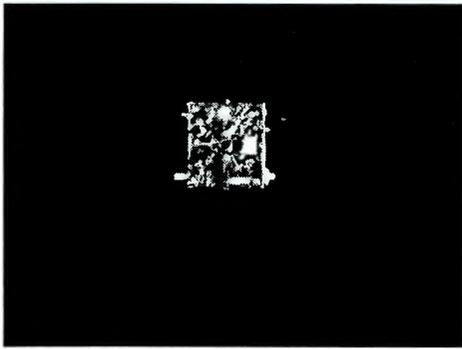


Fig. 6 An Image of Target Satellite

As a advanced rendezvous experiments, a fly around inspection experiment is planned. The plan is to release the target satellite from the chaser satellite and then conduct “fly-around and visual inspection experiment” of the target satellite.

3.2 Capture and Berthing with manipulator

ETS-VII has a capability to feed back the on-board hand eye camera’s information at the rate of 2 [Hz] to a path planning controller of a robot arm. A hand eye camera image taking a special vision marker is once converted into a black and white image under a threshold which is commanded from the ground. By extracting the marker’s sizes and centroids from the black and white image, the relative position and orientation between the camera and the marker are measured at the rate of 2[Hz] with a “H32” 32bit space qualified processor. Once the marker’s position in a image is found, the search area on a CCD is narrowed to avoid miss-detection of the marker from the next search. The relative velocity between the camera and marker is taken into account to decide the search area on the CCD.

Based on the measurement, an arm planning controller generates a desired arm position and orientation command automatically. Digital filtering techniques are used to abandon a faulty data while keeping the rapid tracking capability, since on-board camera images are much influenced not only by an aspect of direct lights from the sun and the earth but also by unpredictable reflections. The major functions of the digital filter is as follows:

- Low-pass filter
low-pass filter with “ if then loop ” to avoid an excessive motion of the arm and also to abandon faulty data
- feed forward
Compensation of the time delay of the vision sensor

The block diagram of the ETS-VII visual servoing is shown in Fig. 7.

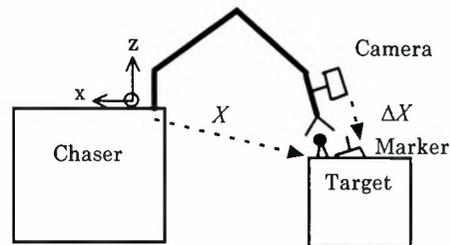
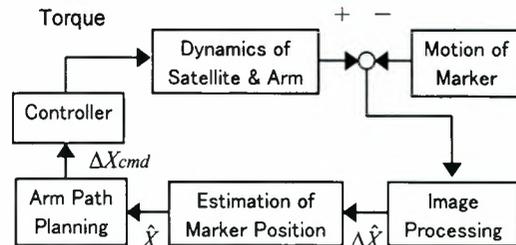


Fig. 7 Block Diagram of the Visual Servoing

There are basically three control modes to keep the satellite attitude while the arm is moving. First one is basic feed back control independent from the arm control. The second one is feed-forward control in which predicted disturbance angular momentum due to the arm motion is fed forward to the satellite attitude controller. The third one is free motion mode in which no attitude control is conducted. Appropriate attitude control mode is chosen depending on the task performed.

The basic function of the ETS-VII visual servoing was confirmed on January 1999 under a real on-orbit lighting condition. The sequence of event of an experiment was carefully designed so that the visual marker used for the image processing is stably illuminated by the Earth albedo. In the experiment, a robot arm on the chaser satellite approached and grasp the handle on the target satellite automatically under the condition that the target satellite is fixed to the chaser by a docking mechanism. The approach initiated from a position approximately 0.7 [m] above the capturing position with 0.3 [m] position error in off-axis direction. The robot has a function to start a capturing sequence by closing the fingers of the endeffector after confirming that the arm reached the aiming point relative to the target marker. The mode transition of this automatic capturing function is shown in Fig. 8.

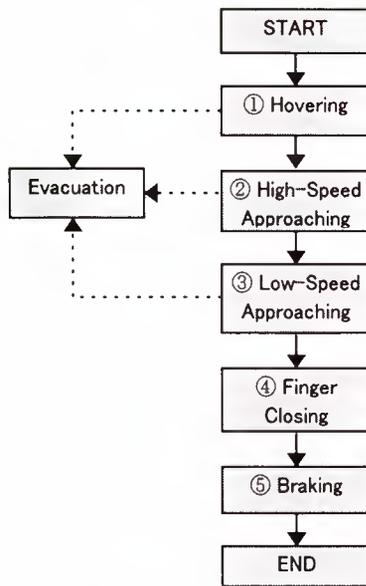


Fig. 8 Mode transition of Automatic Capturing

Images of hand-eye camera during the experiment are shown in Fig. 9.

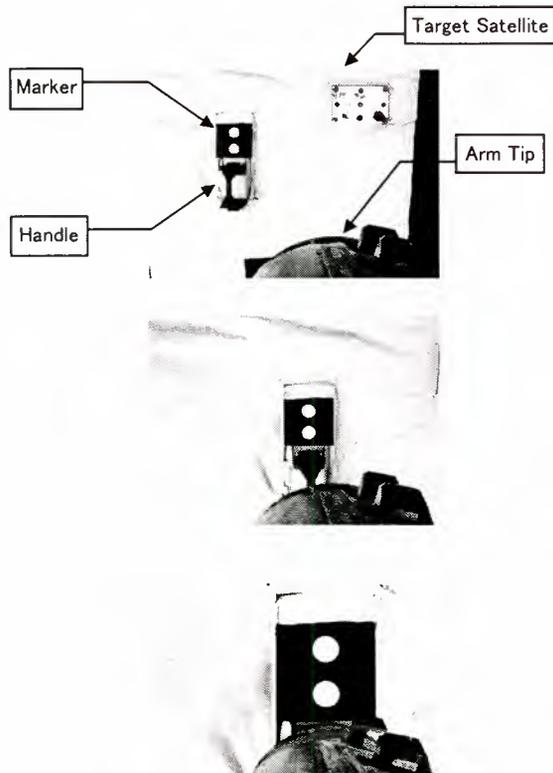


Fig. 9 Hand-eye Images during Visual Servoing

Based on the data obtained from the experiment and other preparatory experiments conducted, an automatic satellite capturing experiment using robot arm is planned as one of the extra experiment in the extended mission life of ETS-VII satellite. The target satellite of ETS-VII which is separated and

floating on an orbit will be captured by the robot arm on the chaser satellite using this visual servoing technique. An image of satellite capturing experiment is shown in Fig.10.

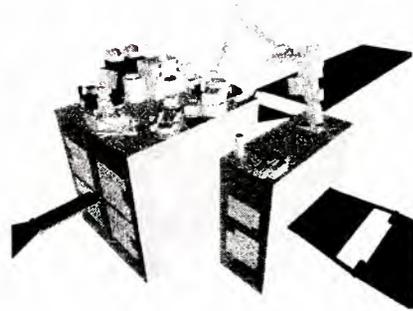


Fig. 10 Automatic Satellite Capturing

3.3 ORU Change and Refueling

An experimental On-orbit Replacement Unit (ORU) was separated from an ORU port attached to the satellite surface and connected to the port again by the manipulator. Fig. 11 shows the on-board camera image taken during the experiment.

Pairs of an electrical connector and a liquid quick disconnecter are installed between the ORU and the ORU port. Disconnection and re-connection of these connectors were surely confirmed during the experiment.

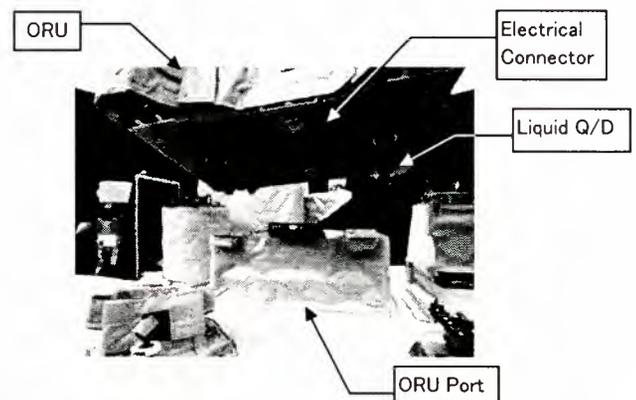


Fig. 11 Image of the ORU replacement

An fuel transfer experiment was also conducted. Because of its physical characteristics similar to hydrazine, water was used for this fuel transfer experiment. Approximately 0.7 [kg] of water inside of a supplying tank was transferred to the receiving tank. The receive tank is inside of the ORU while supplying tank is inside of the ORU port. These two tanks were connected by the quick-disconnector. The procedure of the fuel transfer experiment is illustrated in Fig. 12.

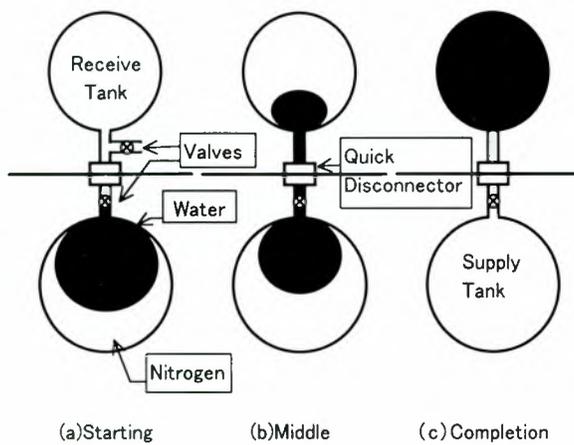


Fig. 12 Procedure of the Refueling Experiment

4. Future Scope

Although there are several merit of the on-orbit servicing system in terms of space environment and also the cost in a long term taking an inexpensive future reusable launch system into account, it is fact that there is no urgent and concrete requirement from users to use it. This is mainly because the “user penalty” is still too large to receive the on-orbit services. Therefore, following actions are required to promote the on-orbit servicing system.

4.1 Technologies to be Studied

● Reducing User penalties

It is highly required to study a rendezvous technique to approach and capture a non-cooperative target which does not have special sensors nor equipment for rendezvous mission. Unnecessity of those sensors and equipment greatly reduces the user penalty.

A radio radar or laser radar is a good candidate for a rendezvous sensor in a long range while image sensor which can measure relative distance and orientation from a CCD image of a target is a candidate for the sensor in proximity operations.

A rendezvous technique in a non-circular orbit is also required to serve a satellite in non-circular orbit such as a faulty inserted satellite.

● Robotics work site technology

It is important to choose an appropriate robotics

work site interface to keep the total cost of the on-orbit servicing system minimum by avoiding a servicing robot to be too dexterous while reducing the user penalty to an acceptable level.

4.2 Political Strategy

● Demonstration Mission

A demonstration to users and taxpayers is important to show the feasibility of more convenient on-orbit servicing capability and to persuade them to choose the on-orbit servicing concept. The target satellite for the demonstration mission will be an actual failed satellite on-orbit or an experimental satellite specially designed for the mission. It is needless to say that the former is more sensational than the latter if possible.

● Future Vision

It is also important to lead the users by showing the long term vision and merit of choosing on-orbit servicing that sweep away a choice based on a short term benefit.

Unfortunately, the fear of space debris problem does not seems to be an urgent factor that can persuade general users to choose on-orbit service immediately. Because some studies show that the continuation of the current launch rate does not lead to a phenomena of cascading of debris in near future.

References

- (1)Oda, M. “ETS-VII a Rendezvous Docking and Space Robot Technology Experiment Satellite.” Proc. of the 46th International Astronautical Congress, Oslo, Norway, 1995, IAF-95-U.2.01.
- (2)Clohessy, H., and Wiltshire, S. “A Terminal Guidance System for Satellite Rendezvous”, Aero/Space Science, Vol.,27, 1960. P563.
- (3)Kevin, G. “The dynamics of solar maximum mission spacecraft capture and redeployment on STS41-C.” AAS-85-060.
- (4)STS 07700, vol. XIV appendix.7,NASA.
- (5)Inaba, N., and Wakabayashi, Y. ” Automatic Satellite Capture and Berthing with Robot Arm”, I-SAIRAS’ 94, pp205-208, 1994.

Vision & Interactive Autonomy Bi-Lateral Experiments on the Japanese Satellite ETS-VII

Daniele G. Galardini, Konstantinos Kapellos and Eric Maesen
 TRASYS Space - Leuvensesteenweg 510, Bus 43, 1930 Zaventem, Belgium
 Gianfranco Visentin and Frédéric Didot
 ESTEC – Keplerlaan 1, 2201 Noordwijk, The Netherlands

ABSTRACT

In April 1998 the Vision & Interactive Autonomy Bi-Lateral Experiments (VIABLE) project on the ETS-VII engineering satellite of NASDA, the Japanese Space Agency, entered in its final phase. This first collaboration between ESA and NASDA has the scope to test the Interactive Autonomy concept for space robotics and to investigate advanced vision-based techniques for robot positioning and calibration. Thanks to the ETS-VII environment, several experiments have been performed in April 1999 using the ETS-VII robot arm and the on-board available utilities. Fine positioning, compliance control, camera calibration, robot calibration, force/torque sensor calibration are at present the experiences foreseen.

The project development is carried out by TRASYS Space as prime contractor. The ground station for robot monitoring and control has integrated ESA developments as well as university S/W toolbox, and experiences have been extensively studied and validated with a robot simulator prior to the operations phase. During the operations phase, the robot tasks are executed in the satellite environment and the actual robot movements are shown concurrently to the simulation. Besides the interest of the experiences themselves, VIABLE would show the possibility to adapt existing software for high-level robot task control to a robot using low-level control statements used to command the ETS-VII on-board robot.

Finally, the re-use/adaptation of existing products and the collaboration between different Space Agencies has provided a series of technical and management returns (lessons

learned), from which will benefit future robotics space missions.

The paper is organized as follows. Section one will describe the on-board/on-ground environment and the constraints to which the developers have to adapt the control system. Solutions to implement the VIABLE on-ground control station will then be given in Section 2. A description of the experiments and of the main attempts will be presented in Section 3. Considerations on managing international projects and final remarks will conclude the paper.

Section 1: System Description

Figure 1.1 shows the overall ETS-VII robot environment. Following paragraphs describe briefly the available hardware to support both the Interactive Autonomy experiments and the Vision-Based Control experiments.

1.1 ETS-VII Robot Arm

The ETS-VII robot arm (ERA), mounted on the satellite platform, is a six degrees of freedom manipulator of approximately 2 m in a fully stretched configuration. A force/torque sensor is mounted on the last joint. The performance characteristics of the robot arm are shown in Table 1.1.

	POSITION	ATTITUDE
Pose accuracy	10 mm	1 deg
Path accuracy	30 mm	3 deg
Pose repeatab.	2.5 mm	0.13 deg
Max. tip speed	50 mm/sec	5 deg/sec
Speed acc.	10 %	10 %

Table 1.1: Robot arm characteristics

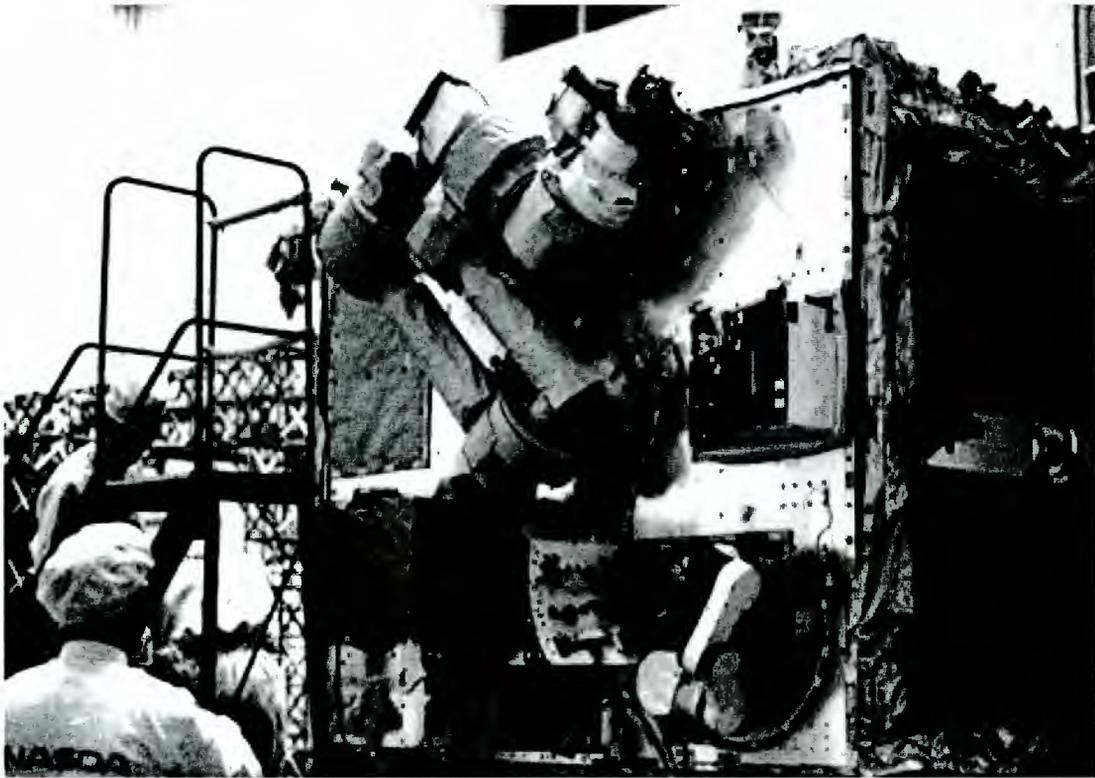


Figure 1.1: ETS-VII Robot arm and its environment

1.2 Arm End-Effector

The end-effector is attached to the *f/t* sensor. It can be equipped with different grapple fixtures (GPF) to handle payloads (P/L). As a result, the end-effector provides the following functions:

- grasps a GPF attached to the P/L;
- provides torque to P/L via GPF;
- provides electrical interface with P/L via GPF.

For VIABLE experiments it is allowed to use the standard GPF only.

1.3 Vision System

The on-board vision system consists of two sets of cameras. The arm hand camera (AHC) set is mounted on the end-effector and the arm monitor camera (AMC) set is mounted on the first joint of the robot arm. Each set contains two B/W CCD cameras, a primary one and a redundant one. All cameras have a 668 by 485 pixel resolution and a fixed focal length. Video data of any two cameras are sent to ground each 250 ms. JPEG video data compression is used to reduce the video data size to 1.2 Mbps.

A video data processing (VDP) unit is present on-board for measuring misalignments by using a 3-point alignment marker and computing the relative position and attitude of three white circle images of the marker in the video image. This computation takes less than 0.5 sec. For experiment execution, video data transmission to the ground control station can

occur in two modes: either 4 Hz AHC + 1 Hz AMC or 2 Hz AHC/AMC + 2 Hz AHC/AMC.

1.4 On-Board Controller

The robot mission on-board controller (RMOC) together with the arm drive electronics (ADE) controls ERA. While the latter performs joint servoing at a rate of 5 ms and sensor signal processing, the former is responsible for arm trajectory control and compliance control (maximum rate 250 ms).

Note that the robot arm motion plan, i.e. path and speed, is modified in order not to disturb the satellite attitude motion.

1.5 Taskboard

The taskboard consists of functional items supporting the evaluation of the robot arm performance and arm motion characteristics. Experiments with the different items can only be performed with the Taskboard Tool Handling (TBTL) device, which is to be attached to the external robot arm (ERA) end-effector.

The functional items are:

- force/torque sensor calibration unit (spring coil with scale) whose the displacement can be measured with the arm hand camera as it is located close to a linear scale;
- a peg, permanently fixed to the TBTL, diameter 18 mm, with two different holes (diameters 18.4 and 19 mm);

- a slider handle to be operated with the peg for testing contact motion;
- a linear scale (1mm resolution);
- a sine curve surface to test contact motion;
- and a small chained floating ball.

1.6 Arm Control Modes

The arm can be controlled in three different modes:

1. joint position control mode (absolute or relative);
2. Cartesian position control mode (straight line);
3. Cartesian compliance control mode.

Ground station commanding is possible:

- either by uplinking continuous (every 250 ms) incremental setpoints (position & attitude or joint angle commands), called point of resolution (POR),
- or by uplinking robot telecommands (straight path motion, single joint movements).

1.7 Modes of Operations

The ETS-VII robot supports three modes of operation:

1. pre-programmed execution mode: the robot motion plan consists of robot telecommands whose validity (correctness, collisions and interference with satellite attitude control) is verified prior to execution;
2. telemanipulation: incremental positions (POR) are sent each 250 ms and interpolation is performed by the on-board robot controller;
3. real-time execution mode: used for the floating ball capturing experiment assisted by the vision data processing unit.

The on-board robot arm will be controlled from a ground control station at NASDA Tsukuba Space Center, Japan. As said the ground operator will receive real time images of the robot arm motion that he can use together with the predictive CAD simulator for supervision.

The time delay in the communication link is around 4 s.

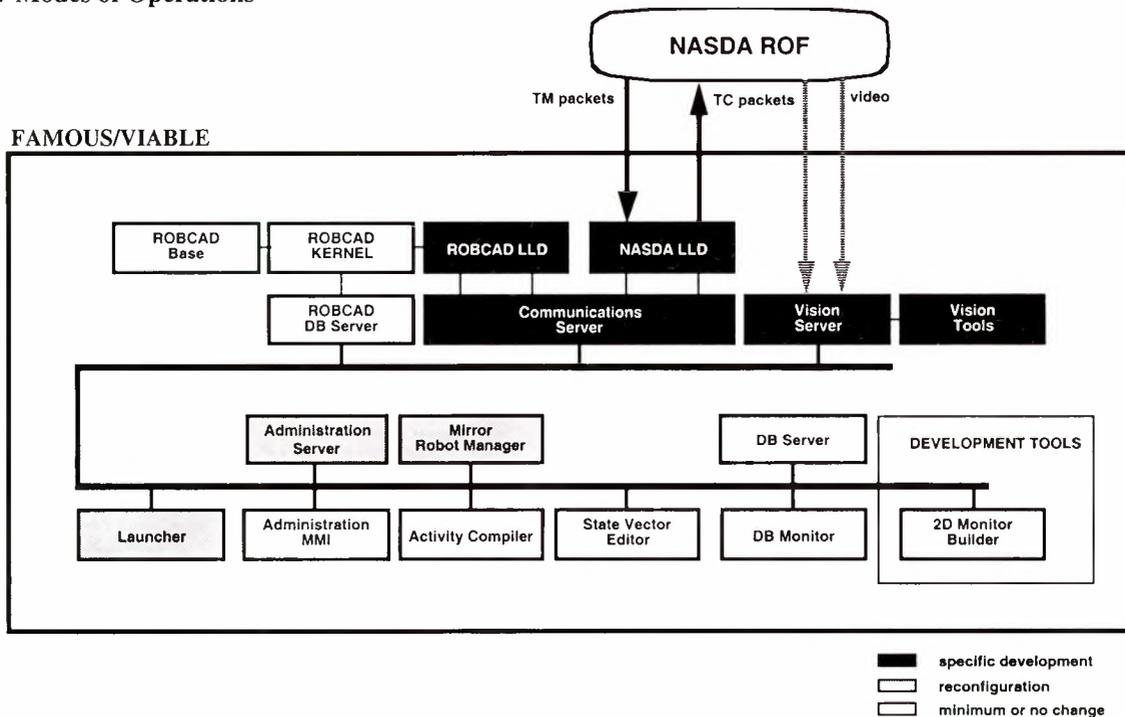


Figure 2.1 – VIABLE station general description

Section 2: ETS-VII Ground Testbed

The VIABLE station is integrating the FAMOUS/Generic system, a previous ESA development which has led to a general purpose and project independent software designed to actively support an on-ground operator for the preparation and execution of space-based robotic experiments. The framework has to be *instantiated* (by

addition of dedicated software or by configuration of existing software) to support the VIABLE experiments. This is mostly due to different and project specific robot languages and communication protocols. Figure 2.1 shows the resulting VIABLE station architecture and the interface with the NASDA Robot Operator Facility (NASDA ROF), i.e. the NASDA mission control center that has the final authority in uploading commands. Grayed boxes identify the specific VIABLE

software components. The VIABLE station is organised as a series of servers and clients components:

- servers provide standard services to the calling components (e.g. access to the workcell model, or communications with the robot via the NASDA ROF);
- the clients are basically MMI with (very) limited processing capabilities, from which the user operates or monitors events.

The components are mostly based on Tcl/Tk, and are integrated with each other using Tcl-DP. The Communications Server is the central monitoring and control component for VIABLE. The Low-Level Drivers (LLD) transform high-level requests into basic telecommands for the real NASDA robot and/or to the simulated robot. ROBCAD is the kernel software for simulation and workcell representation.

Section 3: VIABLE Experiments

3.1 IA Robot Manipulation

The Interactive Autonomy (IA) operational concept has been devised in order to enable the use of robots by non-experts. It includes two aspects:

Interactivity. A first case of interactivity is when an user has to adapt his experiment plan with new parameters. A second one of interactivity concerns the anomalies or divergences between the expected experiment results and the actual ones. The interactions are performed from the ground, making use of pre-programmed nominal or recovery sequences. In both cases, the operator does not interact at servo-level but using pre-programmed sequences that can be considered as macro commands.

The nominal task of the robot is prepared on ground using an off-line programming system, supported by CAD simulation. The safe execution (e.g. against collision) is particularly verified. Once the tasks has been validated, these are ready for non-interrupted execution.

Autonomy. Due to the functionality offered by the NASDA ETS-VII robot, the resulting autonomy is reduced to the execution of simple low-level commands. To increase the abstraction level of each task an interpreter between high-level commands (compound tasks) and low-level commands (ERL commands) will be provided. As shown in Figure 3.1, such interpreter also uses the telemetry data to control what the low-level commands need to be uploaded. Therefore, from the end-user's point of view the

compound task abstraction level will be maintained high when from the developer's point of view the system will present a low level of abstraction even in the compound task description.

Remember that compound tasks are sets of commands whose execution is carried out in a completely autonomous way. The interaction between the operator and the robot is limited to initiate these tasks and to monitor their execution.

IA operations and expected advantages

For VIABLE, the IA operations are broken down into four different phases:

- Station development: encoding of control concepts in modular form and creating templates for supporting the envisaged operations (on-ground);
- Compound tasks preparation: interactive programming of high level commands (compound tasks) and verification by simulation (on-ground);
- Mission simulation: use of the VIABLE station to simulate operations (e.g. for training);
- Mission operations: autonomous execution (on-board) of the above-prepared compound tasks with high level monitoring (on-ground) by the VIABLE station. Moreover, contingency procedures for releasing the control to NASDA and, if possible, for recovering the operations (on-ground) have been established;
- Post-mission analysis: the post-mission data analysis.

In other words, the operator by means of an off-line programming tool composes each compound task (i.e. macro command). Then the compound task is parameterised and validated by simulation during the mission simulation phase.

For the VIABLE mission, the major advantages foreseen using IA instead of telemanipulation are:

- safety of the execution that is guaranteed by validation of the tasks by the programming and simulation tool in addition to the monitoring by the human operator of telemetry data communicated to the ground segment;
- more predictable performance and better reliability;
- lower demand on operator skill and workload, enabling a scientist to concentrate only on his experiment;
- safe task execution due to testing and validation of pre-programming effort;
- faster task execution.

Notice that a prerequisite for applying IA is the availability of a robot with very predictable Cartesian and dynamic behaviour. In case of poor robot repeatability and accuracy, as on

ETS-VII, a higher level of interaction by operator is foreseen when precision positioning has to be guaranteed. Otherwise, higher autonomy could be maintained.

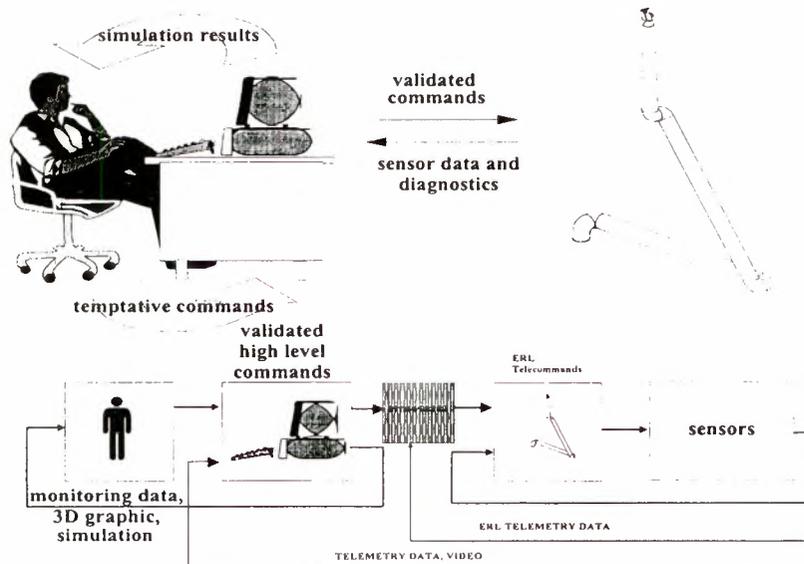


Figure 3.1: Control loops in VIABLE Interactive Autonomy.

3.2 VBRC Experiments

The operation phases for Vision-Based Robot Control (VBRC) experiments are the same as for IA experiments.

The major expected advantages of the VBRC experiments are:

- to assist the operator during the IA experiments, e.g. to refine positions, to retrieve the slider position;
- to calibrate the cameras parameters that are unknown. The completion of this item is required before the use of the Vision Tools for combined IA/VBRC experiments;
- to validate an innovative algorithm to obtain eye-hand calibration from images only;
- to provide material for post-mission robot calibration.

The following examples will help appreciate the expected additional benefits.

VBRC/IA Peg-Into-Hole Experiment

The robot moves blindly to a position above the hole. The cameras take images and these are augmented by Vision Tools. This means that the markers and the holes from the CAD model will be displayed directly on the image. Also the impact point of the peg will be drawn on the image. Then, an ellipse fitting is applied to the original images (only in a restricted area: close to the projection of the CAD hole) and from the difference between the hole in the

images and the projected impact point, an update for the robot position is computed.

The robot performs this update, images are taken and these are again augmented. If necessary, a new update is performed and the robot finally uses force feedback to insert the peg into the hole. This will help the operator in the execution of a safe experience, reducing torques and forces on the robot end-effector. An example of the features offered by Vision Tools is given in Figure 3.2 which shows a first Breadboard Prototype with marker position superimposed.

Eye-Hand Calibration

An innovative algorithm has been developed to obtain eye-hand calibration from images only without any need to touch the grapple.

The procedure computes the eye-hand calibration in two stages. First the translation between camera and robot-tip is computed, then the rotational component is searched for.

For the eye-hand translation, 3 equidistant views are taken from the markers and an Euclidean reconstruction is found. Then, the robot is rotated with a known rotation, again 3 equidistant views of the markers are taken and the reconstruction is computed. From both reconstructions and the known rotation, the translation between camera and robot can be found. For the eye-hand rotation, a starting point for the camera is selected. Then, the camera is moved in 3 orthogonal directions

and for each direction two images are taken. From these views, the rotational component of the eye-hand calibration is found using vanishing points.

This procedure is considered as an advanced experiment with high scientific significance.

Conclusions

VIABLE is the result of a collaboration between NASDA and ESA. From a Project Management point of view, the following points are worth to mention:

1. direct, and even personnel, contact between the technical people of both sides dramatically enhance mutual understanding (and therefore solving) of the difficulties;
2. the whole management chain should actively relay and support communication of up-to-dated technical information (documents, engineering models, and even piece of code) or, when this is unfortunately the case, dare to quickly and explicitly declare an information as missing or lost;
3. common definitions and conventions must be clearly stated and published to avoid misunderstandings between partners;
4. what are the allowed and not allowed operations must be clearly defined and explained by the participating agencies, for example in the form of typical and representative utilisations scenarios

We would like to stress again the importance of a project management that acts even at technical level following the exchanges between the partners and avoiding project re-engineering.

Technically, VIABLE re-uses previous ESA developments for the ground station implementation.

Finally, VIABLE has been the first project demonstrating in-orbit the Interactive Autonomy mode of. The operations (Figures 3.3-7) at NASDA Space Center in Tsukuba, Japan, between April 3 and 6, 1999. The scientific analysis of VIABLE mission is foreseen in next months in collaboration with University research centers.

Acknowledgements

The authors would like to thank all NASDA officers and particularly Dr M. Oda, Mr. N. Inaba and Dr. N. Doi for the help and support in the succesful realization of the VIABLE mission.

References

1. Oda, M., et al - ETSVII, Space Robot in-Orbit Experiment Satellite. Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis,
2. NASDA - ETS-VII NASDA ground system for DARA(ESA) ground system interface, Issue B, 2-Sept-97
3. NASDA - Interface specification between NASDA and foreign space agencies on ETSVII robot experiments, 10 May 1996
4. NASDA - Interface Control Specification (ICS) Part II On-Ground Systems and Mission Operations (Preliminary Draft), 13-May-96
5. TRASYS-VIABLE Operation Plan Document, VIA-TRA-TN-OPP V1.0, 20/08/98
6. TRASYS-VIABLE Experiment Definition Review VIA-TRA-TN-EDRD V1.0, 11/06/98

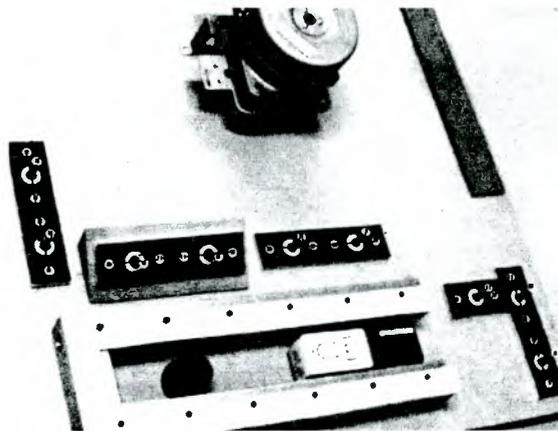


Figure 3.2: Prototype with marker position superimposed

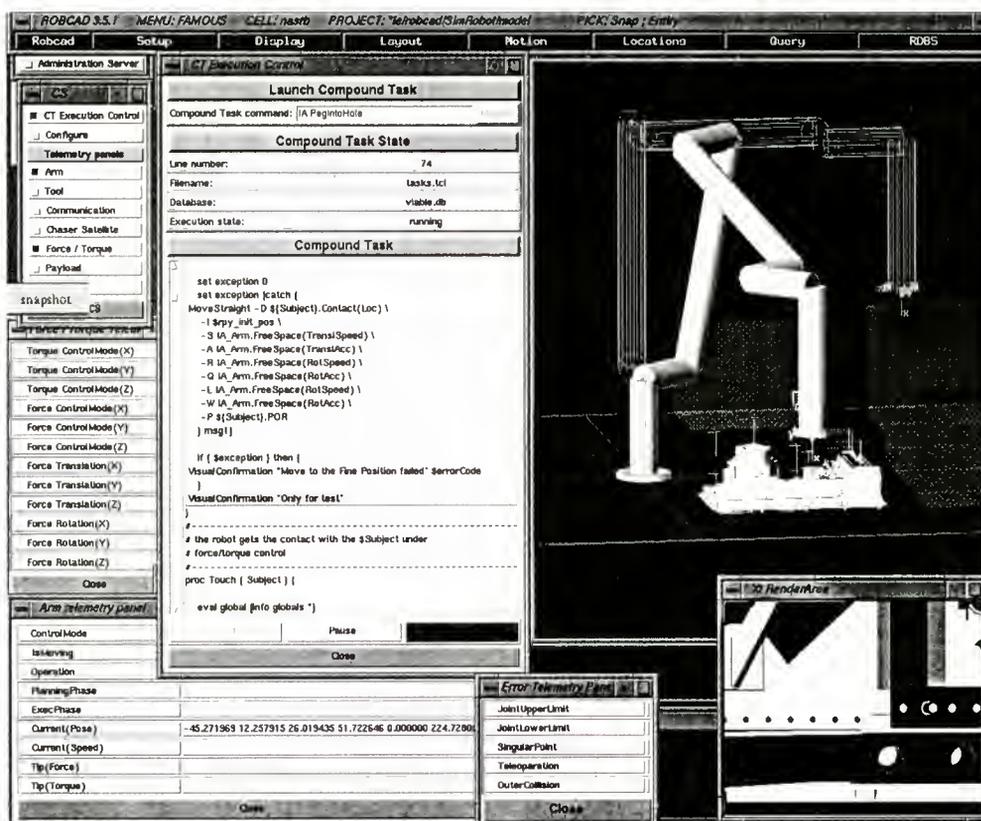


Fig 3.4: The VIABLE Station layout



Fig. 3.4: The ESA/TRASYS team during the operation in Tsukuba



Fig. 3.5: The NASDA Team during Operations

THE RANGER TELEROBOTIC SHUTTLE EXPERIMENT: AN ON-ORBIT SATELLITE SERVICER

Joseph C. Parrish

NASA Headquarters Office of Space Science, Advanced Technologies & Mission Studies Division
Mail Code SM, Washington, DC, 20546-0001, U.S.A.

phone: +1 301 405 0291, fax: +1 209 391 9785, e-mail: jparrish@hq.nasa.gov

Abstract

The Ranger Telerobotic Shuttle Experiment (RTSX) is a Space Shuttle-based flight experiment to demonstrate key telerobotic technologies for servicing assets in Earth orbit. The flight system will be teleoperated from onboard the Space Shuttle and from a ground control station at the NASA Johnson Space Center. The robot, along with supporting equipment and task elements, will be located in the Shuttle payload bay. A number of relevant servicing operations will be performed—including extravehicular activity (EVA) worksite setup, orbit replaceable unit (ORU) exchange, and other dexterous tasks. The program is underway toward an anticipated launch date in CY2001, and the hardware and software for the flight article and a neutral buoyancy functional equivalent are transitioning from design to manufacture.

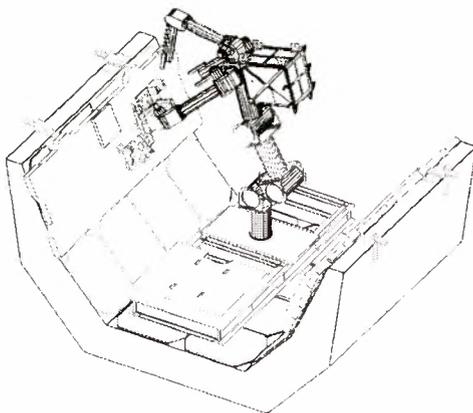


Figure 1: Ranger in Space Shuttle payload bay.

1 Introduction

As space operations enter the 21st century, the role of robotic satellite servicing systems will increase dramatically. Several such systems are currently in development for use on the International Space Station, including the Canadian Mobile Servicing System (MSS)[2] and the Japanese Experiment Module Remote Manipulator System (JEM-RMS). Another Japanese system, the Experimental Test System VII (ETS-VII)[1], has already demonstrated the ability for rendezvous and docking, followed by ORU manipulation under supervisory control. Under development by the United States, the Ranger Telerobotic Shuttle Experiment (RTSX)[3] is progressing toward its mission on the NASA Space Shuttle to demonstrate telerobotic servicing of orbital assets.

The missions envisioned for the Ranger class of servicers are for attached (e.g., to a Space Station) and free-flying (e.g., to a communication satellite in geostationary orbit) operations such as inspection, maintenance, refueling, and orbit adjustment. The approach being taken with the first flight deployment of a Ranger spacecraft is for attached operation on a cargo pallet in the payload bay of the Space Shuttle, as shown in Figure 1.

The robot will perform a series of representative tasks, ranging from simple taskboard operations to very complex EVA worksite setup using hardware that was never intended for robotic handling. In addition to obtaining performance data on these task operations, a major aspect of the Ranger mission is to compare performance via local and remote teleoperation. Several of the tasks will be repeated with varying control modalities and time delays to compare these effects. The robot will be controlled from flight and ground control stations, with commands and telemetry transferred via the normal Shuttle communications path (Figure 2).

The experiment is sponsored by Telerobotics Pro-

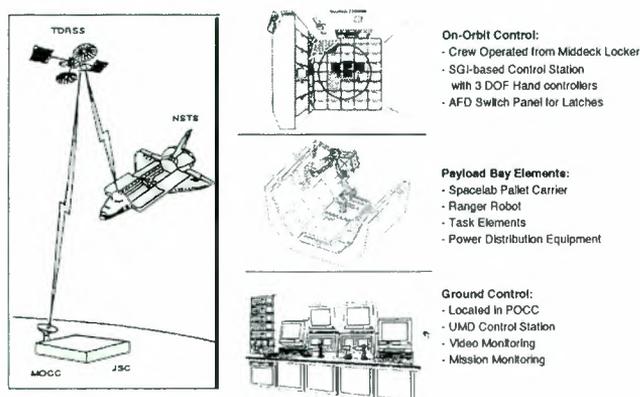


Figure 2: RTSX mission overview.

gram in NASA's Office of Space Science, and is executed by the University of Maryland under a cooperative agreement. In addition to the ground-breaking demonstrations of telerobotic servicing capabilities, the Ranger program also serves as a training program for young engineers in a truly hands-on environment. The Space Systems Laboratory at the University of Maryland College Park campus has an operational neutral buoyancy version of the Ranger robot—designed and built largely by students—and is gathering operational experience with the system at their own neutral buoyancy facility and at other NASA centers.

2 Mission Objectives

The RTSX mission objectives address three major areas[4]. The first is demonstrating a series of tasks that are representative of a wide variety of extravehicular operations, thus showing the utility and application of a dexterous robotic servicer. Second are the human factors effects of controlling space telerobots, including time delay, microgravity, and advanced control interfaces. Finally, the RTSX mission will provide flight data for comparison and correlation to hundreds of hours of data from ground-based computer and neutral buoyancy simulations.

2.1 Task Demonstrations

The first set of task operations involve tasks that have been designed with robotic compatibility in mind. These tasks provide collocated grasp points and fasteners, along with visual cues to support grasp point acquisition and fastener status indication. They are typically performable with a single manipulator arm,

freeing a second manipulator (if available) for stabilization functions or as a functional spare. These tasks obviously have the lowest relative complexity and the highest chance of mission success. However, the RTSX experiment is attempting to define the limits of space telerobots, so a more challenging set of tasks will be attempted.

A second set of operations involve tasks that were originally designed only for EVA astronauts. Although EVA astronauts lack the dexterity of humans in a shirt-sleeve environment, they do have greater dexterity than most robotic systems envisioned for space operation. EVA tasks can require multiple arms for performance, and typically don't provide integrated handholds with fasteners.

A major objective of the RTSX mission is to demonstrate that space robots (equipped appropriately to interface with the hardware) can perform tasks having no special provisions beyond general EVA compatibility. This would greatly increase the set of conceivable tasks, including setup and teardown of EVA worksites—which add considerably to the overhead of EVA operations without directly contributing to the achievement of maintenance objectives.

2.2 Human Factors

Figure 3 shows the overall human factors science strategy for the RTSX mission. The two upper boxes represent operations performed on-orbit, while the lower two boxes represent operations performed from the ground. The three main effects on human factors—time delay, microgravity, and advanced operator interfaces—are decoupled to allow a clear assessment of their relative influences.

The time delay associated with ground controlled operations on the Space Shuttle may range from 5–7 seconds[5]. Any time delay greater than 0.3 seconds causes the operator to adopt a “move-and-wait” control strategy that increases the task performance time[6]. A set of robotic tasks will be performed on-orbit without time delay and then repeated with varying levels of time delay, giving a direct assessment of the effect of time delay. The effects of time delay on teleoperation has been an active topic of research at the Space Systems Laboratory.

Another significant difference between ground and on-orbit operations is the effect of microgravity. Clearly, this has a dramatic effect on the dynamics of the manipulators and manipulated elements, but there may also be effects upon the operator. It is possible to adequately restrain the operator to permit stable interaction with the control station, but the more subtle issues of lost vestibular cues and their impact on sit-

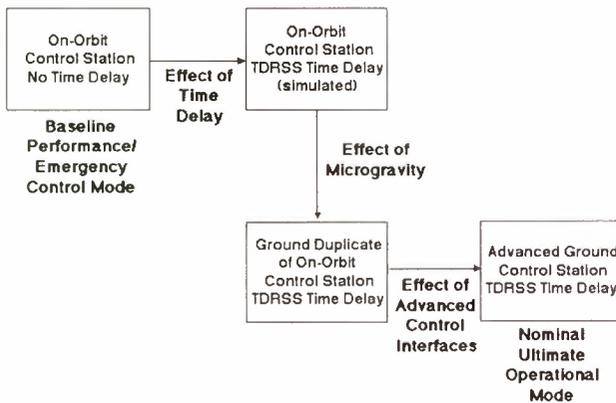


Figure 3: RTSX science strategy.

uational awareness are not well understood. Very few applicable research results are available in this area. To address this issue, functional duplicate control stations will be used on the ground and on-orbit, with equal time delay effects programmed. Therefore, the effect of time delay will be masked, and the effect of the microgravity environment may be directly measured.

Thus far, the input devices used to control space telerobots have been standard 2x3 degree-of-freedom (DOF) hand controllers; therefore, only a single manipulator can be controlled by a single operator. The only output devices have been simple monoscopic video and text displays. Initial research results[7] suggest that with intuitive 6 DOF input devices and higher fidelity output devices, it will be possible for a single operator to coordinate the operation of two 6+ DOF manipulators. A number of advanced output devices—such as head-mounted displays and stereo vision devices—promise to give operators a greater sense of telepresence than that offered by straight video and text. It may also be possible to mitigate the effects of time delay through the use of predictive displays. The ground control station will incorporate two sets of input and output devices; the first set will replicate the basic hand controllers and video displays of the on-orbit control station, while the second set will incorporate more advanced input and output devices, along with predictive displays for time delay compensation. The intent is to provide the most capable ground control station possible, and the “basic” control station will serve as the reference system.

2.3 Correlation of Flight Data to Ground Simulations

Clearly, on-orbit operational time for the RTSX mission will be limited. A number of ground simulations

have been developed to support the development of the RTSX flight hardware, assist in training the flight and ground crews, and support anomaly resolution during the mission. By correlating the RTSX flight data to the database obtained from ground simulations, it will be possible in the future to use the “calibrated” ground simulators to predict on-orbit performance for tasks that have not yet been envisioned.

The simulators take the form of graphical computer displays, and also as a neutral buoyancy equivalent to the RTSX system, known as the Ranger Neutral Buoyancy Vehicle (RNBV). A free-flying RNBV is already operational and collecting data on human factors and task operations. A second-generation RNBV which closely resembles the RTSX flight article configuration and system architecture is currently under construction. Once operational, this system will be shared between crew training and task operation data collection.

3 System Configuration

3.1 Cargo Bay Equipment

The Ranger robot, task equipment, and support equipment (Figure 4) will be carried to orbit on a Spacelab Logistics Pallet (SLP), and will remain anchored in the payload bay for the duration of the mission[8]. In the event of a contingency that prevents the safe return of the payload, the entire pallet can be jettisoned remotely. There are also provisions for EVA contingency servicing if sufficient mission resources are available.

The Ranger robot consists of a body and four manipulators. The body serves as the mounting point for the manipulators and end effectors, houses the main computers and power distribution circuitry, and is the anchor point for the manipulator launch restraints and the body latches. The body is made from aluminum sheet; the manipulator attachment structure is a monocoque, while the electronics housing is a framework with body panels. This construction is stiff, robust, and allows for easy serviceability.

The Ranger robot has three types of manipulators—two dexterous manipulators, one video manipulator, and one positioning leg. The dexterous manipulators are a 8 DOF R-P-R-P-R-P-Y-R design, 48 inches in length, and capable of outputting approximately 30 pounds of force and 30 foot-pounds of torque at their endpoints. A suite of interchangeable end effectors are available for the diverse task set. The video manipulator is a 7 DOF R-P-R-P-R-P-R design, 55 inches in length, and carries a stereo video camera pair at its distal end. The positioning leg is an actively-braked 6 DOF R-P-R-P-R-P design, 75 inches in length, and

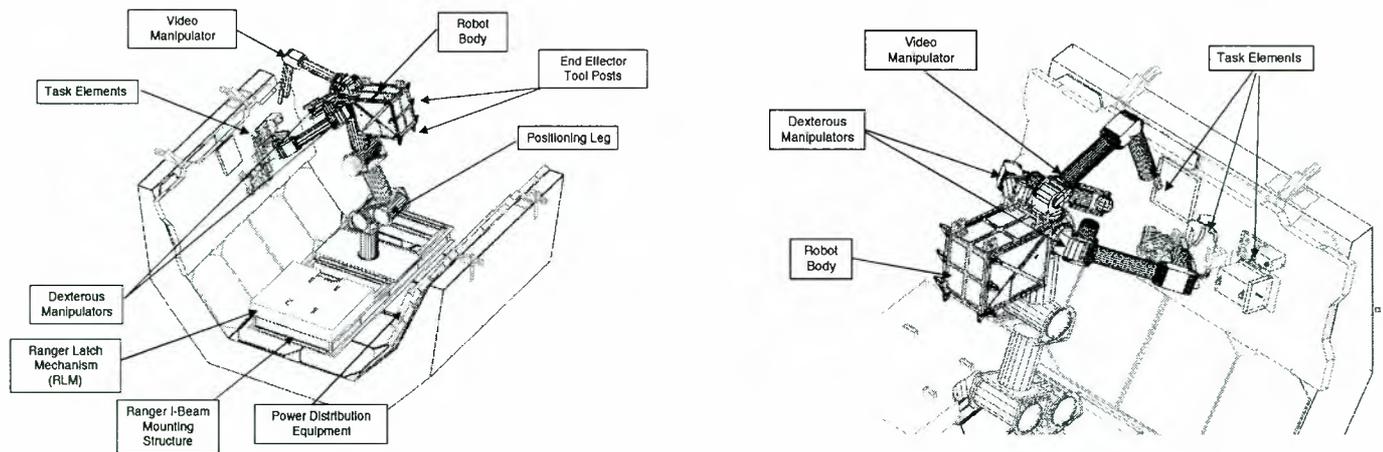


Figure 4: Ranger robot and task equipment on Spacelab Logistics Pallet.

capable of outputting 25 pounds of force and 200 foot-pounds of torque at its endpoint. In a braked condition, it can withstand a 250 pound load applied at full extension. It is permanently attached to the Spacelab Logistics Pallet for the RTSX mission.

The task element suite consists of the following components:

- International Space Station (ISS) Remote Power Controller Module (RPCM)
- Hubble Space Telescope (HST) Electronic Control Unit (ECU)
- ISS Articulated Portable Foot Restraint (APFR)
- Robotic task board

The RPCM changeout is considered to be a robot-compatible task. The RPCM ORU was designed from the outset to be serviced robotically, and incorporates misalignment tolerance, visual aids, and collocated grasp points/fasteners. The task can be performed with a single manipulator using simple motions.

The ECU is an ORU-style box that was changed out on the first HST servicing mission in 1996. It does not have collocated grasp points/fasteners, and will require coordinated dual arm operations.

The APFR is a complex, jointed device, designed to support EVA operations. It is by far the most difficult task on the RTSX mission, requiring four different end effectors, multiple arm coordination, and numerous task steps. Successful execution of this task on-orbit will help to validate the concept of telerobotic setup of EVA worksites.

The task board is comprised of a number of smaller task operations, including a set of calibrated force and torque measuring sensors, a contour-following task, a

peg-in-hole task, and a visual inspection task board provided by the NASA Jet Propulsion Laboratory.

The support equipment on the SLP include electrical power conditioning and switching units, a body and manipulator latching system, and a contingency stowage box. The electrical power equipment includes DC-DC converters, filters, and relays to support the robot and the latching system. The latching system is based on a flight-proven design used for NASA's SPARTAN free-flying satellite; it secures the robot body and manipulators for launch and re-entry.

3.2 Crew Cabin Equipment

Most of the RTSX-related crew cabin equipment is located in the Middeck. Figure 5 shows the Shuttle Middeck, with RTSX flight control station (circled) deployed and attached to the middeck lockers, facing forward. The RTSX flight control station consists of a Silicon Graphics, Inc. O2™ workstation, keyboard, four flat-panel graphics and video displays, hand controllers, and networking and video processing equipment. The flight control station is stowed in Middeck lockers when not in use; the keyboard, hand controllers, and displays are deployed for RTSX operations. Additional RTSX-dedicated items in the Middeck include a Payload General Support Computer (PGSC) for monitoring Orbiter parameters, and video and still cameras to document RTSX operator interactions with the payload.

The switches that control the payload retention latches and the payload jettison function are located on switch panels in the Aft Flight Deck. If an observer is deemed necessary for experimental data collection or safety purposes, they would use direct out-the-window views and/or video displays from the Aft Flight Deck.

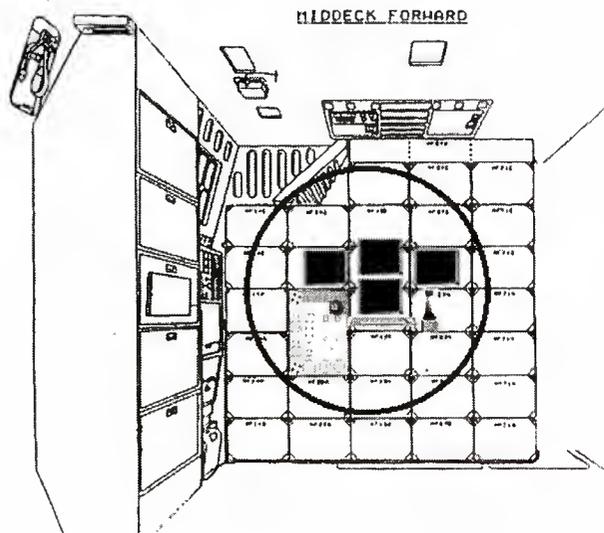


Figure 5: RTSX flight control station.

3.3 Ground Equipment

The ground control station (Figure 6) has two operator stations to support the requirements for a functional duplicate of the flight control station and an advanced control station. The ground control station will be located in the Payload Operations Control Center (POCC) at the NASA Johnson Space Center. It will tie into the payload data network and will serve both in the operational function described in Section 2.2 and as a monitor and archive for data when the flight control station is active.

Like the flight control station, the ground control station is based on Silicon Graphics, Inc. workstations. The included peripherals are graphics and video display monitors, hand controllers and other input de-



Figure 6: RTSX ground control station.

vices, and video and data processing and archiving equipment. The architecture of the ground control station is modular; the main robot control modules are also used in the flight control station. Some other modules are unique to the ground control station; these include the interfaces to the advanced input and output devices, the simulation modules, and a module that forwards mission data back to the University of Maryland.

Figure 7 shows the user interface for the ground control station. It is highly graphical, and has the ability to display video from the downlinked data stream. A subset of the ground control station functionality will be implemented on the flight control station; the flight control station will lack the advanced input and output devices and predictive displays, and will be optimized for a single operator.



Figure 7: Ground control station user interface.

The Ranger Neutral Buoyancy Vehicle supports both operational and scientific objectives in the RTSX mission. While the first generation RNBV shown in Figure 8 is a free-flying configuration, the second generation RNBV is a functional equivalent of the RTSX robot, and is deployed on a neutral buoyancy mockup of the SLP and its associated task equipment. The RNBV structure is similar in form to the RTSX robot. The manipulator arms are almost exact duplicates of the flight arms, except for seals in the joints and surface finishes. The neutral buoyancy environment poses several significant challenges, namely the need to waterproof all exposed elements and to ensure that structure is strong enough to withstand pressure effects and the rough treatment inherent to the underwater environment. The RNBV will be surface-supplied with

pressurized air, electrical power, and fiber optic data and video lines.

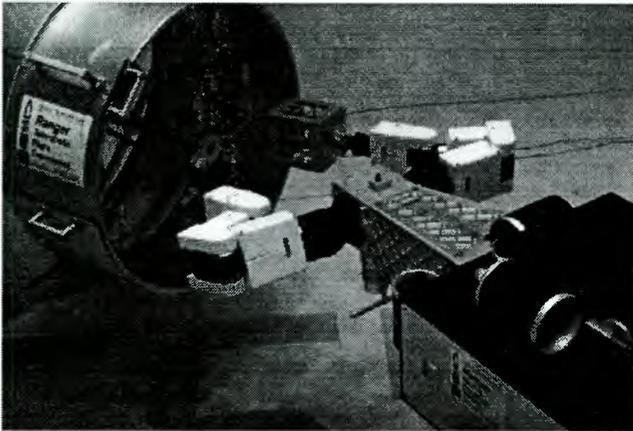


Figure 8: Ranger Neutral Buoyancy Vehicle.

Operationally, the RNBV should be an excellent replica of the flight system. Manipulator motions can be kept slow to minimize water drag effects, and the task elements can be made neutrally buoyant to simulate weightlessness. However, it will be difficult to replicate the on-orbit lighting conditions, and external flotation may be required to make the manipulators and end effectors neutrally buoyant. These issues notwithstanding, neutral buoyancy is the best simulation medium for on-orbit dexterous robotic operations, and the RNBV is a key element of the RTSX mission.

4 Operations Concept

4.1 Mission Operations

RTSX is expected to be either a primary payload or a complex secondary payload, due to crew time requirements. The RTSX mission is expected to involve approximately 48 hours of operations, divided between ground and flight control. (The flight control station will be in a monitoring mode during ground controlled operations, and vice versa.) For mission day planning and crew fatigue considerations, the 48 hours will be divided into approximately 12 four-hour sessions.

The RTSX does not have fine pointing requirements, but does expect a relatively benign thermal environment during task operations; therefore, a payload bay-to-Earth flight attitude has been requested. Orbiter thruster firings are expected to be deferred so as not to disturb task operations. Finally, no EVA operations are required for the nominal RTSX mission; however, EVA may be used to recover from an RTSX failure

that prevents safe return if crew resources and mission time are available.

4.2 Session Operations

The twelve test sessions are designed to support the population of the test matrix (ground vs. on-orbit, predictive display vs. no predictive display, etc.) while achieving mission success at the earliest possible time. Only one IVA crewmember will be required to operate the flight control station, although an additional crewmember(s) may serve as a safety monitor or video/still camera operator.

A typical four-hour session will consist of robot power-up and checkout operations (approximately 30 min.), task operations (approximately 3 hrs. 15 min.), and robot stowage and power-down operations (approximately 15 min.). The task operations segment may be further sub-divided to account for ground and flight control, or to sequence through more than one on-orbit operator. If the ground control station is active, control will automatically revert to the flight control station if communications are interrupted.

4.3 Task Operations

Figure 9 gives two representative views of a task operation. This particular task is a changeout of the Hubble Space Telescope (HST) Electronics Control Unit (ECU); the left view is as might be provided by a Ranger body-mounted camera; the right view is as provided by a video manipulator camera. Although the task operations will be extensively practiced via computer and RNBV simulations, the robot will be teleoperated on-orbit. Only a few operations, such as robot deployment/stowage and end-effector changeout, will be automated. Time to complete a particular task will range from a few minutes—in the case of the task board elements and the RPCM—to possibly several sessions for the APFR task.

The RTSX hardware and software design are strongly influenced by the requirement to ensure that the robot does not pose a hazard to the Orbiter or its crew. The hazards include inadvertent contact between the robot and the Orbiter, excessive loads into task equipment, inability to safely stow the robot for landing, and potential hazards to EVA crewmembers. The NASA Jet Propulsion Laboratory is developing a methodology[9] to detect potential collisions between the Ranger and itself or with its surrounding environment. The RTSX computer architecture is highly failure tolerant, and has a hierarchical monitoring approach that permits any processor to shut down an adjacent upstream or downstream processor. The control

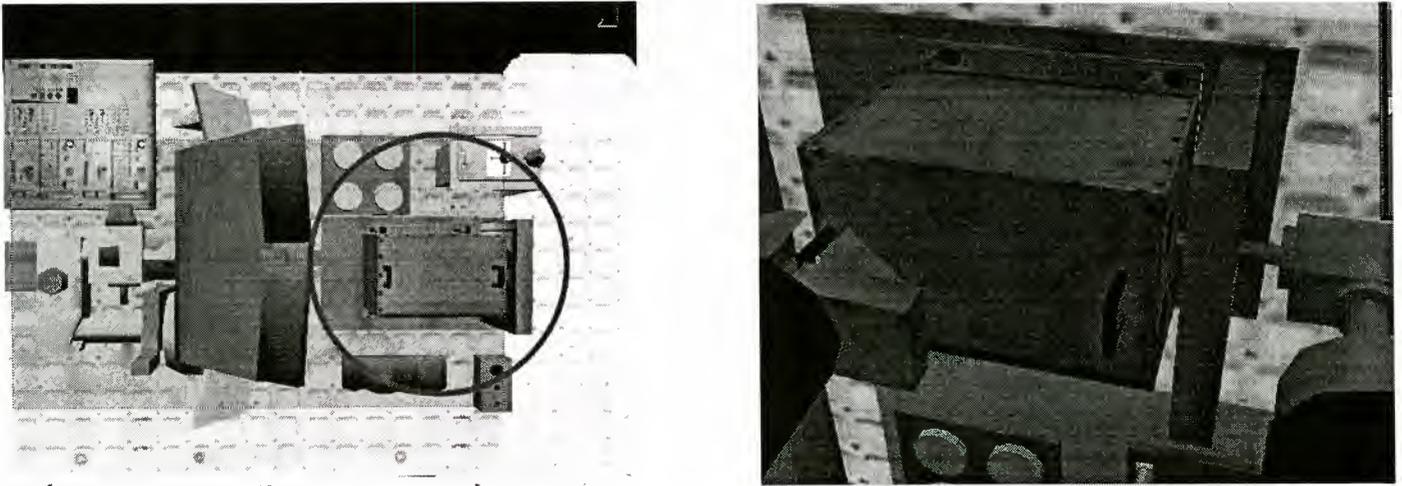


Figure 9: Ranger performing ORU changeout operation.

stations play no active role in the safety of the system, and an inadvertant operator command or loss of communication will not result in a hazardous condition.

5 Outlook

5.1 RTSX Mission Outlook

The RTSX project has completed the preliminary design phase and is undergoing detailed design. The manipulators are leading the development process, with the body and associated subsystems following shortly thereafter. The body structure for the RTSX-equivalent Neutral Buoyancy Vehicle has been manufactured and is awaiting outfitting with power, data, and pressurization subsystems. Hardware and software integration for the flight article are planned for late CY1999, with environmental testing in middle CY2000 in anticipation of a Space Shuttle launch opportunity in CY2001.

5.2 RTSX Follow-on Mission Outlook

A successful RTSX mission will set the stage for several possible follow-on scenarios. A logical follow-on to the pallet-based RTSX configuration would be a free-flying system, named the Ranger Telerobotic Flight Experiment (RTFX), which has already been conceptually designed[10]. Another possible scenario would be to deploy Ranger to a long-duration platform such as the International Space Station to extend the experimental database. Finally, there are a number of candidate assets in Earth orbit that could benefit from servicing; the lowest risk approach would be to demonstrate

free-flying servicing on a failed spacecraft that would not otherwise be recoverable. These scenarios are, of course, dependent on a successful first mission with the RTSX, and this is where the Ranger development team is focusing its efforts.

Acknowledgements

The Ranger Telerobotic Flight Experiment (NCC5-243) is funded by the Space Telerobotics Program of the NASA Headquarters Office of Space Science, Advanced Technology & Mission Studies Division (Code SM).

The author would like to gratefully acknowledge Martin Devaney, J. Corde Lane, and Cristin Sawin of the University of Maryland Space Systems Laboratory for contributing many of the figures used in this manuscript.

References

- [1] M. Oda, K. Kibe, and F. Yamagata. ETS-VII: Space robot in-orbit experiment satellite. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 739–744, Minneapolis, MN, April 1996.
- [2] W. Brimley, D. Brown, and B. Cox. Overview of international robot design for space station freedom. In S. B. Skaar and C. F. Rouff, editors, *Teleroperation and Robotics in Space*, volume 161 of *Progress in Aeronautics and Astronautics Series*, pages 411–441. AIAA, Washington, DC, 1994.

- [3] J. C. Parrish. The Ranger Telerobotic Shuttle Experiment: Status report. In M. R. Stein, editor, *Telemanipulator and Telepresence Technologies V*, volume 3524, pages 1–12. SPIE, 1998.
- [4] University of Maryland Space Systems Laboratory, College Park, MD. *Ranger Telerobotic Shuttle Experiment (RTSX) Experiment Definition and Mission Requirements Document*, DT07-0020 edition, March 1999.
- [5] G. Hirzinger et al. Sensor-based space robotics—ROTEX and its telerobotic features. *IEEE Transactions on Robotics and Automation: Special Issue on Space Robotics*, 9(5):649–663, 1993.
- [6] R. Held, A. Efstathiou, and M. Greene. Adaptation to displaced and delayed visual feedback from the hand. *Journal of Experimental Psychology*, 72:887–891, 1966.
- [7] J. C. Lane, C. Carignan, and D. L. Akin. Reconfigurable control station design for robotic operations. In *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, Orlando, FL, October 1997.
- [8] University of Maryland Space Systems Laboratory, College Park, MD. *Ranger Telerobotic Shuttle Experiment (RTSX) Critical Design Review Data Package*, May 1999.
- [9] B. Bon and H. Seraji. Real-time model-based obstacle detection for the nasa ranger telerobot. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- [10] J. C. Parrish and D. L. Akin. The Ranger Telerobotic Flight Experiment: Mission, technologies, and programmatic. In L. A. Demsetz, editor, *Robotics for Challenging Environments: Proceedings of RCE II, the Second Conference*, pages 136–142, Albuquerque, NM, 1996. ASCE.

THE ISS INSPECTOR MISSION

D. Wilde, U. Brüge
Daimler-Benz Aerospace Infrastructure (Germany),

Sytin O.,
RSC-Energia (Russia).

Paper not available

Rover Systems (2)

AUTONOMOUS NAVIGATION FIELD RESULTS OF A PLANETARY ANALOG ROBOT IN ANTARCTICA

Stewart Moorehead, Reid Simmons, Dimitrios Apostolopoulos and William "Red" Whittaker

The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA. 15213
U.S.A.

phone: 1-412-268-7086, fax: 1-412-268-5895, email: sjm@ri.cmu.edu
phone: 1-412-268-2621, fax: 1-412-268-5576, email: reids@ri.cmu.edu
phone: 1-412-268-7224, fax: 1-412-268-1488, email: dalv@ri.cmu.edu
phone: 1-412-268-6556, fax: 1-412-682-1793, email: red@ri.cmu.edu

ABSTRACT

The Robotic Antarctic Meteorite Search at Carnegie Mellon is developing robotic technologies to allow for autonomous search and classification of meteorites in Antarctica. In November 1998, the robot Nomad was deployed in the Patriot Hills region of Antarctica to perform several demonstrations and experiments of these technologies in a polar environment.

Nomad drove 10.3km autonomously in Antarctica under a variety of weather and terrain conditions. This paper presents the results of this traverse, the ability of stereo vision and laser scanner to perceive polar terrain and the autonomous navigation system used.

1 INTRODUCTION

From the Lunakhods on the Moon to Sojourner on Mars [6], mobile robots have demonstrated their usefulness to planetary exploration. As future missions become more ambitious, mobile robots will be required to do more tasks in shorter periods of time necessitating an increased level of autonomy. In particular, mobile robots will be called upon to drive long distances with little or no supervision to achieve the goals of planetary science.

As one of the harshest environments on Earth, Antarctica is a unique place to test planetary robotic technologies. The low temperatures, lack of communications and remoteness make it an interesting terrestrial analog of the Moon and Mars. In November of 1998, the robot Nomad (Figure 1) was deployed to the Patriot Hills region (80S, 81W) of Antarctica. This deployment was part of Carnegie Mellon's Robotic Antarctic Meteorite Search program [3] which is developing robotic capabilities to perform Antarctic meteorite searches from a mobile robot. The expedition demonstrated autonomous naviga-

tion in polar terrain and meteorite detection/classification [9]. Experiments were also performed on characterizing laser and stereo sensors [14], systematic patterned search [10], ice and snow mobility, landmark based navigation and millimeter wave radar [1]. Foot search by the expedition found two meteorites [5].



Figure 1: Nomad at the Patriot Hills

Very few robots have been deployed to Antarctica. TROV [13] and SARA [7] explored the underwater coastal regions and Dante I [15] the volcano Mt. Erebus. However, to the authors' knowledge no robot for cross country navigation in polar terrain has been demonstrated. This meant that many factors were unknown before the expedition such as the ability of stereo and laser sensors to see obstacles on snow and ice fields. This uncertainty necessitated the development of a robust autonomy system.

This paper presents a description of the autonomy system implemented on Nomad in Antarctica and presents the results of its autonomy tests.

2 NAVIGATIONAL AUTONOMY SYSTEM

The autonomy system drives Nomad through a series of waypoints while avoiding any obstacles too large for the robot to drive over. It is descended from that found on Ratler [12] and Nomad in the Atacama [16] but differs in several ways. An error recovery module has been added which lets Nomad backup and turn when it is blocked by obstacles or exceeds its roll and pitch specifications. The representation of terrain has been changed to indicate how good it is to occupy a cell and the certainty of that goodness. Finally, the laser has been fully integrated into the autonomy system. These changes have made the system more reliable and robust.

Figure 2 shows the structure of the autonomy system. Except for the controller each box is a separate Linux process running on a single Pentium Pro 133 located on Nomad. The arrows indicate interprocess communications using the Task Control Architecture's (TCA) message passing capability [11] and the arrow labels indicate the type of information passed. Messages can also be passed with TCA over a wireless ethernet link to user interface processes running on an external computer. The controller is implemented on a 68060 running VxWorks and performs the low level motor control.

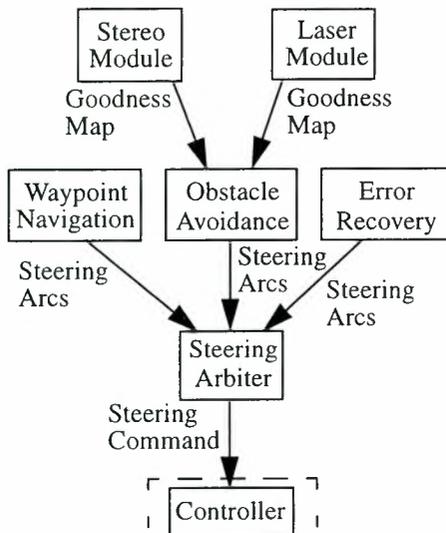


Figure 2: Autonomy System

2.1 GOODNESS MAPS

To model the environment it passes through, Nomad uses a map structure called a goodness map (an example map can be found in Figure 4). Goodness maps are fixed reso-

lution grid based maps (Nomad currently uses a 50cm grid resolution) where each cell contains two numbers: a goodness score indicating the desirability of the robot occupying that cell and a certainty score which indicates the reliability of the goodness score. Each of these numbers are normalized between 0 and 1. Additionally, any cell with a certainty less than a lower threshold is considered unknown and certainty and goodness values are set explicitly to zero.

Multiple goodness maps can be combined by taking an average of the goodness values in corresponding cells, weighted by their certainties. If the maps are created by different sources, then a weight for the confidence in that source is also used in the average. For example, on Nomad both the stereo and laser modules create goodness maps from their sensor readings. The obstacle avoidance module maintains a local terrain map by combining these sensor produced goodness maps, weighted by our confidence in that sensor, with its own map.

Nomad only uses the perceived roughness of the terrain or traversability to determine the goodness of a cell. However, the goodness map representation is general enough to incorporate other measures in the determination of cell goodness. By using multiple criterion when determining cell goodness, a goodness map provides a unified format to balance competing goals. For example, a goodness map which combines the terrainability, science interest and the potential for solar power of a cell would help the robot make trade offs between the three criterion and choose paths which satisfy all of the constraints.

2.2 STEREO MODULE

Nomad has four Sony XC-77 640x480 B&W CCD cameras mounted on a sensor yard 1.67m above the ground at the front of the robot (Figure 3). Each camera has a Computer HAS3616APC auto iris, 3.6mm focal length lens. To operate in the cold temperatures of Antarctica the cameras are enclosed in insulated, heated boxes.

Since Nomad is quite wide and able to turn relatively sharply the four cameras are set up as two stereo pairs - one pair looking right the other looking left. They are strongly calibrated using the procedure in [8]. The raw images are first dewarped to remove radial lens distortion and then rectified so that the epipolar lines lie on the scan lines.

To reduce the cycle time only a small number of rows in the image are examined by the stereo module. These rows correspond to distances of 4.5m to 8.5m in front of the robot. The stereo module computes the disparity map in this region and takes the (x,y,z) pixel coordinates to create a goodness map by using a plane fitting technique. For

each cell in the goodness map, stereo fits a plane to the data in a region equal to the size of the robot (a 5x5 grid cell area) centered at the active cell. Smaller planes are also fit to each cell in this 5x5 submap. The goodness score of the center cell is then determined by the roll and pitch of the planes as well as the residual from fitting the planes. The certainty is derived from the number of data points used to create these planes. This process produces a goodness map where the goodness of a cell is the lowest goodness of all cells in a 5x5 area. Therefore obstacles are expanded into configuration space format allowing planning to consider Nomad as a point robot [4]. The goodness map created depends only on the current stereo image.

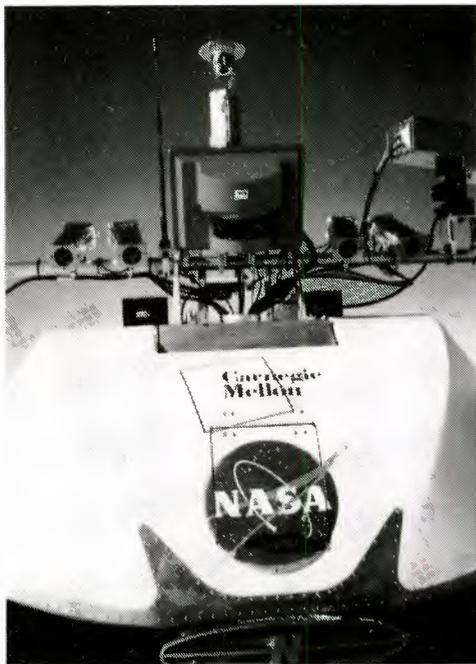


Figure 3: Nomad's sensor yard with 4 CCD cameras and SICK LMS 220 single line scan laser unit.

2.3 LASER MODULE

Nomad uses a SICK LMS 220 single line scan laser unit as a second sensor to detect obstacles. It is capable of generating distance measurements in a 180° field of view in increments of 0.25°. In practice, the autonomy system uses a scan of 100° in increments of 1°.

The output of the laser module is a goodness map which indicates the terrainability of the map squares illuminated by the laser sensor. The goodness map is created by first fitting a line to the complete laser scan using a least squares method. This line is considered as the ground. Next, the deviation of each laser measurement from the ground line is computed. The goodness of a cell is then

inversely proportional to the average deviation of all the laser measurements in the cell. Cell certainty is proportional to the number of measurements present in the cell. Cells with goodness values below 0.5 are expanded to fill the 5x5 cell area around them, providing configuration-space obstacles in the map [4]. A large change in the level of the ground line from the previous scan indicates a step feature - such as a cliff - so in this case all map cells with laser measurements are marked with low goodness. Other than the previous ground level, the goodness map produced is based entirely on the current scan.

2.4 OBSTACLE AVOIDANCE

The obstacle avoidance module, named Morphin, is the heart of the navigation system. It maintains a goodness map of the environment around the robot. This map is generated by merging the goodness maps created by the stereo and laser modules. Unlike the sensor goodness maps, Morphin's map contains data from previous sensor module maps. When a new sensor module map arrives Morphin ages its current map by multiplying the certainties of each cell by a number less than 1. It then merges in the new data using the cell certainties and sensor type to weight each goodness value. In this way new data is added to Morphin's world view and older data becomes less sure until finally it disappears from the map.

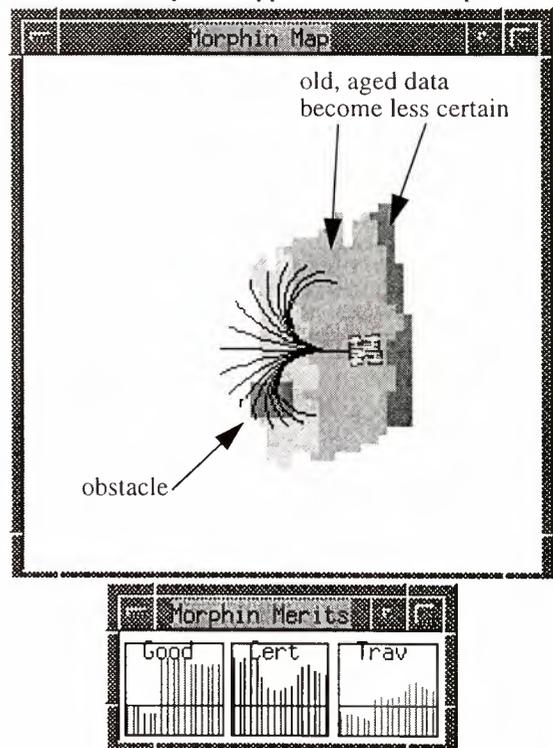


Figure 4: Morphin goodness map with potential driving directions. The votes for each driving direction are indicated by the height of the bars in the Trav window.

Using its goodness map, Morphin evaluates a set of steering arcs. The arcs represent how Nomad would travel on the terrain if it were next commanded to steer in a given direction. Since Nomad turns slowly relative to its nominal travel speed of 30cm/s, Morphin takes steering latency into account when computing travel paths. Each arc is given a score on how good it is to travel along it. If an impassable obstacle is present along the path, the arc is vetoed. The arc votes are then sent to the steering arbiter.

A typical Morphin goodness map with driving arcs and their votes is shown in Figure 4. The map displays the goodness values as different colors and the certainty as different brightnesses. The dark square at the bottom left is an obstacle, expressed in configuration space. Older data is aged or made less certain. This is shown by the darkening of the cells from left to right (the robot is driving to the left). The arcs which Morphin evaluates are drawn over the map starting at the current vehicle position. The Morphin Merits window below the map shows the sum of the goodness and certainty along each arc in the Good and Cert frames. The Trav frame combines the two criterion and is the final vote from Morphin for each arc. Votes below the horizontal line indicate vetoed arcs, and correspond to those arcs passing through the obstacle.

2.5 WAYPOINT NAVIGATION

The waypoint navigation module takes a list of differential global positioning system (DGPS) coordinates as input from a remote human operator. Waypoint prefers Nomad to drive straight towards the current waypoint. It generates votes on the same set of steering arcs as Morphin. The magnitude of the votes are distributed as a Gaussian centered in the direction of the goal. These votes are then sent to the steering arbiter. Once the robot's position is within some specified error radius of a waypoint, the next point in the list becomes the current goal.

2.6 ERROR RECOVERY

The error recovery module has two purposes. The first is to monitor the status of the robot, detecting when a problem has arisen. The second is to initiate an action that will help solve the problem.

In its current form the error recovery module is able to monitor for two problems. The first is when the robot is unable to move because all of the possible travel directions are vetoed due to obstacles. The second is to monitor the roll and pitch of the vehicle to determine when the robot has driven on bad terrain missed by the terrain sensors. This second mode is also referred to as blind driving.

If either of the two problems is detected, error recovery will suspend Morphin and waypoint navigation and initiate a back up maneuver. This causes the robot to back up along its previous route (since Nomad has no sensors looking back this is the safest way). After a fixed time backing up, Nomad will turn in the direction opposite to where it had been driving and then re-enable Morphin and waypoint navigation.

2.7 STEERING ARBITER

The steering arbiter takes the votes provided by Morphin, and the waypoint navigation modules and combines them to decide on Nomad's actual steering direction. Each module is given a weight indicating its importance. If any module vetoes an arc, arbiter will not select that arc. The arc with the highest vote is chosen and an appropriate steering command is issued to the controller.

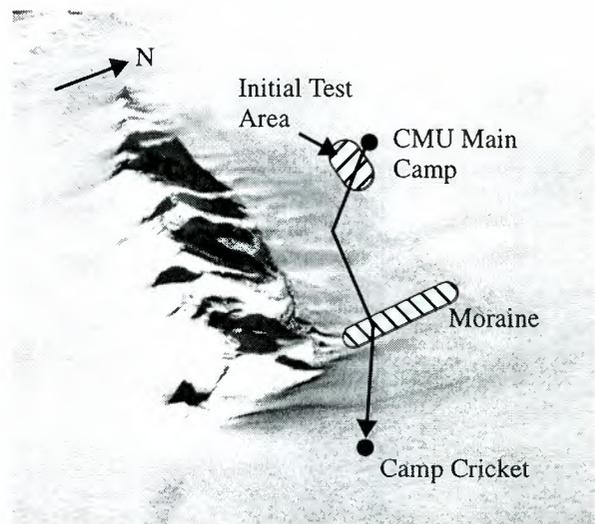


Figure 5: Patriot Hills, Antarctica. The map shows the major areas of operation and the path is Nomad's autonomous trip to the east end of the hills.

3 EXPERIMENTAL RESULTS

The autonomy system described in section 2 was tested on Nomad at Patriot Hills, Antarctica. Patriot Hills contains examples of three common Antarctic terrain types: snow, blue ice and moraine. The snow fields generally consist of hard packed snow which has been sculpted by the wind to form small dunes called sastrugi. Most sastrugi in the area were 10 to 20cm in height allowing Nomad to drive over them without difficulty. The blue ice fields are generally very flat with small (5cm diameter) depressions, called sun cups, covering the surface. Moraines are the most difficult terrain for robot navigation but also the most interesting for meteorite search. Moraines are areas on the blue ice fields where there are

large collections of rocks. Rock size and density varies depending on the moraine but the Patriot Hills moraine was sparsely distributed with most rocks being 40cm or more in diameter and posing a hazard to Nomad.

Taking advantage of Patriot Hills' varied terrain, Nomad's autonomy system was tested in all three terrain types, driving autonomously for a total of 10.3 km during the expedition. Of this distance, 4.7 km was spent driving in the snow field south of the main camp and the moraine. The remaining 5.6 km was made up of the trek from the main camp to the east end of the Patriot Hills (Figure 5).

3.1 EVALUATION OF TERRAIN SENSORS

The ability of a robot to sense its environment is an important capability for autonomous navigation. Thus an important component of Nomad's autonomy tests was the evaluation of its terrain modeling sensors - stereo and laser - in the different terrains and weather conditions of Antarctica.

As on the physical parts of the robot, Antarctica is harsh on traditional outdoor robotic terrain sensors. Sensors must be placed inside sealed, heated enclosures to prevent damage from snow and cold. The reflective property of the ground varies from the Lambertian snow fields to the specular blue ice fields and everywhere the color is an almost uniform white or blue.

During Nomad's tests in the Atacama and Pittsburgh, stereo provided the most information to the navigation system. This was because it provided terrain information from an area rather than just the single line from the laser. The stereo system was tested on snow, blue ice and moraine at the Patriot Hills as well as in three different weather conditions: sunny, cloudy and blowing snow. In all conditions stereo was not able to produce sufficiently dense disparity maps to be useful for navigation. Polarizing filters did improve performance on blue ice but still results were not sufficient for navigation. The terrain type had very little effect on the results (the moraine was sparse enough that most of a scene would be blue ice and not rocks). The weather however did have a large impact on the stereo results. Sunny days provided the best results with blowing snow a close second. Overcast conditions proved the most difficult for stereo. The clouds diffused the sunlight which, combined with the Lambertian surface of a snow field, made the illumination almost uniform everywhere. There was no contrast and it was very difficult, even for humans, to see depth. This phenomenon is referred to as a white out in [2]. During these conditions stereo was able to match very few points.

The single line scan laser unit was tested in the same conditions as the stereo system. The laser was unaffected by terrain type working as well in Antarctica as in pre-trial

tests in Pittsburgh. Even the specular surface of the blue ice fields had no effect on the return signal. Overcast conditions also had no effect on the active laser sensor. The laser did, however, have problems during periods of blowing snow. The laser could reflect off the snow flakes. If it reflected back to the laser unit a short distance would be measured. If it reflected away, no return signal would be received. During mild levels of blowing snow filtering was able to remove these effects. However in heavy storms, filtering did not work and the laser could not be used.

A more complete presentation of sensor results from Antarctica can be found in [14].

3.2 EVALUATION OF NAVIGATION AUTONOMY

For the duration of the expedition, stereo did not provide enough information to use in navigation and obstacle detection. Thus all of the navigation results were obtained using only the single line scan laser for obstacle detection. The navigation system was robust enough to handle the absence of stereo with only small changes to a configuration file of Morphin (unknown terrain's negative impact to an arc's score was reduced to zero).

The first set of navigation tests were performed on the snow fields near the main camp. During these tests the waypoint navigation module was given four waypoints in a rectangle 50x100m. Nomad continually drove around this course. Periodically, a human "volunteer" would step in front of Nomad. Since the laser sensor does not look far enough ahead to allow Nomad to turn and avoid an obstacle Morphin would veto all arcs when the person was seen. This would trigger the recover module which started a backup maneuver. Nomad successfully saw people, backed up, turned, drove past them and then resumed its rectangular course.

After these initial tests, Nomad embarked on a trek to the eastern end of the Patriot Hills. The trek proceeded in two segments. The first, from the main camp to the moraine, used the laser as its only sensor. The second leg, from the moraine to Camp Cricket, was performed during heavy snow which made the laser useless. The error recovery module's blind driving mode was the only sensor in use.

Since the moraine offered the highest density of impediments to travel for Nomad, several tests were performed there. Using only the laser, Nomad was commanded to drive to various places in the moraine. During these tests, Nomad encountered 12 rocks. It saw, and successfully avoided 9 of them. The other 3 rocks were not seen and required using the emergency stop button. These three rocks were missed because they got between the laser and the robot while Nomad was making a sharp turn.

4 CONCLUSIONS

Antarctica is a challenging environment for autonomous mobile robots and terrain sensing modalities. Stereo vision works poorly or not at all here. The vast majority of the terrain is made up of snow and ice fields which provide little texture for disparity matching. Stereo is further hampered in overcast conditions where the diffuse nature of the light eliminates all contrast, making it difficult even for human vision to work. The laser sensor works well on all terrains but heavy blowing snow reflects the beam causing false readings.

Despite the absence of stereo data, the autonomy system on Nomad was robust enough to drive 10.3km, detecting and avoiding several rocks with just the laser sensor. Nomad was driven on three major terrain types, snow, blue ice and moraine and in all weather conditions. The tests performed demonstrated the capability of autonomous navigation in polar terrains which is an essential component in the robotic search for meteorites in Antarctica.

Performance of the autonomy system can still be improved. Morphing should consider unknown terrain between the robot and the laser scan to be untraversable. This will help solve the problem of unseen obstacles getting between the laser scan and the robot during sharp turns. Another solution to this is to actively tilt the laser providing a scan over an area instead of just a line.

5 ACKNOWLEDGEMENTS

This work is based on grant NAG5-7707 from the Teleroobotics Program of NASA's Office of Space Science, Advanced Technology and Mission Studies Division.

The authors would like to thank the expedition team: Alex Foessel, Nicolas Vandapel, Christian Puebla, Fernando Valdes, Danilo Bassi, Bill Cassidy, Liam Pedersen, Mike Parris, Matt Deans, Pascal Lee and Mark Sibenac for their help during the field experiments. Thanks also go out to the other members of the Nomad development team.

A special thanks goes out to Mark Maimone and others who helped develop the autonomy system on Nomad in previous projects.

6 REFERENCES

- [1] Foessel, A. and S. Chheda, "Short Range MMW Radar Perception of Polar Terrain", *Field and Service Robotics Conference*, Pittsburgh, PA., 1999.
- [2] Horn, B. K. P., *Robot Vision*, The MIT Press and McGraw-Hill, 1986.
- [3] <http://www.frc.ri.cmu.edu/projects/meteorobot/>
- [4] Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publishers, 1991.
- [5] Lee, P. et. al., "Search for Meteorites at Martin Hills and Pirrit Hills", *Lunar and Planetary Society Conference*, 1999.
- [6] Mishkin, A., J. Morrison, T. Nguyen, H. Stone, B. Cooper and B. Wilcox, "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover", *Proc. IEEE Aerospace Conf.*, 1998.
- [7] Papalia, B. et. al. "An Autonomous Underwater Vehicle for Researches in Antarctica", *Oceans '94 Conference*, 1994.
- [8] Robert, L., "Camera Calibration without Feature Extraction", *INRIA Tech. Report No. 2204*, Feb. 1994.
- [9] Pedersen, L. et. al., "Robotic Rock Classification Using Visible Light Reflectance Spectroscopy: Preliminary Results from the Robotic Antarctic Meteorite Search Program", *Lunar and Planetary Society Conf.*, 1999.
- [10] Shillcutt, K., D. Apostolopoulos and W. Whittaker, "Patterned Search Planning and Testing for the Robotic Antarctic Meteorite Search", *Meeting on Robotics and Remote Systems for the Nuclear Industry*, 1999.
- [11] Simmons, R., "Structured Control for Autonomous Robots", *IEEE Transactions on Robotics and Automation*, 10:1, Feb. 1994.
- [12] Simmons, R. et. al., "Experience with Rover Navigation for Lunar-Like Terrains", *IROS*, p. 441-446, 1995.
- [13] Stroker, C., "From Antarctica to Space: use of telepresence and virtual reality in control of a remote underwater vehicle", *SPIE Proceedings 1995, vol 2352, Mobile Robots IX*, 1995.
- [14] Vandapel, N., S.J. Moorehead, W. Whittaker, R. Chatila, R. Murrieta-Cid, "Preliminary Results on the use of Stereo, Color Cameras and Laser Sensors in Antarctica", *Int'l Symposium on Experimental Robotics*, 1999.
- [15] Wettergreen, D., C. Thorpe and W. Whittaker, "Exploring Mount Erebus by Walking Robot", *Robotics and Autonomous Systems*, 11(3-4), 1993.
- [16] Wettergreen, D., D. Bapna, M. Maimone and G. Thomas, "Developing Nomad for Robotic Exploration of the Atacama Desert", *Robotics & Autonomous Systems Journal*, vol 26, pp. 127-148, 1999.

SOLAR POWER EXPERT FOR REMOTE ROBOTIC EXPLORERS

Kimberly Shillcutt

Robotics Institute, Carnegie Mellon University
Field Robotics Center, Pittsburgh, PA 15213, USA
phone: +1-412-268-7086, fax: +1-412-268-5895, e-mail: kimberly@ri.cmu.edu

William Whittaker

Robotics Institute, Carnegie Mellon University
Field Robotics Center, Pittsburgh, PA 15213, USA
phone: +1-412-268-6559, fax: +1-412-268-1338, e-mail: red@frc.ri.cmu.edu

ABSTRACT

Robotic exploration of remote areas to assist or replace human exploration reduces the cost, hazard and tedium of such exploration. For remote explorers, power is the most critical resource, and the most common source of that power is solar energy. Information about the robot configuration, the planned path, the terrain and the position of the sun can be processed by a solar power expert software module to calculate the power provided by a given plan of action. Using this information to select the best plans will enable remote robotic explorers to extend their lifetimes. This paper presents the development of a solar power expert and its implementation on a simulator. Several patterned path plans are evaluated with various solar panel configurations, starting times and locations, concentrating on polar regions.

1. EXPLORATION ROBOTS

Capable and adaptable robots are needed for exploring areas too dangerous or costly for humans to visit. Planets, moons, and remote earthly locations such as polar regions and volcanic craters are some such areas. The absence of close human presence in these places requires some degree of autonomy, particularly when the work area is dynamically changing and not fully known ahead of time. Preprogramming the robot is not an option when insufficient information is available. Teleoperation in remote areas causes multiple difficulties: time delays prevent rapid reactions to dangerous situations and lengthen multi-stage tasks unacceptably, while limited sensor information frustrates human interpretations and reactions. Autonomous operation allows robots to reason about their ability to perform various tasks as well as the probable results of performing those tasks, all with respect to the current, possibly changing environment.

To succeed in their mission of exploration, much work needs to be extracted out of few resources. The financial budget for robotic explorers is always limited, creating size and weight restrictions for any robot destined to be launched into space or placed into other remote areas. Thus, efficient use of resources can be of vital importance to robotic explorers, making the difference between success and failure of the mission.

The most critical exploration resource is power, and optimizing its generation is crucial. For this class of remote, outdoor, exploring robots, solar power is the mode of choice [Colozza]. The use of non-renewable batteries or fuel cells alone is not feasible for extended missions, due to the enormous weight and volume that would be needed to transport the required amount. Nuclear power is another option, but one which requires extensive safety reviews, and public and governmental approval which may take years to complete. Despite drawbacks of solar power such as inefficiencies due to material composition, dust storms or clouds, solar power is a prime power source in the inner solar system, ranging from the Sun to Mars.

To generate solar power for use by a mobile robot, several factors need to be considered. Solar power generation depends not only on the static solar panel configuration, but also on the changing orientation of those panels with respect to the sun, and the current visibility and strength of the sunlight reaching the panels. The motion of the sun over time combined with local terrain maps will indicate whether or not the sun will be visible at a given location, and at which angle the sunlight will be incident on the panels. A solar power "expert" can simulate these environmental factors while evaluating a given plan of action, combining the software simulation with the robot configuration to estimate the amount of solar power that will be generated while enacting that plan. This estimate can then be used by the robot to determine which plan to choose and what actions to perform.

2. EXPLORATION MODES

For robotic explorers, extensive travel through a region is a primary task, often accomplished using complete coverage patterns. Such patterns enable the robot to pass over every portion of the area, either physically or with its sensors. Coverage of an area is a common theme in several earthly applications, whether pursued robotically or by humans, such as landmine detection and meteorite searches. Understanding the application aids in determining the best way to cover the area.

The application for which this work was performed is a robotic Antarctic meteorite search. Antarctica is one of the most remote locations on Earth. Its cold and pristine environment makes it one of the best places to find meteorites. Wind scours off the top layers of ice flows blocked by mountains or other obstructions, revealing concentrations of meteorites [Cassidy]. The extreme conditions also make it difficult for humans to work there, but a robot designed to explore this area can provide great scientific returns.

A primary characteristic of this application which affects the mission profile is the polar location. The same consequences discussed here are equally relevant to polar locations on the Moon and other planets. While planetary and lunar surface missions to date have been near-equatorial expeditions, polar regions are increasingly of interest to researchers. For example, searching for frozen volatiles such as water ice on the lunar south pole is currently under investigation [Deans], and the Mars Polar Lander will land near 75° south latitude on Mars later this year.

Solar power generation near the equator is simplified, but in the polar regions, the sun remains low on the horizon, producing long shadows and enhancing the effect of terrain features. Calculating the location and movement of shadows is thus complicated, but necessary to ensure adequate generation of solar power. In some cases, long periods without sunlight may occur, at which time dependence on other power sources or hibernation must occur.



FIG. 1: Vertical Solar Panels on Robot

Due to the low sun angles, solar panels must be more vertical than horizontal to gain the most power. When movable panels are not possible or desirable, choosing the best orientation for the solar panels is vital. One configuration considered for the meteorite searching robot is pair of vertical panels back to back, placed along the spine of the robot (see Figure 1).

3. SOLAR POWER EXPERT

The first step in calculating the amount of power which can be generated by an exploring robot is determining its position throughout a plan of action. A simulator was designed which takes a desired coverage pattern and updates the position of the robot every second as it follows the pattern. Two basic types of coverage patterns have been implemented so far: a straight rows pattern, and a spiral pattern, as shown in Figure 2.

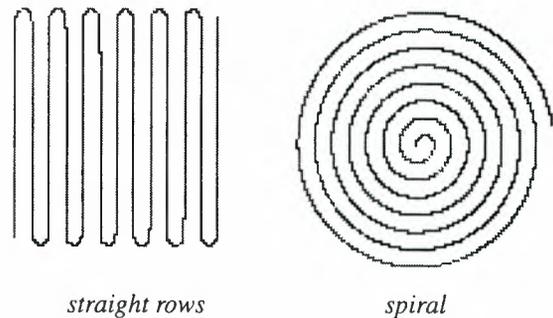


FIG. 2: Coverage Pattern Types

Each pattern is defined by a module which determines the steering angle with which the robot should drive to continue the pattern. By knowing the current location of the robot, such as with differential GPS, and knowing certain parameters of the pattern, such as starting and end points and row lengths, the pattern modules can correct for deviations from the pattern and bring the robot back on track.

The second step is to determine the position of the sun relative to the robot's current location. The current basis of the solar power expert is a function from the SatTrack program [Bester] which calculates the right ascension, declination and distance of the sun and moon from the Earth's center for a given Julian date. The results from this function are transformed to give the altitude and azimuth of the sun as seen from a given location on the Earth's or moon's surface for a given date and time.

Terrain features are then taken into account. By comparing the surrounding terrain elevations in the direction of the sun from the robot, any occlusions of

the sunlight can be determined (see Figure 3). Two levels of shadow finding are proposed, based on high and low resolution terrain maps. A small, high resolution area can be processed quickly to determine if any local obstacles are blocking the light. This terrain map will be generated in realtime, as the robot explores its surroundings with various sensors and adds new obstacles and elevation information to its database. The second level of shadow finding is based on a lower resolution map of the area containing information gained beforehand from remote sensing or digital elevation maps. This type of map can provide information about large areas of shadow which can be used in planning where to start exploring first.

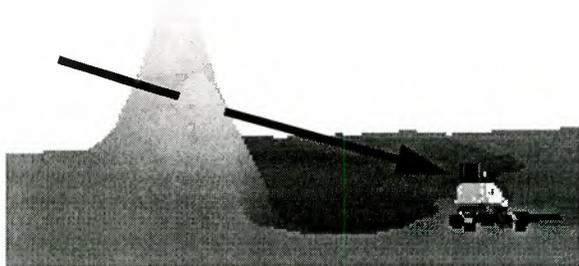


FIG. 3: Determining Terrain Occlusions of Sun

Finally, the robot's pose and solar panel configuration are used to determine the amount of solar power which can be generated. The normals to the robot's solar panels are defined by the configuration. The robot's roll, pitch and yaw are then used to calculate the current direction in which those normals point relative to the angle of sunlight. The power generated varies with the cosine of that relative angle, decreasing as the angle increases.

The power generation calculation is performed at each step of the way along the coverage pattern. Recalculating every second provides an accurate enough evaluation considering the speed of the robot and the speed of the sun's movement across the sky. Both the instantaneous power produced over time and the total energy generated for the pattern can be calculated and used to compare different coverage patterns.

4. SIMULATION RESULTS

Two types of tests were performed with the solar power expert. For the first type of test, the two different coverage patterns, straight rows and spiral, were evaluated for six different latitudes on Earth's surface, at three times of year for each latitude. The

time of day was chosen such that the sun was near its peak elevation at that time. The starting heading of the robot was chosen such that the vertical solar panels were perpendicular to the angle of sunlight, producing the greatest amount of power at that time. In addition to the back-to-back vertical solar panel configuration, an additional configuration of a single solar panel placed horizontally on top of the robot was considered, to provide more information for the lower latitude test cases. All three solar panels were assumed to be 1 meter square in size, producing a maximum power output of 100 Watts.

The strength of sunlight reaching the robot can differ greatly depending on the sun's elevation above the horizon, due to diffraction effects. While this difference was not accounted for in the energy sums, the main values considered were the ratios of total energy for the different patterns and different solar panel configurations. Since both patterns are performed with the same sun angles, for both configurations, the sunlight strength cancels out.

The pattern simulations were based on rows or spirals 8 meters apart, on flat ground. The straight rows pattern used rows 100 meters long, and a total width of 100 meters. At a speed of 0.15 m/s, this pattern requires approximately 2 hours and 40 minutes. The spiral pattern was performed for the same amount of time. Table 1, at the end of this paper, shows the results of the simulations. A longitude of 0 W was used for all simulations. For southern latitudes, January 1st, 2000 was used for summer, October 1, 1999 for spring, and July 1, 2000 for winter. For northern latitudes, January 1, 2000 was used for winter, October 1, 1999 for fall, and July 1, 2000 for summer.

While the purpose of the solar power expert is to evaluate possible plans of action during the mission for the current terrain and area, the evaluations of these test cases show some basic trends which might lead to heuristics for simplifying the online planning, as well as aid in determining the best solar panel configuration to use for the designated location. For example, the side solar panels are preferable for higher latitudes and during seasons with lower sun angles, as expected. For intermediate locations, simulations can quantitatively predict which configuration is best.

For the side solar panel configuration, the differences in power generation between the two patterns are greater during the winter than the summer, for all latitudes. Generally, the differences are also greater for

higher latitudes, during the same season. The commonality between these trends is lower sun angles, implying that power generation is more sensitive to the type of pattern when the sun angle is low. For the top solar panel configuration, both patterns produce the same power, as the robot's orientation is irrelevant.

In all cases except the summer at 40 N latitude, the straight rows pattern generates more power than the spiral pattern. Of course, this depends on the initial heading of the straight row with respect to the sun. When evaluating plans for a specific time and location, the solar power expert can calculate just how much better a given coverage pattern is and in which direction to start.

The second type of test concerns the effect of terrain on sun visibility for three different latitudes on Earth. For the terrain map, a digital elevation map of the moon's Tycho crater [Margot] was modified by reducing the vertical scale by a factor of 100, reducing the horizontal scale by a factor of 1000, and "placing" the crater region at several locations on Earth, to generate graphs which clearly demonstrate the effect of latitude on terrain shadowing. The straight rows coverage pattern used is superimposed on the crater region in Figure 4. The pattern is 100 meters by 80 meters, with rows 8 meters apart. The simulated robot traveled at 0.15 m/s.

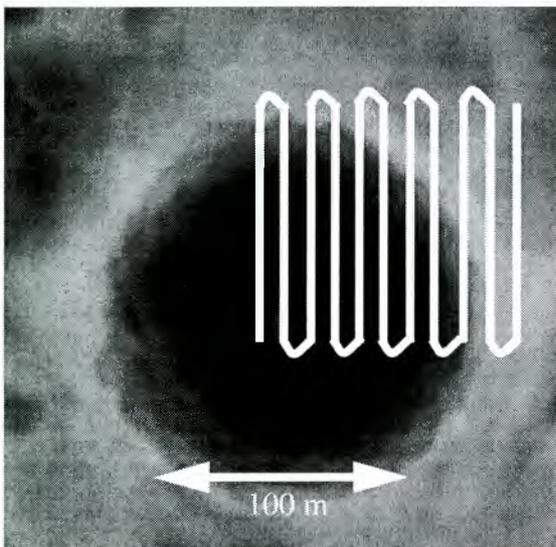


FIG. 4: Coverage Pattern over Crater

Traversability of the region and the effect of slopes on robot locomotion and pose were not considered, as the purpose was to gain an understanding of the shadowing caused at different latitudes by the terrain

features. The power generated by two back to back vertical solar panels and one top horizontal solar panel, each of size 1 meter square with a maximum output of 100 Watts, is calculated for latitudes of 85 S, 45 S, and 0, on December 1, 1999.

The Tycho crater is deep and abrupt, even with the scale reduced, as can be seen by the plot of vertical elevation versus time in Figure 5. This causes the robot to generate power sporadically, as it passes in and out of shadows. Even with the relatively high sun angles at the equator, as high as 68 degrees at this date, shadows still cover portions of the crater. The sun for this simulation was coming from the lower left quadrant of the image, causing the central peak of the crater to shadow the robot as it passes to the upper right of the peak on the second and third rows of the pattern. The rim of the crater causes more shadows during the final rows.

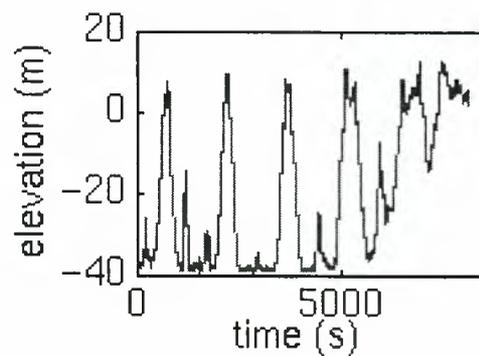


FIG. 5: Elevation of Robot during Pattern

Figure 6 compares the power generated at the three different latitudes on Earth. The x axis shows the time passed as the robot progresses through the coverage pattern, while the y axis shows the instantaneous power being produced. For lunar locations, with the moon's smaller curvature, distant terrain features will not occlude the sun as much. The lower sun angles at polar locations will counteract that effect, however, causing longer shadows.

5. ADDITIONAL CONSIDERATIONS

In the above simulations, terrain effects on robot locomotion and pose were ignored. In reality, a robot will not be able to drive straight up a steep crater wall, and any slopes will affect the relative angle between the sun and the robot's solar panels. Incorporating slopes into the simulation and calculating their effect on the robot's roll and pitch is the next step.

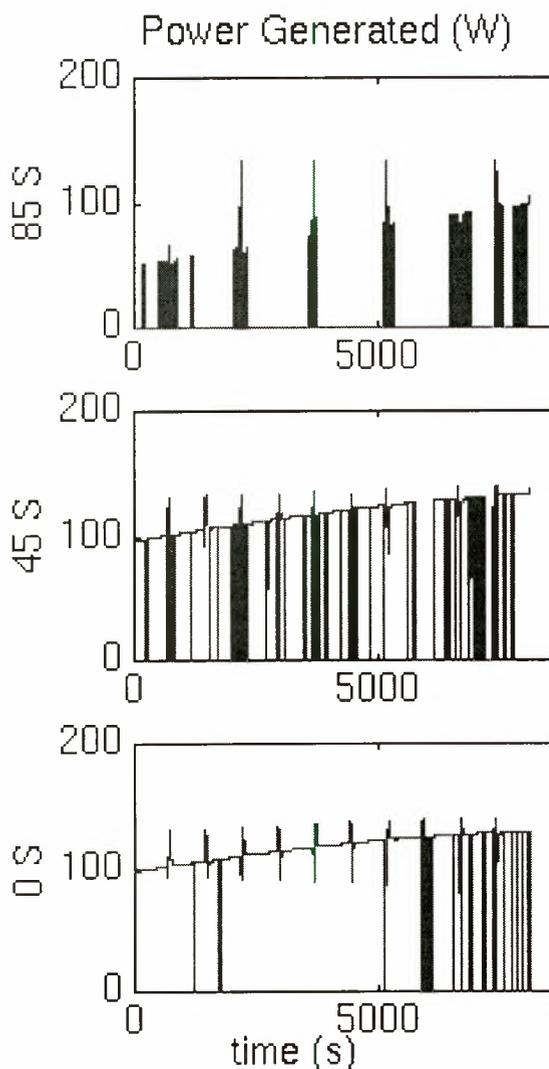


FIG. 6: Solar Power Generated within Crater

In addition, for uneven ground, uncertainties in the terrain will cause uncertainties in the power evaluations. Previously unknown obstacles will cause path deviations, and smaller rocks and hills will temporarily affect the orientation of the robot. Strategies for accounting for uncertainty include considering a range of possible locations and orientations of the robot, or taking actual field data and comparing it to predicted data, converting the observed variations into uncertainty.

Another consideration is the variation in the robot's power consumption. Not only does the amount of power generated need to be calculated, but also the amount of power consumed to ensure that enough power is available to accomplish the plan of action. The type of steering activity, the terrain to be covered,

and environmental affects such as wind will all affect the power requirements of the system.

Different coverage patterns require different amounts of steering changes, thus requiring more or less power. Tests with the robot used in the Antarctic meteorite search show that nearly 4 times the amount of power is needed for a skid steered point turn than a 12 meter radius skid turn [Shamah]. Uneven or soft terrain may cause more slippage and power draw as well. Finally, the configuration of the robot may affect the amount of power consumed. Depending on the wind direction relative to the robot's heading, a vertical solar panel sail as described for the simulations will cause varying amounts of air resistance, and therefore variations in power consumption. However, for locations on the moon, and possibly Mars, wind will be irrelevant or negligible.

The two pattern types shown here are only a small sample of the plans that can be evaluated by the solar power expert. Patterns with intermediate amounts of curvature between the straight and spiral patterns can be developed. One such pattern is a polar sun-following pattern, where the robot turns continually to maintain the optimum orientation of its solar panels to the sun. Other patterns may be based on the terrain, such as boundary-following patterns. Creating a wider repertoire of patterns will be pursued in the future.

The simulations above demonstrate the capability for an on-board solar power expert, allowing the robot to make decisions based on the current environment. This software has, in fact, been implemented on the robot used in the Robotic Antarctic Meteorite Search project, which will be deployed for a second time in Antarctica in the 1999-2000 season [Shillcutt]. Solar power evaluations will be produced and compared to actual solar power generation based on observed sun visibility and test solar panels.

By evaluating multiple coverage pattern options with respect to power considerations, explorer robots can select the plan which allows the best chance of surviving. With the limited resources typical of space missions, even minor improvements in power usage can determine whether or not a mission is successful. Using this solar power expert's information to select the best plans will enable remote robotic explorers to extend their lifetimes and produce greater returns.

ACKNOWLEDGEMENTS

This research was performed for, and with the assistance of the members of the Robotic Antarctic Meteorite Search group at the Field Robotics Center of Carnegie Mellon University, which is led by Dimi Apostolopolous. The Tycho crater digital elevation map used in the simulations was produced by Dr. Jean-Luc Margot, of Arecibo Observatory in Puerto Rico.

BIBLIOGRAPHY

Bester, M., *SatTrack* (computer program), <http://www.bester.com/sattrack.htm>

Cassidy, W. A., R. P. Harvey, J. W. Schutt, G. Delisle and K. Yanai, "The Meteorite Collection Sites of Antarctica," *Meteoritics*, Vol. 27, 1992, pp. 490-525.

Colozza, A. J., *Design and Optimization of a Self-Deploying Single Axis Tracking PV Array*, NASA CR 189132, 1992.

Deans, M., S. Moorehead, B. Shamah, K. Shillcutt, and W. Whittaker, "A Concept for Robotic Lunar South Pole Exploration," *Sixth International Conf. on Engineering, Construction, and Operations in Space*, Albuquerque, NM, April 26-30, 1998, pp.333-339.

Margot, J. L., D. B. Campbell, R. F. Jurgens, M. A. Slade, "The Topography of Tycho Crater", *Journal of Geophysical Research*, in press, 1999.

Shamah, B., *Experimental Comparison of Skid Steering Vs. Explicit Steering for a Wheeled Mobile Robot*, Master's Thesis, CMU-RI-TR-99-06, Carnegie Mellon University, March 1999.

Shillcutt K., D. Apostolopoulos, and W. Whittaker, "Patterned Search Planning and Testing for the Robotic Antarctic Meteorite Search," *International Topical Meeting on Robotics and Remote Systems for the Nuclear Industry*, American Nuclear Society, Pittsburgh, PA, April 25-29, 1999.

TABLE 1: Simulation Results

Latitude	Season	Sun Elevation During Pattern (degrees)	Energy Generated Straight Rows, Side Panels	Ratio of Side / Top Panels (Straight : Spiral)	Ratio of Straight / Spiral Patterns
89 S	Summer	23.8 - 24.0	784,534 J	2.009 : 1.415	1.420
	Spring	3.9 - 4.1	857,086 J	12.624 : 8.884	1.421
	Winter	-	-	(sun not above horizon)	-
80 S	Summer	32 - 33	755,820 J	1.452 : 0.975	1.490
	Spring	12 - 13	880,349 J	4.152 : 2.774	1.497
	Winter	-	-	(sun not above horizon)	-
70 S	Summer	38 - 43	619,053 J	0.977 : 0.718	1.359
	Spring	18 - 23	791,702 J	2.283 : 1.626	1.404
	Winter	-	-	(sun not above horizon)	-
60 N	Summer	44 - 53	502,388 J	0.682 : 0.543	1.258
	Fall	19 - 27	733,628 J	1.887 : 1.449	1.302
	Winter	1 - 7	866,990 J	10.364 : 7.300	1.420
50 N	Summer	49 - 63	370,243 J	0.454 : 0.403	1.126
	Fall	26 - 37	700,281 J	1.352 : 1.002	1.349
	Winter	9 - 17	839,614 J	3.524 : 2.493	1.414
40 N	Summer	53 - 73	255,119 J	0.259 : 0.298	0.867
	Fall	32 - 47	605,869 J	0.959 : 0.736	1.303
	Winter	17 - 27	786,878 J	2.041 : 1.457	1.400

ROAMS : Rover Analysis, Modeling and Simulation

Jeng Yen, Abhinandan Jain and J. Balaram

Jet Propulsion Laboratory

California Institute of Technology

4800 Oak Grove Drive, Pasadena, CA 91109

phone: 01 818 354-4744, fax: 01 818 393-5007, e-mail: Jeng.Yen@jpl.nasa.gov

Abstract

In this paper, we present the development of ROAMS system for real-time simulation of mobile robotic vehicles. The purpose for the simulation is to provide a virtual testing ground for various subsystems and components of the robotic vehicle, which includes a mechanical subsystem, an electrical subsystem, internal and external sensors, and on-board control software. Using the DARTS/DSHELL framework, the real-time simulator can be applied to both operator-in-the-loop and off-line simulation. This flexibility permits ROAMS to be utilized for various rover tasks in planetary exploration missions, including those of system engineering, scientific research, and operation teams. However, to achieve real-time in the simulation of complex physical systems is non-trivial. Efforts have been made to build the rover model for an efficient and stable simulator. Currently, the rover model is comprised of its mechanical, electrical, and sensor subsystems, all connected with the on-board software. With additional terrain and rock models, we developed a novel solution technique that leads to real-time simulation of the rover traversing Mars-like terrain.

1 Introduction

ROAMS is constructed upon JPL's DARTS/DSHELL [4], a multi-mission spacecraft simulation software, for simulation of robotic vehicles. Inherent to DSHELL is the development environment for modeling sensors, actuators, electrical and mechanical subsystems. Expanding these capabilities, ROAMS models closed-loop mechanisms and contacts between the vehicle's wheels and the terrain. On a class of Mars-like terrain, it can efficiently solve the configuration of the rover traversing the terrain and rocks. The results are used for

feeding the sensor data back to the on-board subsystems and other devices. ROAMS is developed for real-time simulation of the rover system. For the development purposes, we used Rocky-7, a Mars rover prototype, as the base model of the robotic vehicle [6].

The Rocky-7 Research Platform

The Rocky-7 rover configuration is shown in Figure 1. Like Sojourner rover in the recent Pathfinder Mass mission, it is designed for carrying out various tasks in planetary exploration. The mobility system is a modification of the Rocker-Bogey design used in previous rovers at JPL. It consists of two rockers hinged to the sides of the main body. Each rocker has a steerable wheel at one end and a smaller bogey at the other end. Unlike its predecessors Rocky-3 and Rocky-4 (and the Sojourner flight rover) that have four steerable wheels, Rocky-7 has only two. Rocky-7 has a closed-loop mechanism (rocker-differential) designed to give it high mobility in rough terrain. The kinematic model is presented in Figure 2, where the internal constraints are described in details.

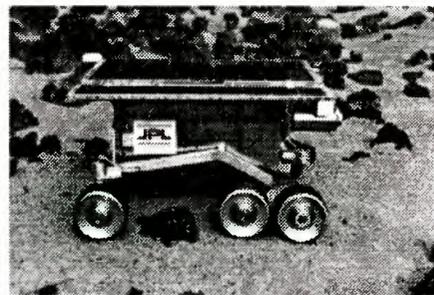


Figure 1: Rocky-7 Side View

Novel Solution Technique

One of the essential component of the rover simulation is the terrain, where the position of the wheels deter-

mines rover's configuration. We developed a novel numerical solution for the configuration kinematics of the rover that arise from driving on the Mars-like terrain. The configuration kinematics of the rover is posted as a constrained optimization.

The objective function is composed of the driving function and the loop-closure equations. The constraints are resultant from the contacts between the six wheels and the terrain. Depending on the contact rule, e.g., non-penetrative or penetrative, and the terrain profile, e.g., smooth or non-smooth, the constraint equations can be problematic for a Mars-like environment. For instance, all six wheels may not be in contact with the ground at all times, because of physical limitations, such as joint stops of the rocker-differential-bogey mechanism. This problem is resolved by using a bound-constraint method [7], which fixes the joint angles at their limits. In particular, the active contact condition is maintained if the Newton iteration converges. If the Newton iteration diverges, we check the joint limits and estimate the condition number of the iteration matrix to decide either to accept or to reject the current solution. Based on the progress of the iteration, the set of active constraints can be renewed. This approach is very effective for solving the inverse kinematics of the rover.

The terrain is represented by a parametric surface with a rock field, where the rock is hemispherical. For both the smooth (e.g. with continuous second order derivatives) and non-smooth terrain, the above numerical solution is applied with slight modifications. On a smooth terrain, the solution is carried out using a Newton-type method, which is globalized by a backtracking line-search at each iteration. For non-smooth terrain, we apply an inexact Newton iteration [2] that uses an approximated iteration matrix to obtain the solution. This class of Newton-type methods can produce robust results. It is efficient enough to apply to a real-time simulation of the rover on a workstation computer platform. Detailed solution technique is contained in Section 3.

Applications of ROAMS

Building on this novel solution technique, we have applied ROAMS for a closed-loop simulation with the Rocky-7 on-board operating system. The navigation state machine of the Rocky-7 rover was used in driving the model against a terrain with randomly distributed rocks. Using DARTS/DSHELL modeling libraries, we implemented the models for some of the hardware, including a panoramic spectrometer, a sun

sensor, a tilt sensor, an obstacle detection camera, a solar panel, and a battery. These models provide high-fidelity synthetic data to stimulate the on-board subsystems owing to an efficient and stable numerical solution. This tool permits a much improved environment for the research and development of the rover system.

In a recent project, ROAMS has also been integrated with a planning system that directs multiple rovers to perform some complex tasks. Three rovers are managed by the planner in a science mission. The extension of ROAMS is to integrate the high-fidelity rover models with the Rocky-7 software and the planning system. For this purpose, we developed additional device models for avoiding collision among the rovers, detecting obstacles, and monitoring power level. These device models provide high quality sample data for the planning model. The integrated system has great potentials for many advanced applications in areas of design, engineering and planning of mobile robotic systems. More detailed descriptions of this integrated architecture are contained in [5].

2 Rover Model Development

The mechanical model of the Rocky-7 rover consists of 12 bodies, including a chassis, two pairs of rockers and bogeys, a differential and six wheels.

Coordinate Frames and Variables

Coordinate frames and variables are as defined in Figure 2. The unconstrained rover's degrees-of-freedom (dof's) are seen to be three translational, three rotational, three internal ($\gamma_0, \gamma_1, \gamma_2$), two steering (λ_1, λ_2), and six drive (ψ_1, \dots, ψ_6). Contact interactions at each wheel constrain these dof's to result in the rover typically having two translational dof's (x, y) and one angular dof (heading) when in full contact with the ground. Notice the closed kinematics chain consisting of the main differential, the spherical pinned joints on the near side of rover, the main rocker axis, the spherical pinned joints on the far side of the rover, and back to the differential. The constraints for the closed-loop mechanism are given by fixing the distance between the attachment points (center of spherical joints) of two plastic clips (clevis) connecting differential and the rockers. In order to derive the forward kinematics it is necessary to solve for the rocker axis angle in terms of the differential angle. An analytic solution is possible and is graphed in Figure 3, where the

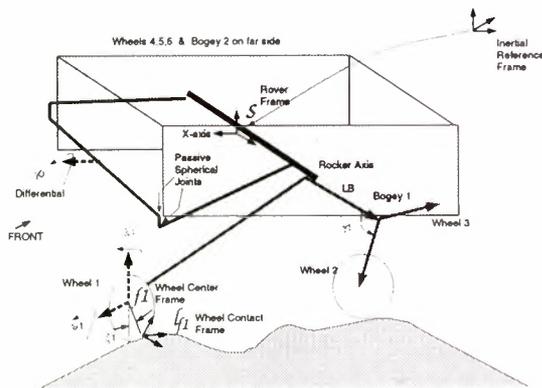


Figure 2: Rocky-7 Kinematics

nominal position of the rocker axis angle on each side of the rover is taken to be zero. Notice that at large angles for the differential, the solutions are no longer symmetric. However, for small differential angles a symmetric linear approximation is possible and used henceforth.

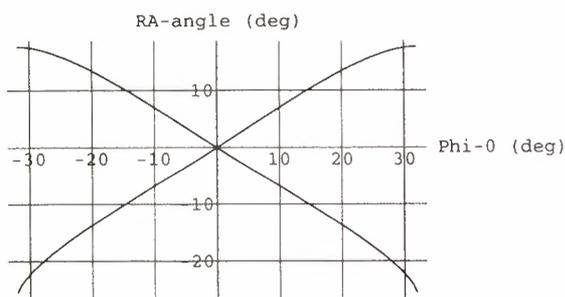


Figure 3: Solution of Internal Closed Kinematic Chain

DSHELL models

On-board Rocky-7, sensors, actuators, cameras, and electrical subsystems are modeled with the standard DSHELL library [4]. The DSHELL models, navigation sensors, and custom electronics constitute a testbed for Rocky-7's system software.

The sun sensor, accelerometer, and wheel encoders are used to provide the internal odometry of the rover. The sun sensor and an obstacle detection CCD camera provide information for the autonomous path generation. The commands from the on-board software drive six wheel-motors and two steering motors. Extracting from the commands, ROAMS computes the incremental change of the rover's position, and feeds back the wheel encoder and other sensor data. By completing the loop for Rocky-7's autonomous path motion, ROAMS furnishes an effective testbed for the

on-board navigation algorithm and software.

Rocky-7 has camera pairs with a 5 cm baseline at both ends of the vehicle, enabling bi-directional driving with obstacle avoidance. The camera model is implemented in ROAMS using a search of the synthetic rock field via bounding-box methods. Based on the height of the rocks in the approximated camera range, the signal of obstacle detection is sent to the rover navigation system. For multiple rovers, the obstacles include other rovers. When an obstacle is detected, the signal will trigger the vehicle navigator to generate an alternative path to the target location.

The power unit of the rover consists of a battery model, a power draining model, and a solar panel model. Using the nominal power usage of the motors, electrical subsystems, an estimated power level of the battery is obtained. The solar panel model computes, using the panel's attitude and the sun position, readings of the solar power generation.

Synthetic Terrain and Rocks Models

Two types of terrain, *smooth* and *piece-wise smooth*, are currently available in ROAMS. It is in the form of a parametric surface, where the elevation (z -coordinate) is a function of the x - and y -coordinates. On the terrain, a rock field can be applied to make up a synthetic Martian environment. The rock profile is represented by a half-sphere

$$z = \begin{cases} \sqrt{r-d}, & d \leq r \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

where $d = \sqrt{(x-x_c)^2 + (y-y_c)^2}$, and (x_c, y_c) is the center location, and r is the radius of the rock. When the contact location is within the rocks radius, equation (2.1) is the additional elevation of the wheel contact point. Note that the derivatives become increasingly degenerated when approaching the rock's edge. Care must be taken when the wheel is moving in and out of the rock's circumference. This is dealt with a line-search strategy in the Newton iteration.

3 Configuration Kinematics

It is straightforward to formulate the inverse kinematics of the rover traversing a terrain in the optimization problem:

$$\min_{q \in R^n} f(q) \quad (3.1a)$$

$$\text{subject to } g(q) \geq 0 \quad g \in R^k \quad (3.1b)$$

some wheel(s) may not be in contact with the terrain. Moreover, on a rough terrain, multiple contacts may occur that yields additional difficulties. These and related computational problems are dealt with using a nominal contact point and a modified line-search mentioned above. Choosing the contact point on the wheel to be $x_w = x_c$ and $y_w = y_c$, we simplified (3.3a) to the form:

$$z_c - z(x_w, y_w) - r_W = 0 \quad (3.4)$$

where $z(x_w, y_w)$ is the elevation of the contact location on the terrain, and r_W is the wheel radius. Using equation (3.4) for the contacts, the modified Newton iteration exhibited an enhanced convergence. For non-smooth terrain, the solution of equation (3.4) may become discontinuous, which may cause divergence in Newton iteration. Nevertheless, the solution method is effective on piece-wise smooth terrain and flat terrain with a rock field. We are developing new direct search methods for traversing rough terrain.

As noted before, equation (2.1) induces a singularity to the Jacobian of (3.4) when the contact location is at its circumference. To treat this problem, we apply a regularization of the derivatives of equation (2.1)

$$\frac{dz}{dx} = \begin{cases} -\frac{(x-x_c)}{\rho} & \text{if } d \leq r - \epsilon \\ -\frac{(x-x_c)}{r\epsilon}, & \text{otherwise.} \end{cases} \quad (3.5a)$$

$$\frac{dz}{dy} = \begin{cases} -\frac{(y-y_c)}{\rho} & \text{if } d \leq r - \epsilon \\ -\frac{(y-y_c)}{r\epsilon}, & \text{otherwise.} \end{cases} \quad (3.5b)$$

where $d = \sqrt{(x-x_c)^2 + (y-y_c)^2}$, $\rho = d\sqrt{r-d}$ and ϵ is a small number.

4 Applications of ROAMS

Testing Tool for Rocky-7 On-board Software

One direct application is to use ROAMS for testing the Rocky-7 on-board software. For development purposes, the system software is ported from a real-time operating system to the Unix platform. The system clock is set to use the standard Unix system time, permitting the (simulated) sensor feedback being synchronous.

Upon the rover receiving the command to move to a desired location, the navigation algorithm generates a sequence of *way-points*, turns the rover toward the goal, and executes obstacle avoidance activities. Connected to ROAMS, the navigation state

machine is tested against the synthetic terrain made of a flat base with the Viking Lander 1 rock field. The navigator of Rocky-7 produces the commands for its four wheel-motors, moving toward the next way-point [11]. ROAMS applies the commands to approximate the next position of the rover, then solves the inverse kinematics for the configuration of the rover. Using the position and attitude, the sensor outputs are obtained and sent back to the rover software system. A simple diagram illustrates the configuration of the testbed in Figure 4. The testbed results have given qualitative measurements to the robustness of the state machines of the software system.

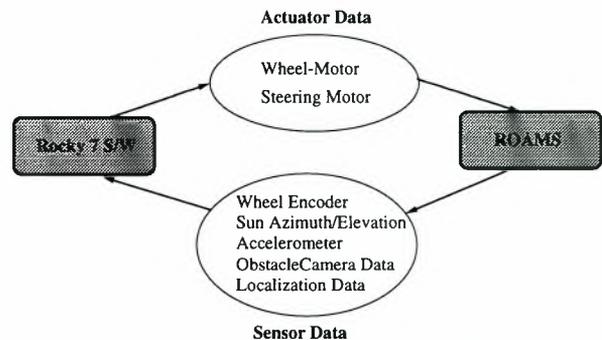


Figure 4: Rocky-7 Testbed Data Flow

Virtual Environment for Cooperative Rovers

ROAMS is also used for an integrated architecture being developed at the NASA Jet Propulsion Laboratory for utilizing multiple cooperating rovers. It provides a virtual testing ground for planning and coordinating multiple rovers in performing a complex task for planetary surface exploration. For the multiple rover architecture, we extended the single rover simulation to support several rovers, and developed additional hardware models, which approximate the resources of each rover. In the current application of cooperating rovers, we evaluated the architecture using a model of terrain and a set of science goals over that model. Correspondingly, a panoramic spectrometer model was developed to produce spectral measurements. Utilizing the development of Rocky-7 testbed, we constructed additional output data to the planning and scheduling system. As shown in Figure 5, the rover model contains the power unit, including a solar panel, a battery, and a power draining model. The solar panel model generates a voltage computed by the relative angle between sun position and the panel's surface. Power draining is based on the nominal power usage of electrical components. These and the battery model can give a high-fidelity prediction of the power level, allowing robust planning activities.

where $f(q)$ represents the $L2$ -norm of the internal kinematic equations and the driving functions, and $g(q)$ consists of the joint limits and the wheel-terrain contact equations. For example, in our case, $f(q)$ is comprised of the squared $L2$ -norm of the internal constraints and the path constraints, i.e., rover's body-coordinates (x, y) and its heading angle following the path. The constraints in equation (3.1b) describe the distance function between terrain and six wheel. Four joint limits of the rockers and bogeys are implemented without additional equations.

Newton-Type Iterations

It is well-known that nonlinearity of the constraints (3.1b) can considerably increase the difficulties in the solution of (3.1) [7]. Furthermore, the objective of a real-time simulation requires an highly efficient solution. To achieve the required efficiency, we have applied the class of *first derivative methods* to (3.1), and solve the resultant nonlinear system via Newton-type methods. The gradient of (3.1a) and the constraints of (3.1b) yield the system of nonlinear equations of the form

$$F(q) = \begin{bmatrix} \nabla f(q) \\ g(q) \end{bmatrix} \quad (3.2)$$

where ∇f is the gradient of $f(q)$, and the components of $g(q)$ are of the form $w_i g_i(q)$ for some $i \in 1, 2, \dots, 6$ and $w_i \geq 0$. The variable q is made of the generalized coordinates of the rover model in the computation engine DARTS [1]. In particular, the function $F(q)$ consists of three kinematic drivers, two internal constraints, and six to eighteen equations in $g(q)$. The contact condition, e.g., non-penetrative or penetrative, determines the number of $g(q)$. Nominal contact of the wheels yields 6 constraints in $g(q)$. For non-penetrative contact, $g(q)$ contains 18 equations and each wheel adds two more parameters to q for the wheel to terrain contact model. At each wheel, the contact is occurred with a positive *weight factor* w_i . During the iteration, the weight factor w_i may approach zero to relax the non-penetrative contact between the wheel and the ground. At the event that any wheel leaves the ground, its weight factor can be set to zero. The re-scaling of w_i is coupled with a global search at the each Newton step. Another feature is the limitation of joints. This is implemented by fixing the corresponding joint variable in q . The resultant sub-problem induces the search direction that does not move these joint variables. These modifications furnish an effective numerical solution of (3.2) for the configuration kinematics of the rover, and permit real-time simulation of the rover.

Properties of Line-Search Algorithm

For a robust Newton convergence, we implemented a step selection strategy using backtracking, see [3], pp. 120-126. This algorithm ties to a Newton step at the each iteration. For a smooth nonlinear system (3.2), the sequence of solution generated by the iteration will converge very fast, i.e., quadratically or superlinearly, to a local minima of equation (3.1). For the rover inverse kinematics, the application of backtracking line-search algorithm is particularly effective. As explained, the nonlinear system (3.2) is of a specific structure such that the nonlinear contact equations are loosely coupled with the mostly linear driving functions. This implies that the contact equations should be the most difficult part to resolve. For this reason, the maximum step is set to the corresponding arclength of the wheel's surface at one time unit.

Another feature associated with the step selection is the detection of an ill-conditioned iteration matrix. During the simulation, the Newton direction may become irregular when the rover steps through a non-smooth region. For instance, at the edge of a rock on a flat terrain, or at the boundary of a piece-wise smooth patch, the Jacobian of equation (3.2) may become near ill-conditioned. The resultant search direction will therefore be irregular (often contains very large components). Using a new search direction, the iterative solution may overcome the local irregularity. In the preliminary test, it often reaches a nearby solution that is good enough for the application.

Handling Terrain and Rocks

The terrain profile is a *parametric surface*. Each point on the terrain can be written as its Cartesian coordinate $[x, y, z(x, y)]^T$. On the i th road wheel, a non-penetrative contact yields

$$g_i^c = (x_w - x_c)^2 + (y_w - y_c)^2 + (z_w - z_c)^2 \quad (3.3a)$$

$$g_i^n = \begin{bmatrix} n_w t_c \\ n_w t_c \end{bmatrix} \quad (3.3b)$$

where $[x_w, y_w, z_w]^T$ is the position on the wheel and the contact position on the terrain is $[x_c, y_c, z_c]^T$, and t_w, t_c are tangent and n_w, n_c are normal vectors of the wheel and the terrain at the contact position, respectively. Equation (3.3a) is the contact condition, while equation (3.3b) is the *non-penetrative* condition that constrains two tangent planes at the contact points to be co-linear. For Mars-like terrain, equation (3.3b) is often eliminated because of non-smoothness in the terrain and rock field.

Equation (3.3a) is in fact an *inequality* because

It is shown through the development of this integrated architecture that ROAMS is a flexible and effective tool for modeling and testing of robotic vehicles.

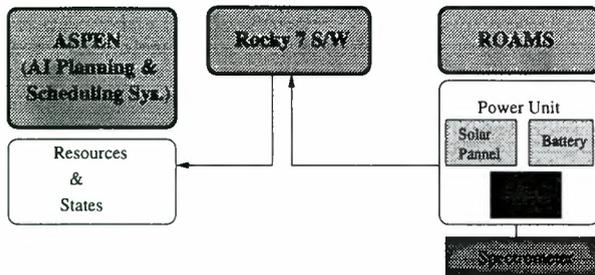


Figure 5: Cooperating Rovers Testbed Data Flow

Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] A. Jain and G. Man, *Real-time simulation of the Cassini spacecraft using DARTS: functional capabilities and the spatial algebra algorithm*, in 5th Annual Conference on Aerospace Computational Control, (Jet Propulsion Laboratory, Pasadena, CA.), Aug. 1992.
- [2] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp.400-408.
- [3] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.
- [4] J. Biesiadecki, D. Henriquez and A. Jain, *A Reusable, Real-Time Spacecraft Dynamics Simulator*, in 6th Digital Avionics Systems Conference, (Irvine, CA), Oct 1997.
- [5] T. Estlin, H. Das, S. Hayati, et al., *An integrated architecture for cooperating rovers*, to appear in proceedings of the symposium of iSAIRAS '99.
- [6] S. Hayati, R. Volpe, et al., *The Rocky-7 Rover: A Mars Sciencecraft Prototype*, Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque NM, April 20-25 1997.
- [7] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press Inc. (London) Ltd., 1981.
- [8] D.J. Montana, *The kinematics of contact and grasp*, The International Journal of Robotics Research, Vol. 7, No. 3, June 1988.
- [9] R.M. Murray, Z. Li and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.
- [10] R. Volpe, J. Balaram, T. Ohm and R. Ivlev, *Rocky-7: a next generation Mars rover prototype*, Advanced Robotics, Vol. 11, No. 4, pp. 341-358 (1997).
- [11] R. Volpe, *Navigation Results from Desert Field Tests of the Rocky-7 Mars Rover Prototype*, to appear in International Journal of Robotics Research, Special Issue on Field and Service Robots, 1999.

AN INTEGRATED ARCHITECTURE FOR COOPERATING ROVERS

T. Estlin, J. Yen, R. Petras, D. Mutz, R. Castaño, G. Rabideau,
R. Steele, A. Jain, S. Chien, E. Mjolsness, A. Gray, T. Mann, S. Hayati, H. Das

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
phone: +1 818 393-5375, fax: +1 818 393-5244, email: {firstname.lastname}@jpl.nasa.gov

ABSTRACT

This paper presents a rover execution architecture for controlling multiple, cooperating rovers. The overall goal of this architecture is to coordinate multiple rovers in performing complex tasks for planetary science. This architecture integrates a number of systems and research efforts on single rovers and extends them for multiple rover operations. Techniques from a number of different fields are utilized, including AI planning and scheduling, real-time systems and simulation, terrain modeling, and AI machine learning. In this paper, we discuss each architecture component, describe how components interact and present the geological scenario we are using to evaluate the overall architecture.

1. INTRODUCTION

This paper describes an integrated architecture being developed at the NASA Jet Propulsion Laboratory for solving planetary surface exploration problems through the utilization of multiple cooperating rovers. Utilizing multiple rovers for science and exploration activities has a number of advantages. First, we can greatly increase mission science return by simultaneously using complementary instruments on different rovers and efficiently dividing science-gathering tasks between the rovers. Second, multiple rovers can perform tasks that otherwise would not be possible using a single rover, such as taking wide baseline stereo images. Third, multiple rovers would enhance mission success through increased system redundancy. If one rover fails, then its tasks could be quickly taken over by another rover, helping to ensure mission success.

This paper presents work in demonstrating how multiple rovers as compared to a single rover can more effectively explore a selected site and return more science data per communication cycle. The described architecture utilizes research results on single rovers (i.e. command sequence generation, navigation, control, science operations, ground

control, etc.) and extends them to multiple rovers. An integrated system architecture has been developed that can automatically generate interesting science goals, plan for and coordinate multiple rover activities, and monitor and update activities in response to anomalous events. This architecture also utilizes a multi-rover simulation environment and control software from the NASA JPL Rocky 7 rover [Volpe et al., 1997]. Techniques from several different fields are combined including Artificial Intelligence (AI) planning and scheduling, real-time systems and simulation, terrain modeling and system kinematics/ dynamics, and AI machine learning.

The organization of this architecture consists of the following. An AI planning and scheduling system (CASPER) takes as input a set of science goals for exploring a particular terrain and then automatically generates plans (i.e. command sequences) that coordinate a team of rovers in successfully completing the goals and exploring the requested areas. Each rover plan is then relayed to the onboard control software and executed in a multi-rover simulation environment (ROAMS) that is used to simulate the rover terrain and rover operations within that environment. The simulator also generates sensor feedback from the rovers which is relayed back to the planner. This feedback is utilized to determine the success or failure of certain activities and any changes in resources or states. If unexpected changes have occurred, the planning system can perform re-planning to fix the original plan and ensure the successful achievement of the goals. An AI clustering algorithm analyzes any science data gathered by the rovers and then uses this analysis to produce new science goals for the rovers to accomplish. This architecture is currently being evaluated using a geological scenario where rovers are used to examine and classify terrain rocks.

The remainder of this paper is organized as follows. We begin by characterizing the multiple cooperating rovers application domain and describing the particular science scenario we are using to evaluate our integrated system.

Next, we present out multi-rover execution architecture which controls and coordinates operations for a team of rovers. We then describe each of its individual components and any interactions between them. In the final sections, we discuss related work, planned future work, and present our conclusions.

2. COOPERATING ROVERS FOR SCIENCE

Utilizing multiple rovers on planetary science missions has several important advantages:

- *Force multiplication.* Multiple rovers can collect more data than a single rover and can perform certain types of tasks more quickly than a single rover, such as: performing a geological survey of a region or deploying a network of seismographic instruments. We call these *cooperative* tasks.
- *Simultaneous presence.* Multiple rovers can perform tasks that are impossible for a single rover. We call these *coordinated* tasks. Certain types of instruments, such as interferometers, require simultaneous presence at different locations. Rovers landed at different locations can cover areas with impassable boundaries. Using communication relays, a line of rovers can reach longer distances without loss of contact. More complicated coordinated tasks can also be accomplished, such as those involved in hardware construction or repair.
- *System redundancy.* Multiple rovers can be used to enhance mission success through increased system redundancy. Several rovers with the same capability may have higher acceptable risk levels, allowing one rover, for example, to venture farther despite the possibility of not returning. Also, because designing a single rover to survive a harsh environment for long periods of time can be difficult, using multiple rovers may enable missions that a single rover could not survive long enough to accomplish.

In all cases, the rovers can behave in a cooperative or even coordinated fashion, accepting goals for the team, performing group tasks and sharing acquired information.

Coordinating distributed rovers for a mission to Mars or other planet introduces some interesting new challenges for the supporting technology. Issues arise concerning interfaces, communication, control and individual onboard capabilities. For example, different software components must successfully interface onboard the rovers to provide the needed autonomous functionality. In addition, mission designers will need to decide on interfaces among the rovers, to the lander and/or orbiter and to the ground operations teams. Decisions will need to be made on communication capabilities, which will limit the amount of information shared between rovers and the lander/orbiter. A distributed control protocol will need to be selected that defines how tasks are distributed

among rovers and the "chain of command" for the rovers. Finally, the onboard capabilities will need to be considered, including computing power and onboard data-storage capacity.

Many of these design decisions are related, and all of them have an impact on the onboard technologies that can be utilized by the mission. The interfaces determine what activities can be planned for each rover and what data or sensor feedback can be utilized by the onboard software. The amount of communication available will determine how much science or terrain data can be shared among rovers and will affect how much each rover can coordinate with other rovers to perform tasks. In addition, communication capabilities will affect the amount of onboard autonomy required. If bandwidth is low and reaction time is critical, a rover will need to react intelligently to the environment, including performing autonomous navigation and replanning for its own activities in response to unexpected events. The control scheme will determine which rover executes which activities and which rovers coordinate and monitor activities of the others. Decisions on the onboard capabilities of each rover limit the independence of the rover. With little computing power, a rover may only be able to execute commands. More power may allow it to plan command sequences, replan if necessary, and analyze gathered data. Some rovers may also perform these activities as a service to other rovers or in cooperation with them.

To evaluate the architecture presented in this paper, we have initially chosen the configuration of a team of three rovers where each rover has a planning and data-analysis tool onboard as well as low-level control software for tasks such as navigation and vision. Each rover can thus plan for its assigned goals, execute and monitor generated commands, collect the required data, perform re-planning if necessary, and perform science analysis onboard to direct its future goals.

Currently we are evaluating our framework by testing its ability to build a model of the distribution of surrounding terrain rocks, classified according to composition as measured by a boresighted spectrometer. Science goals consist of requests to take spectral measurements at certain locations or regions. These goals are prioritized so that, if necessary, low priority goals can be preempted (e.g., due to low battery power). Science goals are divided among the three rovers. Each rover is identical and is assumed to have a spectrometer onboard as well as other resources including a solar panel that provides power for rover activities and a battery that provides backup power when solar power is not available. The

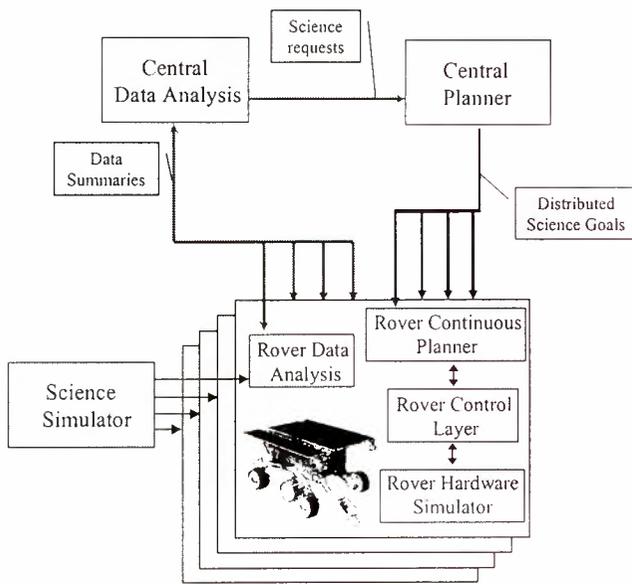


Figure 1: Multi-rover Execution Architecture

possible. Collected science data can be transferred to an orbiter where it is stored in memory.

3. MULTI-ROVER EXECUTION ARCHITECTURE

The overall system architecture is shown in Figure 1. The system is comprised of the following major components:

- **Planning:** A dynamic, distributed planning system that produces rover-operation plans to achieve input rover-science goals. Planning is divided between a central planner, which efficiently divides up science goals among rovers, and a distributed set of planners, which plan for operations on individual rovers and can perform re-planning if necessary.
- **Rover Control Software:** Control software from the NASA JPL Rocky 7 rover that handles execution of low-level rover commands in the areas of navigation, vision and manipulation. This software performs low-level monitoring and control of each rover's subsystems.
- **Multi-Rover Real-Time Simulator:** A multi-rover simulation environment that is used to simulate the planetary terrain and rover hardware operations within that environment. This simulator models rover kinematics and generates sensor feedback which is relayed back to each rover planner.
- **Data Analysis:** A distributed machine-learning system which performs unsupervised clustering to model the distribution of rock types observed by the rovers. This distribution is also used for prioritizing new targets for exploration by the rovers.

- **Science Simulator:** A multi-rover science simulator that models different geological environments and rover science activities within them. The science simulator manages science data for the current terrain, tracks rover operations within that terrain, and reflects readings by rover science instruments

The overall system operates in a closed-loop fashion. Science goals (e.g., a spectrometer reading at a certain location) are given to a central planner which assigns them to individual rovers in a fashion that will most efficiently serve the requests. Each rover planner then produces a set of actions for that rover which will achieve as many of its assigned goals as possible. These action sequences are executed using the rover low-level control software and a multi-rover hardware simulation environment which relay action and state updates back to each onboard planner. If necessary, each onboard planner can perform re-planning when unexpected events or failures occur.

Action sequences are also executed within the science simulator and any gathered data is sent to the rover data-analysis modules. These modules form local models of the observed data that are broadcast to the central analysis module. This module forms a global rock-distribution model and generates a new set of observations goals that will further improve the accuracy of the model. In this way, the data analysis system can be seen to take the role of the scientist driving the exploration process. New science goals are then sent to the centralized planner and the overall cycle continues until enough data is gathered to produce distinct models for any observed rock types.

In the next few sections, we discuss each of the architecture components in more detail.

3.1 DISTRIBUTED, CONTINUOUS PLANNING

To produce individual rover plans for a team of rovers, we have developed a distributed planning environment utilizing the CASPER planning system [Chen et al., 1999]. CASPER (Continuous Activity Scheduling, Planning, Execution and Replanning) is an extended version of the ASPEN system [Fukanaga et al., 1997] that has been developed to address dynamic planning and scheduling applications. CASPER employs techniques from AI planning and scheduling to automatically generate the necessary rover-activity sequence to achieve the input goals. This sequence is produced by utilizing an iterative repair algorithm [Minton and Johnston, 1988; Zweben, et al., 1994] which classifies conflicts and attacks them each individually. Conflicts occur when a plan constraint has been violated where this constraint

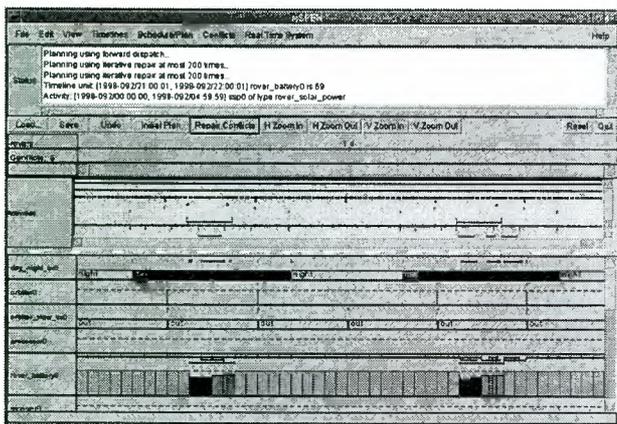


Figure 2: Example Rover Plan

constraint has been violated where this constraint could be temporal or involve a resource, state or activity parameter. Conflicts are resolved by performing one or more schedule modifications such as moving, adding or deleting an activity. Examples of conflicts include a rover that is at the incorrect location for a scheduled science activity or having too many activities scheduled for one rover, which oversubscribes its power resources. Figure 2 shows an example rover-plan displayed in the CASPER GUI.

To support missions with multiple rovers, we developed a distributed planning environment where it is assumed each rover has an onboard planner. This allows rovers to plan for themselves and/or for other rovers. And, by balancing the workload, distributed planning can be helpful when individual computing resources are limited. Our approach to this task was to include a CASPER continuous planner for each rover, in addition to a central, batch planner.

The central planner develops an abstract plan for all rovers, while each agent planner develops a detailed executable plan for its own activities. The central planner also acts as a router, taking a global set of science goals and dividing it up among the separate rovers. For example, a science goal may request an image of a particular rock without concern for which rover acquires the image. The central planner could assign this goal to the rover that is closest to the rock in order to minimize the traversals of all rovers. This master/slave approach is just one approach to distributed planning which could be utilized for this architecture; we are also experimenting with several other forms of distributed planning for this task [Rabideau, et al., 1999].

In order to enhance the quality of the produced schedules, we have implemented heuristics for assigning rovers to

goals and for deciding on the order in which to visit each of the specified locations. The heuristics borrow from algorithms for finding solutions to the Multiple Traveling Salesman Problem (MTSP) [Johnson et al., 1997]. With multiple rovers covering the same area, the planner prefers paths that minimize the total traverse time of all the rovers.

To achieve a high level of responsiveness for each onboard rover planner, we also utilize a continuous planning approach. Rather than considering planning a batch process in which a planner is presented with goals and an initial state, each rover planner has a current goal set, a current state, a current plan, and state projections into the future for that plan. At any time, an incremental update to the goals or current state may update the current plan. This update may be an unexpected event or simply time progressing forward. Each onboard planner is then responsible for maintaining a plan consistent with the most current information obtained from the rover sensors and low-level control software. The current plan is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, the planner stands ready to continually modify the plan to bring it back into sync with the actual rover state.

3.2 ROVER CONTROL SOFTWARE

To handle low-level rover control issues, we utilize the Onboard Rover Control & Autonomy Architecture (ORCAA) software developed for the NASA JPL Rocky 7 rover [Volpe et al., 1997, Hayati & Arvidson, 1997]. In the ORCAA software, asynchronous rover activities are initiated by a queue of rover commands. These activities are represented using asynchronous finite state machines (FSMs) and synchronous data-flow control loops. When the rover receives a command sequence, these commands cause state transitions in one of three main state machines: Navigation, Vision and Manipulation. For example, in the Navigation FSM, possible states include "Idling", "Steering", "Driving", etc. State transitions in these FSMs are used to run different execution methods and are often used to begin the execution of synchronous processes, which perform monitoring and control of the rover's subsystems.

This software also relays sensor information and command updates back to the overlying planning system. This information includes command updates such as whether a command was successfully executed and sensor values such as the current sun angle or level of battery power.

3.3 MULTI-ROVER REAL-TIME SIMULATION

In order to accomplish preliminary testing of this architecture, a real-time simulation environment has been developed using the DARTS/Dshell software [Biesiadecki, et al., 1997]. The Rover Analysis Modeling and Simulation (ROAMS) [Yen et al., 1999] extension of DARTS/Dshell was first slated towards modeling single-rover operations and is based on the Rocky 7 Mars rover. Currently, the simulator rover model is comprised of its mechanical, electrical, and sensor subsystems, and is connected with the on-board (Rocky 7) software. Several terrain models have been incorporated and we have developed solution techniques that permit a real-time simulation of the rover traversing a Mars-like terrain on a workstation platform.

The basic component of the simulator is the solution of inverse kinematics for the rover traversing a Mars-like terrain. Building on this novel solution technique, we have applied the ROAMS rover simulator to testing the Rocky 7 on-board software. The control and navigation algorithms of the control software are used to drive the Rocky 7 rover model against a terrain with randomly distributed rocks. Applying the DARTS/Dshell methodology, we implemented models for hardware devices, such as a panoramic spectrometer, sun sensor, tilt sensor, obstacle detection camera, solar panel, battery, etc., to feed the subsystems. Also, based on the numerical solution of inverse kinematics, the hardware instrument models provide high-fidelity synthetic data to test the control and navigation code. Overall, this environment permits a fast and better design and implementation of the rover's software subsystem.

For the multiple rover architecture, this single-rover simulation model has been extended to support several cooperating rovers. An example situation involving three rovers is shown in the ROAMS interface in Figure 3. For use with this architecture, we developed additional hardware models, including a collision avoidance model, an obstacle detection model, models of power units, and the capability for running multiple rovers in ROAMS. Due to the stability and accuracy of the numerical solution, these device models can provide high quality sample data for the control software and ultimately the planning system. For example, the power source of Rocky 7, including a solar panel and a battery, can produce accurate reading of the power level due to the prediction of the panel's attitude and the wheel's motor output. As explained above, these and other sensor values can be fed back to each rover planner so that a valid command sequence can be consistently maintained for each rover.

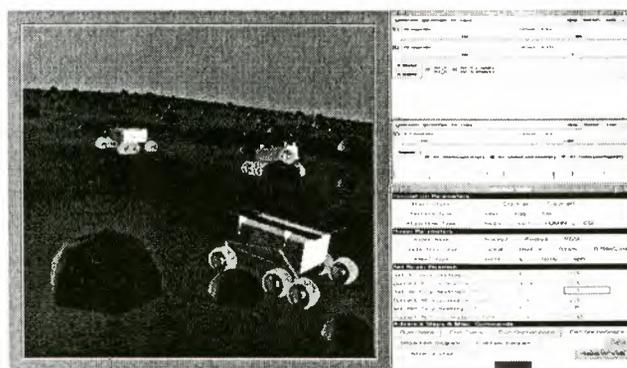


Figure 3: Three rovers in sample terrain

3.5 SCIENCE-DATA ANALYSIS

To perform science analysis, we use a machine-learning system which performs unsupervised clustering to model the distribution of rock types in the observed terrain [Estlin, et al., 1999]. Clustering is performed by a distributed algorithm where each rover alternates between independently performing learning computations using its local data and updating a global-distribution model through communication among rovers. The model used for this distribution is a simple K -means-like unsupervised clustering model, where each cluster represent a different rock type in the sensor space. Currently, each sensor reading is a spectral measurement returning values at 14 wavelengths; learning takes place in the full 14-dimensional continuous space. A sample cluster model (shown for 2 of 14 dimensions) is shown in Figure 4.

After a new set of science readings is acquired, each rover sends a small set of parameters, which summarizes its local data, to the central clusterer. The central module then integrates this data into an updated global model and broadcasts that model to all rovers in the system. This process continues iteratively until convergence.

Output clusters are also used to help evaluate visible surfaces for further observation based upon their "scientific interest." Specifically, the system tries to increase the accuracy of the clustering model by obtaining data readings in regions that are likely to improve the model. Each update of the global clustering model determines a new set of interesting science goals, i.e. planetary locations to be explored by the rovers. These observation goals are then sent as formal goals by the learner to the planner. Thus, the science analysis system can be viewed as driving the science process by analyzing the current data set and then deciding what new and interesting observations should be made.

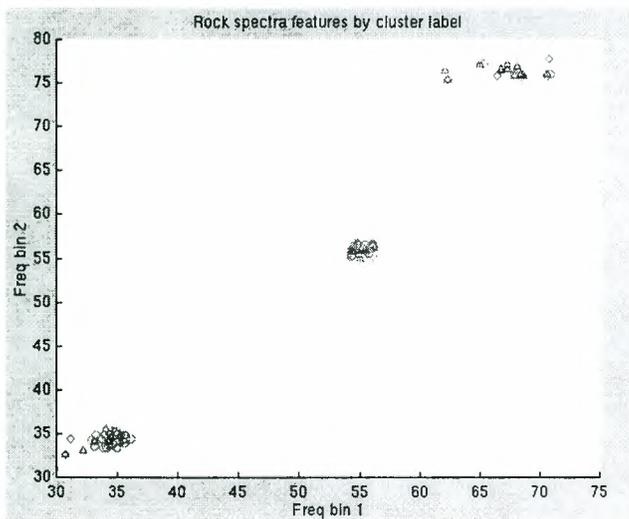


Figure 4: Example spectra-feature space

3.5 SCIENCE SIMULATOR

A science simulator designed for this particular geological scenario provides data for the science-analysis system by simulating the data-gathering activities of the rover. Different Martian rockscapes are created for use in the simulator by using distributions over rock types, sizes and locations. The size and spatial distributions of the rockfield were developed by examining distributions of rocks observed by the Viking Landers, Mars Lander and Mark Pathfinder. The distribution of minerals that can occur in rocks was developed in collaboration with planetary geologists at JPL, and the spectra associated with rocks are generated from the spectra of the component minerals via a linear-mixing model. When science measurements are requested from a terrain during execution, rock and mineral spectral models are used to generate sample spectra based on the type of rock being observed. This data is then communicated to the relevant rover data-analysis module.

4. RELATED WORK

While there has been a significant amount of work on cooperating robots, most of it focuses on behavioral approaches that do not explicitly reason about assigning goals and planning courses of action. One exception is GRAMMPS [Bummitt and Stentz, 1988], which coordinates multiple mobile robots visiting locations in cluttered, partially known environments. GRAMMPS also has a low-level planner on each robot, however it does not look at multiple resources or exogenous events.

It also does not utilize a learning system to analyze gathered data and deduce new goals.

Many cooperative robot systems utilize reactive planning techniques [Mataric, 1995; Parker, 1999]. These systems have been shown to exhibit low-level cooperative behavior in both known and “noisy” environments. However, these systems have not been shown useful for mission planning where a high-level set of science and engineering goals must be achieved in an efficient manner.

The idea of having a scientific-discovery system direct future experiments is present in a number of other systems [Rajamoney, 1990; Nordhausen and Langley, 1993], however none of these have been utilized for multiple-robot scenarios. In our architecture, the data-analysis system is integrated with a planning system and real-time simulator, which plan and execute detailed activity sequences needed to perform each experiment. The data-analysis system also directly interacts with the environment and is specialized to problems and scenarios in planetary science.

5. FUTURE WORK

We have a number of planned extensions to this work. First, we intend to extend the overall architecture to be more robust and able to handle rover failure situations. For instance, if a rover fails, the distributed planning system should recognize this failure (e.g., the rover has not responded for a certain amount of time), refrain from sending any new goals to that rover, and re-assign any current goals assigned to that rover.

Another important addition is to integrate the Envelope Learning and Monitoring using Error Relaxation (ELMER) system [Decoste, 1997] to model rover-resource use such as battery power or onboard memory. ELMER uses statistical machine-learning techniques to learn and refine input-conditional limit functions from historic and/or simulated data. These limit functions define context-sensitive upper and lower boundaries, within which future resource-data is expected to fall. This system will enable more accurate resource modeling, which can be used by the planner to better estimate future resource levels.

We also plan to increase the fidelity of the simulation by adding models of onboard cameras and other instruments, and extending the simulator to model communication between each rover. Currently, it is assumed rovers share science data through the central data-analysis model, however this communication is not explicitly represented

in the simulator. We would also like rovers to share plan information, which would allow them to directly coordinate with each other during plan execution and would allow us to experiment with different forms of distributed planning that require communication among agents [Tambe, 1997; Sandholm, 1993].

Last, we plan on testing the overall architecture in a more realistic setting using actual rovers as opposed to the hardware and science simulators described previously. This testing will occur in the JPL Mars yard and/or in outside field tests using rovers such as JPL's Rocky 7 and Rocky 8.

6. CONCLUSION

In conclusion, using multiple rovers can greatly increase the capabilities and science return of a mission. In this paper we have presented an integrated architecture that combines techniques from several fields to effectively plan for and coordinate rover activities, execute these activities in a real-time environment simulator, monitor rover-execution status, and effectively respond to unexpected events through re-planning. This integrated system exhibits great potential for advanced applications in areas of design, engineering, and distributed planning for mobile robotic systems.

ACKNOWLEDGEMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

- Biesiadecki, J., Henriquez, D., and Jain, A., "A Reusable, Real-Time Spacecraft Dynamics Simulator," *Proceedings of the Sixth Digital Avionics Systems Conference*, Irvine, CA, October 1997.
- Bummit, B., and Stentz, A., "GRAMMPS: A Generalized mission planner for multiple robots in unstructured environments," *Proceedings of the IEEE Conference on Robotics and Automation*, 1988.
- Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Integrated Planning and Execution for Autonomous Spacecraft," *Proceedings of the 1999 IEEE Aerospace Conference*, Aspen, CO, March, 1999.
- Estlin, T., Gray, A., Mann, T., Rabideau, G., Castano, R., Chien, S., and Mjolsness, E., "An Integrated System for Multi-Rover Scientific Exploration," *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, FL July 1999.
- Decoste, D., "Mining Multivariate Time-Series Sensor Data to Discover Behavior Envelopes," *Proceedings of the Third Conference on Knowledge Discovery and Data Mining (KDD-97)*, Newport Beach, CA, August 1997.
- Fukanaga, A., Rabideau, G., Chien, S., and Yan, D., "Toward an Application Framework for Automated Planning and Scheduling," *Proceedings of the 1997 International Symposium of Artificial Intelligence, Robotics and Automation for Space (iSAIRAS-97)*, Tokyo, Japan, July 1997.
- Hayati, S., and Arvidson, R., "Long Range Science Rover (Rocky 7) Mojave Desert Field Tests," *Proceedings of the 1997 International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS-97)*, Tokyo, Japan, July 1997.
- Johnson, D. and McGeoch, L., "The Traveling Salesman Problem: A Case Study in Local Optimization." *Local Search in Combinatorial Optimization*, Aarts, E. and Lenstra, J., eds., John Wiley and Sons, London, 1997, pp. 215-310.
- Mataric, M., "Designing and Understanding Adaptive Group Behavior," *Adaptive Behavior*, 4(1):51-80, December 1995.
- Minton, S., and Johnston, M. "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems," *Artificial Intelligence*, 58:161-205, 1988.
- Nordhausen, B., and Langley, P., "An Integrated Framework for Empirical Discovery," *Machine Learning* 12:17-47.
- Parker, L., "Cooperative Robotics for Multi-Target Observation," *Intelligent Automation and Soft Computing*, 5(1):5-19, 1999.
- Rabideau, G., Estlin, T., Chien, S. and Barrett, T., "A Comparison of Coordinated Planning Methods for Cooperating Rovers," Submitted to the *AIAA99 Space Technology Conference*, Albuquerque, NM, September, 1999.
- Rajamoney, S., "A Computational Approach to Theory Revision," *Computational Models of Scientific Discovery*, Shrager, J., and Langley, P., eds., Morgan Kaufman, 1990.

Sandholm, T., "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations," *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., July 1993.

Tambe, M., "Towards Flexible Teamwork," *Journal of Artificial Intelligence Research*, 7:83-124, 1997.

Volpe, R., Balam, J., Ohm, T., and Ivlev, R., "Rocky 7: A Next Generation Mars Rover Prototype," *Journal of Advanced Robotics*, 11(4), December 1997.

Yen, J., Jain, A., and Balam, J., "ROAMS: Rover Analysis, Modeling and Simulation Software," *Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, The Netherlands, June 1999.

Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," *Intelligent Scheduling*, Zweben, M., and Fox, M., eds., Morgan Kaufman, 1994, pp.241-256.

Spacecraft Autonomy

Integrated Planning and Execution for Satellite Tele-Communications

Kanna Rajan* Christian Plaunt* Barney Pell† Nicola Muscettola‡

{kanna,plaunt,pell,mus}@ptolemy.arc.nasa.gov

NASA Ames Research Center, Mail Stop 269-2

Moffett Field, CA 94035-1000

Abstract

The next generation of communications satellites may be designed as a fast packet-switched constellation of spacecraft able to withstand substantial bandwidth capacity fluctuation ranging from unstable weather phenomena to intentional jamming of communication. We have designed and partially implemented an architecture for managing satellite telecommunications network resources. Our approach supports advance reservations and dynamic requests, negotiation and fulfillment of prioritized Quality of Service (QoS) contracts, graceful degradation in the presence of dynamic tasks and environmental changes, and optimization of geometrically constrained resources. Our integration of planning and execution to address this task uses planning to avoid resource contentions among requested activities and to configure an independently competent execution system. Our system can be used in routine operations or as a simulation-based design tool.

1 Introduction

The current revolution in information technology continually produces new advances in communications capability. In its vision for the future, the US Department of Defense (DoD) perceives information as critical to tactical and strategic decisions and satellite communication as an essential operational component (Department of Defense, Space Command 1997). One of the critical technologies being closely scrutinized is the application of Asynchronous Transfer Mode (ATM) technology to satellite communications systems. Satellites are limited and expensive communications resources, and ATM technology, through quality-of-service (QoS) contracts and statistical multiplexing, offers greater flexibility and capacity than existing circuit-switched systems currently used for military satellites.

However, extending ATM to support military communication satellites requires innovations beyond standard ATM networks. Unlike most quality of service work, the military domain requires advance guarantees and hierarchical resource allocation. One of our major goals is to support these domain requirements while increasing the efficiency of resource utilization and supporting unplanned resource allocations. In addition, we must also support geometric constraints and optimizations resulting from satellite beam management.

The DoD is in the process of evaluating the design parameters needed for such a system using simulation based design. One of the tools needed as part of this design analysis is a prediction and execution component. For this, we are proposing the use of the Planner/Scheduler (PS) and Smart Executive (Exec) subsystems of the Remote Agent (RA) (Bernard *et al.* 1998; Pell *et al.* 1998a; Muscettola *et al.* 1998b). The RA will be the first artificial intelligence-based autonomy architecture to reside in the flight processor of a spacecraft (NASA's Deep Space One (DS1)). We have built from these components a new system, the Remote Agent for Satellite Tele-Communications (**RAST**).

Similar to other high-level control architectures (Bonasso *et al.* 1997; Wilkins *et al.* 1995; Drabble *et al.* 1996; Simmons 1990; Musliner *et al.* 1993), **RAST** clearly distinguishes between a *deliberative* and a *reactive* layer. In the current context PS develops a schedule based on requested bandwidth allocations known a priori. Planning/Scheduling is used to smooth out resource consumption resulting from future requests, to establish configurations to enable future requests (like needing beam coverage to make a subsequent call request), and to set execution priorities to support efficient responses to dynamic requests, taking into account environmental projections. Planning and execution must support quality of service guarantees in highly dynamic environment. PS negotiates among requests in advance, and Exec negotiates contracts and adjusts and sheds tasks based on variable priorities at

*Caelum Research Corporation

†Marketplace.net

‡RECOM Technologies

run time in the dynamic environment. As a result, the execution system handles planned and unplanned dynamic resource requests and supports load balancing, quality of service, fast responses, and graceful degradation.

1.1 Integrated Planning and Execution

In addressing these issues, we have explored a novel and interesting integration of planning and execution. There are number of ways to integrate planning and execution that have been explored:

- plans as coordination routines for multiple agents (including humans).
- planners generate tasks networks, which are then executed by doing the right task at the right time.
- plans as programs, which are run by the executive (e.g. planning in CIRCA (Musliner *et al.* 1993) generates a program comprised of test-action pairs).
- plans as advice, which the executive uses in running is own goals (e.g. planner produces a navigation map, which exec uses when it is heading to targets).

Our integration has aspects of several of these. The advance planning of resource allocations tells users when they should place their calls, thus preventing resource conflicts before they happen and guaranteeing resource availability. The plans themselves have task networks with explicit configuration actions (beam configuration activities) and also advice in terms of execution priority updates and projections used for monitoring plan execution.

Most planning and execution work addresses specific resource requirements, whereas this work addresses multiple types of quality-of-service contracts, with resource sharing (statistical multiplexing), preemption, and even reconfiguration (in the case of beam migration and repositioning).

Finally, our executive has independent competence, and can run with or without a plan, but performance can be enhanced with a plan.

1.2 Organization

In Section 2 we describe the overall problem in greater detail. Section 3 describes the **RAST** architecture. In Section 4 the details of the approaches taken in the Planning/Scheduling component. Section 5 covers the run-time execution system. We explore work related to this project in Section 6, and in Section 7 we consider the open issues and future work to which this project points. Finally, in Section 8 draw our conclusions.

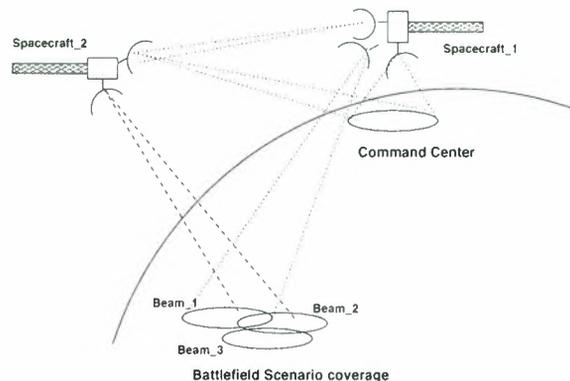


Figure 1: A simplified satellite telecommunications scenario supported by an ATM networked constellation

2 The Domain

2.1 Motivation

We are motivated by the requirements of complex, mission critical satellite tele-communications systems. In this domain, there are several conflicting goals which influence many levels of design choices (for instance, guaranteed connections versus maximal network throughput, fluctuating bandwidth, conflicting demand patterns, quality of service). These considerations make this a particularly interesting domain for our exploration. A communication network in this domain must be highly configurable and controllable in order to handle the strategic needs of the user, and also be highly autonomous in order to function efficiently in the potential absence of such control.

The objective of this overall effort is to build an operational system which can also be used as an analysis and design tool, capable of both controlling or simulating and analyzing multiple configurations, topologies, and environments in the unstable environment of mission critical communications with the purpose of controlling or designing a future generation satellite based telecommunications system. When used as a design tool, the agent generates output for designers to evaluate operations policy and provides flexibility in the operational constraints modeled. Rapid iteration of the system design is possible by comparison of throughput performance results for candidate designs. Moreover, a network planning and execution agent can optimize the policy for users and potential customers can be advised in their planning for network usage. At present, satellite communications network planning is a computation and labor intensive element of operations. The model-based planning and execution agent could improve efficiency and reduce cost and effort.

The work described in this paper is further moti-

vated by our interest in several research aspects of this domain. Issues include using planning and scheduling to smooth out resource consumption resulting from future requests, establishing configurations to enable future requests (e.g. requiring beam coverage to enable a subsequent call request), and setting execution priorities to support efficient responses to dynamic requests, taking into account environmental projections.

2.2 A Brief Background on ATM

The domain consists of a constellation of spacecraft which act as ATM switches directing and controlling traffic flow from a number of sources to a number of destinations (Figure 1). Traffic is based on an ATM model with different *contract types* and *priorities*. Contracts ensure a Quality of Service (QoS) so that guarantees can be made a priori about specific call connections. The user must inform the network upon connection setup of both the expected nature of the traffic and the type of QoS contract required. The idea is to ensure that critical calls, that need to get through under all circumstances, are guaranteed bandwidth capacity while those of lower priority — regardless of contract type — or of a non-critical nature are allocated bandwidth on an as available basis. Following are some terms from the ATM literature (see (Varma 1997) for a concise tutorial) we will use in this paper:

- CBR (Constant Bit Rate): Bit rate remains constant over duration of connection; requires delay, jitter and bandwidth guarantees¹.
- VBR (Variable Bit Rate): Intended for supporting bursty data and defined by peak and average rate.
- ABR (Available Bit Rate): Intended for data transmission which do not preserve any timing relationship between source and destination.

In addition, in this domain we also deal with call priorities. For instance, critical calls that need to get through under all circumstances will have the highest priority. Such calls may be of any contract type, depending on the nature of the call (voice, video, data, etc.). Less critical calls might request an “expensive” contract (e.g. CBR), but also be willing to accept a less expensive contract (e.g. ABR) if that is the best contract available.

Currently, such communication is managed by restricting the identity, time, and bandwidth allocations of people and equipment that can use system to communicate. Multiple high priority channels are reserved

¹We distinguish here between different peak and average bandwidth requirements among QoS contracts. E.g., CBR 2 requires roughly twice as much bandwidth as CBR 1.

just in case an important message needs to be sent. In this approach not only is the complete bandwidth allocation preallocated as a “pipe” (i.e once allocated the resources are completely tied to the user), but dynamic request allocations can only be accepted if the request is of a high enough priority, to preempt an ongoing call when enough capacity is not available. Needless to say, this is a highly suboptimal approach, especially in the forward tactical areas where frequently a large amount of bandwidth is needed on demand and where no accurate predictions can be made a priori.

3 System Architecture

The system architecture consists of several modules, as shown in Figure 2. The architecture is based on the components of the Remote Agent architecture (Pell *et al.* 1998a), plus several domain specific components (or simulators) which are used either at plan-time or run-time. In this section we discuss the various components of the system architecture with each module annotated as in Figure 2.

3.1 Plan-Time Components

As an operational system (see Figure 2), the Planner/Scheduler (3) takes input in the form of authorization requests from a Request Generator (1) and estimates of the effects of environmental conditions on bandwidth capacity fluctuations at run-time from a plan-time Environmental Expert (2). From this input, the PS generates a plan which includes the reservation schedule, beam movements, required configuration, policies, priority schedules, and so on, that will be required to carry out the authorized calls while maximizing dynamic potential. The schedule produced from these inputs is supplied both to the users of the system (in order to regulate usage by informing users whether their reservation has been accepted or not and when to place their call) and to the run-time execution system. Thus, the plan time components configure usage patterns as well as system resources and priorities.

3.2 Run-Time Execution Components

The run-time execution components monitor and execute the execution schedule while responding to dynamic requests and environmental changes. The major tasks at run-time are (i) to determine whether a call request can and should be admitted to the system, and (ii) to administer those call requests which have already been admitted to the system.

The execution schedule is executed by the Plan Runner (4), as follows. The primary form of configuration change is to move a beam to a new location, by sending the corresponding command to the Beam Manager

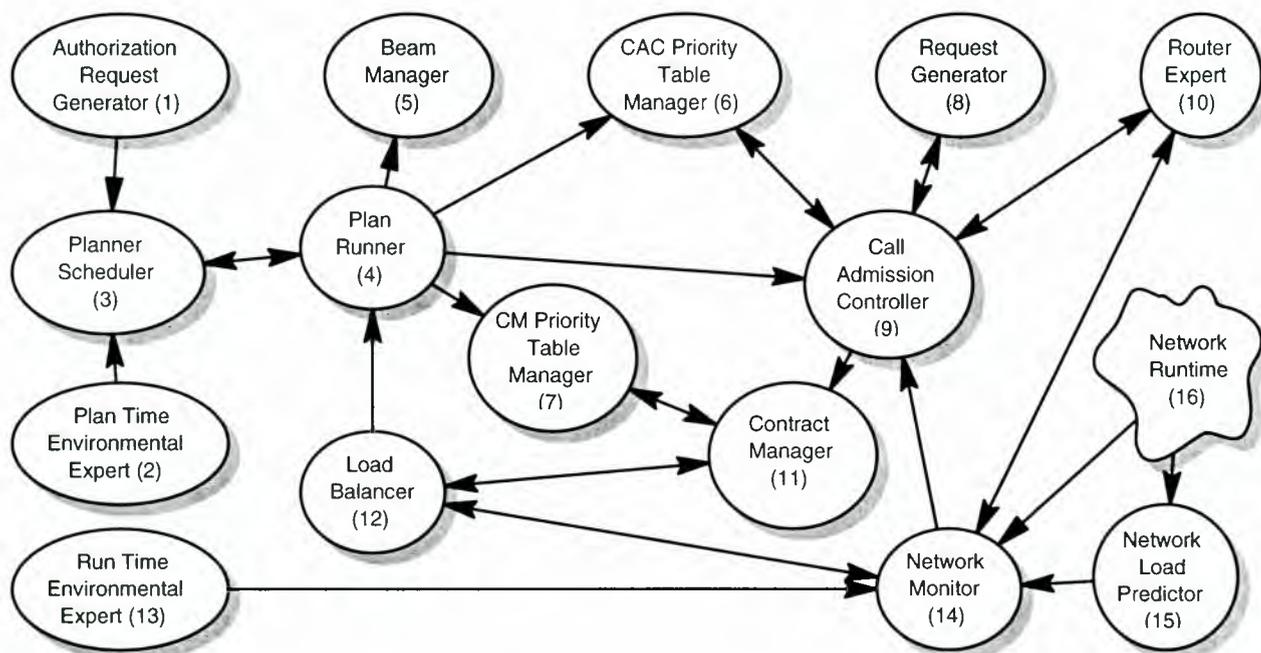


Figure 2: The **RAST** application architecture for a Modular High Level ATM Network Controller. This architecture can also be used as a simulation based design tool by simulating the shaded components in this figure.

(5). Policy and priority changes are issued by sending the commands to the corresponding priority table manager, one for the Call Admission Control (CAC) Priority Manager (6), and one for the Contract Manager (CM) Priority Table Manager (7). These priority tables, which are consulted dynamically at run time, control the behavior of the major run-time execution components of the system, the Call Admission Controller (9) and the Contract Manager (11). The Plan Runner uses information from the Load Balancer (12) to determine if the plan execution is proceeding within the bounds of the planner. If not, it continues with the current policy while requesting a new plan.

At run time, the Call Admission Controller (9) receives initiation requests from distributed users, represented here as a dynamic Request Generator (8). These call initiation requests are typically a variable mixture of scheduled and unscheduled requests. Each such request specifies information about the call contract requested, which includes quality of service contract types and parameters. When the system is running as simulation, the mixture of these requests is designed to simulate various "real world" probability distributions.

The CAC decides how to handle the requests based on (i) the policy specified by the CAC Priority Table Manager (6), (ii) the state of the network (i.e. current coverage, capacity, and usage) as reported by the

Network Monitor (14), (iii) the availability of communication resources as reported by the Router Expert (10), and (iv) the allowable types of contracts in the call request received. The initiation requests can be (i) serviced as requested, (ii) serviced but with some alternative contract type (as allocated by the Router Expert (10) and accepted by the call requester), or (iii) denied.

The Router Expert (10) allocates (or denies) connection contracts in response to requests from the Call Admission Controller (9). It decides whether or not to allocate such contracts based on several factors, including the state of the network reported by the Network Monitor (14) and the Network Runtime (16), availability of point to point virtual circuits, and so on. When a call request is accepted, the call and its allocated contract are passed to the Contract Manager (11) which tracks the calls thereafter.

The other major functionality provided by the run-time system is contract management, embodied here in the Contract Manager (11) and Load Balancer (12).

The Contract Manager is the run-time module which keeps track of all the contracted calls which have been received from the Call Admission Controller (10). The Contract Manager, based on the priority policy in the CM Priority Table Manager (7), and the current state of the network as reported by the Load Balancer (12), controls all of the "in progress" call traffic.

If at any time insufficient resources exist to support the current calls with sufficient robustness, The Contract Manager interacts with the Load Balancer (12) to free up resources. The Load Balancer keeps network usage within capacity by migrating call among different possible routes, reducing the bandwidth of calls with contracts which allow this, shedding low priority calls, and potentially repositioning beams to optimize ground coverage. In conjunction with the Contract Manager, lower priority calls can be moved or killed to make way for higher priority calls. This ability to migrate or shed calls becomes particularly important when the network is operating in an unstable, dynamic environment where network capacity can fluctuate enormously.

3.3 Network Components

The Network Monitor (14) is the interface between the run-time execution system and the network itself. Based on input from the run time Environmental Expert (13), the Network Runtime (16), and the Network Predictor (15), it reports the current total bandwidth capacity and current actual usage to the Load Balancer (12), the Router Expert (10) and the Call Admission Controller (9) at run time. The run time Environmental Expert (13) simulates changes of the environment which affect bandwidth capacity on the beams (e.g. weather changes, hardware problems, jamming, etc.). The Network Predictor (15) is a traffic expert which can be used by the plan-time and run-time Environmental Experts (2, 13) for better network usage predictions.

Finally, the Network Runtime (16) is the real (or simulated) network. Primarily, it feeds the Network Monitor (14) with run time fluctuations in network load, capacity, congestion, outages, and so on.

3.4 Simulation-Based Design Tool

The system can also be used as a simulation-based design tool. This is accomplished by simulating user and environmental factors (see the shaded components of Figure 2). Our modular design enables the external interface points to be unaware of whether the input is coming from a simulator or from operational use. For example, rather than running of real authorization requests, input to the planner can be provided by a statistical Authorization Request Generator. Similarly, dynamic calls can be generated in accord with alternative test cases for usage patterns, rather than coming from real users, and environmental conditions such as failures, weather changes, or jamming can be simulated. Together, this supports use of the system for 'what-if' analysis, in which we run different networks,

policies, and assumptions through simulated operational contexts and collect statistics such as throughput and call completion rates.

Clearly, this architecture is divided between plan-time and run-time. The focus of the plan-time components is to smooth the fluctuations in the actual run-time call requests as much as possible. The focus of the run-time components is to respond to just such fluctuations.

4 The Planning/Scheduling Component

The objective of the Planner/Scheduler (PS) is to schedule system resources and requested traffic allocations as optimally as possible. The Exec then takes this generated schedule and changes system configuration to support the scheduled calls and to meet the demands of dynamic real-time traffic to the extent possible. Using PS in the loop helps to optimize run time configuration and allocation and also permits dynamic call initiation by reconfiguring the network (antennas) to cover critical regions.

The PS is a timeline based non-linear temporal constraint posting planner which uses chronological back-track search. Temporal information in the plan is represented within the general framework of Simple Temporal Constraint networks, as introduced by Dechter, Meiri, and Pearl (Dechter *et al.* 1991) in a Temporal Database (TDB). Details of the HSTS planner/scheduler and TDB can be found in (Muscuttola 1994).

4.1 The Scheduling Process

The PS component generates a schedule of calls based on a domain model. The model describes the set of constraints that all the calls have to satisfy. The schedules consist of several parallel *timelines*, each of which consist of a sequence of *tokens*. A timeline in this domain describes the condition of each channel over time. Each call is a token on a timeline. In our domain there are primarily three token types; a call request token which specifies all the request parameters necessary for scheduling, a beam capacity token type which gives instantaneous capacity at any time and a beam location token type which specifies to the planner where the beam coverage is. Beam slewing (when the spacecraft's beam is to be transitioned from one area of coverage to another) is assumed to be instantaneous so no token is required.

Beam Scheduling The PS receives as input a traffic request allocation which specifies for each call request, the contract type, priorities, requested capacity, duration of the call and the source and destination

target areas. The PS then tentatively builds a partial plan based on the requested start times and duration. A constraint is posted on the beam timeline specifying a region of beam coverage which will satisfy the call constraints. Given a set of such requests, the planner searches through the space of possible configurations of the limited set of beams in order to optimize coverage. A simple example is shown in Figure 3. Calls (represented as tokens) assigned to some channel (represented as timelines) request bandwidth (not shown) and beam coverage. As the partial schedule is built both the location and the duration of the beam at those requested locations get refined. When no more beam requests are to be satisfied the PS can then determine slew boundaries when the spacecraft can move the beam from one area of coverage to another. Scheduling beam coverage as a result, is a matter of ensuring that most (if not all) requested calls are covered by some beam. Those calls that are not covered will be rejected.

Bandwidth Scheduling We currently use a simple forward dispatching strategy which is adequate to schedule all calls. As a result calls scheduled on a specified channel take up the 'real estate' on that channel. Any subsequent call also requiring capacity on that channel and intersecting temporally with a previously scheduled call will currently be rejected at the scheduling phase. Such rejected calls however have the opportunity to request bandwidth at run time where lower priority and contract type calls can be shed. In the future however, the problem that needs to be tackled is complicated by the introduction of contract types and priority. In that event, contract types and priority schemes will allow preemption of scheduled calls already placed on the timelines. So for instance if a CBR request is posted to a temporal duration $[t_1, t_2]$ and if the bandwidth capacity exists, this call could be accommodated within the temporal duration. If not, any previously scheduled ABR or VBR calls would need to be rescheduled to accommodate this incoming CBR call. Correspondingly if a non-CBR request comes in after a CBR call capacity is satisfied, then depending on the request type, its duration and requesting range, the new call request could be either moved or rejected outright. This strategy will have to ensure that a CBR will always have the capacity reserved for it when scheduled, while a ABR could be shed at execution time. Effectively this calls for a CAC (9) style priority table manager, *but at schedule time*. Policies for this table can then be adjusted to allow selection of different scheduling strategies by the user.

4.2 Model Representation

The plan model consists of definitions for all the timelines, definitions for all the tokens that can appear on those timelines, and a set of temporal constraints that must hold among the tokens in a valid schedule. The planner model is described in a domain description language (DDL) (Mussettola 1995), and is represented as part of the planner's TDB.

Temporal constraints are specified in DDL by *compatibilities*. A compatibility consists of a *master token* and a boolean expression of temporal relations that must hold between the master token and *target tokens*. An example is shown in Figure 4. The first constraint specifies that a call request master token can only be satisfied if its peak bandwidth capacity is satisfied, and it is within the confines of some beam which provides coverage. Additionally, another call is to follow (precede) it on this channel.

Heuristics tell the planner what decisions are most likely to be best at each choice point in the planner search algorithm, thereby reducing the search. In HSTS, the heuristics are closely intertwined with the model and can be used to specify which compatibility to place on the planners agenda mechanism to focus its search. In the current system acquiring good heuristics to make the planner search computationally tractable is still an issue.

5 Run-Time Execution

5.1 Dynamic Policy Enforcement

The run-time execution system's objective in **RAST** is to enforce a small number of communication policies in a variety of environmental network loading situations in order to analyze their effects on the system. That is, the Exec's job is (1) to enforce policy on priority-based bandwidth allocation, (2) within that policy, to service the scheduled allocations and configuration changes generated by PS, and (3) to service unscheduled bandwidth allocation requests for bandwidth dynamically as (1) and (2) allow.

In particular, this means that the active run-time policy will determine the default behavior of the Exec (and the behavior of the communications system) when (1) there is no plan available (for whatever reason), (2) between the time when a plan is broken and a new plan is received, and (3) when there is not enough bandwidth to satisfy the current plan, etc.

Currently, the communication policy of interest is (1) to service all dynamic communication requests, scheduled or not, in highest priority first order until either all are serviced or bandwidth capacity is reached; and (2) when bandwidth capacity is exceeded, shed

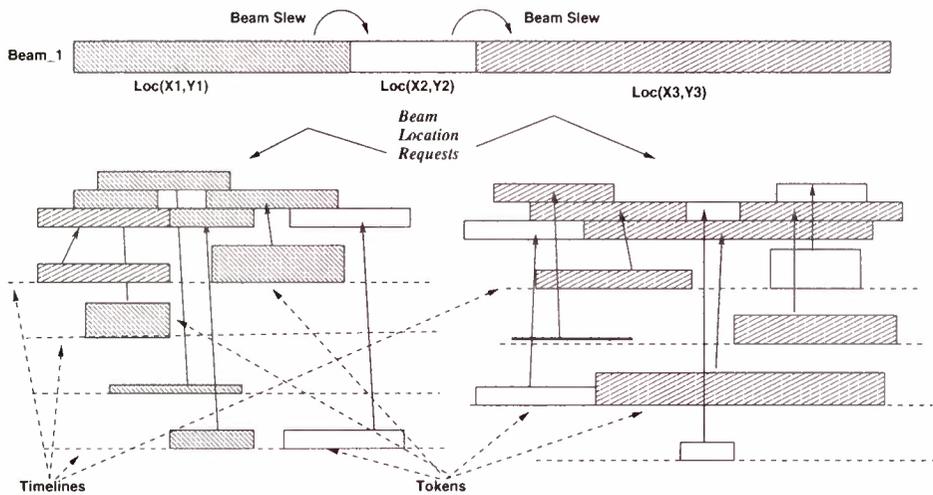


Figure 3: A partial schedule with calls requesting bandwidth and beam coverage. Height of a token indicates amount of bandwidth requested while shading corresponds to a specific beam coverage location. Merging of beam location requests results in the PS scheduling beams as shown in the top of the figure.

```
(Define_Compatibility ;; compats on Call Request
(Call_Request ?ID ?Contract_Type ?Priority ?Cap ?Call_Source
 ?Call_Dest ?Est ?Lst ?Duration ?Beam)
:parameter_functions ( ( (?_duration_ <- ?Duration) ) )
:compatibility_spec
(AND
 ;; requires a specific amount of bandwidth capacity
 (AND (equal (DELTA MULTIPLE (Capacity) (+ ?Cap Used)))
 ;; needs to request a beam location based on the call source
 (contained_by (MULTIPLE ((Beam Beam_1_Pointing_SV)) ((Beam_Loc (?Call_Source))))
 (contained_by (SINGLE ((Beam Beam_1_Pointing_SV)) ((Beam_Loc (?Call_Source)))) )
 ;; is followed either by another call immediately or a NOOP
 (OR (met_by (SINGLE ((Call_UL CALL_1_SV)) (Call_Request)))
 (met_by (SINGLE ((Call_UL CALL_1_SV)) (No_Call_Activity))) )
 ;; is preceded either by another call immediately or a NOOP
 (OR (meets (SINGLE ((Call_UL CALL_1_SV)) (Call_Request)))
 (meets (SINGLE ((Call_UL CALL_1_SV)) (No_Call_Activity))) )
 ;; and allocates an equivalent bandwidth for the downlink phase
 (equal (SINGLE ((Call_DL CALL_1_SV)) ((Call_DL_Request (?Id ?Contract_Type ?Priority
 ?Cap ?Call_Source ?Call_Dest
 ?Est ?Lst ?Duration ?Beam)))))) )

(Define_Compatibility ;; compats on beam pointing/location
(SINGLE ((Beam Beam_1_Pointing_SV)) ((Beam_Loc (?Call_Source))))
:compatibility_spec
(AND
 ;; precedes and succeeds another beam pointing token
 (met_by (SINGLE ((Beam Beam_1_Pointing_SV)) (Beam_Loc)))
 (meets (SINGLE ((Beam Beam_1_Pointing_SV)) (Beam_Loc))) )

(Define_Compatibility ;; compats on no activity fillers
(SINGLE ((Call_DL CALL_1_SV))(No_Call_Activity))
:compatibility_spec
(AND
 (meets (SINGLE ((Call_DL CALL_1_SV)) (Call_DL_Request)))
 (met_by (SINGLE ((Call_DL CALL_1_SV)) (Call_DL_Request))) )
```

Figure 4: An example of a compatibility constraint in the **RAST** Planner model.

communication allocations in lowest first priority order until it is no longer exceeded.

At run time, whenever a conflict arises over bandwidth allocation in either the Call Admission Controller or the Contract Manager, they consult a dynamic table of priorities to determine which call(s) are accepted, migrated, denied, or shed. An example of such a table is shown in Table 1.

Two such tables are maintained for use by the CAC and CM, which consult them in order to increase or decrease bandwidth usage. These tables each have a "manager" which the Plan Runner commands in order to set and reset these tables.

Given a clear policy on such priorities, the run-time system will work even in the absence of a plan. Further, there can be multiple policies which the Exec can enforce, perhaps depending on various environmental or experimental circumstances.

5.2 Run-Time Execution

At run-time, the Exec accepts a stream of call requests, some scheduled in advance, others not. Requests are either to start or release a connection. Start requests contain data about the call's requested contract, assigned priority, origin, destination, and so on.

When the CAC receives a call, with the help of the Router Expert and the Network Monitor, either a route and a contract are granted or denied. If the contract is granted, the call is connected via the route (uplink beam, downlink beam, etc.) assigned at the granted bandwidth and QoS contract, and the call and contract are passed on to the Contract Manager. In the case of a release request, the relevant parts of the system are notified, and the call (and its associated resources) are released.

The Contract Manager administers all of the "in progress" traffic in the system. In order to keep bandwidth usage within capacity, it has the ability to migrate calls among beams, reduce (or "squeeze") the bandwidth usage of calls with certain contracts, or to terminate calls. For example, if bandwidth becomes unexpectedly restricted, the CM can migrate, squeeze, and shed calls in reverse priority order to preserve as many virtual circuits as possible within the bandwidth available (Figure 5) or reduce ABR rates to keep usage within network load capacity. Conversely, when usage falls below capacity, ABR rates can be increased ("unsqueezed") to use the extra bandwidth.

6 Related Work

This paper is among the first work concerned with the problem of integrating planning and execution to support both advanced reservations and dynamic requests

for quality-of-service (QoS) style resource-allocation problems. QoS requirements have emerged mainly in the telecommunications domain and have come to the forefront in the context of Asynchronous Transfer Mode (ATM) communication networks. Major areas of research on intelligent agents in telecommunication applications (Albayrak 1998) include *network configuration, call admission control, and routing*.

Hayzelden (Hayzelden & Bigham 1998) describes a heterogeneous multi-agent architecture for ATM networks. The architecture is similar to **RAST** in that it integrates a deliberative planning layer with a reactive execution layer. The problem focus is somewhat different, however, as they address the problem of network configuration (dynamically adjusting the network topology), while **RAST** addresses call admission control and load balancing (accepting and shedding calls). Also, their approach does not deal with advanced scheduling based on call reservations; rather, their planning agents watch over the network and plan modifications to it based on observed usage patterns. Similar comments pertain to the ARCHON multi-agent system (Jennings *et al.* 1996), which was also applied to network monitoring and configuration.

One aspect of network configuration that does fall within the present scope of **RAST** is *beam management*. Unlike the context of terrestrial ATM networks, where every source and destination have automatic coverage, the satellite context especially requires advanced reservations as supported by **RAST**, as the beams must be pointed to cover an area to enable call initiation from that area. Optimizing beam positioning in both planning and scheduling present interesting problems in computational geometry that also differ from network configuration problems addressed in standard ATM networks. Nielsen (Nielsen *et al.* 1997) addresses the problem of obtaining maximally efficient coverage given a set of antennas and regionally varying load requirements. This could enhance our approach to beam planning and beam migration in **RAST**.

Brown and Tong (Brown & Tong 1998; Tong & Brown 1998) address the problem of call admission control and QoS guarantees by means of reinforcement learning. Verma (S. Verma & Garcia 1998) approaches the problem by means of distribution and mathematical optimization. Both these approaches could be used to enhance the priority and policy table update mechanisms in **RAST**, although the advanced call scheduling of **RAST** is still necessary for a full solution to our problem.

Much research on load management addresses the problem of packet and call routing in telecommunication applications. Approaches include reinforcement

Rank	1	2	3	4	5	6	7	8	9	10
Contract	CBR 1	CBR 2	VBR 1	CBR 1	CBR 2	CBR 1	ABR 1	ABR 2	ABR 1	ABR 2
Priority	high	high	high	medium	medium	medium	high	high	medium	medium

Table 1: An example of the first several entries in a priority table. Priority rank is determined as a function of assigned priority and the QoS contract. That is, the number one ranked calls are high priority CBR 1, the second rank are high priority CBR 2 calls, and so on.

```
[Setting Beam Capacity for BEAM-2 to 15]
Handling network event at 5050.00 for <BEAM-2 18/15 in use (120.0%), 5 calls>
Migrating <CALL 6 :LOW_PRIORITY (13) :VBR_1 (4 s/s) :AREA_C to :AREA_D (BEAM-2)> to BEAM-1 at 5050.01
<BEAM-1 26/100 in use (26.0%), 9 calls>
Done handling network event at 5050.02 for <BEAM-2 14/15 in use (93.3%), 4 calls>
[several transactions elided]
Looking for 2 s/s on BEAM-1 for <CALL 26 :HIGH_PRIORITY (1) :CBR_1 (2 s/s) :AREA_A to :AREA_B (BEAM-1)>
Looking for 3 s/s on BEAM-2 for <CALL 12 :MED_PRIORITY (6) :VBR_1 (4 s/s) :AREA_C to :AREA_D (BEAM-1)>
Can't find 3 s/s on BEAM-2 to reclaim.
Shedding <CALL 12 :MED_PRIORITY (6) :VBR_1 (4 s/s) :AREA_C to :AREA_D (BEAM-1)> at 5077.96
<BEAM-1 26/30 in use (86.7%), 10 calls>
Accepted <CALL 26 :HIGH_PRIORITY (1) :CBR_1 (2 s/s) :AREA_A to :AREA_B (BEAM-1)> at 5077.97
<BEAM-1 28/30 in use (93.3%), 11 calls>
```

Figure 5: A trace of the run-time execution system which demonstrates call migration and shedding. CALL 16 is moved when network capacity changes, and CALL 26 is accepted after CALL 12 is shed.

learning (Boyan & Littman 1993; Tong & Brown 1998), market-based routing (Gibney & Jennings 1998), and ant-colony optimization (Bonabeau *et al.* 1998). Since our current work operates at a higher level of abstraction (call admission and modification, not packets and routing), this work could be plugged into our architecture in a modular fashion. It would be interesting to see whether the advanced reservations managed by **RAST** could be exploited by these routing mechanisms for performance improvements.

Much of the emphasis of QoS resource allocation involves reasoning about real-time cpu, bandwidth, latency, and jitter requirements, often in the presence of geometric constraints. Boddy (Boddy & Goldman 1994) addressed many of these issues in generating a scheduler to produce real-time schedules for the BOEING 777 aircraft. The CIRCA system (Musliner *et al.* 1993) also generates plans with real-time execution guarantees. Boddy and Musliner (Boddy 1996) describe a constraint-based distributed scheduling process for air traffic control. Each designated region of airspace is managed by a separate resource manager that allocates spatio-temporal windows to pilots requesting the resource. They applied similar ideas to task distribution and data volume management for distributed processing (Musliner & Boddy 1997).

Finally, several AI systems have been developed to support closed-loop plan execution. In contrast with **RAST**'s current plan execution component, the ap-

proach taken in 3T (Bonasso *et al.* 1997) has the planner watch over each step of execution. Hence the planner itself serves as an integral participant in the plan execution capability. Bresina (Bresina *et al.* 1996) describes APA, which has separate components for generation and execution of temporal plans, in which the executive is competent to carry out activity in the absence of plans, similar to the approach in **RAST**. Reece and Tate (Reece & Tate 1994) developed an execution agent for the O-Plan (Currie & Tate 1991) planning system. The combined system supports a plan repair mechanism (Drabble *et al.* 1996) that is more sophisticated than that supported by **RAST** at present, as it allows the planner to edit any unexecuted portion of the currently executing plan. Our redesigned plan execution component (Pell *et al.* 1998b) will support a similar editing capability, based on the work in O-Plan and also in Cypress (Wilkins *et al.* 1995). Finally, Lockheed's Tactical Planning and Execution System (TPES) (Mitchell 1997) is an interesting related system that supports many execution and replanning capabilities with a high level of human interaction.

7 Open Issues and Future Work

7.1 Open Issues

There are a number of open issues this domain has brought out. While we have addressed how to reconcile advanced reservations and dynamic requests within

an unpredictable environment, we have as yet to determine how our design scales up to a constellation of spacecraft. Traffic patterns and routing efficiencies are bound to affect the performance of the system. One interesting issue to explore would be to perform Machine Learning for load prediction and apply it to the Network Predicting (15) component. Determining schedule quality and ensuring that the PS generates a dispatchable schedule for the Exec (Muscettola *et al.* 1998a) are two other interesting tasks.

7.2 Future Work

What we have described in this paper is, in part, work in progress. We have developed the PS models and the Exec interfaces to most of the run time monitoring and execution software, and are running the Exec in a standalone mode with no planner input.

We are currently only demonstrating a modest scenario with 2 beams and 20 channels per beam, though we subsequently plan to increase the number of beams and hence the number of call requests this system can handle. In the near term we will be injecting various failure scenarios into both the plan-time and run-time environment (e.g. restricting the bandwidth because of jamming or atmospheric phenomena) and modeling the uplink and downlink segments separately. The latter would allow us to analyze throughput rates for each spacecraft which is acting as an ATM switch by changing the on board buffering capacity that each spacecraft provides.

8 Conclusion

We have reported here on a partially implemented architecture for managing satellite tele-communications network resources. We have used an approach which supports advance reservations and dynamic requests, negotiation and fulfillment of prioritized quality of service (QoS) contracts, graceful degradation in the presence of dynamic tasks and environmental changes, and optimization of geometrically constrained resources. Our integration of planning and execution addresses multiple types of quality-of-service contracts, with resource sharing (statistical multiplexing), preemption, and even reconfiguration (in the case of beam migration and repositioning).

We have explored an interesting integration of planning and execution, which combines several techniques, including plans as advice, coordination routines and task networks. Finally, our system can be used in routine operations or as a simulation-based design tool.

Acknowledgment We wish to thank Gregory Dorais of NASA Ames for useful technical discussions.

Scott Sawyer, Laura Plice, Tom Fall, Marilyn Golden and Gregory White of Lockheed Martin provided the necessary domain knowledge and the framework necessary for our understanding of this problem.

References

- Albayrak, S., editor (1998). *Intelligent Agents for Telecommunications Applications*. Lecture Notes in Computer Science. Springer-Verlag.
- Bernard, D. E., G. A. Dorais, C. Fry, E. B. G. Jr., B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. P. Nayak, B. Pell, K. Rajan, N. Rouquette, B. Smith, & B. C. Williams (1998). Design of the remote agent experiment for spacecraft autonomy. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO. IEEE.
- Boddy, M. S. (1996). Contract-based distributed scheduling for a next generation air traffic management system. Technical report, Honeywell Technology Center.
- Boddy, M. S. & R. P. Goldman (1994). Empirical results on scheduling and dynamic backtracking. In D. Atkinson, editor, *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS)*, Pasadena, California. Jet Propulsion Laboratory.
- Bonabeau, E., F. Henaux, S. Guerin, D. Snyers, P. Kuntz, & G. Theraulaz (1998). Routing in telecommunications networks with ant-like agents. In (Albayrak 1998).
- Bonasso, R. P., D. Kortenkamp, D. Miller, & M. Slack (1997). Experiences with an architecture for intelligent, reactive agents. *JETAI*, 9(1).
- Boyan, J. A. & M. L. Littman (1993). Packet routing in dynamically changing networks: A reinforcement learning approach. In J. D. Cowan, G. Tesauro, & J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 671–678. Morgan Kaufmann, San Francisco CA.
- Bresina, J., W. Edgington, K. Swanson, & M. Drummond (1996). Operational closed-loop observation scheduling and execution. In L. Pryor, editor, *Procs. of the AAAI Fall Symposium on Plan Execution*. AAAI Press.
- Brown, T. X. & H. Tong (1998). Reinforcement learning for admission control. In *Procs. of Snowbird 98*.

- Currie, K. & A. Tate (1991). O-plan: the open planning architecture. *Art. Int.*, 52(1):49–86.
- Dechter, R., I. Meiri, & J. Pearl (1991). Temporal constraint networks. *Art. Int.*, 49:61–95.
- Department of Defense, Space Command (1997). *Advanced Satellite Communications Capstone Requirements Document*.
- Drabble, B., A. Tate, & J. Dalton (1996). O-plan project evaluation experiments and results. Oplan Technical Report ARPA-RL/O-Plan/TR/23 Version 1, AIAI.
- Gibney, M. & N. Jennings (1998). Dynamic resource allocation by market based routing in telecommunications networks. In (Albayrak 1998).
- Hayzelden, A. & J. Bigham (1998). Heterogeneous multi-agent architecture for atm virtual path network resource configuration. In (Albayrak 1998).
- Jennings, N. R., E. Mamdani, J. M. Corera, I. Laresgoiti, L. F. Perriolallat, P. Skarek. & L. Z. Varga (1996). Using archon to develop real-world dai applications, part 1. *IEEE Expert*, 11(6):64–70.
- Mitchell, S. W. (1997). A hybrid architecture for real-time mixed-initiative planning and control. In *Procs. of AAAI-97*, pages 1032–1037, Cambridge, Mass. AAAI, AAAI Press.
- Muscettola, N. (1994). HSTS: Integrating planning and scheduling. In M. Fox & M. Zweben, editors, *Intelligent Scheduling*. Morgan Kaufmann.
- Muscettola, N. (1995). *HSTS Domain Description Language v1.2 User Manual*.
- Muscettola, N., P. Morris, & I. Tsamardinos (1998a). Reformulating temporal plans for efficient execution. In *Proc. of Sixth Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*.
- Muscettola, N., P. P. Nayak, B. Pell, & B. C. Williams (1998b). Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1/2). To Appear.
- Musliner, D. & M. S. Boddy (1997). Contract-based distributed scheduling for distributed processing. In *Working Notes of the AAAI Workshop on Constraints and Agents*, Providence, RI.
- Musliner, D., E. Durfee, & K. Shin (1993). Circa: A cooperative, intelligent, real-time control architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6).
- Nielsen, F., P. Calégari, F. Guinec, & P. Kuonen (1997). Combinatorial optimization algorithms for radio network planning. In *Proc. 10th Franco-Japanese Franco-Chinese Conf. Combin. and Comp. Sci.* Palaiseau, FRANCE.
- Pell, B., D. E. Bernard, S. A. Chien, E. Gat, N. Muscettola, P. P. Nayak, M. D. Wagner, & B. C. Williams (1998a). An autonomous spacecraft agent prototype. *Autonomous Robots*, 5(1).
- Pell, B., G. A. Dorais, C. Plaunt, & R. Washington (1998b). The remote agent executive: Capabilities to support integrated robotic agents. In A. Schultz & D. Kortenkamp, editors, *Procs. of the AAAI Spring Symp. on Integrated Robotic Architectures*, Palo Alto, CA. AAAI Press.
- Reece, G. & A. Tate (1994). Synthesizing protection monitors from causal structure. In *Procs. AIPS-94*. AAAI Press.
- S. Verma, R. K. P. & A. L. Garcia (1998). Call admission and resource reservation for guaranteed qos services in internet. *Computer Communications Journal*, 21(4). Special issue on Building Quality of Service into Distributed Systems.
- Simmons, R. (1990). An architecture for coordinating planning, sensing, and action. In *Procs. DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 292–297, San Mateo, CA. DARPA, Morgan Kaufmann.
- Tong, H. & T. X. Brown (1998). Optimizing admission control and routing while ensuring quality of service in multimedia networks via reinforcement learning. In *IEEE Infocom'99*, New York. Submitted.
- Varma, A. (1997). Tutorial on Traffic Management in ATM Networks. In *MILCOM 1997*.
- Wilkins, D. E., K. L. Myers, J. D. Lowrance, & L. P. Wesley (1995). Planning and reacting in uncertain and dynamic environments. *JETAJ*, 7(1):197–227.

SATELLITE TELE-COMMUNICATIONS SCHEDULING AS DYNAMIC CONSTRAINT SATISFACTION

<p>Christian Plaunt Caelum Research Corporation NASA Ames Research Center Mail Stop 269-2 Moffett Field, CA 94035, USA phone: +1 650 604 2928 fax: +1 650 604 3594 plaunt@ptolemy.arc.nasa.gov</p>	<p>Ari K. Jónsson RIACS NASA Ames Research Center Mail Stop 269-2 Moffett Field, CA 94035, USA phone: +1 650 604 2799 fax: +1 650 604 3594 jonsson@ptolemy.arc.nasa.gov</p>	<p>Jeremy Frank Caelum Research Corporation NASA Ames Research Center Mail Stop 269-2 Moffett Field, CA 94035, USA phone: +1 650 604 2524 fax: +1 650 604 3594 frank@ptolemy.arc.nasa.gov</p>
--	---	---

ABSTRACT

We consider a single satellite telecommunications problem in which a dynamic set of calls must be assigned to beams of the satellite. These assignments must satisfy beam-coverage constraints, capacity constraints and requirements based on the priorities of incoming calls; additionally, the satellite must respond quickly to the changing call load and environment. We show how this problem can be solved by using constraint satisfaction technology. We model the problem as a Dynamic Constrained Optimization Problem (DCOP) and present an algorithm inspired by hill-climbing search. We present empirical results from a simulation showing that the algorithm meets the requirements imposed by the problem domain, and finds solutions that are within 2% of the optimal.

1. INTRODUCTION

Advances in satellite communications technologies have given rise to new challenges for automated allocation techniques for dynamic resources such as satellite and bandwidth availability. In this paper, we look at a particular class of satellite tele-communications scheduling problems, and present an approach for solving those problems effectively within strict real-world time limits. The approach is based on viewing the satellite communications scheduling problem as a Dynamic Constraint Optimization Problem (DCOP). This well-known class of automated reasoning problems provides us with a well-defined framework and a number of possible approaches to solving such problems.

We present an automated dynamic scheduling method based on an encoding of this problem as a dynamic constrained optimization problem. The algorithm we selected to solve this problem was inspired by the local search paradigm, which provides good solutions to such problems in real-time. Experimental results show the scheduling method to be very effective at

finding near-optimal solutions, even in the face of degraded communications capabilities.

Our problem domain consists of a single satellite with b communication links or beams, which cover a set of g ground stations. Each call is assigned a priority, and there are p priorities available. The satellite must support a dynamically changing set of call requests. Each call request consists of a source ground terminal, a destination ground terminal, a number of units of bandwidth, and a priority. A call must be assigned to both an uplink beam and a downlink beam or be rejected. The uplink beam must cover the source ground terminal and the downlink beam must cover the destination ground terminal. Each beam has a maximum capacity for uplinked and downlinked calls. That is, the total bandwidth of uplinked calls on a beam may not exceed a capacity c_u , and the total bandwidth of downlinked calls on the same beam may not exceed a capacity c_d . Furthermore, calls of priority p_i outweigh all calls of strictly lower priority; if there is not enough space capacity on a beam for an incoming call of high priority, the system is required to disconnect enough calls of lower priority to accommodate the new, higher priority call, if possible. Each call has a finite but unknown duration, so calls periodically are released, thereby freeing more capacity for new calls. Calls arrive at arbitrary intervals. A call does not change priority, source or destination station while it is in progress. However, the capacity of beams may increase or decrease, forcing some calls to other beams or requiring the premature termination of some calls.

The problem of managing calls on the satellite consists of the following:

1. When a new call arrives, the satellite must decide whether to accept the call or reject it; if the call is accepted, the satellite must decide which beams the call will utilize.
2. When the capacity of a beam changes, the satellite must decide whether to move or terminate any

calls, and if so, which ones.

If we consider the satellite at any instant, there is a list of call requests in the system. The problem is to decide which calls to accept and which to reject, and then to assign each call uplink and downlink beams in such a way that each call's coverage requirements are satisfied and the capacity constraints on the satellite beams are met. A problem of this type can be encoded as a *Constraint Satisfaction Problem* or CSP. Informally, a CSP consists of a set of variables, a description of the possible values each variable can take on, and a list of constraints which define valid assignments to sets of variables. The problem we have described also includes a preference for assigning calls of high priority to the satellite. Adding such a preference order among solutions satisfying the constraints results in a *Constrained Optimization Problem* or COP. CSPs and COPs have been heavily studied, and many theoretical and practical results can be brought to bear to address such problems; for work on CSPs in general see Haralick & Elliot (1980), Nadel (1989), and for a specific application see Banerjee & Frank (1996). However, as calls arrive and depart, we have not just one but a *sequence* of such problems. These problems are closely related, as each problem in the sequence is derived from an earlier problem by the termination of an existing call, the addition of a new call, or the reduction in capacity on a beam. A modification of CSPs known as *Dynamic Constraint Satisfaction Problems* or DCSPs can be used to encode the sequence of problems which results from the arrival and departure of call requests.

There are a number of methods for solving CSPs and DCSPs. However, in this domain, the solver must meet performance requirements imposed by the telecommunications application, as the satellite must be able to respond rapidly to new call requests as well as to changes in the available bandwidth on the beams. The solver must nonetheless provide good solutions (i.e. allocate high priority calls) which are also valid (i.e. meet the coverage constraints and do not exceed beam capacity). In addition, the fact that we are presented with a sequence of closely related problems suggests that any algorithm to solve the sequence of problems *reuse* the solution to the previous problem to increase the speed of the solver. *Hill-climbing* algorithms for CSPs operate by perturbing solutions in order to find better solutions which are nearby. These methods have good problem solving performance in general Selman *et al.* (1992), Minton *et al.* (1990), and also promote the reuse of solutions between successive problems in the DCSP framework Freuder & Wallace (1998).

The rest of the paper is organized as follows. In Section 2 we formally define CSPs and DCSPs and discuss methods of solving these problems, including hill-

climbing methods. In Section 3 we formally describe the satellite telecommunications problem as a Dynamic Constrained Optimization Problem (DCOP). In Section 4 we present a hill-climbing algorithm for responding to new connection requests and reductions in the capacity of beams. We also give a bound on the complexity of this procedure, and establish that it can respond to changes in real-time. In Section 5 we present the results of applying the hill-climbing algorithm to a set of telecommunications requests taken from a real satellite telecommunications problem. We show that hill-climbing can consistently find solutions within 2% of the best possible solution. In Section 6 we conclude and discuss some opportunities for future work.

2. DYNAMIC CONSTRAINT SATISFACTION AND OPTIMIZATION PROBLEMS

In this section we present the formal machinery we will use to solve the satellite telecommunications problem. We shall first formally define CSP, COP, DCSP and DCOP, then discuss hill-climbing algorithms to solve these problems.

2.1 CONSTRAINT SATISFACTION PROBLEMS

A *Constraint Satisfaction Problem* or CSP is a triple $P = (V, D, C)$, where:

1. $V = \{v_1, \dots, v_n\}$ is a set of variables
2. $D = D_{v_i} \mid i \in \{1, \dots, n\}$ are the domains of the variables, where each D_{v_i} is a finite set of possible values of v_i .
3. C is a set of constraints (Y_j, R_j) , where each constraint consists of a scope $Y_j = \{v_{i_1}, \dots, v_{i_k}\} \subseteq V$ and a relation $R_j \subseteq \prod_{\nu=1}^k D_{v_{i_\nu}}$.

It is worth mentioning that if the domains D_{v_i} are large and the scope contains many variables, then explicitly enumerating the relations of the constraints may be quite cumbersome. Consequently, relations are often written in a condensed form. For instance, if the variable domains are subsets of the integers, we can write relations as equations such as $C_1 = \{(x, y), x + y < 5\}$ rather than enumerating all the legal pairs of values of x and y .

A *valid solution* to a constraint satisfaction problem $P = (V, D, C)$, where $V = \{x_1, \dots, x_n\}$, is an n -tuple $(v_{x_1}, \dots, v_{x_n})$, such that:

1. $v_{x_k} \in D_{x_k}$ for $k = 1, \dots, n$, and
2. For any $(Y, R) \in C$ with $Y = \{x_{i_1}, \dots, x_{i_k}\}$, we have $(v_{x_{i_1}}, \dots, v_{x_{i_k}}) \in R$.

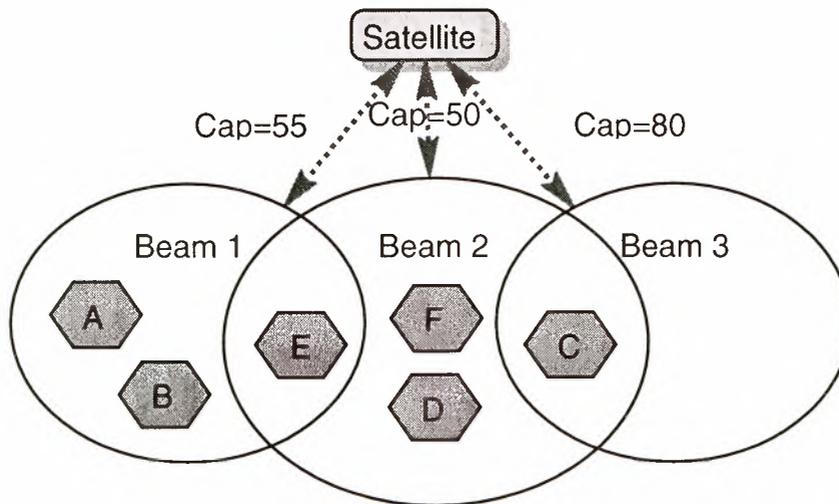


Figure 1: A network consisting of one satellite with three uplink/downlink beams. Beam 1 of capacity 55 covers stations A, B and E, beam 2 of capacity 50 covers stations D,E,C and F, and beam 3 of capacity 80 covers station C.

2.2 CONSTRAINT OPTIMIZATION PROBLEMS

Many problems consist of both constraints and an *optimization criteria* which differentiates between valid solutions. For example, in our telecommunications domain, an allocation of calls to the beams of the satellite may satisfy the constraints, but we prefer assignments which assign more high priority calls to the satellite. To formalize this concept, we define a *Constrained Optimization Problem* or COP as a pair (P, g) where P is a constraint satisfaction problem and g is a function that maps every valid solution of P into \mathbb{R} . The goal of constraint optimization is to find a valid solution that maximizes g .

As an example of a COP, let us consider a simple scheduling problem consisting of two low-capacity beams and a few call requests (disregarding geographic constraints for simplicity). The calls must be assigned to beams or rejected, and the constraints limit the overall bandwidth requirements for all calls assigned to a beam. Finding a satisfactory assignment of calls to beam 1, beam 2 or rejection, is a CSP. If we now specify a preference among solutions, or equivalently, specify an optimization function on the set of valid assignments, we have a COP. In other words, the CSP defines a set of valid call assignments, and the preference functions defines an ordering among solutions.

2.3 DYNAMIC CONSTRAINT SATISFACTION PROBLEMS

As mentioned earlier, the satellite telecommunications problem is not a CSP, since calls are constantly being added and deleted. However, the changing set of calls can be represented as a sequence of closely related

CSPs. To formalize this notion, let $P = (V, D, C)$ be a constraint satisfaction problem. Any problem of the form $Q = (V', D', C')$ such that $V' \supseteq V$ (i.e. there are more variables), $D'_v \subseteq D_v$ for each $v \in V$ (i.e. there are fewer legal values for variables) and $C' \subseteq C$, (i.e. there are fewer legal combinations for variables in a constraint) is a *restriction* of P . Any problem of the form $Q = (V', D', C')$ such that $V' \subseteq V$ (i.e. there are fewer variables), $D'_v \supseteq D_v$ for each $v \in V$ (i.e. there are more values for variables) and $C' \supseteq C$ (i.e. there are more legal combinations for variables in a constraint), is a *relaxation* of P . A *Dynamic Constraint Satisfaction Problem* or DCSP is a sequence of constraint satisfaction problems P_0, P_1, \dots , such that each problem P_i is either a *restriction* or a *relaxation* of P_{i-1} . This definition is consistent with similar definitions given in Dechter & Dechter (1988) and Verfaillie & Schiex (1994).

Not surprisingly, it is relatively straightforward to generalize the idea of dynamic constraint satisfaction to dynamic optimization problems. Formally, a *Dynamic Constrained Optimization Problem* or DCOP is a sequence of optimization problems, such that each entry is a relaxation or a restriction of the previous problem. This means that the optimization function remains unchanged throughout, but the set of variables, domains and constraints may change.

2.4 SEARCH METHODS AND HILL-CLIMBING

There are two main families of procedures for solving CSPs and COPs. *Complete* methods are guaranteed either to find a valid assignment of values to variables or prove that no such assignment exists. Complete methods frequently exhibit good performance, and guarantee

a correct and optimal answer for all inputs. Unfortunately, they require exponential time in the worst case, which is not acceptable for the satellite telecommunications domain.

Recently, researchers have become interested in *incomplete* search methods which do not guarantee correct answers for all inputs. These methods can find satisfying assignments for solvable problems with high probability. These incomplete algorithms have gained popularity in recent years, due to their simplicity, speed and observed effectiveness at solving certain types of problems.

Hill-climbing is one of the most popular incomplete approaches to solving constraint satisfaction problems. These algorithms map assignments to a set of assignments by making minor changes to the original assignment. Each element of the set is evaluated according to some criteria designed to move closer to a valid assignment and/or improve the evaluation score of the state. The best element of the set is made the next assignment. This basic operation is repeated until either a solution is found or a stopping criteria is reached. A hill-climbing algorithm requires two components: a *candidate generator* which maps one solution candidate to a set of possible successors, and a *evaluation criteria* which ranks each valid solution (or invalid full assignments), such that improving the evaluation leads to better (or closer to valid) solutions.

To take a concrete example of hill-climbing, consider the following scenario for a slightly unrealistic satellite telecommunications problem. We have a satellite with only one beam and one station. Assume we have assigned a call of priority 5 requiring a bandwidth of 2, to a single given beam with capacity of 4 units. Two calls are currently rejected, one with priority 3 and bandwidth requirement of 3, and the other with priority 7 and bandwidth requirement of 1. This current solution could be described as $(\{C_{5,2}\}, \{C_{3,3}, C_{7,1}\})$, with the first set being calls assigned to the beam and the second set consisting of rejected calls.

Let us then choose a simple optimization function, which sums up $(10 - p) \cdot b$, where p is the priority (1 highest, 8 lowest) and b is the bandwidth used. Our current solution then evaluates to $(10 - 5) \cdot 2 = 10$. Our successor function might then give the following options:

$$(\{C_{3,3}\}, \{C_{5,2}, C_{7,1}\})$$

which evaluates to 21, and

$$(\{C_{5,2}, C_{7,1}\}, \{C_{3,3}\})$$

which evaluates to 13. We therefore pick the first candidate as the new current solution.

A second hill-climbing iteration would then result in

$$(\{C_{3,3}, C_{7,1}\}, \{C_{5,2}\})$$

which is indeed an optimal solution at 24.

Hill-climbing algorithms do not always find optimal solutions for real problems. However, hill-climbing methods have the distinct advantage that they can often provide a valid solution at any time-point. This makes the technique very suitable for systems that must perform with real-time guarantees. An added bonus is that the more time the hill-climbing process is given, the better the solution will typically be. Gent & Walsh (1993) Finally, hill-climbing is especially attractive for DCSPs, because it is likely that the solution to problem C_i is a good starting assignment for problem C_{i+1} . Freuder & Wallace (1998) For these reasons, we chose to base our solution to the satellite telecommunications problem on hill-climbing.

3. THE PROBLEM AS A DCOP

As mentioned earlier, our problem domain consists of a single-satellite, multiple ground-station communications network with variable connection coverage and varying bandwidth due to technical glitches, maintenance and other factors. In order to handle a communications request, we need to allocate sufficient bandwidth from the source terminal node to the central node, along a link that covers that node, and from the central node to the destination node, along a link covering the destination.

Let us assume that we are given a single satellite communication assignment problem with n calls, b beams, g ground stations, p priorities and s ground stations. Each beam has capacity of d_i slots for downlink, u_i slots for uplink.

A satellite communication assignment problem is defined by:

- $\{t_1, \dots, t_g\}$, a set of ground stations
- $\{l_1, \dots, l_b\}$, a set of links between a set of ground stations and the satellite
- $\text{cap}_u(l_i)$, function identifying the uplink capacity of each link
- $\text{cap}_d(l_i)$, function identifying the downlink capacity of each link
- $\text{cover}(l_i, t_j)$ a predicate indicating whether beam i covers the location of ground station j
- $\{c_1, \dots, c_n\}$, set of calls, including start times and durations.
- $p(c_i)$, function giving priority of call
- $\text{use}(c_i)$, function indicating bandwidth required for call

- $\text{source}(c_i)$, indicates the terminal source of call
- $\text{dest}(c_i)$, indicates the terminal destination of the call

To describe our problem as a DCOP, we first determine what our variables and values are. The key decisions are how each call is routed, i.e. which beams the uplink and downlink are assigned to. We define two variables u_i and d_i for each call that is in the system, the uplink-beam and the downlink-beam. The values for these variables include all possible beams, but we also need a value to represent that a call is rejected. So, for each call variable u_i or d_i , we have a set of links that we can assign to it, $\{l_1, \dots, l_b\}$, and a flag indicating that the call is rejected. To facilitate the specification of this problem, let us represent this domain as the numbers from 0 to b , with 0 standing for the rejection flag and $i \in (1, \dots, b)$ standing for (l_1, \dots, l_b) respectively.

The constraints that must be satisfied are the following:

- For each link l_i ,

$$\sum_{j=1}^n I(u_j = i)(\text{use}(c_j)) \leq \text{cap}_u(b_i)$$

- For each link l_i ,

$$\sum_{j=1}^n I(d_j = i)(\text{use}(c_j)) \leq \text{cap}_d(b_i)$$

- For each call c_i , if $u_i = j$ then either $j = 0$ or $\text{cover}(b_j, \text{source}(c_i))$
- For each call c_i , if $d_i = j$ then either $j = 0$ or $\text{cover}(b_j, \text{dest}(c_i))$
- For each call c_i , $u_i = 0$ if and only if $d_i = 0$

where I is the indicator function. The optimization function is defined as follows:

$$g(\langle (c_1, b_{j_1}), \dots, (c_n, b_{j_n}) \rangle) = \sum_{i=0}^n p(c_i) I(u_i = 1)$$

The problem is dynamic in that calls arrive and are accepted or rejected; calls are terminated or completed; and the beam capacity changes. In terms of DCOPs, the relaxations that can occur are:

- The two variables corresponding to an existing call are deleted from the problem along with all associated constraints. This occurs if a call is either rejected, terminated or completed.
- The capacity of a link increases.

The restrictions that can occur are:

- Two new variables corresponding to a new call are added to the problem along with all associated constraints.
- The capacity of a link decreases.

4. SOLVING THE DCOP USING LOCAL SEARCH

We are now ready to describe our solution to the satellite telecommunications DCOP. When a relaxation occurs, we do nothing; the solution to the previous problem is always adequate when the problem is relaxed. There are two categories of restrictions in this problem: call arrival and capacity reduction. When a new call arrives, it is assigned to a pair of beams which have the appropriate coverage and have the most remaining capacity. The resulting assignment may overload one or both beams, which also happens when the capacity of a beam is reduced. Consequently, the main issue is moving or terminating calls in such a way that we preserve the high priority calls and no beams are overloaded.

We solve the problem of reassigning calls on overloaded beams using hill-climbing. Recall that a hill-climbing algorithm requires two components: a candidate-generation component to take an initial assignment and generate new assignments, and an evaluation function which ranks the new assignments. The best of the candidate assignments according to the objective function is then selected as the new assignment. We first discuss these two components then show how the algorithm works as a whole.

We generate new candidates by trying to move the lowest priority calls on overloaded beams. Let L be the set of all of the lowest priority calls which could be moved to relieve the capacity of any overloaded beam, and let m be the highest priority call in L . (Should there be several calls of the same priority, assume that each call has a unique identifier and pick the one with the smallest id.) Now let B be the set of beams satisfying the coverage requirements for this call. The candidates that are generated consist of moving the call to each of these beams in turn or rejecting the call. Notice that there is always at least one option because we can always reject a call.

Now let us see how we compare the candidates. Our preference is to keep calls of high priority; for the objective function we interpret each priority as an integer and sum the priorities of calls which are assigned to beams. We do this by counting the highest priority calls on each beam until the capacity is reached. Under this scheme, the priorities of calls must be chosen so that calls of lower priority are appropriately comparable to calls of higher priority; for instance, if 2 calls of priority p are

worth more than 1 call of priority $p+1$ then $p+1 < 2p$. Additionally, we do not count a call if either its uplink or downlink is on an overloaded beam and is of low enough priority that it might be terminated.

In some cases the candidates can all have the same rank. If the call being moved has a priority low enough that it would not stay on any beam, we reject the call to remove it from the system. In any other situation, we randomly select one of the best candidates as the new configuration.

We now have an operation which determines what to do with a single call on an overloaded beam. The call is moved to another beam if it is of high enough priority to displace calls of lower priority or if there is excess capacity. Otherwise, the call is terminated. We execute this operation repeatedly until no beams are over capacity any longer. Figure 2 shows the sketch of the full algorithm, which we call *load-balance*.

```

procedure load-balance()
  O = set of overloaded beams
  while O is not empty
    L = set of calls to be bumped from O
    m = highest priority call from L
    B = set of beams m can move to
    for b in B
      if b satisfies coverage requirements of m
        rank moving m to b
    if rank of terminating m == rank of best move
      terminate m
    else make best move
  update O
end # while
end

```

Figure 2: The load-balance hill-climbing algorithm.

The *load-balance* procedure may be called many times, since moving a call may exceed the capacity of some other beam, and several calls may be required to reduce the capacity on a single, heavily overloaded beam. We now provide a worst-case complexity of the number of times *load-balance* will be called in order to satisfy the capacity constraints on all the beams. Let C be the total number of calls in the system at the time that load balancing occurs. We shall show that no call is ever handled by the procedure more than once. If *load-balance* rejects a call it is never manipulated again, so let us consider what happens if the procedure moves part of a call c_i . Recall that c_i is the highest priority call of all the calls which must be moved from any overloaded beam. Because c_i is moving and not being terminated, we know that either there is space on the destination beam, or some other calls can be moved from

the destination beam. But all of these calls are of strictly lower priority than c_i . Therefore, no call moved after c_i can displace c_i from its new home. Consequently, in the worst case, each component (uplink and downlink) of a call would have to move once. Since each call is either terminated or each component of a call is moved only once, *load-balance* is called fewer than $2C$ times.¹ Also notice that the complexity of any single call manipulation is $O(b)$ since there are b beams and each call manipulation must consider all beams in the worst case. So overall the algorithm requires $O(Cb)$ elementary operations.

We now return to the issue of call acceptance. When a new call arrives, the satellite must decide whether to accept or reject the call. The procedure to do this requires first finding a pair of uplink-downlink beams which satisfy the coverage requirements. We tentatively assign the call to those beams with the highest capacity path, and then call *load-balancing*. Once load balancing is done, if the new call request is part of the assignment, then the call is accepted and all changes required to realize the new solution are taken. If the new request is *not* part of the new solution, which can happen when one of the beams is at or near capacity and the new request is of low priority, the call is denied.

5. EMPIRICAL RESULTS

In this section we present the results of a simulation using call requests from a real telecommunications application. The scenario we experimented with uses a satellite with 2 beams covering 5 ground stations. Three of the stations are covered by both satellites, and the remaining two stations are covered by only 1 satellite each. Calls have 8 possible priorities, with priority 1 calls the highest. As the simulation proceeded, we ratcheted down the capacity of the uplink and downlink capacity of both the beams from 1000 to 400 units of bandwidth over the course of 1000 seconds. This scenario was designed to show that, as the capacity of this beam changes, our hill-climbing algorithm terminates calls in lowest-first priority order.

We encoded the problem as a DCOP as we have indicated in the previous sections of the paper. In addition, we carefully crafted an objective function in order to ensure that calls of the highest priority stayed on the beams as capacity changed.

Figure 3 shows a graph of bandwidth usage by relative priority² over time on one of the downlink beams during a test run of 7769 dynamic call requests over a period of 33 minutes (an average of about 3.9 calls per

¹We can put a tighter bound on the number of manipulations at runtime, but space prohibits us from including these results.

²These are not the priority values used to compute the objective function during search.

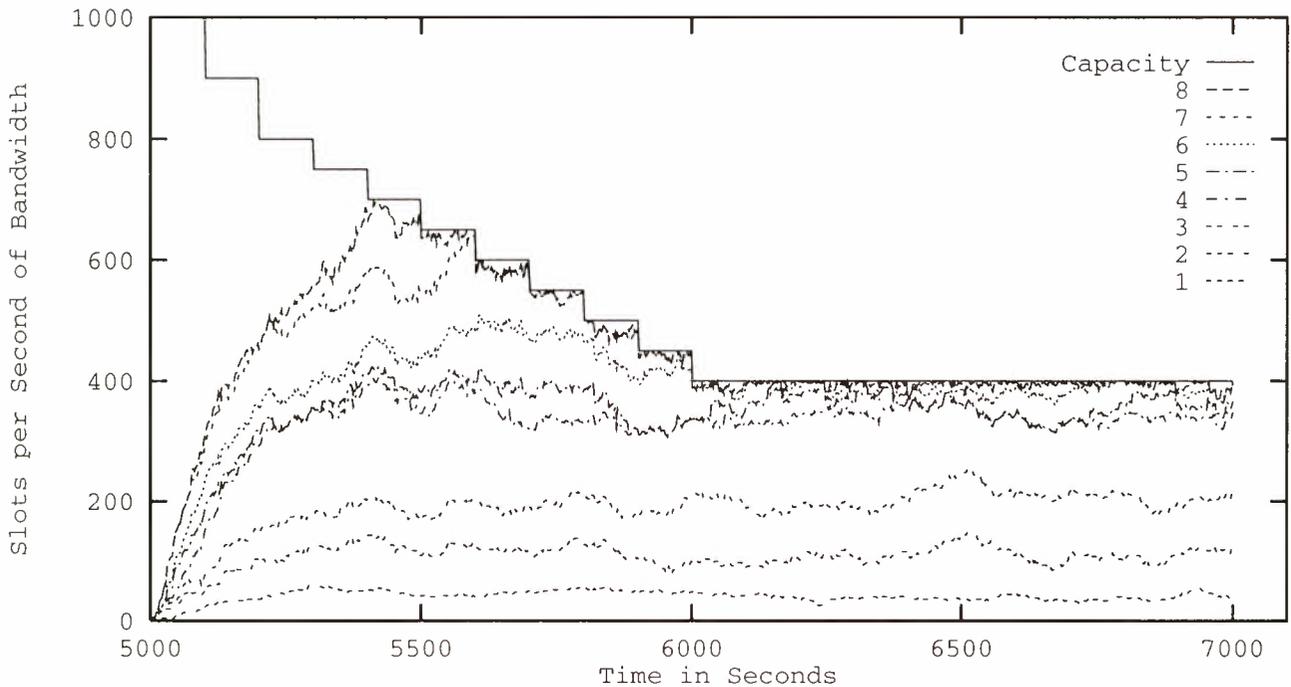


Figure 3: A graph of bandwidth capacity and usage by priority over time on a downlink beam. The highest priority calls (1) are at the bottom, with successive lower priority call in increasing order above (2-8), total available bandwidth capacity at the top.

Time	% Opt. Score	% Opt. Bandwidth
5450	99.98	97.37
5550	99.96	98.29
5650	99.76	95.69
5750	99.88	97.95
5850	99.67	95.99
5950	99.69	96.89
6050	99.27	93.50
6200	99.80	99.50
6500	99.33	99.00
6700	98.14	100.00

Figure 4: Closeness of the solution found by load-balance to the optimal solution as simulation time increases.

second). Note that the highest priority calls are graphed at the bottom, with successive ranks of lower priority calls in increasing order above. The top-most (solid) line shows the total available bandwidth capacity of the beam during the run. As we see from the figure, the calls of low priority are terminated to keep the high priority calls on the system, as desired.

As we have said before, hill-climbing algorithms may not always find the optimal solution to a COP. To test how close we came to the optimal solution, we created

the COP induced at a particular time instant of the simulation and found the optimal solution. Figure 4 shows the results. The percentage is the value of the solution found by the load balancer over the optimal value. We see that throughout the simulation the hill-climbing algorithm was able to consistently find solutions within 2% of optimal.

We also analyzed the bandwidth use achieved by the hill-climbing load balancer. Figure 4 also shows the percentage of optimal bandwidth achieved by the load balancer. We see that the load balancing algorithm consistently uses more than 94% of the bandwidth used by the optimal solution.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced the satellite telecommunications, and showed how it can be modeled as a Dynamic Constrained optimization Problem. We then designed and implemented a hill-climbing algorithm to solve the problem. Our empirical results show that hill-climbing is capable of solving this problem very well, as it is consistently able to achieve within 2% of the best possible solution.

There are several future directions for this work. One direction is to consider the impact of moving beams. As beams change position, ground station coverage patterns will change, introducing a new set of relaxations

and restrictions. The framework we have described is adequate to address this type of dynamism, and we believe that the results will be as good as those presented here. Another research question involves increasing the number of beams in the simulated satellites. More beams and more ground stations may result in larger search problems, thereby causing more difficulties as the hill-climbing search must work harder to solve the problem instances that arise over time. A third option is to consider situations with multiple satellites. This problem presents a challenge to modeling via DCOPs, because now a call must take several "hops" to get from its source ground station to its destination ground station. In theory, this should be addressable simply by adding some more variables to the problem, but in practice it may prove more difficult.

Another possible research direction is to consider using other algorithms to solve this problem. In the previous section we used a complete search algorithm with hand-crafted heuristics to generate the optimal solutions to the optimization problem induced at fixed timepoints. We found that while it would often take this program a long time to find the optimal solution, it frequently found good solutions early. This raises the prospect of using other algorithmic techniques for constraint satisfaction to address this problem.

REFERENCES

- Banerjee, D. & J. Frank (1996). Constraint satisfaction in optical routing for passive wavelength route networks. In *Proceedings of the 2nd International Conference on the Principles and Practices of Constraint Programming*, pages 31–45. Springer Verlag. Lecture Notes in Computer Science.
- Dechter, R. & A. Dechter (1988). Belief maintenance in dynamic constraint networks. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 37–42, Palo Alto, CA. Morgan Kaufmann.
- Freuder, E. & R. Wallace (1998). Stable solutions for dynamic constraint satisfaction problems. In *Proceedings of the 4th International Conference on the Principles and Practices of Constraint Programming*, pages 447–461. Springer Verlag. Lecture Notes in Computer Science.
- Gent, I. P. & T. Walsh (1993). Empirical analysis of search in GSAT. *Journal of Artificial Intelligence Research*, 1:47–59.
- Haralick, R. & G. Elliot (1980). Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263–313.
- Minton, S., M. D. Johnston, A. Phillips, & P. Laird (1990). Solving large scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 17–24. AAAI Press.
- Nadel, B. (1989). Constraint satisfaction algorithms. *Computational Intelligence*, 5:188–224.
- Selman, B., H. Levesque, & D. Mitchell (1992). New method for solving hard satisfiability problems. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 440–446. AAAI Press.
- Verfaillie, G. & T. Schiex (1994). Solution reuse in dynamic constraint satisfaction problems. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 307–312, Cambridge, MA. MIT Press.

ADAPTIVE RESOURCE PROFILING

Dennis DeCoste

Machine Learning Systems Group

Jet Propulsion Laboratory / California Institute of Technology

4800 Oak Grove Drive; Pasadena, CA, 91109; USA

<http://www-aig.jpl.nasa.gov/home/decoste/decoste@aig.jpl.nasa.gov>

Abstract

This paper addresses the problem of learning *resource profiles*: upper and lower prediction bounds for engineering resources (e.g. power). We argue for data-driven techniques which specifically learn interval-valued predictions (i.e. best and worst-case bounding functions), as opposed to probabilistic (i.e. soft) predictions. We present and discuss a simple preliminary example using actual data from a rover prototype in a laboratory test-bed environment. We discuss extensions to this work, including integration with an onboard planner that could use these resource profiles, toward improving overall science throughput.

1 Introduction

This work addresses the problem of automated learning and updating of resource models (e.g. battery power availability) using sensor data. Robust spacecraft/rover autonomy requires the ability to maintain resource models onboard, to reflect changing environmental and degrading spacecraft conditions with minimal ground attention. Furthermore, future planned missions and economic constraints present increasing pressures to deal with largely unknown environments and short design-build-launch cycles (with minimal time for rigorous testing). Thus, careful pre-flight manual preparation of resource models is likely to be infeasible and inadequate.

The traditional ground-based approach (send all data to ground, perform trending and statistical modeling manually, update models) is both suboptimal and impractical. It results in reduced science throughput, due to both spacecraft-ground communication delays and the need to use excessively conservative resource margins. Furthermore, for key future mission contexts such as multiple cooperating rovers,

spacecraft fleets, and Deeper Space missions (such as planned Pluto flybys), the telemetry bandwidth requirements and/or communication delays would be enormous.

1.1 Adaptive Resource Profiling

To address such problems, we have developed machine learning and data mining techniques to both learn initial resource models from historic sensor data (e.g. testbeds, simulations, early mission behavior) and to continually adapt them using online sensor data. Specifically, we have adapted our earlier work in abnormality/fault detection via learning red-line envelope functions ([1,2]) to the task of *resource profiling*: learning upper and lower bounds on expected future resource availability over time.

Each profile projects how much of a resource may be available over future time points, based on the current resource level and on the durations of actions which can produce (e.g. activate solar panel) or consume (e.g. turn on motor) the target resource (e.g. battery power). To reflect uncertainty in the impact of such actions (due to both unobserved-yet-contextually-significant effects and routine sensing noise), these predictions can be based on learned context-sensitive interval-valued (rather than nominal mean-valued) estimates of the production and consumption rates of such actions.

The end result is an envelope profile showing the best-case and worst-case resource availability over time. Such profiles are useful both for plan execution monitoring (i.e. when actions are observed) and planning/scheduling (when future actions are planned). These models allow reasoning under both best and worst-case scenarios, to guide aggressive attempts toward maximum science throughput while avoiding controlling dangerously close to worst-case limits (e.g. heading into night-fall without sufficient battery charge to run critical night-time operations or experiments).

Details for some techniques to learn bounding func-

tions from data can be found in ([1],[2]). The key property of these techniques is that they: 1) result in few false alarms (e.g. they properly contain all the data within the resulting bounding intervals) while otherwise striving to be as tight as possible and 2) overcome key limitations of other data-driven alternatives. For example, the common approach of *error bars* (e.g. neural network predictions of means and variances) make strong assumptions about the nature of the prediction error distribution (e.g. symmetric Gaussian noise). More general non-parametric *probably density estimation* overcomes that problem, but tends to be very “data hungry” and spends significant effort modelling the nature of the data between the extrema values.

In contrast, our bounding techniques essentially view the problem as a form of constrained optimization: make predictions which are as close to the target (e.g. in the least squared error sense) while ensuring that those predictions are always above (or below, in the case of low bounds) the target values. Our technique does not spend effort modelling the entire probability spread (only predicting the context-sensitive extrema values), nor does it make strong assumptions about the nature of the prediction error distribution, except that the maximum error is bounded (i.e. finite range between the tails of the distribution), as is typically the case in practice for digitally-sampled engineering data.

2 Example: Mars Rover Battery Drain

As a concrete example, consider a key resource for a Mars rover: power. Solar panels provide power, loads (e.g. motors, cameras) consume it, and the battery stores it. There is uncertainty in the rate at which the solar panels recharge the battery, due to conditions such as dust accumulation and Sun position. The loads also have uncertain consumption, due to variabilities that existing sensors are inadequate to capture. In our experiments (running the Rocky 7 rover prototype in the JPL Mars Yard), the possible training inputs are: 1) sensed quantities such as battery voltage, wheel motor torques and currents, and solar intensity and 2) the times of various actions (such as turning on/off cameras, which do not have their own sensors of currents).

From actual sensed data of such inputs over time, our system learns to predict bounds on the battery power at any given time $T+\Delta T$ into the future. In our experiments with Rocky 7 so far, these predictions are based on the current battery voltage (and other selected sensors) at time T , as well as some fixed lagged time values in the past (e.g. at $T-L_1$, $T-L_2$,

...). In practice, a prediction target of the remaining kilo-watt/hours of power (instead of voltage levels per se) is more meaningful. This requires computing backwards from a final (0 KW/hr) battery dead state, computation of load watt requirements (i.e. from observed current and duration data), and integrating to compute target values of “remaining power” over each sensed time point. For simplicity, the experiments discussed below focus on predicting the voltage level.

2.1 Example Performance

Figure 1 shows the training data, consisting of 23 sensors over about 7.5 hours. This data was gathered over six independent trials of Rocky 7, under various load and solar conditions, and combined into one time-series data-set.¹ Each trial was run from a full battery charge until the battery power dropped so low that the CPU and data sampling shut down. The solar panel on the Rocky 7 prototype is actually insufficient to recharge the battery, even with no loads other than CPU; so, it merely slows down the power drain rate. Thus, the plot of the battery voltage sensor (labelled **MezVoltage-batteries**) shows 6 distinct periods of high-to-low voltage drop, one for each trial.

Figure 2 shows the same sensors, for the single test (seventh) trial. The test trial was about one third the duration of a nominal (no load) battery drain trial, due to especially heavy loads (i.e. much wheel motor activity). Figure 3 shows the evaluation of the learned battery resource profiles when applied to the test data. Those high and low resource bounds were learned using only the training data, for a prediction forward lag of 1 minute (i.e. $\Delta T=60$). The inputs for this example were various lagged values of the battery voltage (specifically, $T, T-1, T-2, T-4$, and $T-8$). The test data completely fits within the bounds. The noticeable looseness is a result of having learned bounds which contain all 6 of the training trials; a profile learned for this test trial alone would be much tighter (but more prone to not fit future data). The looseness is especially obvious for the high bound. This arises from the fact that the training trials involved a variety of loads, some much less than for the test trial. Including relevant action events (such as motors being on or off) as inputs to these bounding functions would lead to tighter predictions.

¹Data gap periods of 1000 seconds (not shown in plots) were inserted between each of the six training data subsets, to avoid lag vectors from crossing any trial boundaries.

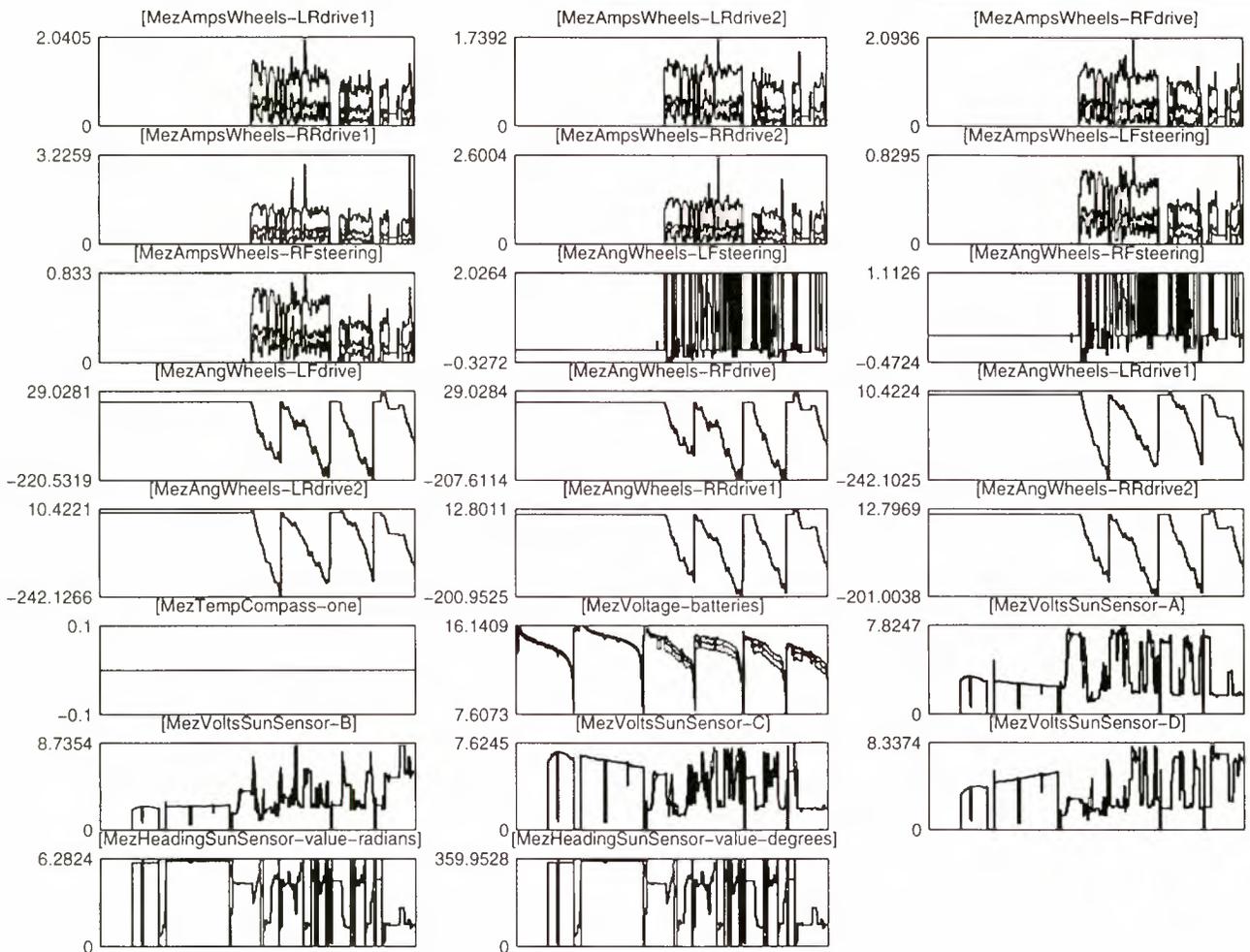


Figure 1: Plot of training data

Each box is a time-series plot of the named sensor. These time-series represent a concatenated sequence of six test-bed experiments, as suggested by the battery voltage (plot box which is 6th down, 2nd across) starting six times at high (full) values (near 16 volts).

3 Discussion

For more accurate, context-sensitive resource profiling, the inputs to the bounding functions should include quantities related to the actual loads over time. However, simply using the relevant raw sensor data (e.g. motor currents) will generally not work well. For resources such as battery power, integration over time windows greater than a few local samples is often effectively required, to model with sufficient precision the contributions and depletions of the underlying resource quantity (e.g. power).

Thus, we are investigating using features representing the total duration of various actions (e.g. camera on, motors on) between the current time T and the predicted time $T+\delta T$. We believe that using such

aggregate durations for each type of load activity (e.g. number of seconds motor 1 is on between time T and $T+\delta T$) as inputs, instead of the sensed quantities of those loads (e.g. actual electric current values at each motor over time) per se, also provide more useful models for use in resource management by planners. This is because a planner will reason at the level of such actions, and our model must itself be able to map those actions into worst-case and best-case consumption rates. Our use of such load-activity durations as inputs does reflect an assumption that the resource consumption is an additive function of such durations. Balancing the predictive imprecision that results from such abstractions, while still providing useful abstracted interfaces for planners (i.e. not at the detailed level of sensed load quantities), is our

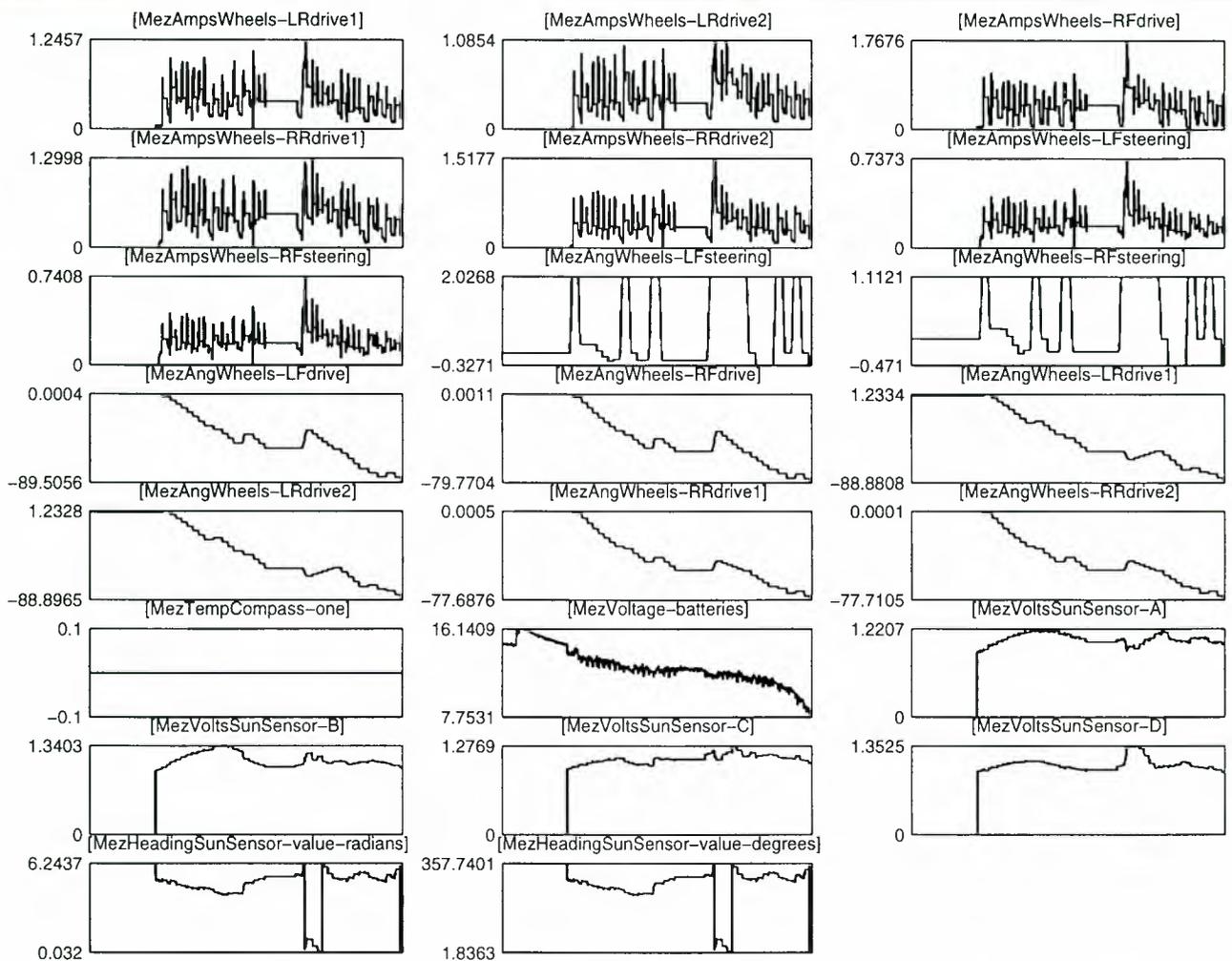


Figure 2: Plot of test data

current focus in ongoing research.

To overcome the expense and limited sample size of current testbed testing, we are currently evaluating these techniques on simulated Rover data, under a variety of load and action contexts. We plan to more tightly integrate this resource profiling capability with existing automated planning capabilities over the coming months (for preliminary architecture for such integration, see [3]).

4 Acknowledgements

Eddie Tunstel and John Szijarto provided significant feedback and assistance on this work.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

5 References

- [1] Dennis DeCoste. "Mining multivariate time-series sensor data to discover behavior envelopes". Proceedings of the Third Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, CA, August 1997.
- [2] Dennis DeCoste. "Automated learning and monitoring of limit functions". Proceedings of the Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-97), Japan, July 1997.
- [3] Tara Estlin et al. "An Integrated Architecture for Co-operating Rovers". Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-99), The Netherlands, June 1999.

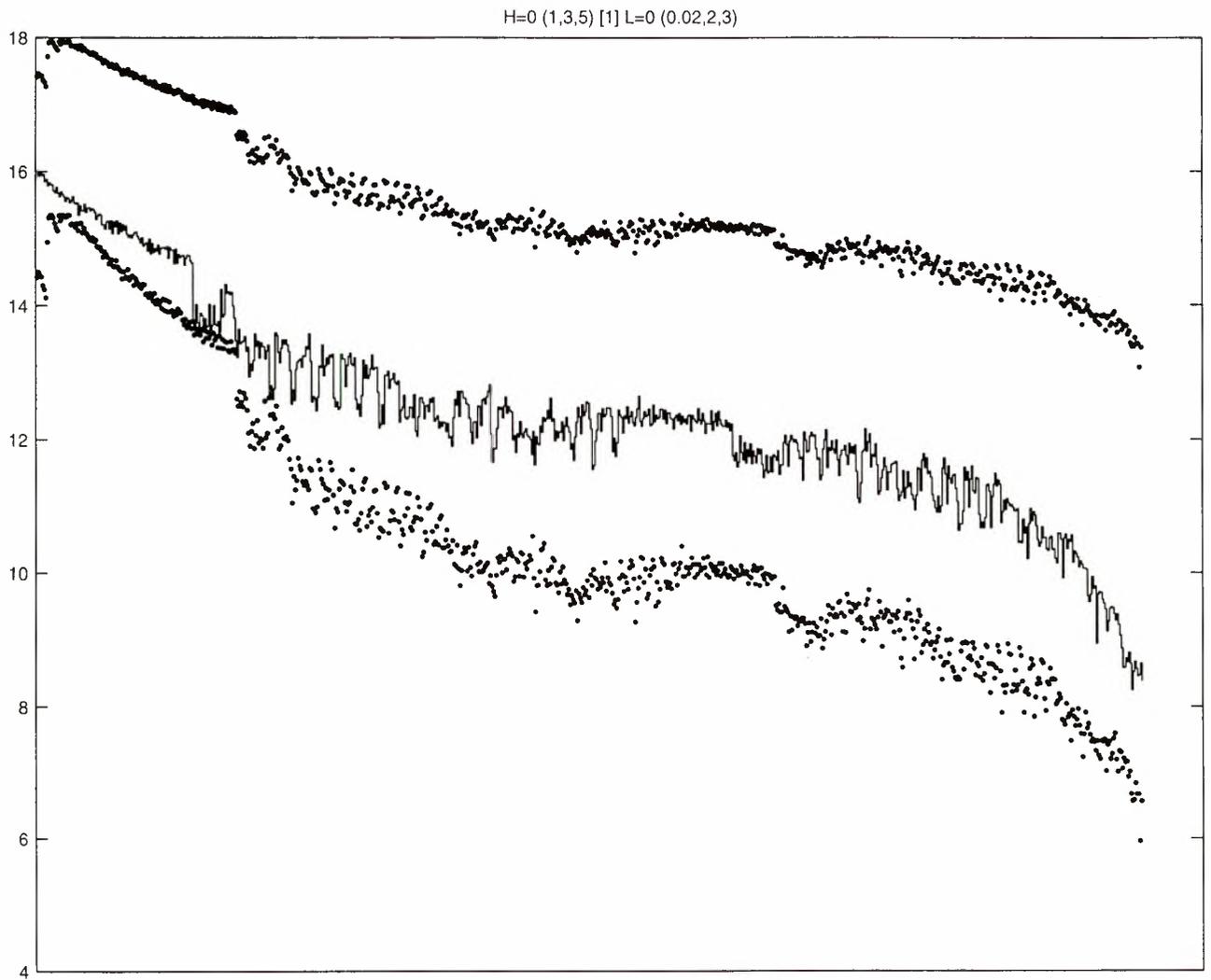


Figure 3: Plot of battery envelopes for test data

Time-series plot of upper and lower profile bound values, for 1 minute look-ahead prediction. The actual test data is between these bounds over all time points.

AUTONOMY ARCHITECTURES FOR A CONSTELLATION OF SPACECRAFT

Anthony Barrett

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 126-347, Pasadena, CA 91109-8099
phone: +1 818 393-5372, fax: +1 818 393-5244, e-mail: anthony.barrett@jpl.nasa.gov

ABSTRACT

This paper describes three autonomy architectures for a system that continuously plans to control a fleet of spacecraft using collective mission goals instead of goals or command sequences for each spacecraft. A fleet of self-commanding spacecraft would autonomously coordinate itself to satisfy high level science and engineering goals in a changing partially-understood environment – making feasible the operation of tens or even a hundred spacecraft (such as for interferometer or magnetospheric constellation missions).

1. INTRODUCTION

Until the past 5 years, missions typically involved fairly large expensive spacecraft. Such missions have primarily favored using older proven technologies over more recently developed ones, and humans controlled spacecraft by manually generating detailed command sequences with low-level tools and then transmitting the sequences for subsequent execution on a spacecraft controller.

This approach toward controlling a spacecraft has worked spectacularly on previous NASA missions, but it has limitations deriving from communications restrictions – scheduling time to communicate with a particular spacecraft involves competing with other projects due to the limited number of deep space network antennae. This implies that a spacecraft can spend a long time just waiting whenever a command sequence fails. This is one reason why the New Millennium program has an objective to migrate parts of mission control tasks onboard a spacecraft to reduce wait time by making spacecraft more robust [Muscettola et al. 97]. The migrated software is called a “remote agent” and can be partitioned into 4 components:

- a mission manager to generate the high level goals,
- a planner/scheduler to turn goals into activities while reasoning about future expected situations,
- an executive/diagnostician to initiate and maintain activities while interpreting sensed events through reasoning about past and present situations, and
- a conventional reactive controller to interface with the spacecraft to implement an activity’s primitive actions.

In addition to needing remote planning and execution for isolated spacecraft, a trend toward multiple-spacecraft missions points to the need for remote distributed planning and execution. The past few years have seen missions with growing numbers of probes. Pathfinder has its rover (Sojourner), Cassini has its lander (Huygens), Cluster II has 4 spacecraft for multi-point magnetosphere plasma measurements. This trend is expected to continue to progressively larger fleets. For example, one proposed interferometer mission [Mettler&Milman 96] would have 18 spacecraft flying in formation in order to detect earth-sized planets orbiting other stars. Another proposed mission involves 5 to 500 spacecraft in Earth orbit to measure global phenomena within the magnetosphere.

To describe the 4 software components of autonomous spacecraft and constellations, the next section describes a master/slave approach toward autonomously controlling constellations. While being a conceptually simple extension to single-spacecraft autonomy, this approach has several problems that motivate the next section on teamwork. Teamwork replaces masters and slaves with leaders and followers, where a follower has the autonomy to look after its teammates. The fourth section discusses ways to expand teamwork to let each spacecraft function both as a leader and a follower, and the last section concludes by discussing hybrids of the three architectures.

2. MASTER/SLAVE COORDINATION

The easiest way to adapt autonomous spacecraft research to controlling constellations involves treating the constellation as a single spacecraft. Here one spacecraft directly controls the others as if they were connected. The controlling “master” spacecraft performs all autonomy reasoning while the slaves only transmit sensor values to the master and forward control signals received from the master to their appropriate local devices (fig. 1). The executive/diagnostician starts actions and the master’s reactive controller manages actions either locally or remotely through a slave.

The 3 modules above the reactive controller essentially follow the standard belief-desire-intention (BDI) framework [Rao&Georgeff 95]. The mission manager takes a set of *beliefs* and generates *desires* (goals) for the

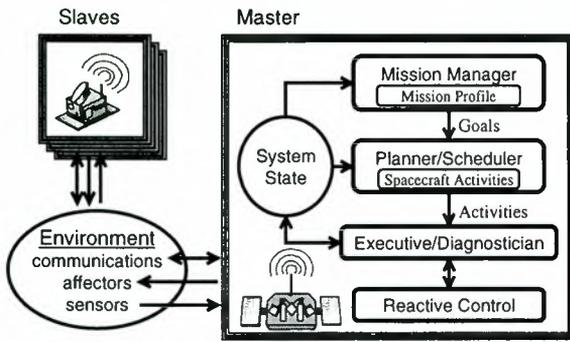


FIG.: 1 Architecture for Master/Slave Coordination

planner/scheduler, which translates them into *intentions* (plans) for execution. Gat describes 3T [Gat 97], another architecture with three layers to deliberate, sequence, and control. While deliberation combines mission management and planning, the other two layers match the executive and the reactive controller. EVAR [Schoppers 95] illustrates another case where the executive subsumes both the planner and mission manager. In general, we can describe most autonomous agent research as variants on the BDI model with different approaches toward implementing the modules and their interactions.

2.1. EXECUTIVE/DIAGNOSTICIAN

At the lowest level the executive/diagnostician (or just “executive”) takes an activity sequence, incrementally feeds activities to the reactive controller and monitors results to update the system’s state – a model of the constellation and its environment. Since performing an activity might have unintended situation dependant results, blindly feeding primitive activities to the reactive controller is unreliable. The issue here is that the Executive must rapidly diagnose and respond to detected contingencies.

EVAR [Schoppers 95] resolved the problem by compiling large sequences into *universal plans* – a clever encoding of state/response rules that enumerates all states and their appropriate responses. Unfortunately this approach only works in restricted domains where we can make a practical representation that implicitly enumerates all states.

Another approach involves robustly implementing each activity as a *reactive action procedure* (RAP) – an encoding of state/response rules for *anticipated* states [Firby 87]. Here activities fail when the current state falls outside the anticipated set, and failure forces the executive to abort the sequence and inform the planner. The issue now involves how many actions to feed the executive at a time.

For instance, one system uses variable size planning windows to generate sequences where one activity is to plan for the next window [Pell et al. 97], and another

system runs the planner continuously and feeds individual activities to the executive as they become executable [Ambrose-Ingerson&Steel 88]. While these examples show that the planner’s continual operation is optional, all systems must continually run the executive to actively monitor and diagnose the reactive controllers. This involves using a production system to appropriately apply state/response rules to affect the system state or reactive controller.

2.2. PLANNER/SCHEDULER

While the executive reasons about current and past activities, the planner/scheduler (or just “planner”) reasons about future command sequences. Given the heavy use of time and metric resources in spacecraft planning domains, we use a heuristic iterative-repair strategy [Rabideau 99] towards building and maintaining command sequences. This approach takes a complete plan at some level of abstraction and manipulates its actions to repair problems detected by envisioning how the plan would execute on the spacecraft. One type of problem involves multiple simultaneous actions with conflicting resource needs. For example, simultaneously activating too many sensors might cause a bus fault by drawing too much power. Repairing this problem would involve either deleting or moving sensor activation activities in the plan.

At any given moment the mission manager can suggest tasks for the planner/scheduler to add to the constellation’s future behavior. Since these tasks are often abstract and might conflict with other established tasks the scheduler continuously debugs its tasks and sends actions to the executive (fig. 2). The planner essentially maintains a set of tasks that are abstract in the far future and become progressively more detailed as their execution times approach. For example, a suggested task to take a picture of a target might involve slewing and possibly calibrating the camera prior to acquiring the image. This task is detailed as its execution time approaches. By continuously detailing the earliest tasks, the planner assures that it always has actions to send to the executive.

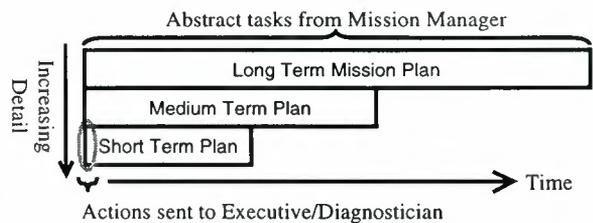


FIG.: 2 Continually updating the spacecraft activities

As time progresses, activities move from the future plan through current execution into the past. During this process an activity’s expected outcomes get replaced with

its sensed outcomes, and the constellation's actual state will drift from the expected state and cause future expectations to drift as well. The planner repairs the tasks whenever this drift causes a conflict.

2.3. MISSION MANAGER

This module facilitates high-level spacecraft commanding by maintaining beliefs involving the high-level mission profile. This profile contains a high level behavioral description for the spacecraft. This description can take many forms from a simple set of temporally constrained goals to an elaborate production system that asserts goals upon detecting user specified scientific opportunities by analyzing parts of the constellation & environment model.

For instance, the spacecraft would have periodic goals to transmit data to Earth. These goals would be temporally constrained in order to synchronize with a ground station. They also have to be high level to determine how to communicate based on the specific state of the spacecraft prior to preparing for a downlink. As another example, the mission manager might apply a feature detection algorithm on a previously captured picture and generate observation goals based on the results.

While a spacecraft can operate entirely autonomously with a mission profile. Humans analyzing the science results will tend to suggest changes to mission goals for answering questions arising from their analysis. We can even vary the constellation's level of autonomy by varying the abstractness of the mission profile. When using primitive action sequences, the profile can short-circuit the planner to allow absolute commanding. Adding abstract tasks to the profile lets the spacecraft adapt its behavior to its local environment, and adding data analysis for rule based autonomous goal generation makes a spacecraft detect and respond to scientific opportunities.

3. TEAMWORK

While the master/slave approach benefits from conceptual simplicity, it relies on an assumption that the master spacecraft's reactive controller can continuously monitor the slaves' hardware, and this relies on high-bandwidth highly-reliable communications. Since unintended results occur fairly rarely, one way to relax the bandwidth requirements involves putting reactive controllers on the slaves and only monitoring unexpected events. Unfortunately, this disables the ability to monitor for unexpected events between spacecraft and leads to a host of coordination problems among the slaves [Tambe 97]. Also, failures in the communications system can result in losing slaves.

We can apply teamwork models [Tambe 97, Stone& Veloso 98] to reduce the communications problem by giving the slaves their own executives (fig. 3). This replaces the master/slaves relationship with one between a

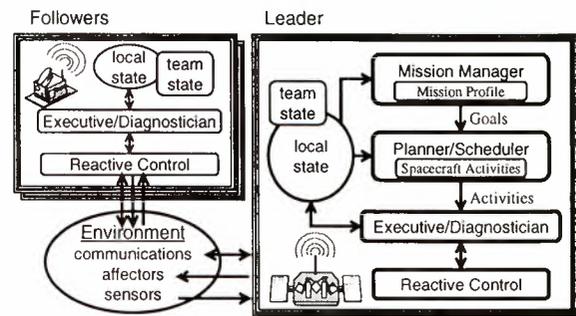


FIG.: 3 Architecture for Teamwork

team leader and its followers. Here each follower can monitor its own performance and selectively transmit results to the leader. Partitioning the system's state into local spacecraft states and shared team-states facilitates this selective transmission. While the spacecraft keep their local states private, they communicate to keep team-states consistent across teams in the constellation.

3.1. REPRESENTING TEAM PLANS

Instead of sending separate actions to each follower for execution, the leader broadcasts the entire reactive team plan¹ to all followers. This lets each follower actively monitor its own progress and passively track its teammates' activities. This passive monitoring process maintains robustness while reducing communications.

In addition to regular activities found in the master/slave approach, reactive team plans also include *team activities*. These define coordination points where the team synchronizes before and after executing the team activity. For instance, a 3 spacecraft interferometer has a combiner spacecraft to generate pictures by processing light reflected from two collector spacecraft. A reactive team plan to control the constellation might have 3 team activities (fig. 4) to coordinate the 3 spacecraft while making an observation, and each activity has 2 or 3 sub-activities defining how the constellation behaves during the joint activities. As illustrated, team activities have brackets and those suffixed with an asterisk only apply to subsets of the team. In this case the subset denotes the combiner spacecraft. The activities in this plan subsequently make the constellation attain a rough formation, dress up the formation for finer tolerances to make a measurement, and transmit the results to Earth.

While this interferometer's impoverished number of spacecraft do not sufficiently motivate the need for teamwork, other interferometer mission proposals describe over a dozen, or even a hundred, collectors to support the combiner. To support teamwork for these larger missions,

¹ Given our heavy use of Tambe's formalism, we adopt his terminology and call a sequence a *reactive team plan*.

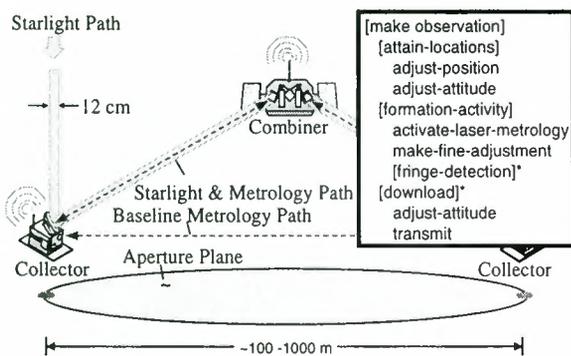


FIG.: 4 Structure of a reactive team plan for a 3 spacecraft interferometer

we must alter the executives' underlying architectures to manage each spacecraft team's associated team-state. We illuminate these changes by describing the machinery underlying team activity execution.

3.2. EXECUTING TEAM ACTIVITIES

A team of spacecraft contains a leader and one or more followers that jointly intend to accomplish some task by executing a team activity. Teams dynamically form when team activity execution starts and dissolve upon completion. When a team performs a task, it shares a team-state. This state contains facts like a list of teammates, their roles in performing the joint task, and other information to coordinate team activity.

Depending on the action, execution can manipulate the reactive controller and alter the local and team-state information. Since team-states are replicated across all teammates, a spacecraft must broadcast all team-state changes to maintain consistency. The standard protocol for changing a team-state is a 3-step process where one spacecraft broadcasts the change, all teammates broadcast acknowledgements in turn, and all teammates update their copies upon hearing everyone else. If a teammate does not respond before a time-out interval, the original spacecraft rebroadcasts the change.

While only transmitting team-state changes reduce communications, the number of broadcasts still implies bandwidth problems as the spacecraft population increases. Stopping spacecraft from broadcasting a change when teammates can infer it from observation further reduces communications [Huber&Durfee 95, Tambe 97]. For instance, the combiner in our interferometer example does not have to signal the end of a formation activity. The mere act of slewing to downlink the results tells the collectors that the formation activity is over.

3.3. GENERATING AND REPAIRING TEAM PLANS

Although reactive team plans might look like an extension on standard hierarchical plans by virtue of the bracket

syntax, techniques for building and managing hierarchical plans, like those described earlier, also apply to generating reactive team plans. As such planning does not change much when moving from master/slave plans to reactive team plans. Just like in master/slave coordination, there is a spectrum of ways to generate plans and feed them to the executives. At one extreme the lead spacecraft can generate a whole plan and then feed the resultant sequence to its executives, and at the other extreme it repairs the plan incrementally and maintains a copy in the shared team-state.

The real difference between the two approaches involves limiting the knowledge to plan from. Where the master knew everything about the constellation, the team leader only knows a subset of everything. The issue now becomes a matter of what status information to put in the subset and how fresh to keep it. While increasing the information and its freshness improves the leader's results, it also increases the communication overhead as the constellation's status changes.

A second issue involves whether the information belongs in the team-state, and whether it should be transmitted privately to the leader. While putting information in the team-state increases the followers' abilities to keep track of each other, it also increases the communications overhead. Where changing the team-state involves a broadcast followed by waiting for multiple acknowledgements, changing the leader's local state involves one transmission followed by waiting for the leader's acknowledgement.

One planning approach has the leader managing the team plan and follower roles in the team-state, but lets the followers privately transmit state updates to the leader. Here the leader changes the team plan and roles based on projecting its expected results given the privately received status information.

Another approach still has the leader managing the team plan's activities with heuristically assigned roles in the team-state, but followers keep status information local and submit change requests as they perform their roles in the evolving team plan [Fujita&Lesser 96]. While we can assign and reassign roles at random, a better approach involves auctioning off the unassigned roles to the teammates. The teammates bid on these roles based on local information as well as currently assigned roles, and the leader can either change the plan or assign roles based on these bids.

4. PEER-TO-PEER COORDINATION

The approach to alter communication overhead by distributing execution monitoring across the constellation can extend to also distributing the planning process. This addresses the possibility where the lead spacecraft is disabled. For interferometers this is not an issue because

losing the combiner spacecraft ends the mission anyway, but missions like a 50 satellite constellation are functionally redundant and should not end when any one spacecraft is disabled.

One way to increase robustness involves giving the other spacecraft backup planners and mission managers (fig. 5). While this lets the next spacecraft in a designated chain of command replace a disabled leader, these extra modules are underutilized. Instead of transmitting data to a central spacecraft for planning, we can use the extra planners to move parts of the planning process closer to the data. This makes the spacecraft symmetric and coordination becomes a collaborative effort among peers.

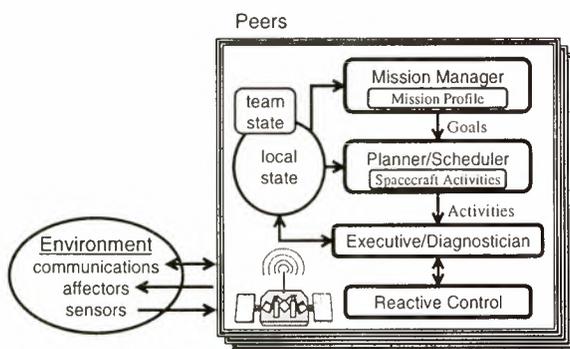


FIG.: 5 Architecture for Peer-To-Peer Coordination

This architecture works particularly well with constellations of satellites that loosely coordinate. For instance, a constellation of picture taking satellites might coordinate to partition desired targets, but each satellite runs in isolation to take its picture. Here the mission managers coordinate to partition the goals, and the planners and executives run in isolation. This class of loose coordination problem is common in the mobile robot community, and some systems even call this module a cooperative planning (or social) module [Müller 96].

4.1. LEVELS OF AUTONOMY

In teamwork or a chain of command, one spacecraft plans how to perform a task and its followers accept and execute the results. Combining loose coordination with teamwork facilitates letting different spacecraft act as leaders for different tasks. Here all spacecraft know about all tasks, and each task has a designated lead spacecraft. Research on autonomy levels [Martin&Barber 96] generalizes this idea. We can give each spacecraft a copy of the plan with tasks annotated with one of 5 autonomy levels:

- Observer: spacecraft does not participate,
- Command-driven: spacecraft serves as a follower,
- Consensus: spacecraft collaboratively plans with others,
- Local: spacecraft plans to perform task alone, and
- Master: spacecraft plans and serves as a leader.

As the 5 definitions imply, autonomy levels specify whether or not a spacecraft can change a task. For instance, a team's leader has tasks annotated with "master", and its followers' tasks have "command-driven" annotations. Given these annotations, a spacecraft can simultaneously serve as a leader and a follower in two separate teams. A spacecraft can even plan and perform tasks in isolation while participating in teams.

While autonomy levels specify which constellation members plan out mission manager requested tasks. These levels are not static – a spacecraft can communicate with the constellation to change a task's autonomy level annotations. For instance, a mission manager might always assign tasks to its spacecraft at the "local" autonomy level. If a team is needed to perform the task, the spacecraft will have to change the annotation to "master." As Martin points out [Martin&Barber 96], this change involves communicating to find spacecraft willing to accept "command-driven" annotations.

Using autonomy levels, we can treat the plan and state information as a shared database where each spacecraft has varying capabilities to modify tasks based on their autonomy-level annotations. Softening the distribution requirement from full to partial plan sharing makes a constellation operate as a team at one point and as multiple independent spacecraft as another. The change involves letting spacecraft keep locally planned and executed tasks private.

4.3. COLLABORATIVE PLANNING

Unlike the other annotations where a single spacecraft plans a task, the "consensus" annotation implies that multiple spacecraft collaboratively plan to perform a task. Collaborative planning involves distributing the plan across the constellation and letting each spacecraft detect and repair problems. The question now becomes a matter of how to keep the plan consistent across the constellation while all spacecraft are updating it. The main objective is to minimize communications overhead while planning.

One approach would fragment the plan and distribute the fragments [Corkill 79]. Since the fragments are disjoint, their union would be consistent. Each spacecraft would expand its own fragment and communicate to detect and resolve interactions. To detect interactions, each spacecraft broadcasts its fragment's effects upon determining them. When a spacecraft hears of an effect that either helps or hinders its own fragment, it initiates a dialog with the broadcasting spacecraft to add signaling actions to their plans to coordinate the interaction. Thus the required bandwidth depends the amount of interaction.

An alternative approach would give every spacecraft a copy of the plan and have them maintain consistency by broadcasting changes as they make them. The main

problem with this approach involves communication overhead – the spacecraft would spend most of their time responding to each other's updates.

These two approaches define a whole spectrum of collaborative planners depending on the amount of shared plan and state information. While the first case shared all state information in the form of advertised effects the second shared all plan information.

5. CONCLUSIONS

This paper described several autonomy architectures for an autonomous constellation of spacecraft. Such a constellation would continually plan to control its spacecraft using collective mission goals instead of goals or command sequences for each spacecraft. The first architecture made use of research relating to a single autonomous spacecraft by treated the constellation as a single master spacecraft with virtually connected slaves.

The utilized research describes implementations in terms of 4 interacting modules, and the master/slave architecture placed all modules on the master. While the teamwork and peer-to-peer architectures keep the 4 modules, they progressively give the slaves more authority by replicating more of the modules across the constellation.

While this paper described each architecture in isolation, these architectures can coexist within a constellation. Such a constellation would have 3 classes of spacecraft: leaders, followers, and slaves. Where leaders have the ability to plan and collaborate, followers can only execute plans and watch out for each other. Both leaders and followers can have virtually attached slave spacecraft.

ACKNOWLEDGEMENTS

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES

- [Ambrose-Ingerson&Steel 88] J. Ambrose-Ingerson and S. Steel, "Integrated Planning, Execution and Monitoring," Proceedings of AAAI-88.
- [Corkill 79] D. Corkill, "Hierarchical Planning in a Distributed Environment," In Proceedings of IJCAI-79.
- [Firby 87] R. Firby, "An Investigation into Reactive Planning in Complex Domains," Proceedings of AAAI-87.
- [Fujita&Lesser 96] S. Fujita and V. Lesser, "Centralized Task Distribution in the Presence of Uncertainty and Time Deadlines," In Proceedings of ICMAS-96.
- [Gat 97] E. Gat, "On Three-Layer Architectures," *Artificial Intelligence and Mobile Robots*, D. Kortenkamp, R. Bonasso, and R. Murphy, eds., AAAI Press.
- [Huber&Durfee 95] M. Huber and E. Durfee, "On Acting Together: Without Communication," AAAI Spring Symposium on Representing Mental States and Mechanisms, 1995.
- [Martin&Barber 96] C. Martin and K. Barber, "Multiple, Simultaneous Autonomy Levels for Agent-based Systems," In Proceedings of the Fourth International Conference on Control, Automation, Robotics, and Vision.
- [Mettler&Milman 96] E. Mettler and M. Milman. "Space Interferometer Constellation: Formation Maneuvering and Control Architecture," In SPIE Denver '96 Symposium.
- [Müller 96] J. Müller, *The Design of Intelligent Agents, A Layered Approach*. Lecture Notes in Artificial Intelligence, Springer-Verlag, 1996.
- [Mussettola et al. 97] N. Mussettola, et al. "On-Board Planning for New Millennium Deep Space One Autonomy," Proceedings of IEEE Aerospace Conference 1997.
- [Mussliner 93] D. Musliner, *CIRCA: The Cooperative Intelligent Real-Time Control Architecture*, PhD Thesis, University of Michigan, 1993.
- [Rabideau et al. 99] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations Using the ASPEN System," in Proceedings of iSAIRAS-99.
- [Rao&Georgeff 95] A. Rao and M. Georgeff, "BDI Agents: From Theory to Practice," Proceedings of ICMAS-95.
- [Pell et al. 97] B. Pell, E. Gat, R. Keesing, N. Mussettola, B. Smith, "Plan Execution for Autonomous Spacecraft," Proceedings of IJCAI-97.
- [Schoppers 95] M. Schoppers, "The use of dynamics in an intelligent controller for a space faring rescue robot," *Artificial Intelligence*, 73 (1995) 175-230.
- [Simmons 88] R. Simmons, "A Theory of Debugging Plans and Interpretations," In Proceedings AAAI-88.
- [Stone&Veloso 98] P. Stone and M. Veloso, "Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork," Submitted to ICMAS'98.
- [Tambe 97] M. Tambe, "Towards Flexible Teamwork," *Journal of Artificial Intelligence Research*, 7:83-124.

Robotic Satellite Servicing Experiments on ETS-7



RESULTS OF THE ETS-7 MISSION - RENDEZVOUS DOCKING AND SPACE ROBOTICS EXPERIMENTS

Toru Kasai, Mitsushige Oda and Takashi Suzuki

National Space Development Agency of Japan (NASDA), ETS-VII project team,
2-1-1 Sengen, Tsukuba-shi, Ibaraki-ken, Japan
phone:+81-298-52-2863, fax:+81-298-60-6718
email: Kasai.Toru@nasda.go.jp

Abstract

On orbit servicing is the indispensable function for future space activities such as building and operation of the international space station, inspection and repair of orbiting satellites, and conducting lunar/planetary explorations. The Rendezvous Docking (RVD) technology to meet and to connect two spacecraft in-orbit is also an essential technology for the future space missions such as logistic support of the international space station. Therefore, National Space Development Agency of Japan developed and launched an engineering test satellite named ETS-VII (Engineering Test Satellite #7) on November 28th, 1997 to conduct the RVD and space robot technology experiments. This paper shows an overview of the ETS-VII project and the results of various experiments on it.

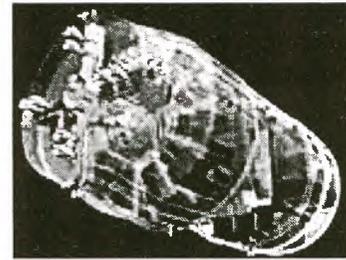


Fig. 2 Artist's image of the H-II transfer vehicle (HTV)

1.2 Expectation for the space robots

There are many tasks to be conducted in space such as building and operation of the international space station, inspection and repair of orbiting satellites, and conducting lunar/planetary explorations. Some of these tasks are currently conducted by astronauts. However most of these tasks are highly risky and expensive. Therefore space robot becomes attractive tool to conduct these tasks. The space shuttle's remote manipulator system (Ref.1) is used many times to deploy and capture satellite to/from space. Similar manipulator systems are to be mounted on the international space station to handle large element of the space station. (Ref.2, Ref.3) These space robot will expand ability of the astronauts. However these space robot are manipulated by astronauts and the limitation of the available manpower of the astronauts limits the capability of the space robots. If the robot can be teleoperated from the ground, the limitation of the astronaut's manpower will disappear. NASDA defined such robots as the second generation space robot. (Ref.4) The first generation space robot what is manipulated by the onboard astronaut(s) such as the shuttle remote manipulator system and the space station's manipulator systems. The third generation space robot is what is highly autonomous one. Planetary exploration robot that does not require frequent assistance from operator would be an example of the third generation space robot and will be realized in near future.

1.3 ETS-VII satellite

Mission of the ETS-VII is to conduct the unmanned automated RVD experiments and the space robot

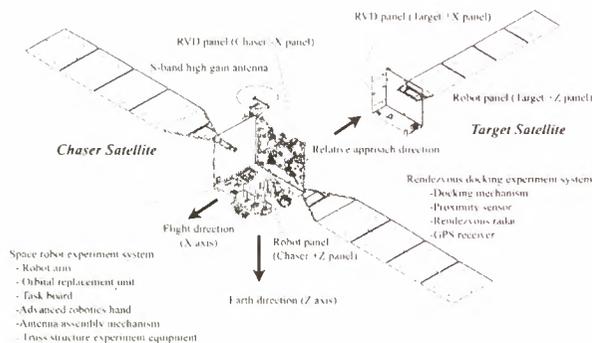


Fig. 1 ETS-VII chaser and target satellite

1. Introduction

1.1 Expectations for the rendezvous docking

In building a large space station in orbit, or providing maintenance services for orbiting satellite such as repair and fuel supply, the RVD technology is necessary. In operating the international space station, NASDA is to deliver Japanese logistic module using NASDA's H-II rocket. Fig.2 shows H-II transfer vehicle named HTV, which deliver the Japanese logistic module to the space station. Even though NASDA has a long history of research on RVD technology, NASDA did not have any experience to conduct the RVD in space. Therefore, NASDA decided to develop the Engineering Test Satellite No.7 (ETS-VII) to conduct the RVD and space robot technology experiments.

technology experiments. Since there was not appropriate satellite in orbit that can be used as target for the RVD experiments, chaser satellite and the target satellite were launched together. They are also called as HIKOBOSHI and ORIHOME respectively which mean a hunter boy and a weaver girl who were separated by the milky way and were allowed to meet each other once a year on 7th of July in an old Japanese love story. Mass of the chaser and the target satellite are 2.5t and 0.4t respectively.

ETS-VII satellite was launched by H-II rocket on November 28th, 1997. The orbit of the satellites is 550km altitude and 35degrees inclination. Mission life of the ETS-VII satellite was 1.5 years after the launch. However, satellite's status after 1year in orbit was good, mission period was extended to two years after the launch. The ETS-VII satellite on the H-II rocket is shown in Fig.3.

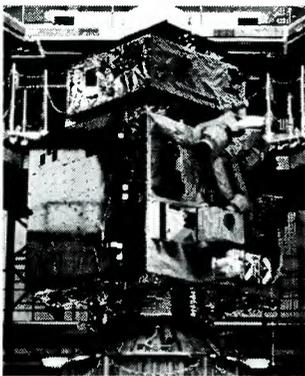


Fig. 3 ETS-VII on board H-II F6

1.4 ETS-VII project

Since ETS-VII is a rare opportunity to conduct the space robot experiments in space, following national agencies were invited to participate the project. These agencies developed their own onboard robot experiment equipment and on-ground control systems to conduct their own space robot experiments. Those are;

- Ministry of International Trade and Industries (MITI) conducts the advanced robotic hand experiments.
- National Aerospace Laboratory (NAL) conducts handling of truss structures by the tele-manipulated robot arm.
- Communication Research Laboratory (CRL) conducts antenna assembling experiments using the onboard robot arm.

Details of experiments by these agencies are shown in Ref.5, 6 and 7.

1.5 ETS-VII experiment system

ETS-VII experiment system consists of the satellite mounted rendezvous docking and robot system, the on-ground control system and the communication network, which connects the both systems. The communication between the onboard system and the ground control system which is located at NASDA's Tsukuba space

center is realized using a data relay satellite (NASA's TDRS) in the geo-stationary Earth orbit. This communication network is shown in Fig.4. It is realized by the computer network of large number of computers at various sites. This computer network based communication cause time delay. In the case of ETS-VII, the time delay is about 6 to 7 seconds in return. A similar time lag can be observed on the internet-based communication.

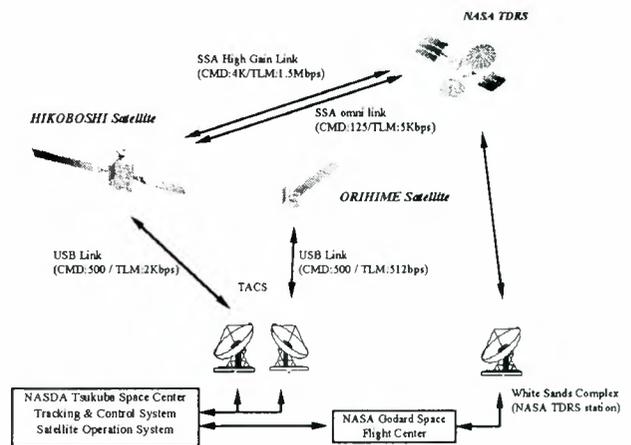


Fig. 4 ETS-VII experiment system

2. Rendezvous docking mission of ETS-VII

2.1 Mission objective

Purpose of the ETS-VII's RVD experiments are to conduct following technology experiments and provide confidence in conducting the following HTV project as mentioned in the section 1.2.

- Unmanned autonomous RVD between the chaser and the target satellites.
- Experiment of the remote piloting of the chaser satellite from the on ground control station.
- Conduct rendezvous flight which simulates the rendezvous against the international space station.

These RVD mission objectives were decided from the following considerations.

2.1.1 Ways of rendezvous

There are several ways of the RVD method. Direction of approaching the target is a matter of consideration. The Russian logistic support vehicle (Soyuz, Progress) approaches the Mir space station from either in front of the station or behind the station. The space shuttle also used similar approach in the past mission that recovered orbiting satellites. However the international space station adopted new way. Spacecraft that approach the space station must approach the station from beneath of it. This is to increase safety (decrease risk of collision of both spacecraft) during the final phase of rendezvous. In this way of approach, it is possible to find a trajectory that does not intersect the orbit of the space station. Approach from in front of or from behind must take a trajectory that intersects orbit of the target.

2.1.2 Way of docking

Way of docking is another point of consideration. The already established way of docking which is used by Russian Soyuz / Progress spacecraft and the space shuttle is “docking” which makes connection of two spacecraft using kinetic energy caused by the relative speed of two spacecraft. The relative speed at docking is in the order a few cm/sec. However this type of docking produces vibration of the flexible appendages such as solar panel. It is not adequate to apply this way of docking to the in-orbit connection of satellite, since satellite are generally built by light weight material and are not rigid enough to conduct the above type of docking. Therefore, a new type of docking mechanism and way of approaching are required.

2.1.3 Unmanned rendezvous docking

Way of control of the RVD is also matter of consideration. Rendezvous docking of NASA’s space shuttle or Russian Mir station is controlled or monitored by astronauts onboard the space craft. However in the case of ETS-VII, it is unmanned satellite and continuous monitoring / control can not be realized without using two or more data relay satellite in geostationary orbit. At the time of planning the ETS-VII satellite, only one experimental data relay satellite was foreseen to be used by ETS-VII. Therefore ETS-VII’s RVD system was decided to be autonomous one.

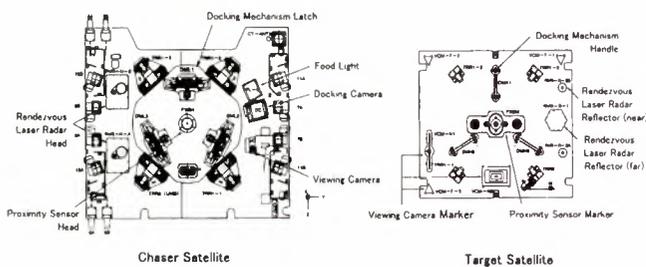


Fig. 5 RVD equipment on docking panel

2.2 GNC function of ETS-VII RVD

To perform autonomous RVD, the navigation function to measure and estimate relative position and velocity are needed. ETS-VII has three navigation sensors which are GPS receiver (GPSR), Rendezvous laser Radar (RVR) and Proximity sensor (PXS), and selects main navigation sensor according to relative range. ETS-VII GNC function is divided into three phase. One is the docking phase using the PXS. Another is the final approach phase using the RVR. The other is relative approach phase using the GPSR. The overview of these function as follows. (Ref. 8,9) Table-1 shows GNC performance required in each phase.

Within 2m distance, the relative position and relative attitude of the target satellite are measured by the PXS mounted on the chaser satellite. The chaser satellite approaches or goes away to/from the front of target satellite using this data. In the separation phase, The

Table-1 GNC performance required in each phase.

		requirement
TF Injection Accuracy		
Relative position	$((X-TF)^2+Z^2)^{1/2}$ (m)	less than 80
	Y(m)	less than 60
Relative velocity	$V_{x,y,z}$ (m/sec)	0 ± 0.2
VP Injection Accuracy		
Relative Position	X(m)	$VP\pm 0.3$
	Y(m)	0 ± 0.3
	Z(m)	0 ± 0.3
Relative Velocity	X,Y,Z (m/sec)	0 ± 0.01
DM Capture Condition		
Relative Position	X(m)	0.532 ± 0.043
	Y,Z(m)	0.0 ± 0.025
Relative Velocity	V_x (m/sec)	0.01 ± 0.0005
	V_y (m/sec)	0.0 ± 0.0002
	V_z (m/sec)	0.0 ± 0.0005
Relative Attitude	roll(deg)	0.0 ± 1.9
	pitch, yaw(deg)	0.0 ± 1.6
Relative Attitude Rate	roll, pitch, yaw (deg/sec)	0.0 ± 0.1

chaser satellite drifts out of capture area of Docking Mechanism Latch (DML), the chaser satellite automatically starts 6 DOF control based on PXS navigation. In the approach phase, the chaser approaches for the target at the velocity of 1cm/sec with 6 DOF control. On injection capture area, the DML automatically closes and captures the Docking Mechanism Handle (DMH) mounted on the target satellite. On the capture area of DML, both satellite's Reaction Control System (RCS) are disabled.

Between 2m to 520m distance, the relative distance and the direction of the target satellite are measured by the RVR. In this phase, reference trajectory guidance and Line Of Sight (LOS) control by the RVR navigation are adopted. The RVR measures the relative range and LOS angle. The Guidance Control Computer (GCC) estimates the relative position and velocity by processing the RVR measurement and earth sensor (ESA) data. The GCC issues the position and velocity command along the fixed reference trajectory along the V-bar. The chaser approaches/departures at the velocity of 5-10cm/sec along the V-bar. During V-bar approach/departure, the chaser executes LOS control against the target. At the end of the approach phase, the chaser is injected to Vicinity Point (VP:2m) to hand over the docking phase.

Beyond 500m distance, the relative distance and the relative velocity are measured by the GPSR mounted on both satellites and a flight trajectory of the chaser satellite is automatically generated by the GCC. In this phase (relative approach or departure phase), the C-W guidance based on the relative GPS navigation is adopted. The chaser satellite GPSR receives absolutely navigation data of the target that is transmitted from the

target GPSR via the inter-satellite communication link between the chaser and the target satellite. The chaser GPSR computes relative position and velocity. The GCC computes delta-V command to inject the aimed position based on the GPS relative navigation data. To minimize the injection error, mid course maneuvers are also executed. Each the thruster firing automatically cut off based on the velocity increment measured by the accelerometer (VIC: Velocity Increment Cut off guidance). The injection accuracy by the GPS relative navigation is about 80m. On the other hand, the measurement range of RVR is 600m and the FOV of RVR is 3deg for the relative range. Accordingly, in the worst case, the LOS angle is over 4deg. Therefore we fixed the acquisition point by RVR (Terminal phase Finalization point :TF point) at 520m on the V bar, and the Chaser control LOS pointing based on the GPS relative position to pick up the Target by RVR.

Beside the above autonomous RVD, remote piloting of the chaser satellite is planned to supplement the autonomous functions.

2.3 ETS-VII Flight Management Function

The flight management function is very important function for autonomous RVD. ETS-VII flight management system (FMS) is realized by RendezVous docking Flight Software (RVFS) installed on Guidance Control Computer (GCC). The transition among these modes are executed by FMS or command from ground support facility. The ground support crews monitors both satellite conditions before transition to the next phase and sends "GO" command. During the RVD experiment flight, the FMS ensures the safety of the Chaser and the Target i.e. prevent from collision. In the ETS-VII, two fail safe criteria is adopted within 30m from the Target. To ensure this criteria, not only FMS is installed the fault tolerant computer but also safe approach trajectory and closing velocity are selected.

The FMS has the five functions, mode control, time control, fault detection, fault isolation/recovery and safety control. The mode control function change the mode/sub-mode according to fixed condition about relative position, relative velocity, sensor/actuator status and system status. The time control function calculates the elapsed time from beginning of each experiment and relative time based on GPS time. The fault detection function detects the failure of sensors (PXS, RVR, GPSR earth sensor and inertial reference unit) and actuators (reaction control system, docking mechanism). The failures are detected by monitoring change of sensor data and status. The fault isolation/recovery function discriminates the fail sensor or actuator and switches it to redundant component. If redundant sensor or actuator is fail (two failure), the mode transition to Safety Ensuring Mode (SEM) is required. The safety control function detects the failure of satellite system and changes it to SEM if necessary. The system failures are as follows. 1)navigation failure, 2)guidance failure,

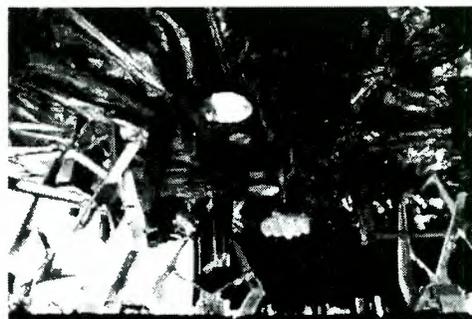
3)Chaser/Target, Chaser/TDRS inter satellite communication link failure 4) Guidance Control Computer (GCC)failure, 5) power supply system failure.

If mode transition to SEM is required, a Disable Abort (DA) or a Collision Avoidance Maneuver (CAM) is executed automatically according to relative position. The DA is adopted beyond 30m range from the Target. In this region, non-collision trajectory, witch both satellites never collide if the Chaser orbit maneuver disable (DA) is selected. CAM is adopted within 30m range from the Target because non-collision trajectory can not be selected within the area . In this region, the approach velocity along the V-bar is limited ($\leq 5\text{cm/sec}$) under the safe velocity witch the CAM can be executed.

2.4 Experiment Result

2.4.1 First RVD Experiment Flight

On July. 7 1998, the first autonomous RVD experiment flight (Flight Path 1 (FP-1)) are successfully completed. The FP-1 is executed during one event of the TDRS (1 event is approximately 40min.). The ground support crews sent the "separation command" at 7:09(JST) and the Chaser separated the Target at the speed of 1.8cm/sec. When the Chaser drifted out the capture area of Docking Mechanism Latch (DML) , the chaser started to control automatically and the Target started earth pointing control. Then the Chaser kept the relative position at Vicinity Point (VP:2m) for 15 minutes. After position keeping at VP, the ground crews sent the "Target control off" "approach command" and the Chaser approached the Target at the speed of 1cm/sec. On entering DML capture area, the Chaser automatically cut off its thruster and the DML captured the docking mechanism handle on the Target and mated it at 7:33 . Photo- 1 shows docking camera on the Chaser



Upper : The Target, Lower : The Chaser, Left bright sphere : The earth

Photo- 1 Capturing Target satellite before contact

2.4.2 Second RVD Experiment Flight

On Aug. 7 1998, the second autonomous RVD experiment flight (Flight Path 2 :FP-2)) was performed. In the original plan, the Chaser would make a flight over range of 0 to 500m using the RVR. It would take four hour from separation to docking.

In the first sight of TDRS, separation operation, which had been already demonstrated in FP-1, was executed and the Chaser kept the position at VP for 10 minutes. During the position keeping, the Chaser switched its navigation system from PXS to RVR. After checking performance of the Line Of Sight control using RVR, ground support crews sent “VP departure command” at 3:13 (JST) . The Chaser departed along the V-bar at the speed of 10cm/sec.

In the second event of TDRS, the Chaser was departing at the range of 450m. Then the Chaser reached at 525m point. After monitoring FMS’s mode change and status of both satellites, ground support crews sent “TF departure command” at 4:57(JST). The Chaser approached along the V-bar at the speed of 10cm/sec.

In the third event of TDRS, the Chaser would be approaching at the range of 30m from the Target. But in the acquisition of sight from TDRS, the Chaser change its flight mode to Safety Ensuring Mode (SEM), and made flight at the range of 1.6km from the Target to retreat point. During invisible time after second event, the Chaser’s thruster miss firing was happened. Then the Chaser caused attitude anomaly. As a result, the Chaser automatically change its mode to SEM and executed the Disable Abort (DA).

After this anomaly, five times attitude anomalies were happened during V-bar approaches and position keeping maneuvers. During the experiment, the Chaser made a flight over range of 0 to 12km using not only RVR but also GPS relative navigation. Thruster miss firing, witch caused attitude anomalies, always happened on the Z direction thrusters. The Chaser assigned three thrusters for three degree of freedom such as +roll, +pitch and +Z. If the Z thruster is miss firing, the control of 3 degree of freedom must be performed by two thruster. Therefore, attitude anomaly was happened.

Many approaches were tried to make re-docking with the Target. To avoid the Z thruster miss firing, it was necessary to limit to use the Z thruster. Therefore, only X thruster were used to inject into the TF point as possible, and V-bar approach range was made short. In the original plan, the Chaser would switched its navigation from relative GPS navigation to RVR navigation at TF(520m) point. But, fortunately, the relative GPS navigation was much more accurate than our expectation. Then the Chaser could be injected to TF’(150m) point by C-W guidance or manual delta X maneuver using relative GPS navigation. Additionally, RVFS was modified to minimize possibility of attitude anomaly by the Z thruster miss firing. As a result, on Aug. 27th, the Chaser successfully mated with the Target. Fig. 6 shows trajectory in the FP-2 experiment.

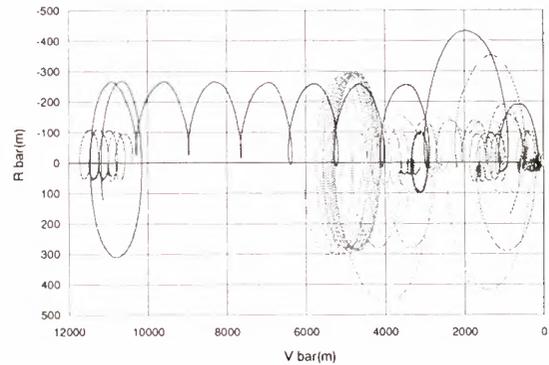


Fig. 6 trajectory in the FP-2 experiment.

3. Space robot mission of ETS-VII

3.1 Mission objectives

ETS-VII robot experiments are to conduct following experiments and to show taxpayers that the teleoperated space robot or the second generation space robot is a useful tool for future space mission.

- Performance evaluation of the onboard robot system: ETS-VII robot arm works in space more than one year without maintenance.
- Experiment of the coordinated control of the satellite attitude and onboard robot arm.
- Teleoperation of the onboard robot arm from ground.
- Demonstration of the in-orbit satellite servicing such as visual inspection, equipment exchange, fuel supply, target satellite handling and others:
- Provide space robot experiment opportunity to national laboratories outside NASDA since opportunity to conduct space robot experiments in space is quite rare. (Ref.5-7)

3.2 ETS-VII robot experiment system

3.2.1 Onboard robot system

ETS-VII onboard robot system consists of 6 DOF robot arm and a set of robot arm's payloads which are shown in Fig.7. These equipment are mounted on an Earth looking surface of the chaser satellite. This is to use the reflected sun light from the Earth atmosphere as the light source for the cameras. The reflected sunlight from the earth atmosphere is preferable since it is scattered light and makes little shade. (Ref.10)

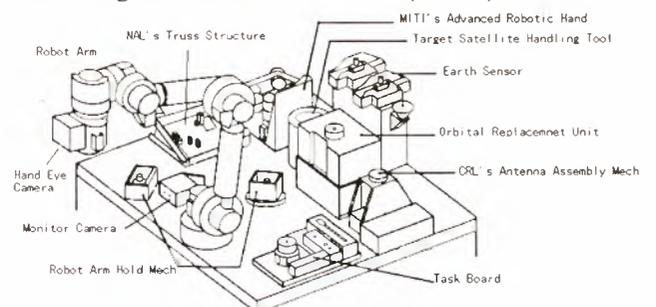


Fig. 7 ETS-VII onboard robot system

(1) Robot arm

The ETS-VII robot arm is about 2m stretched length and its joints are driven by combination of the DC brush-less motor, the harmonic-drive-gear[©] and a resolver.

ETS-VII robot arm has following control modes;

- Arm tip position control mode
- Joint angle control mode
- Compliance control mode (incl. force control, active limp and impedance control)

The compliance control is realized by the onboard robot control system using data from the force-torque sensor on the robot arm. Instructions to the onboard system are given by the on-ground robot control system using robot language command such as "move A to B". On-board robot control system generates trajectory to realize the instructed robot arm's tasks and calculates joint angle to realize the required robot arm's motion. Joint velocity or arm tip velocity commands are sent in forms of joint position or arm tip position commands. This is to assure safety against sudden disruption of the communication link during the robot arm's motion. These control modes were tested in orbit and showed good performance. The positioning accuracy (repeatability) of robot arm was better than 1.5mm.

(2) Video cameras on the robot arm

A hand eye camera is mounted on the end effector. Another monitor camera is mounted on the first joint of the robot arm. The first joint acts as camera's pan unit. Up to five video images out of two cameras per a second (5 frame/second) can be sent to ground using the JPEG compression format.

(3) Add-on tools

ETS-VII robot arm's end-effector is most suitable to handle medium size/weight equipment, which is attached on the same platform with the robot arm. It is not suitable to handle small equipment or to grasp a floating object. Therefore, ETS-VII robot arm uses additional tools to handle these payloads. A taskboard handling tool, which has two fingers and a fixed peg, is used to handle various equipment on the taskboard such as slider, switch, and hole for peg-in and others. A target satellite handling tool, which has large two fingers, is used to grasp the target satellite. Photo-2 shows the robot arm with the taskboard handling tool is handling a metal ball with a metal chain on the taskboard. (Ref.11)

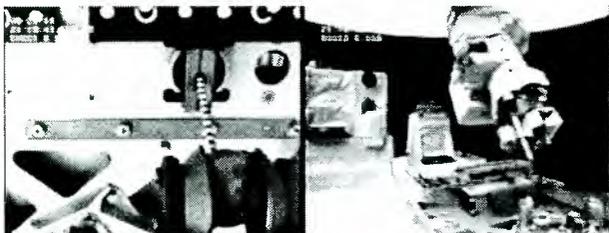


Photo- 2 Handling of small parts using add-on tool

(4) Orbital Replacement Unit (ORU)

The orbital replacement units (ORU) are widely used on the internal space station to exchange equipment in orbit. ETS-VII carries one ORU as an experimental payload of the robot arm. Size and mass of the ORU are similar with those of a microwave oven. It housed fuel tanks, valves, liquid connector and electrical connectors which were used by the fuel supply experiments. Photo-3 shows ORU handling by the onboard robot arm.

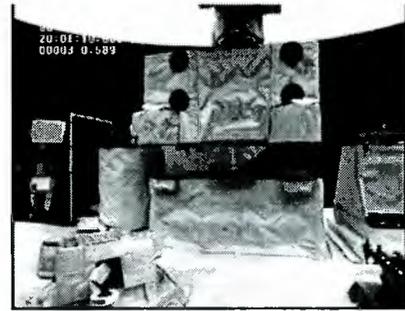


Photo- 3 ORU handling by the ETS-VII robot arm.

3.3 On-ground robot control system

Design requirement for the robot teleoperation system of ETS-VII is;

- Number of operators necessary to operate the system should be minimum and the hours to be required to learn and to prepare operations should be minimum.
- Any dangerous action such as collision against other object or too fast motion which distributes satellite attitude stability should automatically be prohibited even if the operator try the action without knowing the influence of the planned action.

Since ETS-VII's data communication is realized by the packet data transmission through a lot of computers at various sites which are shown in Fig.4, the overall time delay in the robot control loop is 6 to 7 seconds.

Command data to the onboard robot system can be sent in two modes, the non-interval command mode and the interval command mode. In the time interval command mode, command data can be sent at each 250msec. However to maintain a data rate within the allocated data rate, limited volume of data can be sent in this mode. In the non-interval command mode, larger command data can be sent at larger time interval. ETS-VII's teleoperation system is designed as follow to realize the above mentioned requirements and these constraints. (Ref. 12-16)

3.3.1 Teleoperation Mode

According to the two command data transmission modes, ETS-VII robot arm system has two teleoperation modes, the "supervised control mode", which uses non-interval commands and the "telemanipulation mode", which uses time interval commands. In the supervised

control mode, instruction to the onboard robot system can be sent in codes which mean like "Move from A to B at a speed of C, acceleration D, compliance parameters of E and, etc....." The onboard robot control system decodes this instruction to generate robot arm's tip trajectory, to calculate joint angles using the inverse kinematics, and controls individual joints.

If the robot arm's working environment and the tasks to be conducted are well defined, the automatic task execution is realized using this control mode. In this mode, the command sequences can be written using GUI (graphical user interface) into a flowchart. This commands sequence is verified using the on-ground robot simulator, which simulates the actions of the onboard robot system. The verified command sequences are stored in the robot operation facility. In the actual operations, necessary command sequences are selected by an operator and instructed to start when they are required. Then each command is sent out automatically each after the previous command is successfully conducted. This operation method is simple and safe, and is recommended for most of space robot's tasks, which are well defined.

3.3.2 Telemanipulation mode

In the telemanipulation mode, instruction to the onboard robot system are sent in the form of the robot arm's tip position and pose at each 250msec. These instruction are generated from input by two 3-dof joysticks, which are similar with those of the space shuttle's remote-manipulator system. The onboard robot control system will generate robot arm trajectory by interpolating these data. If one data is missing by a communication error or other reasons, the onboard robot system will interpolate the missing command. If more than two commands are missing, the onboard system will stop robot arm's motion assuming that the commands from the on-ground station were stopped by the operator or by the communication error. If commands from the on-ground station does not arrive constantly, the onboard computer's FIFO (first-in-first-out) buffer adjusts the time interval.

Teleoperation under the time delay of 6 seconds is not easy. ETS-VII's on-ground robot control system uses following operator aids to assist telemanipulation.

- Predictive computer graphics, which show how the robot arm will move if a command will be executed.
- Shared control between the telemanipulation and the automatic control.
- Imaginary guide plane to guide the robot arm motion to a desired position and to inhibit other motions.

3.3.3 Telemanipulation by a shuttle astronaut

In March 1999, NASDA astronaut, Mr. Wakata was invited to conduct telemanipulation of the ETS-VII robot arm. He operated the shuttle manipulator to

recover a free flyer from orbit in 199 and his skill to operate the shuttle manipulator is highly recognized within NASA. His given task in the telemanipulation on ETS-VII was to trace surface of the experimental equipment on ETS-VII by the onboard robot arm keeping the push down force around 20 Newton in the telemanipulation mode. With help of the compliance control of the onboard robot arm, he conducted the task very smoothly, even though he could spent only two days for training including lecture and training on the robot teleoperation system. This shows ETS-VII robot system is a user-friendly system, which is easy to learn and to operate.

4. Coordinated satellite attitude & robot arm control

The mass of the ETS-VII chaser satellite is about 2.5t. The ETS-VII's robot arm handles payloads of a few kg to 400kg (target satellite). Attitude of the satellite platform must be maintained within a few tenth degrees by the reaction wheels and the gas jet thrusters even against the robot arm's reaction. This is to maintain the communication link through the data relay satellite and to generate electrical power from its solar arrays. However, if the reaction of the robot arm motion is too large, the satellite attitude control system can not maintain the proper satellite attitude. Therefore, the coordinated control of the satellite attitude and the robot arm is realized through the coordination of the onboard satellite attitude control system, onboard robot control system, and the on-ground robot control system. Fig.8 shows this coordinated control system.

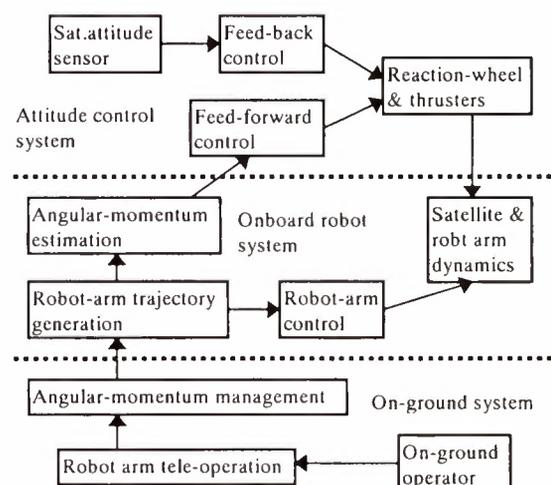


Fig. 8 Coordinated satellite and robot control system

The onboard robot control system estimates the angular momentum that the planned or commanded robot arm motion will produce. This estimated angular momentum is provided to the onboard satellite attitude control system to conduct the feed-forward angular momentum compensation. The on-ground robot control system also estimates the angular momentum which the planned robot arm motion will produce. If the estimated angular momentum is too large for the satellite attitude

control, then the planned or instructed robot arm motion will be canceled or modified to prevent the excess satellite motion beyond the capability of the satellite attitude control system. This assures satellite attitude stability even if the robot arm motion is instructed by telemanipulation which the robot control system can not predict its motion. Detail of the coordinated satellite attitude and robot arm control is shown in Ref.17-19.

4.1 Handling target satellite by the robot arm

An experiment to handle the target satellite by the onboard robot arm was conducted. One of the aims of the experiment was to test the coordinate satellite attitude and robot arm control capability. Since mass of the chaser satellite is about 2.5t and that of the target is 0.4t. Therefore the reaction from the robot arm when it manipulate the target satellite is large enough to disturb stability of the chaser satellite's attitude. Photo-4 shows the experiment. Images are from onboard cameras and the computer graphics, which animates the robot arm and satellite motions. The robot arm moved the target satellite to disconnect from the chaser satellite and then moved back to the docking position. The first move was under the traditional satellite attitude control and the latter was under the coordinated satellite attitude and robot control. It is clear that the coordinated control worked well.

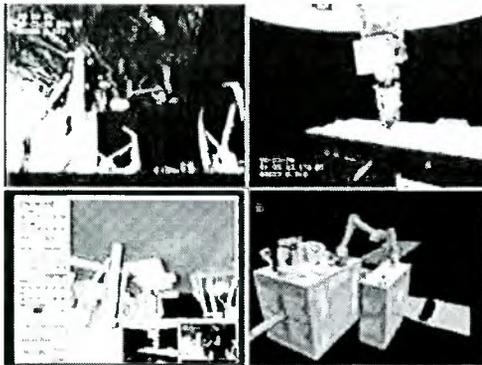


Photo- 4 Target satellite handling experiment

5. Conclusion

In this paper, we introduced an over view of the project and experiment results of the ETS-VII which is conducted the unmanned automated rendezvous docking experiments and the second generation space robot experiments. The most up-to-date information on ETS-VII experiment will be got on the Internet at "<http://oss1.tksc.nasda.go.jp/ets-7/>".

References

- (1) R.Ravindran, K.H.Doetsch, "Design aspect of the shuttle remote manipulator system", Proc. of the AIAA 1982 Guidance and control conference, pp.456-465, AIAA-82-1581, 1982
- (2) D.A.Basesett, et al., "Canadian Space Agency Mobile Servicing System Assembly, checkout and operation", Proc. of the 46th International astronautical congress, Oslo, Norway, Oct.2-6,1995, IAF-95-T.3.004
- (3) T.Matsueda, et al., "Development of Japanese Experiment Module Remote Manipulator System", Proc. of the 3rd International conference on artificial intelligence, robotics and automation in space (i-SAIRAS'94), Pasadena, CA., USA, Oct.18020,1994, pp.183-186
- (4) T.Iwata, M.Oda, et al., "Development of the second generation space robot", The 40th International Astronautical Congress, 1989, IAF-89-051
- (5) K.Machida, H.Nishida and K.Akita, "Precise Telerobotic System for Space Experiment on ETS-VII," IAF-98-U.5.05, 49th Int. Astronautical Congress, pp.1-10, (1998)
- (6) K.Matsumoto, et al., "Truss structure telemanipulation experiment using ETS-VII", Proc. of the 3rd int. symp. on artificial intelligence, robotics and automation in space, Oct.18-20,1994, Pasadena,CA,USA,pp.275-278
- (7) S. Kimura, T. Okuyama, Y. Yamana, Y. Nagai, and H. Morikawa: Teleoperation System for Antenna Assembly by Space Robots, Telemanipulator and Telepresence Technologies V, ed. M. R. Stein, SPIE, 14-23 (1998)
- (8) I. Kawano et al., "First result of autonomous rendezvous docking experiments on NASDA's ETS-VII satellite", Proc. of 49th International Astronautical Congress, Sept.28-Oct.2,1998, Melbourne, Australia, IAF-98-A3.09
- (9) M.Mokuno I.Kawano and T.Kasai, "Experimental Results Of Autonomous Rendezvous Docking On Japanese ETS-VII Satellite", 22nd AAS GN&C Conf., February 1999, Breckenridge, USA.
- (10) M.Nishida and M.Oda, "Space telerobot experiment system based on NASDA's ETS-VII satellite", Proc. of AIAA Guidance and Control Conference, Aug. 1997.
- (11) M.Oda, et al.; "Development of the EVA End Effector, Grapple Fixtures and Tools for the Satellite-Mounted Robot System", Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS'96), Nov.1996, Osaka, Japan, pp.1536-1543
- (12) P.G.Backes, et al., "A Local-Remote Telerobot System for Time-delayed Traded and Shared Control", Proc. of the 1991 Int. Conf. on Advanced Robotics, 1992
- (13) W.Kim, et al., "Force-reflecting and Shared Compliant Control in Operating Telemanipulators with Time Delay", IEEE Trans. Robotics and Automation, Vol.8, No.2, April 1992, pp.176-185
- (14) J.Funda, et al., "Teleprogramming: Toward Delayed Invariant Remote Manipulation", Presence, Vol.1, No.1 MIT Press, 1992
- (15) M.Oda., "System engineering approach in designing teleoperation system of the ETS-VII robot experiment system", Proc. of IEEE International Conf. On Robotics and Automation (ICRA97), pp.3054-3061, Apr.20-25, 1997, Albuquerque, New Mexico, USA
- (16) M.Oda and T.Doi, "Teleoperation system of ETS-VII robot experiment system", Proc. of IEEE/RSJ International Conf. on Intelligent Robotics and Systems (IROS'97), pp.1644-1650, Sept.7-11, 1997, Grenoble, France
- (17) M.Oda, "Coordinated Control of Spacecraft Attitude and its Manipulator", Proc. of 1996 IEEE Int. Conf. on Robotics and Automation", April 22-28, 1996, Minneapolis, USA, pp.732-738
- (18) M.Oda and Y.Ohkami, "Coordinated Control of Spacecraft Attitude and Space Manipulators", Control Engineering Practices, Elsevier Science Ltd.,
- (19) M.Oda, "Motion control of the satellite mounted robot arm which assures satellite attitude stability", Acta Astronautica Vol.4, No.11, pp.739-750, 1997

ANTENNA-ASSEMBLY EXPERIMENTS USING ETS-VII

Shinichi Kimura and Shigeru Tsuchiya

Communications Research Laboratory

4-2-1 Nukui-kitamachi, Koganei, Tokyo 184-8795, Japan

phone: +81-42-327-7514, fax: +81-42-327-6699, e-mail: shin@crl.go.jp, tsuchi@crl.go.jp

ABSTRACT

The construction of large and precise antennas is one of the most important challenges in the development of space infrastructures. Using robots is an effective method for constructing very large and precise antennas in space. The Communication Research Laboratory (CRL) developed an antenna designed to be assembled in space for use in future space communication infrastructures and has been studying teleoperation technologies for controlling space robots. The initial experiments for testing the assembly of this antenna in space are being conducted using Engineering Test Satellite VII (ETS-VII), which was launched in November 1997. ETS-VII which is the first robotic satellite was developed by the National Space Development Agency of Japan (NASDA); it is equipped with six-DOF manipulators and antenna-assembling mechanisms developed by the CRL. Here, we briefly describe these experiments, including master-slave experiments using space robots, and our evaluation of the audio feedback system, which uses an eye-mark recorder.

1. INTRODUCTION

Antenna assembly will be an important application of robotics in constructing future space infrastructures. Using robots to assemble antennas is an attractive alternative to using inflatable and deployable antennas as a way to construct large-scale antennas on space stations, space platforms, planetary bases, etc. This is because

- (1) they enable antennas to be constructed using prefabricated parts,
- (2) they enable antennas to be constructed using a limited number of launchers by reusing the launchers and assembling the antenna in stages,
- (3) the sections of the antenna can be easily collimated after they are assembled, and
- (4) such antennas can be easily repaired by simply replac-

ing the defective components.

At the Communications Research Laboratory (CRL), we have developed an antenna designed to be assembled in space to investigate the technologies needed for constructing large space antennas[1]-[4]. We are currently testing the assembly of this antenna in space by using Engineering Test Satellite VII (ETS-VII), which is Japan's first robotic satellite. Our aim is to evaluate the mechanisms we designed for achieving sub-millimeter-accuracy assembly and teleoperation. After basic checkout of the satellite, 16 assembly experiments were performed by the CRL in cooperation with the National Space Development Agency of Japan (NASDA). The basic functions of the assembly mechanisms and teleoperation system were confirmed. In this paper, we describe our antenna-assembly experiments using ETS-VII and present some of the experimental results.

2. ANTENNA ASSEMBLY EXPERIMENTS USING ETS-VII

2.1 ETS-VII

ETS-VII is an engineering test satellite developed by NASDA. It was launched aboard an H-II rocket in 1997. It weighs 2.8 t and has a circular orbit with an altitude of 550 km. It was designed to test two technologies- rendezvous docking and space robotics-using a 2-m-long 6-DOF (degrees of freedom) robotic arm. The space robotics experiments included "Advanced Robotic Hand", planned by the Ministry of International Trade and Industry; "Deployable/Detachable Truss Assembly", planned by the National Aerospace Laboratory; and "Experiment on Antenna Assembling Mechanism", planned by the CRL to test mechanisms for assembling structures in space and to study the basic technologies of space robotics needed to assemble an antenna by using a satellite.

The robotic arm is controlled remotely from NASDA's

Tsukuba Space Center via intersatellite communication through NASA's TDRS. The telemetry information is delivered in 68-byte telemetry packets at 10 Hz through a 9600-bps RS-232C line; it includes the end-effector position, joint angles, force, torque, and status information. Commands for the robotic arm are transmitted at a maximum of 4 Hz through a different 9600-bps RS-232C line. The robotic arm can be moved in not only position-control mode, but also in force-control mode, such as compliance mode or active-limb mode. The time delay (round trip time) is about 6 seconds. Images are obtained by on-board cameras and delivered through two NTSC lines. The images are updated every 2 Hz over each line.

2.2 Antenna-Assembling Mechanism

The antenna-assembling mechanism (AAM) we designed (Fig. 1) consists of three parts: a "Fixed Part (FP)", a "Combining Part (CP)", and a "Catcher for Emergency". The "Fixed Part" and the "Catcher for Emergency" are attached to the outside of the satellite, and the "Combining Part" is connected by a latch to the "Fixed Part". In the experiments, the antenna-assembly process was simulated by disassembling then reassembling the connection between the FP and CP. The CP is captured by the robotic arm by using a grapple. Latching is performed by a very simple motion of the robotic arm. When the latch mechanism is moved straight toward the latch pin, the stopper of a rotary camera, which is moved by a spring, is snapped off and automatically latched. This latching mechanism does not need a power supply, so it has a virtually unlimited lifetime.

To achieve accurate and reliable assembly, the AAM has two adjusting mechanisms. One is a visual guidance mechanism that uses a target mark, and the other is a mechanical guidance mechanism that uses a guide cone (Fig. 1). The image view, which is captured by a camera mounted on the hand of the robotic arm, changes according to the relative drift between the CP and FP. This drift is automatically calculated by analyzing the image of the target mark. The guide cone and compliance mechanism, which is a spring system located between the grapple and the CP, adjust the vibrations between the CP and FP and adjust for the small drift that cannot be handled by the visual guidance mechanism. These mechanisms are important not only for assembly using teleoperation, but also

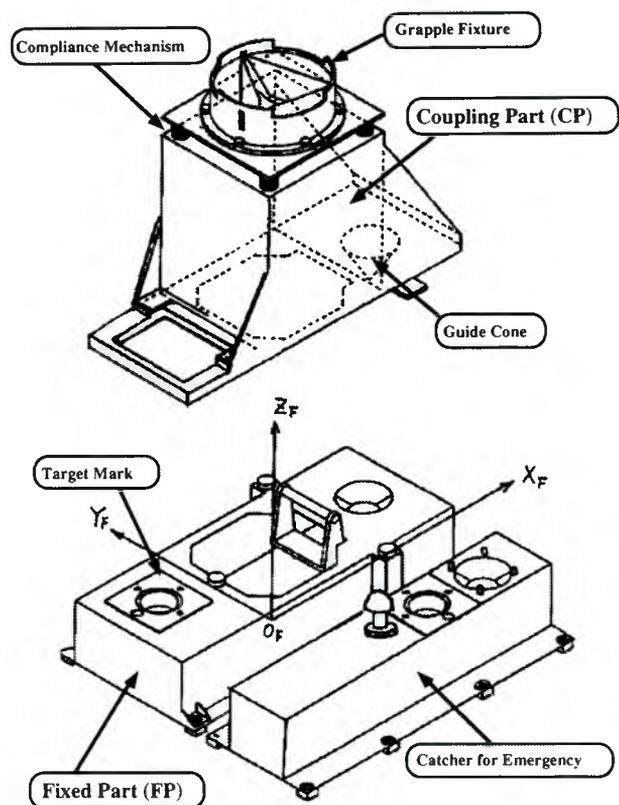


Fig. 1: Antenna-assembling Mechanism (AAM)

for fully automatic assembly.

2.3 Teleoperation system

Given the circumstances under which space robots operate, it is essential to generate motion commands safely and reliably. However, because the duration of each experiment conducted using ETS-VII is limited to 20 minutes by the orbital movement of the satellite, the operator needs to generate motion commands swiftly and efficiently. Therefore, we developed a teleoperation system for our antenna-assembly experiments based on the following considerations in order to achieve reliable and efficient operation (Fig. 2).

1) Semi-automated document processing.

A macro program written in Japanese is automatically converted to SOP, SOE, robot language, and the language of the operation database.

2) Library of motions

Verified motions are stocked as parts of the macro language.

3) Multi-modal interface of telemetry data

Force & tactile (master arm)

Audio (status: voice, force & torque, motor-noise tone)

Visual (downlinked & processed images, 3D simulation models)

The equipment for the telemetry audio interface is quite simple. An RS-232C interface between the interface computer and NASDA's operations system is diplexed to a personal computer called a telemetry analyzer. When a status change in the telemetry is detected by this analyzer, it gives a spoken warning. The magnitude of force on the end-effector is also presented as a motor-noise tone synthesized by an audio sampler. The operator can thus easily recognize the state of the robotic arm without having to pay much attention to the telemetry display.

3. RESULTS

3.1 Basic Assembling Process and Assembling in Compliance Control Mode

The basic disassembling-assembling experiments were conducted using the following processes, as illustrated in Figure 3, which shows downlinked images during the experiments. (1) The robotic arm captures the fixture mounted on the AAM-CP (Capture). (2) The rotary cum of the AAM-CP is rotated by a socket wrench installed on the end-effector of the robotic arm (Unlatch). (3) The robotic arm moves the AAM-CP up from the AAM-FP (Disassemble). (4) The robotic arm positions the AAM-CP at the assembly position by using the visual guidance provided by the Target Mark (Position). (5) The robotic arm moves the AAM-CP straight towards the AAM-FP and the latch

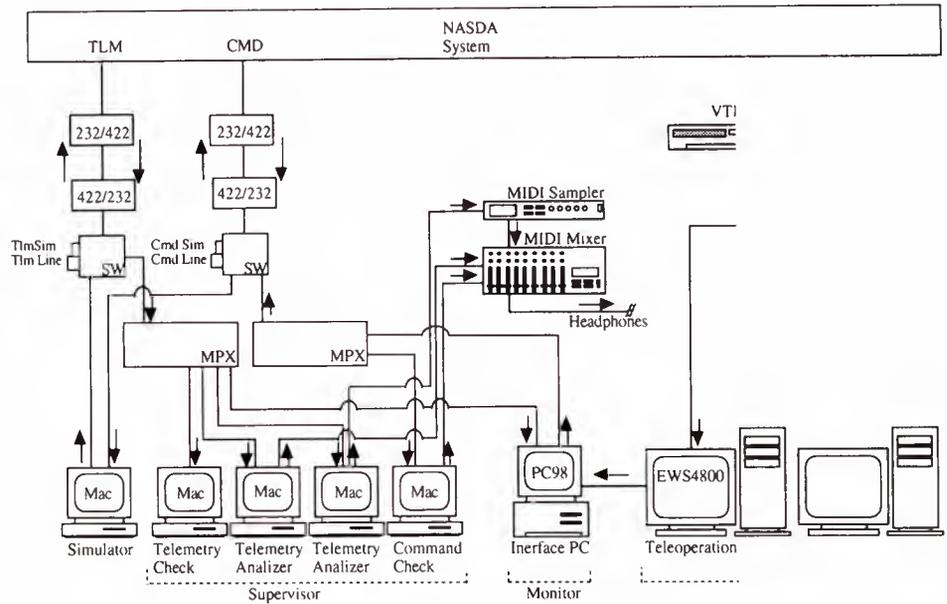


Fig. 2: Teleoperation System

mechanism automatically connects the AAM-CP to the AAM-FP (Reassemble). (6) The robotic arm releases the AAM-CP fixture (Release).

Figure 4 shows the reactive force at the end-effector of the robotic arm during disassembly and assembly. The AAM was disassembled in about 2430 seconds and reassembled in about 2480 seconds. The assembly was done using very simple operations in which the robotic arm moved the AAM-CP toward the AAM-FP. The AAM was reliably reassembled using a very small force (a maxi-

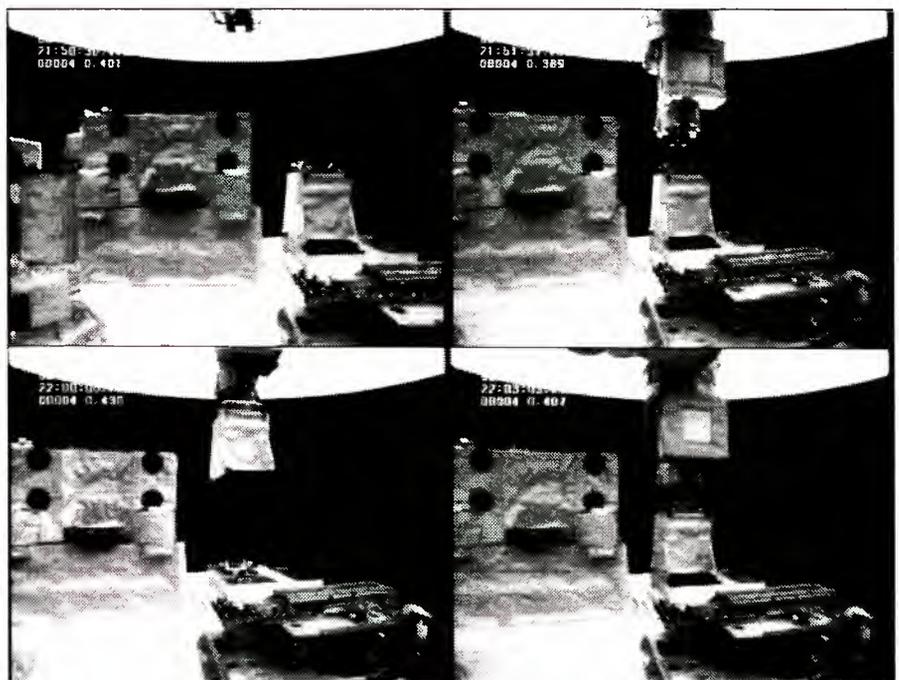


Fig. 3: Downlink Images during assemble-Disassemble Process

imum of 8 N). The disassembly-reassembly process was performed using the satellite 40 times and no mechanical trouble occurred. The mechanical performance of the AAM was very good and did not vary over the year and a half of testing.

3.2 Assembling in Compliance-control and Non-compliance Modes

The robotic arm of ETS-VII supports force feedback control modes, such as compliance-control mode and active-limb mode. In compliance-control mode, the end-effector moves flexibly according to the following control algorithm, which takes into consideration the force and torque at the end-effector.

$$X = F / (MS^2 + CS + K)$$

$$R = T / (MS^2 + CS + K)$$

The X and R are the shifts in the end-effector, and the F and T are the force and torque on the end effector, respectively. Parameter S is the time interval of the control system. The compliance-control mode was quite effective for performing the capture tasks.

The antenna-assembling mechanism also contains the mechanical compliance system. This system enables easy and reliable assembly using the robotic arm, which alone is low cost, less accurate, and unable to support compliance mode. The system is effective for following very fast and unexpected motions, whereas software compliance, such as the compliance control of ETS-VII, cannot follow motions much faster than the control cycle of the on-board controller, whose calculation ability is limited. If the compliance control is effective, the maximum speed of the robotic arm is 2 mm/s. However, with software compliance it is possible to change the compliance parameters, whereas the mechanical compliance system must be tuned for to the task beforehand. Two compliance-control parameters ("Flex" and "Hard") are available for each axis of the robotic arm.

We conducted an experiment to evaluate assembly using the compliance-control mode and confirmed that effective compliance parameters were generated for the antenna-assembly process. This experiment is important for understanding the detailed assembly dynamics of the antenna-assembling mechanism.

In all cases, harmful, unstable motion was not observed, so the mechanical compliance does not conflict

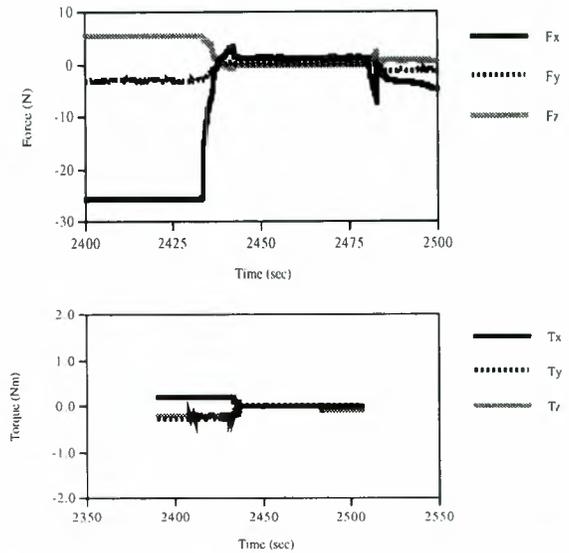


Fig. 4: Force and Torque at the End-Effector during Assembling- Disassembling Process

with the software compliance. However, one of the assembly status indicators did not change regularly after assembly using the flex-compliance mode. The end-effector position was shifted during assembly, whereas the shift of the attitude was negligible. The X axis is the axis along the length of the AAM, the y axis is the axis along the width of the AAM, and the z axis is the approach axis. Rotation around the z, y, and x axes indicates, respectively, roll, pitch, and yaw. To confirm these results, we evaluated using the flex-compliance mode along the XYZ axis and the same mode only along the Z axis. The assembly status did not change regularly after the assembly along the XYZ axis, whereas it did after the assembly along the Z axis. This suggests that the compliance parameter should be "hard" for the antenna-assembling mechanism.

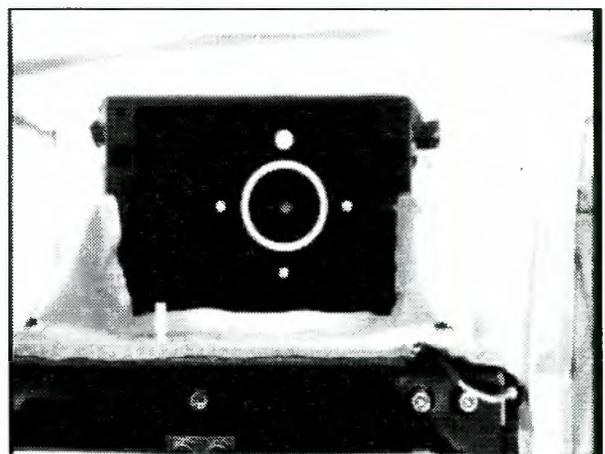


Fig. 5: Downlink Image of Target Mark through Hand Camera

3.2 Measurement of Combinable Area

To check the mechanical mechanism for adjusting the combining position, the combining position was deliberately shifted from the proper one. The AAM was designed to adjust the combining position even when it is out of position by up to 6 mm in any direction and by an altitude of up to 2 degrees in any direction. Considering the accuracy of the robotic arm for positioning, ten cases, namely x and y axis shifts of ± 5 mm and roll, pitch and yaw shifts of ± 1 degree, were checked. In all cases, the mechanical compliance system adjusted the combining position effectively, and the assembly was performed successfully and reliably. The force needed for assembly never exceeded 8 N. These experiments confirmed the abilities of the AAM's mechanical adjusting mechanism.

3.3 Check of Auto-Positioning Using Image Processing

To perform fully automatic assembly, the teleoperation system for the antenna assembly experiment has an auto-calibration function that uses the downlinked image of the "Target Mark" on the AAM-FP (shown in Fig. 1). This image is taken by the camera mounted on the robot hand, so it is affected by the position and attitude of the robotic arm. The teleoperation system calibrates the robotic arm position from the downlinked image and automatically adjusts the position of the arm.

As the first step in testing fully automatic assembly, we checked the auto-positioning function of the teleoperation system at five positions. The calibration error in the x - y direction was 2.26 mm in the worst case, which is much lower than the 5 mm required for fully automatic assembly; that in the z direction was 1.85 mm in the

worst case, which is also much lower than the tolerance of 10 mm. Figure 5 shows a downlinked image of the "Target Mark" used for checking the auto-positioning function. These results suggest that the auto-positioning function of the teleoperation system worked well and is suitable for fully automatic assembly.

3.4 Checkout of Master-Slave Control

The CRL and NEC planed to perform experiments on advanced control of space robots using a master arm. [5]-[6] For the first evaluation of the master-slave control system, we checked the stability and effectiveness of the master-slave control system during simple operations. This is the first case in which master-slave control was utilized for space robots. Figure 6 illustrates how the operation works. The estimated bilateral control function was confirmed: the end-effector was successfully guided to the top of the cone. This guidance increases the operation efficiency. These results suggest that the master-slave control system is quite stable and effective.

3.5 Evaluation of Audio Feedback System using Eye Mark Recorder

To evaluate the effectiveness of the audio feedback system, the operator's viewpoints were recorded by using an eye-mark recorder during the capture task with and without audio feedback (Fig. 7). The time required to cap-

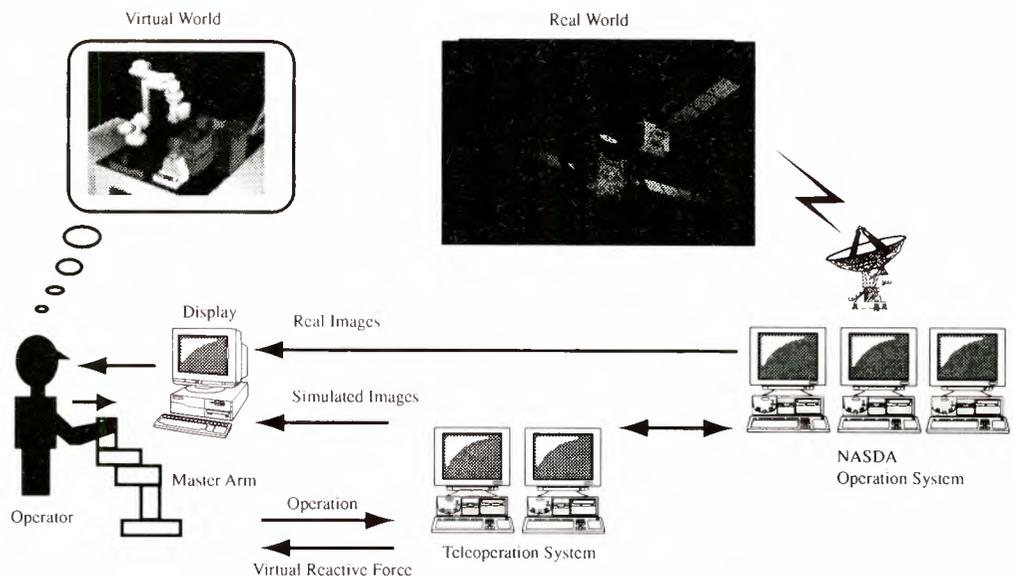


Fig. 6: Virtual Bilateral Control

ture the target when using audio feedback was shorter than when audio feedback was not used. When audio feedback was used, the operators paid attention to various information inputs, including the downlinked images, while without audio feedback, the operators primarily concentrated on watching the telemetry. These results suggest that the audio feedback system is quite effective because it reduces the need to concentrate on the telemetry data. We are now continuing our experiments and analyzing the use of the audio feedback system and the eye-mark recorder.

4. CONCLUSION

We have described the basic antenna-assembly experiments we conducted using ETS-VII and presented some initial results. The latch mechanism of the antenna-assembling mechanism, the mechanical compliance system, and the visual guidance system were all successfully checked out and their effectiveness was confirmed. The antenna-assembling mechanism was successfully assembled when there was displacement of up to 5 mm in position and 1 degree in attitude by using its own adjustment mechanism. We plan to perform further experiments on teleoperation at later stages of the satellite's mission.

References

- [1] T. Iida, K. Okamoto, Y. Ohkami, S. Kibe, H. Koshishi, M. Naka, and H. Yamamoto, "A Space Station Experiment on Large Antenna Assembly and Measurement", *Acta. Astronautica*, 12, pp. 335-344 (1985).
 - [2] Y. Suzuki, S. Kimura, T. Takahashi, K. Nakamura, and H. Morikawa, "Antenna-Assembling Mechanism Test on ETS-VII", In *Proceedings of Third International Symposium on Artificial Intelligence, Robotics and Automation for Space*, Pasadena, U.S.A., pp. 279-284 (1994).
 - [3] T. Okuyama, S. Kimura, and M. Takahashi, "Teleoperation System for Experiment on Antenna-Assembling Mechanism on ETS-VII", In *Proceedings of Fourth International Symposium on Artificial Intelligence, Robotics and Automation for Space*, Tokyo, Japan., pp. 69-72 (1997).
 - [4] S. Kimura, T. Okuyama, and H. Morikawa, "Teleoperation of Space Robots for Antenna Assembling Experiment on ETS-VII", *Space Ops* 98, 2a007 (1998).
 - [5] H. Morikawa and N. Takanashi, "Ground Experiment System for Space Robots Based on Predictive Bilateral Control", *IEEE Int. Conf. on Robotics & Automation*, pp. 64-69 (1996).
 - [6] N. Takanashi et al., "A High-sample-rate robot control system using a DSP based numerical calculation engine", *IEEE Int. Conf. on Robotics & Automation*, pp. 1168-1173 (1989).
- S. Kimura, T. Okuyama, Y. Yamana, Y. Nagai, and H. Morikawa, "Teleoperation System for Antenna Assembly by Space Robots", *Telemanipulator and Telesence Technologies V*, ed. M. R. Stein, SPIE, 14-23 (1998).

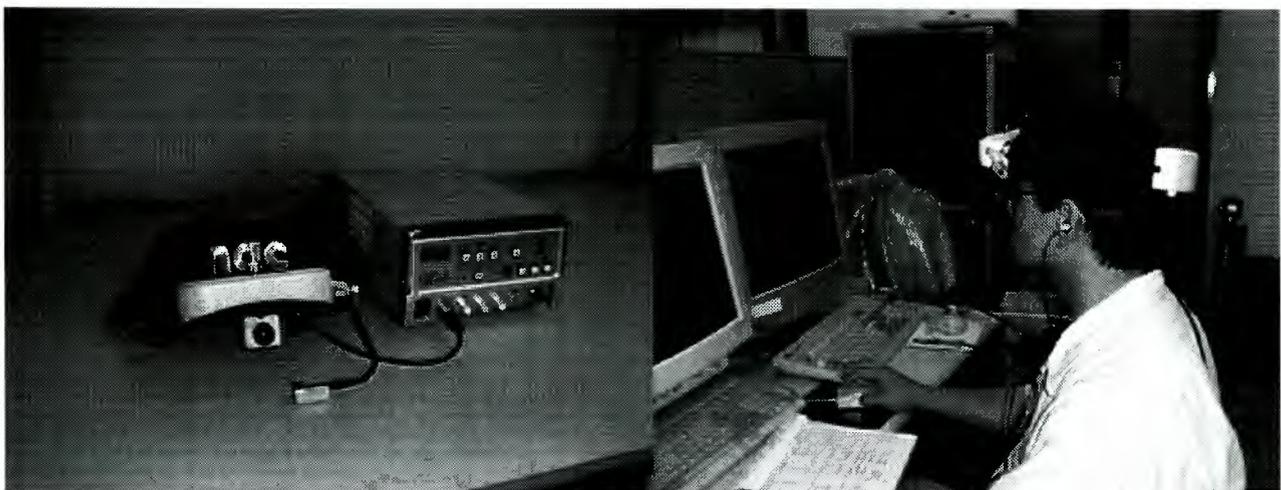


Fig. 7: Evaluation of the Teleoperation System Using Eye Mark Recorder

TELEOPERATION CONTROL OF ETS-7 ROBOT ARM FOR ON-ORBIT TRUSS CONSTRUCTION

Kohtaro Matsumoto, Sachiko Wakabayashi, Luis F Penin, Masahiro Nohmi

Space Technology Research Group, National Aerospace Laboratory
7-44-1, Jindaiji-higashi, Chofu, Tokyo, 182-8522, Japan
phone: +81-422-40-3170, fax:+81-422-40-3149, Email: matumoto@nal.go.jp

Hiroshi Ueno, Tetsuji Yoshida, Yuhtaro Fukase

Space Systems Division, Shimizu Corporation
2-3, Shibaura 1-chome, Minato-ku, Tokyo, 105-07, Japan
phone:+81-3-5441-8951, fax:+81-3-5441-8955

ABSTRACT

In this paper, we will present an overview and the results of the teleoperation experiments of our truss structure experiment on the Engineering Test Satellite 7, that have been done from last March 1998. The major purpose of our experiment is to establish the basic and advanced teleoperation technology that will be useful for on-orbit truss construction for the future large scale space system.

1. INTRODUCTION

At Nov. 28, 1997, NASDA launched the ETS-7 (Engineering Test Satellite 7) (Fig.1). It has been developed to demonstrate two major missions for the Japanese future space activities, these are the rendezvous docking and the space robotics.

For ETS-7, NASDA has developed the baseline teleoperation robot systems on the satellite, such as the arm, vision, communication, controller, and the ground facility¹⁰⁾. For this space robotics experiment, other three national institutes, ETL/MITI, CRL¹¹⁾ and NAL, have cooperated with each experiment, with own experiment apparatuses on the satellite, and with own ground facility for the ground teleoperation experiment.

We, NAL, have implemented the TSE (Truss Structure Experiment) components on the ETS-7, and the

ground teleoperation facility at Tsukuba Space Center. Until the last May 1999, we have finished almost all TSE planned experiments, and are preparing the phase 2 advanced experiments in the rest of this year.

2. GROUND TELE-OPERATION OF DEXTEROUS TASKS BY SPACE ROBOT

From the viewpoint of the expectation for the space robots¹⁾, current space robots have two problems, those are the lack of the ground teleoperation, and the lack of the dexterity of their works.

Even in the space station era, the astronauts' on orbit manipulation are the primary operation for the station robots. The main reason is the lack of the technological maturity of the teleoperation, although the main tasks of the station robots will be the handling of the modules, that are designed for the robot works.

Most of the EVA tasks, such as the repair or assembling, will still remain for the human tasks, since those tasks require the dexterous human operation. For the Hubble telescope repair mission, the space robot possibility was also considered instead of the human repair, and was not implemented because of the lack of maturity of the dexterous tasks.

We have selected the truss structure tele-handling tasks as the model tasks on the ETS-7 for the future space robots, since the truss structure and its joints,



Fig.1 ETS-7 before Launch

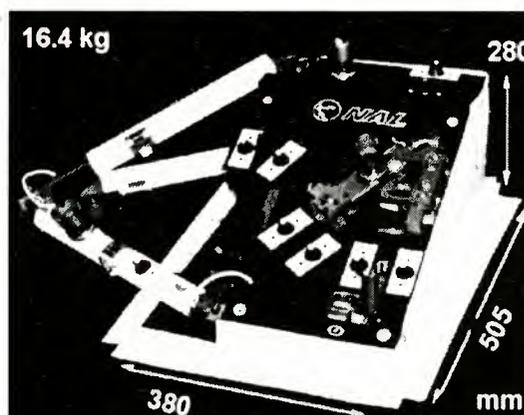


Fig.2 TSE (Truss Structure Experiment)

require the dexterous handling capability of a small target in its teleoperation⁵⁾. The next generation space station, the solar power satellite, and the space hotel in the far future, will essentially require the space robot as its constructors to solve the large scale assembling work requirement.

3. ETS-7 ROBOT ARM AND ITS TELEOPERATION

The ETS-7 robot arm was introduced from the JEM small fine arm with a little modification in its length, the tool, and controller. Its tip position accuracy is quite worse than the ground robot in a factory, because of the vacuum lubrication. For the work monitoring, a pair of hand eye and a pair of shoulder camera were implemented, with compressed monochrome JPEG at 4 Hz. The endeffector tool has two modes of capturing the grapple fixtures. The finger open operation captures the standard GPF-S and GPF-M. The finger close operation captures the GPF-N that is designed for TSE. (Fig. 6)

The ETS-7 teleoperation has two modes; those are the program control and the direct teleoperation control. Before sending the program control command to the ETS-7, NASDA's ground facility checks its safety verification using the simulator. In the direct teleoperation control, the arm tip motion will be controlled at 4 Hz directly from NAL's ground teleoperation facility. On-line verification against collision, singular attitude, speeds, acceleration, and so on is done within NAL's facility.

For the force control, the ETS-7 arm has four modes; those are position control, compliance control, active limp control, and force command control.

4. TSE: TRUSS STRUCTURE EXPERIMENT

Our TSE (Truss Structure Experiment apparatus) on the ETS-7 is designed and developed to implement the two basic truss teleoperation works; those are the deployment of the truss structure and assemble of the truss joint. In addition to these tasks, TSE launch lock is also designed to be teleoperated by the ETS-7 robot. TSE is composed by three major parts. (Fig. 2)

(1)**Launch Lock (LL):** TSE LL, which locked the

deployable truss element and the truss assembly joint, was released by the 90 degrees ETS-7 arm tip rotation. The major difficulty of this task was the GPF-N capture itself, since the capture task was the first operation for NAL, without any trial or experience at the ground.

(2)**Deployable Truss structure (DT):** The latch component of DT is same as the DTB (Deployable Test Bed)²⁾ for JEM. The DT is one section of a triangle truss structure that can be deployed and folded. (Fig. 4)

The arm deploys DT along a 3 dimensional spline curve under closed link movement (Fig. 3). The operational difficulty is to move the arm along the 3D trajectory within limited tip force and torque. The closed link movement along a strictly constrained trajectory is the first operation for ETS-7.

(3) **Truss assembly Joint (TJ):** TJ was originally designed for STAR*Bay-2 truss³⁾ to be operated by human. Its mechanism was modified for one hand robot operation without any hand-over.

TJ assembly task is similar to the "peg-in-hole", but harder because of its mechanical obstacles. During TJ assembly, even if it is positioned at the center of the inserting hole, TJ assembly task requires about 10-20N force to surmount the mechanical obstacles of the joint. A mechanical guide was also attached to the joint receiver (JR) to compensate the insufficient accuracy specification of ETS-7 arm tip control.⁵⁾ (Fig.5)

(4)**Grapple Fixture for NAL (GPF-N):** GPF-N is designed as the smallest grapple fixture only for TSE truss strut. The GPF-N diameter is only 38mm,

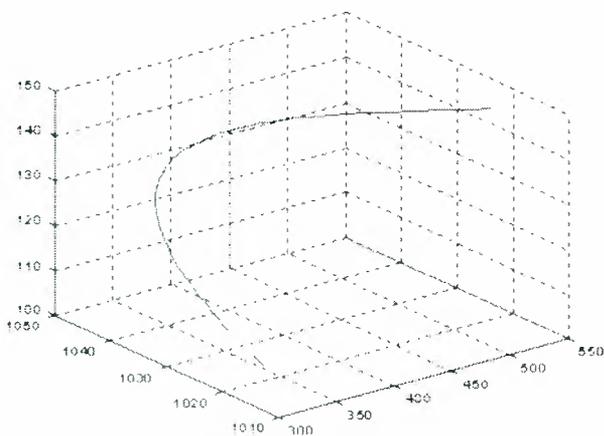


Fig. 3 3D Spline Trajectory of DT Deployment

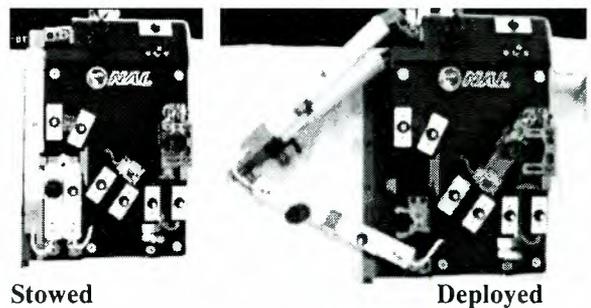


Fig.4 Truss Deployment and Stow

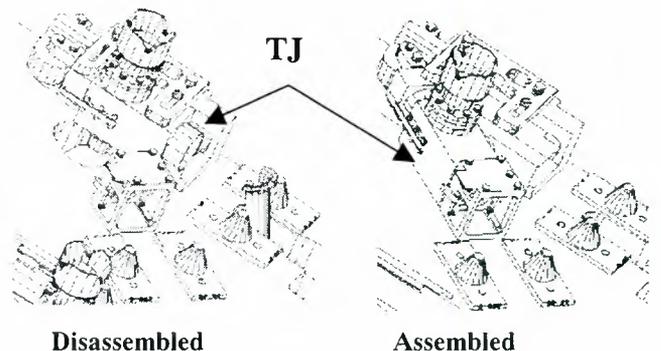
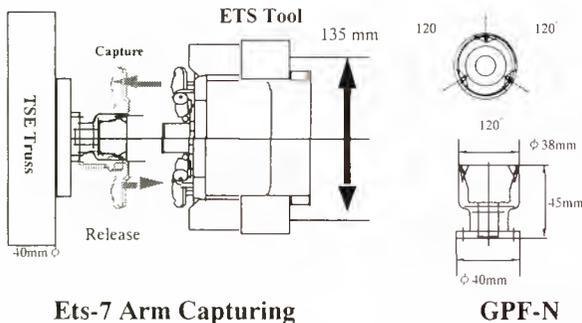


Fig.5 Truss Assembly Joint

although the diameter of other GPF-S&M is 138mm. The finger close capturing method is also different for GPF-N, as described before. (Fig. 6)



Ets-7 Arm Capturing

GPF-N

Fig.6 GPF-N and its Capture

5. NAL GROUND TELEOPERATION FACILITY

We have implemented our tele-operation facility using NASDA's ETS-7 robot facility as a transparent monitor. The major functions of our teleoperation facility are the followings; (Fig.9)

- (1) Generation of the program mode teleoperation command for the ETS-7 robot arm.
- (2) Generation of the direct teleoperation command at 4Hz with online safety verification.
- (3) Graphic simulation for the predictive display, and safety checking of collision and singular attitude.
- (4) Joystick for the direct teleoperation by human operator from NAL facility.
- (5) The precise measurement of the arm tip position and attitude by target marker image processing.
- (6) Hardware simulator of the ETS-7 robot arm with the TSE-EM (engineering model).
- (7) Advanced research software I/F.

The hardware simulator is used to verify every teleoperation procedure and algorithms, to train the operator before the real teleoperation, and to improve the programs, procedure, and/or operation skill under more realistic teleoperation environment than software

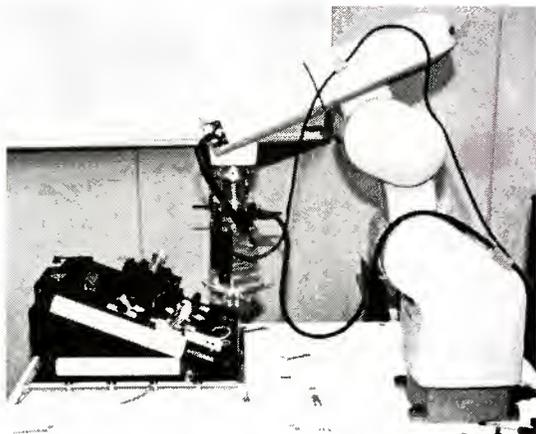


Fig.7 NAL-TSE Hardware Simulator

simulator. This simulator is composed of the industrial robot and the TSE-EM, and can partially simulate the motion and control of the ETS-7 arm. (Fig.7)

The advanced research software I/F: We prepared this I/F to enable the innovative telerobotics algorithm from the laboratory into the real space robot experiments with minimum and earliest preparation.

For the additional, but essential tasks, such as on-line safety check, arm setting, and human operator's override, the standard NAL facility's function is utilized to keep NAL facility's safety reliability.

The command and telemetry I/F: Two RS-232C lines with 9600 bps are used. One RS232C is used for teleoperation command at 4Hz, and others for the telemetry data distribution at 10Hz. On board TV images are delivered at 4Hz by NTSC. (Fig.8)

6. BASIC TELEOPERATION FOR TSE

For TSE teleoperation, we have implemented and tried the following methods as basic experiments.

- (1) **Program Mode Teleoperation:** Using the ETS-7's program control, every teleoperation command is programmed just like a "robot control language". Every arm tip motion is specified by a straight line movement controlled by the on-board arm controller.

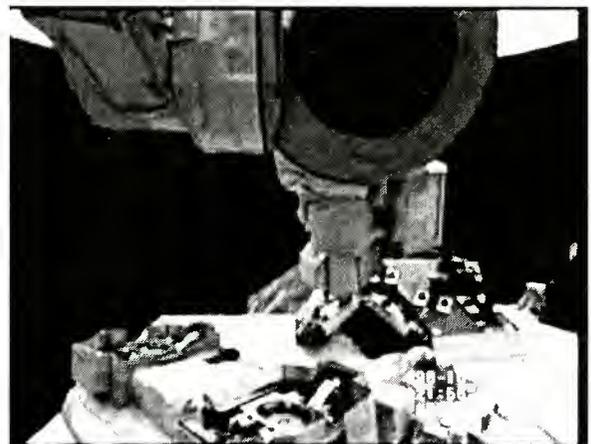


Fig.8 TSE from Shoulder Camera during Deployment



Fig.9 NAL Ground Teleoperation Facility

This control mode is the basic mode of ETS-7. However, since our TSE's tasks are basically closed link operations, this control mode has the following inconvenience for TSE teleoperation.

1. Discontinuous tip force control: After every tip movement, the arm controller will keep the final tip position based on the arm joint encoder. Thus, for the compliance force control tasks with relatively large working force, this discontinuity of tip force could not be ignored.
2. Safety check for the command trajectory: The verification time is linear to the length of the trajectory, and might increase the additional time delay for teleoperation.

The initial experiments were done only by this control mode. The initial LL release, DT deployment and stow, and TJ assemble and disassemble were done successfully. However, for TJ assembling we needed a slight change of assemble sequence to avoid the effect of the tip force discontinuity.

- (2) **Programmed direct teleoperation control**: We developed the program control language over the direct teleoperation control.

In this control mode, the continuous transfer between programmed control and Joystick (JS) direct teleoperation can be achieved. JS is used for the human operators to override to the programmed control. This mode could prevent the inconveniences of the program mode teleoperation, mentioned above.

The major problem of this control is;

1. Low control frequency: Because of the low command frequency, at 4Hz, and low telemetry data rate, at 10Hz, the tip motion is limited with relatively slow speed except when free tip motion.
2. Over safety for the time delay: Because of the long time delay between the ground and ETS-7, up to 5.5 sec, we have to use the over-safety criteria, that can assure the safety under the worst motion.

By this programmed direct teleoperation control, the DT deployment and stow, and the modified TJ assemble and disassemble were tried successfully with



Fig.10 Visual Aid using Predictive Force Method

better performance in its working force.

7. HUMAN DIRECT TELEOPERATION BY JOY-STICK FOR UNKNOWN ENVIRONMENT

In the programmed direct teleoperation control, the human operator overrides the operation by using JS and tries to adjust slight miss-match between the ground model and the on-orbit situation.

In addition to such override function, the major expectation of the human operator is its adaptive and flexible control capability under a situation of malfunction or unknown trajectory. However, if a working target has some closed link characteristics, such as our truss structure, its ground teleoperation becomes drastically complex for human teleoperation without a suitable operation aid system.

Thus in addition to the basic JS operation, we have developed the several advanced aid systems for the human direct teleoperation from the viewpoint of the "tip force", described in the followings.

(1) Visual Aid using Predictive Force Method¹⁰⁾

For the direct teleoperation of the unknown closed link trajectory, we have developed a new visual aid system that assists the human operator based on the "tangential direction" of the past trajectory and "predictive force". For the smooth closed link operation, we assume that the best/optimal direction of the operational force should be the tangential direction of the trajectory. (Fig.10)

The "tangential direction" can be defined from the past trajectory telemetry data, using various estimation method such as the least square approximation or the moving average. The "predictive force" is defined as the theoretical compliance force using the subtraction of the arm tip positions from the current command and the latest telemetry data.

In our aid method, only the past arm tip trajectory history and the last teleoperation command are used, and the design database is never used, although it is the essential parameter in a usual predictive aid system.

The appropriate joystick inputs have two components. One is in parallel with the tangential direction of the current point to apply the deployment force (tangential input). The other is in the vertical direction of the tan-

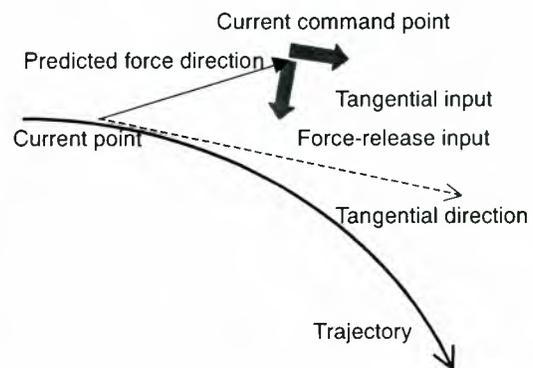


Fig. 11 Joystick Assists by Predictive Force

gential input to release the excessive force (force-release input). These diagonal input directions are converted into the joystick coordinate. By this aid system, the operator can easily handle the complex 3D spline closed link trajectory without move-and-wait, and within allowable force. (Fig.11)

(2)Ground Loop Tele-Control based on Predictive Force Method¹⁰⁾

We have extended this predictive force method to the ground loop control. When the operational force is specified, the target command point can be determined uniquely on the tangential line. The suitable next command is decided toward the target point within the arm tip speed limitation and control gain.

The predictive force is controlled within the specified force. The time delay might be the major disturbance for this control, but its effect is not so severe when the control gain is set as small enough. Furthermore, this control method might decrease the excessive operational force of the usual pre-programmed control based on the design data, because it essentially does not rely on any design model.

In Fig.12, the direct human teleoperation shows that our visual aid successfully assists the operator to trace the correct trajectory. And the result by the ground looped programmed predictive force control shows the successfully generation of the appropriate arm tip movement without any design data.

(3) Teleoperation through Virtual Force Reflection

It is well known that the direct force reflection (FR) (bilateral control scheme or force feedback) might be effective only with delays of less than 1 second and so with heavy degradation and is not suitable for the space robot teleoperation. However, the use of FR to the operator is expected to improve the human hand task performance. We have used a FR system to display virtual forces for the continuous teleoperation guide of on-orbit manipulators, with long communication time delay up to 7 seconds. A two DOF force reflecting joystick have been used as the FR hand controller (FR-HC).

The FR-HC guides the operator hand movement based

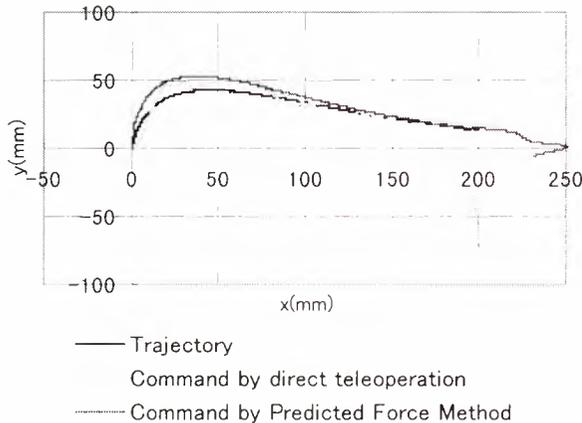


Fig. 12 Result of the Predicted Force Method Human JS Operation vs Programmed PFM

on the following virtual forces, instead of the real/telemetry force data. Using this FR guide, capturing the GPF-N and the TJ assembling were successfully completed.

a) Potential virtual force fields

In this FR, the CAD data is only used as the potential field to decrease the effect of inaccuracy and time delay, and to increase the human operator's judgement based on other data. The operator keeps the control during the movement, can stop it, reverse it or overrun the guiding force reflection. Using the tip position measurement, by the ground image processing, the operator will update the force potential field. (Fig.13)

b) Virtual force as physical constraints

The potential virtual force field is also generated from the closed link trajectory, that constraints the arm tip movement physically.

For these virtual physical constraints, FR robot model and "snapshot" procedure are developed to reflect the virtual force field to the operator, according to the real robot position in the work environment. (Fig.14)

c) Adaptive Virtual force by probing environment

Finally, since the TJ assembling looks like a too severe "peg-in-hole", the TJ insertion task requires more precise positioning than a known design database. Thus in this virtual force approach, we implement the probing the environment task before the insertion task itself.

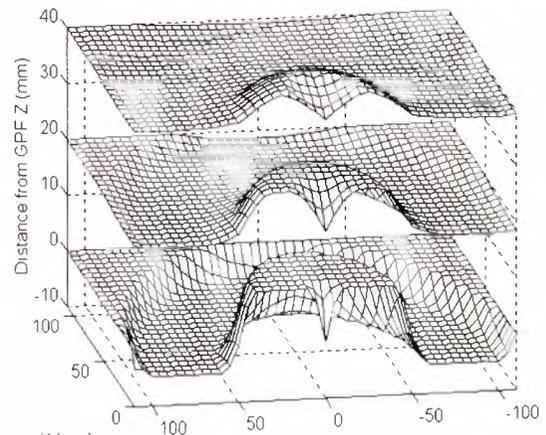


Fig. 13 Force field for GPF grasping

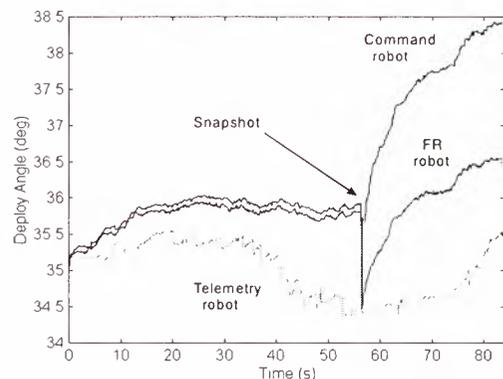


Fig.14 Snapshot during disassembly of the TJ

From the force and position telemetry data by the probing, an on-line map of the TJ peg-in-hole environment is built as a virtual force map to guide the operator through the task.

(4) Force Accommodation Control

In addition to the compliance control, ETS-VII robot arm can be controlled by the force/torque command, called "force accommodation control". In this control algorithm, the robot arm will continue to move until the external force/torque at the arm tip will reach the specified force/torque value.

Since the force accommodation control is implemented in the on-orbit controller, the following merits will be expected in teleoperation with time delay.

1. Excessive force and torque over the command can be suppressed.
2. Trajectory information is not essential.

Since the force accommodation control is fixed to the arm tip frame, the force direction deviates gradually from the correct trajectory during the truss deployment. Because of this trajectory deviation, the external force at the arm tip reaches the specified control force, and stops the robot movement. At this point the force accommodation control parameter shall be changed to the next one.

Fig.15 shows the result of deploying DT using the force accommodation control. When the telemetry force reaches the commanded force, the tip velocity becomes slow.

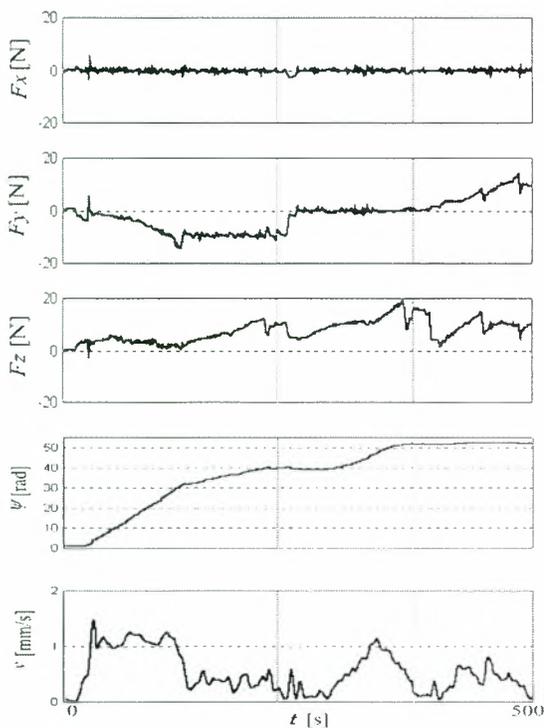


Fig.15 Experimental result of truss deployment

8. CONCLUDING REMARKS

Through the Truss Structure Experiments (TSE), the complex and dexterous space robot teleoperation for a small work object has been demonstrated on ETS-7. In the TSE experiments, two conventional teleoperation systems and three advanced aid systems have been tested and shown their better performance for the truss handling. Through the TSE experiment, the space robot capability and problems have been examined and demonstrated for the future truss structure construction.

The ground facility for the TSE experiments has been used and has shown its reliability for the space robot teleoperation. Through the TSE experiments, the advantage and the importance of the hardware simulator at the teleoperation facility are affirmed. The advanced research software I/F has also been utilized in the advanced teleoperation aid experiments.

For the phase 2 advanced experiments of ETS-7, we will continue to develop advanced teleoperation support technology, using the force feedback joysticks and other innovative aid systems until next November 1999.

REFERENCES

- [1]"Advancing Automation and Robotics Technology for the Space Station and for the U.S. Economy", NASA TM87566, 1985
- [2]S. Kibe, et al "Deployment/Retraction Support Mechanism of DTB", Proc. of 17th ISTS, 1990
- [3]T.Yoshida, K.Takagi,"Truss Joint System for Space Structures STAR*Bay 3", Proc. of 16th ISTS, 1988
- [4]S.Wakabayashi & K.Matsumoto, "Teleoperation with Additional Time Delay Mechanism for ETS-7 and JEM-EF", 47th IAF, Beijing, 1996
- [5]K.Matsumoto, et al, "Truss Structure Teleoperation Experiment by ETS-7 Robot", Proc. I-SAIRAS-97, Tokyo, 1997
- [6]T.B.Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis", IEEE Trans. On Robotics and Automation, Vol.9, No.5, Oct. 1993
- [7]S.Wakabayashi, K.Matsumoto, "Generalized Visual Aid for Direct Teleoperation Applied to ETS-7 Truss Manipulation Experiment", Proc. I-SAIRAS-99, Nordvijk, 1999
- [8]L.F.Penin, K.Matsumoto, S.Wakabayashi, "ETS-7 Robot Teleoperation through Virtual Force Reflection", Proc. I-SAIRAS-99, Nordvijk, 1999
- [9]M.Nohmi et al, "Truss Deployment by ETS-7 Robot Arm using Force Control", 16th Int. Symposium of Automation and Robotics in Construction. Madrid, Sept. 1999
- [10]M. Oda, "Space robot experiments on NASDA's ETS-VII satellite", Proc. of the 29th International Symp. on Robotics (ISR'98),1998, Birmingham, UK
- [11]S.Kimura, T.Okuyama, et al: Teleoperation System for Antenna Assembly by Space Robots, Telemanipulator and Tele-presence Technologies V, ed. M. R. Stein, SPIE, 1998

Performance Evaluation of Advanced Robotic Hand System in Space Experiment

Kazuo MACHIDA* (machida@etl.go.jp), Kenzo AKITA**

Keitaro OHNO***(ohno@stars.flab.fujitsu.co.jp),

Masayoshi MORIYA***, Hirotaka NISHIDA***, Tsutomu OHSAWA***

* : ETL, MITI, ** : Institute for Unmanned Space Experiment Free Flyer (USEF)

*** : Fujitsu Limited

Abstract

This paper presents the overview of a space test of the world's first precise extravehicular telerobotic system named ARH (the Advanced Robotic Hand system). It was boarded on Engineering Testing Satellite VII (ETS-VII) developed by NASDA, and was launched into low-earth orbit in November 1997. MITI/ETL has been conducting the researches on a precise telerobotic system. This time MITI along with the Institute for Unmanned Space Experiment Free Flyer (USEF) have developed the ARH system and carried out the space experiment on ETS-VII to prepare the robot technologies for efficient industrial utilization of space in the near future.

The objectives of the ARH space experiments are to evaluate the capability of the semidexterous robot hand for executing precise and delicate tasks and to validate the related technologies implemented in the system, which are multi-sensory, multi-DOF, and multi-finger control. Almost all of the experiments were carried out successfully and the results of them were found to be the expected ones.

1. Introduction

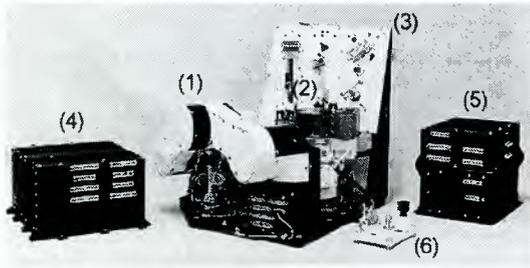
It has been a while since we human being started using a word like a space age. Space technology has been surprisingly developed and they say that it must not be just a dream for a person in the street to go for a space trip. International Space Station program is now ongoing and other such kind of space program which will have to be constructed or maintained on orbit will become accomplished in the near future. Such construction or maintenance so far has been achieved mainly as the U.S. space shuttle mission using space robot controlled by the crew on the space shuttle. Even the crew himself or herself sometimes has taken an extravehicular actions. It could be said that the extravehicular activities by a person are not only very dangerous but also expensive although it is supposed to be the most precise working action to take on orbit. Meanwhile the

space robot so far had no ability to handle rather small parts such as sample cartridges, bolts or electric connectors. If we human beings succeeded in developing such high performance space robot as an unmanned precise extravehicular telerobotic system, which is durable enough to be able to work for long time, then cost, time, and also jeopardy for the crew would be saved.

This kind of space robot should have an ability of precise work, for example, of carrying an instrument to an unexpected place or of replacing an abnormal instrument with new one handling some electric connectors and/or bolts. It would also be required some skills to handle a flexible object like wire, solar cell sheet, and even a floating object would be required to be grasped. Moreover, considering the insufficient tele-operability caused by the communication time lag, it should have adequate autonomy using several sensors as well as adequate performance of the ground system. The ARH system⁽¹⁾ was designed and fabricated to evaluate the capability of such kind of possible robot hand for precise work and to validate the related technologies we have been studied so far. In 1993, German program called ROTEX⁽²⁾ had succeeded in performing an onboard precise space robot experiment. The major differences between our ARH experiment and the ROTEX's are that the ARH is expected to i) be an extravehicular space robot, ii) work for long time up to one and a half years, iii) be a space robot of which the end effector or the hand can be attachable to the different kind of space robot arm on orbit.

2. System Description

The ARH system consists of a mini arm of around 0.7m, a hand which is able to be attached/detached to/from the mini arm, a task board, a control computer, a power unit, a task panel, and a ground operation system (see figure 1 and 2).



(1)mini arm, (2)hand, (3)task board, (4)control computer, (5)power unit, (6)task panel

Figure 1: ARH Onboard Components



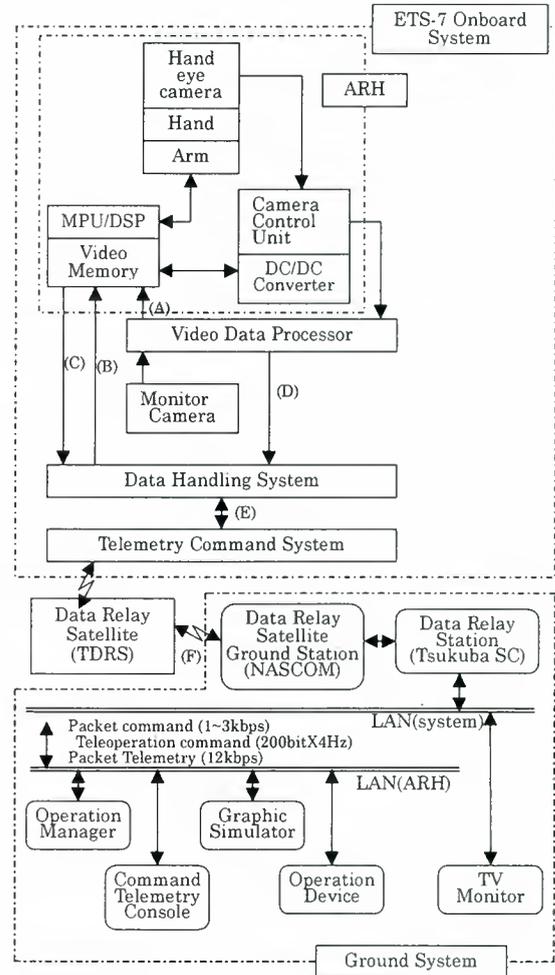
Figure 2: Ground System Configuration

Onboard experiment has been carried out in both configurations of i)stand-alone mode where the hand is attached to the ARH's mini arm and ii)long-arm-connected mode where the hand is connected to the NASDA's long arm called ERA⁽³⁾. The former mode was mainly focused on obtaining some expertise to control the precise semi-autonomous tele-robotics system. And the latter one was focused on the feasibility study of attaching the highly precise ARH's hand to the ERA in an attempt to give the ability of precise work to it, which has rather coarse positioning accuracy compared to the ARH's one.

The ground system was set up at NASDA Tsukuba Space Center and has controlled all of the ARH's tasks. The communication link between the ARH onboard system and its ground system has been established using the geo-stationary data relay satellite via the NASA Goddard Space Flight Center (see figure 3). The data rate of the uplink is around 4kbps and that of the downlink is around 1.5Mbps for which the video signal is dominant. The overall communication time lag is 5-6 seconds including uplink and downlink.

2.1. Mini Arm

The mini arm is a 5-degree-of-freedom



- (A) Hand eye camera image data 4f/s
- (B) Packet command
- (C) Packet telemetry
- (D) Compressed video data 1.2Mbps
- (E) Packet Data
- (F) Up-link 4kbps / Down-link 1.5Mbps

Figure 3: System Block Diagram

(5-DOF, R-P-P-P-R) space robot of around 0.7m and its positioning accuracy is around 1mm, allowing it to perform precise space experiments. It also has a 6-DOF force torque sensor to achieve delicate tasks. One of the distinctive features is that the hand could be attached/detached to/from this mini arm so that the hand could be attached to another 2m long arm or ERA of which the positioning accuracy is around 10mm. This reconfiguration was carried out as one of the ARH's extended experiment – long arm connected mode- described later (see chapter 3).

2.2. Hand

As is shown in figure 4, the hand⁽⁴⁾, which is the key component of the ARH system, has three fingers and four kinds of sensors for precise works. It has a hand-eye CCD camera and three LASER range finders as non-contact sensor for determination of working positions, and a wrist compliance sensor, and grip force

sensors as contact sensor for fine positioning and delicate force control in gripping, attaching, and detaching objects. Using this multi-sensory hand, ARH system can work as semi-autonomous robotic system as well as telerobotic system.

2.3. Task Board

As is shown in figure 5, the task board is an experiment panel for the evaluation of the

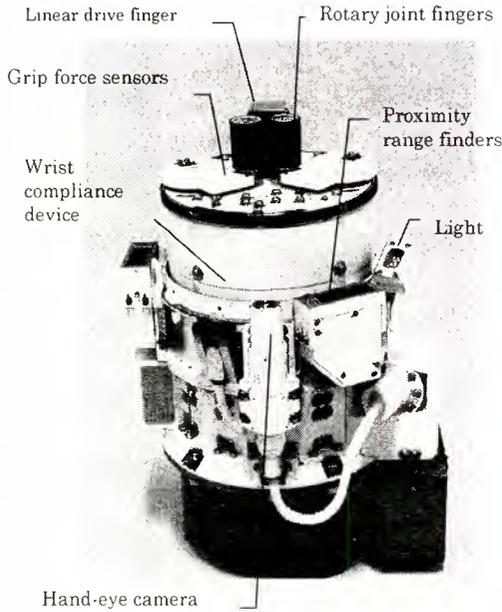


Figure 4: Hand configuration

performance of the system. This panel has a connector with an LED and its switch to check if the connector insertion is completed properly. It also has a bolt, a floating object, a solar cell sheet, a thermal blanket, and a wire to be handled with the hand.

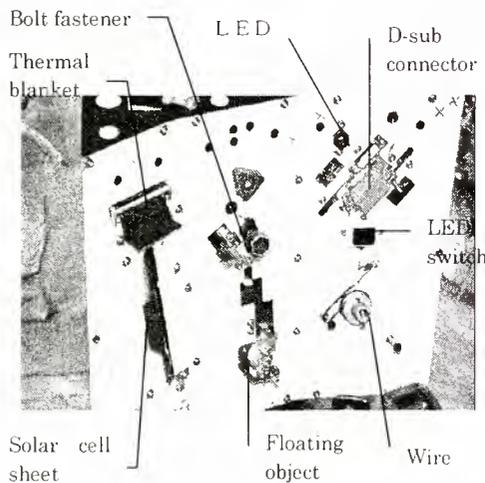


Figure 5: Task Board Configuration

2.4. Control Computer and Software

The control computer consists of 80386+80387 as the MPU with memory of 128kB and 256kB for ROM and RAM respectively, and a DSP for controlling the mini arm. The software⁽⁵⁾ of each of the experiments such as a bolt or connector mate/demate experiment is to be installed separately on the multi-task operating system using application program interface or API, where these software on the RAM is able to be rewritten by the telecommand so that the system could be flexible and be executed with limited hardware resources. This software structure can also make the system serve as a flying test bed for a space robot, where users can perform some experiments using their own logic.

2.5. Task Panel on Target Satellite

The task panel is an experiment panel boarded on the target satellite of ETS-VII, which is used during the long-arm-connected mode. It consists of an electric connector and a bolt, which are handled with the hand connected to the ERA. Figure 6 is the view of the task panel through the CCD camera of the ERA.

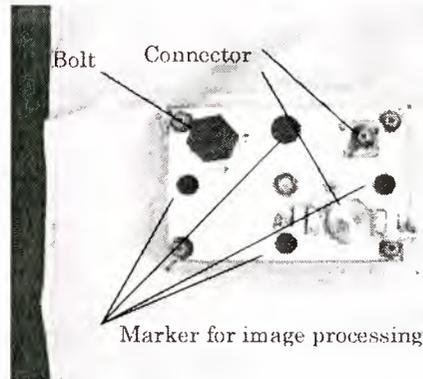


Figure 6: Task Panel (onboard)

2.6. Ground System

As was shown in the figure 2 and 3, the ground system consists of an operation manager, a telemetry and command processor, and a graphical simulator. In order to watch the onboard status, we constructed real-time images of the robot on the display with computer graphics using the telemetry data. The motion prediction images are also visualized on the same display using the telecommand data. Moreover, the onboard sensor status are visualized so that the operator can get almost all the onboard status with just a

glimpse of the computer display. This computer-generated camera, of which we call a "virtual-hyper camera"⁽⁶⁾, provides much more important information than a real TV camera, which also requires much more data to transfer.

Regarding the master device to operate the robot, we chose a mouse-like device as shown in the figure 7, where our former study⁽⁷⁾ with respect to the master device for a space robot concluded that the best one was the mouse type on the grounds that the space robot does not necessarily move in 3 dimension at the same time. Other type of master device has some problem of operability from the point of view of operator's fatigue.

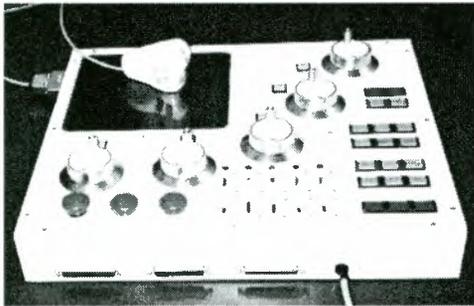


Figure 7: Mouse Master Device

3. Technologies to Evaluate

All the experiments have been carried out in two configurations as is shown in Table 1: an ARH stand-alone configuration and a long-arm-connected configuration as was described in chapter 2. And also there are three operation modes to carry out these experiments: autonomous operation mode, tele-operation mode, and fusion mode.

In this table, the technologies to evaluate are also shown. In order to achieve these experiments, total system performances are required, not only the positioning accuracy of the mini arm or the fingers, but also the image processing performance, resolution and stability of all the sensors including the LASER range finders, force torque sensors, and so on.

3.1. Stand-alone mode

Considering the utilization of a semi-autonomous and tele-operable space robot in the near future, the expertise of handling an electric connector, bolt fastener, or even a floating object has to be obtained through the experiments as described in the introduction. The connector installed on the task board as a handling object is just the same D-sub connector as we use for the space components. It allows only around $\pm 0.5\text{mm}$ as the

Table 1: Space experiment and Key Tech.

Mode & Task	Technologies to Evaluate
Stand-alone mode	
Connector mate/demate (A,T)	Precise positioning, precise part insertion and grip force control
Bolt fastener mate/demate (A,F)	Screw motion control under force monitoring and re-grasping.
Floating object capture (A,T)	3D-position measurement and capturing a floating object in 0g.
Solar cell sheet handling for power generation (A)	Handling technology of flexible sheet
Wire handling (T)	Handling technology of flexible wire and teleoperation by virtual operator method
Working in eclipse (A)	Effectiveness of virtual-hyper camera
Long-arm-connected mode	
Connector mate/demate	Error compensation of long arm
Sample return	Work environment measurement and error compensation of long arm

The uppercase letter A, T, and F following the description of the "mode & task" stands for "autonomous", "tele", and "fusion" operation mode respectively, which means the operation mode to be applied.

positioning error to mate. So sub-millimeter level control is requisite to perform this task. Without any aid, it would be very difficult to achieve this experiment considering the positioning accuracy of the space robot. So the precise positioning performance with the aid of the image processing technology should be evaluated. Also the grip force control technology should be evaluated in order to mate the electric connector. With respect to the bolt fastener mate/demate experiment, it would be very difficult to detect whether the fastener is in mesh. So the screw motion control and re-grasping technology including wrist and grip force control should be verified. Regarding the capturing of a small floating object under the zero gravity environments, the 3D-position measurement and capturing technology using the LASER range finder should be verified. Furthermore, in order to handle a flexible object like a wire, which is difficult to be handled autonomously because of its every-changing shape, the effective teleoperation with the aid of the motion prediction technologies is required.

In the experiment called working in eclipse, the feasibility of carrying out the tasks

under the eclipse with the aid of an LED illuminator installed on the side of the hand should be verified. This is on the grounds that the workable time of the space robot would be seriously limited if it were not able to work in eclipse as it is now.

3.2. Long-arm-connected mode

By handling the connector and bolt installed on the task panel (see section 2.5) using the hand connected to the ERA, the feasibility of attaching the highly precise hand to the rather not so precise space robot should be verified. Rather larger-scale space robot would not have enough positioning precision to handle an electric connector or a bolt.

4. Results

Several experiments were carried out over five times of experiment windows, which has been planned to accomplish over one and a half years. Each window consists of around three-five consecutive days.

4.1. Reconfiguration on orbit

First critical event, which was carried out in March 1998, was the reconfiguration of the robot system on orbit including launch lock relief and hand attaching sequence because the mini arm and the hand were locked separately by mechanical cramps when they were launched (see figure 8).

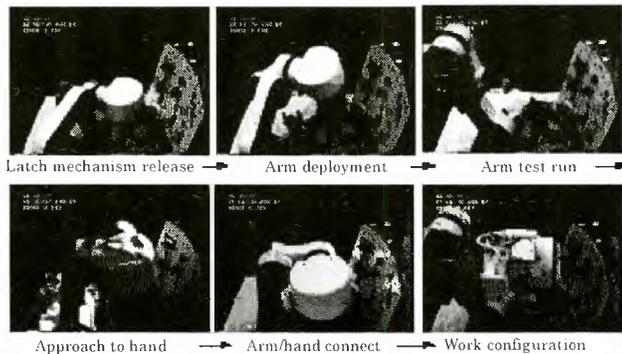


Figure 8: Reconfiguration

With respect to the launch lock relief sequence, we feared the cold welding as well as the hot welding, which might be caused by the evacuated thermal cyclic environment and mechanical vibration respectively. Metallic contact surface between these cramp mechanisms and both of the mini arm and the hand were processed by solid state lubricant for the measures. The telemetry data of the wrist force and torque sensor for the experiments showed rather lower level of maximum 18N and

0.9Nm compared to the ground test result. This means there was no welding on orbit.

With respect to the subsequent hand attach test, the major concern was the position error of the hand attaching location and orientation because the exact position in the zero gravity field was not able to be measured on the ground. By adjusting these errors using force feedback control technology, this task was carried out with no trouble. Figure 9 shows the wrist sensor and the arm position. The maximum load to the arm was around 15N. The axial force F_z reached 17N when the interface of the hand contacted that of the arm. These results were found to be lower level compared to the ground test. Initial checkouts for all mechanisms including all the sensors were carried out after the hand was attached.

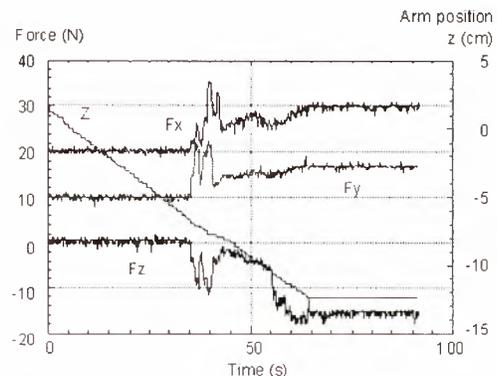


Figure 9: Wrist Force in Hand Connection

Through these system checkouts, we could confirm that the mini arm and the hand system could work well in space. Also we could find that the noise level of all of the sensors were low compared to the ground test.

4.2. Electric Connector Mate/Demate

Electric connector mate/demate experiment was carried out as the first major task for this system in July 1998 on the second window. This experiment was carried out in both modes of autonomous operation and teleoperation.

First, under the autonomous operation mode, using the LASER range finder measuring three points on the task board, the attitude of the hand was set vertical to the task board. Then the local coordinates of the task board were created using the hand eye CCD camera measuring the several circular markers on the task board. This approach, we think, will be very efficient under the condition that the absolute position and orientation of the parts like connector are unknown, where the only thing we should know is the local position and

orientation. The positioning accuracy we could obtain by using this technology was around $\pm 1\text{mm}$. This accuracy seems enough for the other handling objects, but the electric connector is an exception, which requires at least 0.5mm accuracy.

As the next step, we took an image of the receptacle of the connector itself using the hand eye camera (see figure 10) and processed the image of the pinholes of the connector (see figure 11) in order to determine the precise location and orientation of the connector.

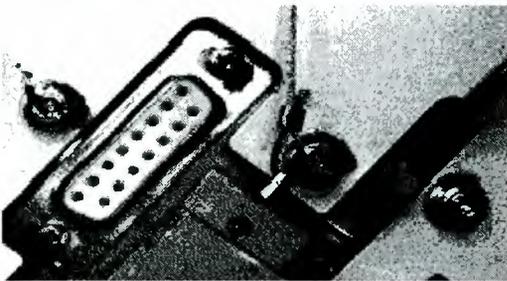


Figure 10: Connector image picture

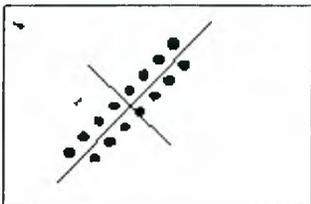


Figure 11: Processed image

By using this technology we could obtain the positioning accuracy of around $\pm 0.25\text{mm}$. After measuring the accurate position of the receptacle, the connector was inserted to it using the passively controlled compliance mechanism, which would compensate the residual positioning error. Figure 12 shows the wrist force sensor when the connector was inserted.

Through this experiment, we could confirm the electrical connector could be handled with the multi-sensory space robot autonomously. And we could confirm the effectiveness of this image processing technology, although we also found this needs more considerations. We had some trouble to adjust the shutter speed of the camera to the every-changing lighting condition for obtaining the best image. This might be caused by the lack of the pixel numbers of the pinholes of the connector or by the effect that the sunlight incidents directly into the pinholes, which hinder sampling the pinholes.

The experiment under the tele-operation mode was carried out on the following day. The simulator, displaying the attitude and

position of the mini arm as well as the fingers based on the command of 4Hz and the telemetry of 8Hz , was found to make the tele-operation easier, although the communication time lag of around 5-6 seconds was expected to hinder the timely operation. We could demate the electric connector successfully, but the connector mate experiment by tele-operation was unsuccessful.

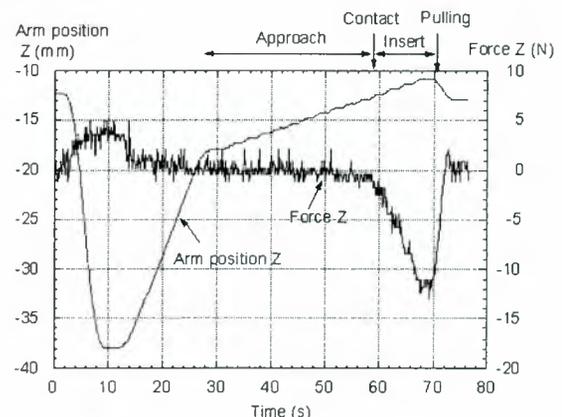


Figure 12: Connector Insertion

By analyzing the experiment results, we found that the real position of the connector was off to the side of around 4mm compared to the position that the graphical simulator indicated. We estimate that this may be caused by the several following reasons. One is that we omitted such models as the passively controlled wrist compliance mechanism from the simulator. Second is that the CPU load of the graphical simulator on which the operator could only depend became high during the operation, and the commanding interval kept changing, which made the operability worse. Third is that the force torque sensor output with some bias noise, which operator could not detect the position where the insertion was completed. We think this might be a lesson to learn about how to carry out a simulator-based tele-operation for a space robot.

4.3. Bolt Fastener Mate/Demate

Bolt fastener mate/demate experiment was carried out on the third window in October 1998. In order to mate the bolt, the axial center of the rotational axis of the mini arm and that of the bolt should be coincident. Also the status whether the bolt is in mesh properly should be detected. This experiment was carried out in fusion operation mode, where either the autonomous mode or the tele-operation mode was applied according to the circumstances.

First, the local coordinates were created as the same procedure as described in

section 4.2 to know the location and orientation of the bolt, which was installed into the lower receptacle. Then the bolt head was grasped. The relative location of the bolt against the hand was detected by knowing the position of each finger. Using these data, the axial center of the mini arm was adjusted to the bolt autonomously. Then the bolt was screwed to the loosening direction to the movable limit of the arm of 180degree. After that the bolt was pulled to the axial direction in order to confirm the bolt was demated. By iterating this procedure, the bolt was demated properly.

After inserting the tip of the bolt to the another upper receptacle, the bolt was screwed to the tightening direction. Then the bolt was pulled to the axial direction as the same way in demating in order to check if the bolt was in mesh. By that time, the experiment was carried out in autonomous operation mode. After that time, we switched the operation mode to fusion mode in order to confirm the effectiveness of it, which compensates the technologies that are insufficient in the tele-operation mode. In this operation mode, the local autonomy of the onboard system, for example the grip force control, was used.



Figure 13: Bolt handling experiment

(1)Image through hand eye camera. (2)Image through camera on ERA. (3)Graphical simulator. (4)Operation scene.

The bolt was pulled to the axial direction with the force of around 7N. Then the bolt was screwed to the loosening direction monitoring the wrist force in order to detect the meshed angle, which was found to be 27 degree. Released and inserted to the lower receptacle again, the bolt was screwed to the fastening direction. And the meshed angle was checked, which was found to be more than 30 degree. Then the bolt was fastened properly. Figure 13 shows a scene of this experiment.

On the next 4th experiment window, the bolt handling experiment was carried out using the advanced function⁽⁹⁾ of the graphical simulator, which is expected to improve the

tele-operability. In this experiment, the simulated 3-D computer graphic was overlaid on the real video image of the hand eye camera (see figure 14). And the action for the onboard system to take was indicated with the mouse of the computer using a drag&drop-like operation style, where we call it a tele-programming operation. This is just the same way as the mechanical engineers design the hardware using the CAD system on their personal computer.

Through this experiment, we could confirm that the bolt fastener could be handled with our technology in both ways of autonomous operation and tele-operation. Also we could confirm the effectiveness of the advanced function of the graphical simulator.

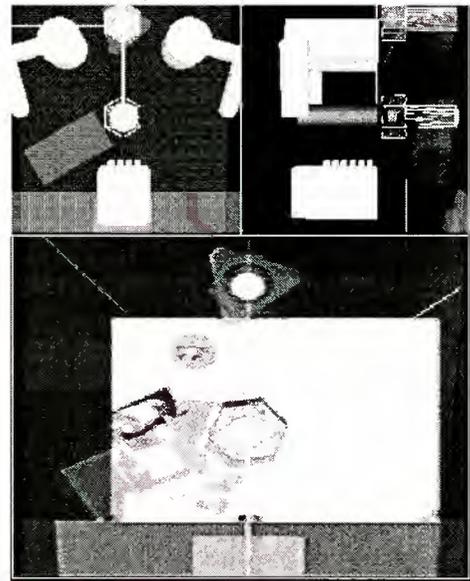


Figure 14: Simulated image overlaid on the real video image

4.4. Floating Object Capture

The floating object is an aluminum sphere of around 30mm installed onto the task board as shown in figure 5. This experiment was carried out as the tele-programming experiment.

After released, the floating object was supposed to be searched with the LASER range finder scanning the appropriate region where the floating object was expected to be. After detecting the approximate position of the floating object, which could be calculated from the distance measured with the LASER range finder and the attitude of the arm, the floating object was supposed to be captured. Actually the experiment was aborted and postponed because we have had a trouble after grasping it. The grip force sensors indicated unexpected value when it was grasped. We have been

analyzing the experiment data and we suppose that the floating object was adhered to the launch-lock mechanism. However, through this teleoperation experiment, we think we could confirm the effectiveness of the teleprogramming operation.

4.5. Working in Eclipse

Under the eclipse environment, the only thing we could depend is the virtual-hyper camera or the graphical simulator because all the video monitors would not work for lack of the sunlight. Turning on the LED installed on the side of the hand, we carried out the same image processing procedure of the connector experiment as described in section 4.1. As the result, the image obtained through the hand eye camera was dark as we expected, but was enough to process by CPU (see figure 15). Every step of the procedure was carried out with no trouble and we could confirm that the LED of around 1W is enough to get the images for the image processing. Also the effectiveness of teleoperation using the virtual-hyper camera was confirmed. Furthermore, we could find that the lightning condition was rather better than that in the sun side of the Earth because there is no more every changing situation in eclipse.

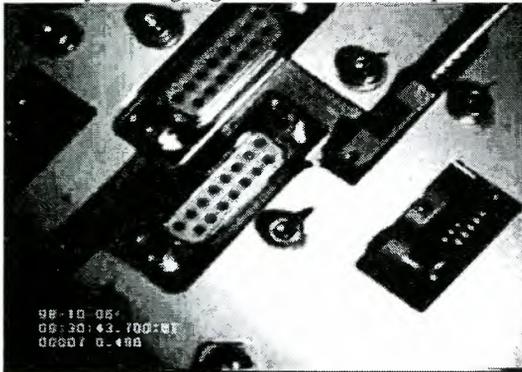


Figure 15: Image in eclipse

4.6. Summary of Other Experiments

The Solar cell sheet handling experiment was carried out by the multisensory autonomous control, and the LED illumination powered by the solar cell was confirmed through the hand-eye camera. The wire handling experiment was achieved by teleoperation from the ground. The details are presented in the accompanied paper⁽¹⁰⁾. In the long-arm connected mode, the connection of the hand to the long arm ERA has been successfully carried out in May 1999, and precise in-orbit servicing to the task panel on the target satellite has been performed.

5. Conclusion

We have carried out several in-orbit servicing experiments in order to evaluate the capability of the semidexterous robot hand for executing precise and delicate tasks as well as to validate the related telerobotic technologies implemented in the system.

Through these experiments, we could confirm that the three-finger multisensory hand of this robot system is valid enough to carry out the precise tasks autonomously in orbit.

We could also confirm that the teleoperability of this system is efficient enough to carry out in-orbit precise works with the aid of the local autonomy of the onboard system as well as the computer-graphics-based telerobotic function of the ground system in spite of the insufficient communication capacity, the unavoidable communication time lag, and unsatisfactory information of the video images.

More detailed information about the ARH including the results of the space experiments is on the Internet at <http://www.etl.go.jp/~5822/ARH/ARHEng>.

References

- 1)K.Machida, et.al. "Development of Advanced Robotic Hand System for Space Application," I-SAIRAS'94, 1994
- 2)G.Hirzinger, B.Brunner, J.Dietrich, and J.Heindl, "Sensor-Based Space Robotics-ROTEX and Its Telerobotic Features," IEEE Trans. Robotics and Automation, Vol.9 No5, 1997
- 3)M.Oda, "Engineering Test Satellite VII: Rendezvous, Docking and Space Robot Satellite," I-SAIRAS'97, 1997
- 4)K.Machida, et.al. "Precise Space Telerobotic System Using Three-finger Multisensory Hand," IEEE Int. Conf. On Robotics and Automation, 1995
- 5)M.Moriya, et.al. "Flight Software of the Advanced Robotic Hand system," 48th International Astronautical Congress, 1997
- 6)K.Machida, et.al. "Precise Telerobotics for Long Distance Space Mission," International Conference on Mobile Planetary Robots, 1997
- 7)N.Tsuda, et.al. "Development of Master Manipulator for Space Robot," Proc. of The 13th Annual Conf. of Robotics Society of Japan p401-p402, 1995.
- 8)K.Machida, H.Nishida and K.Akita, "Precise Telerobotic System for Space Experiment on ETS-VII," IAF-98-U.5.05, 49th Int. Astronautical Congress, pp.1-10, 1998
- 9)A.Noda, et.al. "Ground Control System for Precise Telerobotic Operation of the Advanced Robotic Hand." The 9th International Conference on Advanced Robotics, 1999. (to be issued)
- 10)N.Matsuhira, et.al. "A wire Handling Experiment Using a Teleoperated Advanced Robotic Hand on ETS-VII," I-SAIRAS'99, 1999

Rover Control (1)
[Control of Special Tasks]

Autonomous Rock Tracking and Acquisition from a Mars Rover

Mark W. Maimone, Issa A. Nesnas, Hari Das
 Jet Propulsion Laboratory
 Pasadena, CA 91109

Email: mark.maimone@jpl.nasa.gov

Phone: +1 (818) 354 - 0592 Fax: +1 (818) 393 - 4085

<http://robotics.jpl.nasa.gov/tasks/pdm/papers/isairas99/>

Abstract

Future Mars exploration missions will perform two types of experiments: science instrument placement for close-up measurement, and sample acquisition for return to Earth. In this paper we describe algorithms we developed for these tasks, and demonstrate them in field experiments using a self-contained Mars Rover prototype, the *Rocky 7* rover. Our algorithms perform visual servoing on an elevation map instead of image features, because the latter are subject to abrupt scale changes during the approach. This allows us to compensate for the poor odometry that results from motion on loose terrain.

We demonstrate the successful grasp of a 5 cm long rock over 1m away using 103-degree field-of-view stereo cameras, and placement of a flexible mast on a rock outcropping over 5m away using 43 degree FOV stereo cameras.

1 Introduction

NASA is engaged in a series of missions designed to study the planet Mars. The current schedule calls for 5 pairs of orbiter/lander probes to be launched approximately every two years, starting with the Mars Pathfinder mission of 1997. The 2003 and 2005 missions, in particular, call for a rover with the ability to traverse more than 1 kilometer away from its landing site, acquiring samples along the way.

Autonomous robotic operations can greatly increase the science return of such planetary missions. As these operations become more adaptive, the burden of planning a sequence of motions is moved from the human operator to the onboard control system, allowing a greater number of targeted experiments to be achieved. In this paper we describe algorithms that allow a rover to autonomously approach and collect (or analyze) a sample at a human-specified target

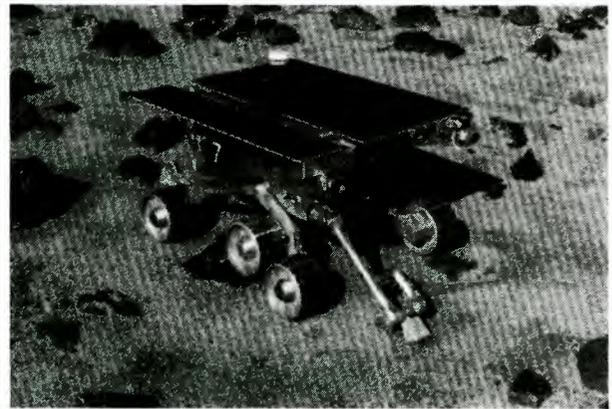


Figure 1: The *Rocky 7* rover

location.

Our approach combines vision processing with vehicle and arm control. The target is identified in an image by a human operator, and its 3D location is computed onboard using stereo vision. A curved path toward the target point is planned, and executed in small steps. The shape of the terrain immediately around the target is used to reacquire the target at each step; we servo on the elevation map instead of image features, because the latter are subject to abrupt scale changes during the approach. This allows us to compensate for the poor odometry that results from motion on loose terrain, by visually reacquiring the target at each step. Vehicle motion stops when the target appears within the workspace of the arm that will be used to grasp or study it.

In the sections that follow, we survey related work that uses visual servoing to guide end-effector motion, describe the general algorithm, and detail the experimental results from field tests performed on the *Rocky 7* Mars Rover prototype (see Figure 1).

- | |
|---|
| <ol style="list-style-type: none"> 1. Acquire stereo image pair with body navigation cameras 2. Send the left image over wireless network to host 3. Scientist/Operator selects target rock on left image 4. Target location and intensity threshold sent to rover <p style="text-align: center;"><i>All subsequent processing occurs onboard</i></p> <ol style="list-style-type: none"> 5. Identify 3-D location of rock based on calibrated camera models and onboard stereo image processing 6. Compute single-arc rover trajectory to target 7. Drive rover toward target 8. Periodically (every 10 cm) poll the target tracking software to update target location using new stereo pair and current odometry 9. Redirect rover toward the new target location using new single-arc trajectory, and repeat until target is within 1 cm of goal position. 10. Deploy sampling arm and pick up rock. |
|---|

Table 1: Algorithm for small-rock acquisition

2 Related Work

First described in [WSN85], visual servoing strategies incorporate vision sensing with the actuation of motors in a robotic system. Often simple image-processing filters are used to locate a target of interest, and knowledge of the camera system geometry and manipulator kinematics are used to control motor current. This technique has been applied successfully to the active placement of a manipulator at high frame rates (e.g., in [HGT95], [PK93], [Nis90], and [THM⁺96]). In this application the distance of the target from the camera system usually remains the same, so the relative size of the object will remain constant throughout the servoing process.

In our case the entire robot, not just a manipulator, is being directed toward a goal point. Visual servoing for vehicle motion should be a useful tool, because the uncertainties introduced by motion over unknown terrain could potentially be eliminated by the visual tracking. However, as the vehicle approaches the target, the target's image size grows dramatically between updates, and a correlation search on the intensity image tends to fail. Therefore approaches such as [WTB97] work well at long distances, but are less reliable at the final approach to the object.

3 Approach

The general problem we attempted to solve is the identification and collection of an interesting rock sample, in a control architecture that meets the constraints of interplanetary operation. This latter requirement is summarized as follows: there will be a high latency in communication between the operator and rover (from 4 to 21 minutes one-way), and the number of messages sent must be minimized. For ex-

ample, during Mars Pathfinder operations in 1997, logistical constraints on the Deep Space Network dictated that only two 5-minute communications windows were available each day.

This general problem can be broken down into a series of steps: Target Selection, Rover Motion toward the Target, Target Visual Reacquisition (these two steps might repeat a number of times), and Target Grasping. The first of these steps, Target Selection, is an extremely difficult task to automate, because it would require the rover to determine which samples are scientifically interesting. We felt this was a task best left to scientists, and therefore designed our system to require a single round-trip transmission to allow a human scientist to perform it. We felt that the remaining steps could be made sufficiently robust to be implemented entirely onboard the rover.

A summary of our algorithm for sample collection can be found in Table 1. The following subsections describe each component of the algorithm in detail, and refer back to the numbered steps in Table 1.

3.1 Target Selection

Target Selection is the first step of our sample acquisition process (steps 1-4 in Table 1). We assume the rover is already deployed in the area of interest, and has taken a stereo pair of images of the terrain in front of it. We transmit the left image from this stereo pair over the wireless network to a human operator who inspects the image, locates an interesting sample (a surface rock small enough to be grasped by the robot arm), selects it with the mouse, and transmits its image location back to the rover. Figure 2 illustrates a sample target selection. This step requires one round-trip communication between the rover and operator.

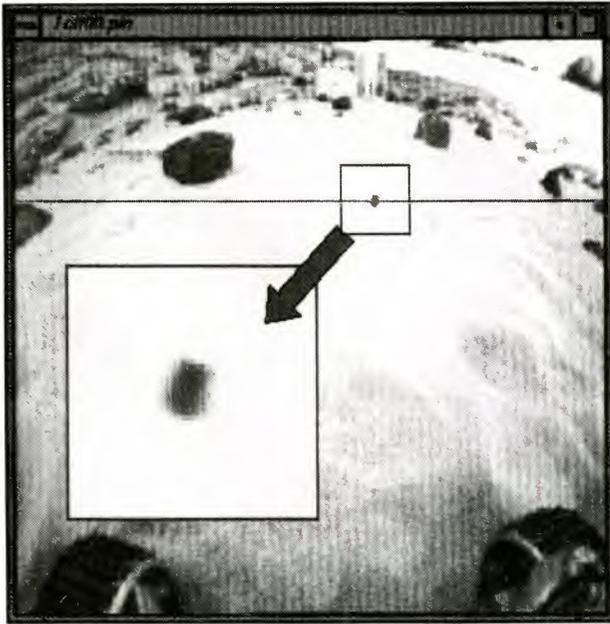


Figure 2: Sample target selection in Java GUI display. The selected target is shown zoomed in.

We found it necessary in later processing to segment out the rock from its background using brightness-based intensity thresholding. So in addition to the image coordinates of the target rock, the operator communicates a brightness threshold and range to the rover (e.g., “pixels with 8bit intensity darker/lighter than 145 should be considered rocks”).

3.2 Rover Motion toward the Target

Next the rover performs computations and moves toward its target (steps 5-7 and 9 in Table 1). Once the rover receives the goal point in image coordinates, it uses stereo image processing and a geometric camera model to compute the (X,Y,Z) location of the target in the rover reference frame. Details of the JPL Stereo Vision algorithm can be found in [XM97]. Note that the goal location is stored in the 3-D rover reference frame, not a 2-D image frame.

Having computed a location in world coordinates, a single arc is computed that should bring the rover close enough to the target that it appears within the workspace of the arm (see Figure 3). Our experimental arm had only 2 degrees of freedom, so it was important that the rover be positioned correctly to within a small tolerance, i.e., about 30% of the size of the 2 DOF gripper.

The rover is then commanded to move a short distance along the arc (10 cm or the remaining distance to goal, whichever is smaller), and its position

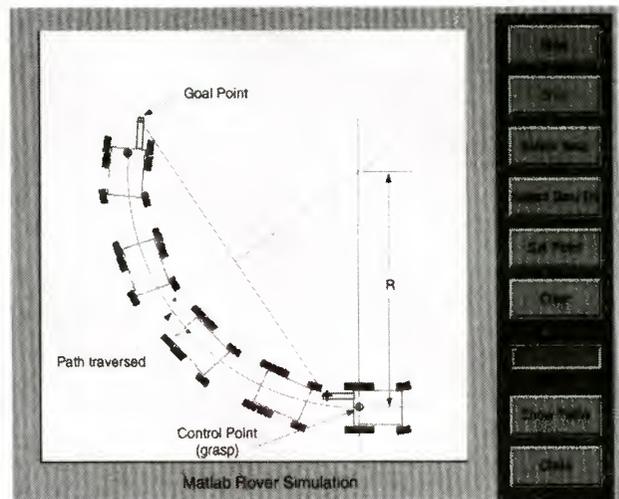


Figure 3: Single arc trajectory generation

is reevaluated in the next step.

3.3 Target Visual Reacquisition

Having made partial progress toward the goal, the rover stops to evaluate its current position (step 8 in Table 1). This update is initialized by subtracting the motion just taken from the target location in the rover frame. The motion just taken is estimated by computing vehicle odometry from wheel rotations. This is a very noisy estimate, because nothing is known about the surface on which the rover is moving; it could consist of pebbles, sand, sticky tar, or solid rock.

A starting point in a fresh stereo image pair is computed from this new estimated location, and a small window around that point is searched in an attempt to locate the target. However, instead of searching the raw intensity image we automatically compute a range image from the stereo image pair, and search the resulting elevation map for the *shape* of the target, rather than its visual appearance. In particular, we assume that any target rock will be resting higher on the ground than its nearby surroundings, and lock in on the local elevation maximum as the new, refined 3D target point. We may not always achieve a completely dense elevation map from the range data, so before searching for the local maximum we linearly interpolate any data missing from the range image. Given this dense, interpolated elevation map, we start at the best estimate of the target location and “climb” to higher elevations until we reach a local maximum.

Unfortunately, early experiments showed that on

a sandy surface, the error in the odometry estimate was sufficient to cause this method to lose the target. That is, the search window was centered too far away from the target rock for a simple gradient-ascent climb to recover it, even after relatively small motions. A general solution to this problem would be to incorporate more effective position and pose sensing and estimation into the rover. We anticipate that the work described in [Bal99] will provide such estimates and will be incorporated onboard the *Rocky 7* rover soon, but it was not available during the timeframe of our project.

Instead, we took advantage of the fact that our targets were visually distinct from the background sand, and used an intensity filter to focus attention in the elevation map. Given the search window centered at the (noisy) estimated target location, pixels in the image window are classified in one pass as either BACKGROUND or ROCK according to the threshold value set by the operator. The ROCK pixel nearest the center of the search window is then treated as part of the target, and the enclosing blob of ROCK pixels are relabeled TARGET pixels. Finally, the centroid of all TARGET pixels is computed, and its range value (perhaps an interpolated value) is used as the starting point for the climb to the local elevation maximum. Using the centroid preserves the scale-invariance of our method. In fact, any pixel classification technique can be used instead of brightness: on a flight mission one might use spectral filters to distinguish rocks from non-rocks, as in [PAW⁺98].

If no range data are available, then no refinement is done, and the vehicle odometry is assumed to be correct.

The new target location is fed back into the Rover Motion toward Target step, and vehicle motion continues until the target is found to be within the workspace of the arm.

3.4 Target Grasping

Finally, having determined that the target lies within the workspace of the arm, the arm is deployed and the target grasp is attempted (step 10 in Table 1). We use the difference between the actual and commanded trajectories from the motor encoders to tell when the arm makes contact with the target or ground, then close the gripper on the target. Instead of lifting off right away, we raise the arm a small amount and continue to close the gripper until it stops, several times more. This redundancy helps ensure that the gripper has a good hold on the target.

4 Experimental Results

As testbed for these algorithms, we used the *Rocky 7* Mars Rover prototype [Vol99] (see Figure 1). *Rocky 7* is a 6-wheeled vehicle with rocker-bogey suspension and one set of steerable wheels. Batteries and solar cells provide about 50 Watts of power. A small 2 DOF arm with 2 DOF gripper mounted on one side of the vehicle is used for digging and grasping rock samples, and an extendible 3 DOF mast provides stereo image views from as high as 1.5 meters above the ground. For terrestrial work, communication is via a 1 Mbit/sec wireless ethernet bridge or a 10 Mbit/sec coax hard line. Onboard processing consists of a 60 Mhz 68060 CPU running the Vx-Works 5.3 operating system in 16 megabytes of RAM. Vision sensors include three pairs of stereo cameras: one body-mounted pair faces the arm, another body-mounted pair is on the other side of the vehicle, and the third pair is mounted near the end-effector on the extendible mast. All cameras are 480x512 CCD board cameras (but currently only half-resolution images are used), and the body-mounted cameras have an effective FOV of 103 degrees, while the mast cameras have an effective FOV of 43 degrees. The body-mounted cameras are approximately 30 cm above the ground, point downward at an angle of approximately 45 degrees, and are used primarily for detection of nearby obstacles. During these experiments the vehicle moved approximately 5 cm/sec and paused briefly during the image acquisition and path generation steps.

We performed several experiments in JPL's Mars Yard¹, and successfully demonstrated the autonomous acquisition of small rocks (3-5 cm) located over 1 meter in front of the rover. Figure 4 shows a sample tracking sequence, with the target indicated in each frame by a dark square. Execution of the entire sequence (Target Selection, 8 - 10 iterations of Target Reacquisition, and successful Target Grasping) typically completed within one minute when the target was just over 1 meter away.

Many experiments were run, and 14 complete image/odometry datasets were collected. When run over these datasets, the visual tracker succeeded in maintaining target lock through 10 complete sequences. Primary failure modes were due to abrupt intensity changes because of indoor lighting or rover shadow. All but one of the failures were corrected by simply re-running the visual tracker with a more appropriate intensity threshold; in the final failed sequence the target was the same color as the back-

¹<http://marscam.jpl.nasa.gov/>

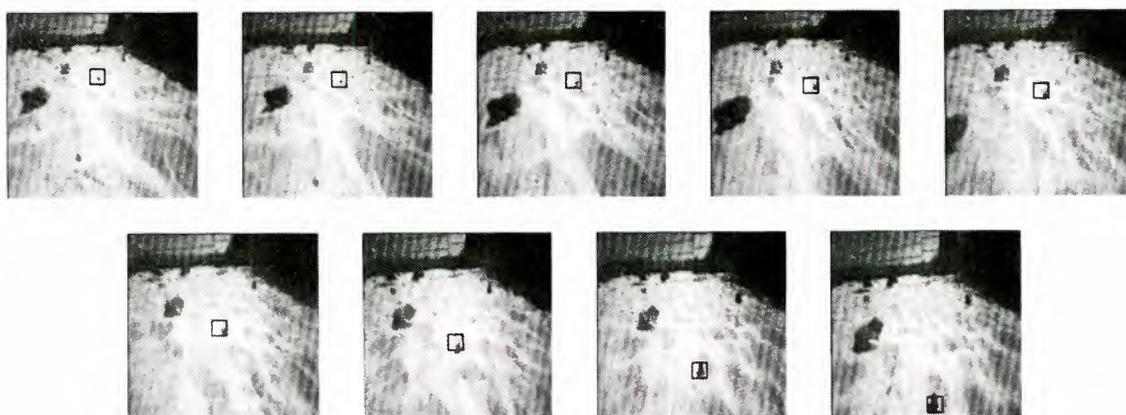


Figure 4: Sample tracking sequence.

ground.

In general, failures can occur when:

- The target leaves the camera FOV, so no range data is available and tracking depends entirely upon noisy odometry.
- The target is visible, but no range data is computed. This can happen if the stereo optics are not properly set for current lighting conditions.
- Multiple targets are visible in the search window and odometry is poor. Additional filtering based on range data could alleviate this, as could matching based on more than a single shape feature (i.e., not just the elevation maximum).
- The target is visible but outside the search window. This happens when the rover climbs over very hilly terrain, if the pose is not measured and used to predict the search window starting point. One could search again using revised motion parameters, or improve the pose sensing.
- Tracking is fine, but the rock is not picked up. This can occur if the rover gets stuck in a servoing loop, attempting to make small changes in position. On sandy soil, such maneuvering introduces much positional uncertainty.
- The target is the same color as the background, so the intensity filter is irrelevant or misleading.

4.1 Mast Placement

This algorithm was also applied successfully to the placement of *Rocky 7's* flexible mast arm on a rock outcropping. The limited degrees of freedom in *Rocky 7's* mast dictate that the vehicle must face the

target point's tangent plane on the surface of a boulder to enable complete coverage by the end-effector. For general targets (anywhere on the surface of a boulder) the surface normal is computed from the range data at closest approach, and a two-arc trajectory generated to ensure that the vehicle approaches the rock normal to the tangent plane of the target. However, since this algorithm servos on the local elevation maximum, only targets on the tops of rocks were able to be specified.

During several trials in the Mars Yard *Rocky 7* successfully tracked targets (the tops of boulders 20–50 cm tall) over 5 meters away using the 43-degree FOV stereo cameras in the mast head and successfully placed the end effector on the target. For this application Target Reacquisition occurred after every 50 cm of motion. Execution of the entire sequence (Target Selection, 8–10 iterations of Target Reacquisition, and successful Mast Placement following the two-arc path generation) typically completed within four minutes when the target was just over 5 meters away.

5 Future Work

In the future we hope to reduce our dependence on the brightness-based filter by matching the entire shape of the terrain around the target (not just its peak) using the technique of [Ols99], and by improving the position and pose estimates using visual feature tracking on the whole scene using a technique from [Mat89]. These improvements should allow tracking of targets anywhere on a rock, enabling a more general mast placement capability, and should also enable tracking of targets that leave the field of view. We would also like to be able to specify multiple targets in a single image, and enable the rover

to keep track of (and acquire) them accurately even if they leave the field of view of the cameras. [PK93]

6 Acknowledgements

We are grateful to Long Range Science Rover (LRSR) task for providing the *Rocky 7* rover and time to use it, and to the other members of the LRSR team for their work in developing and supporting the vehicle, especially Samad Hayati, Rich Volpe, Bob Balaram, Bob Ivlev, Sharon Laubach, Alex Martin-Alvarez, Larry Matthies, Clark Olson, Rich Petras, Rob Steele, and Yalin Xiong. The work described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract to the National Aeronautics and Space Administration.

References

- [Bal99] J. Balaram. Kinematic state estimation for a mars rover. *To appear in special issue of Robotica on Intelligent Autonomous Vehicles*, 1999.
- [HGT95] Gregory D. Hager, Gerhard Grunwald, and Kentaro Toyama. *Intelligent Robotic Systems*, chapter Feature-Based Visual Servoing and its Application to Telerobotics. Elsevier, Amsterdam, 1995. V. Graefe, editor.
- [Mat89] Larry Matthies. *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University Computer Science Department, October 1989. CMU-CS-89-195.
- [Nis90] H. K. Nishihara. Real-time implementation of a sign-correlation algorithm for image-matching. Technical report, Teleos Research, February 1990.
- [Ols99] Clark F. Olson. Subpixel localization and uncertainty estimation using occupancy grids. In *International Conference on Robotics and Automation*, pages 1987–1992, 1999. <http://robotics.jpl.nasa.gov/people/olson/abstracts/icra99.html>.
- [PAW⁺98] L. Pedersen, D. Apostolopoulos, W. Whittaker, T. Roush, and G. Benedix. Sensing and data classification for robotic meteorite search. In *SPIE Photonics East*, Boston, 1998.
- [THM⁺96] D.A. Theobald, W.J. Hong, A. Madhani, B. Hoffman, G. Niemeyer, L. Cadapan, J.J.-E. Slotine, and J.K. Salisbury. Autonomous rock acquisition. In *AIAA Forum on Advanced Development in Space Robotics*, Madison, WI, August 1996.
- [Vol99] Richard Volpe. Navigation results from desert field tests of the Rocky 7 mars rover prototype. *International Journal of Robotics Research*, Accepted for Publication, Special Issue on Field and Service Robots 1999. <http://robotics.jpl.nasa.gov/people/volpe/papers/JnavMay.pdf>.
- [WSN85] L. E. Weiss, A. C. Sanderson, and C. P. Neuman. Dynamic visual servo control of robots: an adaptive image-based approach. In *International Conference on Robotics and Automation*, pages 662–668, March 1985.
- [WTB97] David Wettergreen, Hans Thomas, and Maria Bualat. Initial results from vision-based control of the Ames Marsokhod rover. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1377–1382, Grenoble, France, September 1997. <http://img.arc.nasa.gov/papers/iros97.pdf>.
- [XM97] Yalin Xiong and Larry Matthies. Error analysis of a real-time stereo system. In *Computer Vision and Pattern Recognition*, pages 1087–1093, 1997. <http://www.cs.cmu.edu/~yx/papers/StereoError97.pdf>.

Autonomous Sample Acquisition for Planetary and Small Body Explorations

Ali R. Ghavimi, Frederick Serricchio, Ben Dolgin, Fred Y. Hadaegh

Jet Propulsion Laboratory

Pasadena, CA, USA

Email: ali.r.ghavimi@jpl.nasa.gov

Tel: (818) 354-0470 / Fax: (818) 393-4440

Abstract. Robotic drilling and autonomous sample acquisition are considered as the key technology requirements in future planetary or small body exploration missions. Core sampling or subsurface drilling operation is envisioned to be off rovers or landers. These supporting platforms are inherently flexible, light, and can withstand only limited amount of reaction forces and torques. This, together with unknown properties of sampled materials, makes the sampling operation a tedious task and quite challenging. This paper highlights the recent advancements in the sample acquisition control system design and development for the in situ scientific exploration of planetary and small interplanetary missions.

1. Introduction

Sample Acquisition systems are envisioned to be an essential part of future NASA plans for both the sample return missions and in situ scientific exploration of planetary and small body objects. Near-term NASA missions requiring autonomous sample acquisition capabilities include Mars Sample Return ('03, '05, '07, '09 missions) and Space Technology 4/Champollion launching in 2003. The sample acquisition operation for these missions is required to be off rover or lander systems. These platforms are expected to be light, to be flexible with low-frequency structural modes, and to have little resistance to sampling induced reaction forces and torques. Furthermore, the environments possess uncertain surface substrate with unknown mechanical

properties. The stability of the sampling mechanism and the supporting base platform require that forces and torques must be limited and controlled to certain prescribed threshold, which in turn, impose additional complexity. In light of all these constraints, the sample acquisition operation can be quite challenging and key control system technologies are necessary to ensure stability and performance of the integrated sampling system.

The Jet Propulsion Laboratory (JPL) has developed a rock coring system and a subsurface drilling mechanism that are capable of acquiring samples from flexible and light supporting platforms. A smart sampling control system has also been devised for autonomous and robust operations of these mechanisms. The following highlights a summary of the proposed sampling control system design. Actual test results of the integrated coring and drilling control system design are available demonstrating the autonomous control of the smart sampling control system.

2. Problem statement and Requirements

The minimal requirements for small body exploration are taken from those of the Space Technology 4/Champollion flight project. The basic approach for sample acquisition on small bodies is expected to be lander-mounted drilling platforms. Because of many uncertain characteristics of small bodies, aside from the challenging task of

successful landing, landers must be anchored to the surface to avoid detachment from the surface. In this case, anchors can only provide nominal retention force and torque for lander-based drilling. To ascertain the dynamic stability of the lander-drill system, the drill mechanism must be actively controlled to retain reaction forces and torques generated during sampling.

The primary requirements for planetary sample acquisition are derived from those of the Mars Sample Return missions. The general strategy for planetary core sampling is envisioned by means of rover-mounted coring platforms. By design, rovers are expected to have lightweight and flexible modes with little damping. Further, rover system can have excessive suspension deadband and backlash. Moreover, the expected axial drilling force is comparable with the effective weight of the rover. Therefore, the coring system must be actively controlled to limit the effect of reaction forces and torques imposed by uncertain interactions between the coring bit and the sampling surface.

To satisfy performance requirements for in situ scientific studies, applied drill forces must track commanded force profiles to within certain specified accuracy. The telemetry data such as penetration and rotation rates can be used to derive information about material properties of the sampled rock or substrate drilled. This together with other uncertainties and structural dynamics impose challenging constraints on the end-to-end operation of the coring or drilling control system.

3. Hardware Development

JPL has developed prototypes of two generations of a drill and a coring system. The ESB drill was developed under the Exploration of Small Body Task and is consisted of three modes of operations: 1.

Drill Axis for penetration, 2. Rotation Axis for drilling, and 3. Arm Axis for indexing. The ESB drill has been integrated and tested on a rigid platform, a one-dimensional landing leg, a three-legged lander, and a super light rover with variable mass and flexible modes.

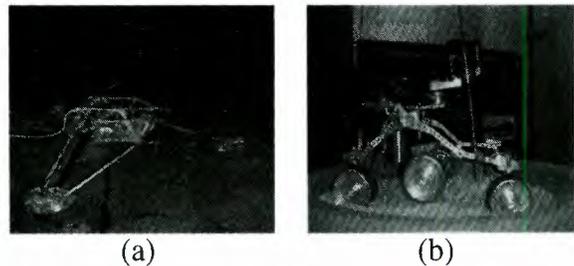


Figure 1. (a) ESB drill on a Lander system.
(b) ESB drill on a flexible rover.

The RDC Mini-Corer was developed under the Robotic Drilling and Containerization Task and is consisted of four axes of operation: 1. Drill Axis for penetration, 2. Rotation Axis for drilling, 3. Break-Off Axis for core breaking, and 4. Push-Rod Axis for core ejection. The RDC Mini-Corer has been integrated and tested on rigid platforms, on a mock rover with variable mass and stiffness, and on the FIDO rover system developed by the Exploration Technology Task. The mounting interface to the platforms is accomplished by a pitch and yaw mechanism that allows motions in two directions.

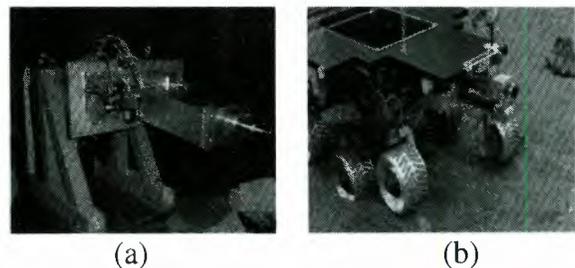


Figure 2. (a) RDC Mini-Corer on a mock rover.
(b) RDC Mini-Corer on FIDO Rover.

The selection of hardware components, e.g., motors, gears, encoders, drivers, electronics, sensors, and other mechanical properties, was accomplished by extensive trade studies between the mechanical design specification options and the control system objectives. The hardware setup in each case played an important role in the development of the key control system technologies.

4. Control System Architecture

The main objectives of the sampling control system are to ensure stability, performance, and efficiency throughout the sampling operation. The following highlights important constraints that must be accounted for by the sampling control system. Rovers and landers can withstand limited reaction forces and torques. Sampling operations can excite structural modes of the associated rover or lander platform. Sampling objects have uncertain mechanical properties, e.g., hardness cohesiveness, porosity, or texture, and can induce fast transient responses. Live rocks and surfaces are inherently unstable and impose additional uncertainty in the sampling process. Thus, a viable drilling initiation strategy, together with a stable and robust drilling control algorithms, becomes a crucial part of the sampling control system.

The control system architecture is composed of three levels of decentralized controller design, where an executive controller commands each individual local servo and accordingly an associated low-level servo in each axis of operation. The executive controller is event-based driven and ensures dynamic stability during the sampling operation and provides fault detection and hazard avoidance. The local servo in each axis is unique and operates independently of other axes. The local servos accept commands from the executive controller and provide closed-loop control of position, rate, force, or torque. The operating plant in each local servo is an independent low-level

motor/encoder servo loop. The controller in each local servo utilizes commanded data, position, rate, force, or torque, to zero out the tracking error between the commanded data and the actual motion profile. The controller associated with each low-level servo loop is designed so that the servo is stable, meets tracking performance, and has a proper and fast reaction response. The overall control system design is fully autonomous and incorporates a hybrid of appropriate controller design methodologies. The low-level and local servos perform continuous control of the sampling dynamics and employ highly optimized loop shaping methods. The executive controller responds to event-based scenarios, fault protection, and fault recovery.

5. Servo Loop Shaping Design

The stringent stability and performance requirements in each axis of the sampling operation demand a highly optimized loop-shape design for each servo subsystem. In general, the control system loop design is function of the sampling frequency, the structural resonance of the system, and sensor noise. The expected sampling frequency provided by the real-time operating system is expected to be between 50 to 100 Hz. Therefore, the control bandwidth is limited to be less than 5 to 10 Hz, respectively. Moreover, the control loop gain must be gain stabilized as the Nyquist frequency is of the same order of the structural modes or lower. The strategy in this case is to design the control loop bandwidth as wide as possible for maximum performance and disturbance rejection. Because of the presence of low-frequency structural modes, the high-frequency portion of the loop shape must have a fast roll-off to ascertain sufficient gain margin at the structural resonance frequency. This however reduces phase margin at the crossover frequency violating stability requirements. This limits the loop bandwidth

further as sufficient gain margins and phase margins are required in the vicinity of the crossover frequency. Under the above constraints, the application of Bode-Step loop-shaping method [1] is quite applicable for the optimal loop shaping of the sampling control system. Figure 3 illustrates the proposed loop design, where f_b denotes the feedback bandwidth frequency and f_{st} is the lowest structural mode frequency. The slope of the low-frequency asymptote is about 10 dB per octave and the that of the high-frequency asymptote is about 18 dB per octave. The Bode step gain extends over one octave providing sufficient separation between the low-frequency and high-frequency rolloffs. This stability margin specifications are about 10 dB gain margin and 30 degrees phase margin. In this respect, once the frequency of the structural resonance is determined for each axis of the operation, the associated loop bandwidth and consequently, a rational transfer function realization can easily be computed in each case.

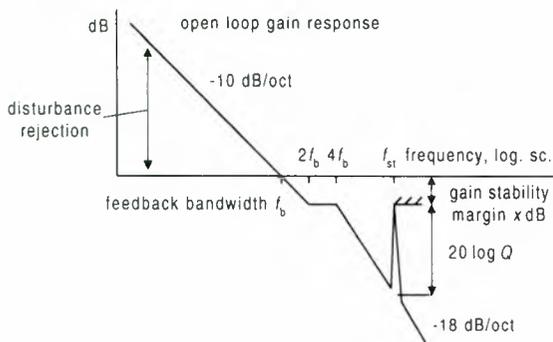


Figure 3. Bode-Step method for servo loop design

Once specific loop shape is designed for each servo subsystem, an appropriate controller can be designed by compensating for the effect of each plant transfer function. The net result is a highly optimal linear controller for each axis of the sampling operation. However, the actual system is

nonlinear and is subject to great deal of uncertainties. Best results can be achieved when a nonlinear dynamic compensation is used together with the resulting linear controller in each case. The actual details of the nonlinear controller design are beyond the scope of this paper and will not be presented any further.

6. Control Software Implementation

The sampling control system software is integrated in the VxWorks real-time operating system running on a dedicated processor. A graphical user interface is designed on an NT or Unix host for command processing, development, data archiving, and post-data analysis. The interface between the target processor and the host is established via an Ethernet or a wireless Ethernet connection. All control algorithms, codes, and device drivers are implemented in C/C++ running on the target processor. Figure 4 shows the high-level schematic of the sampling control system for the ESB drill system. Figure 5 illustrates a block diagram representation of the RDC Mini-Corer control system.

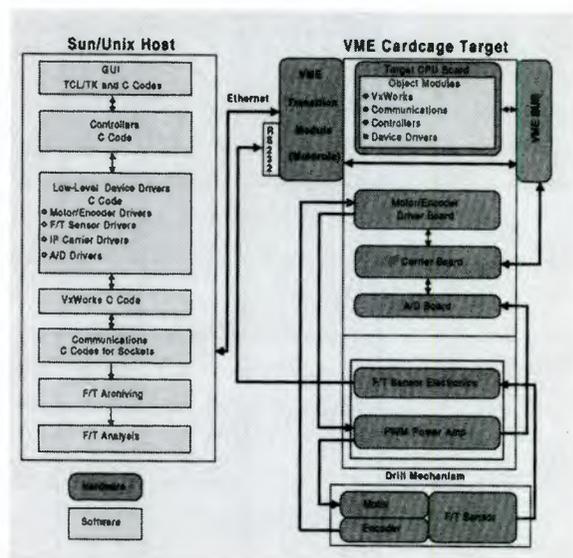


Figure 4. Schematic representation of the real-time sampling control system for the ESB drill.

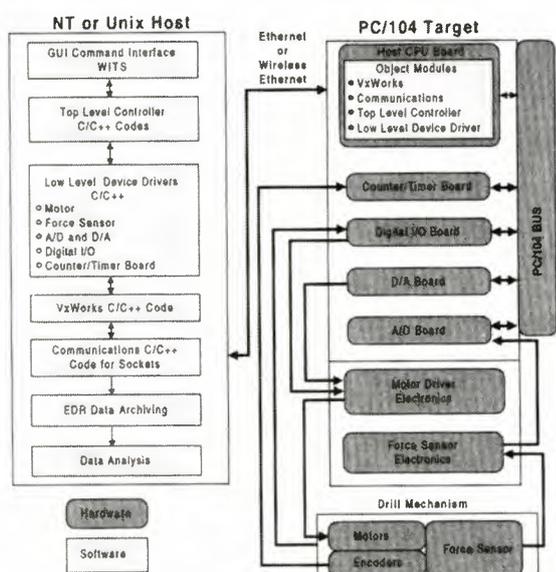


Figure 5. Schematic representation of the real-time sampling control system for the RDC Mini-Corer

7. Test Results

The ESB drill and the RDC Mini-Corer have been tested extensively on various platforms. Numerous test materials have been considered under various operating conditions. The sampling control system strategy proves quite efficient and satisfies system stability specifications, as well as, science performance requirements.

The ESB drill system was used to drill a set of cometary test samples. Real data was archived in each case and was used to generate a material characterization baseline for these test samples. The analyses of the data revealed a great deal of information on the drilling performance, as well as, the sampling material. The results can be classified into an archive to deduce optimal drilling parameters, e.g., thrust force or rotation rate, when drilling unknown materials. Figure 6 demonstrates the results.

The complete analysis of the results is beyond the scope of this paper and will not be presented. The end-to-end sampling operations of these two coring and drilling systems and will be presented in videotapes.

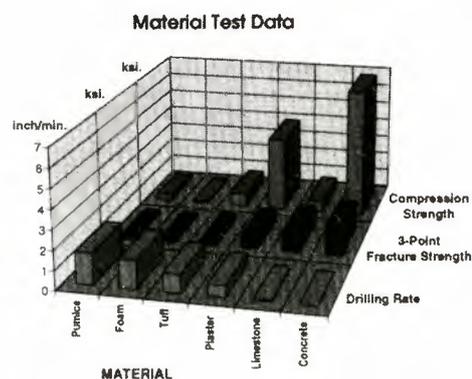


Figure 6. Material characterization for comet simulant materials.

8. Conclusions

The sampling control system design provides the enabling technologies necessary to accomplish autonomous in-situ scientific exploration of planetary and small interplanetary objects. Two generations of sampling mechanisms are developed to perform rock coring and subsurface drilling. Future design efforts for a next generation drill mechanism involve automating a process to add segments to the top of a drill stem for an automated deep drilling.

9. Acknowledgements

This work was performed at Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

10. References

- [1] B. Lurie, A. Ghavimi, F. Hadaegh, *System Trades using Bode-Step Methods*, AIAA, Boston 1997.
- [2] D. Sevilla, A. Ghavimi, F. Serricchio, *Drilling and Sampling Technology for the Exploration of Small Bodies*. International Symposium on Artificial Intelligence, Robotics and Automation in Space, Japan, 1997.

EVOLUTION OF AUTONOMOUS SELF-RIGHTING BEHAVIORS FOR ARTICULATED NANOVERS

Edward Tunstel

Jet Propulsion Laboratory, California Institute of Technology,
4800 Oak Grove Dr., MS 107-102, Pasadena, CA 91109 USA
phone: (818) 393-2666, fax: (818) 354-8172, e-mail: tunstel@robotics.jpl.nasa.gov

ABSTRACT

Miniature rovers with articulated mobility mechanisms are being developed for planetary surface exploration on Mars and small solar system bodies. These vehicles are designed to be capable of autonomous recovery from overturning during surface operations. This paper describes a proposed computational means of developing motion behaviors that achieve the autonomous recovery function. Its aim is to reduce the effort involved in developing self-righting control behaviors. The approach is based on the integration of evolutionary computing with a dynamics simulation environment for evolving and evaluating motion behaviors. The automated behavior design approach is outlined and its underlying genetic programming infrastructure is described.

1 INTRODUCTION

Recent advances in micro-technology and mobile robotics have enabled the development of scientifically capable rovers of mass on the order of tens or hundreds of grams. Development of such *nanrovers* will permit mobility-based science surveys on planetary surfaces with a small fraction of the science payload expected for currently planned, and future, rover missions. Nanrovers have been proposed as possible payloads on landers used for missions to Mars, small bodies, or the moons of gas giant planets [1]. They could be used as individual units or cooperative teams to survey areas around a lander, or even to conduct long-range exploration involving measurement of surface mineralogic and morphologic properties. Research efforts are underway to develop nanrovers that include mobility, computation, power, and communications in a package of several hundred grams in mass [1]. Thus far, a functional nanover prototype has been developed that is capable of autonomous mobility, science data gathering, and transmission of telemetry to an operator control station [2]. A flight version of the rover is currently under develop-

ment as a technology experiment on an asteroid sample return mission called MUSES-C. The MUSES-C flight mission is being implemented by Japan's Institute of Space and Astronautical Science (ISAS) and NASA [3]. In addition to the flight development effort, the nanover concept and design are being refined through ongoing technology research efforts. The aim is to develop miniature, but scientifically capable, rovers that could easily fit within the projected mass/volume reserves of future missions to Mars and small planetary bodies.

The current nanover prototype features a novel wheeled mobility mechanism that allows it to execute motions beyond conventional rolling and turning. Its articulated mechanism of wheels on posable-struts can be thought of as a hybrid wheeled-legged mobility system. With this design, the rover is capable of operating with its chassis upside down, recovering from accidental overturning, and even hopping in very small gravity fields. Herein, we focus on the important mobility control feature of autonomous self-righting and present an approach to automatic discovery of associated motion control behaviors. We use the term *self-righting* to refer to the act of maneuvering the rover's articulated mobility mechanism to effect recovery from an initial overturned state to its nominal upright driving configuration. Due to the wide range of possible motions permitted by its mobility mechanism, considerable time and effort could be spent designing general self-righting motion sequences for the nanover. The problem is complicated further when resource limitations (e.g. available power, time, etc) or certain flight constraints must be considered in the solution. A control software design approach is proposed that is aimed at reducing the effort involved in developing self-righting behaviors that are sensitive to on-board resource limitations. The approach is based on the integration of evolutionary computing with a dynamics simulation environment for evolving and evaluating suitable motion behaviors. The

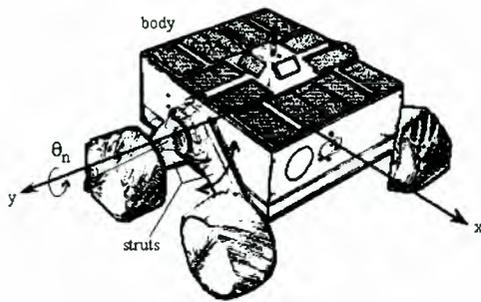


FIG. 1: Articulated nanorover prototype.

automated behavior design approach is outlined and the software infrastructure necessary for implementing the strategy is described.

2 NANOROVER MOBILITY

The current nanorover prototype is illustrated in Fig. 1. The rover's mobility mechanism is comprised of four wheels on articulated struts. Each wheel and strut can be actuated independently. The largest dimension (length) of the rover is 20 cm which makes it 30% the size of Sojourner, the Mars Pathfinder micro-rover. Each aluminum wheel contains a drive motor within, and is cleated with a helical tread on the outer surface to enhance traction and skid-steering performance. In addition to basic functionality for forward/reverse driving and turning, the high-mobility articulated mechanism provides the rover with the capability to self-right, as well as operate with its body/chassis upside down. This implies the ability to recover from overturning, and allows body pose control for preferential pointing of on-board science instruments. Aside from the rover's apparent miniature size, it is the capability to self-right which distinguishes it from many other planetary rover designs. Moreover, this capability enhances its survivability, and hence, the likelihood of mission success.

The rover has an on-board computer that can be programmed to execute autonomous sequences of strut, body, and wheel motions, which cause the vehicle to self-right (as well as perform other useful behaviors). Its suite of attitude sensors and motor actuators permits simultaneous coordinated control of strut articulation and body pose. As indicated in Fig. 1, the four struts can rotate in two directions about a common pivot axis (y -axis in the figure), however, struts on either side cannot rotate past one another. In addition to strut rotations, the body can be actuated to pitch about the same axis. These rotations constitute the articulation degrees of freedom θ_n ($n = 1-5$); the wheel motions provide four rolling degrees of freedom. Strut angles are measured by potentiometers;

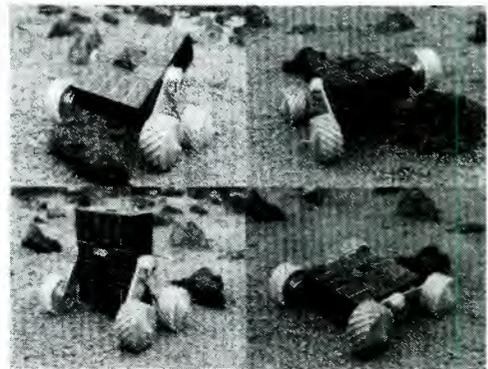


FIG. 2: Posable-strut and chassis configurations.

wheel rotational displacements are measured by encoders. The flight rover design includes sensors at each wheel for detecting proximity to, and contact with, the ground. It also includes a sun sensor for detecting body orientation relative to the sun. A variety of pose configurations that are possible with this mechanism are shown in Fig. 2.

Due to the flexibility of the mobility mechanism and chassis, a number of feasible motion sequences can be executed that result in successful self-righting from an initially overturned state. One possible sequence is illustrated in Fig. 3, in which the motion progresses from (a)–(f). From the initial overturned state in (a), the rover actuates its struts towards the terrain until its wheels make contact, (b). The same strut motion continues until the configuration in (d) is achieved. At this point, the body is actuated to its nominal upright configuration, (e)–(f). A single fixed sequence such as this may be inadequate as a general self-righting solution. While effective on relatively flat local terrain, it may fail if attempted in very close proximity to large rocks. A more general solution calls for an algorithm or set of control rules that assesses the overturned configuration via sensory perception, and produces expedient actuator controls. For completeness, the behavior should be able to prescribe control responses for the range of possible sensor stimuli. This can be achieved efficiently with behavior control rules that accept inputs that are partitioned into intervals, or even fuzzy sets [4].

2.1 PRACTICAL ISSUES

Some of the motion sequences that can be executed with the posable-strut mechanism are more favorable than others with regard to the total number of motions necessary (and therefore, power required), and the required execution time. Nanorovers used for flight missions rely on solar energy as their primary electrical power source. The flight nanorover

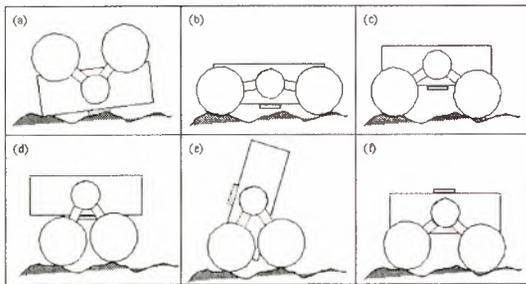


FIG. 3: Example self-righting sequence.

is designed to have most sides of its chassis populated by solar panels, with the primary solar panel located on the nominal top side. This design ensures that sufficient power will always be available for actuation of motors needed to self-right. The maximum size of the primary solar panel for these rovers is limited by the small footprint of the vehicles. As such, nanorovers must operate within the constraint of relatively low power budgets. Sufficient available on-board power for mobility actuators, science instruments, and communications is of primary concern for nanorovers. Designs for self-righting and other motion behaviors must be sensitive to on-board power constraints. Some of the most intuitive solutions (such as that in Fig. 3) may not sufficiently account for realistic on-board resource limitations. Therefore, it behooves the rover control engineer to explore the space of feasible solutions for behaviors that would minimize power consumption and comply with other operational constraints or flight rules. Execution time required for self-righting is also of concern since the frequency of unintentional overturning may be significant for nanorovers operating in certain environments and terrain-types. The cumulative time spent recovering from frequent overturning could easily detract from time allotted for science data gathering and navigation goals. An additional concern for nanorovers is the negative impact that dusty environments can have on solar panel efficiency. Due to their low profile relative to the terrain, dust could accumulate over time on the rovers' solar panels. The problem is only compounded each time the rovers overturn. This issue is currently being addressed by a dust mitigation approach planned for the flight rover, which is based on the use of an electronic dust rejection apparatus.

As an alternative to the tedious effort of examining all of the possible motion sequences, an automatic computational method of self-righting behavior design is proposed in the following section. The goal and expected result of the approach is the discovery of one or more viable self-righting behaviors that can be used as is, or as a starting point for further refine-

ment. The advantage is a savings in time and effort that would otherwise be spent searching the space of possible motion sequences.

3 SELF-RIGHTING EVOLUTION

In this section, we outline an approach to artificial evolution of self-righting behaviors. More specifically, we propose genetic programming for off-line learning of self-righting behaviors for nanorovers. A genetic programming (GP) system [5] computationally simulates the Darwinian evolution process by applying fitness-based selection and genetic operators to a population of candidate solutions, which are represented as computer programs or subroutines. The main distinction between genetic programming and genetic algorithms is that the former adapts hierarchical *symbolic* data structures (e.g. computer programs), while the latter adapts linear numerical data structures (e.g. bit strings or arrays of integers or reals). For our purposes, the computational structures undergoing adaptation are sets of condition-action rules of dynamically varying size and structure. That is, the population consists of behavioral rule sets, each represented as a tree data structure, of different numbers of rules. Tree nodes, or *genes*, may consist of components of a generic if-then rule construct and common logic connectives (e.g. **AND**, **OR**, and **NOT**), as well as input/output variables and parameters associated with the problem. Each set of rules constitutes a motion behavior that maps articulation, orientation, and wheel-contact sensor values into strut and body motions.

The objective of the GP system is to create a population of candidate self-righting behaviors, evaluate behaviors via dynamics simulation, and improve the population through artificial evolution until one or more highly fit solutions is discovered. All behavioral rule sets in the initial population are randomly created from syntactically valid combinations of genes. Descendant populations are created by genetic operators — primarily reproduction and crossover. For the reproduction operation, several behaviors selected based on superior fitness are copied from the current population into the next, i.e. the new generation. The crossover operation starts with two parental rule sets and produces two offspring that are added to the new generation. This operation selects a random portion of each parental tree structure and swaps them (while maintaining valid syntax) to produce the two offspring. GP cycles through the current population evaluating the fitness of each behavior based on its performance in computer simulations of the control system. After a numerical fitness is determined for each behavior, the genetic operators are applied to

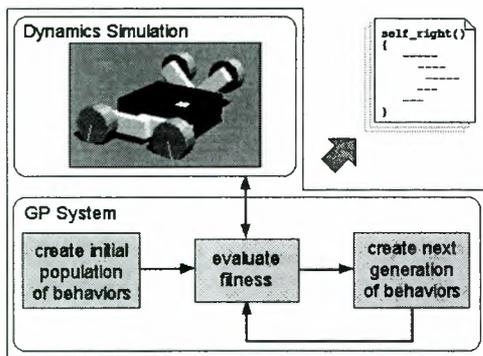


FIG. 4: Behavior evolution architecture.

the fittest behaviors to create a new population. This cycle repeats on a generation by generation basis until satisfaction of termination criteria (e.g. discovery of a highly fit behavior, lack of improvement, maximum generation reached, etc). The end result is the best-fit behavior that appeared in any generation.

The overall process is summarized as illustrated in Fig. 4. Candidate self-righting behaviors in the population evolve in response to selective pressure induced by their relative fitnesses for implementing the desired motion behavior. This population-based approach is particularly suitable for global search and optimization in large and/or multi-modal search spaces. The key distinction between such evolutionary search methods and a conventional gradient descent based approach is that, in the former, multiple points in the search space are sampled in parallel. The approach has been verified through numerous examples reported in the literature. In the definitive GP text [5], Koza has applied genetic programming to evolve computer programs that solve a number of interesting control problems. The same techniques have been successfully applied to search and optimization of robot manipulator trajectories [6], mobile robot control and navigation behaviors [7], and collective behaviors for multi-robot systems [8]. Each implementation differs in various problem-dependent ways. However, for robotic system applications, a common characteristic is the formulation of a fitness measure that drives the evolution and is coupled to a motion simulation. The viability of evolved behaviors is a function of the thoroughness of the evaluation process. Performance is based solely on evaluation of behavioral responses predicted by the simulator, and is computed by a user-prescribed fitness function. As such, the success of the approach depends in large part on the fitness function employed and the fidelity of the simulation environment. Each of these integral aspects is discussed further below.

3.1 BEHAVIOR EVALUATION

In order to apply evolutionary algorithms for behavior design, a measure of behavior fitness must be formulated to drive the process. It is important that the fitness function map observable parameters of the problem into a spectrum of values that differentiate the performance of behaviors in the population. If the spectrum of fitness values is not sufficiently rich, the fitness function may not provide enough information to guide GP toward regions of the search space where improved solutions might be found. For problems involving simulation of controlled behavior, a variety of performance attributes can be considered for inclusion in the fitness measure. Examples include a maximum number of time steps, explicit error tolerances, terminating physical events such as task success or failure, and penalties/rewards thereof. In general, selected performance attributes can be weighted to emphasize their relative importance in the search for candidate solutions. The fitness function is analogous to the performance measure of optimal control theory, or more generally, the objective function of optimization theory.

One approach to evaluating evolving candidate self-righting behaviors is to test them against a number of fitness cases, tabulate a performance score for each case, and average the scores to determine an overall fitness value. The initial postures for each fitness case should be chosen to represent an overturned configuration that can occur in the target environment. The number of fitness cases should be chosen such that they represent the search space sufficiently to allow the evolved strategy to generalize (i.e. handle unforeseen initial conditions). Fig. 3a is one example of a fitness case for the self-righting problem. A few additional examples are illustrated in Fig. 5. For each fitness case the goal is the same — recovery from an initial overturned state to achieve the nominal upright driving configuration.

Given the practical points expressed in Sect. 2.1, it would be prudent to formulate a fitness score based primarily on the estimated power consumed by motors (p), the time elapsed during execution (t), and the percentage of progress made ($\psi \leq 100$). Each of these performance attributes is measurable at the end of each fitness case. It is possible, however, to formulate the fitness evaluation such that performance is measured during fitness case execution. This was done in [8] where a reinforcement learning function was coupled with fitness evaluation to install a progress indication during fitness trials. Power consumption can be estimated from knowledge of motor performance characteristics and usage during execution. Elapsed time is determined based on simulation

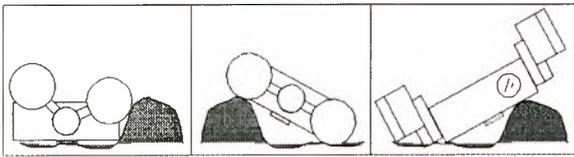


FIG. 5: Example fitness cases.

ticks starting from the beginning of the self-righting maneuver to the end of the trial. The amount of progress made is indicated by the percentage of angular displacement achieved by the chassis from the initial posture towards the desired nominal driving configuration. Secondary performance attributes from among the aforementioned examples can also be included in the formulation. With selected attributes defined, a fitness score $f(p, t, \psi)$ can be computed for trial runs through each fitness case. The overall fitness of a candidate self-righting behavior would be computed by averaging the scores over the total number of fitness cases defined. Suitable fitness formulas for self-righting would reward behaviors that consistently achieve (or come close to) the desired upright configuration in a timely manner, while minimizing power consumption.

3.2 DYNAMICS SIMULATION

A simulation environment is a key component of the approach described above. This is particularly true for evolution of rover behavior(s). One of the challenges of evolutionary robotics is the successful evolution of robust controllers in simulation. It was pointed out in [9] that the use of simulation environments of questionable fidelity tend to result in evolved behaviors that are not easily transferable to real robots. However, for developing rover systems designed to operate in unknown space environments, evolution in simulation is often the most practical option. Behaviors evolved in simulations must, however, be validated and verified to some extent on real rovers. The use of rover and environment simulators of reasonably high fidelity can mitigate such concerns. Pre-existing simulators are particularly useful in streamlining rover control and navigation software development efforts when prototype/flight hardware is unavailable or inaccessible.

A high-fidelity dynamics simulation system is available at JPL for use in this work. It is based on the JPL-developed DARTS/DSHELL [10] simulation tools. DARTS/DSHELL is a multi-mission spacecraft simulator with a real-time computational engine for flexible multi-body dynamics. It includes libraries of hardware models for various sensors, actuators, and motors. Its simulation infrastructure allows for interfaces

to a 3D animation viewer and rover research/flight software. The interface between rover software and the simulator enables software to issue control updates to the simulator and receive state/sensor data from the simulator. The computational engine computes dynamics of multi-body systems based on inertial properties of the bodies in the system and forces applied to those bodies. In this dynamics simulation system, the nanorover is modeled as a multi-body system of wheels, struts, and a chassis. Different friction models can be created to simulate characteristics of wheel-terrain interactions, and the gravitational acceleration can be varied as well. Currently, the DARTS/DSHELL spacecraft simulation tools are being leveraged to develop a related software simulation toolkit that is more germane to rovers [11]. These systems provide suitable environments for rover/terrain modeling and simulation that are useful for flight software design and development. When integrated with a genetic programming system, as described above, high-fidelity simulators provide a fitness evaluation medium for artificial evolution of rover behaviors.

4 ISSUES FOR SMALL BODIES

The approach as described thus far is nominally focused on the basic discovery of self-righting behaviors that might be feasible on Earth and Mars. The importance of a self-righting capability is magnified in the case of surface exploration on small bodies like asteroids. In this case, the gravitational fields are substantially weaker than those of Earth or Mars, and the likelihood of unintentional overturning is substantially higher. Before the proposed approach can be applied to evolve effective behaviors for small-body exploration, additional considerations must be factored into the dynamics simulation. Most notable among these are appropriate gravitational effects and terrain characteristics.

When accurate data about small bodies of interest are unknown, assumptions about gravity and terrain characteristics must be made. In a recent preliminary study [12], the mobility performance of a nanorover operating within a small-body gravity field was examined using a commercial dynamics simulation software package. In that study, assumptions were made about the environment of the near-Earth asteroid Nereus (4660), the primary target of the MUSES-C flight mission, which is less than one kilometer in diameter. The surface gravity of Nereus is expected to be $8\text{--}80\mu g$ [3]. In [12], $20\mu g$ was assumed. The aim of this small-body mobility study was to predict the rover's ability to maintain adequate tractive forces with the ground surface to achieve forward progress. Two wheel-terrain interaction models were consid-

ered. The first was based solely on Coulomb friction (with a friction coefficient of 0.5); the second was a combination of Coulomb friction and adhesive forces (thought to arise due to electrostatic attractions between the wheels and a dusty surface). To computationally evolve self-righting behaviors for such environments, the simulator used for behavior evaluation must be capable of representing different gravity fields and terrain types. The dynamics simulator mentioned above offers this flexibility.

Until additional facts are learned about Nereus, data presented in [3] and assumptions made in [12] will be used as a baseline for our computational behavior evolution experiments. For the upcoming flight mission, relevant new findings will be factored into the design of control and navigation behaviors for mobility on the target asteroid. The various desirable attributes of viable evolved behaviors will be identified for possible realization on the flight rover. This activity will be supported by high-fidelity computer simulations as well as hardware-based low-gravity simulations that focus on evaluating behaviors in the context of relevant mission scenarios and constraints.

5 SUMMARY AND CONCLUSIONS

Nanorovers with articulated mobility mechanisms are capable of a variety of maneuvers besides conventional rolling and turning. This paper has focused on the problem of autonomous self-righting and has expressed some of the practical aspects of the problem. An automated software design approach has been proposed for developing rover control behaviors for self-righting. Genetic programming is advocated as a means for offline learning using a high-fidelity dynamics simulation of the rover and environment. The proposed approach can be used to synthesize self-righting behaviors and optimize them based on performance feedback from the simulator, which can be interfaced with prototype rover control software. The integrated system would be beneficial for streamlining rover software design and development efforts.

In addition to self-righting behaviors, the approach can be applied to develop other functionalities for which solutions are not already well-defined. The necessary software infrastructure consists of an evolutionary computation kernel and a simulator of reasonable fidelity. The interested reader can find source code for implementing GP in the LISP programming language in [5]. Public domain implementations that are written in C or C++ are also available on the World Wide Web.

ACKNOWLEDGMENTS

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The dynamics simulation environment is being developed by Jack Morrison and Jeffrey Biesiadecki of the Jet Propulsion Laboratory.

REFERENCES

- [1] B. Wilcox et al. Nanorovers for planetary exploration. In *AIAA Robotics Technology Forum*, pages 11-1-11-6, Madison, WI, Aug 1996.
- [2] E. Tunstel, R. Welch, and B. Wilcox. Embedded control of a miniature science rover for planetary exploration. In *7th Intl. Symp. on Robotics with Applications, WAC'98*, Anchorage, Alaska, May 1998.
- [3] R. Jones et al. NASA/ISAS collaboration on the ISAS MUSES C asteroid sample return mission. In *3rd IAA Intl. Conf. on Low-Cost Planetary Missions, IAA-L98-0506*, Pasadena, CA, Apr 1998.
- [4] E. Tunstel. Mobile robot autonomy via hierarchical fuzzy behavior control. In *6th Intl. Symp. on Robotics and Manufacturing, WAC'96*, pages 837-842, Montpellier, France, May 1996.
- [5] J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- [6] Y. Davidor. *Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization*. World Scientific Publishing Co., Teaneck, NJ, 1991.
- [7] E. Tunstel and M. Jamshidi. On genetic programming of fuzzy rule-based systems for intelligent control. *International Journal of Intelligent Automation and Soft Computing*, 2(3):273-284, 1996.
- [8] S. Calderoni and P. Marcenac. Genetic programming for automatic design of self-adaptive robots. In W. Banzhaf et al., editors, *Genetic Programming: Proc. of the First European Workshop*. Springer, Paris, France, 1998.
- [9] M. Matarić and D. Cliff. Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems, Special Issue: Evolutional Robotics*, 19(1):67-83, Oct 1996.
- [10] J. Biesiadecki, A. Jain, and M.L. James. Advanced simulation environment for autonomous spacecraft. In *Intl. Symp. on Artificial Intelligence, Robotics and Automation in Space*, Tokyo, Japan, Jul 1997.
- [11] J. Yen, A. Jain, and J. Balaram. ROAMS: Rover analysis modeling and simulation software. In *Intl. Symp. on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, Jun 1999.
- [12] E.T. Baumgartner et al. Mobility performance of a small-body rover. In *7th Intl. Symp. on Robotics with Applications, WAC'98*, Anchorage, Alaska, May 1998.

Spacecraft Autonomy
Experiments on *Deep Space One*

VALIDATING THE DS1 REMOTE AGENT EXPERIMENT

P. Pandurang Nayak[†]

Douglas E. Bernard[¶]

Gregory Dorais[‡]

Edward B. Gamble Jr.[¶]

Bob Kanefsky[‡]

James Kurien^{||}

William Millar[‡]

Nicola Muscettola[§]

Kanna Rajan[‡]

Nicolas Rouquette[¶]

Benjamin D. Smith[¶]

William Taylor[§]

Yu-wen Tung[¶]

Abstract

This paper describes the validation of the Remote Agent Experiment. A primary goal of this experiment was to provide an on-board demonstration of spacecraft autonomy. This demonstration included both nominal operations with goal-oriented commanding and closed-loop plan execution, and fault protection capabilities with failure diagnosis and recovery, on-board replanning following unrecoverable failures, and system-level fault protection. Other equally important goals of the experiment were to decrease the risk of deploying Remote Agents on future missions and to familiarize the spacecraft engineering community with the Remote Agent approach. These goals were achieved by successfully integrating the Remote Agent with the Deep Space 1 flight software, developing a layered testing approach, and taking various steps to gain the confidence of the spacecraft team. In this paper we describe how we achieved our goals, and discuss the actual on-board demonstration in May, 1999, when the Remote Agent took control of Deep Space 1.

1 Introduction

May, 1999, represents a milestone in the history of the development of spacecraft autonomy. In two separate experiments, the Remote Agent, an AI software system, was given control of an operational spacecraft and demonstrated the ability to respond to high level goals by generating and executing plans on-board the spacecraft, all the time under the watchful eye of model-based fault diagnosis and recovery software.

[†]RIACS, NASA Ames Research Center, MS 269-2, Moffett Field, CA 94035.

Corresponding author: nayak@ptolemy.arc.nasa.gov

[§]Recom Technologies, NASA Ames Research Center, MS 269-2, Moffett Field, CA 94035.

[‡]Caelum Research, NASA Ames Research Center, MS 269-2, Moffett Field, CA 94035.

[¶]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

^{||}NASA Ames Research Center, MS 269-2, Moffett Field, CA 94035.

Current spacecraft control technology relies heavily on a relatively large and highly skilled mission operations team that generates detailed time-ordered sequences of commands or macros to step the spacecraft through each desired activity. Each sequence is carefully constructed in such a way as to ensure that all known operational constraints are satisfied. The autonomy of the spacecraft is limited.

The Remote Agent (RA) approach to spacecraft commanding and control puts more “smarts” on the spacecraft. In the RA approach, the operational rules and constraints are encoded in the flight software and the software may be considered to be an autonomous “remote agent” of the spacecraft operators in the sense that the operators rely on the agent to achieve particular goals. The operators do not know the exact conditions on the spacecraft, so they do not tell the agent exactly what to do at each instant of time. They do, however, tell the agent exactly which goals to achieve in a specified period of time.

Three separate Artificial Intelligence technologies are integrated to form the RA: an on-board planner-scheduler, a robust multi-threaded executive, and Livingstone, a model-based fault diagnosis and recovery system [5; 4]. This RA approach was flown on the New Millennium Program’s Deep Space One (DS1) mission as an experiment. The New Millennium Program is designed to validate high-payoff, cutting-edge technologies to enable those technologies to become more broadly available for use on other NASA programs.

The DS1 Remote Agent Experiment (RAX) had multiple objectives [2]. A primary objective of the experiment was to provide an on-board demonstration of spacecraft autonomy. This demonstration included both nominal operations with goal-oriented commanding and closed-loop plan execution, and fault protection capabilities with failure diagnosis and recovery, on-board replanning following unrecoverable failures, and system-level fault protection. These capabilities were demonstrated using in-flight scenarios that included ground commanding and simulated failures.

Other equally important, and complementary, goals of the experiment were to decrease the risk (both real and perceived) in deploying RAs on future missions and to familiarize the spacecraft engineering community with the RA approach to spacecraft command and control. These goals were achieved by a three-pronged approach. First, a successful on-board demonstration required integration

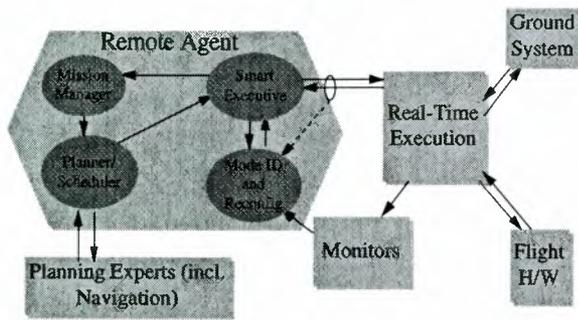


Figure 1: Remote Agent architecture.

of the RA with the spacecraft flight software. This integration provided valuable information on required interfaces and performance characteristics, and alleviates the risk of carrying out such integration on future missions. It also served to familiarize systems engineers and flight software engineers with the integration of RAs with traditional flight software. Second, a perceived risk of deploying RAs is related to its ability to synthesize new untested sequences in response to unexpected situations. We addressed this risk by demonstrating a layered testing methodology that serves to build confidence in the sequences synthesized by the RA in a variety of nominal and off-nominal situations. Third, the experiment was operated with close cooperation between RA team members and DS1 ground operators. This served to familiarize the ground operations community with benefits and costs of operating a spacecraft equipped with an RA.

The RAX was successfully executed on-board DS1 during the week of May 17–21, 1999. There were a few surprises along the way, which are discussed in a later section. These surprises pointed out some areas for improvement for future deployments of the remote agent. They also gave the team an opportunity to show off a number of benefits that the technology provides in terms of robust execution despite unexpected events, the ability to query the system to understand its state, as well as the ability to rapidly create and execute new mission profiles.

The remainder of the paper is organized as follows: Sections 2 and 3 describe the RA and the RAX scenarios; Section 4 describes RAX flight preparations for flight; Section 5 discusses the flight experiment itself, including surprises and the responses to these surprises; finally, Section 6 summarizes the paper.

2 Remote Agent Architecture

The RA architecture and its relation to flight software is shown in Figure 1. Viewed as a black-box, RA issues commands to real-time execution flight software (FSW) to modify spacecraft state, and receives state information through a set of monitors (MON) that filter data streams into a set of abstract properties. The RA itself is comprised of four components: a Mission Manager (MM), a Planner/Scheduler (PS) [3], a Smart Executive (EXEC) [6], and a Mode Identification and Reconfiguration module (MIR) [8].

MM formulates near-term planning problems based on a long-range mission profile representing the goals of the

mission. MM extracts goals for the next scheduling horizon, combines them with a projected spacecraft state provided by EXEC, and formulates a planning problem for PS. This decomposition into long-range mission planning and shorter-term detailed planning enables RA to undertake an extended mission with minimal human intervention.

PS takes as input a plan request from MM and produces a flexible, concurrent temporal plan for execution by EXEC. PS constructs plans using domain constraints and heuristics in its knowledge base; planning experts participate in the planning process by requesting new goals or answering queries posed by PS.

EXEC executes a plan by decomposing the high-level plan activities into primitives, sending out commands, and monitoring progress based on direct feedback from the command recipient or on inferences drawn by MIR. If some task cannot be achieved, EXEC may attempt an alternate method or may request a recovery from MIR. If the EXEC is unable to execute or repair the current plan, it cleanly aborts the plan and attempts to bring the spacecraft into a safe state while requesting a new plan from MM.

MIR is responsible for mode identification (MI) and mode reconfiguration (MR). MI observes EXEC issuing commands, receives events from MON, and uses model-based inference to deduce the state of the spacecraft and provide feedback to EXEC. MR serves as a recovery expert, taking as input a set of EXEC constraints to be established or maintained, and uses declarative models it shares with MI to recommend a single recovery action to EXEC.

All communication between RAX and the flight software was mediated by the RAX manager, a software task belonging to DS1 flight software. The RAX manager was also responsible for starting the RAX Lisp task at the start of the experiment. When RAX is terminated, either normally or by ground controllers, the RAX manager immediately stops any further communication between RAX and the flight software, and then stops the RAX Lisp task. The ability to tightly control RAX activity through the RAX manager was an important factor in convincing the DS1 project that ground controllers could easily recover control of the spacecraft from RAX.

3 Remote Agent Experiment scenarios

The design of the RAX scenarios was driven by the need to demonstrate the RAX validation objectives. The RAX scenarios were originally designed in mid-1997, and were largely unchanged until early 1999. However, in response to new operations constraints levied by DS1 and an unexpected anomaly during the experiment, we were forced to significantly redesign the scenarios. Our ability to quickly redesign the RAX scenarios provides objective evidence of the flexibility of the RAX technology. In this section we describe the validation objectives and the various RAX scenarios.

3.1 RAX validation objectives

The DS1 project required formal validation objectives from each of the 12 technologies being validated on DS1. The validation objectives for RAX were broken down into specific objectives for each of the three engines as follows.

PS's validation objectives were to: (a) generate plans on-board; (b) reject low-priority, unachievable goals; (c) replan following a failure; (d) generate back-to-back plans; and (e) enable modification of mission goals from ground. EXEC's validation objectives were to: (a) provide a low-level commanding interface; (b) initiate on-board planning; (c) execute plans generated both on-board and on the ground; (d) recognize and respond to plan failure; and (e) maintain required properties in the face of failures. MIR's validation objectives were to: (a) confirm executive command execution; (b) demonstrate model-based failure detection, isolation, and recovery; and (c) demonstrate ability to update MIR state via ground commands.

3.2 Original RAX scenarios

The original RAX scenarios consisted of a 12 hour scenario and a 6 day scenario. The 12 hour scenario was designed as a confidence builder for the DS1 project. It involved neither on-board planning nor thrusting with the Ion Propulsion System (IPS). Rather, the plan was to be generated on the ground, uplinked to the spacecraft, and executed by EXEC and MIR. The scenario included imaging asteroids with the MICAS camera to support optical navigation, a simulated sensor failure* scenario, and demonstration of low-level commanding to flip a switch. The planning of optical navigation imaging provided the planner the opportunity to reject low-priority, unachievable goals since the optical navigation windows had time only to image a subset of the asteroid goals.

The 6 day scenario was to be run following successful completion of the 12 hour scenario. The 6 day scenario included both on-board planning and operating the IPS, and was the full-up test of RA. The scenario was divided into 2 horizons. At the start of the scenario, PS generated a plan for the first horizon which included MICAS imaging for optical navigation and IPS thrusting. Execution of the first plan also included a ground command to modify the goals for the second horizon. At the end of the optical navigation window PS planned to switch off the MICAS camera. However, a stuck on failure injection in the camera switch prevented RA from turning off the camera, leading to a plan failure. This led to a replan, which produced a second plan with the camera being left on. The second plan also included an activity to produce a plan for the second horizon (the third plan in the scenario), which was to be executed back-to-back with the second plan. While the second plan was being executed, the switch failure injection was undone and ground informed MIR that the switch is now fixed. The execution of the third plan included IPS thrusting, optical navigation imaging, and two simulated failures, a communication failure on the 1553 bus, and a thruster valve stuck closed failure.

Together, these two scenarios demonstrate all RAX validation objectives.

3.3 2 day RAX scenario

The 12 hour and 6 day scenarios were used for all RAX integration and testing until the beginning of March, 1999. At that point, we were informed by the DS1

*All failure scenarios were simulated failures, though they appeared to be real to RAX.

project that they did not want us to switch off the MICAS camera due to concerns about thermal effects. Furthermore, we were required to provide only about 12 hours of IPS thrusting, to ensure that DS1 would be on track for its asteroid encounter in July, 1999. These changes meant that the 6 day scenario had to be changed at this late date, since it switched off the camera 3 times (not including the failed attempt during the failure injection) and thrusted for a total of about 4 days. We responded by developing a 2 day scenario. The 2 day scenario was similar to a compressed 6 day scenario, except that the simulated MICAS switch failure was active for the whole duration of the scenario. This prevented RA from ever switching off the camera. Furthermore, the 2 day scenario had only about 12 hours of IPS thrusting. Our ability to quickly develop a new scenario in response to these new constraints was viewed very favorably by the DS1 project.

3.4 6 hour RAX scenario

An anomaly was encountered while executing the 2 day scenario on-board DS1 which led to early termination of the 2 day scenario (see Section 6). At this time, approximately 70% of the RAX validation objectives had been achieved. To achieve the remaining 30% of the objectives, we quickly put together a 6 hour scenario which included IPS thrusting, three failure scenarios, and back-to-back planning. This scenario was executed on the spacecraft a little over 2 days later, thus completing RAX validation. The remarkable thing about this scenario was not just that we could quickly design and test it at such short notice, but rather that the DS1 project had already gained enough confidence in the RA that they allowed on-board execution of this new scenario within days of conception!

4 Preparing the Remote Agent Experiment for flight

We took a number of steps to prepare RAX for flight. In this section we highlight some of the key steps, including preparing the Lisp for flight, testing RAX, software change control, special considerations involved in testing PS, and the operational readiness tests. A comprehensive discussion of our integration methodology, a central element in preparing RAX for flight, is beyond the scope of this paper. Suffice it to say that developers acted as front-line testers during our various integration efforts, and hence identified and resolved a significant number of bugs (often unreported in our formal problem reporting system). As a result, formal testing on high fidelity platforms found few bugs, since most of the problems on these platforms had been discovered and resolved during integration.

4.1 Preparing Lisp for Flight

One important aspect of the RAX preparation for flight was the preparation of Lisp for flight. The RAX software development and runtime environment was based on CommonLisp, in particular the Harlequin Lispworks product []. The use of Lisp was appropriate given the background of the RAX developers, the early inheritance of code libraries, and the hardware independence of the high-level software interfaces between RAX and the rest

of flight software. However, with the choice of Lisp came some unique challenges. These challenges fell into two rather broad categories: resource constraints and flight software interfaces.

Like all spacecraft, DS1 placed constraints on computational and telecommunication bandwidth (both uplink and downlink) resources. For computational resource, DS1 has a total of 128 MB RAM, 16 MB EEPROM, and a 20 MHz RAD6k. During the RAX experiment time, the uplink and downlink data rates were about 1 kbps and 4 kbps, respectively. Based on early estimates, RAX was allocated 32 MB of RAM, 16 MB of file space and up to 45% of the CPU. At the time of this allocation it was not clear if RAX could meet these resource constraints.

To fit within the 32 MB memory allocation and the CPU fraction constraints, the RAX team thoroughly analyzed their code for memory and performance inefficiencies and employed a “tree-shaking/transduction” process to the Lisp image. The analysis is, of course, common for any high performance software. However, transduction is Lisp-specific and arises from the tight coupling of the Lisp runtime and development environments. Transduction removes the unneeded parts of the development environment, e.g., the compiler, debugger, windowing system. The result is a significantly smaller image, both in terms of file system and runtime memory. During RAX testing, peak memory usage was measured at about 29 MB, which was more than was actually observed in flight.

To reduce the uplink time and the spacecraft file system usage, we employed a custom Lisp image that supported ground-based compression and spacecraft-based decompression. Upon completion of the transduction process the RAX Lisp image was compressed by a factor of about 3 to 4.7 MB and uplinked to the spacecraft. On-board decompression was initiated at the start of each RAX run, with the file being inflated directly into the 32 MB RAX memory space. Use of this custom compression drastically reduced the file uplink time and kept the RAX file space usage within the agreed upon limits.

Besides the resource constraints, we also dealt with a complicated flight software interface. The flight software was written in the ‘C’ programming language and ran on the VxWorks operating system. Lisp and ‘C’ interacted through Lisp’s foreign function interface. This interface was the source of many early problems, primarily caused by discrepancies between data structure alignments assumed by the Lisp and ‘C’ compilers. These problems were quickly discovered and resolved with the help of an extensive test suite that tested a large number of function parameter variations.

Another problem arose in preparing the Lisp multi-threading system for flight. Originally, the Lisp thread scheduler relied on a high frequency external, periodic wakeup call, issued at interrupt level. However, this went against the design principles of the DS1 flight software. Hence, we had to significantly change Lisp’s approach to thread preemption to use a lower frequency wakeup call implemented with flight software timing services.

Most of the late integration problems with RAX Lisp arose because of the VxWorks port. As RAX moved from testbed to testbed, ever closer to the final spacecraft configuration, low-level Lisp problems arose. The problems were consistently of two types: a function as-

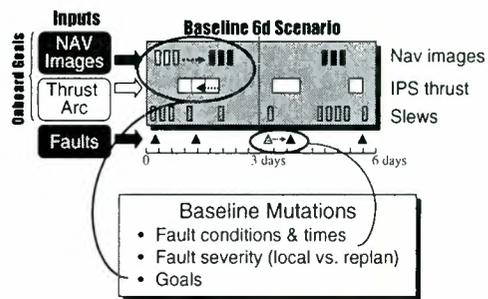


Figure 2: Baseline Variations

sumed by Lisp to be present was not present or a function was present but did not perform as expected by Lisp. The first type of problem was resolved by consistent application of a detailed RAX and FSW build process. The second type of problem was addressed on a case-by-case basis. Solutions to these problems were made difficult due to the reduced debugging visibility as testbeds assumed the spacecraft configuration. We benefited from the dedicated efforts of both Harlequin and the DS1 FSW team.

4.2 Testing the Remote Agent

Autonomous systems, such as the RA, need to respond robustly in a wide range of situations. In order to verify that they respond correctly in all situations would require a huge number of test cases. To make matters worse, the tests should ideally be run on high-fidelity testbeds, which are heavily oversubscribed, difficult to configure correctly, and cannot run faster than real time, e.g., we could run only 10 tests in four weeks on one of DS1’s high-fidelity testbeds. To address these problems, we employed a “baseline testing” approach to reduce the number of tests, and exploited several lower-fidelity testbeds to increase the number of tests we could run [7].

The baseline scenarios we used were the 12 hour and 6 day scenarios discussed above. We tested a number of nominal and off-nominal variations around these scenarios. The variations comprised variations in spacecraft behavior that we might see during execution and changes to the baseline scenario that might be made prior to execution. This included variations to the goals in the mission profile, variations in when faults might occur, and variations in the FSW responses (see Figure 2).

The tests were distributed among the low, medium, and high fidelity platforms. The two low-fidelity platforms were the “babybed” and “radbed”. The babybed had a non-radiation hardened PowerPC CPU and ran with simulators written by the RAX team. The radbed was identical to the babybed, except that it had a flight RAD6000 CPU. The medium fidelity platform was the “papabed”, which had a flight CPU, bus, and memory and official DS1 simulators. The highest-fidelity platforms, the “hotbench” and “testbed”, had flight computers and were connected to flight hardware (flight spares)

Platform	Fidelity	CPU	Hardware	Availability	Speed
Spacecraft	Highest	Rad6000	Flight	1 for DS1	1:1
DS1 Testbed	High	Rad6000	Flight spares + DS1 sims	1 for DS1	1:1
Hotbench	High	Rad6000	Flight spares + DS1 sims	1 for DS1	1:1
Papabed	Med	Rad6000	DS1 simulators only	1 for DS1	1:1
Radbed	Low	Rad6000	RAX simulators only	1 for RAX	1:1
PowerPC	Lowest	PowerPC	RAX simulators only	2 for RAX	7:1

Table 1: DS1 Testbeds

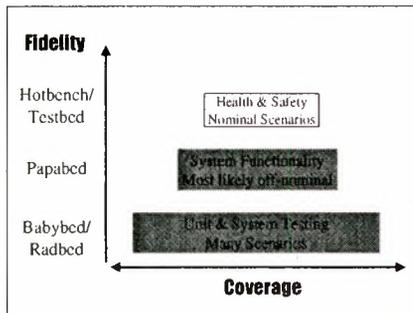


Figure 3: Pyramid Testing Approach

where feasible (see Table 1).

The architecture of RA allowed us to run certain tests on lower-fidelity testbeds and be confident that their results would hold on higher-fidelity testbeds. Specifically, the RA commands and monitors the spacecraft through well-defined interfaces with the FSW. Those interfaces were the same on all platforms, as were the range of possible responses. Only the fidelity of the responses improved with platform fidelity. This allowed us to exercise a wide range of nominal and off-nominal behaviors on the babybeds and radbed, test the most likely off-nominal scenarios on the papabed, and test only the nominal scenarios and certain performance and timing related tests on hotbench and testbed. This “pyramid” approach to testing is summarized in Figure 3.

The remainder of this section describes the tests on each of the testbeds, and discusses the effectiveness of our testing approach given the benefit of hindsight.

Babybed and radbed testing

Each of the RA modules devised a test suite of nominal and off-nominal scenarios that isolated and exercised key behaviors in each module. This involved testing about 200 variations of the initial state and goals of the planner, while exercising MIR in hundreds of the likeliest failure contexts. The PS and MIR tests were used for testing EXEC, and the system-level interaction of all modules was exercised by a suite of twenty additional scenarios. These tests were run rapidly on the babybeds and radbed, with simulators that permitted faster than real-time execution and exploited RA’s ability to “warp” over long periods of idle time. Even with this increased speed, running a scenario was a time-consuming and error-prone process. To alleviate this, we designed an

automated testing tool that accepted an encoded scenario description as input, controlled the simulator and ground tools to execute the scenario, stopped the test when appropriate by monitoring the telemetry stream, and stored all logs and downlinked files for later examination. This rapid data collection led to a total running time of about one week for all tests, since tests could be scheduled overnight and required no monitoring. Analyzing the results of the tests, however, was still a time consuming process. These tests were run after each major RAX software release. We identified (and resolved) over 800 bugs in six months.

Papabed testing

Once we delivered a “frozen” version of RA, we ran six off-nominal system test scenarios on the papabed. These corresponded to the most likely and highest-impact scenarios. No bugs were detected in these scenarios, probably because RA responses to off-nominal situations were well tested on the babybed.

Hotbench and testbed testing

The hotbench and testbed was reserved for testing the nominal scenarios, and for testing a handful of requirements for spacecraft health and safety. RAX was designed with a “safety net” that allowed it to be completely disabled with a single command sent either by the ground or by on-board FSW fault protection. Hence, the only ways in which RAX could affect spacecraft health and safety was by consuming excessive resources (memory, downlink bandwidth, and CPU) or by issuing improper commands. We tested the resource consumption cases by causing RAX to execute a Lisp script that consumed those resources. We guarded against improper commands by having subsystem engineers review the execution traces of the nominal scenarios, and doing automated flight rule checking. The nominal scenarios were run in conditions that were as close to flight-like as possible.

4.3 Software change control

As the date of the flight experiment drew closer, our perspective on testing changed. Throughout 1998 the main goal of testing was to discover bugs in order to fix them in the code. Starting in January 1999 the discovery of a bug did not automatically imply a code change to fix it. Instead, every new problem was reported to a Change Control Board (CCB) composed by senior RAX project members. Every bug and proposed fix was presented in detail, including the specific lines of code that needed to change. After carefully weighing the pros and cons of making the change, the board voted on whether or not to allow the fix. Closer to flight, DS1 instituted its own CCB to review RAX changes.

As time progressed, the CCB became increasingly conservative and the bias against code modifications significantly increased. This is demonstrated by the following figures. In total, 66 change requests were submitted to the RAX CCB. Of these, 18 were rejected amounting to a 27% rejection rate. The rejection rate steadily increased as time passed: 8 of the last 20 and 6 of the last 10 submitted changes were rejected.

The reason for this increase in conservatism is easily explained. Every bug fix modifies a system that has already gone through several rounds of testing. To ensure that the bug fix has no unexpected repercussions, the modified system would need to undergo thorough testing. This is time consuming, especially on the higher fidelity testbeds, so that full revalidation became increasingly infeasible as we approached flight. Therefore, the CCB faced a clear choice between flying a modified RAX with little empirical evidence of its overall soundness or flying the unmodified code and trying to prevent the bug from being exercised in flight by appropriately restricting the scenario and other input parameters. Often, the answer was to forego the change.

4.4 Testing the PS module

As discussed above, the PS module had undergone extensive testing throughout 1998 using variations of the 12 hour and 6 day scenarios. To generate these variations, we started by identifying the parameters that define a scenario. Test cases were generated using the "Latin squares" method [1] that ensured every pair of parameter values occurred in some test case. This approach was very effective in finding bugs, and resulted in a majority of the 211 PS problem reports filed in that period.

However, as we entered 1999, new problems were discovered in PS outside of the formal testing process. This resulted in 22 change requests submitted to the RAX CCB, a little over 9% of the total PS problem reports. The vast majority of these problems consisted of PS operating correctly but being unable to find a plan within the allocated time limit since its search was "thrashing". These problems were particularly serious since they could easily arise in off-nominal situations during flight.

There were several reasons for this situation:

1. The ranges of some parameters turned out to be different than those assumed by PS testing, e.g., PS testing assumed turn durations were at most 20 minutes, while actual turns could take over an hour. This created stress situations not considered by PS testing.
2. Planning problems became more challenging when we transitioned from the 6 day scenario to the 2 day scenario. The temporal compression led to the disappearance of slack time between activities. In the 6 day scenario PS could exploit this slack to achieve subgoals without backtracking. In the 2 day scenario backtracking became necessary, revealing additional brittleness in the PS chronological backtracking search.
3. A more fundamental issue was the independence between the PS test generator and the structural characteristics of the domain model. This led to the test generator missing a number of stress cases. For

example, one problem depended upon the specific values of three continuous parameters: the time to start up the IPS engine, the time to the next optical navigation window, and the duration of the turn from the IPS attitude to the first asteroid. An equation relating these parameters can crisply characterize the stress situations. Unfortunately, automatically generated test cases based on covering pairwise interactions of parameter values cannot reliably detect such problems.

Given the late date at which these new problems were discovered, it was not feasible to modify the test suite to test extended variations around the new baseline. Instead, we focused on the just the most crucial variation: the time at which replans might occur. The objective was to ensure that the planner was robust to any replanning contingency. Two steps were needed to accomplish this. First, the new 2 day scenario was designed to guarantee that the harmful constraint interactions of the PS domain model would be avoided under any hypothetical replanning contingency. The idea was to ensure that PS could always return a plan within the given time limit. Second, a new PS test suite was carefully designed and run to ensure that this was indeed the case.

The design methodology for this new PS test suite is instructive. Exhaustive generation of all possible plans was clearly impossible. Instead, using our knowledge of the PS model, we manually identified boundary times at which the topology of the plans would change. We identified 25 such boundary times and generated a total of 88 test cases corresponding to plans starting at, near, or between boundary times. This led to the discovery of two new bugs. Furthermore, analysis of the test results showed that PS would fail to find a plan at only about 0.5% of all possible start times. Although the probability of this failure was extremely low, contingency procedures were developed to ensure that the experiment could be successfully continued even if this PS failure actually occurred.

We used the above test suite design methodology only toward the end of RAX, after the PS model and code had been frozen. However, we believe that this (currently manual) analysis method can be generalized and extended to provide an automatic PS testing procedure throughout the development process for new application domains.

4.5 Operational Readiness Tests

In addition to testing RAX, preparing RAX for operations involved preparing operational procedures and sequences for running the experiment and identifying contingency procedures. The operational readiness tests (ORTs) were a "dress rehearsal" of the procedures and contingencies, and were intended to familiarize the operations team with the procedure and to identify problem areas.

We performed two ORTs. The first ran through the first several hours of the 12 hour scenario and was primarily intended to exercise the procedures for starting RAX. This involved configuring the spacecraft, filesystem, and memory to the state required to start RAX. The second ORT ran through the entire 2 day scenario. The operations team monitored key events in the scenario, with breaks in between. This proved to be an

effective way to monitor the experiment without unduly taxing the operations team. During actual spacecraft operations we followed a similar approach, though the RA team monitored the experiment around the clock.

The other purpose of the ORT was to exercise the RAX ground tools in an operations environment. During the two ORTs, RAX was run on the hotbench and the data was sent to workstations in the mission control center, some of which were running the RAX ground tools. The tools performed well, although we did identify a number of shortcomings, which we proceeded to resolve prior to flight.

5 The Remote Agent Experiment in flight

RAX was scheduled to be performed on DS1 during a three week period starting May 10, 1999. This period included time to retry the experiment in case of unexpected contingencies. On May 6, 1999, DS1 encountered an anomaly that led to spacecraft safing. Complete recovery from this anomaly took about a week of work by the DS1 team, both delaying the start of RAX as well as taking time away from their preparation for the asteroid encounter in July, 1999. In order not to jeopardize the encounter, the DS1 project also decided to reclaim the third RAX week for encounter preparation, leaving only the week of May 17th, 1999, for RAX. However, to maximize the time to try the more important 2 day experiment, they agreed to go ahead with the 2 day experiment without first doing the confidence building 12 hour experiment. This decision was strong evidence that the DS1 project had already developed significant confidence in RAX during pre-flight testing.

On Monday, May 17th, 1999, at 11:04 am PDT, we received a telemetry packet that confirmed that the 2 day RAX scenario had started on DS1. Shortly thereafter, PS started generating the first plan. The first plan was generated correctly, but not before an unexpected circumstance created some apprehension in us.

PS telemetry indicated that PS was generating the plan following a different search trajectory than what we had observed in ground testing. Since the conditions on the spacecraft were practically identical to those on the ground testbeds, there was no apparent reason for this discrepancy. Subsequently, the cause for this discrepancy was traced back to the spacecraft and papabed differing on the contents of the file containing asteroid goals; PS was actually solving a slightly different problem than it had solved on the ground! Thus, this unexpected circumstance allowed us to demonstrate that PS problem solving was robust to last minute changes in the planning goals, increasing the credibility of the autonomy demonstration.

The 2 day scenario continued smoothly and uneventfully with the simulated MICAS switch failure, the resulting replan, long turns to point the camera at target asteroids, optical navigation imaging during which no communication with DS1 was possible, and the start of IPS thrusting. However, around 7:00 am on Tuesday, May 18, 1999, it became apparent that RAX had not commanded termination of IPS thrusting as expected. Although plan execution appeared to be blocked, telemetry indicated that RAX was otherwise healthy. The spacecraft too was healthy and in no apparent danger.

The decision was made to use EXEC's ability to handle low-level commands to obtain more information regarding the problem. Once enough information had been gathered, the decision was made to stop the experiment. By this time an estimated 70% of the RAX validation objectives had already been achieved.

By late Tuesday afternoon the cause of the problem was identified as a missing critical section in the plan execution code. This created a race condition between two EXEC threads. If the wrong thread won this race, a deadlock condition would occur in which each thread was waiting for an event from the other. This is exactly what happened in flight, though it had not occurred even once in thousands of previous races on the various ground platforms. The occurrence of this problem at the worst possible time provides strong impetus for research on formal verification of flight critical systems. Once the problem was identified, a patch was quickly generated for possible uplink.

Following the discovery of the problem, we generated a 6 hour RAX scenario to demonstrate the remaining 30% of the RAX validation objectives. This new scenario was designed, implemented, and tested, together with the patch, on papabed overnight within about 10 hours. This rapid turn around allowed us to propose a new experiment at the DS1 project meeting on Wednesday. The DS1 project decided to proceed with the new scenario. However, they decided not to uplink the patch, citing insufficient testing to build adequate confidence. In addition, based on the experience on various ground testbeds, the likelihood of the problem recurring during the 6 hour test was deemed to be very low. Nonetheless, we developed and tested a contingency procedure that would enable us to achieve most of our validation objectives even if the problem were to recur.

The DS1 project's decision not to uplink the patch is not surprising. What was remarkable was their ready acceptance of the new RAX scenario. This is yet more evidence that the DS1 project had developed a high level of confidence in RA and its ability to run new mission scenarios in response to changed circumstances. Hence, although caused by an unfortunate circumstance, this rapid mission redesign provided unexpected validation for RA.

The 6 hour scenario was activated Friday morning. The scenario ran well until it was time to start up the IPS. Unfortunately, an unexpected problem in some supporting software failed to confirm an IPS state transition, thus causing RA to (correctly) stop commanding the IPS startup sequence. The underlying cause of this problem was still under investigation as of May 28, 1999. Since this situation was out of scope for RAX, the resulting RA state was inconsistent with spacecraft state. Fortunately, the discrepancy proved to be benign. Hence, RA was able to continue executing the rest of the scenario to achieve the rest of its validation objectives.

As a consequence of the two flight scenarios, RAX achieved 100% of its validation objectives.

6 Summary

The primary goal of RAX was to demonstrate that Artificial Intelligence technologies could achieve high-level autonomous control of a spacecraft including:

- goal-oriented commanding;

- closed-loop planning and execution;
- spacecraft state inferencing and failure detection;
- closed-loop model-based failure diagnosis and recovery;
- on-board re-planning as a response to unrecoverable failures; and
- system-level fault protection.

Familiarizing the spacecraft engineering community with these technologies and laying the foundation for more extensive applications of RA were also important goals. These goals were achieved by the design of RA, its integration with the DS1 flight software on spacecraft testbeds, its layered testing, two operational readiness tests with ground control personnel, and successful commanding of the spacecraft during the week of May 17-21, 1999.

As a result of the Remote Agent project, we believe that the willingness of NASA missions to deploy highly-autonomous systems has increased. Moreover, the NASA Ames Research Center and the Jet Propulsion Laboratory have recognized this contribution by nominating RA for NASA's prestigious Software of the Year award.

Acknowledgments

We gratefully acknowledge the DS1 team and Harlequin, without whom the Remote Agent Experiment would not have been possible. We would also like to thank the many past contributors to the Remote Agent adventure and its many supporters over the past four years. This paper describes work performed at the NASA Ames Research Center and at the Jet Propulsion Laboratory, California Institute of Technology, under contract from the National Aeronautics and Space Administration.

References

- [1] D. M. Cohen, S. R. Dalal, J. Parelius, and G. C. Patton. The combinatorial design approach to automatic test generation. *IEEE Software*, pages 83-88, September 1996.
- [2] Douglas E. Bernard *et al.* Design of the remote agent experiment for spacecraft autonomy. In *Proceedings of the IEEE Aerospace Conference*, 1998.
- [3] Nicola Muscettola. HSTS: Integrating planning and scheduling. In Mark Fox and Monte Zweben, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.
- [4] Nicola Muscettola, P. Pandurang Nayak, and Brian C. Williams Barney Pell. Remote Agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103:5-47, 1998.
- [5] Barney Pell, Douglas E. Bernard, Steve A. Chien, Erann Gat, Nicola Muscettola, P. Pandurang Nayak, Michael D. Wagner, and Brian C. Williams. An autonomous spacecraft agent prototype. *Autonomous Robotics*, 5(1), March 1998.
- [6] Barney Pell, Erann Gat, Ron Keesing, Nicola Muscettola, and Ben Smith. Robust periodic planning and execution for autonomous spacecraft. In *Proceedings of IJCAI-97*, 1997.
- [7] Benjamin Smith, William Millar, Julia Dunphy, Yu wen Tung, P. Pandurang Nayak, Edward B. Gamble Jr., and Micah Clark. Validation and verification of the remote agent for spacecraft autonomy. In *Proceedings of the 1999 IEEE Aerospace Conference*, 1999.
- [8] Brian C. Williams and P. Pandurang Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of AAAI-96*, pages 971-978, 1996.

Flight Validation of On-Demand Operations: The Deep Space One Beacon Monitor Operations Experiment

Jay Wyatt, Rob Sherwood, Miles Sue, John Szijjarto

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, MS 301-270
Pasadena, California 91109
Phone: 818-354-1414, Fax: 818-393-6004
Email: e.j.wyatt@jpl.nasa.gov

ABSTRACT

The Beacon Monitor Operations Experiment is one of twelve new technologies currently being flight validated on NASA's Deep Space One mission. The technology enables a spacecraft to routinely indicate the urgency of ground contact using a tone signal rather than telemetry while also summarizing onboard data until a telemetry downlink is required. The two subsystems (tone communication and onboard summarization) have been deployed on DSI and are in the final stages of flight-testing. The system can be used by missions to lower operational cost and in some instances to decrease mission risk. NASA will see a measurable unburdening of the antenna network if several missions use the technology. This paper provides a description of the technology and shares the results to date.

1.0 INTRODUCTION

The budget environment that has evolved since the advent of NASA's Faster, Better, Cheaper initiative has caused mission risk policies and mission designs to change in ways that have been conducive to the inception of new operations concepts and supporting technologies. Such was the case when the beacon monitor concept was conceived to enable a mission to Pluto to be achieved within the budget constraints passed down from NASA. The technology was accepted into the New Millennium Program and baselined for flight validation on the Deep Space One Mission. As the technology was being developed for DSI, the NASA community has expressed a growing interest and acceptance of adaptive operations and onboard autonomy.

In traditional mission operations, the spacecraft receives commands from the ground and in turn transmits telemetry

in the form of science or engineering data. With beacon monitoring, the spacecraft sends a command to the ground that instructs the ground personnel how urgent it is to track the spacecraft for telemetry. There are only four such commands. Thinking of beacon operations in this way creates a paradigm shift over the way we traditionally approach operations. Also, it is very important to not think of the tone message as just a little bit of telemetry. If one does this, it is easy to make the argument that a little more telemetry is better. Our approach is one where telemetry is only transmitted when it is necessary for ground personnel to assist the spacecraft or otherwise very infrequently if the spacecraft is fortunate enough to go long periods (a month or so) without requiring ground assistance. When telemetry tracking is necessary the intelligent data summaries contain the most relevant information to provide full insights into spacecraft activities since the last contact. The key challenge has been to develop an architecture that enables the spacecraft to adaptively create summary information to make best use of the available bandwidth as the mission progresses such that all pertinent data is received in one four to eight hour telemetry pass.

2.0 DSI BMOX SUBSYSTEMS

It was required that two subsystems be designed and developed to implement the desired functionality for the DSI experiment. These are, in fact, standalone innovations. Although they are being presented here primarily in support of cruise phase operations, there has also been interest in applying these technology components to other domains. Other potential applications include using in-situ beacons at Mars, adapting tone messaging and summarization to earth orbiters, using beacons for science event detection and notification, and in utilizing the tone system to reduce mission risk due to spacecraft operability constraints.

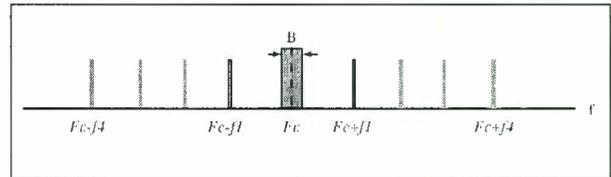
2.1 TONE SYSTEM

There are four tone signals and each uniquely represents one of the four urgency-based beacon messages. The DS1 tone definitions are summarized in Table 2.1.1. These tones are generated as the spacecraft software reacts to real-time events.

Table 2.1.1
Tone Definitions

Tone	Definition
Nominal	Spacecraft is nominal, all functions are performing as expected. No need to downlink engineering telemetry.
Interesting	An interesting and non-urgent event has occurred on the spacecraft. Establish communication with the ground when convenient. <u>Examples</u> : device reset to clear error caused by SEU, other transient events.
Important	Communication with the ground needs to be achieved within a certain time or the spacecraft state could deteriorate and/or critical data could be lost. <u>Examples</u> : memory near full, non-critical hardware failure.
Urgent	Spacecraft emergency. A critical component of the spacecraft has failed. The spacecraft cannot autonomously recover and ground intervention is required immediately. <u>Examples</u> : PDU failure, SRU failure, IPS gimbal stuck.
No Tone	Beacon mode is not operating, spacecraft telecom is not Earth-pointed or spacecraft anomaly prohibited tone from being sent.

RF carrier by a squarewave subcarrier using a 90 degrees modulation angle. The carrier (f_c) is completely suppressed. The resulting downlink spectrum consists of tones at odd multiples of the subcarrier frequency above and below the carrier. Four pairs of tones are needed to represent the four possible messages.



B =Frequency uncertainty f_c =Carrier frequency
 f_i =Subcarrier frequency for the i^{th} message

Figure 2.1.2
Tone Signal Structure

The goal is to reliably detect the monitoring messages with zero dB-Hz total received signal-to-noise-spectral-density ratio (P_t/N_o) using a 1000 second observation time. Future missions are assumed to carry a low-cost auxiliary oscillator as a frequency source, instead of a more expensive, ultra-stable oscillator. The downlink frequency derived from the auxiliary oscillator is not precisely known due to frequency drifts caused by on-board temperature variations, aging, and uncorrected residual Doppler frequency. In addition, the downlink frequency also exhibits short-term drift and phase noise. These factors were taken into consideration in the design of the monitoring signal detector.

Urgent Beacon tones on DS1 are sent when the spacecraft fault protection puts the spacecraft in standby mode. This condition occurs when the fault protection encounters a fault that it cannot correct. Standby mode halts the current command sequence, including IPS thrusting. During the DS1 tone experiment, the Beacon tone can be sent regularly at a prescheduled time, i.e., 30 to 60 minutes per day. The tone cannot be operated continuously because DS1 requires as much power as possible for IPS thrusting and the tone transmission uses some of the thrusting power. Routine operational use of the beacon monitor system is currently being explored for the DS1 extended mission, scheduled to begin in September of 1999.

It is important to communicate the urgency of ground response using a telecommunications method that has a low-detection threshold and short detection times. Ease of detection translates to lower cost operations. The signal structure is shown in Figure 2.1.2. Each message is represented by a pair of tones centered about the carrier frequency. Tones are generated by phase-modulating the

2.2 ONBOARD SUMMARIZATION SYSTEM

If the beacon tone indicates that tracking is required, the onboard summarization system provides concise summaries of all pertinent spacecraft data since the previous contact. The summarization system performs three functions: data collection and processing, mission activity determination, and episode identification. The data collection subroutine receives data from the engineering telemetry system via a function call and applies summary techniques to this data, producing summary measures for downlink to the ground. The mission activity subroutine determines the overall spacecraft mode of operation. This determination is used to choose the appropriate data and limits monitored by the episode subroutine. The mission activity is intended to be exclusive. When a new mission activity starts, the previous mission activity is assumed to have ended. The episode subroutine combines summary and engineering data received internally from the data collection subroutine

with the mission activity received from the activity subroutine and compares the data with mission activity specific alarm limits. It is necessary to use the mission activities to determine which data to use for episode identification and to identify the limits of these data. If the limit is exceeded, the subroutine spawns a new episode and collects past relevant data from the data collection subroutine. The past data collected will be one-minute summaries that go back in time as far as the user has defined. (So a five-minute episode would contain summaries starting five minutes before the episode to five minutes after the episode.) At the end of the episode, the subroutine outputs data to the telemetry subsystem for downlink.

Three different types of summarized data are produced onboard: overall performance summary, user-defined performance summary, and anomaly summary. Six different telemetry packets have been defined to contain this information. (See Figure 2.2.1) Taken as a whole, the telemetry packets produce summary downlinks that are used to enable fast determination of spacecraft state by ground personnel. The performance summaries are generated at regular intervals and stored in memory until the next telemetry ground contact. They are computed by applying standard functions, such as minimum, maximum, mean, first derivative, and second derivative, to the data. User-defined summary data can provide detailed information on a particular subsystem and are created at the user's discretion. Anomaly summary data (episodes) are created when the raw and summarized data violate high or low limits. These limits are determined by the subsystem specialist and stored in a table on-board the spacecraft. The limit tables are based on the current mission activity.

Figure 2.2.1
Summarization Telemetry Packets

Telemetry Name	Description	Output Frequency
Activity	Current value of mission activity	Output on change
Data Sample	Records a snapshot of every raw and summarized data channel	Regular interval, i.e., 15 min.
Episode Summary	Records general data about an out-of-limits data condition called an "episode"	One per episode
Episode Channel	Records specific data about a single data channel's behavior during an episode	One or more per episode
Tone Change	Current state of the beacon tone	Output on tone change
Channel Summary	Summary data about a single data channel's behavior since the last downlink	One for each channel out of limits
User Summary	A user-specified packet containing raw and/or summarized data	Duration user-specified

The software also has the capability to use AI-based envelope functions instead of traditional alarm limits. This

system, called ELMER (Envelope Learning and Monitoring using Error Relaxation), provides a new form of event detection will be evaluated in addition to using the project-specified traditional alarm limits. Envelope functions are essentially adaptive alarm limits learned by training a neural network with nominal engineering data. The neural net can be onboard or on the ground. For DS1, envelope functions are trained on the ground and then uploaded to the spacecraft. DS1 spacecraft fault protection will only be based on project-specified static alarm limits but the summary data can be generated based on the adaptive limits.

3.0 BEACON GROUND VISUALIZATION SOFTWARE (BeaVis)

BeaVis is a ground-based visualization environment for viewing summary data and tone state histories. The tool was designed to facilitate quick interaction with data that has been summarized in a remote system. Summary data files (as downlinked telemetry for space missions) contain all of the important information since the last contact. While it is possible that the summary information is just providing confirming status information, for an adaptive or autonomous system there is likely some urgency in understanding the data because it would not have been sent in the first place if the remote system was functioning normally. For this reason, it was imperative that we design a system that would enable an operator to quickly evaluate summary data to arrive at the correct diagnosis of system behavior. The burden here is shared between the remote system's ability to summarize and the ground system's ability to present the information logically to the user.

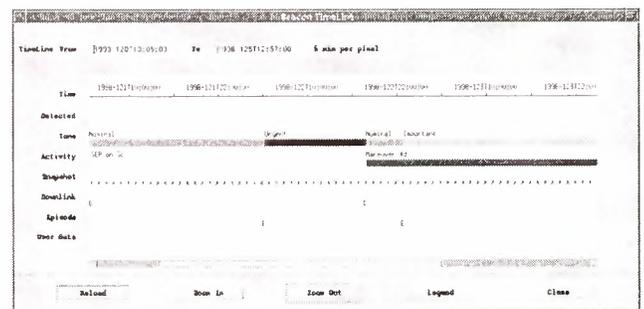


Figure 3.1
BeaVis Timeline Display

The BeaVis delivery for DS1 provides several novel ways of visualizing summary information and includes a timeline display, tabular displays and strip charts. The timeline display, shown in Figure 3.1, provides access to summary downlink data and indicates beacon tone detections that have transpired during the mission. The

tabular and strip chart displays can be accessed via "hypertext" style links from the timeline display. There are GUI elements that show specific summary data components, such as mission activity changes, snapshot telemetry, episode data and user summary data. The environment also includes a tool for creating the parameter tables that are uploaded to the spacecraft.

4.0 EXPERIMENT RESULTS TO DATE

Results so far are showing that the system is performing on par with expectations and functional validation is approximately 90% complete. Validation is defined as functional deployment and means that the system is operational and basically checked out. Full analysis of results is not included as validation by the mission's definition. After the next software upload, currently scheduled to occur in June of 1999, BMOX software will be executed fully. The experiment should reach 100% validation by July 1, 1999. We will continue to conduct the experiment after being fully validated in order to provide data required for performance analysis. Performance evaluations, though still ongoing are yielding some interesting results.

4.1 TONE SYSTEM PERFORMANCE

To date, a total of twelve tone experiments have been conducted. Ten of these experiments were purely tone experiments designed to check out the functionality and characterize the performance of the tone transmission, detection, and delivery systems. These experiments are called Xtone or Ktone experiments for X-band and Ka-band respectively. For these activities, a pre-selected tone (subcarrier) or a sequence of tones was uploaded to the spacecraft prior to the experiment. The tone detection team did not know the tone sequences. During the experiment, the spacecraft commanded the transponder to transmit the tone (or tones) as sequenced. The tone detector at NASA's Goldstone antenna complex, operated remotely from JPL, detected the transmitted tones in near real-time and reported the detection results to various recipients via Email. A total of ten such experiments were scheduled, two of which were not executed because of DS1 schedule changes. Of the eight experiments conducted, six were completely successful and two were deemed partially successful. One of the six successful experiments was a Ktone pass. While data is still being analyzed, preliminary post-pass analysis has validated the correct operation of the tone transmission, detection, and delivery systems. Based on preliminary Xtone detection results, the tone detector appears to have achieved the goal of detecting the signal at 5 dB-Hz. The other two experiments were conducted in conjunction with beacon flight software

experiments, known as Btransmit, where the onboard beacon software analyzes spacecraft engineering data and selects the tone for transmission. Both of these experiments were completely successful.

The signals used for beacon monitor are characterized by three things: (1) the signal strength can be extremely low, (2) the initial tone frequencies, which are derived from an on-board auxiliary oscillator, are not known exactly, and (3) the tone frequencies are constantly drifting. The tone detector is designed to detect these types of signals with a high-level of confidence. Based on data provided by the DS1 telecom personnel, the auxiliary oscillator temperature can undergo a wide range of changes after an OPNAV (optical navigation) maneuver. This results in a very large frequency uncertainty and a very high rate of change (>6 Hz/sec), both of which could exceed the limits of the tone detector (when the signal level is low). A better understanding of the characteristics of the beacon signal in a flight environment (initial frequency uncertainty, drift rate, etc.) and how they affect the performance of the tone detector is being obtained as we complete analysis of the experiment data.

4.2 SUMMARIZATION PERFORMANCE

The end-to-end summarization system is 80% validated, or when weighted against the entire BMOX system, it is 40% out of 50% validated. Functional checkout of the data generation (onboard) and visualization software (ground) has been completed. A detailed performance analysis required to verify that the system is fully operational is ongoing and is 5% out of 15% completed. The tone selection software has been fully checked out.

The initial set of summarization data included 97 'engineering' sensor values sampled onboard once per second. These values were chosen by the Beacon Team based on their importance in detecting major spacecraft anomalies. Five additional sensor values were derived from the original set. Functions including minimum, maximum, mean, first derivative, and second derivative were applied to 16 of the original sensors. High and low limits were applied to 33 of the sensor values.

The data summarization component of beacon has detected several out-of-limit conditions. Several other sensor limit checks gave us false alarms that had to be updated. The ability to rapidly update our alarm limits was part of our validation objectives. We expected false alarms because our limits did not come from the flight team. The first phase of validation used limits from the Beacon Team to test the functionality of the software. The second phase of our validation involves getting limits from the subsystem experts. With more accurate limits, spacecraft engineers

should be able to use summarization data to successfully determine spacecraft anomalies. The data summarization software can provide enough detail for spacecraft engineers using the beacon ground visualization tools to respond appropriately.

One activity that is producing important results involves analyzing summary system performance on DS1 anomalies to date. Although ELMER will not be fully deployed until the next software upload, preliminary results when running ELMER on historical data are showing that adaptive alarm thresholds can track gradual trending of sensor data much tighter than the current DS1 static alarm limits. We see this in monitoring the gradual drift in eight solar array temperatures sensors, one of which is shown in Figure 4.2.1. In comparing traditional limits with ELMER limits during the 81 days of operations, we see that ELMER limits track actual spacecraft performance much more precisely than static limits, which would be off the scale of this chart.

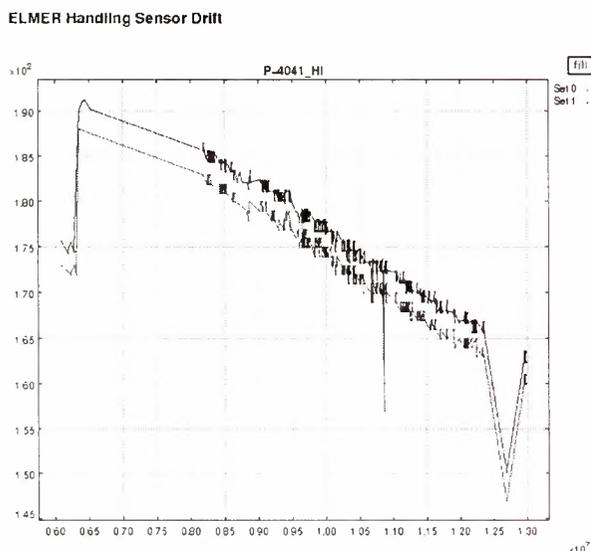


Figure 4.2.1
Tracking of adaptive alarm limit to DS1 solar array temperature

Another validation exercise is confirming that summarization can capture subtle, yet important spacecraft episodes. In ground tests, ELMER detected an unexpected heater turn-on that occurred when the solar panels went off-axis during a spacecraft maneuver. Since ELMER trains across multiple parameters using nominal data, the summarization system detected this event without explicit a priori knowledge of the scenario. This data is shown in Figure 4.2.2.

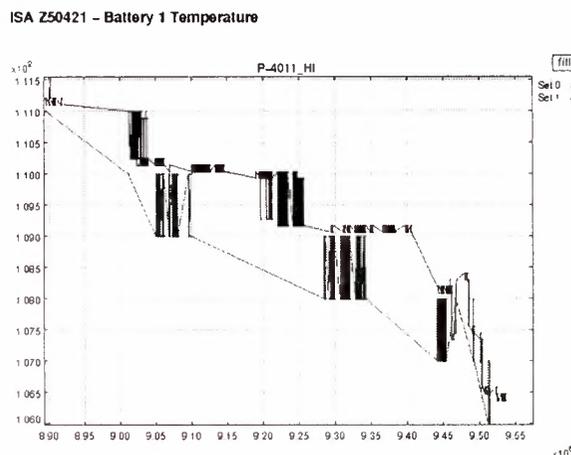


Figure 4.2.2
Battery Temperature Episode Detection

We have already started gathering information from the subsystem engineers for the second phase of the beacon experiment scheduled to begin in June 1999. One of the inputs we have received from the flight team is to trigger limit checks based on a sensor value. For example, start monitoring battery voltage when the battery current is greater than 10A. We will consider adding this capability in the future. The summarization software already has context dependent limit checking, but it uses the overall spacecraft activity, not just the state of one component.

4.3 OPERATIONAL EFFECTIVENESS

The utilization of the ion propulsion system (also called solar-electric propulsion) on DS1 offers an additional justification for baselining beacon operations. The IPS provides continuous thrust for much of the cruise phase. The operational margin for IPS thrusting represents the duration for which IPS could be off and still allow the spacecraft to reach the target asteroid. Due to the low thrust associated with IPS and because actual thrusting did not start until several weeks after launch, the operational margin is only a few weeks. Telemetry downlink passes are becoming less frequent as the DS1 mission progresses. Eventually, there will only be one telemetry pass per week. If the spacecraft experiences a problem that requires the standby mode, the IPS engine will be shut down. It could be up to one week before the flight team has visibility to that standby mode. Using the beacon tone system during the periods between scheduled telemetry downlinks can be a cost effective way to decrease mission risk because it reduces the likelihood of losing thrusting time and not making the intended target. Other future IPS missions have taken note of this fact and are interested in beacon tone services to lower their mission risk.

5.0 LESSONS LEARNED IN DEVELOPMENT

The DSI flight software was redesigned about eighteen months before launch. This decision greatly compacted an already tight development schedule. As a result, the testing of all non-essential software functions was delayed until after launch. The beacon experiment was considered a non-essential piece of software and therefore was only tested pre-launch for non-interference with the other flight software. In post launch testing, routine software problems were found that had to be corrected before turning on beacon software. A nominal amount of system testing prior to launch would have likely prevented these problems from delaying activation of the software and would have decreased development cost.

Before the software redesign, the beacon software was tightly integrated with the DSI fault protection software. The decision was made after the redesign to de-couple the two pieces of software. Previously, the fault protection monitors triggered the beacon tones. After the redesign, the mapping of faults to tones was performed using two different methods. All spacecraft standby modes are now mapped to the urgent beacon tone. The interesting and important beacon tones are mapped using beacon software determined limits. Decoupling the fault protection software from the beacon software gives us maximum flexibility to determine what sensors to monitor. It is unfortunate, however, that our algorithms for determining faults are not nearly as sophisticated as the fault protection monitors. These monitors can look at many different values based on conditional logic before determining what fault has occurred. Complete integration with fault protection would have created a more powerful system.

6.0 FUTURE WORK

There are three facets to future work in this area. One thrust is the operational concept. DSI BMOX is the first major implementation of the components required to achieve adaptive operations on a space mission. Work in this area is expected to continue since beacon operation is baselined for the missions in the JPL Outer Planets Program. Currently, this involves missions to Europa and Pluto scheduled for launch in the next few years. The operational concept can also be extended to implement a science beacon for increased science return and to provide innovative ways to lower the cost of earth-orbiter operations. A second area for future work is in low-cost telecommunications systems for weak signal detection. Today's faster, better, cheaper spacecraft are also often operationally constrained. Providing a weak signal detection service (i.e. a pager service) for these missions is likely to be useful in many ways that we can't even anticipate currently. We say this given the many

suggestions that we have received to date on how such a service could be utilized on DSI and other missions at JPL. The third area for future work is in onboard data summarization. DSI BMOX is the first major effort at JPL to put summarization techniques onboard a spacecraft. This is likely the first step in a roadmap for developing a myriad of summarization technologies to provide a diverse mission set with the tools necessary to capture the most important data for downlink.

7.0 ACKNOWLEDGEMENT

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

8.0 REFERENCES

- Sherwood, R., et. al: 1999, "Lessons Learned During Implementation and Early Operations of the DSI Beacon Monitor Experiment," *Third International Symposium on Reducing the Cost of Ground Systems and Spacecraft Operations*, Tainan, Taiwan.
- Wyatt, E. J., et. al: 1997, "An Overview of the Beacon Monitor Operations Technology," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Tokyo, Japan.
- Wyatt, E. J., et. al: 1998, "Beacon Monitor Operations on the Deep Space One Mission," *Fifth International Symposium on Space Mission Operations and Ground Data Systems*, Tokyo, Japan.
- DeCoste, D.: 1997, "Automated Learning and Monitoring of Limit Functions," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Tokyo, Japan.
- Sherwood, R., et. al: 1997, "Flight Software Implementation of the Beacon Monitor Experiment On the NASA New Millennium Deep Space 1 (DS1) Mission," *Second International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, Oxfordshire, UK.
- Chien, S., et. al: 1997, "Resource Scheduling for a Network of Communications Antennas," *Proceedings of the IEEE Aerospace Conference*, Aspen, CO.
- Staehele, R.L., et. al: 1996, "Pluto Express: Advanced Technologies Enable Lower Cost Missions to the Outer Solar System and Beyond," *International Low Cost Planetary Missions Conference*, Laurel, MD.
- Wyatt, E.J. and J.B. Carraway: 1995, "Beacon Monitoring Approach to Spacecraft Mission Operations," *First International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, Oxfordshire, UK.

NEXT GENERATION REMOTE AGENT PLANNER

Ari K. Jónsson
 RIACS
 NASA Ames Research Center
 Mail Stop 269-2
 Moffett Field, CA 94035, USA
 phone: +1 650 604 2799
 fax: +1 650 604 3594
 jonsson@ptolemy.arc.nasa.gov

Nicola Muscettola
 Recom Technologies
 NASA Ames Research Center
 Mail Stop 269-2
 Moffett Field, CA 94035, USA
 phone: +1 650 604 4744
 fax: +1 650 604 3594
 mus@ptolemy.arc.nasa.gov

Paul H. Morris
 Caelum Research Corporation
 NASA Ames Research Center
 Mail Stop 269-2
 Moffett Field, CA 94035, USA
 phone: +1 650 604 4713
 fax: +1 650 604 3594
 pmorris@ptolemy.arc.nasa.gov

Kanna Rajan
 Caelum Research Corporation
 NASA Ames Research Center
 Mail Stop 269-2
 Moffett Field, CA 94035, USA
 phone: +1 650 604 0573
 fax: +1 650 604 3594
 kanna@ptolemy.arc.nasa.gov

ABSTRACT

In May 1999, as part of a unique technology validation experiment onboard the Deep Space One spacecraft, the Remote Agent became the first complete autonomous spacecraft control architecture to run as flight software onboard an active spacecraft. As one of the three components of the architecture, the Remote Agent Planner had the task of laying out the course of action to be taken, which included activities such as turning, thrusting, data gathering, and communicating.

Building on the successful approach developed for the Remote Agent Planner, the Next Generation Remote Agent Planner is a completely redesigned and reimplemented version of the planner. The new system provides all the key capabilities of the original planner, while adding functionality, improving performance and providing a modular and extendible implementation. The goal of this ongoing project is to develop a system that provides both a basis for future applications and a framework for further research in the area of autonomous planning for spacecraft.

In this article, we present an introductory overview of the Next Generation Remote Agent Planner. We present a new and simplified definition of the planning problem, describe the basics of the planning process, lay out the new system design and examine the functionality of the core reasoning module.

1. INTRODUCTION

The Remote Agent (Muscettola *et al.* 1998) is the first complete autonomous spacecraft control architecture to run as flight software onboard an active spacecraft. In a unique experiment in May of 1999, the Remote Agent was flight-validated onboard the Deep Space One spacecraft. During this experiment, the Remote Agent successfully generated complex plans which included thrusting of the Ion Propulsion System, slewing and taking pictures. The Remote Agent executed the generated plans safely, and correctly handled a number of injected faults during execution.

As discovered during the development of the Remote Agent Planner, the spacecraft domain provides a number of challenges that are typically not addressed in autonomous planning technology development:

- Activities are executed concurrently onboard the spacecraft, so a plan consists of concurrent activity sequences that can safely be executed in parallel.
- Resources, such as power, fuel, data storage, are strictly limited. A planner must guarantee that possibly concurrent activities in a plan will not exceed resource availability.
- Activities have complex interactions and constraints between them, and any plan generated by the planner must satisfy all constraints and take all interactions into account.

- Activity duration is often flexible. A planner must therefore be capable of reasoning about activities that only have bounds on their duration.

To meet these challenges, the Remote Agent Planner was based on an approach to planning that departs from the more classical planning approaches (Bylander 1994) in a number of ways. (1) The planner reasons about parallel activity sequences, each of which represents the changing state of some system attribute. (2) It can reason about activities that have flexible duration, while taking into account quantitative temporal constraints between them. (3) The goal of the planner is not to generate a fixed sequence, but rather to generate a plan description that is suitable for execution. (4) The planner handles a rich action representation language that can describe the complex activities of real-world systems. This language is also unique in that it eliminates the syntactic and semantic distinction between actions and steady-states. (5) The planner allows for a structured domain description language that is sufficiently expressive to describe the rules and interactions in complex real-world domains such as spacecraft.

The applicability of this approach to real-world planning problems was clearly demonstrated in the Remote Agent Experiment. Nonetheless, work continues on the development of the approach, both in terms of the underlying planning framework and in terms of the implemented planning system. The Next Generation Remote Agent Planner is the next step in this development, providing a simpler and clearer definition for the planning framework, and an enhanced, modular implementation of the planning system.

The simplified planning framework is derived directly from the framework underlying the original Remote Agent Planner. It is just as expressive as the original framework, but has been simplified by unifying concepts and simplifying the problem specifications. The implemented planner is also based on the original planner, but a number of interesting enhancements have been made. First of all, it is based on a new modular system design, aimed at making it easy to modify, maintain and enhance the different components that make up the system. Secondly, the interface that the core system provides to the top-level planner search engine has been significantly simplified. Whereas the original system was limited to backtracking search, the new framework, in conjunction with the simplified top-level interface, make it possible to utilize other, possibly more efficient, search techniques, such as repair-based search and dependency-directed search. Third, the new system includes a new constraint reasoning module that allows arbitrary procedural constraints to be used. This speeds up the constraint reasoning, which is a crucial part of

the planning process, and eliminates previous limitations on the set of constraints that can be represented.

In this paper, we describe the simplified planning framework, and give an overview of the new implemented planning system. We first present the planning framework in an informal manner. We then describe the approach used to solve the planning problems, and give an overview of the new planning system. We continue by providing some details about the new constraint reasoning mechanism, and conclude by looking at what has been done and what is on the agenda.

2. THE PLANNING FRAMEWORK

In this section, we will describe the simplified planning framework, on which the Next Generation Remote Agent Planner is built. The planning framework defines the class of planning problems being solved, i.e. what the world looks like to the planner, and what constitutes a valid plan.

Let us start by looking at what the end result of the planning process should be, i.e. what constitutes a plan. Considering the planner as part of the Remote Agent system, a completed plan is a program or a recipe for what activities the Remote Agent Executive should perform and what states should be maintained. In classical flight software systems, such a plan consists of time-stamped tasks, each to be executed at the predetermined time. The problem with that approach is that it requires an explicit tradeoff to be made between robustness and efficiency. If the time allocated to a task is close to the estimated execution, any delay will result in failure. However, if the time allocated to the same task is much more than the estimated execution time, then time is wasted. To resolve this problem, the Remote Agent is capable of handling temporal flexibility in timepoints describing transitions such as going from the engine thrusting to the engine being off. This means that the start of one task can be tied to the completion of another task, minimizing the effect of any delays, while maintaining the robustness of the plan. The end result of this is that the generated plan is defined, not by fixed times for transition timepoints, but rather by bounds on those timepoints and temporal constraints between them. Figure 1 shows what a simplified, small plan might look like.

In order to define the planning framework, we must now specify what "activities" are, what temporal constraints are and what constitutes a valid plan. Since the exact set of activities and rules will depend on the environment in which the planner operates, the planner uses a description of the activities and rules in each environment. Such a description is called a *domain model*, as

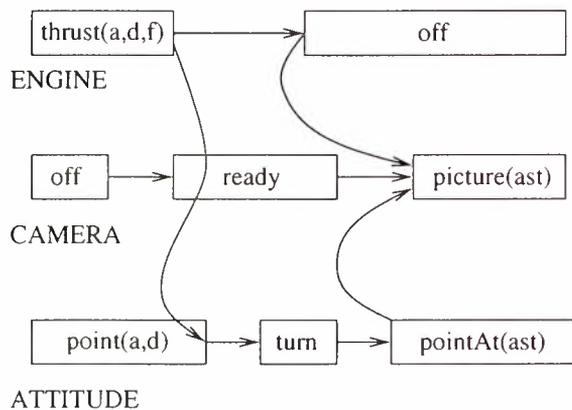


Figure 1: A simplified plan showing activities for engine, camera and attitude. Arrows show temporal constraints between transition timepoints.

it models the domain in which the planner is operating. Describing the planning framework is therefore largely a question of defining what a domain model is.

Many real-world systems, including spacecraft, can naturally be described in terms of components that at each point in time are in a certain state or performing a certain activity. For example, at any point in time, the attitude system can either be holding a specific attitude, or turning from one attitude to another. This natural approach to modeling real-world systems is mirrored in the planner, which plans by reasoning about how the states of such components can change over a given period of time. To generalize this, the basic concept in the domain model is an *attribute* which describes a part of the world that can change over time, e.g. the state of a spacecraft system component.

To specify an attribute, the set of possible values (representing states or activities) must be given. Since states and activities are often fairly complex, the attribute values are described in terms of *predicates* that can have multiple parameters. For example, the attribute value describing the state of holding a constant attitude must have the pointing coordinates as parameters, resulting in a predicate of the form `constantPointing(a, d)`, assuming equatorial coordinates.¹

A predicate is defined by a unique *predicate name*, a sequence of *parameter domains* and optionally a set of *parameter constraints*, which limit the set of valid parameter value combinations. For an example of a predicate, let us consider an attribute describing the amount

¹Technically, these are not predicates, as they do not evaluate to true or false by themselves. However, they can be viewed as shortcuts for the predicates representing that a given attribute has that particular compound value.

of data stored on the onboard data recording mechanism. A predicate describing data being recorded, aptly named `record`, has four parameters; the amount of data at the beginning of the activity, the rate at which the data is being collected, the duration of the activity, and the amount of data at the end. Each parameter takes a value from a given domain; for example, the start-data and the end-data parameters have values between 0 and M , where M is the maximum data storage capacity. Obviously, not all combinations of the possible parameter values give rise to a valid record activity description. Therefore, the final component of the predicate definition is the constraint that for any instantiation `record(s, r, d, e)`, the parameters must satisfy $s + rd = e$.

To structure the domain model, attributes are arranged together as components of model objects, which in turn are instances of model classes. This means that a *model class* is essentially a set of named attributes. For example, a class describing engine objects might have a fuel level attribute, an engine state attribute and a thrust attribute. The *model objects*, such as a specific engine, are then instances of these classes. This allows the same class definition to be used for multiple instances, e.g. in a spacecraft with multiple engines.

Having seen how the predicates describe the values that each attribute can take, let us now turn our attention to the interactions between different attributes. This interaction is the main complicating factor in real-world systems, as many configurations and sequences are either not possible or not safe. For an example of such interactions, let us consider a spacecraft that has an engine and a camera. Since the engine thrust causes vibrations, the camera cannot be taking pictures during the times the engine is thrusting. This leads to the constraint that whenever the camera is taking pictures, the engine must be off. Rephrasing this slightly, the constraint states that any continuous temporal interval where the camera is taking a picture must be contained within a continuous interval where the engine is off.

In order to be able to describe this containment and other relations between intervals, the planner uses quantitative temporal relations. There are twelve possible relations that come in pairs where one is the inverse of the other. The six temporal relations classes are:

- before, after
- startsBefore, startsAfter
- endsBefore, endsAfter
- startsBeforeEnd, endsAfterStart
- contains, containedBy
- parallels, paralleledBy

Quantitative bounds can be placed on the distance be-

tween any two timepoints involved in the interval relation. For example, “before[10,20]” indicates that the first interval must end at least 10 and no more than 20 time units before the second one starts.

To specify rules, such as the one involving the engine and camera, we use a construct called a *configuration constraint*. In principle, a configuration constraint is defined for each possible instantiation of a predicate. Thus, each configuration constraint consists of a predicate instance (attribute value) v and a set of pairs $\{(\tau_1, V_1), \dots, (\tau_k, V_k)\}$, where τ_i is a temporal relation and V_i is a set of instantiations of a predicate. The semantics of such a constraint are that for any interval I where an attribute has the value v , there must, for each $i \in \{1, \dots, k\}$, be an interval J_i where an attribute has one of the values in V_i and the interval pair (I, J_i) satisfies the temporal constraint τ_i .

For an example of such a configuration constraint, let us write up the one for a camera taking a picture of a specific asteroid. In textual form, the configuration constraint can be specified as follows:

```
(camera == picture(asteroid))
  containedBy(engine == off)
  containedBy(attitude == pointAt(asteroid))
  before[0,0](camera == ready)
  after[0,0](camera == ready)
```

The “containedBy” relations specify that each of the engine-off and point-at-asteroid intervals must start no later than at the start of the picture-taking interval and end no earlier than when the picture-taking interval ends.² The “before[0,0]” and “after[0,0]” relations enforce that camera-ready intervals must immediately precede and follow the picture-taking interval. Figure 2 shows a graphical representation of this configuration constraint.

It should be noted that although configuration constraints are conceptually defined for each predicate instantiation, in practice, they are specified in the form of *configuration constraint schemata*. Such schemata specify patterns rather than instantiated attribute values, thus collapsing large sets of constraints into a single schema. The constraints are then instantiated from the schemata whenever sufficient information is available to determine that they are applicable to a given interval.

3. THE PLANNING PROCESS

The Next Generation Remote Agent planning process is based on representing and reasoning about the

²Not displaying the bounds is short-hand for the distance bounds being $[0, \infty]$.

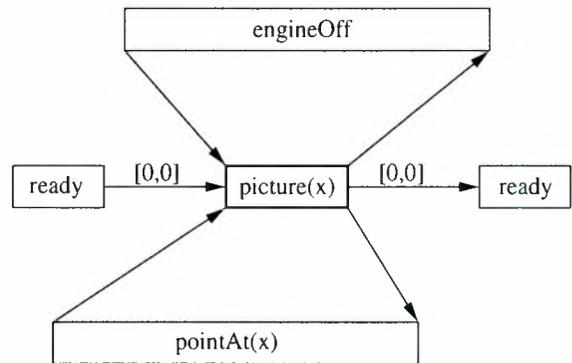


Figure 2: A graphical representation of a configuration constraint. The links indicate temporal constraints that limit the distance from one timepoint to another.

possible developments of each attribute over the time period for which the planner is planning. The goal of this reasoning process is to generate a plan consisting of a network of transitions between attribute values, such that all configuration constraints are satisfied.

The approach used by the planner is to generate and reason about structures called *tokens*. Each token represents a restriction on the set of values that an attribute may take over a specified temporal interval. A *value token* is a special type of token, having the additional restriction that the attribute must maintain a single value throughout the associated interval. Other types of tokens are used in the RA planner, such as constraint tokens which limit the attribute value to a given set, but do allow the attribute value to change during the interval. However, for clarity we will only consider value tokens in this paper. From here on, any reference to a token should therefore be read as referring to a value token.

The planner utilizes variables to represent the different elements of a token. This allows the planner to reason effectively about tokens and their interactions. As a result, a token consists of:

- A predicate name
- A variable representing the start time
- A variable representing the end time
- A variable representing the duration
- A set of parameter variables, one for each parameter to the predicate

In addition to the variables, any applicable parameter constraints are associated with a token, and so is a temporal constraint enforcing that the sum of the start time and the duration is equal to the end time.

Other temporal constraints may then link start and end timepoints from different tokens. These can stem

from configuration constraints, or be instantiated as part of the planning process. Taken all together, the variables and the constraints, both temporal and parameter, form a network of variables linked by constraints, i.e., a *constraint network*. The constraint network is a dynamic entity, as variables and constraints can be added and removed throughout the planning process. The constraint network plays an important role in this approach to planning, since any plan which gives rise to an inconsistent constraint network cannot possibly be extended to a valid plan.

The planner uses *timelines* to represent and reason about the set of possible developments for attributes. For each attribute of each domain object, the planner has exactly one timeline. The reason for utilizing such a specialized construct is that there is a strong relation between tokens that apply to the same attribute of the same object, i.e., the same timeline. Consider any two tokens for the same timeline, each describing a set of valid attribute values for a temporal interval. If the sets of attribute values do not overlap, then the two tokens cannot overlap in time, i.e., one must come before the other. Conversely, if any two tokens necessarily overlap, then they must describe the same interval having the same attribute value. Conceptually, a timeline consists of a sequence of timepoints, each representing a possible transition from one attribute value to another, i.e., the start or end of a token. The interval between any two adjacent timepoints is called a *slot*. During the planning process, a slot will either contain one or more codesignated tokens, or it will be empty.

A set of tokens, along with the associated parameter variable domains, temporal constraints and timelines, describes a *partial plan*. The goal of the planning process is to modify this partial plan, until it is a complete and valid plan. The key observation behind this process is that for any given partial plan, there are only four requirements that can prevent a partial plan from being complete and valid:

1. Parameter variables must be assigned values
2. Tokens must be scheduled onto timelines
3. Configuration constraints must be satisfied
4. Underlying constraint network must be consistent

Any violations of the first requirement can be addressed by selecting a value to assign to each unassigned parameter variable. The second requirement can be enforced by selecting a suitable (not necessarily empty) slot for each uninserted token, and insert the token there. Depending on whether the slot is empty or not, the token will be scheduled between two other tokens or codesignated with a previously scheduled token. The third requirement can be satisfied without any

selection criterion. The simplest approach is to instantiate any tokens required to satisfy a configuration constraint, as soon as a token is inserted on a timeline and all parameter domains have been grounded. If the token is later removed from a timeline or the parameter domains are relaxed, then the instantiated tokens are also removed. Finally, if the constraint network is found to be inconsistent, one or more constraints and value assignments can be removed.

Needless to say, the above methods for enforcing the four requirements interact with one another, one fix causing another break. The process of navigating through these operations is called search, and it can be a complex and expensive process. However, in this framework, there are only three relatively simple operations that require decisions to be made, namely:

- Insert a token on a timeline
- Remove token from timeline
- Modify domain of variable, which includes assigning single values

Although having a simple set of operations does not by itself reduce the cost of searching, it does provide a great deal of flexibility in how the search is done. However, the resulting flexibility may lead to significant reductions in search costs, as more effective search techniques can be brought to bear.

4. THE SYSTEM MODULES

One of the key goals of this work is to design and implement a flexible, extendible and portable planning system that can serve as a research framework for further development of autonomous planning and reasoning techniques, while also providing the core for future applications of the Remote Agent Planning technology. The new system is written in C++, to provide structured programming, fast execution and portability. As of May 1999, the redesign is complete, the implementation is almost complete and testing is under way.

The new implementation is based on a careful object-oriented modular design, which allows modules to be easily replaced, improved and tested. Figure 3 shows an overview of the main modules and the relations between them.

The *constraint network manager* is the constraint reasoning module, responsible for handling the dynamic constraint network described above. The main responsibilities are:

- Add and remove variables.
- Add and remove constraints.
- Manage and reason about variable domains.

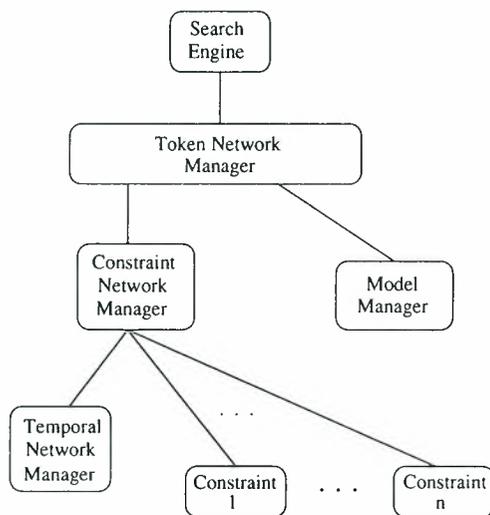


Figure 3: An overview of the key modules in the Next Generation Remote Agent Planner.

- Inform about local and global consistency.
- Provide heuristics for variables and values.

The constraint network manager utilizes the temporal network manager for handling the temporal variables and the temporal constraints that connect them. This is done to allow more efficient algorithms to be applied to the computationally simpler temporal network (Dechter *et al.* 1991). The constraint network also uses external constraint procedure for representing constraints. Such external procedures can represent any constraint, ranging from simple arithmetic equalities to the complicated feasibility evaluations. The design and capabilities of our constraint reasoning framework are discussed further in the next section.

The *model manager* handles all the information relating to the domain model. As a result, it serves a dual role; as the input module responsible for setting up the domain model, and as an information module responsible for providing information about the domain model.

To facilitate the model manager's role as an input module, it has a well-defined input interface that can serve as the single interface for the various different ways in which a model can be specified. As a result, it can be connected to a parser for reading domain descriptions from input files, just as well as it can be connected to a graphical user-interface for building models interactively.

In its role as an information module, the model manager is responsible for effectively responding to queries about the domain model. This includes providing information about the hierarchy of domain classes, the

attribute definitions and the predicate definitions. However, most of the work done by the model manager is in providing information about configuration constraints. As the constraints are described by configuration constraint schemata, the model manager can map any given set of attribute values into the applicable configuration constraint instantiation. Furthermore, to facilitate incremental reasoning, it can also determine what changes occur in the applicable configuration constraints, given any two sets of attribute values.

The *token network manager* handles the top-level planning operations, thus providing the interface that the search engine will use. Its main responsibilities are the following:

- Initialize timelines and tokens according to the domain model and the set of goals to be achieved.
- Add/remove temporal constraints between token timepoints.
- Insert and remove tokens from timelines. This includes inserting into empty slots and codesignating with existing tokens.
- Provide access to parameter variables in tokens so that their domains can be modified and assigned values.
- Automatically generate and eliminate tokens in response to applicable and instantiated configuration constraints.
- Determine consistency and validity for the current partial plan.

Finally, on top of the token network manager, there is a *search engine* that controls the planning process. As mentioned above, only a small set of operations is required to modify the partial plan during the planning process. The role of the search engine is to control the application of these operations, with the goal of finding a valid and complete plan.

Recall that the only required operations were the ability to modify a parameter variable domain and the ability to insert and remove tokens from timelines. Any of these operations can be undone by performing another operation from the set. For example, assigning a single value to a variable can be undone by modifying the variable domain to have the set of values it had before. More importantly, the semantics of the operations guarantee that the effect of undoing an operation is the same as not performing the original operation. This holds regardless of what has been done in between, which is exactly what allows us to utilize non-chronological methods in the search engine.

The added flexibility available to the search engine opens a number of possibilities in making the planning process more efficient. In other domains, various search engines have proven to be effective at solving decision problems such as planning, even in real-world domains. Among the many candidate search techniques that may prove applicable to this planning framework are dependency-directed search (Stallman & Sussman 1977), limited discrepancy search (Harvey 1995), relevance-bounded search (Bayardo Jr. & Miranker 1996), iterative sampling (Langley 1992), heuristic-biased sampling (Bresina 1996) and repair-based search (Minton *et al.* 1990).

5. THE CONSTRAINT REASONING SYSTEM

The Next Generation Remote Agent Planner is based on a redesign of the existing RA planner and thus inherits a number of existing solutions and algorithms. However, a completely new framework has been developed and implemented for doing the constraint reasoning. The new constraint reasoning framework is very general, as it can reason about any set of variables and constraints. At the same time, it is also quite efficient as it combines efficient internal reasoning methods with fast external special-purpose procedural methods.

A constraint network consists of a set of variables, each taking values from a given domain, and a set of constraints connecting the variables. Formally, a constraint is a relation that specifies which combinations of values are allowed for the set of variables in the constraint's scope. However, this is not how constraints are specified in practice, as listing the allowed combinations requires excessive amounts of space. As a result of this, constraints are typically specified using special-purpose constraint descriptions that the constraint reasoning system can understand. In this system, for example, temporal constraints are specified by noting the two variables and the bounds on the distance from one to the other. The problem with this approach is that although it is very efficient and easy to use, it limits the set of constraints to those specifiable in this description language. To solve this problem, without incurring significant efficiency penalties, the Remote Agent constraint network manager can handle external constraint procedures.

A *constraint procedure* is a program that is applied to a set of variables, the scope of the constraint. The procedure implements a mapping that maps each variable domain to a subset (although not necessarily a strict subset) of that domain. In other words, the procedure reduces the set of possible value assignments for the variables, by eliminating values from the domains.

To see how this defines a constraint, let us consider applying the procedure to a set of domains where each domain has only one value, i.e, a variable assignment. The procedure can then either map the set of domains to itself (indicating that this is a valid assignment to the variables) or reduce one or more domains to the empty set (indicating that the given assignment is invalid). A procedure therefore implicitly defines a set of allowed value assignments for the variables in the scope; in other words, it defines a constraint. The only restriction placed on a constraint procedure, in order to make it useful for constraint reasoning, is that it never eliminate any allowed assignments when reducing the domain sets.

The key reasoning task in a dynamic constraint reasoning system is to try to prove the network consistent or inconsistent. This is done by applying a technique called *propagation*, where information about possible and impossible solutions is propagated between variables, through the constraints. In general, correctly determining consistency is NP-complete and will therefore have a worst-case complexity that is exponential in the number of variables. As a result of this, dynamic constraint reasoning is typically done with limited propagation techniques like maintaining arc-consistency.

In its simplest form, *arc-consistency* guarantees that for each value in the domain of a given variable, any single other variable can be assigned some value from its domain, without directly violating a single constraint. Maintaining arc-consistency is therefore the process of eliminating any values that do not satisfy the above condition. This can be accomplished with algorithms that have low-order polynomial complexity. The tradeoff is that inconsistencies may remain undetected, as there is no guarantee that three or more variables can be assigned values without violating a constraint. However, the fact that inconsistencies may remain undetected is not a problem in this planning framework. The reason is that any uninstantiated variables are eventually assigned single values, and in that situation arc-consistency is sufficient to determine the overall consistency correctly.

As in most other dynamic constraint reasoning systems, a propagation algorithm is the core of the constraint network manager. The algorithm we have developed is based on maintaining arc-consistency, but it has been extended so that it can take advantage of other methods that also eliminate values from variable domains. The advantages of this extension are twofold. First, it allows the propagation to directly take advantage of the procedural constraints, which can often eliminate values faster and more effectively than the arc-consistency maintenance. Secondly, the prop-

agation method can be augmented with other efficient propagation algorithms such as the one that performs the propagation within the temporal subnetwork.

The result of all this is not only an efficient framework for performing constraint reasoning, but one that can easily be extended. Constraint procedures can be written separately and simply added to the system, without any modification to the constraint reasoning mechanism. In addition to that, specialized techniques for handling certain parts of the network, e.g. the temporal subnetwork, can be added into the constraint network manager with minimal changes.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an overview of the Next Generation Remote Agent Planner, the next step in the continuing evolution of the RA Planner. The new planning system, with the simplified framework and a modular and flexible design, provides a solid foundation for future applications in autonomous planning for spacecraft, and a framework for further research into the many aspects of autonomous planning for real-world systems.

The development of the Remote Agent planning system is ongoing work, as new challenges arise and better reasoning techniques are developed. This gives us both clear near-term goals and a number of interesting research venues for future work. As of May 1999, the planning framework definition and the modular system design have been completed. The system implementation is close to completion and testing is already underway. Aside from concluding the main system tests, the near-term goals include the development and study of different search engines for driving the planning process. For the longer-term goals, there are too many interesting research questions and application opportunities to list them fully in this paper. However, regardless of which goals are pursued, this new system will provide a solid foundation for both further research into autonomous planning techniques and future applications of the Remote Agent Planner.

REFERENCES

- Bayardo Jr., R. J. & D. P. Miranker (1996). A complexity analysis of space-bounded learning algorithms for the constraint satisfaction problem. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 298–304.
- Bresina, J. (1996). Heuristic-biased stochastic search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.
- Bylander, T. (1994). The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:165–204.
- Dechter, R., I. Meiri, & J. Pearl (1991). Temporal constraint networks. *Artificial Intelligence*, 49:61–95.
- Harvey, W. D. (1995). *Nonsystematic Backtracking Search*. PhD thesis, Stanford University, Stanford, CA.
- Langley, P. (1992). Systematic and nonsystematic search strategies. In *Artificial Intelligence Planning Systems: Proceedings of the First International Conference*, pages 145–52. Morgan Kaufmann.
- Minton, S., M. D. Johnston, A. B. Philips, & P. Laird (1990). Solving large-scale constraint satisfaction and scheduling problems using a heuristic repair method. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 17–24.
- Muscettola, N., P. P. Nayak, B. Pell, & B. William (1998). Remote agent: To boldly go where no ai system has gone before. *ai*, 103(1-2):5–48.
- Stallman, R. M. & G. J. Sussman (1977). Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, 9:135–96.

Demonstrations

MICRO PLANETARY ROVER "MICRO5"

Takashi Kubota*, Youji Kuroda**,
Yasuharu Kunii***, and Ichiro Natakani*

*The Institute of Space and Astronautical Science,
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, JAPAN.
TEL: +81-42-759-8304 FAX: +81-42-759-8304
E-Mail: kubota@cnenet.isas.ac.jp

**Dept. of Mechanical Engineering, Meiji University,
1-1-1, Higashi-mita, Tama-ku, Kawasaki -shi, Kanagawa 214-8571, JAPAN

***Dept. of Electrical Engineering, Chuo University,
1-13-27, Kasuga, Bunkyo-ku, Tokyo 112-8551, JAPAN

Abstract

This paper describes a newly developed rover with small size, light-weight, low power consumption. In recent years, many researchers have extensively studied and developed unmanned mobile robots for surface exploration of the moon or planets. A lunar or planetary rover is required to travel safely over a long distance for many days in unfamiliar terrain. This paper presents scientific signification, requirements, and technology of a lunar or planetary rover. This paper proposes a new mobility system, which has four wheels and one supported wheel. This novel suspension system is a simple and light mechanism like a four-wheeled rover and provides a high degree of mobility like a six-wheeled rover. The performance of the developed rover is shown by some experiments.

1. Introduction

Toward the turn of the century, several schemes sending an unmanned mobile explorer to the moon or Mars are being planned for scientific exploration. In recent years, many researchers have studied and developed lunar or planetary rovers for unmanned surface exploration of planets[1][2][3]. Especially micro-rover missions have received a lot of attention, because small, low-cost missions are typically constrained by mass, budget and schedule. In July 1997, NASA/JPL succeeded in Mars Pathfinder mission and the Sojourner rover could move on the Martian surface and gather and transmit voluminous data back to the Earth[4].

NASA plans to send some rovers to Mars in 2001, 2003, 2005 Missions[5]. NASA/JPL has developed a small rover prototype, called Rocky 7. This microrover is capable of long traverses, autonomous navigation and science instrument control. This rover carries three science instruments and can be commanded from any computer platform from any location using the World Wide Web.

As a part of a development program, teleoperation or autonomous navigation technologies are earnestly studied for realizing a rover to be able to move on an unknown lunar or planetary surface[6]. In recent years, many researchers have earnestly studied and developed planetary rovers for unmanned surface exploration of planets[7][8]. However, there are few navigation systems that can travel safely over a long distance for many days in unknown terrain. There have also been proposed only few practical path planning methods based on sensory data[9][10][11][12].

Recently rover field tests have been performed for evaluating the planetary rover performance. In December 1996, NASA/JPL demonstrated the field tests by the Rocky 7 in the Mojave Desert[13]. The Rocky 7 navigation is based on operator way point designation and on autonomous behavior navigation for movement to the specified targets. In June 1997, CMU rover Nomad navigated 200[km] of the planetary-like Atacama Desert in South America while under the control of operators in North America[14]. The authors also have done a long range test for the perfect autonomous rover at a slag heap in Izu-Ohshima in Japan[15].

The authors have studied a lunar or planetary rover which can travel safely over a long distance on rough terrain. The authors have developed a small, light-weight microrover with a new mobility system, which is called "Micro5". This paper describes the design and implementation of a small rover for future Lunar or Planetary missions requiring long traverses and rover-based science experiments. This paper provides a system overview of a newly developed microrover Micro5.

This paper is structured as follows. In Section 2, the rover mission is discussed. In Section 3, the prototype Micro5 developed for lunar or planetary exploration is explained. Then a new mobility system is proposed in Section 4. Section 5 discusses some experiments and demonstrations for Micro5. Cooperative exploration mission based on buddy system is proposed in Section 6. Finally, Section 7 is for conclusion.

2. Rover Mission

With a new type of launch vehicle, M-V rocket, our Institute (ISAS) has a capability of lunar or planetary exploration. ISAS launched "Nozomi" spacecraft which will be a Mars orbiter. ISAS plans to send Lunar-A spacecraft with penetrators to the moon. ISAS is also promoting SELENE mission with NASDA, which includes a lunar orbiter and lander. The authors have been conducting a wide variety of researches on the rover for the future missions [16][17]. Figure 1 shows an example of the concept of rover missions, where some rovers explore the moon or planets cooperatively.

2.1 Science missions

Candidates for our missions here, not all of which, though, will be accommodated by our rovers, are as follows :

1. Geology by photo images : topographical survey, identifying size, and shape of rocks, composition of rocks, craters etc.
2. Element Analysis : analysis of age using mass-spectrometer, element analysis using X-ray spectrometer, or γ -ray spectrometer, study of mineral composition using visible or infrared reflection spectrometer etc.
3. Wide Area Investigation : studies on magnetic anomalies using magnetometer, gravity anomalies, electro-magnetic structure of the crust using VLF, seismo-logical observation using seismo-meter network etc.

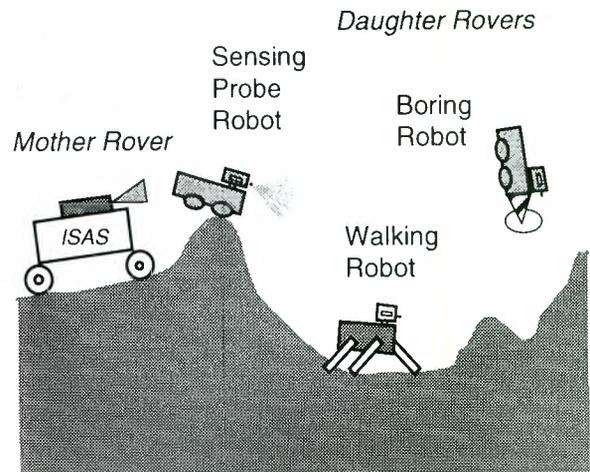


Fig.1 Concept of Rover Mission

4. Investigation by Manipulator : analysis of regolith, measurement of heat flux, element analysis etc.

2.2 Rover requirements

Lunar or planetary rovers are expected to travel in wide areas and explore the surface in detail. Exploration requirements for lunar or planetary rovers are as follows :

1. Large area exploration
2. Underground exploration
3. Long term exploration
4. Sample collection and analysis
5. Placement of scientific instruments
6. Exposed surface exploration such as craters

2.3 Engineering missions

The engineering objective of our rover here is to establish various engineering techniques for the future deep space missions such as :

1. Autonomous soft landing technique
2. Adaptation for planetary environment
3. Reliable mobility development
4. Navigation and guidance
5. Tele-science technology
6. Small, light, Low-power instruments
7. Mission operation technology

3. MICRO5

The authors have developed a small rover Micro5 for future Lunar or Planetary exploration missions requiring long traverses and rover-based science experiments. The overview and specification of the developed Micro5 are shown in Fig.2 and Table 1 respectively. The weight of Micro5 is about 5[kg]. The developed rover measures about 0.53[m] wide, 0.55[m] long, and 0.25 [m] high. The wheel diameter is 0.1[m].

The developed rover is driven by five wheels controlled independently. The steering is controlled by differential of left and right wheels. Those wheels are actuated by small DC motors. The velocity of the rover is about 1.5[cm/s]. This rover has the proposed new suspension system. So the climbable step is 0.13[m] and the climbable slope is about 40[deg]. Power is supplied by solar panel on the top of the rover. The rover is also driven by on-board batteries.

Two stereo cameras are used for a forward terrain sensor. This rover also has some other CMOS cameras around the body for navigation and scientific observation. The rover is equipped with pitch and roll clinometers for attitude detection and encoders for dead-reckoning. Sensor data processing and control are performed by on-board computers. The RISC-CPU's are dedicated to the function of environment recognition, path planning and navigation.

The developed Micro5 has communication system to communicate with the ground system. The rover can send obtained images, house-keeping data, scientific data to the ground system. Operators can control the robot based on image data by teleoperation techniques. Micro5 has the sampling system. The light-weight manipulator with a CMOS camera has been developed, which will be attached to the front of the rover. Some scientific instruments are under development.

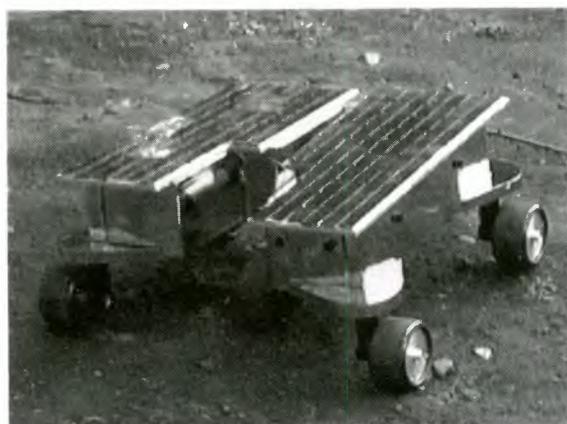


Fig.2 Overview of Micro5

Table 1 Specification of Micro5

Size	0.55[m] (Length) 0.53[m] (Width) 0.25[m] (Height)
Weight	about 5[kg]
Mobility System	Micro5 suspension system Wheel Diameter : 0.10[m] Body Height : 0.13[m]
Mobility Performance	Velocity : 1.5[cm/s] Climable Step : 13[cm] Climable Slope : 40[deg]
CPU	Muultiple CPU's (32bit, 16bit)
Communication	40[Kbps] UHF
Power Supply	Solar Panel : peak power 27[W] Battery : NiCd
Power Consumption	Actuator: less than 3[W] : less than 5[W] (MAX) Computer : less than 4[W]
Payload	8 cameras Sunsensor etc. (to be equipped) Manipulator (to be equipped)

4. New Mobility System

Various kinds of the mobility systems for traverse on rough terrain have been proposed. The suspension system is the key issue for realizing high degrees of mobility. NASA/JPL developed rocker-bogie suspension in a series of the project called "Rocky". That system consists of a pair of two links called the rocker and the bogie which are attached to each other by a passive rotary joint. This combination of the rocker and the bogie makes it possible for the rover to climb rocks 1.5 times its wheel diameter in height smoothly. The rocker-bogie suspension system provides extremely high degree of mobility for the rover. However this is not a perfect system for smaller rover. The rocker-bogie system of Rocky 7 has six wheels. Many-wheels system needs many motors and gears, that causes to increase the weight. Another problem comes from the structure that wheels are attached on the end of the long links and the links are connected by rotary joints as a chain. So very strong stress would act on the links and the joints, even if small force is acted on the wheels. The structure has to be made heavier to endure the strong stress.

A small long-range rover is required to have both a simple and light weight mechanism like 4-wheel drive system and a high degree of mobility like rocker-bogie suspension system. In order to achieve these opposed requirements, the authors propose a new suspension system[18] as shown in Fig.3. The proposed suspension system PEGASUS consists of a conservative four-wheel drive system and a fifth active wheel connected by a link. The fifth wheel which is attached to the end of the link, and the other end of the link is attached to the body with a passive rotary joint. The proposed system is designed to distribute the load of weight equally to all five wheels whenever the rover climb up or down. It means that the fifth wheel supports the load taken to the front wheels when the front wheels climb up rocks, and it also supports that taken to the rear wheels when the rear wheels climb up the rocks. This system can be realized to be simple and light in weight, because the design is based upon a simple 4-wheel drive system.

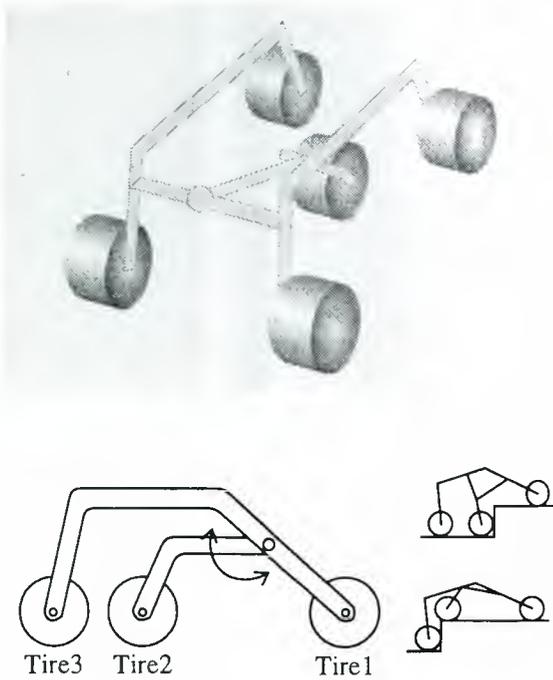


Fig.3 PEGASUS system

5. Demonstrations

Micro5 navigation strategy is based on teleoperation and autonomous behavior. The performance on the mobility of Micro5 is demonstrated by teleoperation.

Figure 4 shows the image sequence of outdoor experiments. The experimental result shows the good performance of the developed microrover.

(a) Experimental Image #1



(b) Experimental Image #2



(c) Experimental Image #3



(d) Experimental Image #4



(e) Experimental Image #5



(f) Experimental Image #6



Fig.4 Field Experimental Results

6. Buddy System

The authors are going to design and develop a lunar or planetary rover based on Micro5 architecture according to mission requirements. The authors are also proposing a multiple rovers mission based on buddy system as shown in Fig.5. The proposed buddy system would lead to higher reliability and safety for exploration mission of the moon or planets. The multiple rovers can also make it possible to extend the exploration areas. Various kinds of tasks such as digging, crater exploration, cliff exploration, sample collection can be realized by cooperation of multiple rovers.

7. Conclusions

This paper described a developed microrover "Micro5" for future Lunar or Planetary missions requiring long traverses and rover-based science experiments. This paper also proposed a new design concept on the small light-weight rover with a novel mobility. Some experiments and demonstrations showed the good performance of Micro5. Cooperative exploration by buddy system is also proposed.

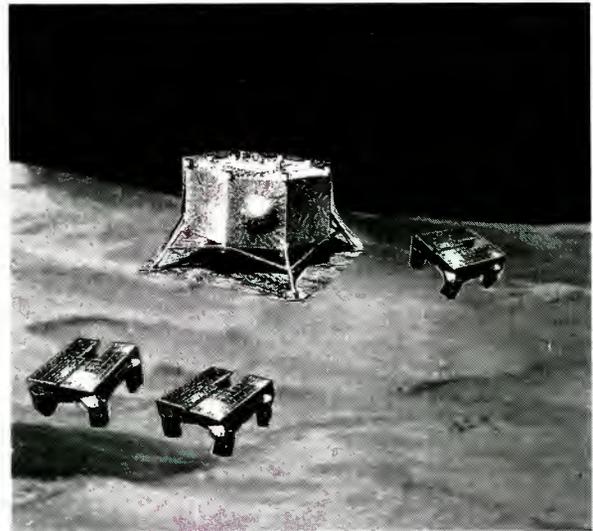


Fig.5 Cooperative Exploration by Buddy System

Reference

- [1] J.L.Loch, R.Desai, "Moose on the Loose: Toward Extended Mission Autonomy for Robotic Exploration of Planetary Surfaces," Proc. of ICAR'93, pp.97-101, 1993.
- [2] G.Giralt, "Remote Intervention Robot Autonomy and Real World Application Cases," Proc. of IEEE Int. Conf. on R&A, pp.541-547, 1993.
- [3] R.Chatila, R.Alami, S.Lacroix, J.Perret, C.Proust, "Planet Exploration by Robots : From Mission Planning to Autonomous Navigation," Proc. of ICAR'93, pp.91-96, 1993.
- [4] <http://mpfwww.jpl.nasa.gov/>
- [5] C.R.Weisbin, D.Lavery, G.Rodriguez, "Robotics Technology for Planetary Missions Into the 21st Century" Proc. of i-SAIRAS'97, pp.5-10, 1997.
- [6] L.Matthies, E.Gat, R.Harrison, B.Wilcox, R.Volpe, T.Litwin, "Mars Microrover Navigation : Performance Evaluation and Enhancement," Proc. of IROS'95, pp.433-440, 1995.
- [7] D.P.Miller, M.G.Slack, R.J.Firby, "Path planning and execution monitoring for a planetary rover," Proc. of IEEE Int. Conf. on R&A, pp.20-25, 1990.

- [8] R.Simmons, E.Krotkov, "Autonomous Planetary Exploration: From Ambler to APEX," Proc. of ICAR'93, pp.429-434, 1993.
- [9] E.Gat, R.Desai, R.Ivlev, J.Loch, and D.P.Miller, "Behavior Control for Robotic Exploration of Planetary Surfaces," IEEE Trans. on R&A, Vol.10, No.4, pp.490-503, 1994.
- [10] T.T.Lee, C.L.Shih, "Robust Path Planning in Unknown Environments via Local Distance Function," Proc. of i-SAIRAS'90, pp.251-254, 1990.
- [11] R.Simmons, E.Krotkov, L.Chrisman, F.Cozman, R.Goodwin, M.Hebert, L.Katragadda, S.Koenig, G.Krishna-swamy, Y.Shinoda, W.Whittaker, "Experience with Rover Navigation for Lunar-Like Terrains," Proc. of IROS'95, pp.441-446, 1995.
- [12] R.Morlanss, A.Liegeois, "A D.T.M based Path Planning Method for Planetary Rovers," Missions, Technologies et Conception des Vehicules Mobiles Planeraires, pp.499-507, 1992.
- [13] S.Hayati, R.Arvidson, "Long Range Science Rover (Rocky 7) Mojave Desert Field Tests," Proc. of i-SAIRAS'97, pp.361-367, 1997.
- [14] W.R.Whittaker, D.Bapna, M.W. Maimone, E.Rollins, "Atacame Desert Trek: A Planetary Analog Field Experiment" Proc. of i-SAIRAS'97, pp.355-360, 1997.
- [15] T.Kubota, I.Nakatani, Y.Kuroda, T.Adachi, H.Saito, T.Iijima, "Izu-Ohshima Field Tests for Autonomous Planetary Rover," Proc. of IROS'98, pp.588-593, 1998.
- [16] I.Nakatani, T.Kubota, T.Adachi, H.Saito, "Navigation Technique for Planetary Rover," Proc. of ICAR'95, pp.201-205, 1995.
- [17] I.Nakatani, T.Kubota, T.Adachi, H.Saitou, S.Okamoto, "Control Technique for Planetary Rover", 3rd Int. Symposium on Artificial Intelligence, Robotics and Automation for Space, pp.103-106, 1994.
- [18] Y.Kuroda, K.Kondo, T.Miyata, M.Makino, "The Micro5 Suspension System for Small Long-range Planetary Rover", ISAS 8th Workshop on Astrodynamics and Flight Mechanics, pp.287-292, 1998.

ACTIVE SURFACE IMAGING SYSTEM (ASIS)

Reinhold Flatscher

Dornier Satellite Systems GmbH, Programs Science and Earth Observation Department
D-88039 Friedrichshafen, Germany
phone: +49 7545 8 4979, fax: +49 7545 8 4105, e-mail: *reinhold.flatscher@dss.dornier.dasa.de*

Andreas Ullrich

Riegl Laser Measurement Systems
A-3580 Horn, Austria
phone: +43 2982 4211, fax: +43 2982 4210, e-mail: *aullrich@riegl.co.at*

Gerhard Paar

Joanneum Research, Institute of Digital Image Processing
A-8010 Graz, Austria
phone: +43 316876 1716, fax: +43 316876 1720, e-mail: *gerhard.paar@joanneum.ac.at*

Gerd-Joachim Ulbrich

ESTEC, Mechanical Systems Division – Optics Section (TOS-MMO)
NL-2200 AG Noordwijk ZH, The Netherlands
phone: +31 71565 5515, fax: +31 71565 5430, e-mail: *gulbrich@estec.esa.nl*

ABSTRACT

A high-resolution imaging and ranging laser sensor for space applications was designed, realized and tested. The work comprised the scanning laser radar hardware, software for quick visualization of range images, software to calculate and display the related digital elevation model, and a test bed equipped with several calibrated mobile targets. Measuring on diffuse reflecting targets the typical ranging distance is up to 300 m. The maximum scan range is 60 x 90 degrees. A bi-directional scan of 60 degrees is performed within 15 seconds acquiring 300 x 300 range pixels.

For demonstration of the measurement capabilities of the sensor device a test bed was established. "Natural" stationary targets as well as mobile reference targets fixed on a flexible rack were used to characterize the sensor. In the future any active or passive range imaging sensor can be compared to the ASIS "reference" employing that test bed.

1. INTRODUCTION AND HISTORY

Space research in Earth orbit, lunar and interplanetary environment has yielded an urgent demand for autonomous technologies in recent years. One major technique in this context is based on active and passive imaging sensors. Measurements for application in robotics, navigation, surface reconstruction and post-launch quality control make all use of computer vision.

Laser sensor instrumentation and in particular the pulsed direct-detection technique is the ideal choice for reliable distance measurements in space, thanks to their low to moderate system complexity, the excellent performance and their high instrument layout flexibility. The achievable accuracy is in the order of millimeters.

Pulsed systems are very attractive for a wide range of space applications. This clearly has been confirmed by the work performed under Work Order No. 01 of ESA contract 9230/90 "Laser Sensors for Planetary Research". The core of this work covered the design and breadboarding of a modular laser sensor for planetary research missions, typically to be used for mid-range topographic mapping and during the descent and landing phase.

Some applications which have gained major interest recently, such as laser sensors for rover vehicles, rendezvous and docking, robotics, etc., impose much more stringent limitations on instrument size, mass and power consumption. The next activity in the laser radar line was an advanced miniaturized pulsed laser-sensor demonstrator for near-range space applications (DEAL). Particular emphasis was given to the compactness of the sensor, to the interfacing of modules by glass fibers and to sophisticated techniques of fast signal processing.

Apart from these laser-sensor scientific developments, ESA carried out work on the establishment of high-resolution 3D terrain models of planetary bodies within the frame of ESA contract 9195/90. That work was based on optical imaging with a view to identify and demonstrate the required methods and algorithms for end-to-end data processing.

Now, both lines of development were brought together comprising the laser opto-electronic front-end and the signal processing back-end. The realized ASIS setup allows the performance capability of an integrated sensor device to be validated in front of representative planetary terrain models serving as test scenes.

2. CONTENT OF WORK

ASIS comprises the realization of an imaging laser sensor and the establishment of a test bed equipped with a stationary "natural" target and several mobile targets. The modular approach of the sensor hardware presents only one electronics unit and one scanner unit both connected via fiber links. Range is measured by using the well-established time-of-flight technique that determines the roundtrip time of a short emitted laser pulse. The start event is defined by the firing of the laser, whereas receiver and subsequent signal processing generate the stop pulse.

A standard personal computer (PC) acts as data display and data collection device. It is also possible to test or reprogram the imaging sensor by the PC via an optional serial interface. Range and intensity images are displayed without geometrical correction in real time on the PC's monitor to get a first impression of the acquired scenery. The final data product is a Digital Elevation Model (DEM) which includes the measured angle information, too. It is also displayed in real time on the PC's display.

The laser transmitter is based on a triple-stack pulsed diode laser running at high repetition rate. High basic range resolution is achieved by short laser pulses of 9 ns generated with the aid of paralleled avalanche transistors. The start event for the time duration measurement is directly taken from the electrical signal firing the laser. No thermoelectric cooling is foreseen to keep the power consumption low.

The receiver relies on the direct detection technique yielding lowest complexity. An optical interference filter suppresses background radiation. The backscattered light is detected by an avalanche photodiode (APD) with stabilized bias. The detector is electronically protected against optical and, thus, electrical overload in case of strong reflections in the near range, e.g. from targets marked by retro-reflectors. The signal of the APD is amplified by a transimpedance amplifier realized in "discrete" technology to give a wide dynamic range and fast recovery at strong light levels in the near range.

In parallel, the amplitude of the received signal is sampled and AD-converted on a shot-to-shot basis. As all involved stages introduce signal dependent internal delays the raw range result must be corrected afterwards. Furthermore, this path provides the intensity of the received signal as an add-on data product beside the range information itself. The intensity signal is logarithmically processed to cover the enormous dynamic range of 130 dB the system works with. A zero-crossing network forming a resonant trigger circuit with the highest dynamic range reported ever gains the stop event for the time-of-flight measurement. The received pulse excites the resonant circuit and the first zero crossing is taken as a stop event. A time to digital

converter evaluates the time interval related to the distance of the target. It is based on a circular counter chain with gate delays. The basic resolution of the converter is 2.5 cm. Any electronic component inside the signal processing chain and related internal delay induces range errors depending on signal amplitude and temperature. Thus, both terms are measured and serve as range correction inputs to cover the system's huge operational dynamic range on a shot to shot basis.

The optical architecture relies on a biaxial design strictly separating transmit and receive paths. So, an excellent near range behavior can be achieved because the strong light level brought back by near targets is attenuated by the poor beam overlap nearby the sensor. In the far field where weak signals are experienced the overlap is perfectly one.

The fast line scan is based on a polygon scanner with a four-facet wheel. So, the scan range in this direction is fixed to $\pm 30^\circ$. Transmitter and receiver beam hit side-by-side the polygon facet. The focal length of both, transmitter and receiver optics, is 82 mm. The rotation speed of the wheel is nominal 5 rps or 20 line scans/s but can be adjusted by software between 2.5 rps and 20 rps. Rotating the entire „mobile“ part of the scanner head with the aid of a stepper motor performs the slower frame scan. So, a very compact scanner unit with extended scan range could be realized. The range of the slow scan is nominal $\pm 30^\circ$ but can be adjusted by software from $\pm 10^\circ$ up to $\pm 45^\circ$ maximum.

A standard personal computer is connected to the ASIS hardware. The PC was also used for testing the demonstrator on system and subsystem level. Range and intensity images are displayed in real-time on the monitor without geometrical correction to get a first impression of the scene. Further statistical evaluations are performed by software modules. The final data product is a Digital Elevation Model including also the measured angle information in two axes. It is displayed on-line and in real time as well. All sensor software packages have been realized with Borland's Delphi and are running in the modern operating system environment NT 4.0 or Windows 95.

The active imaging system serves as a reference sensor in a test-bed. A flexible rack has been built equipped with up to nine surface panels (40 cm x 40 cm size) of different shape, texture and roughness. Dedicated pyramidal, spherical or cylindrical targets allow to test the sensor's spatial resolution, linearity, accuracy and dynamic range. The performance of ASIS is superior and a single shot range resolution of 16 mm (1- σ value) has been measured for high SNR (10^6).

Fig. 1 and Fig. 2 show both main units of ASIS and indicate the portability of the realized system. Table 1 summarizes all features and performance characteristics of ASIS. Fig. 5 shows more details of the Scanner Unit.

3. MEASUREMENT RESULTS

Fig. 3 represents typical range data acquired with ASIS. The data set has been collected by a standard notebook providing color encoded range images for quick visualization. A view of a natural stationary target is shown for comparison. The scan range was 60 degree in both, horizontal and vertical direction. The image contains 333×333 range pixels with an angular separation of 0.2 grad. Areas in which no laser range measurements can be performed due to low reflectivity or beam deflections are shown in black in the false color encoded range image. The distance range to be displayed can be arbitrarily set between 0 and 300 m. In the shown figure the limits were set to 26 m and 75 m. Distances outside of this preselected range are measured as well but indicated in gray, e.g. the trees in the background.

Fig. 4 shows the Digital Elevation Model processed online from a typical planetary target, a sand dune. The image shows a top view of the dune with color-coded height above ground in world coordinates. The displayed detail measures 10 m times 8 m. In addition isolines are overlaid to that picture indicating smooth slopes. Each line indicates a step in height of 15 cm. The photo on top shows the entire scenery with the detail used for further processing.

4. CONCLUSION

An active surface imaging system with extended scan range was realized. Excellent range resolution and accuracy were achieved. A wide dynamic range characterizes the laser sensor and is available on a shot to shot basis. In addition, the receiver is protected against optical overload. Combining a fast rotating polygon scanner with a stepped mobile part yielded a very compact sensor unit with potential to further extend the scan range. The imaging system is portable, easy to setup and easy to use. The system can serve all short range imaging applications in the robotics, rover and rendezvous and docking business.

5. ACKNOWLEDGEMENT

The ASIS project was sponsored by ESA under ESTEC Contract No. 9230/90/NL/PB(SC), CCN09 and funded by the Technical Research Program (TRP). ASIS is the latest activity of active laser imaging technology embedded in a frame contract initiated in early 1991. Beside Dornier Satellite Systems GmbH (D) leading the consortium, also Riegler Laser Measurement Systems (A) and Joanneum Research Institute (A) were involved. Riegler manufactured the laser sensor and provided the required software whereas Joanneum was responsible for the Digital Elevation Model and delivered the mobile targets.

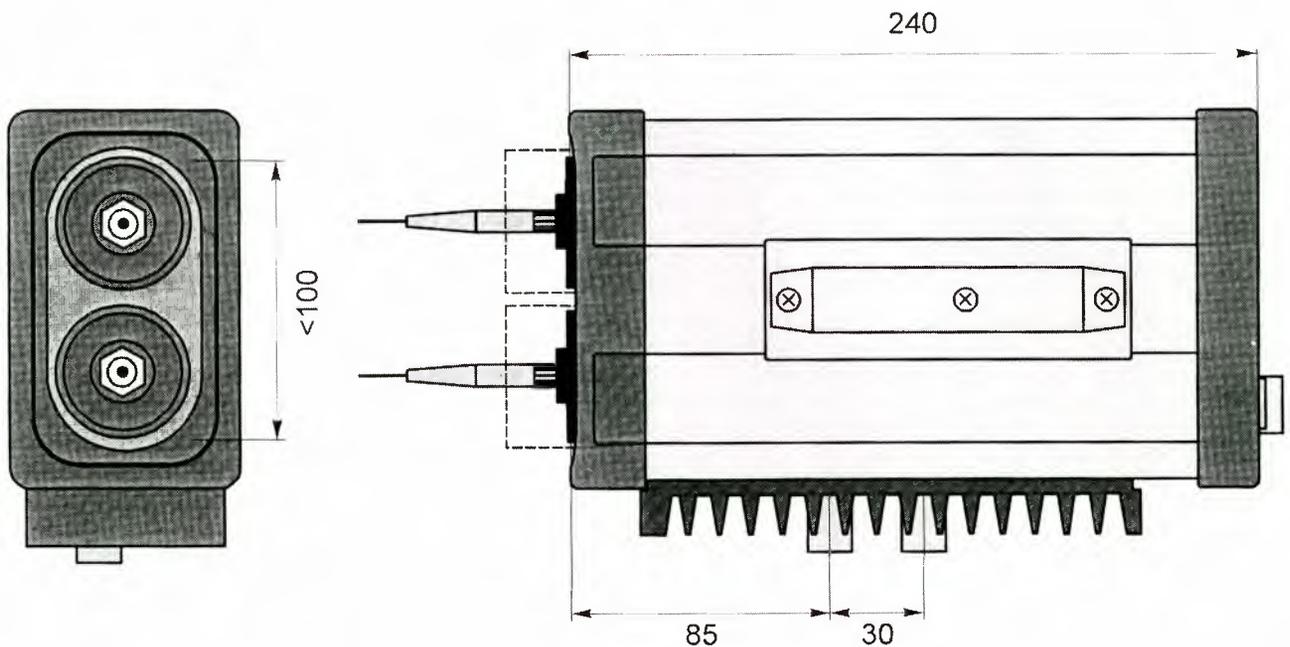


Fig. 1: Electronics Unit (dimensions in mm). Connected to Scanner Unit via glass fibers.

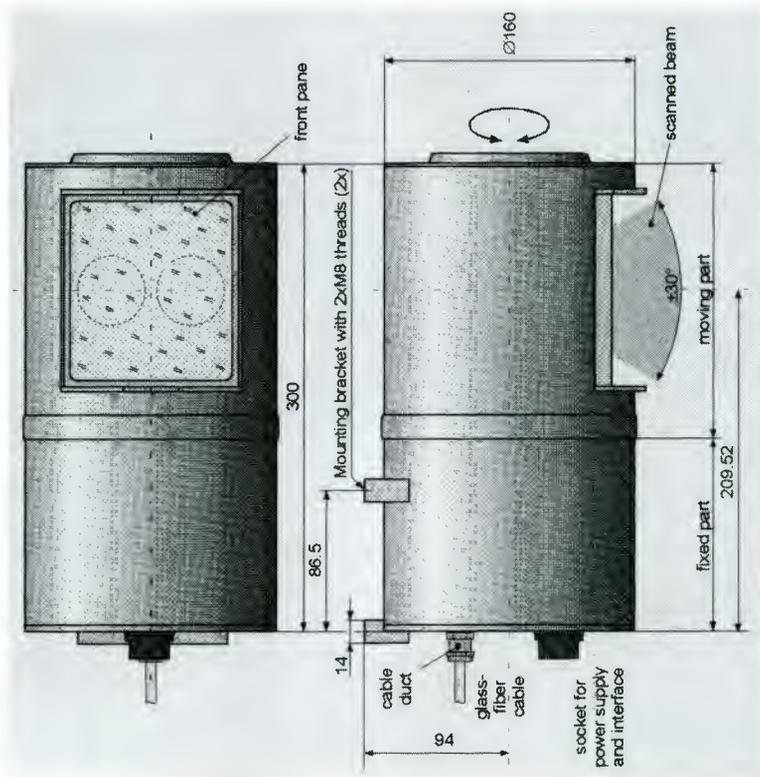
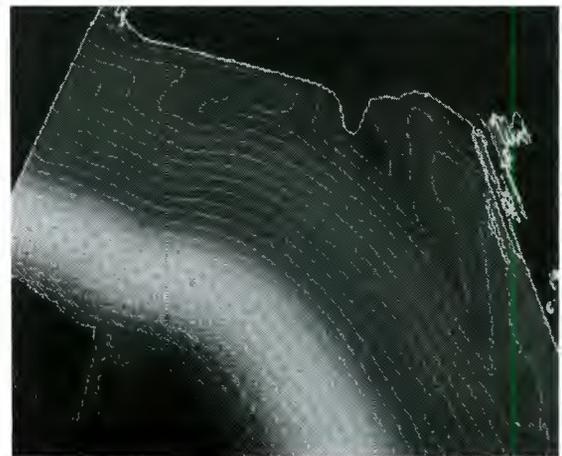
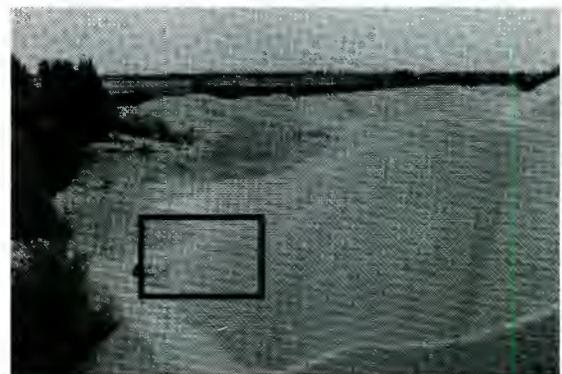


Fig. 2: Scanner Unit (dimensions in mm)



26 m 75 m

Fig. 3: Measured range image of a quarry. Photo on top for comparison.

Fig. 4: Digital Elevation Model of a sand dune overlaid with isolines (height steps of 15 cm).

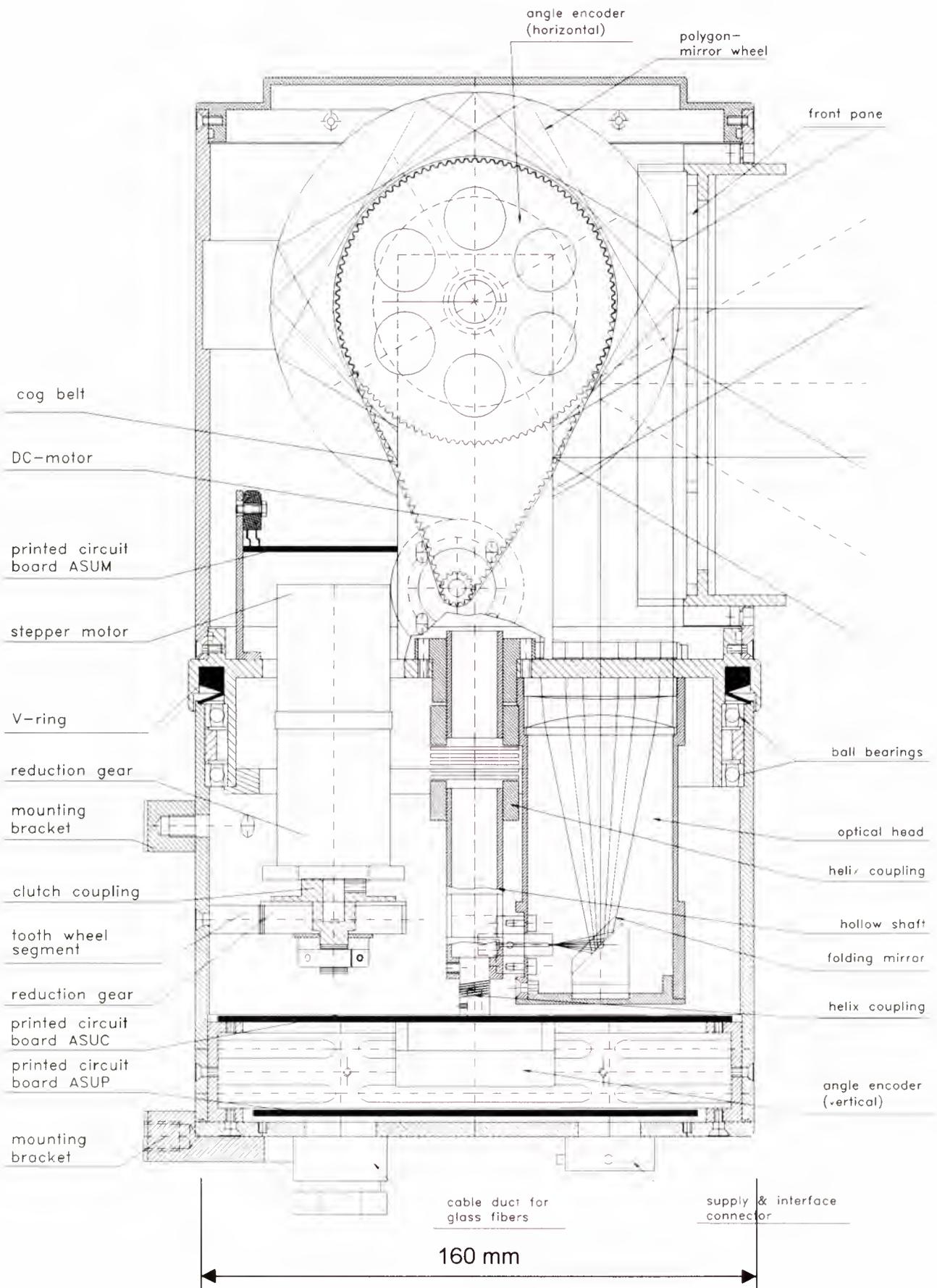


Fig. 5: Cross-sectional view of scanner subsystem.

RANGE MEASUREMENT		
Time-of-flight measurement technique	time-to-digital converter (ring counter with gate delay)	
Microcontroller used for range processing	Intel MCS 296, 16 bit, 50 MHz with basic DSP functions	
Measurement range	0-300 m	
Internal resolution of range measurement	8 mm (all corrections included)	
Range measurement standard deviation	1.6 cm at high SNR	
Range image acquisition time	15 sec. for 60 x 60 degrees (5 sec. for 20 x 60 deg)	
OPTICAL CHARACTERISTICS		
Transmitter / receiver optics (TX / RX)	Ø 42 mm at focal length of 86 mm	
Fiber interface to Scanner Unit	300 µm (TX) and 600 µm (RX) multimode	
Beam divergence	3.5 mrad (TX) and 7 mrad (RX)	
LASER HEAD		
Laser diode	EG&G, PGAU3S09, 3-stack device	
Pulse width	9.2 ns	
Laser repetition rate	18 kHz maximum during normal operation 24 kHz with no drop in energy, up to 36 kHz tested	
Pulse energy in transmitter fiber	266 nJ at 905 nm	
RECEIVER		
Type	direct detection receiver, protected against optical overload	
Detector	EG&G, C30902E, 500 µm diameter	
Dynamic range of system	130 dB electrical on a shot-to-shot basis	
Minimum optical receiver power	1.8 nW at SNR of 9.5 dB optical	
Amplitude detection	logarithmic amplifier and 12 bit A/D-converter	
SCANNER		
Frame scanner range (programmable)	±10° up to ±45° (±30° nominal)	
Scan step	0.2 grad (gon) in both directions	
Frame scan speed	4 grad/s nominal up to 16 grad/s	
Polygon speed (programmable)	5 rps for 20 line scans/s, range 2.5 rps up to 20 rps	
Scanner angle resolution	0.02 grad frame scanner and 0.04 grad line scanner	
INTERFACES		
Data delivery to personal computer	one data pack every scan line with 8 bytes/pixel	
CODING	Range	2 bytes in units of 8 mm
	Inclination	2 bytes in units of 0.01 grad
	Bearing	2 bytes in units of 0.01 grad
	Intensity	12 bits
	Housekeeping	4 bits
Interfaces to PC	1 parallel (ECP) during normal operation 1 serial (RS232) in service or test mode only	
Internal interface between both units	4 wire synchronous serial port (bi-directional)	
Scanner dimensions	300 mm x 160 mm Ø (cylinder)	
Scanner mass and power	5 kg and 12 W at 12 V	
Electronics dimensions	240 mm x 142 mm x 76 mm	
Electronics mass and power	2.5 kg and 7 W	

Table 1: Features and characteristics of realized imaging laser sensor.

DEMONSTRATION OF THE PLANETARY UTILISATION TESTBED

M Van Winnendael
European Space Agency

Paper not available

Control Experiments on ETS-7

ETS-7 SPACE ROBOT TELEOPERATION THROUGH VIRTUAL FORCE REFLECTION

Luis F. Peñín, Kohtaro Matsumoto and Sachiko Wakabayashi

National Aerospace Laboratory (NAL), Space Teleoperation Research Group
7-44-1 Jindaiji Higashi-machi, Chofu City, 182-8522 Tokyo, Japan
phone: +81-422-403173, fax: +81-422-403148, e-mail: penin@nal.go.jp

ABSTRACT

The National Aerospace Laboratory (NAL) is participating in the ETS-7 robot experiments in order to establish the basic technologies for on-orbit truss assembly by ground teleoperation. Several different approaches have been tested so far. This paper describes one of them: the application of force reflecting (FR) hand controllers to improve continuous teleoperation with long communication delays.

After a comprehensive analysis of the current state-of-the-art, four distinct fundamental ideas have been implemented: a) the use of FR to limit speed of command; b) the use of potential force fields for guidance during grasping and inspection; c) the use of the FR hand controller to generate physical constraints and d) the use of the FR device to intuitively correct modeling errors after probing the environment.

All of them have been successfully tested on the ETS-7 robot system using NAL's Ground Teleoperation Facility (GTF) and two 2-DOF Force Reflecting Joysticks. This paper reports the conduction and results of these experiments.

1. INTRODUCTION

Space robot systems and on-orbit telerobotics technology will play an essential role in the construction and maintenance of large-scale structures, such as the International Space Station (ISS). The ETS-7 satellite was launched by NASDA in November 1997 to conduct rendezvous docking and space robot technologies experiments. The National Aerospace Laboratory (NAL) of Japan is participating in the robot experiments in order to establish the basic technologies for on-orbit truss assembly and deployment by ground remote operation of the robot located on the satellite.

It is well known that continuous on-orbit teleoperation of robots by operators on Earth is seriously impeded by signal transmission delays imposed by limits on computer processing at transmission stations and satellite relay stations. For the ETS-7 the time delay is normally between 5-7 seconds.

NAL has been performing research in the last few years on how to overcome the disturbing effect of time delay to achieve smooth and effective teleoperation. The application of the research on the ETS-7 robot arm has produced several promising approaches, as reported in

[1] and [2]. For a detailed description of NAL's complete system and objectives regarding ETS-7 robot teleoperation please refer to the companion paper [3].

This paper describes another of the solutions adopted from conception to final application. The approach makes use of force reflecting (FR) hand controllers to improve continuous teleoperation with long communication delays. It is, to our knowledge, the first extensive application of FR for the ground teleoperation of a space robot.

First, in section 2 a comprehensive analysis of the current state-of-the-art on the use of FR for time delayed teleoperation is presented. The advantages and difficulties introduced by the use of FR are also addressed here. Section 3 is dedicated to describe the system employed to conduct the experiments. Section 4 addresses the tasks being conducted, while sections 5 presents the different fundamental uses of FR implemented so far: a) the use of FR to limit speed of command; b) the use of potential force fields to guide during grasping and inspection; c) the use of the FR hand controller to generate physical constraints and d) the use of the FR device to intuitively correct modeling errors after probing the environment. Finally, section 6 presents the conclusions.

2. A BRIEF REVIEW ON THE USE OF 'FR' FOR TIME DELAYED TELEOPERATION

It is well known that continuous teleoperation can be dramatically improved with the addition of some kind of FR [4]. FR decreases both the time of operation and the forces exerted during contact, making the operation smoother and safer.

It is also known that time delay makes the use of FR extremely difficult. Despite this and owing to its importance there has been extensive research in the field during the last decade, as shown in this section and profoundly explained in references [5] [6]. Proposals for time delayed teleoperation without FR, such as teleautomation [7], tele-sensor programming [8] or control based on a predictive observer [9], are also abundant in the literature.

Existing approaches for time delayed teleoperation with FR can be broadly classified into two groups: proposals for bilateral systems (master and slave coupled in both position/velocity and force with the slave, also known as FFB) and proposals for non-bilateral systems.

Table 1. Conditions of the experiments conducted with the different proposals for FR time delayed teleoperation.

Method	dof	Freq. (Hz)	Task	Control	Delay
[10]	1	500	Hard contact	Pos.	2 s
[11]	1	N/A	Hard contact	Pos.	1 s
[12]	1	1000	Hard contact	Pos.	30 ms
[13]	1	N/A	Basic contact	Pos.	2 s
[14]	6	1	Basic contact	Pos.	500 ms
[15]	1	350	Basic contact	Pos.	320 ms
[17]	6	30	Following of box contour	Pos.	3 s
[18]	1	15	Grapple and fitting	Pos.	2-4 s
[19]	3	N/A	ORU exchange Open door	Vel. Force	5 s
*	2-3	4 / 10	Assembly	Pos	5-7 s

* This paper

2.1 Proposals for bilateral systems

There are two different approaches to eliminate instability of bilateral systems with time delay. The first one makes use of the two-port approach and passivity theories: FFB with time delay based in passivity [10], FFB using wave variables [11] and FFB for ideal kinaesthetic coupling [12].

The second group of approaches tries to solve the problem from the control theory point of view: FFB with tele-monitoring [13], FFB based on a Virtual Internal Model [14] and FFB through a computer network [15] [16].

2.2 Proposals for non-bilateral systems

Non-bilateral FR means that the operator is not kinaesthetically coupled with the slave but instead is receiving force information through another physiological channel (visual and indirect FR) or the force reference is different from the one generated at the remote zone. Regarding the latter case the main proposals are: Teleprogramming [17], Predictive Operator Aid with Force Reflection [18] and Predictive system that tolerates geometric errors.

2.3 Discussion

Table 1 shows the characteristics of the experiments successfully carried out with the different methods mentioned above. It includes the number of DOF, the transmission frequency, the type of task, the type of control (position, velocity or force command) and the maximum time delay supported.

Almost none of these has been tested under the difficult circumstances of operation of space robots: high flexibility of the arm, high backlash, low accuracy, low communication bandwidth, long time delay, short operation time, etc. The last line corresponds to the experiments that are presented in this paper and that have been tested with the ETS-7 robot arm.

First, it is worth noting that almost all of these use position command, which is more suitable for complex and precise tasks and more intuitive than rate control when using FR.

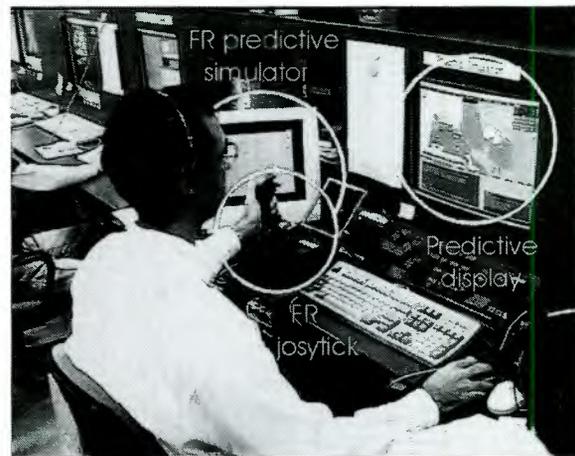


Figure 1. FR Teleoperation of ETS-7 robot arm

It is seen that bilateral schemes are used only in 1 DOF basic contact tasks for delays of up to 1 or 2 seconds, and even so heavily degraded. Hence, it can be affirmed that nowadays they are not well suited for space teleoperation.

On the contrary non-bilateral proposals are operational under several seconds of time delay for more complex tasks and up to 6 DOF. It is true that the main advantage of FR (coupling with the task) is lost but some other advantages, such as intuitiveness, reduction of operating time and safety remain.

Another advantage is that visual aids tend to overload the on-line mental modeling done by the operator, while hand force cues are easily integrated by the human brain with no need of complex processing [20]. Hence FR is a very interesting and direct way of 'displaying' information to the operator and can be used in combination with other visual and acoustic aids without compromising the operator's performance.

Our effort has been to use some of the ideas present in the preceding proposals in combination with new ones to conduct experiments of non-bilateral FR time delayed teleoperation on a real space robot arm. The final aim has been to prove the advantages of this kind of systems in order to overcome time delay and point out the drawbacks that still require more research.

3. DESCRIPTION OF THE SYSTEM

The FR experiments have been conducted using NAL's Ground Teleoperation Facility (GTF) [21] located at the Tsukuba Space Center (TKSC).

Communication between NAL's GTF and the on-board robot arm is done through NASDA's facility. Commands are sent at a frequency of 4 Hz and telemetry is received at a frequency of 10 Hz. Commands in teleoperation mode refer to the arm's tip position. Time delay between command and telemetry is typically 5-7 s.

Figure 1 shows a photo taken during one of the experiments. The main devices involved are shown highlighted. They include 1 or 2 FR joystick, an FR predictive simulator and a predictive display.

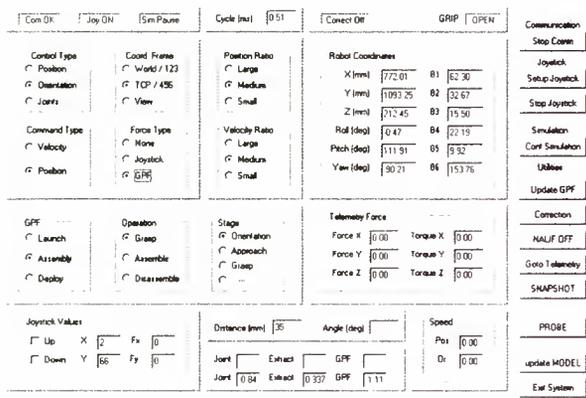


Figure 2 Interface of the FR predictive simulator

• FR Joystick

The FR hand controller is a 2 DOF Joystick from Immersion Inc. It has a workspace of 15.2 x 15.2 cm and is capable of generating output forces up to 8.9 N with a bandwidth of 120 Hz. The joystick deflection in both axes is used as position command for the on-board robot's tip.

Since the joystick workspace is much smaller than that of the robot, it is necessary to scale and re-index the joystick movements to generate the commands. Scaling is particularly useful when carrying out precise tasks. Re-indexing permits the operator to be in a comfortable posture at all times.

On the other hand, since one joystick has only 2 DOF, different modes of operation have been defined to allow each task to be performed in different steps. Experiments have been conducted both with only one or two joysticks, up to a total of 4 DOFs.

• Predictive display

The commercially available simulator Telegrip is used as the predictive display. It makes use of a CAD model of the TSE and of the robot arm.

Three different robots are displayed at the same time during operation: a) command robot, b) telemetry robot and c) FR robot. The command robot shows where the robot is commanded to go. The telemetry robot shows the delayed information of the robot's current position. And the FR robot shows the actual robot model that is used by the FR engine to generate the forces on the joystick. See section 5.4 regarding the *snapshot* approach.

• FR predictive simulator

This computer is the system core and acts as the interface between the FR joystick and the computer responsible for transmitting commands to the robot arm.

It generates position commands for the robot's tip using the joysticks encoders values as a reference. In addition, it is also responsible for giving the appropriate force commands to the joystick actuators to implement FR.

To achieve both functions, the FR predictive simulator includes the following features:

- Force reflection engine.
- Motion planning engine
- CAD model of the environment (i.e. TSE).
- Kinematic description of the various tasks.
- Library of different type constraints (planes, cylinders, etc.)
- Library of different types of contacts (hard, soft, informative, etc.)
- Interface (see Figure 2).

The motion-planning engine calculates the final commands for the robot using the references coming from the joystick and the current mode of operation specified on the interface. Velocity and acceleration motion planning is also performed here to keep them under a specific profile. The cycle time is 4 Hz.

The FR engine calculates the forces to be generated by the joystick actuators. It makes use of the design model static and kinematic features, the FR robot's tip current position and the current mode of operation, as well as the libraries of constraints and contacts, explained in the next section. The cycle time varies, but is commonly between 1000-2000 Hz, which is high enough for the FR to provide a good sense of touch to the operator.

The interface, shown in Figure 2, allows the operator to keep track of all variables deemed important (robot position and speed command, force being generated, telemetry force, parameters specific to the task, etc.). It also permits switching between different modes of operation and activating specific functions, such as the *snapshot* or *update model* features.

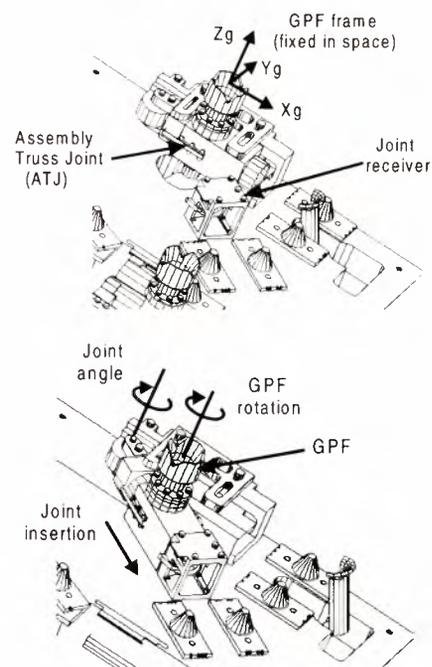


Figure 3 TSE's assembly Truss Joint (TJ) in a) stowed and b) assembled configurations

4. TASKS CARRIED OUT

Three different tasks have been conducted in order to demonstrate the advantages of the new FR system. They have been tested using the ETS-7 robot arm working on NAL's TSE. Apart from the difficulty of each task, one of the main concerns has been execution time, which owing to operability reasons is nominally of 20 minutes. The tasks successfully accomplished are the following:

- *Safe inspection of the TSE unit.*

Movement of the robot close to the TSE surface for visual inspection. The FR capability is used to avoid collisions and follow virtual contours around objects.

- *Vision guided GPF grasping.*

Grasping of the GPF guided by the data provided by the vision system. FR is used to guide intuitively the movement of the operator's hand during grasping.

- *Assembly of the Assembly Truss Joint (TJ)*

Both assembly and disassembly of the Assembly Truss Joint (TJ) have been accomplished.

To assemble the TJ requires the most precise, difficult and strong arm tip motion control of the ETS-7 tasks. This is mainly because of the features of the TJ (Figure 2), which was designed for assembly without any hand-over tasks [22].

The TJ assemble operation has four stages: a) to swing the TJ to the front of the joint receiver (JR), b) to rotate the GPF 60 degree to unlock and extend the TJ top, c) to insert the TJ into the JR and d) to lock the TJ by rotation. The TJ has a spring for automatic stowing.

It is a 3 DOF task. When using a unique 2 DOF joystick, automatic adjustment of roll rotation was done.

- *Deployment of the Deployable Truss Structure (DT)*

This task was successfully carried out at the end of May combining the FR system with the output of the generalized visual aid presented in a companion paper [1]. No detailed analysis of the results have been done yet

4.1 Robot control

Figure 3 shows the reference frames and parameters used. All experiments were conducted by moving the robot in the hand frame system (XYZ). The FR joystick was employed as a generator of position commands to the robot's tip. One joystick ($X_j Y_j$) is always used to move in the robot's hand frame (YZ) plane. If two joysticks are employed, the use of second one depends on the task. The frame $X_g Y_g Z_g$ is a fixed frame located at the TJ GPF stowed position.

5. APPLICATION OF FORCE REFLECTION FOR SPACE TELEOPERATION

Four different uses of FR have been tested so far both in isolated conditions and in combination. They have been successfully applied to the different tasks mentioned.

5.1 Monitoring and limitation of command speed

This is the most basic use of FR. As the FR joystick has been employed as a generator of position commands to

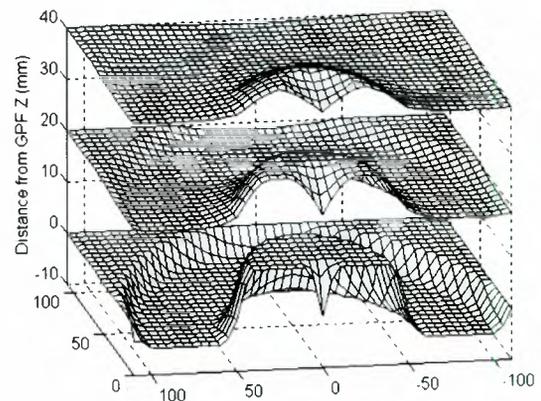


Figure 4 Force field for GPF grasping

the robot's tip, the speed of the robot is linearly related to the speed of the movement made by the hand.

In this context, it is very useful to limit the joystick speed using FR. Two approaches have been implemented: a) generation of a force that keeps the joystick fixed at the position where the violation occurred until the operator decides to continue, and b) generation of a damping force against movement that limits the speed.

5.2 Implementation of a field of virtual forces

The second application for the use of FR has been the development of a guiding system based on the concept of potential force fields.

The basic idea is to generate forces on the FR hand controller to indicate to the operator where he should move to successfully complete the task. He then only has to follow this movement with his hand. The important fact is that he keeps the control over the movement and can stop it, reverse it or even overrun the guiding cue whenever he thinks necessary.

Consider the GPF grasping application. GPF position is known from design data. A 3D force field is created around the GPF like in Figure 4. The GPF grasping point is in the origin of the GPF frame ($X_g=Y_g=Z_g=0$)

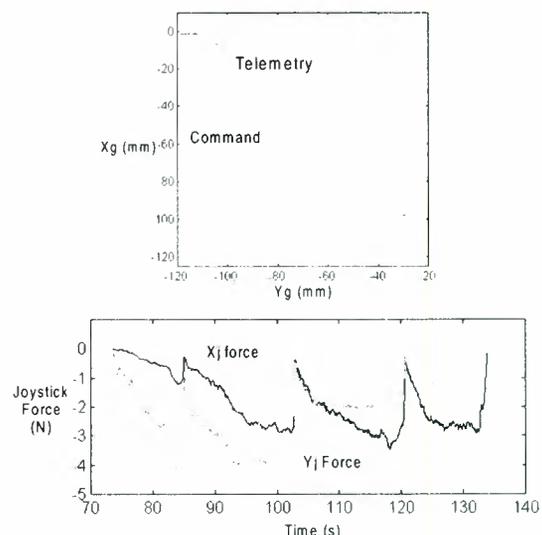


Figure 5 a) Command and telemetry position and b) virtual force generated in the joystick during the TSE inspection

mm). Positive values of force indicate attracting forces to the Z_g axis, while negative values indicate repelling forces. Joystick movement is in the $X_g Y_g$ plane while advance in the Z_g axis is made at a specific speed by pressing a pushbutton of the joystick. Vision data is used to update the GPF location and the corresponding force field along with it. By this method, the grasping is done very smoothly and data from the visual system is conveyed to the operator in an intuitive manner.

An identical approach is used in the orientation domain to move the robot to have its hand frame parallel to the GPF's frame at the beginning of the grasping operation.

Fields of virtual forces can also be used very easily for inspection. They can be put around objects to avoid collision and at the same time to guide the operator along the right path.

A part of the TSE has been safely inspected using this approach. Figure 5a shows the trajectory of both the command and telemetry robots while avoiding collision with the TJ GPF. The robot describes an almost perfect circle around the GPF area. The virtual field pattern felt by the operator is the one shown in Figure 4 with $Z_g=0$; Figure 5b presents the force generated by the joystick axis. It seems somewhat degraded because of the joystick indexing but it does not affect the final results.

5.3 Use of force cues and virtual constraints

To increase the flexibility of the use of virtual forces two new developments have been made: force cues and virtual constraints.

A library of common 3D geometric surfaces has been developed. It includes scalable planes, cones, spheres, etc. They can be easily placed in the environment to represent virtual constraints so that the FR engine can recognize them and generate the appropriate force when contacted by the robot's tip.

Each surface has some special features for FR that can be updated or changed anytime during operation. For example, consider a plane (Figure 6a). Contact can be simulated only in one of either faces (i.e. contact with a stiff surface) or in both at the same time (i.e. constrained movement along the plane). But moreover, FR does not have to be limited to the plane itself, but potential fields can also be applied on the surroundings of the surface.

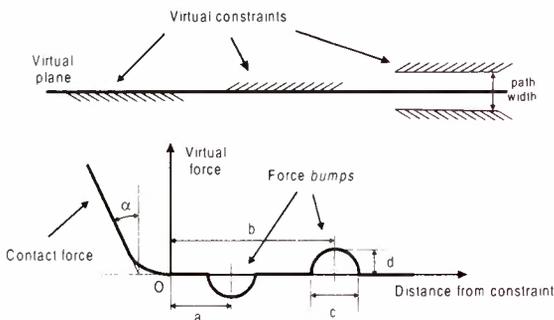


Figure 6 a) Types of constraints in a plane and b) parametric definition of a constraint

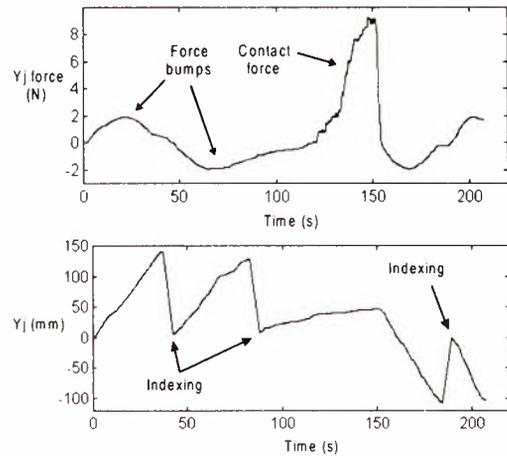


Figure 7 a) Force cues and b) axis evolution of the joystick when approaching a virtual plane.

Figure 6b shows the parametric force profile that is defined for each face of the plane. Before contact there are two *force bumps*. The first one is used to tell the operator that he is approaching contact. The second one helps him to maintain the contact. Finally, different contact behaviors can be specified to simulate complex interactions.

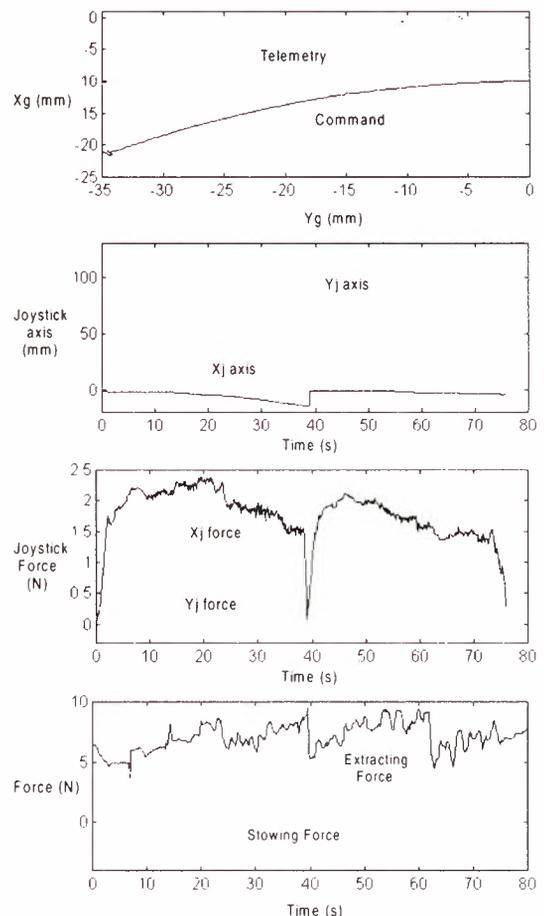


Figure 8 Evolution of different variables during the stowing operation of the TJ using force constraints

Figure 7a shows an actual force pattern of contact with a virtual plane. Note that the abscissa is time. The actual joystick movement is displayed in the lower figure.

Figure 8 shows the application of a constraint in the form of a cylinder for the stowing of the TJ. Both cylinder faces act as virtual walls that guide the movement of the joystick through space. The TJ was pushed constantly to be completely retracted creating a gap between telemetry and command due to the compliance of the arm. One sole indexing operation is required at around 40 s. Force reflected to the operator allows him to follow the path very intuitively. Joystick's DOF are directly related to the robot hand frame, but as orientation is automatically adjusted the operator advances moving one DOF (Y_j) while feeling the constraint in the other (X_j).

The inserting force to keep the joint retracted is perfectly maintained in the 5-10 N range, while the stowing force due to the spring in the joint decreases as the joint is being stowed.

5.4 The *snapshot* approach

The disadvantage of the system presented so far is that it does not account for mismatching between robot command and telemetry due to the high compliance of the robot arm (see Figure 8a). To solve this problem the *snapshot* approach was conceived.

For safety reasons and to help perform contact tasks the compliance of space robots is deliberately high. Nominal values for the ETS-7 robot arm are between 0.2-0.8 N/mm. Such compliance creates an important mismatching between command and telemetry positions of the arm when high forces are present.

This effect is particularly harmful to the approach described in the preceding section. Virtual forces cannot be generated based either in telemetry or in command robot models. A new robot model, called FR robot model, is needed. Virtual forces are generated regarding the FR robot's tip position in the environment.

The idea is to try to use as much as possible FR based on where the robot currently is but without compromising smoothness and speed of operation. In this context the FR robot is always between the command and telemetry robots, trying to be close to telemetry as much as possible.

We will now describe the operation procedure. If no real forces are present in the environment, the three robots will follow the same trajectory and there is no need to perform a *snapshot* (Figure 5a). When forces appear command and telemetry robots tend to separate. But as command separates from telemetry the virtual forces generated by the FR engine lose validity. In this situation a *snapshot* is required.

To make a *snapshot* means that the FR robot is automatically moved to telemetry without changing the command. That is, the relative positions of the FR and command robots change and the computer stores the transformation between them. After the *snapshot* the operator movements and sensation of virtual forces are

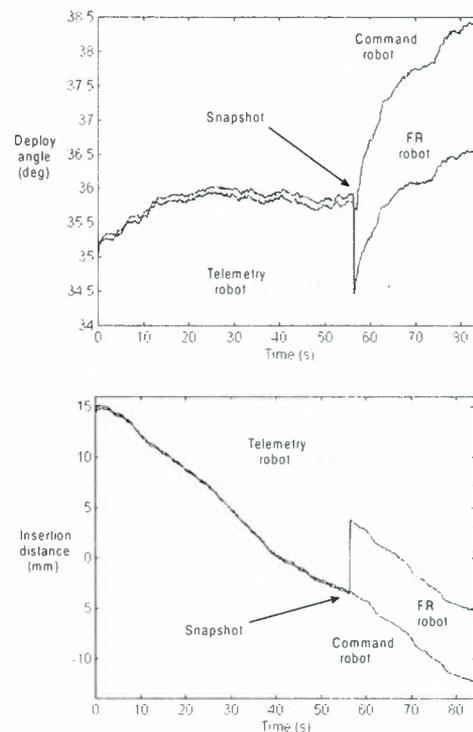


Figure 9 Application of a snapshot during disassembly of the TJ. a) stowing angle and b) insertion distance

close related to where the actual robot is. He continues with the movement of the FR robot and the computer calculates the command relative to it. Eventually the FR robot will depart again from telemetry and it is the operator's decision when to apply another *snapshot* to superimpose telemetry and FR robots again.

This procedure is seen in Figure 9 during disassembly of the TJ using a virtual constrained path (between 35.5 and 36.5 deg). Graphics for both the deployment angle and insertion distance are displayed.

Due to the TJ spring the swing angle of the TJ tends to go to the stowing side. Also, because of friction there is a misalignment in the insertion distance. The operator then performs a *snapshot* to feel where the on-board robot really is. After the *snapshot* the virtual forces guide the FR robot again to the right position to finish extraction and for subsequent insertion. The telemetry robot is brought along with it.

A *snapshot* takes almost no time, although it has to be applied off-line so that the change in virtual forces before and after do not confuse the operator. It can be done as many times as the operator considers necessary. With a little practice the operator soon learns how to take advantage of this novel system.

In the stowing of the TJ (Figure 8) one *snapshot* at the beginning was enough to keep the FR robot as same as the telemetry robot during the entire task. In contrast, several were needed during assembly.

5.5 Probing and model correction

Telemetry force is of great importance because the high flexibility of the arm renders telemetry position calcula-

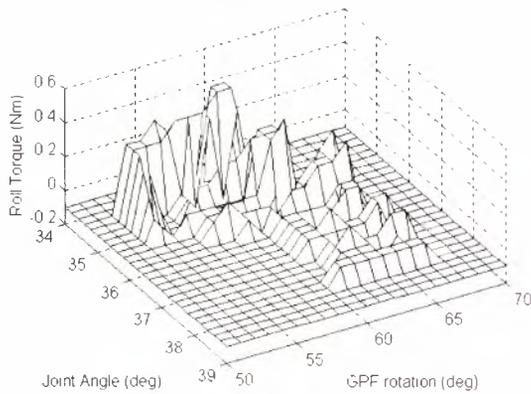


Figure 10. Results of probing the environment to look for the right joint angle for insertion.

tion inaccurate. Using an FR hand controller it is sensible to try to use telemetry force in some way.

We studied several possibilities and the most promising one turned out to be the use the force telemetry to create an on-line map of the environment. We called this the 'probing' approach.

To create a static map of a dynamic magnitude is extremely complex, if not impossible. Furthermore, the poor repeatability and high flexibility of the ETS-7 robot arm would add even greater difficulty.

Instead we decided first to create a map of virtual forces based on the library of constraints, and then update that map using the telemetry force. The development of a general algorithm for more DOFs is still the subject of research, but we have successfully applied this idea to a specific case.

Consider the TJ assembly. One of the main difficulties lies in finding the right deploying angle for insertion. There are inaccuracies in the model and telemetry information cannot be trusted hundred percent. It is then

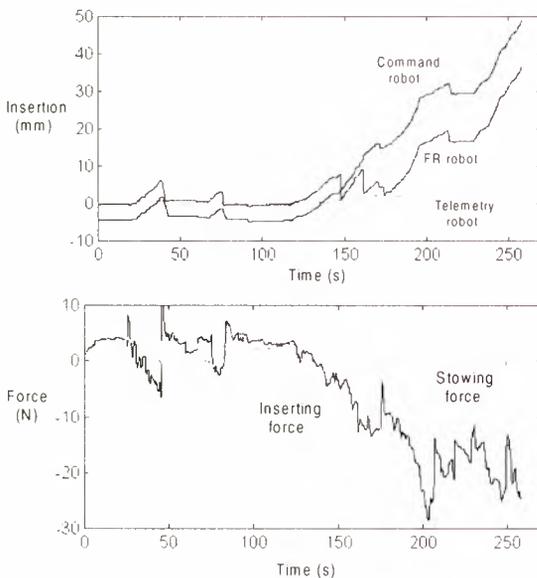


Figure 11 Results of TJ assembly: a) robot models and b) telemetry force.

interesting to probe for the assembly 'hole'.

The operator spends some time moving the robot according to a specific procedure to analyze the environment. The procedure consists in moving slightly in and out along the insertion direction while moving in the deployment one.

When the GPF is rotated a peg comes out of the AJ for insertion. This special feature allows us to use the roll torque as a measure of the contact between the TJ peg and the TJ receiver.

Figure 10 shows the experimental results of probing done with the TJ to look for the right deployment angle for insertion. It is seen that the right angle is around 36 deg, that is, where GPF rotation (extraction) is maximum with less torque.

This map is superimposed with the map of virtual forces already generated to guide the operator through the assembly. Figure 11 shows the final results for both the three robots and telemetry force during successful assembly of the TJ using this method.

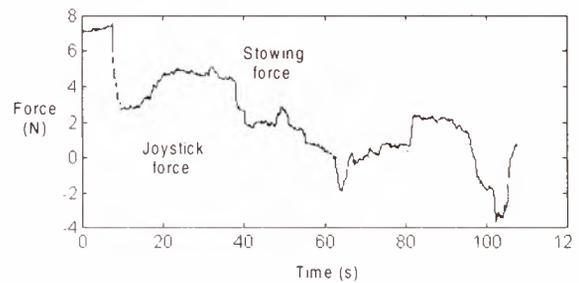


Figure 12. Stowing force and indirect joystick force during TJ assembly using indirect FR teleoperation

5.7 Indirect Force Reflection

We have also performed experiments of indirect FR. Indirect FR means that the telemetry force is reflected on a the hand that is not performing the task.

We have performed the TJ assembly 3DOF task using two joysticks. The right joystick was used to control robot's tip YZ movement. One DOF of the left joystick was used to control the third DOF of the task (the roll axis) while the other DOF reflected the real telemetry force acting on the stowing direction. The value of this force had previously been ruled out as crucial to accomplish an adequate insertion. It had to be under 4 N. Its value is very important to inform the operator where the robot is during insertion.

With indirect FR the operator could *feel* the force to keep it under the 4 N threshold (Figure 12). A non-linear function for FR was employed to warn the operator of the increasing force.

Therefore, indirect FR can be very useful for some tasks. The force reflected does not has to be a one-to-one value of the telemetry force. It is interesting to consider transformations that yield a significant force pattern in 1 or two DOF for the operator.

6. CONCLUSIONS

This paper has presented what is, to our knowledge, the first extensive application of FR for the ground teleoperation of a space robot.

It has been demonstrated that despite time delay there are ways in which FR can be successfully used to improve the performance of the operator.

It has been experienced several times during experiments that continuous teleoperation is extremely important to allow a rapid reaction to unexpected circumstances, or for trying different approaches in a very tight time frame. FR supports this mode operation by increasing its safety and speed. Moreover, it can simulate computer control having the operator in the loop.

On the other hand it is important to consider that FR is by no means a global solution. It should be considered as an important aid for the operator that can easily be combined with visual aids without overloading the operator's decision capability.

Also, it is necessary to study each task and application carefully to decide what is the most practical way to use FR to improve the overall performance.

In the experiments presented in this paper, FR is almost completely used based exclusively on a CAD model of the environment. We tried to overcome this fact by building a force map of part of the environment, although the solution adopted was specific to the task and by no means complete.

This does not mean that FR can only be used combined with a CAD model. For example, FR can be easily and intuitively used to present the operator the output of a guiding system which relies only on telemetry and not on a CAD model, like we did with the one described in [1]. Many other ways of using virtual FR can be thought of depending on the task, objectives and circumstances, being a promising field for new research and new applications.

7. ACKNOWLEDGMENTS

Special thanks are due for the following people that contributed in one way or another to make this work possible and successful: M. Nohmi from NAL, T. Yoshida, H. Ueno and Y. Fukase from Shimizu Co. We would also like to give our thanks to NASDA and its ETS-7 operation facility personnel for their total and outstanding support during the experiments.

8. REFERENCES

[1] S. Wakabayashi, "Generalised Visual Aid for Direct Teleoperation Applied to ETS-7 Truss Manipulation Experiment", *i-SAIRAS'99*, Noordwijk, The Netherlands, June 1999.

[2] M. Nohmi et al., "Truss Deployment by ETS-7 Robot Arm using Force Control", *16th Int. Symposium of Automation and Robotics in Construction*, Madrid, Spain, September 1999.

[3] K. Matsumoto et al., "Teleoperation Control of ETS-7 Robot Arm for On-Orbit Truss Construction", *i-SAIRAS'99*, Noordwijk, The Netherlands, June 1999.

[4] B. Hannaford et al., "Performance Evaluation of a Six-Axis Generalized Force-Reflecting Teleoperator", *IEEE Trans. on*

Systems, Man and Cybernetics, Vol. 21, No. 3, May/June 1991.

[5] T.B. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis", *IEEE Trans. on Robotics and Automation*, Vol. 9, No. 5, October 1993.

[6] L.F. Peñin, "Teleoperation with time delay. A survey and its use in space robotics", *NAL Internal Report (Draft)*, (NAL), Tokyo, Japan, to appear July 1999.

[7] L. Conway, R. Volz and M.W. Walker, "Teleautonomous Systems: Projecting and Coordinating Intelligent Action at a Distance", *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 2, April 1990.

[8] G. Hirzinger, B. Brunner, J. Dietrich and J. Heindl, "Sensor-Based Space Robotics -ROTEX and Its Telerobotic Features", *IEEE Trans. on Robotics and Automation*, Vol. 9, No. 5, October 1993.

[9] K. Brady and T. Tarn, "Internet-Based Remote Teleoperation", *Proc. IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998

[10] R.J. Anderson y M.W. Spong, "Bilateral Control of Teleoperators with Time Delay", *IEEE Transactions on Automatic Control*, Vol. 34, No. 5, May 1989.

[11] G. Niemeyer and J.E. Slotine, "Stable Adaptive Teleoperation", *IEEE Journal of Oceanic Engineering*, Vol. 16, No. 1, January 1991.

[12] T. Yoshikawa and J. Ueda, "Analysis and Control of Master-Slave Systems with Time Delay", *Proc. Int. Conference o Intelligent Robotics Systems*, IROS, 1996

[13] S. Lee y H.S. Lee, "Modeling, Design, and Evaluation of Advanced Teleoperator Control Systems with Short Time Delay", *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, Octubre 1993.

[14] M. Otsuka et al., "Bilateral Telemanipulator System with Communication Time Delay Based on Force-Sum-Driven Virtual Internal Models", *IEEE Int. Conf. On Robotics and Automation*, 1995.

[15] R. Oboe and P.Fiorini, "A Design and Control Environment for Internet-Based Telerobotics", *The International Journal of Robotics Research*, Vol.17, No. 4, April 1998.

[16] K. Kosuge and H. Murayama, "Bilateral Feedback Control of Telemanipulator via Computer Network in Discrete Time Domain", *Proc. International Conference on Robotics and Automation*, April 1997.

[17] J. Funda, T.S. Lindsay and R.P. Paul, "Teleprogramming: Toward Delay-Invariant Remote Manipulation", *Presence*, Vol. 1, No. 1, 1992.

[18] F.T. Buzan, "Control of Telemanipulators with Time Delay: a Predictive Operator Aid with Force Feedback", *Ph.D. Thesis*, MIT, Boston, 1989.

[19] Y. Tsumaki, Y. Hoshi, H. Naruse and M. Uchiyama, "Virtual Reality Based Teleoperation which Tolerates Geometrical Modeling Errors", *Proc. Int. Conference o Intelligent Robotics Systems*, IROS, 1996.

[20] L.F. Peñin et al., "Human Operator Modelling for Master-Slave Teleoperation with Kinesthetic Feedback", *IEEE ICRA'98*, Leuven, Belgium, 1998.

[21] K. Matsumoto et al., "Development of Truss Teleoperation Experiment by ETS-7 Robot", *International Conference on Intelligent Robots and Systems*, IROS'97, Toulouse, 1997.

[22] H. Ueno et al., "On-orbit Construction Experiment by Teleoperated Robot Arm", *14th ISARC*, Pittsburgh, June 1997

A WIRE HANDLING EXPERIMENT USING A TELEOPERATED ADVANCED ROBOTIC HAND ON ETS-VII

Nobuto MATSUHIRA*¹, Mokoto ASAKURA*¹, Yasuo SHINOMIYA*¹
Kazuo MACHIDA*², Kazuo TANIE*³, Hirotaka NISHIDA*⁴, Hiroyuki BAMBA*⁵
and Kenzo AKITA*⁶

*¹Toshiba Corporation, 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 210-8582

*²Electrotechnical Laboratory, MITI, *³Mechanical Engineering Laboratory, MITI

*⁴Fujitsu Ltd., *⁵Aoba Sangyo Co., *⁶Institute for Unmanned Space Experiment Free-Flyer

ABSTRACT

This paper describes the result of the wire handling experiment by the advanced robotic hand (ARH) on ETS-VII. The ARH, a small robot system with multi-sensory hand, has been developed by MITI-ETL to execute precise tasks under extravehicular conditions in space in an ETS-VII robot experiment. The wire handling task has been proposed for the ARH experiment as one of such delicate tasks. It requires teleoperation because wire deformation is difficult to estimate and needs operator's judgement. A new control system has been developed to cope with the teleoperation under time delay and limited information for wire handling task using the ARH. On July 23, 1998, the experiment was carried out by the teleoperation from Tsukuba Space Center to the ETS-VII through the data relay satellite TDRS of NASA. In the experiment, the contact force was controlled within 5 N and the wire handling was done successfully using the ARH under 6 seconds round-trip delay. The validity of our proposed control method was verified.

Key words: Space robot, Advanced robotic hand, Teleoperation, Time delay, Force control, ETS-VII

1. INTRODUCTION

Robotics is expected to play an increasingly

important role in space development. In the engineering Test Satellite VII (ETS-VII) launched in November 1997, many robot experiments have been executed with the aim of realizing the future automation and robotics of the extravehicular activities (EVA) [1]. Besides robot experiments of NASDA, other institutes participated in the experiments such as CRL, NAL, and MITI-ETL of Japan. The ARH experiment has been proposed by ETL [2]. The system has the ability to execute precise manipulation using a three-finger hand and autonomous control using multiple sensors. The robot system is different from the NASDA robot arm system. The wire handling manipulation (HWM) has been proposed one of the experiments of the ARH system [3]. Electric wire handling is considered to be an EVA task. The HWM experiment, involving a precise task for which the judgement of an operator is required for handling the flexible wire, is performed by means of teleoperation from the ground station. Features of the space experiment are time delay, the limitation of control information, and the visual range dependent on the camera location. To cope with these features, a novel control system has been proposed, consisting of a computer graphics (CG) simulator, a hand controller with 5 degrees of freedom (DOF), and a control algorithm for teleoperation using the concept of a virtual operator. This paper describes the results of the space experiment performed in July 1998.

2. HWM EXPERIMENT

The wire handling experiment (HWM) is illustrated in Fig.1. The purpose of the HWM experiment is to show the feasibility of executing precise tasks by teleoperation. The experimental setup consists of a wire, a peg and hole, two pins, and a fixed pin on the task panel. The end of the wire is fixed on the task panel, and the other end of the wire is connected to the peg and wound around the peg. The peg is inserted into the hole and locked by the latch lever using spring force. In the initial condition, the wire is looped around one pin. The HWM experiment is as follows: the ARH grasps the peg, the lock mechanism is released and the peg is extracted from the hole, the wire is moved to loop around another pin, and finally the peg is inserted into the hole again. The wire handling procedure is shown in Fig.2.

The diameter and the length of the peg are 14 mm and 40 mm, respectively. The clearance to the hole is about 0.05 mm. The wire consists of two twisted copper conductors with ETFE coating. The diameter of each conductor is 1.3 mm. The thickness and color are important with respect to recognition through cameras and the flexibility is considered for the manipulation. These conditions

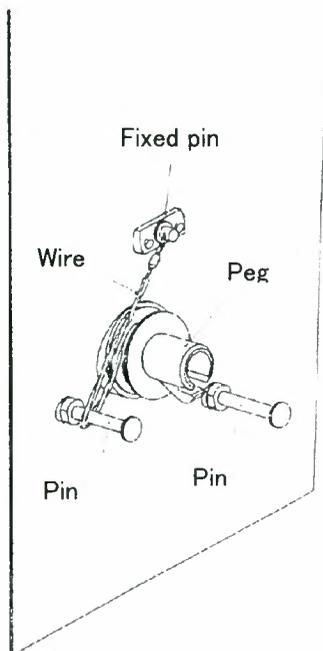


Fig.1 Experimental Setup

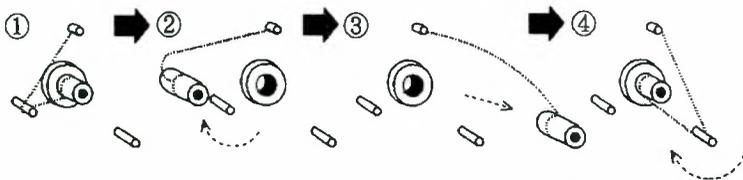


Fig.2 HWM procedure

were determined by using the test-bed at TSC.

The important small tasks in this experiment are to loop around the pin and to lock the peg to the hole certainly. Because if the peg is floating and the wire is not looped around the pin, such situation may disturb the other ARH experiments.

3. ETS-VII AND ARH TOTAL SYSTEM

The total system containing the ETS-VII and ARH is shown in Fig.3. The control station at Tsukuba Space Center (TSC) consists of a graphical display, a hand controller, and monitors. The control commands are transmitted to the ARH system on the ETS-VII through the NASA data transmission satellite TDRS. The ARH is a small robot system with a three-finger multi-sensory hand. The dimension of the ARH system is 500-480-480 mm. The mini-arm has 5 DOF and the hand has 3 DOF with three fingers. For this experiment, a compact hand controller has been developed to control 5 DOF of the ARH, and the software of the simulator which supports the operator and the control algorithm have been specially developed for the HWM experiment.

3.1 Monitors of hand-eye cameras

Two monitors for hand-eye cameras are available for the HWM experiment: hand-eye cameras on the NASDA robot arm and the ARH.

The hand-eye camera mounted on the NASDA robot arm shows the general view of the experimental setup. Since the motion angles of the NASDA robot arm are limited and the pass to the monitoring position may interfere with other experimental setups on the ETS-VII, a suitable position had been checked in advance using the test-bed at TSC and CG simulator. The operator does not watch the monitors during the operation because the time delay would be a source of confusion regarding the operation.

On the other hand, the hand-eye camera of the ARH can show the close-up view on the task panel. But the camera cannot show the gripping position because it is offset from the center of the ARH fingers. This camera is also used to confirm the situation precisely after the completion of each small task and execution of the task.

3.2 Simulator display

The simulator display shows the computer graphics of the experimental setup and three fingers of the ARH. The two types of the finger CG models, virtual finger and actual finger, are both

shown at commanded position from the ground and the actual position. When the operator moves the hand controller, the virtual finger moves to the commanded position first, and after that the actual finger follows the position with time delay. Thus, the operator can estimate the ARH position. Furthermore, the translational force acting on the ARH arm and the gripping force on the fingers are also shown graphically so that the situation is known well. Owing to this information, the operator can easily operate the ARH using the virtual view from the suitable angle without the influence of time delay and poor monitoring.

3.3 Control method for teleoperation

The proposed control system is shown in Fig.4. These software are implemented in the on-board computer of the ARH. Here, the virtual operator, consisting of low proportional control gain and limiter, is the algorithm to generate the desired force value from the ground commands respecting the ARH position. The force control is realized by the model-following control method, i.e., the position trajectory is modified by the desired force and the contact force. The dynamic model of the arm is composed of mass and damper. The ARH moves to the position where the desired force and external force are balanced, and the generated force is limited within 5 N in the HWM experiment.

Owing to this control method, the ARH cannot generate the abrupt motion in the case of a big position error between the commanded position and the actual position, and the ARH cannot generate a large force even in the contact condition. It realizes a safe operation and is suitable for the control of the space robot. Compared with the conventional method, a move-and-wait strategy, our proposed method can be called an extended move-and-wait strategy, because the operator naturally changes the operation modes such as fast operation for rough motion and precise operation for precise tasks. The load of the operator is reduced. The concept of a virtual operator raises the possibility of artificial intelligence development. In this experiment, the software of the virtual operator had to be simple for the limited capacity of the on-board computer. In Fig.4, these double line blocks were newly developed for the HWM and they have to be added to the ARH system.

4. RESULT OF HWM EXPERIMENT

4.1 Experimental pass for HWM

The ETS-VII was launched in November 1997, and the HWM experiment was carried out on 23 July, 1998. The experimental pass is shown in Table 1. Initially, five passes are allocated for the HWM: the camera setup of the NASDA robot arm, HWM program upload, operations, and the storage of the ARH. HWM operations use three passes: the confirmation of the ARH basic function including the pin-lock mechanism and calibration between the CG simulator and the actual environment, wire handling, and the recovery task, if needed. One pass is about 40 minutes containing the line connection procedure of NASA-NASDA, and about 20 minutes can be used for the HWM experiment. In the experiment, since data transmission line was not connected to NASA in the first three passes for preparation, an additional line and additional passes were used and first two passes were executed in parallel in the same pass. As the experiment proceeded successfully, the recovery of the HWM task was not needed.

4.2 Experimental result

Fig.5 shows the overview of the experiment as four segmentations. The whole view from the hand-eye camera of NASDA robot arm is shown at upper right in Fig.5. It is important to confirm the wire position and condition. In the close-up view of the experimental setup from the ARH hand-eye camera at upper left, the view clearly shows the wire and lock condition. In the view of the simulator display at lower left, the setup and two types of fingers are shown. At the control station at TSC at lower right, the operator works the hand controller while watching the CG display. After the execution of each small task, the operator has to confirm the condition by camera views, because it is not possible for the wire to be shown in the CG simulator.

Fig.6 shows the wire conditions before and after the operation given by the ARH hand-eye camera. After the operation, the wire is looped around the other pin and the peg is inserted and locked fast judging by the location of the latch lever.

4.3 Position trajectory during operation

Fig.7 shows trajectories of both commanded position and actual position of the gripping center of the ARH during operation. The trajectory shown in dark is the actual position of the ARH, and the trajectory shown in light is the commanded position from the ground. This shows the features of our proposed control method clearly. In the conventional method, the robot correctly follows the commanded trajectory that the operator made in the graphics display. However, in our method,

the ARH cannot follow accurately the commanded trajectory, because the virtual operator accommodates the trajectory depending on the on-board situation such as commanded force or position. It is quite different from the conventional method that simply plays back the taught trajectory.

Fig.7 also shows the force vector on the trajectory of the ARH position. The maximum length of the vector is 5 N. It shows that the contact force is controlled under 5 N correctly by the proposed control method, and the wire is found to be looped around the pin from the force vector forward to the center of the pin. In the region where the position error is bigger, the operator gives sufficient tension to the wire to reach the hole position. It is a kind of operation skill. Before the insertion of the peg, the direction of the force vector is downward. It means that the peg is pushed to the task panel.

The transmission time delay is assumed to be about 6 seconds, and the experiment time for the confirmation of ARH basic function and the wire handling are 16 min. 46 sec. and 22 min. 27 sec., respectively. The experiment was done within the allowable time for a pass.

5. CONCLUSIONS

We proposed the wire handling experiment as a

precise task using the ARH by teleoperation from the ground. A teleoperated control method has been newly developed which is effective despite time delay and limited information for the space robot. In the ARH experiments, the wire handling experiment was carried out successfully under the condition of round-trip delay time of about 6 seconds, controlling the contact force within 5 N. From the results, the validity of the proposed control method was demonstrated. This is a step toward performing precise tasks by teleoperation of robots.

The authors wish to thank the technical staff of the NASDA robot system for their cooperation.

REFERENCES

- [1] M. Oda, T. Inagaki, M. Nishida, K. Kibe, F. Yamagata, "Design and development status of ETS-7, an RDV and space robot experiment satellite," i-SAIRAS, 1994
- [2] K. Machida, T. Mikami, S. Komada, K. Akita, "Precise EV Robot: Flight model and telerobotic operation," pp.1550-1557, IROS'96, 1996
- [3] T. Kotoku, K. Takamune, K. Komoriya, K. Tanie, N. Matsuhira, M. Asakura, H. Bamba, "Manipulating flexible parts using a teleoperated system with time delay: an experiment," pp.289-292, i-SAIRAS, 1995

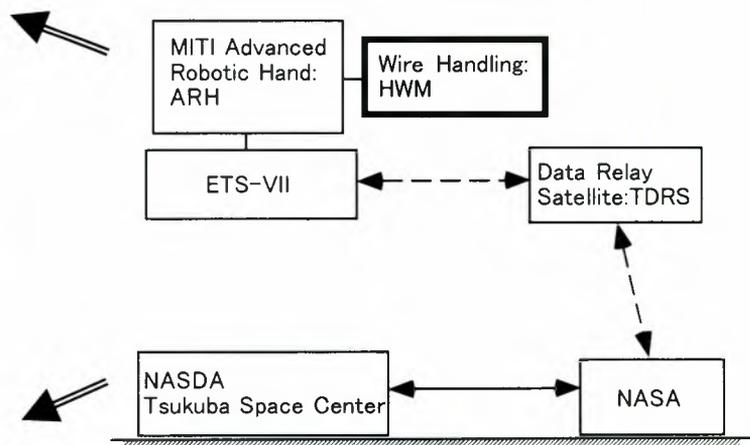
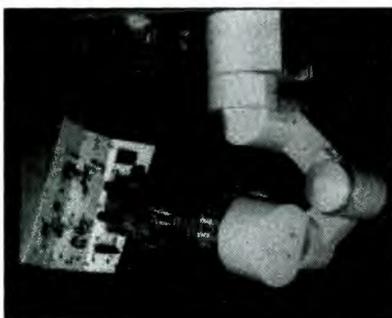


Fig.3 ARH experiment system

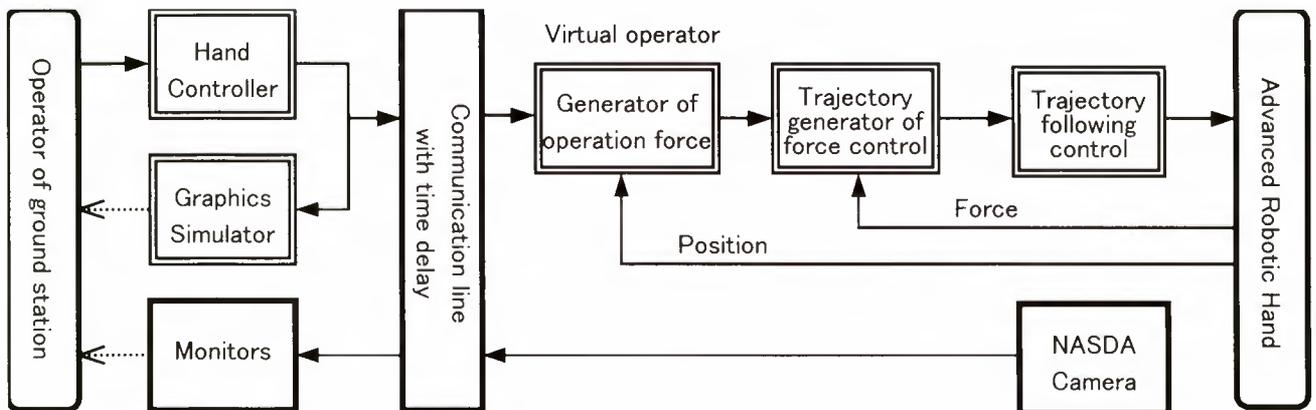


Fig.4 Proposed control system

Table 1 HWM experimental pass

Pass	Time	Plan	Result
1	3:03~3:45	NASDA Camera Setup	Not connected to NASA
2	4:45~5:27	HWM Program Upload	Not connected to NASA
3	6:28~7:10	HWM, ARH Basic Function	Not connected to NASA
4	8:11~8:53	HWM, Wire Handling	NASDA Camera Setup HWM Program Upload
5	9:55~10:37	HWM, Recovery Task ARH Storage	HWM, ARH Basic Function
6	11:37~12:19		HWM, Wire Handling
7	13:30~13:59		ARH Storage

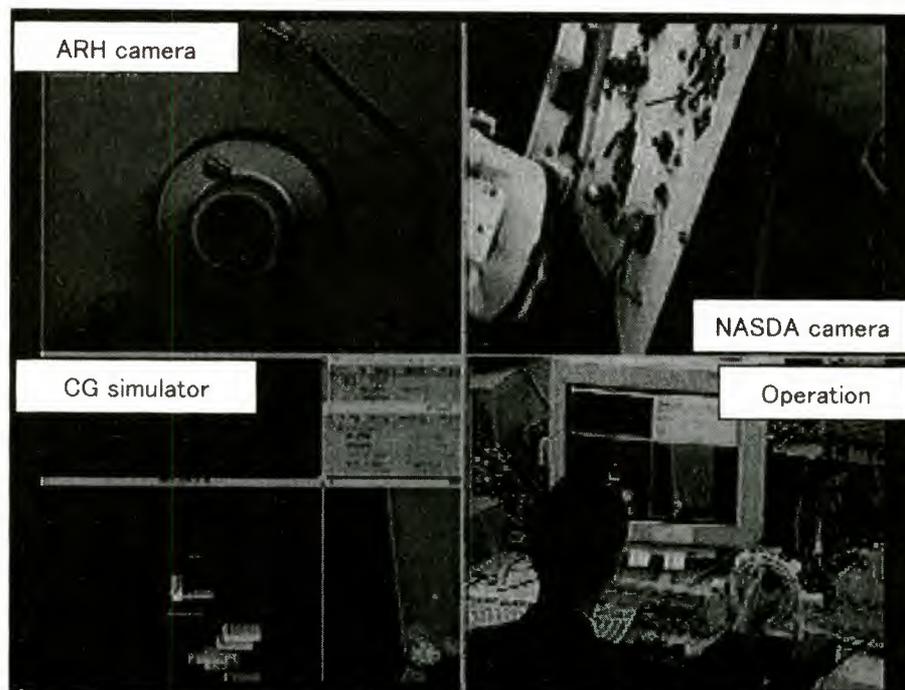


Fig. 5 Overview of HWM experiment



Before the operation



After the operation

Fig.6 Before and after the operation

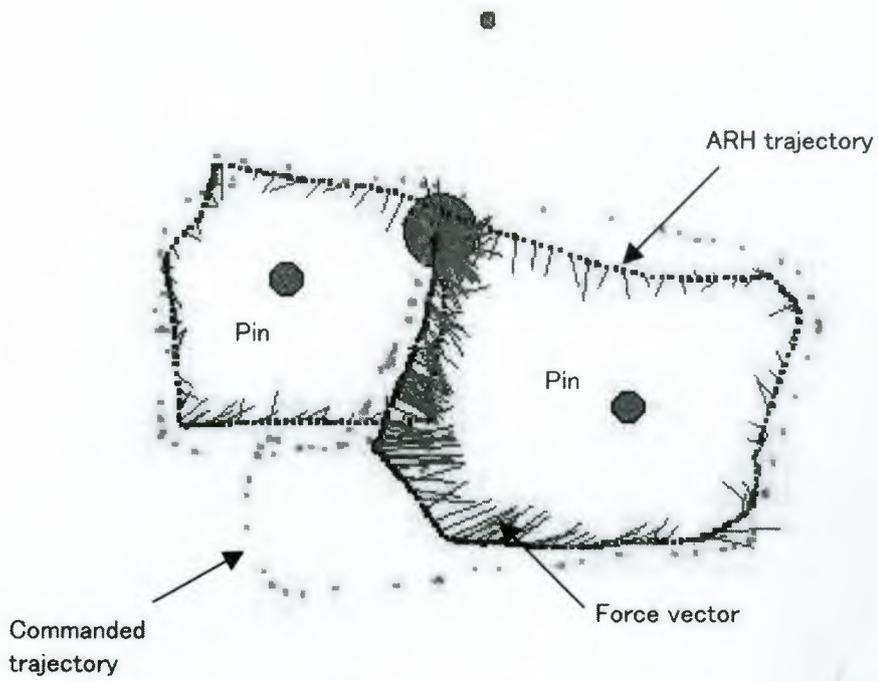


Fig.7 Trajectory of the ARH

GENERALIZED VISUAL AID FOR DIRECT TELEOPERATION APPLIED TO ETS-7 TRUSS MANIPULATION EXPERIMENT

Sachiko Wakabayashi, Kohtaro Matsumoto

Space Project and Research Center, National Aerospace Laboratory
7-44-1 Jindaijihigashi-machi, Chofu, Tokyo 182-8522, Japan
phone: +81-422-40-7131, fax: +81-422-40-3148, e-mail: waka@nal.go.jp

ABSTRACT

This paper reports the latest results of a newly developed visual aid system for direct teleoperation applied to the ETS-7 truss deployment experiment. This aid system does not depend on a designed model of the workplace. It introduces the "predictive force" to calculate the appropriate joystick input, and displays it to the operator in the joystick coordinate system to enable the operator easily follow the direction. This "predictive force method" is extended to automatic programming to make an efficient teleoperation system by combining the direct teleoperation and the program control.

1. INTRODUCTION

For the future space activities like International Space Station, teleoperation is considered one of the most needed technologies to reduce and supplement on-board operation helping the astronauts. And the tasks which teleoperation would substitute are expected to increase in quantity and become more complex.

If there is a precise designed model of workplace on the ground, the program control can perform better than the direct teleoperation by human operator. Still, we need a practical system for direct teleoperation, which effectively supports the operator to conduct the whole operation. This is partly because there is fairly big modeling error in space systems and partly because the program control can not handle unexpected situations.

Thus, we first developed and tested a new visual aid system for direct teleoperation to calculate and display the appropriate input to the operator without using a designed model, by introducing "predictive force".

On the other hand, if an aid system is designed based on the analytical algorithm, there seems to be no clear advantage in direct teleoperation. This is because the program control can do the same or even more precise work, if the next input is determined by calculation. The error of the program control is usually far less than that of the unstable human input. So next, we expanded the algorithm of the predictive force method into automatic programming to realize program control without a designed model.

The direct teleoperation is useful in coping with unexpected situations or making small adjustment during the operation, while the program control excels in

precisely following an algorithm. It seems that the best teleoperation system is a combination of the direct teleoperation and the program control. We think that a teleoperation system first should be equipped with an effective aid system for direct teleoperation, and next, the analytically determined operation should be replaced with program control.

The input device for direct teleoperation is also important. Direct input devices like joystick or master-arm are generally considered flexible and useful in teleoperation system compared with program control. And we adopted joystick as an input device for our truss manipulation experiment on the Engineering Test Satellite 7 (ETS-7). However, it seems that the direct teleoperation using those devices is not so capable without a proper aid system. This is especially true for dexterous operations and the main reason is the difficulty of making complex input. It is better to show the operator the input direction directly connected to the input device.

From the next section, we describe the newly developed aid system, its extension to program control, and their combined system, based on the results of the truss manipulation experiment on ETS-7.

2. TRUSS EXPERIMENT ON ETS-7

ETS-7 (Engineering Test Satellite 7), launched by NASDA Nov. 28, 1997 in Japan, had been developed to demonstrate two major missions, the rendezvous docking and the space robotics (Fig. 1). For the space robotics, it is the world first robotic-arm teleoperation experiment satellite.

NAL have participated the robot experiment along with other three national institutes. The basic robot systems of the satellite and the ground facilities, such as the arm, vision, communication, and the controller, had been developed by NASDA. NAL have developed our own experiment apparatus, TSE (Truss Structure Experiment apparatus), for the satellite and our own ground facilities related to tele-robotic research.

The ETS-7's teleoperation has two modes, the program control and the direct teleoperation control. We have mainly used the direct teleoperation control where the arm tip motion is controlled at 4 Hz directly from NAL's teleoperation facility. For the arm tip force control, the soft or rigid compliance control or the active limp control is used from the final capturing process and all over the handling tasks to avoid excessive force from the arm.



Fig. 1 ETS-7 (Engineering Test Satellite 7)

The TSE experiments have three components, the launch lock, the deployable truss, and the truss joint. The deployable truss (DT) has 10 degrees of freedom, and is a part of a triangle truss structure that is statically determinate and can be deployed and folded (Fig. 2).

The arm grapples the grapple fixture installed on DT and deploys it along a 3 dimensional spline curve under closed link arm control. The technical difficulty is to move the arm along the trajectory within suitable tip force and torque. The closed link movement along a strictly defined trajectory is the first operation for ETS-7.

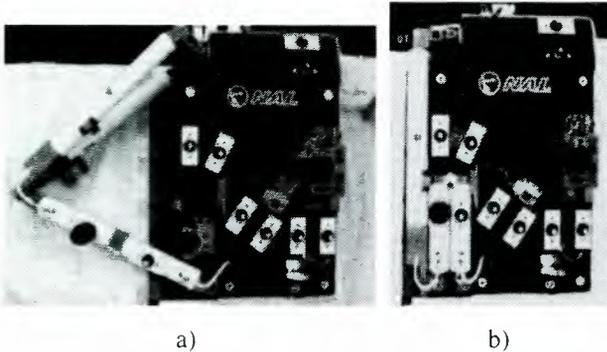


Fig. 2 TSE Deployable Truss
a) deployed position, b) stowed position

3. THE PROBLEM OF THE DIRECT TELEOPERATION OF DT

The trajectory of the deployable truss is a smooth three-dimensional spline curve, and the orientation of the grappling point also changes gradually along the trajectory. The handling of DT has shown that the direct teleoperation by joystick is almost useless without a proper operation aid. This is mainly because of the difficulty of tracing the complex deploying trajectory under communication time delay. It is so difficult to intuitively decide the next input that the operation tends to be a move-and-wait operation to confirm the result. Thus, the aid system should show the operator the proper input directions that are intuitively understandable.

In addition, the designed model of the trajectory is not reliable because of the modeling error caused by the launch impact or the thermal effect. Since

the modeling error of the space structures can usually be more than several millimeters, the use of the designed model is not accurate. Furthermore, considering the efficiency of the future space activities where most of the tasks are not repetitive, the use of the precise designed model for all the tasks seems to be unrealistic, because it requires enormous database and a heavy load of calculation. To realize a practical aid system, it should not rely heavily on a designed model.

4. VISUAL AID SYSTEM USING PREDICTIVE FORCE METHOD

We have developed and tested a new visual aid system for deploying the truss which calculates the appropriate joystick input on-line and shows it to the operator by indicating the directions the joysticks should be moved to.

This method uses only the past trajectory and the present status, without using a designed model of the trajectory, and theoretically estimates the current force executed to the truss. We call this theoretical force as the predictive force. The basic idea is to move the current command point toward the tangential direction of the estimated trajectory calculated from the past data. The appropriate input to apply the needed force for deploying should be in parallel with the tangential direction of the current point (tangential input). When the current command position is not on the tangential line, another input is needed in the vertical direction of the trajectory to release excessive force (force-release input).

As these two input directions are converted into the joystick coordinate system and shown to the operator to follow the direction, the operator easily handles the complex operation without doing move-and-wait.

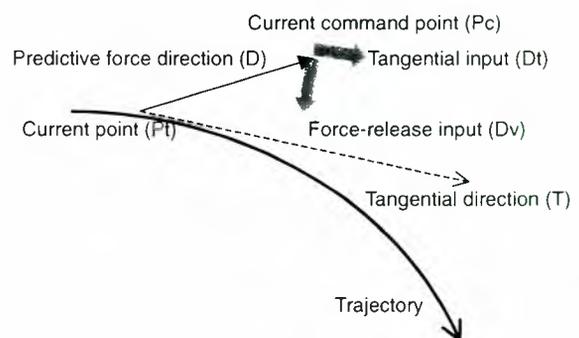


Fig.3 Algorithm of the Command Calculation using Predictive Force method

The algorithm of this aid system is shown in Fig. 3. In the below discussion, the bold symbols indicate vector value.

Fig. 3 shows the situation in the middle of the deployment. The robot is now at the current point P_t , and the current command point is P_c . The vector \mathbf{D} ($= P_c - P_t$) is the difference between the command point and the current point, which theoretically corresponds to the direction of the force. If the parameter of the compliance

control, f_p (the force produced when $|D|=1$), is known, $f_p \times D (=F)$ is the theoretical force vector applied to the truss by the robot arm. We define this theoretical force as the predictive force.

Since the trajectory from past to present is known, the tangential vector of the current point, T , can be calculated by some numerical algorithm. By moving the command point P_c in order for D to overlap with T , the force vector F turns to the tangential direction T and executes only the deploying force.

In other words, the tangential component of $D(F)$, which is in parallel with T , corresponds to the deploying force, and the orthogonal component of $D(F)$ corresponds to the excessive force. The aid system displays to the operator the two directions, the tangential input D_t that increases the deploying force and the force-release input D_v that decreases the excessive force.

P_t , P_c , and T should use the latest value. It is possible to use the current command and the delayed telemetry together, if the time delay is relatively small. But, if the time delay is large, the time difference between P_t and P_c , and the calculation of T should be adjusted.



Fig. 4 Aid Display for TSE DT Operation

Fig. 4 shows the aid display we have used for ETS-7 truss deploying experiment. The upper part of the display shows the overview of the TSE and the lower part shows the operation information. The three windows in lower right shows the force-torque sensor value and the image processing result. The lower left window is the down link image of the on-board hand camera. The two windows in the lower center are the direct input aid for joystick. The left window corresponds to the translation input and the right window to rotation input.

Fig. 5 shows the detail of the input aid. The right window is the translation aid. The bold bar is to indicate the trajectory's tangential direction D_t and the box is to indicate the force-release direction D_v . The thin bar indicates the current operator input. The display matches the physical coordinate system of the input device, joystick in this case, in order for the operator to

quickly follow the direction. The joystick axes are assumed to be vertical to the windows. If the bold bar points to the left, the operator is supposed to push the joystick to the left to execute deploying force. If the box is not around the window's center, the operator is supposed to push the joystick in the red box's direction to release excessive force. Thus, the ideal input should be the direction between the bold bar and the box like the one shown in Fig. 5. The vertical line in the right side indicates the third axis that is vertical to the window.

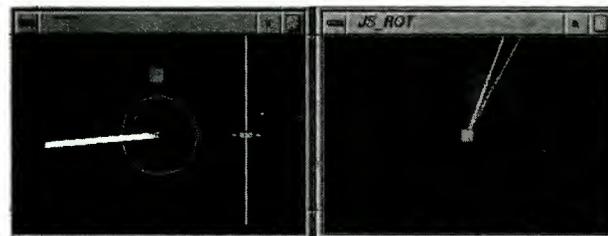


Fig. 5 Joystick Input Aid

The left window is for rotation input. The fan-shaped object shows the roll rotation allowance, because the DT's grapple fixture is not completely fixed and can rotate to latch or unlatch. If the two thin bars, one for current roll and the other for command roll, are within the fan, the excessive roll torque does not appear. The center box just shows the window center. The other box, almost in the center, shows the pitch and yaw input. They are calculated based on the difference between the current value and the command value, which corresponds to the theoretical torque. The pitch and yaw input is expected to reduce the difference.

The aid bars and boxes keep moving during the operation, and the operator follows them.

Fig. 6, 7 show the comparison of the telemetry trajectory and the command trajectory by the direct teleoperation using the visual aid system in stowing operation. Fig. 8 shows the translational force history. The coordinate system is shown in Fig. 9.

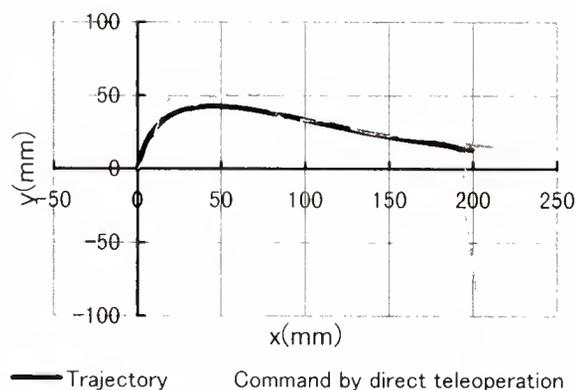
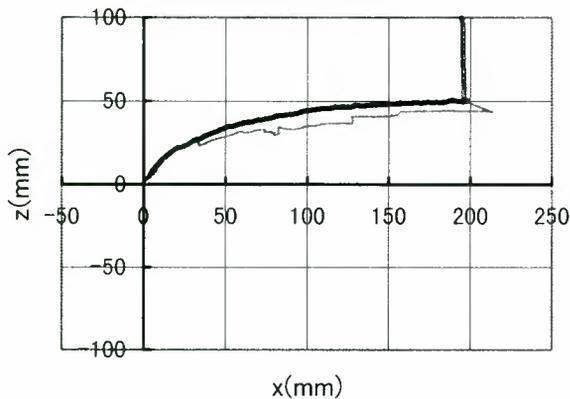


Fig. 6 Comparison of the Trajectory (x versus y)

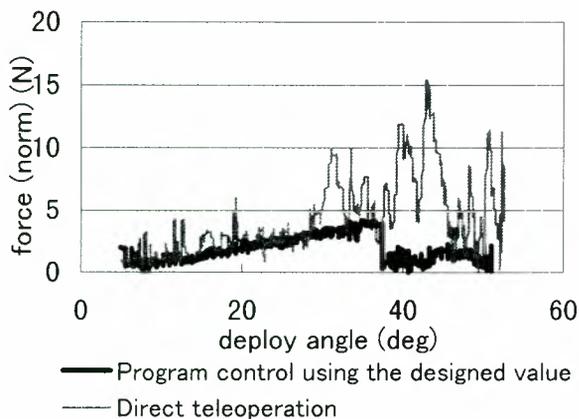
The command trajectory shows that the visual aid system successfully helps the operator to trace the complex trajectory. Though the translational force is

higher than that of the program control, the maximum force is around 15N and it is low enough for safe operation. The required operation time is about ten minutes. It is almost the same as the twelve minutes of program control, thus the move-and-wait operation was successfully avoided.



— Trajectory — Command by direct teleoperation

Fig. 7 Comparison of the Trajectory (x versus z)



— Program control using the designed value
— Direct teleoperation

Fig.8 Comparison of the Translation Force

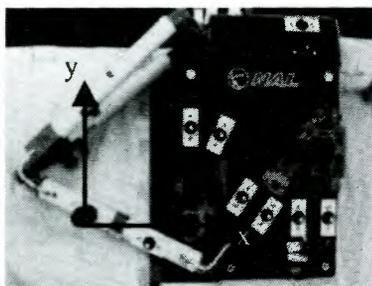


Fig.9 Coordinate System

5. PROGRAM CONTROL OF DT USING PREDICTIVE FORCE METHOD

In the above mentioned aid system, we introduced the idea of the predictive force. Here, we

extend the idea into automatic programming. Fig.10 shows the algorithm. The same symbols in Fig.3 are used. By specifying the maximum allowable force value ($|F_d|$), which the real force should not exceed during the operation, the ideal command point (P_d), which theoretically applies the maximum force to the truss, can be determined uniquely in the tangential direction T . The targeted D is determined as $D' = F_d / f_p$, and the targeted command point P_d is $P_d = P_t + D' = P_t + F_d / f_p$. The move of the command from P_c to P_d usually takes more than one command, and thus P_c approaches P_d gradually with several commands. P_d is kept updated every cycle of commanding using the newly calculated tangential direction T . In this way, the force can be controlled under the specified value $|F_d|$ during the whole operation.

This method performs better than the usual program control using a designed model of the trajectory, because it can avoid the excessive force due to modeling error.

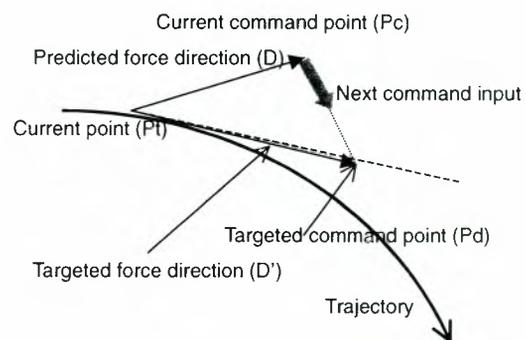


Fig. 10 Algorithm of the Program Control using Predictive Force Method

Fig. 11, 12 show the comparison of the command trajectory of deploying the truss by the program control using the predictive force method with the real trajectory. And Fig.13 shows the translation force history. In this case, the maximum force is specified as 10N and the tangential direction T is calculated by the least square method using the telemetry data from the past to the present. The coordinate system is shown in Fig.14.

The command trajectory by the program control shows that the predicted force method successfully generates the command trajectory without using the designed model. Though the command trajectory overshoot the real trajectory due to the need of executing the deploying force and the communication time delay, the translational force is almost the same as the program control until the middle of the operation and even far lower in the last part of the deployment. This is because the predictive force method compensated the modeling error.

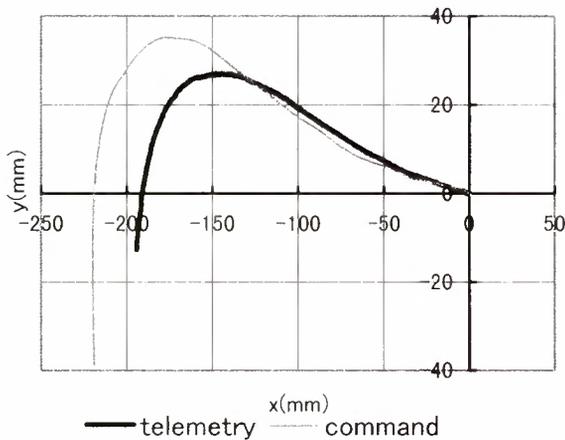


Fig.11 Comparison of the Trajectory (x versus y)

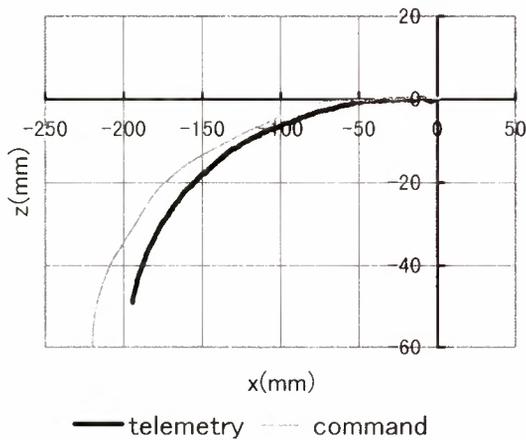


Fig.12 Comparison of the Trajectory (x versus z)

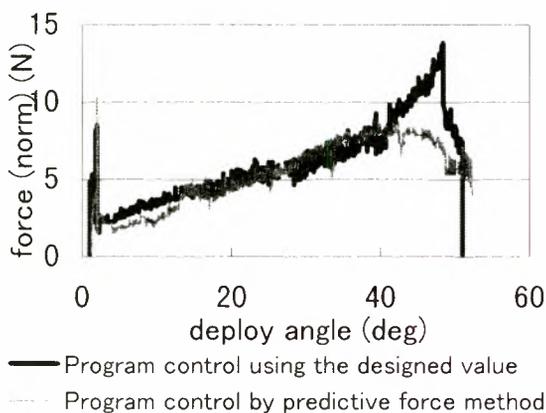


Fig.13 Force History in Deployment



Fig. 14 Coordinate System

6. COMBINED SYSTEM

Direct teleoperation and program control have their own advantage and disadvantage. Direct teleoperation easily handles the discontinuous operation and the unexpected situations, and program control makes stable and precise input in continuous operation. So, we combined them to make an efficient teleoperation system.

Our DT requires two tasks in addition to the 3D spline curve movement. They are releasing the lock by rotating the fixture at the beginning of the stowing and surmounting the temporal fixture at the stowed position. And there is also start/end operation. These discontinuous operations are easy to be specifically programmed but difficult to be generalized. So, we assigned these discontinuous operations to direct teleoperation and the continuous deploying/stowing operation to program control.

Fig. 15 shows the command and information flow of the combined system. This system consists of the direct teleoperation aid system and the program control system both based on the predictive force method. There is an input switching which selects the input or makes overwritten input.

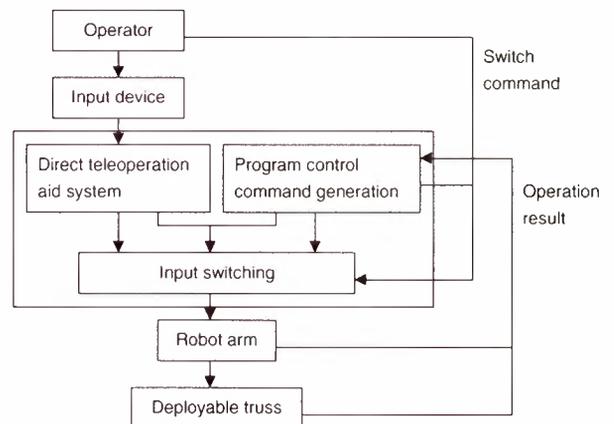


Fig. 15 Input and Information Flow of a Combined Teleoperation System

The operator starts the operation, and when the arm moves enough to start calculating the tangential direction, the operator stops joystick input and the

program control takes over the deploying/stowing operation. During the operation, the operator just monitors it and overwrites the program command if necessary. At the end of the operation, the operator takes over the operation and does the final adjustment. In this way, flexible and continuous operation is realized without relying heavily on a designed model.

For ETS-7 truss deploying experiment, we usually use this combined system. It is very efficient to share the operation by the operator and the program control.

7. CONCLUSION

To realize a practical teleoperation system, we first developed a direct teleoperation aid system, which enables the operator to conduct a complex operation like truss deployment. This aid system is constructed without a designed model and based on the newly developed predictive force method which uses the theoretical force to determine the input for applying the deploying force or releasing the excessive force. The ETS-7 truss deploying experiment shows that the aid system successfully supported the operator to conduct the operation.

Then we extended the predictive force method into program control, which controls the force under the specified value during the operation. This program control makes more precise and stable input than the operator does. The calculation based on the predictive force method is simple and for general use.

Finally, we made a combined system of the direct teleoperation and the program control. In this combined system, the direct teleoperation handles the discontinuous operation like start/end operation or latch operation, and the program control handles the continuous deploying/stowing operation. This system showed great efficiency and has the possibility to be applied to general tasks, because it does not use a designed model and the calculation load is low.

To farther improve this system, the algorithm should take the communication time delay into account in estimating the tangential direction vector. We used the least square method for estimating the tangential direction vector. But, to take the communication time delay into account this method might be inappropriate.

In addition, the teleoperation system must be able to handle unexpected situation efficiently. Through our truss operation, we have the impression that joystick is not necessarily the most appropriate device for direct teleoperation where subtle adjustment or handling of unexpected situation is required. To operate precisely in those situations, we might need other input devices than joystick. The input method itself is also the future subject.

REFERENCES

- [1]K.Matsumoto, etal, "Truss Structure Tele-Manipulation Experiment using ETS-7", Proc. 3rd I-SAIRAS-94, 1994.
- [2]S.Wakabayashi & K.Matsumoto, "Teleoperation with Additional Time Delay Mechanism for ETS-7 and JEM-EF", 47th IAF, 1996.
- [3]K.Matsumoto, etal, "Development of Truss Structure Teleoperation Experiment by ETS-7 Robot", Proc. I-SAIRAS-97, 1997.
- [4]K.Matsumoto etal, "Teleoperation of ETS-7 Robot Arm on On-Orbit Truss Construction", Proc. i-SAIRAS'99, 1999.
- [5]L.F.Penin, K.Matsumoto, S.Wakabayashi, "ETS-7 Robot Teleoperation through Virtual Force Reflection", Proc. i-SAIRAS'99, 1999.

Vision-based Robotics Control Experiment on ETS VII

Maarten Vergauwen, Reinhard Koch, Tinne Tuytelaars and Luc Van Gool
 ESAT-PSI, K.U.Leuven,
 Kard. Mercierlaan 94, B-3001 Heverlee, Belgium
 Phone: +32-16-32.10.64, Fax: +32-16-32.17.23
 vergauwe | koch | tuytelaar | vangool@esat.kuleuven.ac.be

Abstract

This contribution describes the vision-based robotic control (VBRC) experiments executed on the Japanese research satellite ETS-VII. The VBRC experiments were designed to enhance image quality, refine calibration of different system components, facilitate robot-operation by automatically refining the robot-pose and provide data for robot-calibration.

the arm hand camera (AHC) is mounted on the end effector of the robot arm. Each set contains two cameras, one primary and one redundant unit. Both can be utilized as stereo head with 60 mm baseline. Each camera records a grey level image with 668x480 pixel resolution with fixed focal length. The images are compressed with JPEG by a factor of 8.6 to yield a frame rate of 4 images per second on the video down-link. Two NTSC video channels allow access of two camera images simultaneously.

1 Introduction

The vision-based robotic control (VBRC) experiments were executed on the Japanese research satellite ETS-VII in conjunction with the Visual Interactive Autonomy Bi-Lateral Experiments (VIABLE) between ESA and NASDA.

The VIABLE¹ project is the first collaboration between ESA and NASDA with the aim to test the Interactive Autonomy (IA) concept for space robotics and to investigate advanced vision-based techniques for robot-control and calibration.

The ETS-VII satellite is equipped with a 6-DOF robot manipulator and two sets of cameras. The VIABLE experiments had access to a taskboard that allows several tasks to be executed by the manipulator. The taskboard contains a set of 3-point calibration markers with known 3D positions in the taskboard reference frame.

The ETS-VII onboard vision system consists of two sets of cameras. The arm monitor camera (AMC) is mounted on the first joint and

2 VBRC Experiments

The VBRC experiments were designed to

- enhance the image quality to allow better visual control,
- refine calibration of the system components (intrinsic camera parameters, eye-hand calibration) based on the calibration markers,
- perform on-line pose estimation procedures and guide the robot by automatically refining the robot pose,
- aid the operator during the experiments with visual clues using augmented reality techniques,
- provide material for post-mission robot calibration and testing of advanced methods for uncalibrated vision experiments.

3 Viable station setup

An important part of the VBRC experiments is the capability of the Vision Tools to allow operator intervention during execution of a vision

¹The VIABLE consortium consisted of the Belgian companies TRASYS-SPACE and SAS and the K.U.Leuven departments PMA and ESAT-PSI.

task. Image processing and computer vision is a process with possibly many sources for errors that can not all be modeled beforehand. Therefore a user-friendly interface was developed to assist the VBRC tasks. The interface allowed the operator to interact with the vision system to guide and help the automatic processing. While the human operator is very good at interpreting the scene and recognizing qualitative information, the vision system is good at precise quantitative measurements when given the appropriate input data.

To simulate and verify the VIABLE experiments a photo-realistic 3D model of the taskboard and the robot was constructed. This model served as reference for the IA path planning (in ROBCAD) and the VBRC visual simulator (in OpenInventor[1]). The model allowed the realistic visual simulation of all aspects of the experiments.

Verification of this simulation was performed with a mockup taskboard of scale 1:1. It contained all visually significant parts and served as a realistic testbed for the VBRC experiments.

4 Enhancing the image quality

A first set of experiments evaluated the impact of the imaging conditions in space (degradation of the images due to noise, image compression, direct sunlight, etc.) and derived parameters for image preprocessing. Analysis of the images that were taken for this purpose yielded a set of parameters for image-enhancement filters. Evaluation led to the following filter sequence:

1. **a median filter.** This non-linear filter effectively removes spikes and noise in the image but preserves the edges. It was chosen for its capacity to remove the ringing that typically occurs around the edges of an image when JPEG compression is used. Because the JPEG-ringing was quite severe, a window-size of 5 was used for most images.
2. **a binomial filter.** This low-pass filter smoothes the image to remove noise. It has the advantage over standard mean filtering that its frequency response has no ripples.
3. **a sharpening filter.** This unsharp-masking filter cancels the smoothing of the edges caused by the previous filter.
4. **radial distortion compensation.** This filter undoes the quite severe radial distortion of the images.
5. **aspect ratio compensation.** This procedure restores the original aspect ratio of the image which was changed due to the conversion to NTSC.

These preprocessing filters were applied to all incoming images before further processing.

5 Calibration

Online calibration is one of the crucial needs in the VIABLE project because no a priori calibration of the intrinsic camera parameters, the eye-hand, or robot pose is available. Only approximate calibration parameters could be obtained from the specification documents and from a limited set of images taken while the system was still on ground. We therefore designed a set of calibration experiments that verified and refined the approximate calibration from images during the flight segment. These experiments are explained in paragraph 5.2 and 5.3 but first a procedure for invariant-based feature extraction is discussed. This procedure is used in other experiments (like pose estimation or robot calibration) as well.

5.1 Invariant-based feature extraction

The online calibration relies on the 3-point markers and their given 3D position on the taskboard. One of the novel strategies in this task is the reliable extraction of image features and finding the corresponding 3D features. Correspondences are found robustly and completely automatically by exploiting viewpoint invariant relations. Two strategies were used.

1. If enough markers are visible, marker points are extracted as ellipse centers. Collinear points are found by computing the cross-ratio of all sets of 4 points. Lines are formed exploiting the viewpoint-invariant properties of the cross-ratio of 4 collinear points (see[2]). Figure 1 shows the marker points and lines found back by this approach.
2. If the camera is closer to the taskboard and only one 3-point marker is visible, a different approach is used. Because the ellipses

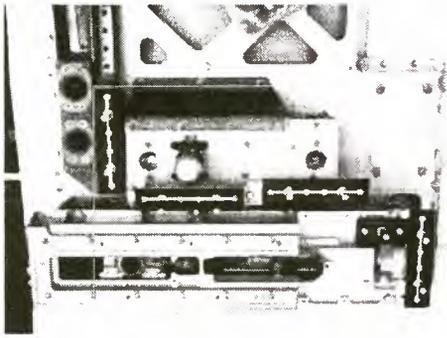


Figure 1: Marker points (found as ellipse centers) are grouped into lines using viewpoint invariant relations. The lines for each marker block are superimposed (in white) over the image for visual confirmation. Correspondences between the lines and marker blocks are also computed which yields 2D-3D relations.

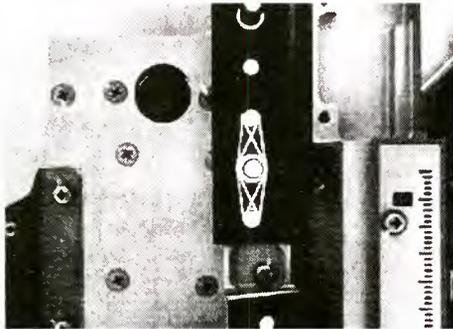


Figure 2: Common tangent lines of ellipses yield tangent points that are invariant under projective transformations. The ellipses and their common tangent lines are superimposed (in white) over the image for visual confirmation. This yields 2D-3D relations.

can be extracted more reliably in this case, we can use them (and not only their center points) to find enough 2D-3D correspondences. The fact is exploited that tangent points of two ellipses with a common tangent are invariant under projective transformations. In figure 2 the common tangent point and lines are superimposed over one of the 3-point markers.

5.2 Camera intrinsic calibration

Calibrating the intrinsic parameters of the camera is an important task in every application where measurements in the image are used to compute 3D spatial information like camera

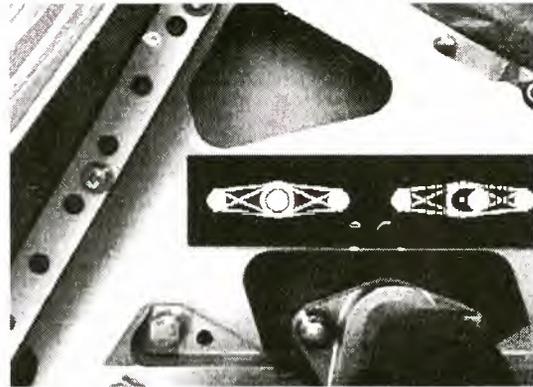


Figure 3: Setup for the eye-hand calibration experiment. The robot is touching the GPF and the AHC is above a 3-point marker. The common tangent points are found and based on these 2D-3D relations the camera pose is computed. This yields the eye-hand calibration of the camera.

poses or 3D reconstructions.

Based on two images of a calibration pattern that were taken by the cameras before the satellite was launched the intrinsic parameters of the cameras were computed.

During the flight segment images of the 3-point markers were taken by the AHC. These markers served as a calibration pattern. The result of the processing of these images was consistent with the precomputed values of both intrinsic parameters and radial distortion.

5.3 Eye-hand calibration

For robot guidance from images the relative transformation between the cameras and the robot tip frame – the *eye-hand calibration* – has to be known. A procedure was developed especially targeted towards the ETS-VII robot. When the robot executes the procedure to grasp the grapple-fixture (GPF), it comes into contact with the taskboard in a predefined position and orientation. In this specific pose, the cameras are approximately aligned with 3-point markers. These markers are exploited to compute the camera poses with the second technique explained in paragraph 5.1. Based on these computed camera poses and the fixed robot pose, the eye-hand calibration can be calculated. Figure 3 shows the setup of this calibration experiment.

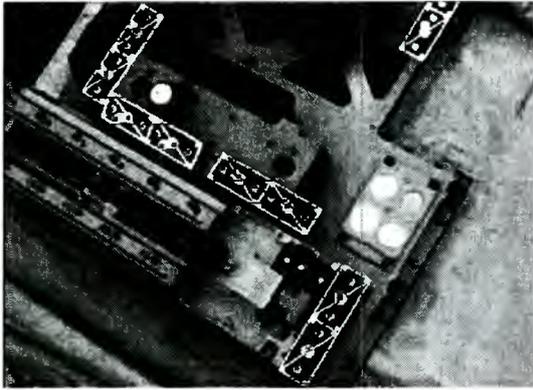


Figure 4: From automatically found 2D-3D relations the camera pose is computed. Parts of the model are superimposed over the real image and give a very good and intuitive verification of the calibration accuracy.

6 Pose estimation and on-line robot guidance

Several experiments concerning pose estimation and on-line robot guidance were performed during the flight segments.

6.1 Calculating pose from known markers

A first experiment consisted of calculating the robot pose from the known 3-point markers. The robot moved to a position where different markers were visible. Using the invariant relations described in paragraph 5.1, 2D-3D relations were found. These relations were the input for a robust camera pose estimation algorithm. An immediate verification of the current calibration status and the accuracy of the computed position could be supplied to the operator by superimposing parts of the given CAD-model with the actual images, using the calculated position. An example of this superimposition can be seen in figure 4.

A second step in this experiment consisted in moving the camera to a position much closer to one of the 3-point markers. The robot was intentionally positioned in a pose not perfectly above the marker. The second invariant method of paragraph 5.1 was used to calculate the camera- (and using the eye-hand transformation also the robot-) pose. Parts of the model were reprojected into the actual image to verify the calculation (figure 5). Using the computed pose, a

relative translation- and orientation-change was computed to position the robot perfectly above the 3-point marker (figure 6)

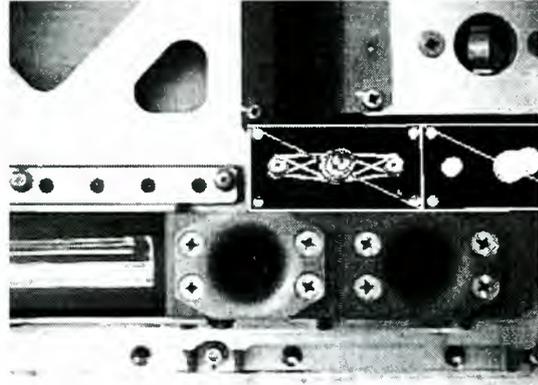


Figure 5: From automatically found 2D-3D relations the camera pose is computed. Verification of the result is possible by reprojecting parts of the CAD-model in the image.

6.2 Insertion of GPF into a hole

The ETS-VII robot has the possibility to attach a grapple-fixtured (GPF) to its end effector and insert it into different holes and a slider on the taskboard. Usually, positioning of the robot is done manually by the operator who uses the artificial markers as a visual clue. During the VIABLE experiments we showed that positioning could be done automatically using the image of the hole or slider only. This is especially important for the case of the slider because its exact position is unknown due to possible previous motions. Using an ellipse-fitting algorithm the hole or slider was extracted and the center point

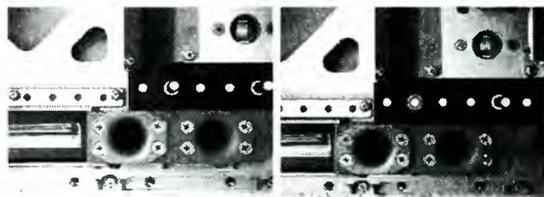


Figure 6: After a relative motion from the current position (left image), automatically computed by the vision-tools, the AHC ends up perfectly above the 3-point marker (right image). This is verified visually by the fact that the central marker tip is centered perfectly with the outer marker ring.

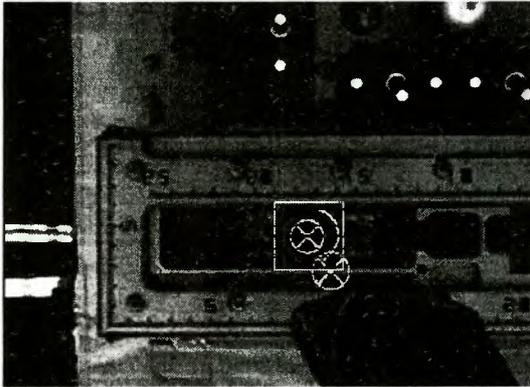


Figure 7: The vision system computes the current impact point of the GPF. The center point of the slider is extracted automatically and the relative movement is computed to position the GPF above the slider. The predicted impact point is shown to fit into the hole.

was found. This allowed the algorithm to compute a relative update of the current pose to position the GPF perfectly above the hole or slider. The image was augmented with the current impact point of the GPF (the point where the GPF would hit the taskboard if it were lowered from its current position) and the estimated impact point after relative motion. During operations the robot was deliberately mispositioned above both hole and slider. The algorithm managed to automatically update the pose to allow insertion. Figure 7 shows both current (misplaced) and estimated impact point.

7 Taskboard calibration and reconstruction

7.1 Calibration of 3-point markers

The 3-point markers on the taskboard are important for vision-based algorithms. The calculation of the camera pose from 2D-3D relations, found by the algorithm, needs the exact 3D coordinates of these markers. Experiments were designed which could retrieve this information.

Because a good estimate of the 3D coordinates of the markers was supplied to us by NASDA, a quick and easy check on the consistency of this data could be performed. We moved the robot over the taskboard to different positions for which different markers were visible in the images. We computed the cam-

era pose based on the markers and reprojected the given 3-point markers in the original image. The estimated mean reprojection error was below a pixel which confirmed the consistency of the marker positions.

The coordinates of the 3-point markers can also be explicitly retrieved from images. This is what was done in another experiment. Three different images, taken from three different poses, showed the same 3-point markers. Based on the given pose of the robot and the eye-hand calibration the camera poses were computed. Based on the identification of the markers given by the invariants, multiple-view matches were found. The markers could then be reconstructed in 3D by triangulation. The resulting data was consistent with the given 3D information (up to the accuracy of the reconstruction of 2.23 mm in x , 1.45 mm in y and 0.84 mm in z).

7.2 Taskboard reconstruction

In an advanced experiment we investigated novel techniques for calibration based on image data alone, without the need to know precise 3D calibration markers. Based on a sequence of images taken from different view points, one can obtain a metric calibration (up to a constant scale factor) of the cameras and the scene (see [3]). This technique allows the handling of a priori unknown objects with little calibration information. For these experiments we recorded predefined image sequences during the flight segment and evaluated these techniques in the post processing phase.

Figure 8 shows some results of the reconstruction of the slider-area of the taskboard. The figure shows views of the reconstruction without any manual refinement. In a post processing step it is easy to obtain reconstructions of parts of the taskboard by human interaction in the image only, using the computed depth data.

8 Robot calibration

8.1 Robot calibration

Robot calibration is a procedure which aims at improvement of the robot accuracy by modifying the robot positioning software, rather than changing or altering the design of the robot or its control system[4]. The procedure that is followed to obtain this goal is

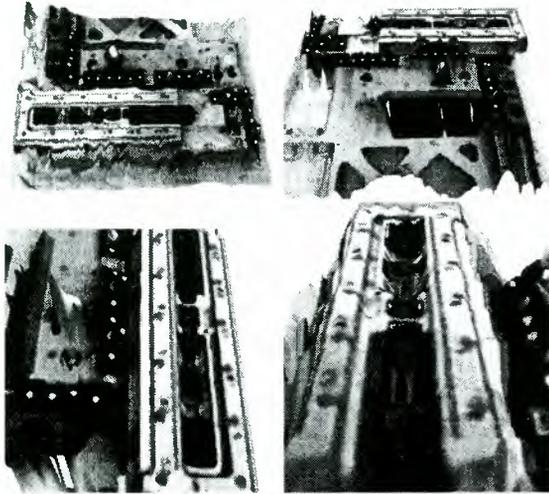


Figure 8: Different artificial views from the reconstruction of the slider-area of the taskboard.

- position the robot in different poses, trying to excite all possible modes,
- measure these poses with a measurement system,
- compute the difference between these measured poses and the pose computed from the joints telemetry by the forward kinematics model of the robot.

If all modes are excited sufficiently this allows to identify updates to be made to the current model.

Standard robot calibration procedures obtain pose measurements from external measuring systems. In the case of the ETS-VII robot no such system is available. Instead we computed the robot poses from the 3-point markers as explained in paragraph 6.1.

Since the taskboard on which all VBRC experiments were conducted is placed in one corner of the ETS-VII satellite, we could excite only a limited range of values in joint space. Ongoing evaluation will show if all joint offsets and link lengths can be identified or if computing a subset will yield better results.

8.2 Repeatability study

During the last temporal window data was gathered to perform a brief repeatability study of the robot. The arm was moved to a certain position

where three 3-point markers were visible. Then the robot was commanded to move a certain distance in a direction and return to the same position. This procedure was executed for three orthogonal directions and each time an image was taken by the AHC. Using the second procedure from paragraph 5.1 the camera- (and using the eye-hand transformation the robot-) pose was computed.

The result of the computations is displayed in table 1. The first pose is the reference pose. Poses 2, 3 and 4 are poses obtained after a motion in x , y and z direction respectively. For each pose the computed position of the camera is shown.

This data clearly shows quite some difference in the x and y direction. It indicates a repeatability-error of a few mm in the robot movements but because of the very limited amount of motions and repetitions no statistical conclusions can be drawn from this data.

pos	x	y	z
pos 1	415.496	152.332	435.001
pos 2	419.604	152.325	435.071
pos 3	417.232	151.359	434.834
pos 4	417.920	154.069	435.442

Table 1: Result of the repeatability study

Acknowledgments

We acknowledge support from the Belgian IUAP4/24 'IMechS' project.

References

- [1] Open Inventor Architecture Group, Open Inventor C++ Reference Manual, ISBN 0-201-62491-5, 1994.
- [2] J. Mundy and A. Zisserman, "Machine Vision", In J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science, Vol. 825, Springer-Verlag, 1994.
- [3] M. Pollefeys, R. Koch, M. Vergauwen en L. Van Gool, Metric 3D Surface Reconstruction from Uncalibrated Image Sequences, In *Proceedings SMILE Workshop (post-ECCV'98)*, LNCS 1506, pp.138-153, Springer-Verlag, 1998.
- [4] K. Schröer, Handbook on Robot Performance Testing and Calibration (IRIS-project), ISBN 3-8167-5200-4, Fraunhofer IRB Verlag, 1998.

Reactionless Manipulations and Proposal to ETS-VII On-Board Experiments

Kazuya Yoshida

Department of Aeronautics and Space Engineering, Tohoku University
Aoba 01, Sendai 980-8579, Japan
yoshida@astro.mech.tohoku.ac.jp

Dragomir N. Nenchev

Department of Intelligent Machines and System Engineering, Hirosaki University
Bunkyo-cho 3, Hirosaki 036-8561, Japan
nenchev@cc.hirosaki-u.ac.jp

Abstract

Space robots provides an interesting characteristics as under-actuated systems. If we operate a space manipulator without a good understanding on the under-actuated dynamics, the operational performance would be much degraded. However with a good understanding on the dynamics, we could operate the manipulator not disturbing the attitude or exciting the vibrations on its foot base. In this paper we propose such nice operation named as Reactionless Manipulations to be examined on the ETS-VII, a free-flying space robot developed by NASDA, currently flying in orbit.

1 Introduction

The interest toward complex robot systems is expanding for new application areas including space robots. A class of such robot systems are so-called under-actuated systems, characterized by the number of control actuators being less than the number of degree of freedom.

One typical example is a free-flying space manipulator (see Figure 1), in which the number of controllable joints are n in general but the number of system DOF is $n + 6$ including the position and orientation of the base body in the inertial space.

Another example of the under-actuated system is a dextrous manipulator arm mounted on a passive flexible base (see Figure 2). In literature, such a system is known under the name of long-reach manipulator [1], or flexible structure mounted manipulator system (FSMS) [2], and in this paper we simply say flexible-base manipulator.

This class of manipulator systems are regarded as a version of macro-mini manipulators. In space applications particularly, the dextrous manipulator sys-

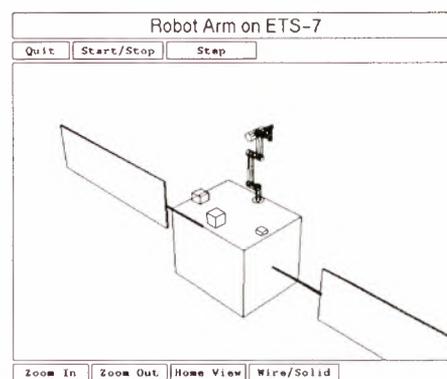


Figure 1 An example of Free-Floating Space Manipulator

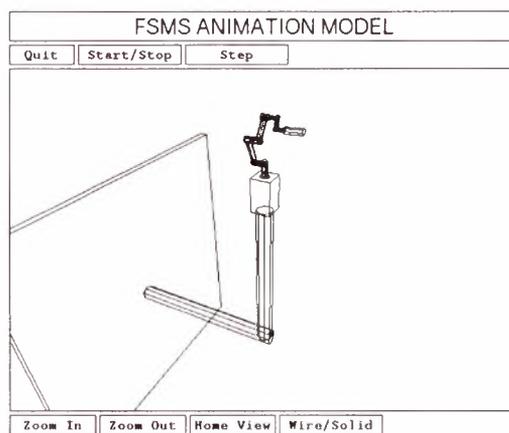


Figure 2 An example of Flexible-Base Manipulator

tem for the international space station, SPDM [3], mounted on the endtip of the space station remote manipulator system, SSRMS, is a good example. A manipulator system on the Japanese module, JEM-

RMS, also takes the macro-mini configuration. When all of active joints work simultaneously and coordinately, these systems are not an under-actuated system. But in the situation of fine positioning after coarse maneuver for example, the joints in the macro part are break locked and only the mini part will be operated most likely. The system is considered as an under-actuated system in such a situation, because the macro part behaves as a passive flexible structure.

One of major issues on the flexible-base manipulators is dynamic interaction between the motion of the dexterous manipulator, the mini part, and the vibrations of the flexible base, the macro part, due to the induced reaction. Such vibrations degrade the operational accuracy and settling time of the end point. One can find that the problem here is similar to the problem in free-floating manipulators at the point that the dynamic reaction of the manipulator arm induces the interactive motion in the supporting base, which is usually undesirable to dexterous operations. The difference between these two systems is that the manipulator base of a free-flying manipulator is a floating inertia, but that of a flexible-base manipulator is an inertia-spring-damper system.

The present authors formulated the dynamic equations for these under-actuated systems, paying attention to the commonality and difference between the free-flying manipulators and the flexible-base manipulators [4][5]. For the commonality, both systems are the first-order non-holonomic system, in which the coupling dynamics is expressed by the momentum equation as the first-order integral.

Getting an insight into this coupling dynamics, the authors discovered that there is a class of manipulator operation to de-couple the manipulator and base dynamics. A mathematical formulation to generate the decoupled dynamics is termed "Reaction Null-Space" (RNS) and such manipulation as "Reactionless Operation," or such motion trace as "Reactionless Path" [6]–[9].

If we operate the manipulator arm mounted on a floating or flexible base along a reactionless path, the manipulator generates zero reaction to the base, then no vibrations on the base and no degradation in the manipulation performance are expected.

The reactionless operation has been experimentally verified with a series of laboratory test bed of a flexible-base manipulator, named TREP-I and II [10]–[12]. However the operation has not been tested on a free-flying manipulator because a free-flying experiment is very difficult in one-G environment. But now, we have a real free-flying space robot in orbit, the robot is the Engineering Test Satellite (ETS-VII) developed and operated by the National Space Development Agency (NASDA), Japan.

This paper aims to propose the reactionless opera-

tion as one of meaningful flight experiments of ETS-VII, in terms for better understanding of the space robot dynamics and better performance of orbital manipulation, with less attitude disturbance and less fuel/time consumption for attitude maintenance.

2 Formulation of Dynamics

2.1 Equation of Motion in General Form

Let us begin with a general discussion considering a system which motion is described by n degrees of freedom of the generalized coordinate $\mathbf{q} \in R^n$ for *active* joints and m degrees of freedom of the generalized coordinate $\mathbf{p} \in R^m$ for *passive* joints. Now, define \mathbf{F}_q as active force/torque (twist) generated on coordinate \mathbf{q} , and \mathbf{F}_p as a passive force/torque exerted on coordinate \mathbf{p} . Also, define \mathbf{x} as a coordinate of a point of interest (the operational coordinate) composed by \mathbf{p} and \mathbf{q} , and let an external force/torque \mathbf{F}_{ex} be applied on \mathbf{x} . Hence, the applied external force/torque is decomposed as $\mathbf{J}_q^T \mathbf{F}_{ex}$ and $\mathbf{J}_p^T \mathbf{F}_{ex}$ onto each generalized coordinate using corresponding Jacobian matrices.

The equation of motion of such system is generally expressed as:

$$\begin{bmatrix} \mathbf{H}_p & \mathbf{H}_{pq} \\ \mathbf{H}_{pq}^T & \mathbf{H}_q \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_p \\ \mathbf{c}_q \end{bmatrix} = \begin{bmatrix} \mathbf{F}_p \\ \mathbf{F}_q \end{bmatrix} + \begin{bmatrix} \mathbf{J}_p^T \\ \mathbf{J}_q^T \end{bmatrix} \mathbf{F}_{ex} \quad (1)$$

where \mathbf{H}_p , \mathbf{H}_q , \mathbf{H}_{pq} are inertia matrices. \mathbf{c}_p , \mathbf{c}_q are non-linear Coriolis and centrifugal forces and they can include gravity forces if necessary.

Kinematic relationship among \mathbf{p} , \mathbf{q} and \mathbf{x} is expressed using Jacobians as:

$$\dot{\mathbf{x}} = \mathbf{J}_p \dot{\mathbf{p}} + \mathbf{J}_q \dot{\mathbf{q}} \quad (2)$$

$$\ddot{\mathbf{x}} = \mathbf{J}_p \ddot{\mathbf{p}} + \dot{\mathbf{J}}_p \dot{\mathbf{p}} + \mathbf{J}_q \ddot{\mathbf{q}} + \dot{\mathbf{J}}_q \dot{\mathbf{q}} \quad (3)$$

The above set of equations are commonly applicable for any type of under-actuated manipulator systems.

2.2 For Free-Floating Manipulators

Now, let us consider a free-floating system composed by a single robot base which is floating in the inertial space without any external force or torque, and a serial manipulator arm at which end point any external force/torque is not apply. For such a space manipulator, the equation of motion is obtained from Equation (1) by replacing the symbols as Table 1.

We then obtain the following equations:

$$\begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{bm} \\ \mathbf{H}_{bm}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_b \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_m \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} \quad (4)$$

Table 1 Symbol replacement for a free-flying manipulator

\mathbf{p}	\rightarrow	\mathbf{x}_b	: the position/orientation of the floating base
\mathbf{q}	\rightarrow	ϕ	: the joint angle of the arm
\mathbf{x}	\rightarrow	\mathbf{x}_h	: the position/orientation of the manipulator end point
\mathbf{F}_p	\rightarrow	$\mathbf{0}$: the external force/torque on the base
\mathbf{F}_q	\rightarrow	τ	: the joint torque of the arm
\mathbf{F}_{ex}	\rightarrow	$\mathbf{0}$: the external force/torque on the end point

$$\dot{\mathbf{x}}_h = \mathbf{J}_m \dot{\phi} + \mathbf{J}_b \dot{\mathbf{x}}_b \quad (5)$$

$$\ddot{\mathbf{x}}_h = \mathbf{J}_m \ddot{\phi} + \dot{\mathbf{J}}_m \dot{\phi} + \mathbf{J}_b \ddot{\mathbf{x}}_b + \dot{\mathbf{J}}_b \dot{\mathbf{x}}_b \quad (6)$$

For a space free-floating manipulator, there is no gravity exerting on the system, then the non-linear term becomes $\mathbf{c}_b = \dot{\mathbf{H}}_b \dot{\mathbf{x}}_b + \dot{\mathbf{H}}_{bm} \dot{\phi}_b$. Integrating the upper set of equation in (4) with respect to time, we obtain the total momenta of the system as:

$$\mathcal{L} = \mathbf{H}_b \dot{\mathbf{x}}_b + \mathbf{H}_{bm} \dot{\phi} = \text{const.} \quad (7)$$

This equation attributes an important characteristics to the free-flying system.

2.3 Reaction Null-Space

The ‘‘Reaction Null-Space’’ is a useful idea to discuss the coupling and decoupling of dynamic interaction between a manipulator and its base. The reaction null-space concept has its roots in the earlier work on free-flying space manipulator by Nenchev et al, where the Fixed-Attitude-Restricted (FAR) Jacobian has been proposed as means to plan and control manipulator motion that does not disturb the attitude of the free-floating base [6].

In Equation (7), if the base does not have any motion, say $\dot{\mathbf{x}}_b = \mathbf{0}$, then the momenta comes from only the manipulator motion. The partial momenta for the manipulator part becomes:

$$\mathcal{L}_m = \mathbf{H}_{bm} \dot{\phi} = \text{const.} \quad (8)$$

It is seen that a constant \mathcal{L}_m indicates no motion of the base $\dot{\mathbf{x}}_b = \mathbf{0}$, then no reaction force or torque is present to yield the base motion.

In case the number of degrees of freedom of the active manipulator joints n is grater than that of the base coordinate m , the solution for the manipulator operation to satisfy $\mathcal{L}_m = \text{const}$ is given by:

$$\dot{\phi}_c = \mathbf{H}_{bm}^+ \mathcal{L}_m + (\mathbf{E} - \mathbf{H}_{bm}^+ \mathbf{H}_{bm}) \xi \quad (9)$$

where $(\cdot)^+$ indicates pseudo-inverse, $\xi \in R^n$ is an arbitrary vector.

The component $(\mathbf{E} - \mathbf{H}_{bm}^+ \mathbf{H}_{bm})$ suggests the mapping onto the null space of the inertia matrix \mathbf{H}_{bm} and this inertial null space is termed ‘‘Reaction Null-Space.’’

In the special case when $\mathcal{L}_m = \mathbf{0}$, Equation (9) becomes much simpler as:

$$\dot{\phi}_{ns} = (\mathbf{E} - \mathbf{H}_{bm}^+ \mathbf{H}_{bm}) \xi. \quad (10)$$

As long as we operate the manipulator using the joint velocities given by (10), no reaction force or torque is generated on the base, therefore no reactive motion or vibration is oscillated in the base. If integrable, the integration of (10) yields ‘‘Reactionless Paths,’’ the trace of the manipulator motion which does not excite the base motion.

On the other hand, the first term of Equation (9) suggests maximum interaction with the base, in contrast with the second term for the reaction null-space. This maximum interaction characteristics can be used to an effective damping of the base vibration for flexible-base systems. For example, using the measurement of the base displacement $\Delta \mathbf{x}_b$ as a feedback signal and G as a gain matrix, we have a simple, but effective vibration suppression law:

$$\dot{\phi}_v = G \mathbf{H}_{bm}^+ \Delta \mathbf{x}_b \quad (11)$$

The above control space is perpendicular to the reaction null-space. Therefore these two operations (10) and (11) can be easily superimposed without interfering each others, just by simple addition:

$$\begin{aligned} \dot{\phi}_c &= \dot{\phi}_v + \dot{\phi}_{ns} \\ \dot{\phi}_c &= g \mathbf{H}_{bm}^+ \Delta \mathbf{x}_b + (\mathbf{E} - \mathbf{H}_{bm}^+ \mathbf{H}_{bm}) \xi \end{aligned} \quad (12)$$

For a space robot which has a 6 DOF manipulator arm on a floating base satellite, $n = 6$ and $m = 6$, therefore the reaction null-space does not exist in general. However, if we care the base attitude only, allowing the base translation during the manipulation, then $m = 3$ and we have the reaction null-space and can find reactionless paths of the manipulator arm.

3 Flight Experiment Opportunity on ETS-VII

The Engineering Test Satellite VII (ETS-VII), developed by National Space Development Agency of

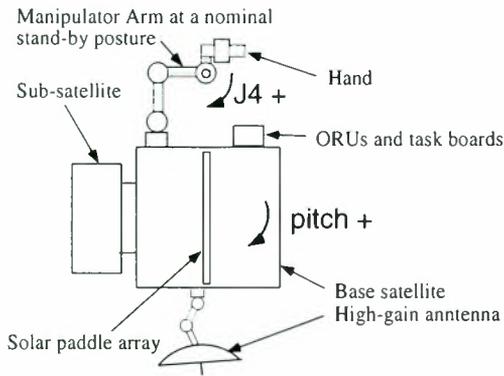


Figure 3 A schematic configuration of ETS-VII

Japan (NASDA) and launched on November 1997, is currently flying on orbit and conducting a lot of interesting experiments with a 2 meter-long, 6 DOF manipulator arm mounted on its un-manned spacecraft. The ETS-VII should be noted as one of remarkable outcomes of research effort on space robots, characterized as a free-flying manipulator system. A plenty of fresh results from ETS-VII will be reported in this symposium.

The officially prepared experiments are completed by the end of May 1999, while the satellite lives in a health condition. Making most of this unique opportunity that a mankind has an operational robot in orbit, NASDA officials recently announced to accept proposals for meaningful options of experiments. And the present authors have proposed the reactionless manipulation, and related operations, to be test by ETS-VII in orbit.

4 Proposed Experiments

The proposed experiments consist of two types of maneuvers for the on-board manipulator. One is the reactionless manipulation, and the other is the attitude change maneuver by the manipulator reaction.

4.1 Reactionless Manipulations

As is presented in Section 2, the reactionless manipulations are obtained by Equation (10) or

$$\mathcal{L}_m = \mathbf{H}_{bm} \dot{\phi} = \mathbf{0}, \quad (13)$$

provided a non-empty null-space of \mathbf{H}_{bm} and an arbitrary ξ .

In practice, we care 3 DOF of attitude disturbance of the base for 6 DOF of the on-board manipulator, then 3 residual DOF are left in ξ and the reactionless manipulations exist in general. In order to uniquely

determine ξ , a relationship with 3 degrees of net freedom can be accepted. Kinematic relationships of the manipulator are such candidate, if paying attention to 3 degrees positions or orientations:

$$\mathbf{v}_h = \mathbf{J}_{mv} \dot{\phi} \quad (14)$$

or

$$\boldsymbol{\omega}_h = \mathbf{J}_{m\omega} \dot{\phi} \quad (15)$$

where \mathbf{J}_{mv} and $\mathbf{J}_{m\omega}$ are partial manipulator Jacobians for the linear and angular velocity of the manipulator end tip, respectively.

Specification of ξ with Equation (14) or (15) is alternatively computed from a direct combination of (13) and (14) or (15), yielding the following solutions:

$$\dot{\phi} = \begin{bmatrix} \tilde{\mathbf{H}}_{bm} \\ \mathbf{J}_{mv} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_h \end{bmatrix} \quad (16)$$

or

$$\dot{\phi} = \begin{bmatrix} \tilde{\mathbf{H}}_{bm} \\ \mathbf{J}_{m\omega} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\omega}_h \end{bmatrix} \quad (17)$$

where $\tilde{\mathbf{H}}_{bm} \in R^{3 \times n}$ is modified from $\mathbf{H}_{bm} \in R^{n \times n}$ in order to focus the attitude motion of the base. Here $n = 6$ then the matrices

$$\begin{bmatrix} \tilde{\mathbf{H}}_{bm} \\ \mathbf{J}_{mv} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{\mathbf{H}}_{bm} \\ \mathbf{J}_{m\omega} \end{bmatrix}$$

are square in 6×6 and conventional inverse exists if non-singular.

Computation of the reactionless manipulations by Equations (16) and (17) are much more simple and practical than by Equation (10) with pseudo inverse and unspecified ξ . For further smooth computation of the conventional inverse continuously along a sequence of motion, the method with an adjoint matrix [13] which is originally proposed for on-line teleoperation can be applied. The method gives a solution for $\mathbf{y} = \mathbf{A}\mathbf{x}$ with a square matrix \mathbf{A} by

$$\mathbf{x} = k \cdot \text{adj}(\mathbf{A})\mathbf{y} \quad (18)$$

where k is an arbitrary constant. This guarantees smooth solutions on and around the singularity with a proper choice of k . And $k = 1/\det(\mathbf{A})$ provides the conventional inverse.

4.2 Specific Proposal and Expected Results

A specific reactionless path of the ETS-VII manipulator arm is computed with Equation (17) from the nominal "stand-by" position toward a positive pitch angle of the hand, while yaw and roll of the hand are kept zero and the positions of the hand are left arbitrary. See Figure 3. Such a motion trajectory in the joint space is depicted in Figure 4, named "Path

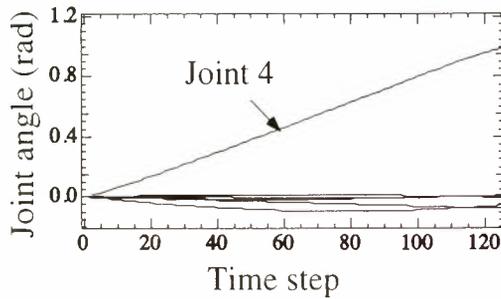


Figure 4 A reactionless maneuver, Path A

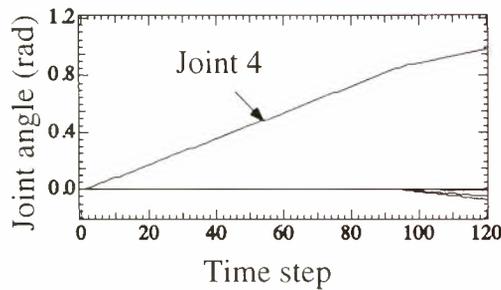


Figure 5 A trivial maneuver, Path B

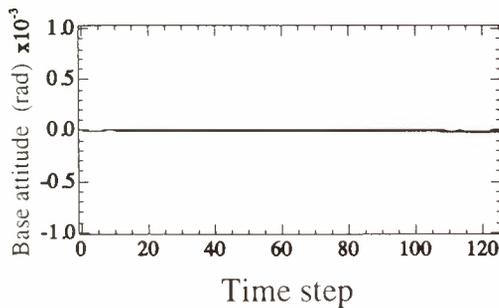


Figure 6 The reaction on the base attitude in Path A operation

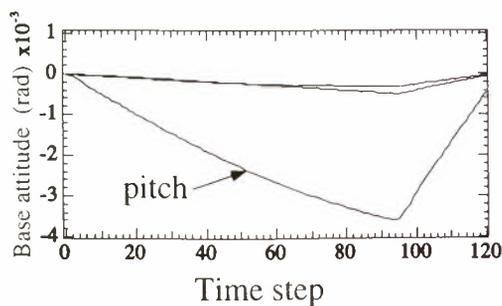


Figure 7 The reaction on the base attitude in Path B operation

A,” where the joint 4 rotates by 50 degrees playing a major role, but other joints also move coordinately to counterbalance the reaction.

For comparison, a trivial operation is planned as shown in Figure 5, named “Path B,” where only joint 4 rotates by 50 degrees first, then an adjustive motion by all joints is made to get the same destination as Path A.

The expected reaction of the base satellite in the absence of the attitude control system are simulated and depicted in Figure 6 and 7 for each manipulation. The simulation is carried out with the software package developed by the Space Robotics Lab. of Tohoku University, “The SpaceDyn,” a MATLAB tool box for space and mobile robots [14] with a relatively precise parameters of ETS-VII.

In these figures, the difference is remarkable with exactly zero reaction in Path A, against non-negligible attitude degradation (0.2 degrees in pitch) in Path B. The proposed motion looks very similar to a frequently observed operation in the manipulation of ETS-VII, when the arm is operated from the nominal stand-by to a posture ready to pick-up an on-board component. Our proposing experiment would prove that a reactionless maneuver, obtained by a small modification from a trivial trajectory, could result in a remarkable stability of the base and saving of the time and energy to recover from the attitude disturbance.

4.3 Attitude Maneuver by Manipulator Reaction

The attitude change or control maneuvers by an effective usage of the manipulator reaction is also proposed. On the contrary to the reactionless maneuvers, a maximum coupling between the manipulator reaction and the base attitude is effective for attitude maneuvers. Such operations are obtained from Equation (11). A good example will be proposed for a possible flight experiment on ETS-VII.

5 Conclusion

In this paper, we propose the reactionless manipulations for a free-flying space robot that would not disturb the base attitude. In the former half of the paper, we present theoretical background and derivation of the reactionless manipulations and related concepts. In the latter half, we propose a specific motion trajectory to yield reactionless maneuver for a possible flight experiment on ETS-VII, a free-flying space robot currently in operation. Dynamic simulations are carried out to check the expected results in the flight experiments. It is clearly seen that the reactionless maneuver in which all joints move coordinately to counterbalance the reaction would yield completely non-zero disturbance on the base, while a trivial maneuver in most of which only one joint

moves at a time, yields non-negligible attitude disturbance.

We do hope the proposed experiments are carried out in orbit by ETS-VII soon.

References

- [1] C.Mavroidis, S.Dubowsky, and V.Raju, "Endpoint control of long reach manipulator systems," in *Proc. IFToMM 9th World Congress*, Milan, Italy, 1995, pp. 1740–1744.
- [2] K. Yoshida, D.N. Nenchev and M. Uchiyama, "Moving base robotics and reaction management control," in *Robotics Research: The Seventh International Symposium*, Ed. by G. Giralt and G. Hirzinger, Springer Verlag, pp. 100–109, 1996.
- [3] C. Vallancourt and C. M. Gosselin, "Compensating for the structural flexibility of the SSRMS with the SPDm," *2nd Workshop on Robotics in Space*, Canadian Space Agency, Montreal, Canada, July 1994.
- [4] Kazuya Yoshida, "A General Formulation for Under-Actuated Manipulators," in *Proc. 1997 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, pp.1651–1657, 1997.
- [5] K. Yoshida and D. N. Nenchev, "A General Formulation of Under-Actuated Manipulator Systems," in *Robotics Research, the Eighth Int. Symp.*, Eds. by Shirai and Hirose, Springer, pp.33–44, 1997.
- [6] D. N. Nenchev, Y. Umetani, and K. Yoshida, "Analysis of a Redundant Free-Flying Spacecraft/Manipulator System," *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 1, pp. 1–6, Febr. 1992.
- [7] K. Yoshida and D.N. Nenchev, "Space Robot Impact Analysis and Satellite-Base Impulse Minimization using Reaction Null Space," in *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, Nagoya, Japan, May 21–27 1995, pp. 1271–1277.
- [8] D.N. Nenchev, K. Yoshida, and M. Uchiyama, "Reaction Null-Space Based Control of Flexible Structure Mounted Manipulator Systems," in *Proc. IEEE 35th CDC*, pp.4118–4123, 1996.
- [9] K. Yoshida and D. N. Nenchev, "Reaction Null-Space Based Control of Under-Actuated Manipulators," in *Proc. 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, Oct. pp.1358–1363, 1998.
- [10] Yoshida et al, "Experiments on the PTP Operations of a Flexible Structure Mounted Manipulator System," in *Proc. 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Osaka, Japan, Nov. pp.246–251, 1996.
- [11] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, A. Konno and M. Uchiyama, "Experiments on reaction null-space based decoupled control of a flexible structure mounted manipulator system," in *Proc. 1997 IEEE Int. Conf. Robotics and Automation*, Albuquerque, New Mexico, April 21-27, pp. 2528 – 2534, 1997.
- [12] A. Gouo, D. N. Nenchev, K. Yoshida, and M. Uchiyama, "Dual-Arm Long-Reach Manipulators: Noncontact Motion Control Strategies," in *Proc. 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, Oct. pp.449–454, 1998.
- [13] Y. Tsumaki, D. N. Nenchev, S. Kotera, and M. Uchiyama, "Singularity-Consistent Inverse Kinematics of a 6 D.O.F Manipulator with a Non-Spherical Wrist," in *Proc. 1997 IEEE Int. Conf. Robotics and Automation*, Albuquerque, New Mexico, April 21-27, pp. 2980 – 2985, 1997.
- [14] <http://astro.mech.tohoku.ac.jp/SpaceDyn>

Rover Control (2)
[Navigation and Piloting]

Command Generation for Planetary Rovers Using Virtual Reality

Theodore Blackmon¹, Charles Neveu², Chris Allport², Laurent Nguyen³, Aaron Kline³,
David Smith¹, Corin Anderson², Vineet Gupta²

1 NASA Ames Research Center

2 NASA contractor with Caelum Research Corporation

3 NASA contractor with Recom Technologies

NASA Ames Research Center

Mail Stop 269-3 Moffett Field, CA 94035-1000

Tel: (650) 604-4710 Fax: (650)

tblackmon@mail.arc.nasa.gov

ABSTRACT

A system architecture is proposed for command generation of planetary rovers through the specification of high-level goals in virtual reality (VR). Through the use of an integrated planning and scheduling system, connected with a high-fidelity simulation model of the rover and its environment, a contingent command sequence could be determined from a set of specified science goals (with given task priorities, constraints, pre-requisites, etc ...). As part of NASA's Advanced Technology Program, this paper describes the work in progress towards enabling such a system and the demonstration of the current status for the Marsokhod '99 Field Experiment in the Mojave Desert.

1. INTRODUCTION

Commanding a vehicle on the surface of another planet such as Mars cannot be accomplished through standard remote control techniques (via teleoperation) due to the inherent time delay, communication bandwidth and uplink/downlink cycles associated with the task. Obviously, increased on-board autonomy and intelligence for a robotic vehicle is a large part of the solution. One might envision a single, white 'GO' button in mission control, which commands the rover to autonomously explore the surface of Mars, only transmitting back a geological sketch map of the area with selected images and a written interpretation. However, is it really practical, or desirable, for us humans on earth to interact with and direct our mechanized servants in such a limited, out-of-the-loop fashion?

By enabling a virtual presence of the remote site through high-resolution 3D modeling of the

terrain, we can gain a greater awareness of the rover's environment and situation, and better determine where we would like the rover to go and what we would like the rover to do. Using VR and web-based technologies, a distributed science team could fluently specify high-level science goals for a team of rover operators to readily implement a conditional sequence of rover commands to accomplish the specified goals. Specification and display of the desired goals and proposed command uplink could be mediated through the same virtual environment, thus providing a digital communication link between the respective science and engineering teams.

Additionally, automated planning and scheduling systems that consider task priority, pre-requisites, time and resource constraints of a semi-autonomous rover could be used in helping to determine a contingent schedule for a selected set of science goals. Furthermore, through high-fidelity simulation, we can gauge what is possible given the rover's capabilities, and partially predict an expected course of action, given a proposed command sequence to be uplinked.

The use of computer models to mediate the human supervisory control of telerobots was recognized some time before the computing power for VR was a reality [Ferrell and Sheridan, 1967]. An early demonstration of the benefits of a model-based approach utilized an on-line computer simulation and graphics display to teach the desired movement and manipulation to a robotic arm [Yoerger, 1982]. To circumvent the inherent difficulties with time delay in teleoperation, a graphical model was superimposed in real-time upon the delayed

video feedback to predict the actions of the telerobot [Noyes and Sheridan, 1984]. Computer model-to-video calibration was further developed [Kim and Stark, 1985] and the predictor display technique has been successfully demonstrated with the advanced teleoperation system at the Jet Propulsion Laboratory [Bejczy et al., 1990] [Kim, 1996].

The concept of 'put-that-there', 'point-and-direct' and 'teleprogramming' for remote robots has been realized using computer graphics simulations to generate high-level goals and command sequences for manipulator arms in structured environments [Schneider and Cannon, 1989] [Wang and Cannon, 1993] [Funda et al. 1992]. More recently, model-based supervisory control has been shown to provide significant benefits for teleoperation, even in the case of minimal time delay [Blackmon & Stark, 1996].

Research at NASA Ames in Virtual Environment Vehicle Control has evolved over the last decade to explore the possibility of using computer models to control planetary rovers in unknown environments [Piguet et al. 1995] [Stoker et al. 1995]. However, the use of 3D computer models for telerobotic control necessitates the generation of accurate models of the remote site, which can present a considerable difficulty in remote exploration of unknown environments.

Using panoramic stereo imagery captured by the cameras of a rover (or lander), it is possible to rapidly build photo-realistic 3D models that can be assembled and visualized at mission control in a VR software interface. MarsMap is such a system that was developed and implemented for Mars Pathfinder, providing the mission scientists and operators with a valuable tool for analysis and planning [Stoker et al., 1999]. Within a half-hour of initial downlink of the stereo panorama images, the data was processed into 3D terrain models, utilizing a process known as the 'Stereo Pipeline'.

Immediately thereafter, the 3D models were loaded into the MarsMap VR system for the mission scientists to fly-over the Pathfinder landing site, and even to project a bird's eye view from above. In addition, sensor information returned from the rover, such as 2D images as well as the sequence and location of science experiments was automatically detected and visually archived within MarsMap (Figure 1). By combining the visualization capabilities with

a simple measurement tool that was calibrated with the Pathfinder lander, 3D information could easily be extracted from the map data to provide relevant locations, distances, angles, and contours of the terrain topology.

For Mars Pathfinder, this 3D measurement information was typically written on pieces of paper by the mission scientists and then used in operations planning, especially for multi-spectral imaging pointing of the Pathfinder lander camera system. Long-range planning for the Sojourner rover was also frequently visualized by the science team utilizing MarsMap, in particular where terrain navigability was in question.

Since Pathfinder, MarsMap has been further augmented with additional user interface elements to now allow the direct specification of science and imaging goals of a mobile rover platform, with the ability to save this goal information to file for later editing and display. This goal information can then be readily assimilated into the command generation user interface elements of a mission, and the results of the planning can subsequently be visualized in MarsMap. Moreover, MarsMap has been combined with a dynamic simulation engine to allow high-fidelity simulation of a rover's interaction with the environment with the appropriate abstraction layer to accept low-level rover commands from the on-board execution software elements.

2. Command Generation with VR

The following two paragraphs present a conceptual overview of a proposed system for utilizing VR as part of the command generation process for planetary rovers.

A collaborative and distributed science team utilizes a VR interface on a limited number of high-fidelity mission operations centers, along with a larger number of desktop web browsers, to specify a collection of science goals for a rover as part of a long-range plan. The distributed science can simultaneously or asynchronously visualize designated goal 'files', thus utilizing the VR system as a communications medium. The respective science PI's (or designated science team members) then work together to select from this larger goal list and possibly modify goals for the next uplink(s). As part of the tools available to the science PI's, an integrated, ground-based

planning and scheduling system connected with a rover simulation model can be used to semi-automatically determine a contingent sequence of rover activities.

The rover operations team oversees and can override this proposed sequence to ensure rover safety, insert appropriate engineering and health commands, and further plan for rover activities where automated planning / task decomposition is insufficient. The resulting proposed uplink is verified by the scheduling system for time and resource constraints, and then sent to the rover simulation to partially predict the resulting outcome of the command sequence. Both the local and distributed science teams can then view the final uplink schedule (as well as rover simulation results).

As part of the research effort to develop and demonstrate the benefits of such a system, a software architecture has been designed and implemented for a modular rover VR planning and display system. Figure 2 illustrates the concepts of this generalized software architecture. The 'RoverVR' interface is made independent of the particular rover base and instrument suite through a set of 'robot', 'instrument' and 'data' objects defined at initialization. Associated with these object classes are specified parameters, including pointers to run-time functions that provide for ...

- Telemetry monitoring and processing of 'data' objects returned from the rover
- Iconic representation of the 'data' in VR
- Display of associated information of the 'data' in VR
- Loading of the 'data' into the VR program
- Methodology of 'data' display in VR
- Interactive manipulation of the 'data' in VR
- Interactive goal planning for defined 'instrument' and 'robot' objects
- Output of a designated goal plan

Upon initialization, defined parameters for the 'robot', 'instrument' and 'data' objects are registered with the 'RoverVR' interface and subsequently used at run-time to enable the data management and planning activities in the VR human interface.

3. Marsokhod '99 Field Experiment

The Marsokhod '99 Field Experiment was designed to develop, demonstrate, and validate

technologies and science strategies for high-science, high-technology performance, and cost-effective planetary surface operations. The results of this blind field test are intended to find direct applications in the NASA Mars Exploration Program, and more generally, in the evolving field of planetary surface exploration. Several highlights are worth noting from this month long field experiment in the Mojave desert.

A hi-resolution stereoscopic imaging system was utilized with a boresighted spectrometer, with similar imaging properties to the 'PanCam / MiniTES' system which is scheduled to fly on the Mars'01 lander and Mars'03 rover platforms. Prior to the field test, a simulated descent imaging sequence was captured utilizing a helicopter and photographic system that closely matched the profile and optical properties of the Mars '98 and Mars'01 descent imaging systems. A flight copy of the Robotic Arm Camera (RAC) which will fly on the Mars '98 and Mars '01 landers was taken to the field for a three days to image a trench dig while the rover / PanCam simulator remained stationary in a 'lander' mode. Finally, the field test wrapped up with the 'ASRO' event, which explored the interaction between a suited astronaut and a rover assistant in a simulated SEP (Science Experimental Package) deployment.

Figure 3 is an information flow diagram of the extended Marsokhod control system for the field experiment. The right portion of the diagram shows the on-board autonomy architecture. Robust navigation of the six-wheeled rover was performed utilizing a visual servo strategy developed as part of an on-going research effort [Wettergreen et. al., 1996]. Of note is the addition of a model-based executive and mode identification / fault monitoring system as part of a larger research effort into increasing the on-board robustness and intelligence of planetary rovers [Bresina et. al, 1999]. This system makes the use of an uplink command sequence format dubbed CRL (contingent rover language) which enables the on-board executive to take action based upon sensed or deduced information regarding the rover state and knowledge of the environment. The remaining sections of this paper provide a system level overview of the relevant ground-based user interface elements as part of the Marsokhod '99 Field Experiment, particularly related to use of VR for the command generation process.

4. Goal Specification Using 'MarsoVR'

One of the most significant uses of VR for rover command planning is the ability to drive a simulation model of the rover over the high-fidelity terrain models generated from stereo imagery, calculating the rover's kinematics along the traverse. This enables a scientist or rover operator to more confidently specify way-points and suitable navigation strategies for the rover base movement in the near to mid vicinity of the most recently captured stereo image sequence (within reasonable accuracy and depth limitations of the stereo camera system). For the Marsokhod field test, the rover operations team in determining navigation heading and distance estimates to designated science targets (Figure 4, upper left) routinely used this capability. In addition, the interactive driving simulation was used to identify terrain areas unsuitable for rover navigation, and to develop combined navigation strategies using the rover's on-board visual servo and dead reckoning with safeguarding capabilities.

The use of an interactive VR model for instrument and arm placement with a rover has similar advantages as with navigation. In cases where the rover base has not moved, a CAD model of the rover instrument arm can be placed with respect to terrain models captured from the same position to confidently determine rock and ground surfaces within direct reach and to specify end-effector coordinates for a safeguarded move. In case where the target object requires movement of the rover base, the VR model is still quite useful for estimating the ability for the rover to reach a target and for determining a strategy on how best to approach a science target with consideration of instrument placement. Figure 4, upper right shows the use of the 'MarsoVR' interface for specifying the arm end-effector and instrument carousel position on a large, flat rock in the direct reach of the rover.

During the field experiment, panoramic imaging and spectrometer experiments, as well as navigation imaging experiments were also specified by the rover operations team in VR. Using the 'MarsoVR' interface, a user sweeps out the pan & tilt extents along with other camera parameters, including pan/tilt step size, camera resolution and image compression (Figure 4, lower left). Beyond simply estimating the downlink data volume, this utility was extremely

useful for visualizing the estimated terrain coverage area, especially when planning for image acquisition sequence to follow a navigation command. For designating spectral targets, the operator points with the 2D mouse and a circle is targeted onto the 3D terrain model showing the desired location and field-of-view for the spectrometer (Figure 4, lower right). The unmodified, original rover image used as the texture on the corresponding 3D model is shown in the planning sub window with a cross-hair on the specified target location.

For the 'MarsoVR' planning interface, all specified science goals and rover commands have a common set of general properties that are specified for purposes planning and scheduling. A constraint on the time window for which the task should be started can be specified if desired. Priorities on task execution as well as downlink of the resulting data are ranked for all tasks. Additionally, pre-requisites for a task, such as other tasks that must precede this task, are also specified. The interface also allows the user to specify general comments that are relevant to the incorporation of this task into the proposed command uplink. These parameters are set through a 2D user interface panel that is integrated with the 'MarsoVR' software program (Figure 5).

'RoverWeb' Interface - As part of the field test web site, a simple image-based, point-and-click interface was implemented to allow the science team members to specify experimental goals remotely. This web-based interface does not require the graphics capability of MarsMap, nor require any special plug-in's such as JAVA, and can thus execute on any standard computer with an Internet browser capability. After selecting a target feature in a rover image along with a goal type (Pan-Cam Image, Navigation Image, Arm Camera Close-Up Image, Spectral Measurement) a scientist completes an Internet form with other information appropriate to the task, and submits this task request. The form information is then translated to an appropriate CRL goal language for command sequence generation and appears for display at mission control at Ames. A copy of the Internet goal form was also posted onto the web so that other scientists could easily monitor the variety of experimental requests made during the mission.

Of course, this electronic method of goal submission was not strictly mandated.

Moreover, as the field experiment evolved, the science team rather made effective use of telecons to typically develop the set of desired goals for the next uplink cycle, and submit the 'science goal requests' as a simple text document detailing the desired set of activities, relevant task parameters, and rationale. This text document with a number of supporting images would be posted into the mission web posting system for other team members to view. After analyzing the proposed set of goals in terms of feasibility, the rover team would respond to this request and often another iteration cycle was required prior to uplink of a sequence that satisfies both science desires and engineering constraints. In terms of effective task planning, it was important for the science team and rover team to have direct verbal interaction, facilitated by a set of central contacts on both the science and rover teams. Verbal communication was essential to ensure that desired goals were understood correctly as well as to provide the science team with better understanding of the rover's capabilities and how to most effectively utilize those capabilities.

5. Command Sequence Generation

Following the submission of rover goals and relevant task parameters, a rover command sequence would be assembled using a 2D forms interface called the 'Command UI'. The 'Command UI' allows the rover operations team to interactively build a contingent sequence of activities to meet the desired goals. The individual activities that comprise the sequence are initially specified at a high-level using the MarsoVR interface, the web-based goal interface, or alternatively specified directly in the 'Command UI'. In fact, in this modular open-architecture design, any custom interface could be used to generate high-level goals for the 'Command UI', given the specification of the 'Contingent Rover Language' (CRL) and the 'Rover Goal Dictionary'.

An operator uses the 'Command UI' to load a set of specified science and engineering tasks for the next uplink into a flat list. Using an iconic representation of the sequence as a hierarchical tree of nodes in time, the operator then selects individual tasks and links these tasks into a sequence. The operator has the ability to place conditional branches into the sequence that are dependent upon run-time information on-board

the rover. A typical example of a conditional sequence for the Marsokhod rover consists of ...

- perform visual servo navigation to desired target
- if visual servo navigation is successful ...
 - capture hi-res image of the target
 - perform 2x2 spectral cube on the target
- else if visual servo navigation fails ...
 - capture images to trouble-shoot failure and re-acquire target
- capture end-of-navigation panorama
- capture end-of-navigation images with fixed low-mounted camera

Screen shots of the 'Command UI' elements are shown in Figure 6. These include panels for selecting among the various proposed goals, editing individual goal properties, and display the resultant timeline of the conditional sequence that has constructed with the 'Command UI'.

Following manual construction of a contingent schedule using the 'Command UI', the sequence of tasks would be further decomposed into a set of low-level rover task primitives using a planning and scheduling system based upon the Just-In-Case approach [Drummond et. al., 1994]. This system accepts a seed schedule and can also automatically add contingent branches to the schedule where appropriate based upon such things as time and resource constraints. More information regarding the automated scheduling and task decomposition is provided in a companion paper written for the Marsokhod '99 field experiment [Bresina et. al., 1999]. This low-level sequence of commands can then be uplinked to the rover and understood by on-board model-based executive.

6. Rover Simulation

Once the rover team has constructed a contingent sequence of rover activities, the various branches of the sequence can be tested using a high-fidelity software simulation of the Marsokhod rover to verify and predict the outcome of the proposed command sequences [Sweet et. al., 1999]. This simulation model is constructed using a hybrid discrete continuous scripting language [Carlson and Gupta, 1998] and is integrated directly with the same terrain model database utilized for the interactive goal specification and task planning in VR. The simulation makes use of a modified rover / terrain 'settling' algorithm presented in [Lincoln,

1996]. A friction model of the rover / wheel soil interaction was adapted from [Andrade et. al., 1998] and also incorporated into the simulation code.

The simulation was developed to accept the same suite of rover commands output by the on-board model-based executive and produce the same telemetry stream available to the executive and mode identification / fault monitoring system. By using the same communications system used on-board, no modification is required to interface the on-board autonomy software programs to the simulation. The resulting simulated rover telemetry (and full state data) can then be visualized and analyzed in 'MarsoVR' with the same tools to view the actual downlink information from the rover. Unfortunately for the field test, the simulation was not heavily utilized due to timing and human resource constraints in the accelerated development schedule. However, it is being utilized more heavily following the field test, especially for continual development of the rover on-board autonomy, prior top validation of the algorithms on the actual rover hardware system.

7. Analysis and Future Directions

Goal generation and task planning in VR was extremely useful for the rover operations team during the Marsokhod '99 Field Experiment. This was particularly true for navigation, manipulation and imaging operations, but less useful for spectral targeting and other strictly image based operations. Without assistance from the rover ops team, the science team made limited use of the VR tools, due to the short duration of their stay at Ames (only 3 days on-site) and the lack of previous familiarization with these tools. Although some use was made early on with the web goal generation tools, the science team largely relied upon the use of telecons and written text files that were handed off to the rover ops team to specify goals for the next uplink.

Currently, the automated planning and scheduling system largely performs a straightforward task decomposition from the goal parameters, with 'canned' engineering contingencies strategically placed to handle expected failure modalities. . Additional contingencies were structured within an uplink through human design of the sequence in the

'Command UI'. There was limited to no automatic planning based upon environmental knowledge and no link with the simulation model. Furthermore, the simulation model was not exploited for command sequence verification and testing prior to uplink.

These and other limitations of the current system were largely due to human resource limitations and the accelerated development schedule for the field test. In addition, somewhat competing goals between science investigation and technology development restricted the testing of technology during the mission. Nevertheless, the excitement of this work and the larger effort by the combined rover operations and autonomy teams continues. To Mars and beyond ...

Acknowledgements

The authors would like to extend acknowledgement to the entire Marsokhod '99 User Interface Team, including those not listed as authors on the paper: Eric Zbinden, Maria Bualat, Linda Kobayashi, John Bresina, Terry Grant, Paul Lucas and Mike Sims.

Thanks and appreciation is also given to other participants in the field experiment, especially to, John Schreiner (Project Lead), Mike Sims (Technical Lead), Hans Thomas (Rover Engineering Lead), Rich Washington (On-board Executive), Carl Stoker (Science Lead) and Nathalie Cabrol (Deputy Science Lead) as well as the participating science teams.

Gratitude is also given to the Computational Sciences Division at the NASA Ames Research Center for extensive support in achieving these research objectives.

REFERENCES

- Andrade, G.; Amar, F.B.; Bidaud, P.; Chatila, R., 1998. Modeling robot-soil interaction for planetary rover motion control. *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications*, Victoria, BC, Canada.
- Blackmon, T.T. and Stark, L.W., 1996. Model-based supervisory control for telerobotics. *Presence: Teleoperators and Virtual Environments*. 5(2), pp. 205-223.
- Bresina, J.L., Golden, K, Smith, D.E. and Washington, R., 1999. Increased flexibility and Robustness of

- Mars Rovers. *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Netherlands.
- Bejczy, A.K., Kim, W.S. and Venema, S.C., 1990. The phantom robot: Predictive displays for teleoperation with time delay. *Proceedings of the IEEE International Conference on Robotics and Automation*. Cincinnati.
- Carlson, B. and Gupta, V., 1998. Hybrid CC and interval constraints. *Proceedings of the International Workshop on Hybrid Systems: Computation and Control*.
- Drummond, M., Bresina, J. and Swanson, K., 1994. Just-in-case scheduling. *Proceedings of the 12th National Conference on Artificial Intelligence*.
- Ferrell, W.R. and Sheridan, T.B., 1967. Supervisory control of remote manipulation. *IEEE Spectrum*, 4(10), pp. 81-88.
- Funda, J., Lindsay, T.S. and Paul, R.P., 1992. Teleprogramming: Toward delay invariant remote manipulation. *Presence: Teleoperators and Virtual Environments*, 1(1), pp. 29-44.
- Kim, W.S., Takeda, M. and Stark, L., 1988. On-the-screen visual enhancements for a telerobotics vision system. *Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics*, Beijing.
- Kim, W.S., 1996. Virtual reality calibration and preview / predictive displays for telerobotics. *Presence: Teleoperators and Virtual Environments*. 5(2), pp. 173-190.
- Lincoln, W., 1996. Simulation and visualization of Martian rover. *Proceedings of the SPIE - The International Society for Optical Engineering (High-fidelity Simulation Training, Test Support, Mission Rehearsal, and Civilian Applications)*, 27(40), pp. 98-105.
- Noyes, M.V. and Sheridan, T.B. 1984. A novel predictor for telemanipulation through time delay. *Proceedings of the 20th Annual Conference on Manual Control*, NASA Ames Research Center.
- Piguet, L.; Fong, T.; Hine, B.; Hontalas, P.; et. al., 1995. VEVI: a virtual reality tool for robotic planetary explorations. *First Workshop on Simulation and Interaction in Virtual Environments*. Iowa City.
- Schneider, S.A. and Canon, R.H., 1989. Experimental object-level strategic control with cooperating manipulators. *International Journal of Robotics Research*, 12(4) pp. 338-350.
- Stoker, C.R., Burch, D.R., Hine, B.P., III; Barry, J., 1995. Antarctic undersea exploration using a robotic submarine with a telepresence user interface. *IEEE Expert*, 10(6), pp. 14-23.
- Stoker, C., Zbinden, E., Blackmon, T.T. et. al. 1999, Analyzing pathfinder data using virtual reality and super-resolution imaging. *Journal of Geophysical Research, Planets*.
- Burch, D.R.; Hine, B.P., III; Barry, J., 1995. Antarctic undersea exploration using a robotic submarine with a telepresence user interface. *IEEE Expert*, 10(6), pp. 14-23.
- Sweet, A., Blackmon, T.T. and Gupta, V., 1999. Simulation of a rover and display in a virtual environment. *Proceedings of the American Nuclear Society 8th International Topical Meeting on Robotic and Remote Systems*, Pittsburgh.
- Wang, C. and Cannon, D.J., 1993. A virtual end-effector pointing system in point-and-direct robotics inspection of surface flaws using a neural network based skeleton transform. *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta.
- Wettergreen, D.; Thomas, H.; Bualat, M., 1997. Initial results from vision-based control of the Ames Marsokhod rover. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, Grenoble, France.
- Yoerger, D.R., 1982. Supervisory control of underwater telemanipulators: Design and experiment. *Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge*.



Figure 1: MarsMap virtual reality (VR) provided Mars Pathfinder mission scientists and operators with a valuable tool for analysis and planning.

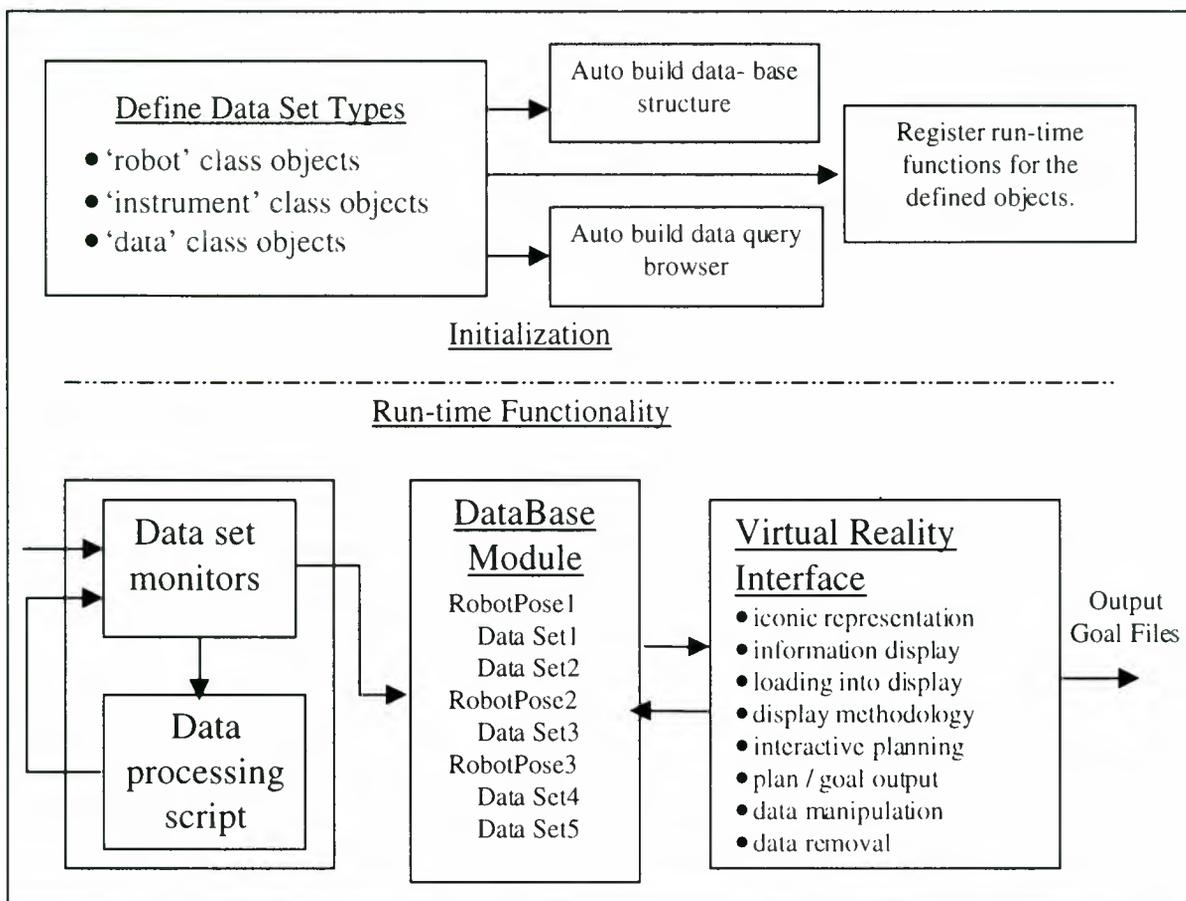


Figure 2: Generalized software architecture that has been developed for a modularized rover VR planning and display system.

Telerobotic System: Marskhod 99 - Mojave Field Test

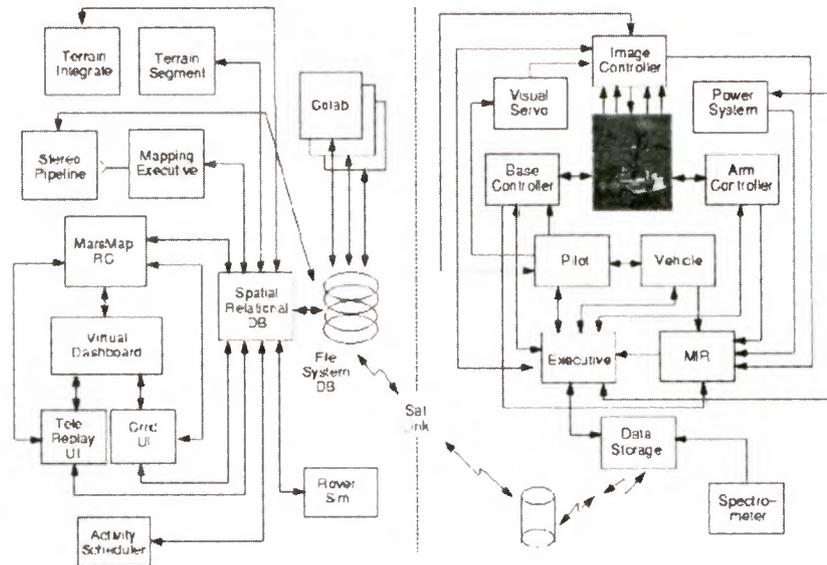


Figure 3: Information flow diagram of the extended Marsokhod control system for the field experiment.

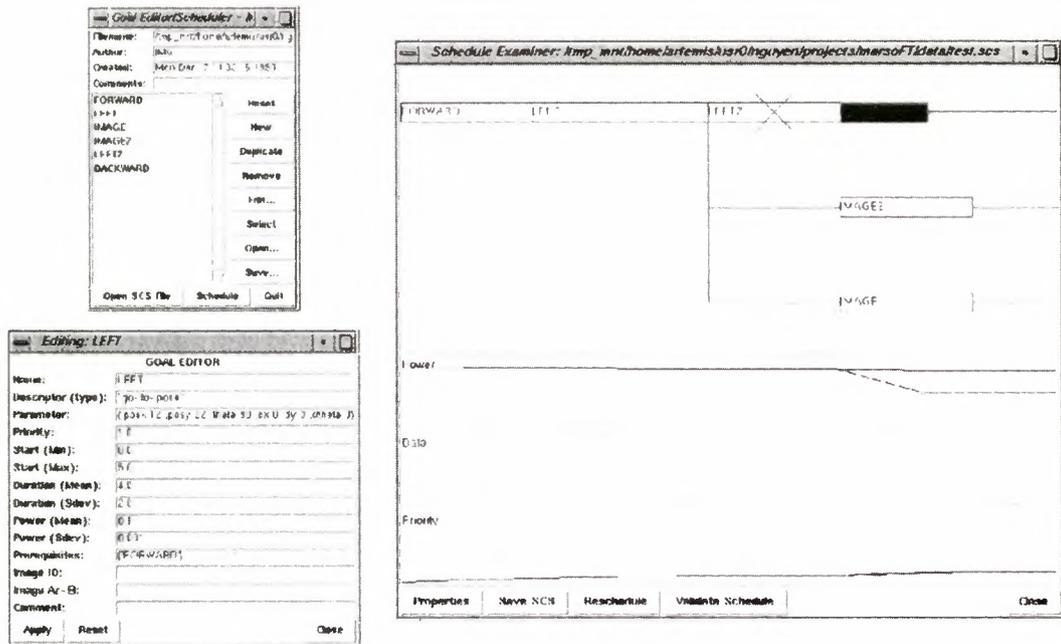


Figure 7: 'Command UI' elements for selecting among the various proposed goals, editing individual goal properties, and display of the resultant timeline of the conditional sequence that has been constructed.

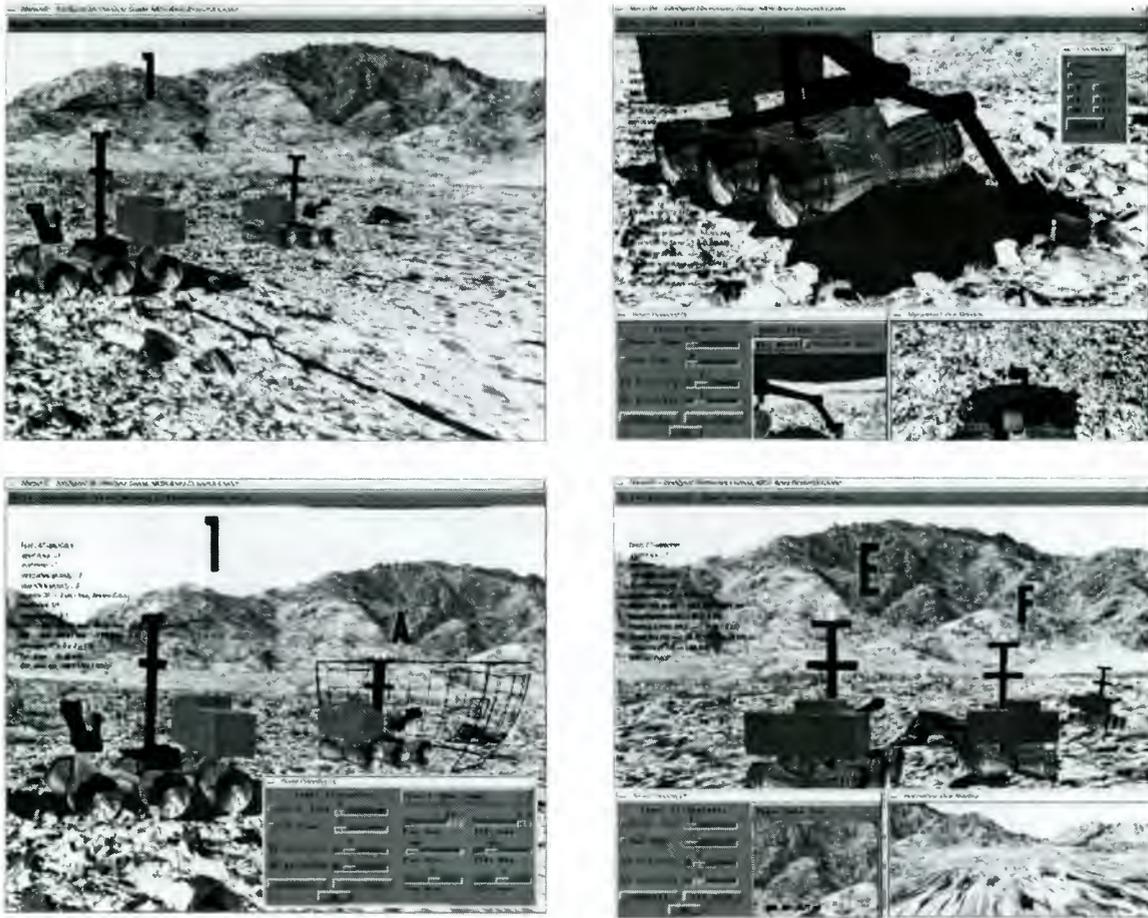


Figure 4: 'MarsoVR' is used to interactively plan rover goals and commands, ranging from navigation heading and distance coordinates, arm placement, panoramic imaging and spectral target selection.

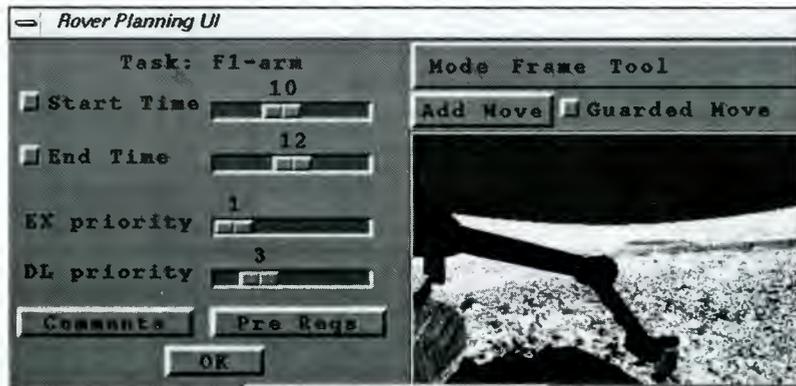


Figure 5: Generic parameters set for every goal include an optional window for task starting time, execution and downlink priorities, task pre-requisites and general comments.

ROVER SELF LOCALIZATION IN PLANETARY-LIKE ENVIRONMENTS

Simon Lacroix, Anthony Mallet and Raja Chatila
LAAS-CNRS

7 Av. du Colonel Roche - 31077 Toulouse Cedex 4 - France

Tel. +33 561 33 62 66 - Fax. +33 561 33 64 55

Simon.Lacroix@laas.fr, Anthony.Mallet@laas.fr, Raja.Chatila@laas.fr

Laurent Gallo

AEROSPATIALE

E/SCS/V - 2 Rue Béranger - 92320 Chatillon - France

Tel. +33 147 46 38 68 - Fax. +33 147 46 32 68

Laurent.Gallo@missiles.aerospatiale.fr

Abstract

The ability for a rover to localize itself with respect to its environment is a crucial issue to tackle autonomous long range navigation. In this paper, we first present and classify the various kind of functionalities a rover should be endowed with to estimate its position during long traverses. We then present a technique that relies on stereovision and pixel tracking to estimate the 6 parameters of the rover displacements, and discuss experimental results obtained with the robot Lama. The paper ends by a brief presentation of a complementary localization function, with respect to an object-based environment model built by the rover as it navigates.

1 Introduction

Future planetary exploration robots will have to explore, map or traverse larger and larger areas. This is a tremendous challenge for roboticists, that must conceive systems endowed with *autonomous long range navigation* capacities. Indeed, the various constraints related to planetary exploration (communication delays, poorly known unstructured terrain) void the possibility to efficiently teleoperate the machine.

At LAAS, we have tackled various aspects related to autonomous long range navigation in unstructured terrains for over ten years, and experimented some in realistic conditions [1, 2]. We are convinced that

to efficiently achieve high level missions defined over a large scale of space and time, a certain degree of *deliberation* is necessary in order to anticipate events, take efficient decisions, and react adequately to unexpected events [3]. In particular, this robot ability to *plan* its activities calls for the building of various environment representations, at several levels of abstraction (topological maps, geometric maps, object representations...). For that purpose, an estimate of the robot position is required, and when executing long missions, sensors on board the lander are no longer helpful to compute it. A position estimate is not only necessary to build coherent environment models, it is also required to ensure that the given mission is successfully being achieved, or to servo motions along a defined trajectory: robot self-localization is actually one of the most important issue to tackle autonomous navigation.

The internal sensors of a robot being always subject to errors and drift, a lot of attention has been paid to exteroceptive data based position correction or estimation algorithms since the very beginning of mobile robotics. Basically, this problem is threefold: (i) the robot has to extract and associate relevant data or models from the gathered data, (ii) he has to process these associations to refine or estimate its position, (iii) and finally, he must be able to *actively* control its perception capacities in order to acquire the relevant data.

We focus in this paper on the first part of the problem, in the context of autonomous navigation in planetary-like environments. The problem is then

very different from indoor environments, the context within which it has essentially been studied up to now: not only the internal sensors data are more noisy - the ground is seldom flat and smooth, but also the environment is not intrinsically structured, as compared to indoor environments where simple geometric primitives match fairly well the reality, and can therefore be "easily" associated from one point of view to the other.

The next section presents a tentative classification of the various position estimation techniques a planetary rover should be endowed with. Section 3 presents in details a technique that enables to estimate the robot motions with an excellent accuracy, using pixel tracking and stereovision. The following section sketch an approach we are currently developing to localize the robot over a long range, on the basis of an object-based environment representation built by the robot. A short discussion concerning the current trend in research related to robot environment modelling concludes the paper.

2 A tentative classification of exteroceptive localization techniques

The various techniques required to compute the robot position as it navigates range from inertial or odometry data integration to absolute localization respecting an initial model. In order to have a better understanding of the problem, we propose here to classify these techniques into four functional categories:

1. *Motion estimation*: it consists in integrating data as a very high pace as the robot moves, similarly to proprioceptive localization¹, in order to estimate the parameters of elementary motions.
2. *Position refinement*: as with proprioceptive localization, exteroceptive motion estimation techniques generate cumulative errors. It is then necessary to rely on the association of elements in the environment (landmarks) perceived from quite different positions to refine the position estimate. The landmark matching problem is here easily solved thanks to the precise enough posi-

tion estimate provided by the motion estimation technique.

3. *Position determination*: even when perceiving and memorizing landmarks, some errors on the position estimate cumulate over a long range of time and space (or after traveling a landmark-free area for instance). Such errors can reach very high values, so that when re-perceiving previously modeled landmarks, one can not rely on the current position estimation to match them. It then calls for *object recognition* to tackle the data association process.
4. *Absolute localization*: in this last category, we put all the techniques that aim at localizing the robot with respect to an initial global model of the environment (such as images or numerical terrain models derived from orbital imagery), a problem often referred to as the "drop-off problem" [4]). If descent imagery can be used to initially localize the lander [5], the problem of absolute localization still has to be tackled when roving over several kilometers.

There are actually five criteria that lead us to establish such a classification of the localization functions: (i) frequency of process activation, (ii) requirements on the precision of the initial robot position, (iii) volume of data required, (iv) necessity to control the data acquisition, and (v) level of abstraction of the processed data. For instance, the motion estimation functionality process a small amount of raw data at a very high frequency, without any control of the data acquisition, and may require a precise initial estimate of the motions (given by the proprioceptive sensors) in order to track and associate successfully the data. On the contrary, the absolute localization function is seldom triggered, requires a high level environment model built upon numerous data sets, for the construction of which data acquisition strategies have been determined, and by definition do not require any precise initial position estimate².

As one can see, the development of several different data processing and environment modeling algorithms is required to tackle the localization problem. All these algorithms are complementary, and provide position estimates with different characteristics: a model of each of these algorithms is required in order to filter the various position estimates into

¹We denote by "proprioceptive localization" all the algorithms that estimate the robot position using proprioceptive sensors - odometers, accelerometers, gyroscopes, inclinometers, etc.

²Note that among these criteria, the abstraction level of the data is actually dubious: we will indeed see in the next sections that we tackle the motion estimation and object recognition functionalities using very similar data (raw grey level images)

a consistent one, and to plan or trigger their activation.

3 Motion estimation using stereovision and pixel tracking

We present here an exteroceptive position estimation technique that is able to estimate the 6 parameters of the robot displacements in any kind of environments, provided it is textured enough so that pixel-based stereovision works well (thanks to progresses on cameras and algorithms, it is even the case for very smooth and flat terrains - the presence of no particular landmark is required). Referring to the classification presented in the former section, this technique is a *motion estimation* function. It is *passive*, in the sense that it does not call for any data acquisition strategy: images are just used as fast as possible. The algorithms therefore do not interfere with any other functionality that makes use of the stereo cameras (obstacle avoidance, map building).

3.1 Principle of the approach

The approach we developed and experimented could be called “exteroceptive dead-reckoning”: it computes an estimate of the 6 displacement parameters between two stereo frames on the basis of a set of 3D point to 3D point matches, established by tracking the corresponding pixels in the image sequence acquired while the robot moves (figure 1). Depending on the time spent by stereovision and on the number of pixels to track, the tracking phase lasts a variable number of frames, which can be reduced to one.

The principle of the approach is extremely simple, but paid we a lot of attention to the selection of the pixel to track: in order to avoid wrong correspondences, one must make sure that they can be faithfully tracked, and in order to have a precise estimation of the motion, one must choose pixels whose corresponding 3D point is known with a good accuracy. Pixel selection is done in three steps: an *a priori* selection is done on the basis of the stereo images (section 3.2); a model of the pixel tracking algorithm is used to discard the dubious pixels during the tracking phase (section 3.3); and finally an outlier rejection is performed when computing an estimation of displacement between two stereo frames (*a posteriori* selection - section 3.4).

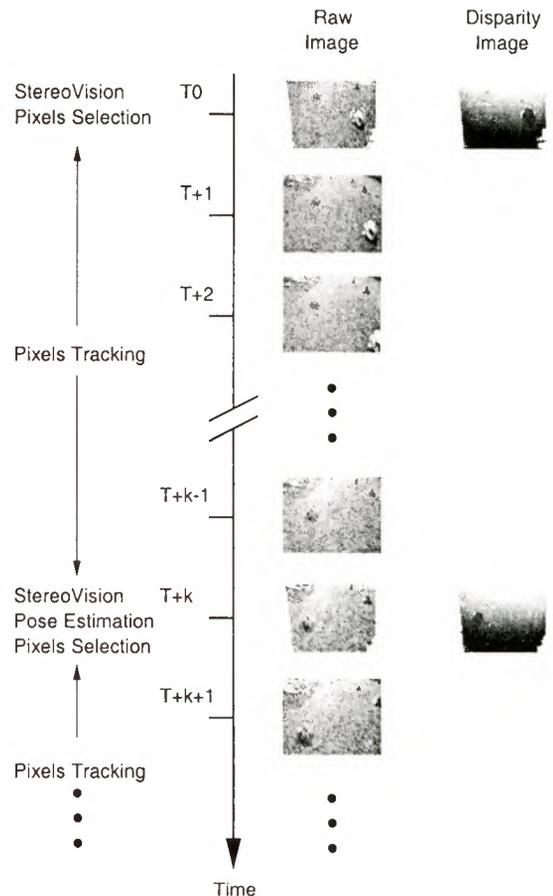


Figure 1: Principle of the approach: at time T_0 , a correlation algorithm computes a disparity image from a stereo pair, and a set of pixels to track is selected. Between T_1 and T_k , the selected pixels are tracked in the image sequence. After the determination of the disparity image at T_k , the set of 3D points correspondences $\{P_0^*, P_k^*\}$ established by the tracking phase is used to compute the displacement $Tr_{0 \rightarrow k}$, and the process starts again.

3.2 Selection of the pixels to track

To initiate the process as a stereo frame comes up, one must select a set of pixels to be tracked. On one hand, one would like to track pixels whose corresponding 3D point is known with a good accuracy: this is done thanks to an error model of the pixel-based stereovision algorithm. On the other hand, one would like to select pixels that are likely to be successfully tracked in the forthcoming image sequence: this is done by studying the behavior of the auto-correlation function in the neighbor of the pixels of the image.

An error model for pixel correlation-based stereovision: A dense disparity image is produced from a pair of images thanks to a correlation-based pixel matching algorithm (we use the ZNCC correlation criteria or a Hamming distance computed on

Census transformed images [6]). False matches are avoided thanks to a reverse correlation and to various thresholds defined on the correlation score curve (essentially on the value of the highest score, and on between this score and the second highest peak in the curve). To get quantitative informations on the precision of the computed disparity (and therefore on the coordinates of the 3D points), we studied a set of 100 images acquired from the same position. As in [7], it appeared that the distribution of the standard deviation on the disparity estimate can be well approximated by a Gaussian. Not surprisingly, the standard deviation on the depth increases quadratically with the depth³. A more interesting fact is that there is a *strong correlation* between the shape of the correlation curve around its peak and the standard deviation on the disparity: the sharper the peak, the more precise the disparity found. This correlation defines an *error model*, that is used during the correlation phase to estimate the error on the computed disparity (figure 2).



Figure 2: A result of our stereovision algorithm: from left to right, original image (only correlated pixels are shown), disparity image, and standard deviation on the disparity estimated with our error model.

However, there are matching errors that occur at the border between two regions of very different intensity values located at different depths (figure 3): as a consequence, the object shape in the disparity image is artificially grown of half the size of the correlation window. These errors, often referred to as “occluding contours artifacts” [8] can not be filtered out thanks to the thresholds on the correlation curve or to a blob filtering algorithm. Moreover, their estimated error tend to be very small: it is practically impossible to avoid the selection of such pixels considering only the stereovision algorithm model.

Selecting good candidates for the tracking algorithm: Planetary environments being highly textured, simple area-based matching techniques are extremely efficient to track pixels in an image sequence (see section 3.3). However, due to noise in

³This would actually be true if the standard deviations on the disparities were not dependent of the depth. In practice, further areas being less textured than closer ones, the disparity standard deviation increases with the depth. As a consequence, the depth standard deviation increases more than with the square of the depth



Figure 3: False matches at the border of a rock: disparity image (left), and correlated pixels (right).

the image and the sampling performed by cameras, the tracking algorithm often eventually drifts: after a few image frames, tracked pixels do not correspond to the same terrain points than the points corresponding to the original pixels. This off course occur especially on smooth, low textured areas, but can also occur on highly textured areas: checking a simple threshold on the standard deviation on the grey levels of the correlation window is not sufficient to ensure that a pixel will be successfully tracked.

To avoid the selection of pixels in the image that are likely to drift during the tracking phase, we defined a measure other the image that represents how similar is a pixel to its neighbors. This measure is based on the computation of the correlation score of one pixel with each of its neighbors, using the same correlation score and window size as the tracking algorithm (auto-correlation). These scores define a correlation peak (a surface), and the shape of this peak indicates how different is one pixel from its neighbors: the sharper the peak, the more different are the neighbors from the pixel. We use the greatest value of the correlation scores found for the neighbors as an indicator of the sharpness of the peak, divided by the theoretical maximum correlation score.

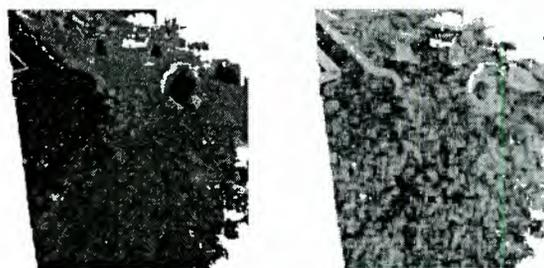


Figure 4: Local similarity measure computed over a whole image. Left: original image, right: similarity measure encoded as grey levels. The darker pixels are good candidates for the tracking algorithm.

Figure 4 presents a result of the computation of this measure over a whole image. One can note that the pixels corresponding to occluding contours are not good candidates for the tracking algorithm: indeed, in the two directions defined by the contour, the correlation windows are very similar. Finally, note that this measure gives an indicator related to

the expected *precision* of the tracking algorithm for a pixel, but not related to the ambiguity (*certainty*): to evaluate an ambiguity measure would require the computation of correlation scores for a wide neighborhood, which is extremely time consuming.

Pixels selection: The set of candidate pixels to track is defined by applying thresholds on the depth standard deviation estimate of the 3D points and on the corresponding pixel similarity measure. The pixels that will actually be tracked are then randomly chosen among the remaining candidates.

3.3 Tracking pixels in an image sequence

Although the pixels to track have been carefully selected, some errors (drifts or false matches) can occur during the tracking phase. In order to avoid such errors, we tested various matching criteria (SSD, ZNCC, Census...) and various template updating strategies on several image sequences to determine the best ones.

Thanks to stereo image sequences, we can detect when a tracking algorithm is drifting by tracking “stereo-corresponding” pixels in the two images, and by checking that after the tracking phase, the returned pixels are still corresponding in the new stereo pair. However, tracking in parallel pixels in a stereo pair takes twice the time to track pixels in one image. We therefore used this possibility to check off-line the tracking algorithm with stereovision, to establish statistics on various tracking algorithms and with various correlation window sizes. This helped to determine the best matching score, template update strategy and optimal window size: we retained the ZNCC correlation score computed over a 11×11 window, and update the template by interpolating the target image around the sub-pixellic matching estimate and with the previous template. Moreover, it allowed us to easily determine the threshold values on the maximum correlation score and on the difference between the second highest peak in the surface, thresholds under which the algorithm is suspected to drift or to return a false match.

The tracking phase is done as follows: given a set of pixel to track and their corresponding 3D points defined on the stereo frame T_0 , the search zone in the image acquired at time T_1 is centered around their predicted position, using the transformation $Tr_{T_0 \rightarrow T_1}$ provided by the robot internal sensors. The size of the search zone is determined according to the uncertainty on the estimated transformation. This prediction is important: it helps to focus the match search in a small area, and therefore reduces the

probability to return a false match. Figure 5 shows the result of tracking a set of pixels in two images acquired from two positions distant of about 0.1 meter. One can see that most pixels have been successfully tracked.



Figure 5: Result of the tracking algorithm on a set of selected pixels. The pixels to track are displayed on the first image (left), and the found pixels on the following image are displayed on right.

3.4 Estimation of the motions

At the date T_k , when a new stereo acquisition is performed, the pixels of the tracked set whose 3D coordinate estimate is now below a certain accuracy are discarded, and the remaining matches are used to compute a first estimate of the 3D transformation $T_{0 \rightarrow k}$, using a constrained least-square method [9]. On the basis of this first estimation, outliers are rejected from the set of matched 3D points, and a new estimate is computed. In our case, the outlier rejection is easy to achieve: indeed, thanks to the *a priori* selection phase and to the thresholds applied during the pixel tracking phase, most of the matched 3D points pairs are consistent.

Tracking pixels over several stereo frames: The obvious drawback of computing elementary motions only between two consecutive stereo frames is that the errors on the motion estimation cumulates over time, just as it happens when integrating the data of the robot’s internal sensor. One way to reduce this errors is to use the possibility to track some pixels over several stereo frames: it allows to determine various displacements parameters every time a stereo image comes up (figure 6).

One could imagine to combine these various displacements estimations using a stochastic filtering technique: this would require the precise knowledge of the uncertainties on every displacement estimation, which is not obvious to obtain. We solved this problem by computing a least square estimation for the whole set of possible 3D points matches in a recursive way: once a position corresponding to time

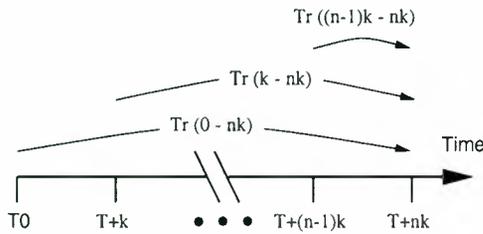


Figure 6: Several displacements between stereo frames can be estimated

T_{nk} is estimated, all the former 3D points coordinates are expressed in this position. At the frame $T_{(n+1)k}$, the matched 3D points are duplicated in order to generate all the possible association sets $\bigcup_{i=0}^{i=n} \{P_{ik} \leftrightarrow P_{(n+1)k}\}$, and the constrained least square method is applied on the whole associations.

The *a priori* pixel selection, performed every time a stereo frame is produced, is then only done to replace the pixels that have been lost (rejected during the tracking phase or as outliers) during the previous stereo-to-stereo cycle.

3.5 Functional architecture

We are currently integrating all the functionalities required by our approach on board the robot Lama. Figure 7 presents the necessary functional modules (integrated under the real-time operating system Vx-Works thanks to GenoM, a software tool developed in our research group to specify and integrate libraries [10]), and the connections between them.

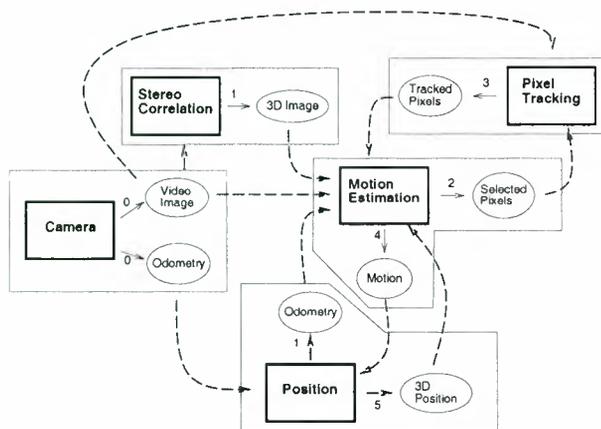


Figure 7: Functional architecture of the motion estimation technique. Functional modules are displayed in bold rectangular boxes, and exported data are displayed in ovals. The numbers indicates the order of data production.

3.6 First experimental results

We have tested the approach with the robot Lama⁴(figure 8), and established comparisons with position records obtained with a differential phase GPS localization system.

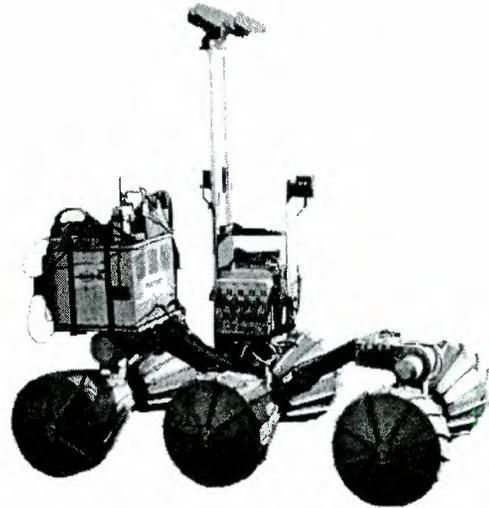


Figure 8: The robot Lama, a Marsokhod robot built by VNIITransmach and equipped at LAAS

The first results are very promising: on some translations of several tens of centimeters, the displacement estimated by the algorithms was close up to 1% to the GPS positions, ie. as precise as the (up to now, we only characterized the translations). We are currently establishing thorough statistics, in order to precisely qualify the precision of the technique. On longer motions, that corresponds to several tens of stereo frames, it appeared clearly that there is a great advantage in tracking pixels over several stereo frames. However, we are not satisfied by the least square estimation algorithm: a Kalman filtering would surely do a better job, all the more since the uncertainty on the 3D points coordinates are well none.

4 Toward unstructured object recognition

We briefly present here a new technique we are currently working on, that allows to identify and register previously perceived objects. It can therefore satisfy both the position refinement and determination processes (section 2). Historically, the first attempts to solve these problems relied on analytical

⁴Lama that is currently lent to us by Alcatel Space Industries

objects models (such as superquadrics for instance). The inadequacy of such models to unstructured objects lead us to study deformable meshes. However if these techniques are well suited for very precise geometric data (such as in medical imaging), they remain useless for robot navigation, where the data range several meters and are much more noisy. We think the recent advances in image registration may be successfully adapted to our problem: instead of aiming at building concise and precise models, these techniques tend to solve the data association problems using either global invariant features [11] or a set of local invariant features [12] determined in the images.

The principle of the method we propose is the following: it consists in building a database of object images (referred to as “aspects”) as the robot navigates. Instead of computing local invariants for these pixels, as it is usually the case when indexing images, we make use of the 3D informations produced by stereovision for all the pixels to predict the object aspect for a *constant* camera distance and orientation⁵. This relaxes the need to compute invariants with respect to image scale and orientation. On this “projected” aspect, a set of discriminant pixels is marked using the autocorrelation function presented in section 3.2 (or using an Harris detector for instance). During the visual aspect database construction, a deformable mesh is determined on the basis of the various 3D points sets. No mesh registration is done: we assume the precision of the robot position given by the motion estimation technique is precise enough to build a mesh that is roughly consistent.

When perceiving an object after a while or a long distance travelled, the problem is to determine whether it has already been perceived or not, and if yes, to determine the robot location with respect to the memorized positions corresponding to aspects. This is done according to the following procedure:

1. The first phase consists in using global attributes to select among all the candidates aspects the ones that are most likely to be matched with the newly perceived object. These attributes are coarse geometric informations (such as the estimate of the object’s volume and inertia moments derived from the mesh), and global photometric informations (texture for instance).
2. The second phase consists in selecting the aspect among the remaining candidates that resembles the most the current aspect. This is

⁵The orientation of the camera with respect to the gravity vector is faithfully provided by inclinometers or the motion estimation technique.

done on the basis of the current aspect prediction for the constant camera distance and orientation chosen during the database building, and by evaluating simple correlation scores (as in the motion estimation technique) for discriminant pixels provided by the auto-correlation function. The originality of the method rely here on the use of the 3D data to project the current aspect to a viewpoint as close as possible to the ones stored in the database, which relaxes the need to compute invariants. Up to now, the best candidate is only chosen using a measure defined on the correlation score for all the marked pixels. Geometric constraints between the marked pixels would probably be helpful when dealing with a large database: the problem is similar to primitive based object recognition techniques.

3. The last phase consists in determining the robot pose with respect to the matching aspect, using a 3D points set of correspondences, as in the motion estimation technique.

One of the critical point of the method is the ability to segment the objects in the data. We have only considered the easy case of rocks lying on a rather flat ground, and developed a simple object detection procedure on the disparity image. It relies on the possibility to quickly compute a virtual disparity image that corresponds to a theoretical flat ground, using the estimate of the robot attitude. A difference between this predicted image and the perceived disparity image exhibits the parts that are above the ground: a simple threshold on this difference lead to a “blob image”, each blob corresponding to a potential object (similar simple segmentation techniques can be applied on the 3D points image). However, one of the advantage of our matching method is that it do not require a faithful segmentation.

5 Conclusions

Rover self-localization is an extremely important issue to tackle in order to endow a robot with autonomous long range navigation capacities. In this paper, we have discussed the various kind of functionalities to develop in order to solve this problem. These functionalities require various data processing and environment modeling algorithms, and may require the determination of data acquisition strategies.

We have presented an approach that is able to estimate elementary robot motions on the basis of stereovision, without building any environment model,

and we have sketched an object modeling approach that can satisfy both the position refinement and determination processes over a long range.

Most investigations concerning localization in outdoor environments relied mainly on geometric characteristics: on rough terrains, a *digital map elevation* is directly used to feed an iconic matching procedure [13, 14], or geometric features extracted from the model are matched [15]. On rather flat terrains where obstacles are easily segmented, some techniques relying on geometric obstacle models have been proposed [16, 17]. However, if geometry is an essential feature to build environment models and indispensable to compute positions, the errors in the data and the models makes the association algorithms very fragile.

The approaches we presented rely essentially on image data, rather than on geometric data. Indeed, the increasing ability to store and rapidly process a large amount of data, due to performances progresses of the computers, lead us to develop techniques that make a strong use of raw image data. As it seems to be the current trend in the perception community, we tend to give up "reconstructionist" approaches that aim at building a concise representation. One can see that if geometry remains an indispensable feature, its role is strongly diminishing in the data association processes.

References

- [1] R. Chatila and S. Lacroix. A case study in machine intelligence: Adaptive autonomous space rovers. In A. Zelinsky, editor, *International Conference on Field and Service Robotics, Canberra (Australia)*, number XII. Springer-Verlag, July 1998.
- [2] S. Lacroix, R. Chatila, S. Fleury, M. Herrb, and T. Siméon. Autonomous navigation in outdoor environment : Adaptive approach and experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, California (USA)*, May 1994.
- [3] R. Chatila. Deliberation and reactivity in autonomous mobile robots. *Robotics and Autonomous Systems*, 16(2-4):197-211, 1995.
- [4] F. Cozman and E. Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *International Symposium on Experimental Robotics, Barcelona (Spain)*, pages 164-172, June 1997.
- [5] L. Matthies, C. Olson, G. Tharp, and S. Laubagh. Visual localization methods for mars rovers using lander, rover and descent imagery. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space, Tokyo (Japan)*, pages 413-418, July 1997.
- [6] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision, Stockholm (Sweden)*, May 1994.
- [7] L. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *IEEE International Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (USA)*, pages 765-768, 1992.
- [8] R. Sara and R. Bajcsy. On occluding contour artifacts in stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 852-857, June 1997.
- [9] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426-1446, Nov/Dec 1989.
- [10] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *Special Issue of the International Journal of Robotics Research on Integrated Architectures for Robot Control and Programming*, 17(4):315-337, April 1998. Rapport LAAS N97352, Septembre 1997, 46p.
- [11] B. Schiele and J. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth European Conference on Computer Vision, Cambridge, UK*, April 1996.
- [12] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), May 1997.
- [13] I.S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278-292, Feb. 1992.
- [14] C. Olson and L. Matthies. Maximum likelihood rover localization by matching range maps. In *International Conference on Robotics and Automation, Leuven (Belgium)*, pages 272-277, May 1998.
- [15] P. Fillatreau and M. Devy. Localization of an autonomous mobile robot from 3d depth images using heterogeneous features. In *International Workshop on Intelligent Robots and Systems, Yokohama (Japan)*, 1993.
- [16] M. Hebert. 3d landmark recognition from range images. In *IEEE International Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (USA)*, pages 360-365, 1992.
- [17] S. Betge-Brezetz, R. Chatila, and M. Devy. Object-based modelling and localization in natural environments. In *IEEE International Conference on Robotics and Automation, Nagoya, Japan*, 1995.

3D Localization for a Mars Rover Prototype*

Stergios I. Roumeliotis¹ and George A. Bekey^{1,2}

stergios|bekey@robotics.usc.edu

¹Department of Electrical Engineering

²Department of Computer Science

Robotics Research Laboratories

University of Southern California

Los Angeles, CA 90089-0781

Abstract

In this paper we consider the problem of localizing a mobile robot on uneven terrain. The localization problem is decomposed into two stages; attitude estimation followed by position estimation. The innovation of our method is the use of a smoother, in the attitude estimation loop that outperforms other Kalman filter based techniques in estimate accuracy. The smoother exploits the special nature of the data fused; high frequency inertial sensor (gyroscope) data and low frequency absolute orientation data (from a compass or sun sensor). Two Kalman filters form the smoother. During each time interval one of them propagates the attitude estimate forward in time until it is updated by an absolute orientation sensor. At this time, the second filter propagates the recently renewed estimate back in time. The smoother optimally exploits the limited observability of the system by combining the outcome of the two filters. The system model uses gyro modeling which relies on integrating the kinematic equations to propagate the attitude estimates and obviates the need for complex dynamic modeling. The Indirect (error state) form of the Kalman filter is developed for both parts of the smoother. The proposed approach is independent of the robot structure and the morphology of the ground. It can easily be transferred to another robot which has an equivalent set of sensors. Quaternions are used for the 3D attitude representation, mainly for practical reasons discussed in the paper. The proposed innovative algorithm is tested in simulation and the overall improvement in position estimation is demonstrated.

1 Introduction

Future missions to Mars will demand long traverses (several km) of rovers to sites of scientific interest. In order to autonomously perform their scientific tasks, these rovers need to know their position precisely. The focus of our research effort is to localize an experimental rover capable of navigating in a 3D environment. Specifically, we

*This work is supported by JPL, Caltech under contract #959816.

are motivated by the problem of localizing the next generation of robot rovers [10] on the surface of Mars. Localization is the problem of determining the position of a mobile with respect to a global or local frame of reference in the presence of sensor noise, uncertainties and potential failures. The basic idea behind many mobile robot localization techniques is to combine sensor data with a priori knowledge about the specifications of these sensors, the structure of the mobile platform, and the environment the vehicle travels in. For example, it is often assumed that a detailed map of the area is known. In this case, the problem of identifying the position of the robot is the problem of finding an area within the map such that the expected sensor values are at all times in accordance with the actual readings.

The assumptions made hereafter are that 1) *No prior maps* of the environment are available and 2) Global Positioning System (GPS) signals are *not detectable* on the surface of Mars. In this case absolute positioning is not feasible. The robot is not capable of determining its position directly by sensing its surroundings (absolute localization). Instead, relative positioning techniques have to be involved. The rover must track its position starting from the landing site through every point of its trajectory.

Many current localization efforts have focused on supporting high quality position tracking. Different sensing devices and odometric techniques have been exploited for this purpose. The common characteristic of these approaches is that they rely on the integration of some kinetic quantity. The main drawbacks of any form of odometry are: 1) Every sensor monitoring the motion of the vehicle has a certain type and level of noise contaminating its signal. Integration of the noisy components causes gradual error accumulation and makes the estimates untrustworthy. 2) The kinematic model of the vehicle is never accurate. For example, we do not know with infinite precision the distance between the wheel axes of the vehicle. 3) The sensor models also suffer from inaccuracies and can become very complicated. For example, the use of complicated models to describe the gyroscope drift. 4) The motion of the vehicle involves external sources of error that are not observable by the sensors used. For example,

slippage in the direction of motion or in the perpendicular direction is many times not detected by the motion sensors. Externally provided or extracted information is necessary from time to time if we wish to keep the error bounded. This group of approaches is also referred to as “dead-reckoning”.

Global (absolute) orientation measurements can drastically increase the accuracy of the position tracking estimate and reduce the rate of growth of the associated uncertainty. In the case of a rover such as Rocky 7, its attitude can be estimated (relative) in real time by integrating the rotational velocity of the vehicle as this is measured by 3 gyroscopes on-board. The problem with this approach is that while the robot is in motion, the rates of roll, pitch and yaw available from gyroscopes are subject to drift and noise. The orientation estimates drift away from their real values and thus they become un-trustworthy. Even small errors in the orientation fast produce large errors in position. As explained in later sections, the absolute (global) orientation of the vehicle can be measured but only intermittently. The focus of this research effort is to provide the best attitude and position estimates when the absolute orientation measurements are not available continuously.

In this paper we address the problem of 3D localization for mobile robots in the absence of absolute positioning information. We concentrate on bounding the attitude uncertainty through periodic use of absolute attitude measurements. As a consequence the position estimate degrades slowly compared to the case when no absolute orientation information is available. The attitude estimate relies on the gyros when the vehicle is in motion while a tri-axial accelerometer is used as an absolute orientation measuring device (roll and pitch) in conjunction with a sun sensor (yaw) when the vehicle is at stop. At the end of each interval of motion a smoother is used which propagates the new absolute orientation information backwards using the previously acquired gyro information. This lowers the uncertainty of the attitude estimate throughout the interval of smoothing; that is when the vehicle was in motion. Both the forward and backward estimators are Indirect (error state) Kalman filters and gyro modeling is used instead of a dynamic model of the robot. Smoothing is being applied here to the mobile robot localization problem for the first time. The proposed smoother based localization algorithm subject to the aforementioned constraints, generalizes across different mobile robot platforms with varying kinematics and dynamics.

In the next section we survey previous work in robot localization. Section 3 examines the dependence of the position estimate on the attitude estimate. We discuss the various attitude measuring devices used, the rationale behind dynamic model replacement and the Indirect Kalman filter and a basic gyro model. Section 4 contains a derivation of the error state equations for the 3-D case using unit quaternions. The linear time-variant equations of the system model and the non-linear equations of the observation model are derived. An Indirect Kalman filter based

on these models is developed. The improvement due to the smoother is demonstrated. Section 5 shows how the position is updated using the improved attitude estimates and section 6 summarizes the contributions of this work and discusses future avenues of research.

2 Previous Work

In order to deal with systematic errors in indoor applications, a calibration technique called the UMBmark test is given in [3]. [4] discusses a technique called gyrodometry, which uses odometry data most of the time, while substituting gyro data only during brief instances (e.g. when the vehicle goes over a bump) during which gyro and odometry data differ drastically. This way the system is kept largely free of the drift associated with the gyroscope. A complementary Kalman filter [6] is used in [9] to estimate the robot's attitude from the accelerometer signal during low frequency motion and the gyro signal during high frequency motion. The attitude information is then used to calculate a position increment. In [1] the authors use a low cost INS system (3 gyroscopes, and a triaxial accelerometer) and 2 tilt sensors. Their approach is to incorporate in the system *a priori* information about the error characteristics of the inertial sensors and to use this directly in an Extended Kalman Filter (EKF) to estimate position.

Examples of absolute localization include [13] in which the localization algorithm is formalized as a tracking problem, employing an EKF to match beacon observations to a map in order to maintain an estimate of the position of the mobile robot; [2] in which the authors use an EKF to fuse odometry and angular measurements of known landmarks and [22] in which a Bayesian approach is used to learn useful landmarks for localization.

Most of the above approaches limit themselves to the case of planar motion. In addition, their accuracy depends heavily on the presence of some form of an absolute positioning system. We consider motion on uneven terrain (3D localization) and propose an estimation algorithm that is capable of incorporating absolute position measurements but is also able to provide reliable estimates in the absence of externally provided positioning information. Our method performs attitude estimation using an Indirect Kalman filter that operates on the error state.

3 Localization and Attitude Estimation

In this section we examine the relation between the attitude estimate and the position estimate. We use an experimental Mars rover prototype (Rocky 7 [10]) as the motivating example throughout this paper. The assumption is that the robot has wheel encoders, 3 gyros, 3 accelerometers and a sun sensor. Since there is no device measuring the absolute position of the rover (there is no GPS on Mars), the position can only be estimated through the integration of the accelerometer signal which has bias and noise. Consider also, that the propagation of the po-

sition relies upon the attitude estimate. Small errors in orientation fast become large errors in position. Formally speaking, the position is not observable and thus the uncertainty of its estimate will grow unbounded. The most promising course of action with this set of sensors is to focus on gaining a very precise attitude estimate. As a result the position uncertainty will grow at a slower rate:

1. The accelerometer measures both the vehicle's acceleration and the projection of the gravitational acceleration on the accelerometer local frame. The relation between these is described by:

$$\vec{p}(t) = \vec{f}(t)/m = \vec{a}_{acc}(t) - A(q(t))\vec{g} \quad (1)$$

where \vec{p} is the vehicle's (non-gravitational) acceleration, \vec{a}_{acc} is the measurement from the 3-axis accelerometer and \vec{g} is the gravitational acceleration. Precise knowledge of the orientation matrix $A(q)$ is mandatory to extract \vec{p} accurately.

2. The next step requires integration of \vec{p} to derive the position. \vec{p} is local (i.e. expressed in a coordinate frame attached to the robot) and in order to calculate the position in global coordinates the attitude information is once again required:

$$\vec{p}(t) = \int_0^t dt' \int_0^{t'} A^T(q(t''))\vec{p}(t'')dt'' \quad (2)$$

3.1 Attitude Measuring Devices

The on-board gyroscopes can be used to calculate the attitude of the vehicle by integrating their signal. On the other hand, the sun sensor directly measures the values of the two components of a two-dimensional vector. This vector is the projection of the unit vector towards the sun on the sun sensor plane. Another sensory input of the same nature is required in order to satisfy attitude observability requirements. While the accelerometer is mainly used to advance the position estimate (Equations 1,2) it can also be used in an alternative way. An accelerometer can measure the local gravitational acceleration, a three-dimensional vector parallel to the local vertical. This provides another orientation fix independent from the sun and thus makes the vehicle's attitude observable. When the vehicle is stopped the accelerometer measures only the gravitational acceleration namely $\vec{a}_{acc} = A(q)\vec{g}$. The roll and pitch of the vehicle can thus be precisely calculated. The sun sensor provides the yaw measurement and thus the matrix $A(q)$ is observable and precisely known when at stop.

This method fails when the rover is in motion. The gravity vector is then "contaminated" by the non-gravitational acceleration of the vehicle (Equation 1). The gravity vector could be extracted while the vehicle is moving if an independent measurement of its own acceleration was available. Research efforts [23, 9] have tried to address this

problem using additional information from odometry. We believe that these approaches are sufficient for indoor applications and can deal with cases of motion over small objects but are not accurate enough for general outdoor environments mainly because of the limited accuracy of the estimates of the non-gravitational acceleration. A more thorough consideration of the problem would require dynamic modeling of the vehicle. An estimator that incorporates a dynamic model of the vehicle [21] could estimate its non-gravitational accelerations.

3.2 Dynamic Model Replacement

In our approach we avoid dynamic modeling and restrict ourselves to **use the accelerometer only when the rover is at stop**. The reasons for avoiding dynamic modeling are: 1. generality, 2. practical estimator size, 3. reported poor payoffs [11] due to dynamic modeling, and 4. complexity. Due to space constraints, we do not discuss these further, the interested reader is referred to [18, 19] for further details.

3.3 The Indirect-feedback Kalman Filter

As mentioned before, Kalman filtering has been widely used for localization purposes. The kinds that usually appear in mobile robot applications are the linear Kalman filter and the Extended Kalman filter (EKF) forms of the full state Kalman filter. In this work we choose to use the error-state form of both the linear Kalman filter and EKF. In the error-state (indirect) formulation, the *errors* in orientation are among the estimated variables, and each measurement presented to the filter is the difference between the INS and the external source data (i.e from absolute orientation sensors). In the following section we derive the equations needed for such a formulation. The primary reasons to pick this formulation are 1. No explicit modeling of the vehicle dynamics is needed, 2. The filter runs at a relatively low frequency, and 3. In case the filter fails, integrated estimates of the INS data continue to be available.

In the feedback form of the Indirect-feedback Kalman filter the updated error estimate is actually fed back to the INS to correct its "new" starting point, i.e. the state that the integration for the new time step will start from. The rationale behind the Indirect Kalman filter as well as the feedback form are discussed in further detail in [18, 19].

3.4 Gyro Modeling

A great difficulty in all attitude estimation approaches that use gyros, is the low frequency noise component, also referred to as bias or drift that violates the white noise assumption required for standard Kalman filtering. This problem has attracted the interest of many researchers since the early days of the space program [15]. Inclusion of the gyro noise model in a Kalman filter by suitably augmenting the state vector has the potential to pro-

vide estimates of the sensor bias when the observability requirement is satisfied. Early implementations of gyro noise models in Kalman filters can be found in [16].

An estimate of the attitude would imply the derivation of the dynamics of the robot, which we wish to avoid for the reasons listed in the previous section. In order to do so we relate the gyro output signal to the bias and the angular velocity of the vehicle using the simple and realistic model [8]. In this model the angular velocity about a particular axis $\omega = \dot{\theta}$ is related to the gyro output ω_m according to the equation:

$$\dot{\theta} = \omega_m + b + n_r \quad (3)$$

where b is the drift-rate bias and n_r is the drift-rate noise. n_r is assumed to be a Gaussian white-noise process with covariance N_r . The drift-rate bias b is not a static quantity but is driven by a second Gaussian white-noise process, the gyro drift-rate ramp noise n_w . Thus $\dot{b} = n_w$ with covariance N_w . The two noise processes are assumed to be uncorrelated.

4 3-D Attitude Estimation

The proposed method in the 3D case is summarized in Figure 1. It should be noted that only the forward filter estimate is available in real-time. The smoother runs off-line (during the times that the robot is halted). This technique is not limited to robots used for Mars exploration. It can be applied to any other autonomous vehicle equipped with an equivalent set of sensors. The mixing of high frequency inertial sensors with low frequency absolute (position or orientation) sensors is becoming common in mobile robotics. Robots equipped with GPS or landmark tracking devices, usually carry additional sensors that can be used for localization when the GPS signal degrades or the landmarks are occluded. Our framework could be used to combine the data from such sensor sets as well.

4.1 Attitude kinematics

We use quaternions to parameterize the robot's attitude for three practical reasons. First, the prediction equations are treated linearly, secondly the representation is free of singularities and finally the attitude matrix is algebraic in the quaternion components, thus eliminating the need for transcendental functions. The reader is referred to [5] for a review on quaternions.

The physical counterparts of quaternions are the rotational axis \hat{n} and the rotational angle θ that are used in the Euler theorem regarding finite rotations. By taking the vector part of a quaternion and normalizing it, we can find the rotational axis, and from the last parameter we can obtain the angle of rotation [7]. Following the notation in [12], a unit quaternion is defined as:

$$q = [q_1 \ q_2 \ q_3 \ q_4]^T \quad q^T q = 1 \quad (4)$$

where the first three elements of the quaternion can be

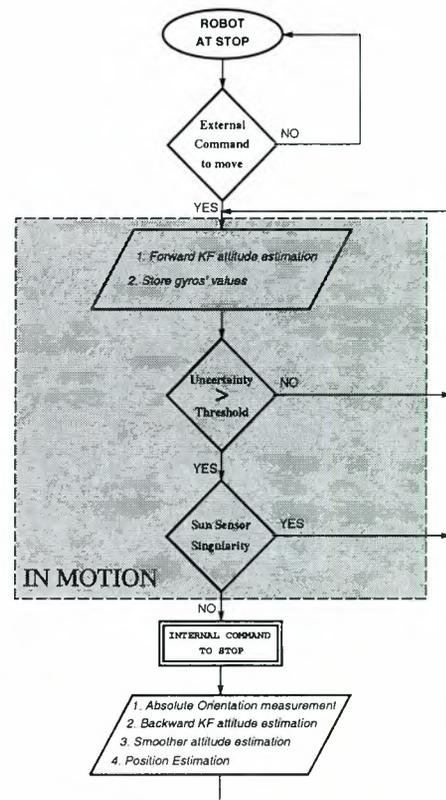


Figure 1: Algorithm Flow Chart: While the robot is in motion the forward Kalman filter uses gyro data to produce (in real-time) a first approximation of the attitude estimate. When the covariance of this estimate exceeds a preset threshold the robot is stopped. An absolute orientation measurement is made using the sun sensor and the three-axis accelerometer. A backward estimate is computed (off-line) and its results are combined (off-line) with the estimate from the forward filter using a smoother. Finally, the position is estimated (off-line) using the (smoothed) attitude estimate for each instant of the trajectory.

written in a compact form as:

$$\vec{q} = \hat{n} \sin(\theta/2) \quad (5)$$

The attitude matrix is obtained from the quaternion according to the relation:

$$A(q) = (|q_4|^2 - |\vec{q}|^2)I_{3 \times 3} + 2\vec{q}\vec{q}^T + 2q_4 \begin{bmatrix} \vec{q} \\ \end{bmatrix} \quad (6)$$

where

$$\begin{bmatrix} \vec{q} \\ \end{bmatrix} = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \quad (7)$$

is a 3×3 skew symmetric matrix generated by the 3×1 vector \vec{q} . The matrix $A(q)$ transforms representations of vectors in the reference coordinate system to representations in the body fixed coordinate system. The rate of change of the attitude matrix with time is given by:

$$\frac{d}{dt}A(t) = \begin{bmatrix} \vec{\omega}(t) \\ \end{bmatrix} A(t) \quad (8)$$

where the corresponding rate for the quaternion is:

$$\frac{d}{dt}q(t) = \frac{1}{2}\Omega(\vec{\omega}(t))q(t) \quad (9)$$

with

$$\Omega(\vec{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (10)$$

At this point we present an approximate body-referenced representation of the error state vector and covariance matrix. The error state includes the bias error and the quaternion error. The bias error is defined as the difference between the true and estimated bias.

$$\Delta\vec{b} = \vec{b}_{true} - \vec{b}_i \quad (11)$$

The quaternion error is not the arithmetic difference between the true and estimated but it is expressed as the quaternion which must be composed with the estimated quaternion in order to obtain the true quaternion.

$$\delta q = q_{true} \otimes q_i^{-1} \text{ or } q_{true} = \delta q \otimes q_i \quad (12)$$

The advantage of this representation is that since the incremental quaternion corresponds very closely to a small rotation, the fourth component will be close to unity and thus the attitude information of interest is contained in the three vector component $\delta\vec{q}$ where

$$\delta q \simeq [\delta\vec{q} \ 1]^T \quad (13)$$

Starting from equations:

$$\frac{d}{dt}q_{true} = \frac{1}{2}\Omega(\vec{\theta}_{true})q_{true} \quad (14)$$

and

$$\frac{d}{dt}q_i = \frac{1}{2}\Omega(\vec{\theta}_i)q_i \quad (15)$$

where $\vec{\theta}_{true}$ is the true rate of change of the attitude and $\vec{\theta}_i$ is estimated from the measurements provided by the gyros, it can be shown [18] that

$$\frac{d}{dt}\delta\vec{q} = \begin{bmatrix} \vec{\omega}_m \end{bmatrix} \delta\vec{q} - \frac{1}{2}(\Delta\vec{b} + \vec{n}_r) \frac{d}{dt}\delta q_4 = 0 \quad (16)$$

where $\vec{\omega}_m$ is the output of the gyros. Using the infinitesimal angle assumption in Equation 5, $\delta\vec{q}$ can be written as $\delta\vec{q} = \frac{1}{2}\delta\vec{\theta}$. Thus Equation 16 can be rewritten as:

$$\frac{d}{dt}\delta\vec{\theta} = \begin{bmatrix} \vec{\omega}_m \end{bmatrix} \delta\vec{\theta} - (\Delta\vec{b} + \vec{n}_r) \quad (17)$$

Differentiating Equation 11 and assuming $\vec{b}_{true} = \vec{n}_w$ and $\dot{\vec{b}}_i = 0$, the bias error dynamic equation is $\frac{d}{dt}\Delta\vec{b} = \vec{n}_w$ which when combined with Equation 17 yields the error state equation:

$$\frac{d}{dt} \begin{bmatrix} \delta\vec{\theta} \\ \Delta\vec{b} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \vec{\omega}_m \end{bmatrix} & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta\vec{\theta} \\ \Delta\vec{b} \end{bmatrix} + \begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \vec{n}_r \\ \vec{n}_w \end{bmatrix} \quad (18)$$

In a more compact form Equation 18 is:

$$\frac{d}{dt}\Delta x = F\Delta x + Gn \quad (19)$$

4.2 Discrete system: Indirect forward Kalman filter equations

4.2.1 Propagation

At this point we define $q_{k/k}$ ($\vec{b}_{k/k}$) as the quaternion (bias) estimate at time t_k based on data up to and including $z(t_k)$, $q_{k/k-1}$ ($\vec{b}_{k/k-1}$) the quaternion (bias) estimate at time time t_{k-1} propagated to t_k , right before the measurement update at t_k . The estimated angular velocity is defined (before and after the update) as:

$$\vec{\omega}_{k/k-1} = \vec{\omega}_m(t_k) - \vec{b}_{k/k-1} \quad \vec{\omega}_{k/k} = \vec{\omega}_m(t_k) - \vec{b}_{k/k} \quad (20)$$

Following [24], the full estimated quaternion is propagated over the interval $\Delta t_k = t_k - t_{k-1}$ as follows:

$$q_{k/k-1} = \left\{ \begin{array}{l} \exp[\frac{1}{2}\Omega(\vec{\omega}_{avg})\Delta t_k] + [\Omega(\vec{\omega}_{k/k-1})\Omega(\vec{\omega}_{k-1/k-1})] \\ -\Omega(\vec{\omega}_{k-1/k-1})\Omega(\vec{\omega}_{k/k-1})\Delta t_k^2/48 \end{array} \right\}^{q_{k-1/k-1}}$$

where the average angular velocity for this interval is approximately

$$\vec{\omega}_{avg} = \frac{\vec{\omega}_{k/k-1} + \vec{\omega}_{k-1/k-1}}{2} \quad (21)$$

The bias estimate is constant over the propagation interval $\vec{b}_{k/k-1} = \vec{b}_{k-1/k-1}$. The propagation equation for the error state covariance is

$$P_{k/k-1} = \Phi(k, k-1)P_{k-1/k-1}\Phi^T(k, k-1) + Q_k \quad (22)$$

If the average angular velocity $\vec{\omega}_{avg}$ is constant over the interval Δt_k , with magnitude ω_{avg} then the discrete system transition matrix $\Phi(k, k-1)$ can be easily calculated from Equation 18 ([18, 20]).

4.2.2 Update

When the rover stops, an absolute orientation measurement is available from the sun sensor and the accelerometer. This is used to update the estimated error state and the covariance [14]. The Kalman gain matrix is given by:

$$K_k = P_{k/k-1}H_k^T (H_k P_{k/k-1}H_k^T + R_k)^{-1} \quad (23)$$

The updated covariance and error state equations are:

$$P_{k/k} = P_{k/k-1} - K_k H_k P_{k/k-1} \quad (24)$$

$$\Delta x_{k/k} = \Delta x_{k/k-1} + K_k \Delta z(t_k) \quad (25)$$

or

$$\begin{bmatrix} \delta\vec{\theta}_{k/k} & \Delta\vec{b}_{k/k} \end{bmatrix}^T = K_k \Delta z(t_k) \quad (26)$$

where $\Delta z(t_k)$ is the measurement residual. The propagated error $\Delta x_{k/k-1}$ is zero because we have implemented

the feedback formulation of the Indirect Kalman filter. Every time we have a measurement the update is included in the full state and thus the next estimate of the error state $\Delta x_{k/k-1}$ is assumed to be zero. This update is:

$$q_{k/k} = \delta q_{k/k} \otimes q_{k/k-1} = [\delta \bar{q}_{k/k} \ 1]^T \otimes q_{k/k-1} \quad (27)$$

where

$$\delta \bar{q}_{k/k} = (1/2)\delta \bar{\theta}_{k/k} \quad \bar{b}_{k/k} = \bar{b}_{k/k-1} + \Delta \bar{b}_{k/k} \quad (28)$$

4.2.3 Observation model

Due to space limitations we omit the equations for the observation model. For a detailed derivation the interested reader is referred to [18, 20].

4.3 Backward filter

In the flow chart shown in Figure 1 we see that the robot stops every time the uncertainty grows over a preset threshold. Then the backward filter is engaged and the last attitude estimate is propagated back in time. This last estimate is very precise because it is heavily based on the absolute orientation measurements acquired when the robot stopped. While the backward filter is close to its starting point it is able to provide estimates of higher confidence than those of the forward filter. In order to derive the equations for the backward Indirect Kalman filter we start from the equations of the system for the forward case:

$$\dot{x} = Fx + Gw \quad \text{and} \quad z = Hx + v \quad (29)$$

By defining $\tau = T - t$, where τ is the backward time variable and $T = t_k - t_{k-M}$ is the time interval of smoothing, the backward system equation can be derived from:

$$\frac{dx}{d\tau} = \frac{dx}{dt} \frac{dt}{d\tau} = -\dot{x} \quad \frac{dx_b}{d\tau} = -Fx_b - Gw \quad (30)$$

Making the appropriate substitutions we get the following equation for the quaternion estimate propagation:

$$q_{b,k-1/k-1} = \left\{ \begin{array}{l} \exp[\frac{1}{2}\Omega(\bar{\omega}_{avg})\Delta t_k] + [\Omega(\bar{\omega}_{k/k-1})\Omega(\bar{\omega}_{k-1/k-1}) - \\ \Omega(\bar{\omega}_{k-1/k-1})\Omega(\bar{\omega}_{k/k-1})]\Delta t_k^2/48 \end{array} \right\}^T q_{b,k/k-1}$$

The bias propagation remains the same as before since the direction of propagation does not affect an assumed constant variable. The backward propagation equation for the covariance is now:

$$P_{b,k-1/k-1} = \Phi_b(k-1, k)P_{b,k/k-1}\Phi_b^T(k-1, k) + Q_{b,k} \quad (31)$$

No new absolute measurements are collected during the backward propagation of the filter and thus, the update equations and the observation model for the backward filter are not considered.

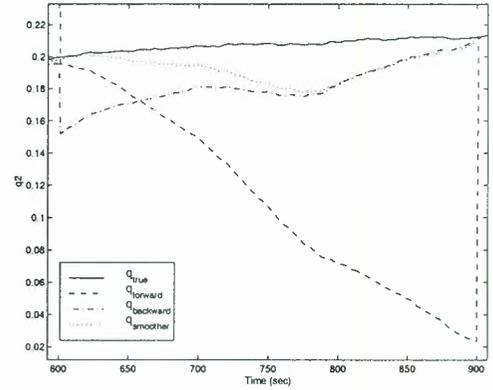


Figure 2: This is the usual outcome due to the bias estimation. The forward filter estimate drifts to the right because it has underestimated the gyro bias. The backward filter overestimates and thus drifts to the left (in the opposite direction). The smoothed estimate outperforms both filters minimizing the average estimation error.

4.4 Smoother

The smoother constructs the best estimate of the state of the system over a time period using all the measurements in that time interval [17]. In our case, the time for which the robot stops to get an absolute orientation measurement allows for post-processing and therefore application of the smoother. In order to calculate the total (smoothed) estimate we use the following equation¹:

$$P_{total}^{-1}\hat{x}_{total} = P_f^{-1}\hat{x}_f + P_b^{-1}\hat{x}_b \quad (32)$$

Each covariance matrix P_f , P_b and P_{total} represents the uncertainty of the corresponding estimate. The higher the uncertainty, the larger the covariance matrix. Equation 32 weighs each of the available estimates (from the forward and the backward filter) according to their certainty. The result is the optimal estimate possible, if all the measurements of the time interval of smoothing were available at once. The significant improvement in the quality of the 3D estimate is shown in Figure 4. Different estimated quantities calculated in a representative trial are depicted in Figures 2 and 3. The overall improvement in attitude estimation is presented in Figure 5.

5 From Attitude Estimates to Position Estimates

The accuracy of the position estimate depends heavily on the accuracy of the attitude estimate. Though the position can be calculated in real-time using the output of the forward Kalman filter we choose not to do that. Instead in our algorithm the position estimation takes place off-line as described in Figure 1. After the vehicle stops to collect

¹Applying this in 3D is somewhat involved because of the particular form of the error quaternion used. The interested reader is referred to [18] for the technical details

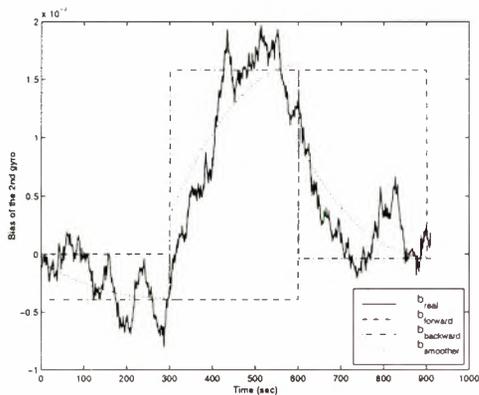


Figure 3: For the second gyro, we show the true bias value, the forward filter’s estimate, the backward filter’s estimate and the smoothed estimate of the bias. The smoothed (total) estimate stays close to the backward filter estimate for the second half of each smoothing interval while for the first part it depends on both the forward filter’s estimate and the backward filter’s estimate. This is due to the fact that the initial bias value for the backward filter is more trustworthy for this time interval than the initial value of the forward filter. The asymmetry is due to the fact that the backward filter works with an “initial” estimate which is actually computed after the motion.

an absolute orientation measurement the off-line smoothing of the attitude estimation is performed. The resulting estimate is accurate and is used to compute the current position. As we mentioned before the attitude estimate is an input to Equations 1 and 2. If the integration step is small, we can simplify this calculation as follows. First the increase in position is calculated due to the sensed acceleration and the current velocity:

$${}^L\Delta p(t_k) = {}^L v(t_k) \Delta T + {}^L a(t_k) \Delta T^2/2 \quad (33)$$

this increment is then transformed to global coordinates using ${}^G\Delta p(t_k) = {}^L A(q(t_k)) {}^L\Delta p(t_k)$, before it can be used to compute the next position using

$${}^G p(t_{k+1}) = {}^G p(t_k) + {}^G\Delta p(t_k) \quad (34)$$

The velocity increment during every measurement cycle is ${}^L\Delta v(t_k) = {}^L a(t_k) \Delta T$. In global coordinates, we have ${}^G\Delta v(t_k) = {}^L A(q(t_k)) {}^L\Delta v(t_k)$. The new velocity is

$${}^G v(t_{k+1}) = {}^G v(t_k) + {}^G\Delta v(t_{k+1}) \quad (35)$$

This result has to be transformed to local coordinates before it is fed back for the next position update:

$${}^L v(t_{k+1}) = {}^L A(q(t_{k+1})) {}^G v(t_{k+1}) \quad (36)$$

The resulting improvement in the estimation of the vehicle’s position is shown in Figure 6.

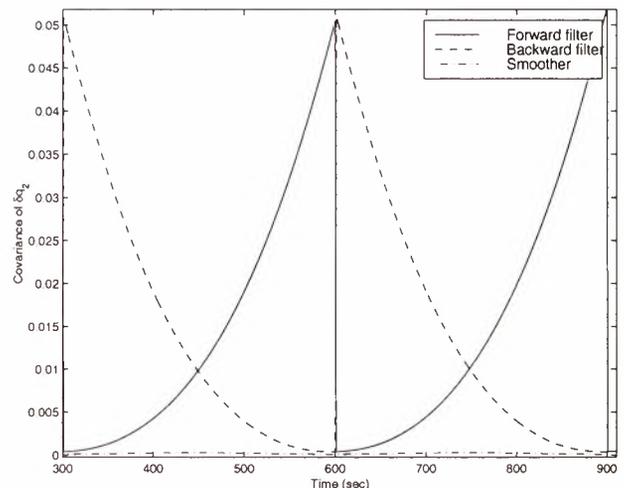


Figure 4: The covariance related to q_2 from the forward filter, backward filter and smoother is shown. At all times the total covariance is lower than either of the corresponding ones calculated from the two filters. Its value remains bounded and varies slightly during the smoothing interval.

6 Conclusion

In this paper we decomposed the localization algorithm into attitude estimation and, subsequently, position estimation. A novel approach that incorporates a smoother was presented. An Indirect (error-state) Kalman filter that incorporates inertial navigation and absolute measurements was developed for this purpose. The dynamic model was replaced by gyro modeling which relies on the integration of the kinematic equations. The error state equations for the three dimensional case were derived and used to formulate the filter’s time-variant system model and non-linear observation model. Quaternions were selected for the three dimensional attitude representation. Finally, the improvement due to the proposed method was demonstrated in simulation. Uniformly smaller values of the covariance of the estimate were sustained throughout each of the trials. It should be noted that due to the lack of vehicle specific dynamic modeling the proposed approach is general and may be used on any vehicle chassis with an equivalent set of sensors. Future directions of research include applications (extensions) of this method to cases where the INS sensors are fused with other absolute sensors that measure position (e.g. vision cues, star sensors, beacons etc.)

Acknowledgments

The authors would like to thank Dr. D. Bayard for several illuminating discussions. We also thank Dr. B. Balaram and Dr. G. Rodriguez for technical discussions.

References

- [1] B. Barshan and H. F. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, June 1995.

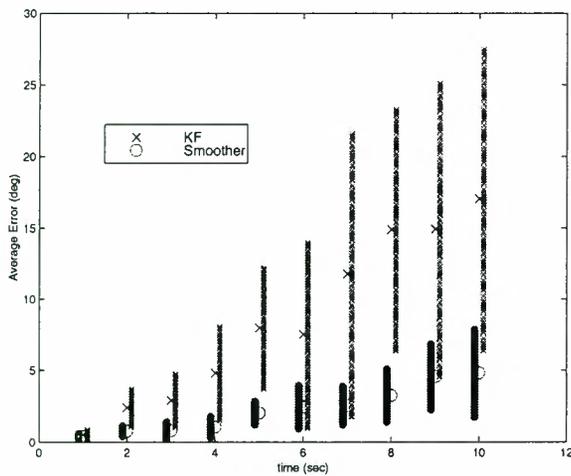


Figure 5: The average orientation error when absolute orientation measurements are available to the robot every 1, 2, ..., 10 seconds, for the case of Kalman filter (x) and smoother (o). The vertical bars represent the 3σ intervals of statistical confidence.

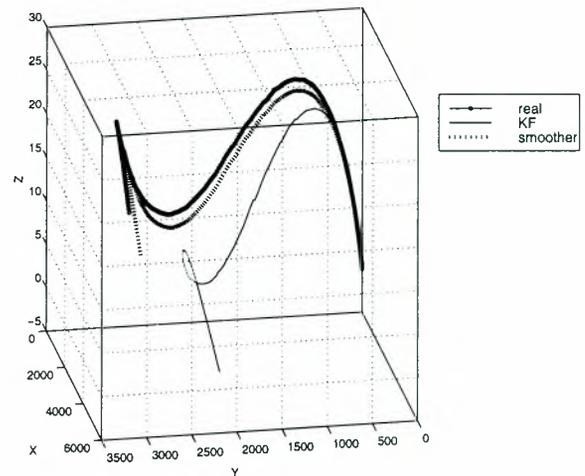


Figure 6: This is an example trajectory in 3D. The thick line (circles) represents the real motion of the vehicle. The thin line is the computed based on the attitude information from the Kalman filter, and the thick dashed is the computed using the attitude information provided by the smoother.

- [2] Ph. Bonnifait and G. Garcia. A multisensor localization algorithm for mobile robots and its real-time experimental validation. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 1395–1400, 1996.
- [3] J. Borenstein and L. Feng. Correction of systematic odometry errors in mobile robots. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, pages 569–574, 1995.
- [4] J. Borenstein and L. Feng. Gyrodometry: A new method for combining data from gyros and odometry in mobile robots. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 423–428, 1996.
- [5] J. C. K. Chou. Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics and Automation*, 8(1):53–64, Feb. 1992.
- [6] S. Cooper and H. F. Durrant-Whyte. A frequency response method for multi-sensor high-speed navigation systems. In *Proceedings of the 1994 IEEE Conference on Multisensor Fusion and Integration Systems*, 1994.
- [7] J. J. Craig. *Introduction to Robotics*, chapter 2, pages 55–56. Addison-Wesley, 2nd edition, 1989.
- [8] R. L. Farrenkopf. Analytic steady-state accuracy solutions for two common spacecraft estimators. *Journal of Guidance and Control*, 1:282–284, July-Aug. 1978.
- [9] Y. Fuke and E. Krotkov. Dead reckoning for a lunar rover on uneven terrain. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 411–416, 1996.
- [10] S. Hayati, R. Volpe, P. Backes, J. Balam, and R. Welch. Microrover research for exploration of mars. In *Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics*, University of Wisconsin, Madison, August 1-2 1996.
- [11] E. J. Lefferts and F. L. Markley. Dynamics modeling for attitude determination. *AIAA Paper 76-1910*, Aug. 1976.
- [12] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, Sept.-Oct. 1982.
- [13] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.
- [14] J. M. Mendel. *Lessons in Digital Estimation Theory*, chapter 17. Prentice-Hall, 1987.
- [15] G. C. Newton. Inertial guidance limitations imposed by fluctuation phenomena in gyroscopes. In *Proceedings of the IRE*, volume 48, pages 520–527, April 1960.
- [16] J. E. Potter and W. E. Vander Velde. Optimum mixing of gyroscope and star tracker data. *Journal of Spacecraft and Rockets*, 5:536–540, May 1968.
- [17] S. R. Reynolds. Fixed interval smoothing: Revisited. *Journal of Guidance*, 13(5), Sep.-Oct. 1990.
- [18] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Smoother based 3-d attitude estimation for mobile robot localization. Technical Report IRIS-98-363, USC, 1998. <http://iris.usc.edu/irislib/>.
- [19] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15 1999.
- [20] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Smoother-based 3-d attitude estimation for mobile robot localization. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15 1999.
- [21] R. E. Scheid, D. S. Bayard, J. (Bob) Balam, and D. B. Genery. On-board state estimation for planetary aerobots. In *12th Lighter-Than-Air Systems Technology Conference, AIAA International Balloon Technology Conference, 14th Aerodynamic Decelerator Systems Technology Conference and Seminar*, San Francisco, CA, June 3-5 1997. AIAA.
- [22] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, Oct. 1998.
- [23] J. Vaganay, M. J. Aldon, and A. Fournier. Mobile robot attitude estimation by fusion of inertial data. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 277–282, 1993.
- [24] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and Space Science Library*. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1978.

STORED IMAGE-BASED MAP BUILDING AND NAVIGATION FOR PLANETARY ROVERS

Shinichi Nakasuka, Hidenori Yamamoto and Akira Tanaka
 Department of Aeronautics and Astronautics, University of Tokyo,
 Hongo 7, Bunkyo-ku, Tokyo, 113-8656, JAPAN
 TEL:+81-3-5841-6590, FAX:+81-3-5841-8560, nakasuka@space.t.u-tokyo.ac.jp

ABSTRACT

A novel method of image-based map building and navigation is proposed for future planetary rovers. The proposed map is not a usual height map, but consists of several images of landmarks connected by the required rover actions from the rover position and orientation from which one of the landmark image is obtained to the position and orientation where another landmark image is obtained. This is similar to the way we usually remember how to get to a certain place from another place, such as "going to the position where building A is seen like this, then turning right until we can see post B and then moving forward until we can find building C", etc. It provides a simple and effective map building especially in the case when a rover wants to make many roundtrips from the key station to experimental sites. The characteristics of this method is that the navigation based on it is robust to the errors in maps as well as in the estimation of current position and orientation of the rovers. Basic concept and algorithm are described, and the results of laboratory experiments are shown and discussed.

1. INTRODUCTION

When an autonomous rover lands on a certain planet, it will be frequently required that the rover must first make a map of the surrounding world for the later navigation purpose. This is because the maps made before the landing will be usually low resolution ones generated from the observations by the orbiters, and so cannot be used for precise navigation to reach certain points within small area. What types of map should be built by what kind of instruments is a hot research issue, for which many ideas have been proposed. This paper proposes one method of such map building, based on how human being constructs a map in his brain in order to remember the way to get to a certain place which he reached before.

The problems of using the most standard maps, namely "height maps", are that the navigation based on them will be highly sensitive to the accuracy of the maps and the current position estimation of the rover. For example, if a rover wants to go from point A to point B and then turn left and go forward to point C, where A, B and C are points on the height map, the rover must know the accurate distance from point A to point B as

well as from point B to point C, and the angle between point A-B line and point B-C line, which should be estimated from the map. Moreover, during the actual moving phase, the rover should know its current position or how much it has already moved from point A, and by how much angle it has already turned. Maps made on planets and navigation system on the rover is predicted to be, however, not so accurate in many cases as to satisfy these requirements.

We, human beings, will do in completely different way to remember and use "a map" in our brain, if we don't have accurate maps or don't have any accurate methods to locate where we are now. We usually remember the way to go from one place to another in this way: "I must first go forward to the direction of that tall building A, and when I reach the post B, I must turn right until I can see the road C. I must go forward until I can see the building D like this size, then turn left to see the destination E. (Fig.1)" For this purpose, we have mental images of building A, post B, road C, building D (also with its size information), and destination E, and the required actions between them. During actually moving, we compare what we actually see now and these stored images to determine what action we should do next. This method is expected to be robust to the errors in maps as well as in the estimation of our current position and orientation.

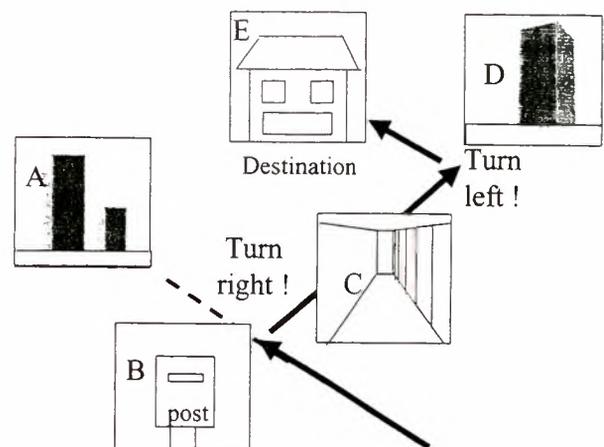


FIG.1 Our Everyday Navigation to Reach Known Place

The proposed map building and navigation method for planetary rovers is based on this simple navigation we usually do in the everyday life. This will provide very powerful and robust navigation especially when a rover

wants to make many roundtrips from the key station to experimental sites.

In the next section, this concept is explored in more detail, and some key algorithms including image processing will be shown in section 3. We have performed some laboratory experiments to verify the feasibility of this concept and evaluate the performance, which will be explained and discussed in section 4. Conclusions and future plans are summarized in section 5.

2. CONCEPT OF IMAGE BASED MAP BUILDING AND NAVIGATION

As discussed in the previous section, the problem of using the usual height maps for planetary rovers lies in that this method is highly dependent on the accuracy of the maps and the estimation of the current position or orientation of the rover, which is in many cases difficult to achieve in simple systems.

In order to compensate for these map and navigation inaccuracies, it is recommended that certain events (such as "beginning to see a certain terrain") should be utilized to trigger certain rover actions (such as "stopping moving forward" or "stopping turning right"). But if the rover only has height maps, such triggering events are difficult to define, or must be generated with much effort. Based on the observation of how we usually navigate ourselves in everyday life, we propose utilization of "images" for this triggering event; for example, the rover is programmed to stop moving forward when it sees a certain landmark in the way as stored in the memory. The size of the landmark, not only its shape, is also important if it is used for triggering, for example, the "moving closer to the landmark" action of the rover. As a result, the map consists of several images connected by the rover actions required to transit from the position and orientation where one image is seen to the other (Fig.2).

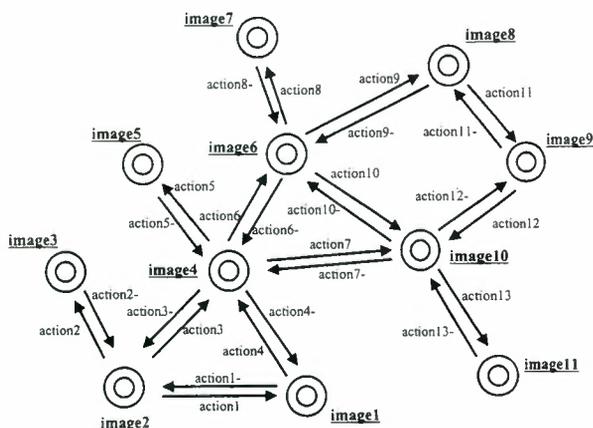


FIG. 2 Concept of Image-based Map

This navigation is highly dependent on the image matching capability of the current view with the stored images, which requires robust image processing system and algorithm. We employed a contour based matching algorithm, which is robust to the change in lighting condition and colors. The details will be given in section 3.

It would be too restrictive if the current view should perfectly match one of the stored images. The image matching subsystem has, therefore, some tolerance to such minute difference. However, if the difference between the current view and one of the stored image is small but cannot be neglected in planning of the next action (for example, "we are now a little too closer to post B to turn right to see road C"), then the system should make some adjusting actions to tune the current view to be matched with the stored one. We usually can do this kind of thing easily, because we know what kind of movement will modify the current view in the desired direction. For example, if the actual image is a little smaller than the stored image, we know that we should go nearer to the target to match the two images. This capability should also be implemented in the system.

The next question is the utility of this type of map and image based navigation. This map can be made only where the rover once visited. In other words, maps can never be made where the rover has never been to. So, is this map really useful for navigation purpose? The answer is, yes for some objectives. Maps will sometimes be used for doing autonomously roundtrip several times from the key position (where a mother-ship is located, for example) to the sites where interesting scientific observations have been made. The proposed map and navigation way will be useful for such objective. Besides, if routes to several different interesting sites have been found and stored in this map form, then these routes can be connected by a certain key position (having a certain same landmark image, such as the position of the mother-ship), so that the rover can move from any points on the map to any other points on it.

3. BASIC ALGORITHM FOR MAP BUILDING AND NAVIGATION

The basic algorithm required for these operations will be as follows.

3.1 Image based map building

When a rover takes an action (during intentional search for a destination place or by chance during random movements, etc.), and this action has been found worth memorizing because, for example, it is found to be one

leg on the route from the key station to the destination place, then the system stores the obtained image before and after the actions and actually performed action itself, in the form of the following triplet.

-
- image before action (called "Image-Before", or "IB")
 - action ("A"),
 - image after action ("IA")
-

FIG. 3 Images and Action Triplet

IB and IA are actually obtained images (such as rocks or terrain) which are considered to be candidates of "landmark images" useful for the later navigation purpose.

In order to do this, the rover should first tentatively store all these triplets during the movement into a kind of "short term memory", and when it finds the right way from a certain point (say, P) to the destination point (say, Q), then it stores only those triplets relevant to this way into the "long term memory." Fig.4 shows one example of a map built in this way. A0 through A5 show the triplets on the found sequence.

3.2 Image based navigation

Once it obtains this "one way triplet sequence," it can follow it to move from point P to point Q by taking the required actions written in the triplets in the sequence, using IA as the stopping condition of each action. To do this, the rover continually captures images and

compares them with the stored images so that it can detect the current situation (the position and orientation of the rover) or detect the stopping condition of currently taken action. Efficient and not so time-consuming image matching algorithm is indispensable for this operation.

Besides, the rover can also move from point Q to point P by following the sequence backwardly and reversing the actions written in the triplets (for example, "move forward" is changed to "move backward."), which is useful for making roundtrips.

3.3 Image adjustment mechanism

As discussed in section 2, the rover sometimes requires adjustment actions to fine-tune the current view so that it coincides with the stored image before taking an appropriate action.

This adjustment can be made using the knowledge as to the relationships between the action (such as "move forward") and how the image changes in the camera frame by this action (such as "the image of the object in front gets larger"). Hopefully, this knowledge had better be obtained autonomously by generalizing experiential data of "action and resultant image changes." This kind of learning of actions and their consequences have been studied in various context [1]-[5], but little research has been performed concerning the relationships between actions and their effect on the image changes. Application of neural network-based

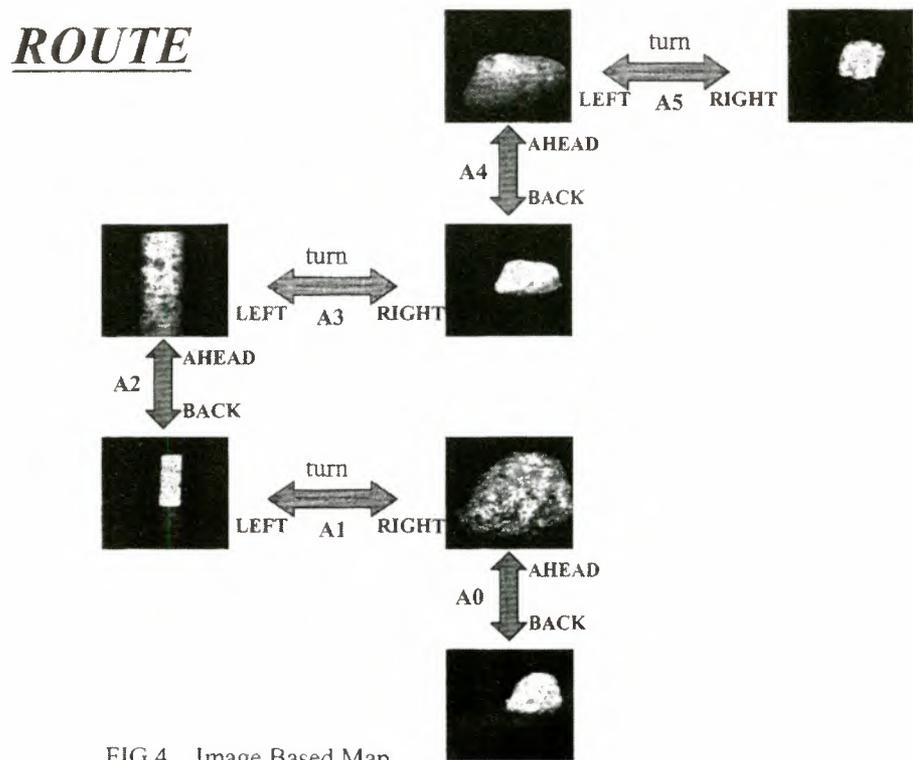


FIG.4 Image Based Map

learning mechanism is being studied in this line, with the objectives of realizing human-like capability of generalizing image changes; we human beings know from our everyday life how a certain object changes its outlook if we see it from the different angles, etc. Much research should be done for realizing it, and in our prototype system, these relationships are pre-coded in program level so that the system can decide which action to take to reduce the difference between the current and the stored images.

3.4 Connection of routes obtained differently

When the system accumulates this kind of "one way image sequences", it may happen that a certain image (say, X) of a sequence (say, P-Q) and a certain image (Y) in another sequence (R-S) almost coincides (such as in Fig.5). In such a case, the system tries to find what kind of motion(s) should be applied to reduce the difference between these two images X and Y, and stores the result in the form of the above mentioned triplet. By this process, the sequence P-Q and R-S get a linkage, and it becomes possible to move from P to R or S to Q. If the rover has some "key station" from which all the explorations start, then all the sequences get connected by way of this point.

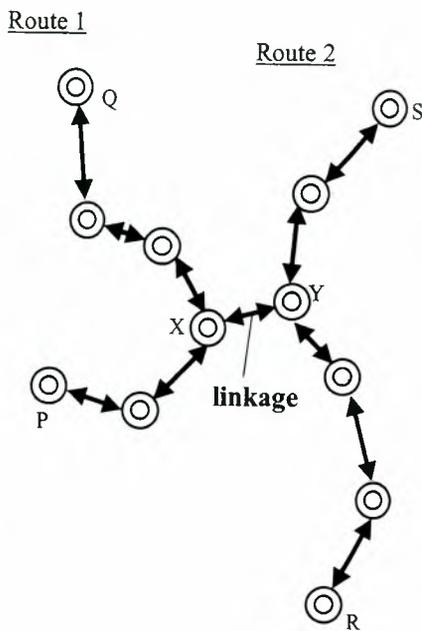


FIG.5 Connection of Differently Generated Sequences

3.5 Image matching algorithm

Image processing for matching the captured image with the stored image is the key ingredient in the overall algorithm. In our system, the following very simple yet efficient algorithm is utilized.

1) An object is extracted as a cluster of pixels from the background using the following clustering algorithm:

- 1-1) 4 by 4 pixels are considered as one "cell," whose brightness in RGB is calculated as the average of the 16 pixels
- 1-2) Cells whose brightness (sum of RGB brightness) is below a certain threshold are considered as background.
- 1-3) Horizontally neighboring cells whose brightness are above a certain threshold are integrated into one cluster
- 1-4) Vertically neighboring clusters are integrated into one cluster

2) The x-y coordinates of the contour of the cluster are calculated, with the origin of the coordinate being the C.G. of the cluster and the length being normalized using the distance from C.G. to the topmost contour point. The number of coordinate points used for matching is fixed at 50 and they are distributed on the contour at almost equal interval.

3) These 50 coordinate points of the captured image is compared with the coordinate points extracted in the same way from the stored image (Fig.6). The Euclidian distance between the corresponding contour points in two images are used to judge the similarity between these two points. Then the number of "similar" point pairs is used to judge the similarity of the two images; if this number is above a certain threshold, the two images are considered to be matched.

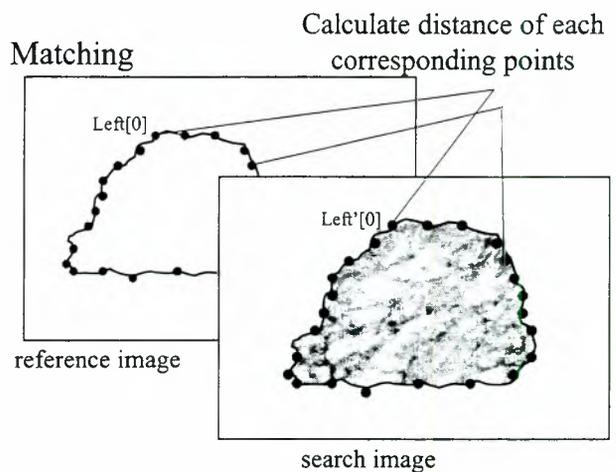


FIG.6 Image Matching Algorithm Based on Contour

4. LABORATORY EXPERIMENT

4.1 Experiment setups

In order to verify this concept, we have developed laboratory experiment system (Fig.7), in which a rover (Fig.8) with an on-board camera and a computer system

is used to experiment the above mentioned map building and navigation on the laboratory floors where many types of actual rocks are placed. (Fig.9)

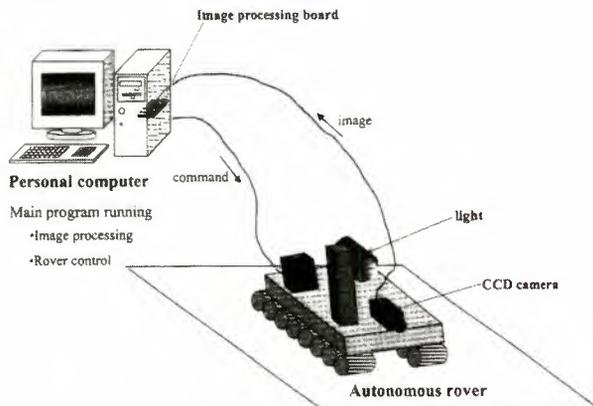


FIG.7 Experimental System Overview



FIG.8 Photo of the Used Rover



FIG.9 Experiment Setup with Rover and Rocks on Floor

The experiment is performed in the following way:

- 1) We control the rover manually from a certain point P to another point Q, during which the rover system gathers triplets. At this point, the map described in Fig.4 is obtained.
- 2) The rover is placed at point P and then it is ordered to follow the same route to reach point Q using the obtained map.
- 3) Then the rover is ordered to move backwardly from point Q to point P, also using the map.

Connection of differently obtained sequences (such as in Fig.5) has not been experimented yet.

4.2 Results of laboratory experiments

The results obtained up to now can be summarized as follows:

- (1) Once a rover finds a way from P to Q, then it can move backward from Q to P and forward from P to Q almost without failure.
- (2) The image adjustment mechanism, which are coded now in the program level, works effectively when the difference of images are rather small, but works poorly when it is large.
- (3) The employed image matching algorithm works rather well, but sometimes makes mismatch in the case that the lighting condition is severely changed from when the map is made to when the actually navigation is performed. The shadows on the rocks often become the cause for such mismatching.

We are now trying to enhance the matching algorithm, especially in terms of compensation for the change of lighting condition. More consideration is now being made as to the utilization of many, not one, objects in the image for matching, in order to make the matching algorithm more reliable. Experiment on connection of differently obtained sequences is planned to be performed shortly.

4.3 Discussions

The drawbacks of this system are (1) the system cannot accumulate information as to the region where the rover has never visited and (2) the system requires high volume of memory to store large number of images. For (1), we think that one of the important objectives of making detailed local maps will be to remember how to get to the places which have been found interesting by exploration. The image adjustment mechanism is considered important for "interpolating" between the images in the map, which will reduce the number of required images per area. We are now studying the application of neural network based learning algorithm for this objective.

The second problem will be solved to some extent by the currently very fast development of larger volume on-board memory. Effective data compression method such as JPEG/MPEG will further reduce the required memory size. In the algorithmic field, we are now also studying about storing the images not in the "image" form, but in already pre-processed form such as the list of contour points. This will reduce the flexibility about how to use the stored image, but tremendously reduce the required memory size. Trading-off between these two factors is now being made to find an appropriate coding method of the stored images.

5. CONCLUSIONS

A new map building and navigation method fully based on obtained images have been proposed, referring to how we human beings remember the way to a certain place in the everyday life using images memorized in our brain. The proposed method is expected to provide very simple and robust navigation method especially for such rovers that make roundtrip from the key station to the experimental sites many times. Laboratory experiments have shown the feasibility of the concept.

The important future works will include the enhancement of the image processing and matching algorithm, efficient and flexible coding of the stored images and application of machine learning to enhance the image adjustment functions. We are continually studying in these lines.

ACKNOWLEDGEMENT

This research is funded by "System Theory of Bio-Informatics and Its Extension towards Engineering Design Theory," which is one project of "Research for the Future of Japan Society for the Promotion of Science."

REFERENCES

- [1] S.Nakasuka, T.Yairi and H.Wajima, Automated Generation of Reflexion-based Robot Controller Using Inductive Learning, Robotics and Autonomous Systems, Vol.17 ,pp.287-305, (1996)
- [2] T.Yairi, S.Nakasuka and K.Hori, Automatic Acquisition of Reactive Behavior for Planetary Rovers, Journal of Space Technology and Science, Vol.12, No.1, pp.17-26, (1998)
- [3] S.Nakasuka, S.Shirasaka, T.Tanaka, T.Maruoka and T.Tanabe, Novel Architecture for Autonomous Generation and Management of Movement Plans of Planetary Rovers, Preprints of 13th IFAC Symposium on Automatic Control in Aerospace, pp.358-363, (1994)
- [4] T.Yairi, S.Nakasuka and K.Hori, Automatic Reactive Behavior Acquisition for Planetary Rovers, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.397-400, (1997)
- [5] S.Nakasuka and S.Shirasaka, Subsumption-based Architecture for Autonomous Movement Planning for Planetary Rovers, Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp.91-94, (1994)

LONG RANGE NAVIGATION FOR MARS ROVERS USING SENSOR-BASED PATH PLANNING AND VISUAL LOCALISATION

Sharon L. Laubach, Clark F. Olson, Joel W. Burdick, Samad Hayati

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

ABSTRACT

The Mars Pathfinder mission illustrated the benefits of including a mobile robotic explorer on a planetary mission. However, for future Mars rover missions, significantly increased autonomy in navigation is required in order to meet demanding mission criteria. To address these requirements, we have developed new path planning and localisation capabilities that allow a rover to navigate robustly to a distant landmark. These algorithms have been implemented on the JPL Rocky 7 prototype microrover and have been tested extensively in the JPL MarsYard, as well as in natural terrain.

1. INTRODUCTION

Mars sample return missions currently being planned call for rovers capable of operation for up to a year. The rovers are required to traverse up to 100m/sol and to reach ground-specified targets accurately. Lessons learned from Mars Pathfinder indicate a need for significantly increased rover autonomy in order to meet mission criteria within severe constraints including limited communication opportunities with Earth, power, and computational resources. Each rover will be working in unknown, rough terrain. Given a goal that cannot be seen from the rover's location, the rover must use its sensors to navigate safely and accurately to the goal using autonomous processes. This will require, in particular, improved motion planning and localisation algorithms.

To address the constraints upon motion planning for Mars rovers, we have developed the *RoverBug* algorithm, which can be considered a "sensorised" version of the classical Tangent Graph (or "reduced visibility graph" [2]) concept. The *RoverBug* algorithm uses two operational modes, *motion-to-goal* and *boundary following*, which interact to ensure global convergence. In addition, a "virtual" submode of

boundary following improves efficiency and handles the limited field-of-view (FOV). Motion-to-goal is typically the dominant behaviour. It directs the robot to move towards the goal using a local version of the tangent graph, restricted to the visible region. After executing the resultant subpath, motion-to-goal begins anew. This behaviour is continued until the goal is reached or the robot encounters a blocking obstacle. In the latter case, the planner switches to the boundary following behaviour.

The objective of the boundary following mode is to skirt the boundary of the obstacle, finding shortcuts where possible. Upon first detecting the blocking obstacle, the algorithm "virtually slides" along the obstacle boundary using gaze control, avoiding unnecessary motion toward the obstacle. Boundary following continues until the robot either completes a loop, in which case the goal is unreachable and the algorithm halts, or the locally visible region contains a new subpath toward the goal. In the latter case, the mode switches back to motion-to-goal. It can be shown that with these two operational modes working together, the *RoverBug* algorithm is guaranteed to reach the goal (or halt if the goal is unreachable) in finite time, is correct, and produces locally optimal (shortest-length) paths. Furthermore, *RoverBug* deals with the limited FOV of flight rovers in a manner which is efficient and minimises the need to sense and store data, using autonomous gaze control.

A complementary rover localisation algorithm is used to determine the change in the rover position by comparing terrain maps generated before and after each subpath is traversed. While the rover plans its movements, the terrain sensed by the rover cameras is compiled into a digital elevation map. After traversing the subpath generated by the planner, the rover senses the terrain through which it has just moved and generates a second terrain map that is registered to the first in order to determine the change in the

rover position. The map registration is performed by determining the relative position that optimises a maximum-likelihood similarity measure. An efficient multi-resolution search is used to determine the optimal registration without examining each position explicitly. By fitting the likelihood function that is computed with a parameterised surface, we compute subpixel localisation estimates. In addition, the uncertainty in the localisation can be estimated in order to combine the result with other sensors, for example using an extended Kalman filter.

Both RoverBug and the localisation algorithm have been implemented on the JPL Rocky 7 prototype micro-rover, a research vehicle designed to test technologies for future missions. Rocky 7, which is roughly the same size as the Sojourner rover now on Mars, has three stereo pairs of cameras for navigation: two body-mounted, and one on a deployable 1.2m mast. The implementation has been tested in the JPL MarsYard as well as in natural arroyo terrain, including traverses for tens of meters requiring multiple iterations of the motion planning and localisation algorithms. Together, these algorithms significantly augment microrovers' autonomous navigation ability, which in turn will aid in producing successful mobile robot missions.

2. PATH PLANNING

The current scenario for a rover sensing system consists of a stereo pair of cameras mounted on a mast, as well as two body-mounted stereo pairs, fore and aft. Typically, the mast cameras have a 30° to 45° field of view (FOV) and the body-mounted cameras an 80° to 100° FOV, and the "visible region" connected with these sensors sweeps out roughly a wedge, with limited downrange radius. On Rocky 7, stereo triangulation is used to generate a wedge-shaped terrain map [5]. A step/slope model [6] is used to detect obstacle pixels within this range image, and the convex hulls of distinct obstacles are computed. Next, the system "grows" the obstacles' convex hulls, accounting for the size of the rover as well as incorporating an empirically-determined safety buffer, to create the configuration space obstacles, or "C-obstacles." (See Fig.2 for an example, in this case using multiple stereo images to form a single combined "wedge" view.) If the goal lies within a C-obstacle, the obstacle's vertices are marked as goals, so an operator can designate a particular rock as a target, e.g., for later instrument placement. Each C-obstacle vertex is also labelled if it lies within another obstacle, or outside the boundaries of the current wedge.

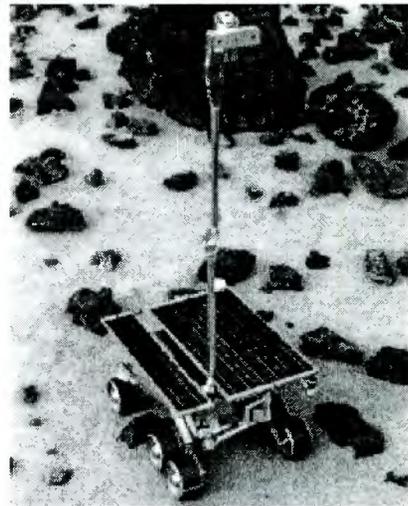


Figure 1: The Rocky7 Prototype Microrover, developed at JPL to test technologies for future missions. It is pictured here in the JPL MarsYard, an outdoor testing arena featuring simulated martian terrain.

Due to severe constraints on computational resources, the RoverBug motion planner is designed to identify the minimal number of sensor scans needed—and which specific areas to scan—to proceed at each step, while avoiding unnecessary rover motion. The planner, based upon the Wedgebug algorithm developed in [3, 4], uses a streamlined local model (the C-obstacles) which is renewed at every step, thus avoiding the issues of maintaining a global map, which taxes the limited memory available and is sensitive to registration errors. However, the algorithm does require good localisation to track the goal position and to determine whether the rover has executed a loop around an obstacle. Hence, the planner has been paired with the on-board localisation algorithm described in Section 4.

The RoverBug algorithm relies upon the construction of the local version of the tangent graph within the visible "wedge." The tangent graph consists of all line segments in freespace connecting the initial position, the goal, and all obstacle vertices, such that the segments are tangent to any obstacles they encounter. A line l is *tangent* to a C-obstacle CB at a vertex x iff in a neighborhood of x , the interior of CB lies entirely on one side of l [2]. Let $LTG(S)$ be the *local tangent graph* within the set S , defined as the tangent graph restricted to S .

The next two subsections give an overview of the operational modes of the RoverBug algorithm. More detail, including the handling of special cases—and the proofs of completeness and correctness of the

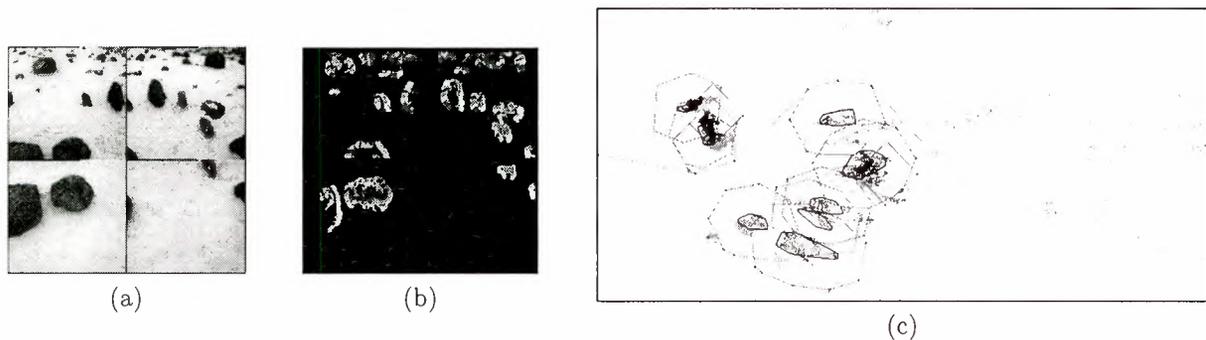


Figure 2: Results from a multi-image “wedge” view. (a) Left images from the mast-mounted stereo pair (b) Height of pixels determined using stereo triangulation (Black pixels indicate no data) (c) Overhead view of elevation map, with detected obstacles’ convex hulls and corresponding C-obstacles, and a computed subpath

“Wedgebug” algorithm underlying RoverBug—can be found in [3]. (Of note, no information, other than explicitly recorded points and parameters, is passed between steps.)

2.1. MOTION-TO-GOAL

The basic thrust of the motion-to-goal mode is monotonic progress toward the goal. At the beginning of the path sequence, an initialisation step records the parameter $d_{\text{LEAVE}} = d(A, T)$, where A is the rover’s initial position, and T is the goal. This parameter marks the largest distance the rover can stray from T during a motion-to-goal segment. A motion segment is composed of a series of steps, consisting generally of a sensing, a planning, and then an execution phase, within a single operational mode.

Each motion-to-goal step proceeds as follows: The rover (at position x) first senses a wedge, $W_0 = W(x, \vec{v}_0)$, where $\vec{v}_0 = \overrightarrow{xT}$ is the vector from x to the goal, and constructs $\text{LTG}(W_0)$. The LTG nodes comprise the convex C-obstacle vertices, the current rover position, and an optional node T_g in the direction of the goal. (Only those vertices within the visible region and on the exterior of the set of C-obstacles are used.) If there are no visible obstacles directly between the rover and the goal, T_g is added so the LTG contains a path directly towards T . The planner searches a subgraph of the LTG, $G1(W_0) = \{V \in \text{LTG}(W_0) \mid d(V, T) \leq \min(d(x, T), d_{\text{LEAVE}})\}$, for the locally optimal (shortest length) path to the goal, using an A* graph search method.

If a path is found, a subpath is generated by truncating the path at the far radius of the visible wedge (leaving an empirical buffer so the rover does not begin the next step directly behind a previously un-

sensed obstacle), and the planner returns the LTG nodes along the subpath (and the point where the path was truncated) as waypoints for the path execution algorithm.

This cycle repeats until either the rover reaches the goal, or no clear path to T exists within the visible region. If the planner detects that the rover cannot make forward progress through the current wedge, the rover must skirt a *blocking obstacle* to reach the goal. RoverBug then switches to its boundary following mode.

2.2. BOUNDARY FOLLOWING

Upon detecting a blocking obstacle O , it is clear that the rover must circumvent the obstacle in order to resume progress toward T . Unfortunately, the Rocky 7 mast is not capable of detecting obstacles reliably within roughly 1m of the vehicle, so boundary following must be accomplished using the body-mounted stereo pairs. These cameras have a limited useful range, roughly 0-1.5m from the vehicle, and cannot generally “see behind” obstacles (as can the mast cameras). Therefore, being close to an obstacle restricts the rover’s already-limited view and can result in tiny incremental steps. In order to efficiently acquire data from the robot’s current position and to avoid as much inefficient motion as possible, we add a submode of boundary following, called “virtual boundary following.”

In essence, the object of “virtual boundary following” is to swing the mast cameras back and forth in a prescribed manner, to search for the “best” place to move and begin “normal boundary following,” thus generating a local shortcut in the rover’s path. First, the planner chooses a temporary “positive” sense of

rotation by selecting the side of the blocking obstacle with the shortest path to T (which will pass outside of the visible region). Next, the rover scans the wedge $W_1 = W(x, \vec{v}_1)$, where $\angle(\vec{xT}, \vec{v}_k) = 2k\alpha_{mast}$ (α_{mast} is the half-angle subsumed by a mast wedge). That is, W_k is the wedge view centered along the vector \vec{v}_k , and k serves as the index of the angle between \vec{v}_k and the direction towards the goal. Let $\overline{W} = \bigcup^{sensed} W_k(x)$ be the conglomerate wedge composed of all of the wedge views which have been sensed so far at position x . The planner computes $LTG(\overline{W})$. We define the wedge boundary as the two rays bounding the visible region; the arc defining the downrange radius is considered interior to the wedge. If \exists a node $V \in LTG(\overline{W}) \cap \partial O$ such that $V \in \text{int}(\overline{W})$, the robot moves to V and begins “normal boundary following,” first recording two features: d_{reach} , the closest distance to T encountered so far on ∂O , and V_{loop} , the intersection of (the near side of) ∂O with the bounding ray in the “negative” direction. If there is no such node V , the planner directs the sensor to scan $W_{-1} = W(x, \vec{v}_{-1})$, constructs $\overline{W} = W_0 \cup W_1 \cup W_{-1}$, and searches the freshly expanded $LTG(\overline{W})$. In this manner, the robot scans back and forth until a suitable node is found, then travels there to begin “normal boundary following.”

“Virtual boundary following” ends when one of two events are detected:

1. $\exists V \in LTG(\overline{W}) \cap \partial O$ such that $V \in \text{int}(\overline{W})$. The robot moves to V , and begins normal boundary following.
2. The latest scanned wedge overlaps a previously scanned region (i.e., $|\angle(\vec{v}_0, \vec{v}_{last})| > \pi$). In this case, the robot is trapped by an encircling obstacle, and the algorithm halts.

“Normal boundary following” uses two views, one toward the goal and one in the direction of travel around the obstacle boundary, to determine whether a clear path towards the goal exists while the robot circumnavigates the obstacle. In this mode, at the start of each step, the rover turns toward the goal and uses its body-mounted cameras to sense W_0 , then searches $G1(W_0)$. If $T \in W_0$, the rover moves to T and the algorithm halts. Otherwise, if there is a clear path to T through W_0 , the planner directs the rover to raise its mast and image toward the goal. (Rocky 7 is unable to have its mast deployed as it moves.) Boundary following exits here if $\exists V \in G1(W_0)$ such that $d(V, T) < d_{reach}$, the *leaving condition*, in which case the planner resets d_{LEAVE} to $d(V, T)$, and begins a new motion-to-goal segment.

If neither of these conditions hold, the rover turns in the positive direction by α_{body} (the half-angle subsumed by the body-mounted cameras), senses a new wedge, and constructs the new conglomerate wedge \overline{W} . If $V_{loop} \in W(x, \vec{t}_x)$, and $V_{loop} \in$ the connected portion of ∂O containing x , the robot has executed a loop—therefore, the goal is unreachable, and the algorithm halts. Otherwise, the planner computes $V \in \partial O \cap LTG(W(x, \vec{t}_x))$ such that $d(x, V) > d(x, V') \forall V' \in \partial O \cap LTG(W(x, \vec{t}_x))$. If $V \in \text{int}\overline{W}$, the robot records d_{reach} , executes this subpath, then begins a new boundary following step. Otherwise, the rover turns again. The rover stops turning either when it has detected V_{loop} , has found a suitable point V , or has turned so far that it is overlapping an area already contained in \overline{W} , in which case the algorithm aborts.

3. TRAVERSE

The execution of each subpath in the implementation of this system on Rocky 7 is accomplished using the “Go-to-Waypoint” algorithm described in [8] as a heuristic collision avoidance mechanism. Future work will incorporate a path-execution algorithm designed to follow a series of waypoints, discarding those passed during collision avoidance manoeuvres, and able to request a replan if the rover strays too far from its computed path.

4. LOCALISATION

After a subpath has been traversed, localisation is performed in order to correct errors in dead-reckoning that have accumulated. This is accomplished by imaging the terrain through which the rover has just moved and comparing it to the map generated prior to the path planning for this subpath. Both terrain maps are generated using stereo vision on-board the rover [5].

4.1. MAP SIMILARITY MEASURE

In order to formulate the matching problem in terms of maximum-likelihood estimation, we use a set of measurements that are a function of the robot position. A convenient set of measurements are the distances from the occupied cells in the local map to their closest occupied cells in the global map. Denote these distances D_1^X, \dots, D_n^X for the robot position X . The likelihood function for the robot position can be formulated as the product of the probability densities of these distances. For convenience, we work in the $\ln L(X)$ domain:

$$\ln L(X) = \sum_{i=1}^n \ln p(D_i^X)$$

For the uncertainty estimation to be accurate, it is important that we use a probability density function (PDF) that closely models the sensor uncertainty. This can be accomplished using a PDF that is the weighted sum of two terms:

$$p(D_i^X) = \alpha p_1(D_i^X) + (1 - \alpha) p_2(D_i^X)$$

The first term describes the error distribution when the cell is an inlier (in the sense that the terrain position under consideration in the local map also exists in the global map). In this case, D_i^X is a combination of the errors in the local and global maps at this position. In the absence of additional information with respect to the sensor error, we approximate $p_1(D_i^X)$ as a normal distribution:

$$p_1(D_i^X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(D_i^X)^2/2\sigma^2}$$

The second term describes the error distribution when the cell is an outlier. In this case the position represented by the cell in the local map does not appear in the global map. This may be due to range shadows that were present when the global map was constructed or outliers that are present in the range data when the local map is constructed. In theory, this term should also decrease as D_i^X increases, since even true outliers are likely to be near some occupied cell in the global map. However, this allows pathological cases to have an undue effect on the likelihood for a particular robot position. In practice, we have found that modeling this term as a constant is both convenient and effective:

$$p_2(D_i^X) = K$$

4.2. SEARCH STRATEGY

A multi-resolution search strategy is used to determine the most likely robot position [1, 7]. This method is guaranteed to locate the optimal position in the discretised search space. The pose space is first discretised at the same resolution as the occupancy grids so that neighboring positions in the pose space move the relative positions of the grids by one grid cell. We then test the nominal position of the robot given by dead-reckoning so that we have an initial position and likelihood to compare against. Next, the pose space is divided into rectilinear cells. Each cell is

tested to determine whether it could contain a position that is better than the best position found so far. Cells that cannot be pruned are divided into smaller cells, which are examined recursively. When a cell is reached that contains a single position in the discretised pose space, then this position is tested explicitly.

To determine whether a cell C could contain a pose superior to the best found so far, we examine the pose c at the center of the cell. A bound is computed on the maximum distance between the location to which a cell in the local map is transformed by c and by any other pose in the cell. We call this distance Δ_C . For the space of translations, Δ_C is simply the distance between c and any corner of the cell. To place a bound on the quality of any position within the cell, we bound each of the distances that can be achieved by features in the local map over the cell. This is done by subtracting the maximum change of the cell, Δ_C , from the distance achieved at the center of the cell, D_i^c :

$$D_i^C = \max(D_i^c - \Delta_C, 0)$$

The values obtained are then propagated through the likelihood function to bound the score that can be achieved by any position in the cell.

$$P_i^C = \ln p(D_i^C)$$

P_i^C is now the maximum score that the i th feature of the local map can contribute to the likelihood for any position in the cell.

A bound on the best overall likelihood that can be found at a position in the cell is given by:

$$\max_{X \in C} \ln L(X) \leq \sum_{i=1}^n P_i^C$$

If this bound does not surpass the best that we have found so far, then the entire cell is pruned from the search. Otherwise, the cell is divided into two cells by slicing it along the longest axis and the process is repeated recursively on the subcells.

4.3. SUBPIXEL LOCALISATION

Using this probabilistic formulation of the localisation problem, we can estimate the uncertainty in the localisation in terms of both the variance of the estimated positions and the probability that a qualitative failure has occurred. Since the likelihood function measures the probability that each position in the pose space is the actual robot position, the uncertainty in the localisation is measured by the rate at which the likelihood function falls off from the peak.

In addition, we can perform subpixel localisation in the discretised pose space by fitting a surface to the peak that occurs at the most likely robot position.

We assume that the likelihood function can be approximated as a normal distribution in the neighborhood around the peak location. Fitting such a normal distribution to the computed likelihoods yields both an estimated variance in the localisation estimate and a subpixel estimate of the peak location. While the approximation of the likelihood function as a normal distribution may not always be ideal, it yields a good fit to the local neighborhood around the peak and our experimental results indicate that very accurate results can be achieved under this assumption.

In addition to estimating the uncertainty in the localisation estimate, we can use the likelihood scores to estimate the probability of a failure to detect the correct position of the robot. This is particularly useful when the terrain yields few landmarks or other references for localisation and thus many positions appear similar to the robot.

4.4. TARGET SELECTION

Prior to performing localisation, the rover analyses the terrain in the map generated at the initial rover position in order to select a *localisation target*. This target is the position in the terrain that the rover looks at in order to generate a new map to match against the previously generated map. We want to select a location that has very distinctive terrain and that allows the localisation to be performed with the smallest uncertainty.

The localisation target is determined by estimating the amount of error present in the map computed at the initial rover position as well as the amount of error that would be generated by imaging the terrain from the final rover position. These errors are encoded in a probability map of the terrain expected to be seen from the final rover position. Each cell in this map contains an estimate of the probability that the cell will be seen as occupied by the rover. By treating this probability map as a terrain map and comparing it to the map generated at the initial rover position, we can predict the uncertainty that will occur in the localisation for any target that the rover may look at to use for terrain matching. The location with the lowest predicted uncertainty is selected as the localisation target.

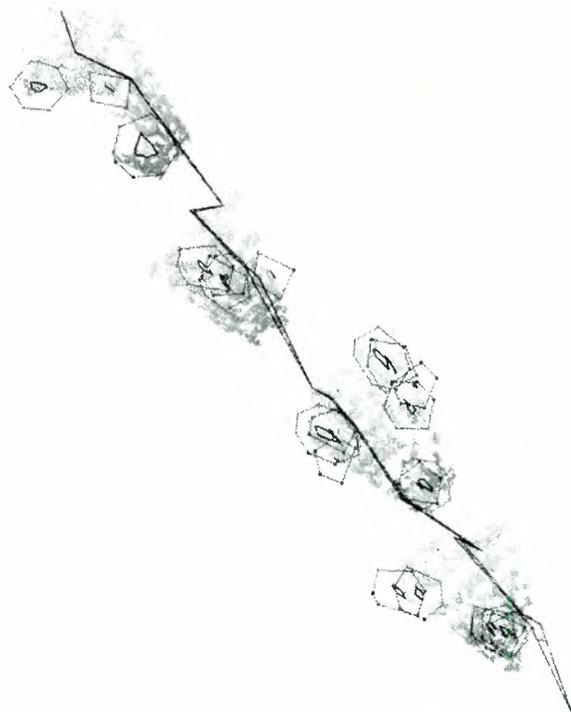


Figure 3: Results from a multi-step run in the JPL MarsYard. The path begins in the lower right corner of the image, toward a goal approx. 21m distant in the upper left. Each (single-image) wedge depicts a rangemap produced from mast imagery, and extends roughly 5m from the imaging position. The obstacles are marked by a black convex hull, and a grey C-obstacle. Each subpath ends with an apparent “jag” in the path; these are not in fact motions, but rather the result of the localisation procedure run at the conclusion of each step. The second line echoing the path is the rover’s telemetry for the run.

5. RESULTS

The implementation of the RoverBug and localisation algorithms on the Rocky 7 prototype Mars rover has been tested in the JPL MarsYard and in natural terrain, for traverses up to tens of meters requiring several iterations of both algorithms. The basic scenario is as follows: the rover is situated in unknown, rough terrain. The remote human operator designates a goal, which is generally outside the range of the rover’s sensors, and sets in motion the autonomous navigation system. The system begins by directing the mast to image towards the goal, generating data for both the localisation and motion planning algo-

rithms. The RoverBug algorithm searches the resulting LTG, and directs the mast to look in the appropriate direction(s) to produce the first subpath. Upon the traversal of the first subpath, the localisation algorithm corrects the rover's position estimate. The cycle repeats, and the system incrementally builds and executes each subpath until the goal is reached.

Fig. 2 demonstrates the generation of a path segment from a multiple mast images, treated as a single "wedge" view. The generated path skirts all of the obstacles and achieves the goal using data from all four of the stereo pairs.

Fig. 3 shows the results of one typical run in the MarsYard. The goal was approximately 21m distant from the initial position, and the radius of each wedge was 5m. The obstacles' convex hulls and silhouettes are computed within each wedge view, and a subpath generated, which is executed before the next wedge view is taken. The steps of the localisation algorithm straddle each path-planning cycle, generating an updated position estimate after the execution of each subpath. The resultant multi-step path runs from lower right to upper left.

6. SUMMARY

The specifications for autonomous rovers for the currently planned Mars missions place strenuous requirements on the rovers' ability to traverse long distances to ground-specified targets safely and accurately. A system able to achieve accurate long-range navigation through planetary terrain is described, which combines sensor-based motion planning and visual localisation. Results from the Rocky 7 prototype rover are presented, which demonstrate good performance of the system.

ACKNOWLEDGEMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work is an element of the Long Range Science Rover project, which is developing technology for future Mars missions. This project is funded as part of the NASA Space Telerobotics Program, through the JPL Robotics and Mars Exploration Technology Office.

References

- [1] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 705–706, 1993.
- [2] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [3] S. L. Laubach. *Theory and Experiments in Sensor-Based Motion Planning with Applications for Flight Planetary Microrovers*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1999.
- [4] S. L. Laubach, and J. W. Burdick. An autonomous sensor-based path-planner for planetary microrovers. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1999.
- [5] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8(1):71–91, July 1992.
- [6] L. Matthies, A. Kelly, T. Litwin, and G. Tharp. Obstacle detection for unmanned ground vehicles: A progress report. In *Proceedings of the 1995 Intelligent Vehicles Symposium*, pages 66–71, 1995.
- [7] C. F. Olson and L. H. Matthies. Maximum-likelihood rover localization by matching range maps. In *Proceedings of the International Conference on Robotics and Automation*, pages 272–277, 1998.
- [8] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev. The Rocky7 Mars rover prototype. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 1996.
- [1] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the Hausdorff dis-

**Intelligent Planning and Control of
Spacecraft Operations**

EFFICIENCY AND FAIRNESS WHEN SHARING THE USE OF A SATELLITE

Nicolas Bataille
Centre National d'Études Spatiales
18, avenue Édouard Belin, 31055 – Toulouse Cedex – France

Michel Lemaître and Gérard Verfaillie
ONERA CERT
2, avenue Édouard Belin (BP 4025), 31055 – Toulouse Cedex – France

ABSTRACT

Resources co-funded by several agents must be exploited in such a way that three kinds of constraints are met: (1) physical problem (hard) constraints; (2) efficiency constraints, aiming at maximizing the satisfaction of each agent; (3) a fairness constraint, which is ideally satisfied when each agent receives an amount of the resource exactly proportional to its financial contribution. This paper investigates a decision problem for which the common property resource is an earth observation satellite. The problem is to decide on the daily selection of a subset of pictures, among a set of candidate pictures which could be taken the next day considering the satellite trajectory. This subset must satisfy the three kinds of constraints stated above. Although fair division problems have received considerable attention for a long time, especially from microeconomists, this specific problem does not fall entirely within a classical approach. This is because the candidate pictures may be incompatible, and because a picture is only of value to the agent requesting it. As in the general case, efficiency and fairness constraints are antagonistic. We propose three ways for solving this share problem. The first one gives priority to fairness, the second one to efficiency, and the third one computes a set of compromises.

1 INTRODUCTION

Due to their cost, large research or industrial projects are often co-funded by several agents (countries, companies, entities ...). Space projects such as earth observation satellites, space stations or space probes are good examples. Once constructed and made operational, the common property resource must be exploited and shared in a way which satisfies three kinds of constraints :

- *physical* constraints: the exploitation of the resource must obey hard constraints;
- *efficiency* constraints: each agent wants to get the highest possible satisfaction in return;
- a *fairness* constraint: each agent must get a return on investments proportional to its financial

contribution to the project; the better the proportionality of returns is achieved, the more the share quality improves.

The first kind of constraints must absolutely be met (hard constraints) whereas the two others are preference constraints (soft constraints). As it can be easily guessed, the efficiency and fairness constraints are antagonistic: the search for a perfect share may lead to poorly efficient decisions, and conversely, decisions which maximize the global satisfaction of agents are often unfair. So, a compromise between the best satisfaction of both constraints must be found.

The usual case involving only one agent (in which case there is no share problem) is a difficult combinatorial discrete optimization problem (NP-hard). Nevertheless, it is a perfectly well stated problem. The multiagent case is also a discrete combinatorial problem, but is actually a multi-objective optimization problem [4]; the first difficulty arises when searching for a meaningful and principled definition of a good compromise between efficiency and fairness.

This article sums up a study, the aim of which was to propose methods to solve a specific share problem, namely the fair and efficient exploitation of an earth observation satellite owned in common by several agents. It is organized as follows. The next section sets the problem more formally. Then we present three quite different methods devoted to the resolution of this share problem. These methods have been simulated on the basis of the expected data for the future Spot5 satellite. The section after reports these simulations. Lastly, we state our conclusions.

2 AN EARTH OBSERVATION SATELLITE SCHEDULING AND SHARING PROBLEM

The studied problem is the following: an earth observation satellite, co-funded by several agents, is exploited in common. These agents make daily requests for pictures they would like to be taken by the satellite. Roughly speaking, the problem consists in selecting each day, among the set of candidate pictures which could be taken the next day considering

the satellite trajectory, a subset of pictures which satisfies all the physical constraints, maximizes the satisfaction of the agents, and respects as far as possible a fairness constraint. Such a selection will be called a *decision*. The satisfaction and fairness constraints will be taken into account over a fixed interval of several days.

Let us describe more formally the problem. First, the data:

- there are n agents; in our real-world problem, n is typically in the range 3 to 8;
- D_{ik} is the set of pictures requested by the agent i for the day k ; let

$$D_{\bullet k} \stackrel{\text{def}}{=} \bigcup_{i=1}^n D_{ik}; \quad (1)$$

the size of a $D_{\bullet k}$ is averaging 200;

- the sets D_{ik} are pairwise disjoint : $D_{ik} \cap D_{jk} = \emptyset$ for all i, j ;
- each picture in $D_{\bullet k}$ could be taken the day k , but all pictures cannot be taken because there are incompatibilities between them: some physical hard constraints must be met (for example no more than m pictures can be taken at once, provided there are only m instruments on board; a transition time between two pictures taken by the same instrument must be respected; on board memory is limited ...); a subset $X \subseteq D_{\bullet k}$ is said *admissible* if all pictures in X satisfy hard constraints (pictures are compatible) and hence can all be taken the considered day;
- $w(x)$ is the weight of the picture x ; it is freely set by the agent requesting the picture, and reflects its importance for the agent;
- $\mathbf{q} = (q_1, q_2, \dots, q_n)$, with $\sum_{i=1}^n q_i = 1$ is the quota vector: q_i is proportional to the financial investment of the agent i .

We characterize now the decisions that we are looking for. Each day $k - 1$, the demands D_{ik} , with corresponding weights, are collected and we must compute the sets of pictures A_{ik} which will be shot for the agent i the day k . These A_{ik} are such that:

- $A_{ik} \subseteq D_{ik}$ (note that A_{ik} are disjoint);
- let

$$A_{\bullet k} \stackrel{\text{def}}{=} \bigcup_{i=1}^n A_{ik}; \quad (2)$$

$A_{\bullet k}$ must be admissible;

- the *cumulative satisfaction* of each agent, measured over a given interval of days I ending on the day k must be as high as possible (efficiency constraints); the satisfaction of the agent i the day k is measured by the quantity $s(A_{ik})$ where

$$s(X) \stackrel{\text{def}}{=} \sum_{x \in X} w(x); \quad (3)$$

hence, the cumulative satisfaction over I for the agent i is

$$cs_i \stackrel{\text{def}}{=} \sum_{k \in I} s(A_{ik}); \quad (4)$$

n	number of agents.
i	agent index.
k	day index.
I	interval of days on which satisfactions and costs are taken into account.
D_{ik}	pictures requested by agent i day k .
A_{ik}	pictures obtained by agent i day k .
\mathbf{q}	quota vector. $\mathbf{q} = (q_1, q_2, \dots, q_n)$.
$w(x)$	weight of picture x .
$s(X)$	satisfaction for an agent receiving the set X of pictures. See eq. 3.
cs_i	cumulative satisfaction. See eq. 4.
cs_i^M	cumulative maximal satisfaction. See eq. 7.
$c(x)$	cost of picture x .
cc_i	cumulative cost. See eq. 16.
\mathbf{cc}	vector of cumulative costs. See eq. 17
j	quality of share criterion. See eq. 21
gcs	global cumulative satisfaction. See eq. 22

Table 1: Main symbols used in this paper.

These satisfactions need to be normalized over agents, if we compare or aggregate them.

- the “quality of the share” over I (to be formalized later) must be as high as possible (fairness constraint).

The problem above is stated as a sequence of multi-objective optimization problem instances. However, the fairness constraint is not yet formally stated. We have investigated three quite different methods devoted to the resolution of this share problem (that is general schemes for computing the A_{ik}). Each one is based on a particular way of taking into account the fairness constraint and the necessary compromise with the efficiency constraints. The first two methods reduce the problem to a sequence of mono-objective optimization problem instances, whereas the third one keeps the multi-objective aspect.

3 FAIRNESS FIRST

The first method searches for fairness first, and then for efficiency. The entitlement to use the resource is shared by allocating observation windows to each agent in turn. Observation windows are merely sequences of successive orbits of the satellite. Each day, the agent i is given the right to freely exploit about $q_i \cdot N$ orbits, where N is the number of orbits daily covered by the satellite. Observation windows can be assigned to agents on the basis of a fixed repetitive procedure. This procedure and the trajectory of the satellite are such that each agent gets opportunity to shoot any place in the world within a bounded number of days.

Following this method, the whole problem can be cast into a set of optimization problem instances, one for each agent each day, because each agent knows in advance his time windows. Assuming that each $x \in D_{ik}$ belongs to the window assigned to agent i the day k , the successive optimization problem instances

consist in maximizing the satisfaction of agents by finding

$$A_{ik} = \operatorname{argmax}\{s(X)|X \subseteq D_{ik}, X \text{ admissible}\}. \quad (5)$$

This problem can be seen as a combination of discrete *constraint* and *optimization* problems. General frameworks such as the *Semiring* and *Valued Constraint Satisfaction Problems* frameworks [1; 9] have been recently designed to capture such mixed problems. Powerful complete and incomplete algorithms, associated to these frameworks, are now available, and research in this area is very active [3; 11; 10; 5].

Our simulations are based on the Valued CSP framework. Almost all windows can be solved to optimality, using a sophisticated algorithm. These simulations show for this method a very good quality share: the number of pictures effectively selected and assigned to each agent is very close to a number proportional to its quota. But the decisions are clearly inefficient, when compared with those resulting from the two following methods, as reported in section 6.

4 EFFICIENCY FIRST

The second method considers the opposite view: first efficiency, fairness if possible. It is based on three main ideas:

1. for efficiency, maximize each day a linear combination of individual satisfactions of the agents;
2. for fairness, choose this combination in a way favoring the fairness constraint;
3. check that each agent has obtained a *fair share*.

The last point is borrowed from the literature on fair division [12; 7; 2]: in this method, we postulate that a decision is fair when each agent receives at least a minimal fair share, defined for the agent i as q_i times the satisfaction it would get if it were the only user of the resource. More formally, the fairness constraint is considered to be satisfied if

$$cs_i \geq q_i \cdot cs_i^M, \quad i = 1, \dots, n, \quad (6)$$

$$\text{with } cs_i^M \stackrel{\text{def}}{=} \sum_{k \in I} s^M(D_{ik}), \quad (7)$$

$$s^M(D_{ik}) \stackrel{\text{def}}{=} \max\{s(X)|X \subseteq D_{ik}, X \text{ admissible}\}. \quad (8)$$

We now turn to the determination of the linear combination of individual satisfactions to be maximized (points 1 and 2). For the moment, assume that the quotas are equal (all agents have equal rights over the common resource). The clue is to consider the weights of pictures as monetary bids. As a first approach, we could select pictures in such a way that the sum of bids for selected pictures, namely

$$s(A_{\bullet k}) = \sum_{x \in A_{\bullet k}} w(x) = \sum_{i=1}^n s(A_{ik}) \quad (9)$$

is maximum (under admissibility constraints). In this way, the higher the bid for a picture is, the more this picture gets some chance of being selected.

But recall that weights are freely fixed by agents. The above function to be maximized must be corrected, in order to take this fact into account. In other words, we must make satisfactions comparable by *normalizing* them. So instead, the function to be maximized will be:

$$s'(A_{\bullet k}) \stackrel{\text{def}}{=} \sum_{i=1}^n s'(A_{ik}) \stackrel{\text{def}}{=} \sum_{i=1}^n \alpha_{ik} \cdot s(A_{ik}), \quad (10)$$

where the coefficients α_{ik} have to be determined. The principle of the normalization is the following : *the maximum of normalized individual satisfaction that an agent would get if it were the only user of the resource is equal for all agents*. Formally stated :

$$s'^M(D_{ik}) = 1, \forall i, k, \quad (11)$$

with

$$s'^M(D_{ik}) \stackrel{\text{def}}{=} \max\{\alpha_{ik} \cdot s(X)|X \subseteq D_{ik}, X \text{ admissible}\}. \quad (12)$$

Obviously we have $s'^M(D_{ik}) = \alpha_{ik} \cdot s^M(D_{ik}) \forall i, k$, hence $\alpha_{ik} = 1/s^M(D_{ik})$.

We must now adapt this normalization to the situation where the agents are entitled to different fractions of the resource (non uniform quotas). The way to do this is simple (see for example [2, section 2.8]): suppose that we have three agents, with quotas $\mathbf{q} = (1/10, 3/10, 6/10)$. This is equivalent to an equal division between a society of 10 fictitious agents, followed by two groupings of 3 and 6 shares for our last two real agents. This argument leads to an adaptation of the previous normalized individual satisfaction: let state

$$s'^M(D_{ik}) = \alpha_{ik} \cdot s^M(D_{ik}) = q_i \quad (13)$$

instead of 1 as in equation 11, hence $\alpha_{ik} = q_i/s^M(D_{ik})$. To sum up, the set of daily selected pictures with this method maximizes the function

$$s'(A_{\bullet k}) = \sum_{i=1}^n q_i \cdot \frac{s(A_{ik})}{s^M(D_{ik})} \quad (14)$$

under admissibility constraints.

With this choice for the coefficients α_{ik} , it is not difficult to see that the selected decisions are independent of the scale of weights used by each agent (in other words, the preference order induced by s' over potential decisions is not changed if some agents multiply their weights by a constant factor). However, the method does not guarantee the satisfaction of the fairness constraint (counter-examples can be easily built). This constraint will have to be checked *a posteriori*. Hopefully, it has a lot of chance to be satisfied, for two reasons :

1. a structural reason : the normalization of the weights tends to favor agents with upper quotas, in a direction favorable to the satisfaction of the fairness constraint; moreover, the fairness constraint is rather soft;
2. a statistical reason : when there is a large number of candidate pictures, not too tightly incompatible, the structural reason can exert its influence;

this is the case with our (realistic) simulation data: the simulations show that the fairness constraint is always widely satisfied (see results in section 6).

The function maximized being a linear combination of individual satisfactions of the agents, decisions selected by this method are Pareto-optimal decisions¹ in the n -dimensional space of individual satisfactions. Such decisions are also called *efficient* decisions. It is impossible to improve a decision selected by this method for one agent without reducing the satisfaction of at least another agent. This property explains the good satisfaction levels obtained with this method in our simulations and justifies the name “efficiency first”.

We have designed a variant of this method, for the case where the fairness constraint would not be satisfied, when requests are poorly distributed and highly incompatible. This variant is inspired by the classical Knaster’s procedure of sealed bids [2, section 3.2], [12, section 8.2]. We compute each day fictitious monetary compensations between agents, reflecting the gap between the actual and ideal shares. An agent having a positive credit is “late” on its quota (it received not enough pictures selected) and conversely, an agent with a negative credit is “ahead” on its quota. These compensations are used to modify the above normalization procedure for the next days in a direction favorable to a fairest share.

This method and its variant can be implemented successfully using the same Valued CSP framework as before. However, the number of instances to be solved is large (all the $s^M(D_{ik})$ must be computed) and the size of the whole instance (for the maximization of $s'(A_{\bullet,k})$) may be very important. Our simulations show that an optimal decision can be computed almost all days in a reasonable amount of time. For very large instances, we have to turn to local search procedures (descent search or simulated annealing).

5 COMPROMISES BETWEEN FAIRNESS AND EFFICIENCY : A MULTI-CRITERIA APPROACH

The third approach does not focus on fairness or efficiency, but computes a set of good compromise decisions. The aim is to help a human decision-maker to take decisions, by providing this decision-maker with interesting compromises.

The most precise way to set the whole problem is to formulate it as a sequence of multi-criteria discrete optimization problems. The criteria to be maximized would be:

- the n agent’s satisfaction criteria $cs_i, i = 1, \dots, n$,
- a criterion j measuring the quality of share, to be defined.

Only the set of Pareto-optimal decisions in this $n + 1$ dimensional space are worth considering. The approach which would consist of collecting this set of decisions is unworkable, because it is very large (in

our application). A straightforward idea is to select the fairest decision within the set of efficient decisions (see for example [6, page 14]). It is as well unworkable because the number of potential decisions is too large to allow exhaustive search.

So, we have to resign ourselves to aggregate some criteria. A sensible solution is to aggregate individual satisfactions into a global cumulative satisfaction gcs , and to keep apart the quality of share criterion j . Eventually, potentially interesting decisions will be presented in the two-dimensional space $j \times gcs$.

5.1 Measuring the quality of share

It is questionable to base the quality of share upon the individual satisfactions obtained by agents, because these satisfactions are not expressed in a common scale, and hence are difficult to compare. A better idea is to base our measure upon some function of the *real cost* of pictures, such as time, memory or power consumption on board. Let $c(x)$ be the cost of the picture x . The cost function is supposed to be independent of the agent requesting the picture, and to have been fixed by mutual agreement between agents. Let

$$c(A_{ik}) \stackrel{\text{def}}{=} \sum_{x \in A_{ik}} c(x), \quad (15)$$

$$cc_i \stackrel{\text{def}}{=} \sum_{k \in I} c(A_{ik}), \quad (16)$$

$$\text{and } \mathbf{cc} \stackrel{\text{def}}{=} (cc_i)_{i=1}^n. \quad (17)$$

The last quantity is just the vector of cumulative costs of pictures selected for the agents over the interval I . We propose to measure the quality of share over I by a “distance” between \mathbf{cc} and \mathbf{q} , the quota vector.

Microeconomists have developed a rich set of *inequality indices* (see for example [6, section 2.6]), that we can use to base our function j measuring the quality of share. The popular Gini indice

$$G(\mathbf{u}) \stackrel{\text{def}}{=} \frac{1}{2n^2\bar{u}} \sum_{1 \leq i, j \leq n} |u_i - u_j| \quad (18)$$

measures the inequality resulting from a vector of utilities $\mathbf{u} = (u_1, \dots, u_n)$. \bar{u} is the average value of the u_i . It can be generalized to the non-uniform case to fit our needs, using an argument similar to the one given in section 4, in the following way:

$$G'(\mathbf{u}, \mathbf{q}) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{1 \leq i, j \leq n} |\bar{u}_i \cdot q_j - \bar{u}_j \cdot q_i| \quad (19)$$

$$\text{with } \bar{u}_i = \frac{u_i}{\sum_{j=1}^n u_j}. \quad (20)$$

Taking

$$j \stackrel{\text{def}}{=} 1 - G'(\mathbf{cc}, \mathbf{q}) \quad (21)$$

finishes the job. We have $0 \leq j \leq 1$, and $j = 1$ when the share is perfect (costs of obtained pictures exactly proportional to quotas).

¹A Pareto-optimal decision always beats any other decision on at least one criterion.

5.2 Aggregating individual satisfactions

As a measure of the global cumulative satisfaction of agents over the interval I , we choose a linear combination of normalized cumulative individual satisfactions :

$$gcs(A_{\bullet k}) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \frac{cs_i}{cs_i^M} \quad (22)$$

It has the following properties : $0 \leq gcs \leq 1$ (the maximum 1 is reached when each agent is satisfied as much as it can be if it were the only owner of the resource); gcs is independent of the individual scales of weights; it is independent of quotas².

5.3 Computing decisions

This method is very costly in term of computational resource. The set of Pareto-optimal decisions in the $j \times gcs$ space can be computed exactly by a branch-and-bound search, or approached by an adapted local search method when the search space is too large.

6 SIMULATIONS

We have used data from the simulated demand concerning the future Spot5 satellite, which will carry three cameras on board. This data, provided originally for the mono-agent case, has been adapted to simulate a demand from $n = 3$ agents. Simulated agents request each day about the same number of pictures. The quota vector for the simulation is $(0.1, 0.3, 0.6)$. Weights are in the range 1 to 100. We dispose of data for 371 days. The most loaded day comprises 427 requested pictures and 18878 binary and ternary admissibility constraints. The cost function is simply $c(x) = 1, \forall x$ (that is, we only count the number of selected pictures). The interval of days I on which cumulative satisfaction and cost functions are based is always $I = [1 \dots k]$, where k is the present day.

agent i		1	2	3
	cs_i^M	112499	112210	115860
Fairness First	cs_i	13196	34078	66819
	cc_i	585	1750	3405
	cc_i (%)	10.2	30.5	59.3
Efficiency First	cs_i	36491	58036	100408
	$q_i \cdot cs_i^M$	11250	33663	69516
	cc_i	1725	3268	5146
	cc_i (%)	17.0	32.2	50.8
Multi-Criteria ($\lambda = 9$)	cs_i	33928	54075	90472
	cc_i	1296	2773	5156
	cc_i (%)	14.0	30.1	55.9

Table 2: Simulation results for the three methods.

The table 2 sums up the numerical results obtained from the simulation. It gives the cumulative

²This option is questionable but seems rather sensible, because we consider that the satisfactions of agents are of equal importance, even if they are entitled with different rights. Note that the quota vector is taken into account by the quality of share.

satisfaction and cost of pictures obtained by agents with each method over the whole simulation interval $I = [1 \dots 371]$. Cumulative satisfactions cs_i should be compared with the maximal possible cumulative satisfactions cs_i^M for each agent, given on the second line.

For the Efficiency First method, we give the minimal fair shares $q_i \cdot cs_i^M$. As it can be seen, the fairness constraint (equation 6) is widely satisfied.

For these results, we have simulated a restricted form of the Multi-Criteria method : instead of building a complete set of non-dominated decisions in the $j \times gcs$ plane, we only look for a decision close to the line of slope $\lambda = 9$ from the (1,1) point (see figure 1).

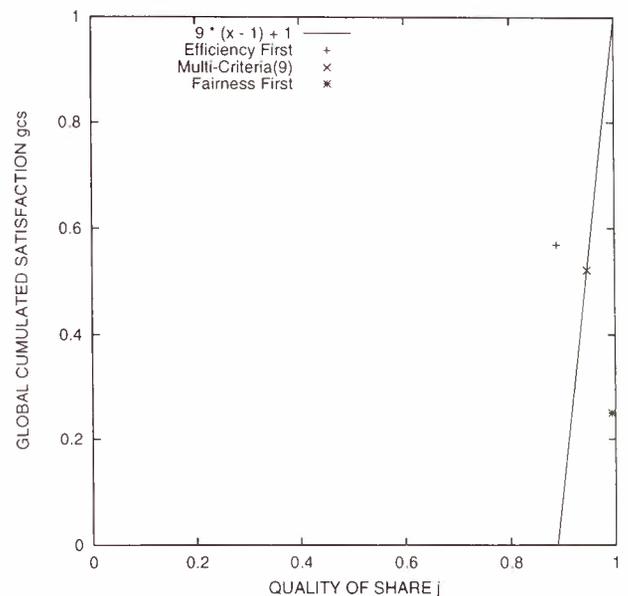


Figure 1: Comparison of methods in the $j \times gcs$ plane.

Finally, the figure 1 sets our three methods in the 2-dimensional plane (quality of share, global cumulative satisfaction). On these two criteria, no method dominates over another. Fairness First provides a quite perfect share, but a poor satisfaction. Efficiency First gives the best satisfaction, but a price in quality of share must be paid for it (this is quite acceptable, since the fairness constraint is satisfied). Lastly, Multi-Criteria($\lambda = 9$) gives a compromise solution between the two others. Other values for λ would allow to get other compromises : this is also of interest in this method.

7 SUMMARY AND CONCLUSIONS

We have described a specific share decision problem involving multiple agents, in which the satisfaction of two kinds of constraints poses a dilemma: efficiency constraints aim at satisfying the agents the most, whereas a fairness constraint watches over equity among agents.

We proposed three different methods to solve this problem. The first method searches for fairness first,

and then for efficiency. It is a simple *a priori* sharing method, allocating observation windows to each agent in turn.

The second method is based on the opposite view : first efficiency, fairness if possible. A global satisfaction criterion is defined and maximized. A “minimal fair share” for each agent is defined *a priori* but only checked *a posteriori*.

The third approach does not favor one constraint or the other, but computes a set of good compromise decisions. This is a multi-criteria approach, based on the computation of the set of Pareto-optimal decisions in the two-dimensional space (global-satisfaction, quality-of-share). This set is computed exactly by a branch-and-bound search, or approached by an adapted local search method when the search space is too large.

These three methods have been simulated on the basis of the expected data for the future Spot5 satellite. In short:

- the first method results in very good shares, but inefficient decisions,
- the second one delivers quite good decisions (minimal fair shares are always achieved and the global satisfaction is high), and uses a tolerable amount of computational resources,
- the last one is very costly in computational resources, but allows a human decision-maker to preview a set of interesting non-dominated compromise decisions.

The overall conclusions of this work are:

- no method can be indisputably put forward; the problem is not to choose a method against another one, it is to present to the agents a set of methods and their properties and to let them decide according to the properties they consider the most important³;
- whereas general methods of sharing can be stated, each share problem is specific and must be studied carefully;
- discrete share problems like this one are computationally very consuming; more specialized combinatorial optimization algorithms are needed to solve them.

ACKNOWLEDGMENTS

The authors would like to thank Lionel Lobjois and Loïc Trégan for their help in the simulation work.

References

- [1] S. Bistarelli, U. Montanari, and F. Rossi. Constraint solving over semirings. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 624–630, Montréal, Québec, Canada, August 1995. International Joint Committee on Artificial Intelligence.

- [2] S. J. Brams and A. D. Taylor. *Fair Division — From cake-cutting to dispute resolution*. Cambridge University Press, 1996.
- [3] E. Freuder and R. Wallace. Partial Constraint Satisfaction. *Artificial Intelligence*, 58(1-3):21–70, December 1992.
- [4] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, 1976.
- [5] J. Larrosa, P. Messeguer, T. Schiex, and G. Verfaillie. Reversible DAC and Other Improvements for Solving Max-CSP. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 347–352, Madison, Wisconsin, July 1998. American Association for Artificial Intelligence.
- [6] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.
- [7] H. Moulin. *Cooperative Microeconomics, A Game-Theoretic Introduction*. Prentice Hall, 1995.
- [8] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, Cambridge, Massachusetts, 1994.
- [9] T. Schiex, H. Fargier, and G. Verfaillie. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 631–637, Montréal, Québec, Canada, August 1995. International Joint Committee on Artificial Intelligence.
- [10] G. Verfaillie, M. Lemaitre, and T. Schiex. Russian Doll Search for Solving Constraint Optimization Problems. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 181–187, Portland, Oregon, August 1996. American Association for Artificial Intelligence.
- [11] R. Wallace. Directed Arc Consistency Preprocessing. In *Constraint Processing*, pages 121–137. Lecture Notes in Computer Science 923, Springer, 1994.
- [12] H. P. Young. *Equity in Theory and Practice*. Princeton University Press, 1994.

³See [8] for a discussion about this point. What we call a method is called by them a *protocol*.

FUZZY LOGIC FOR SPACECRAFT CONTROL: AN EUROPEAN APPROACH

Dr. G. Ortega

European Space Agency (ESA / ESTEC)

Keplerlaan 1, AG 2200, Noordwijk, The Netherlands

E-mail: gortega@esa.int; Tel: +31-71-565-3668; Fax: +31-71-565-5419

Prof. Dr. J.A. Mulder, Prof. Dr. H.H.B. Verbruggen

Technical University of Delft, 2600 GB Delft, The Netherlands

ABSTRACT

As technology allows the growth in size and performance of spacecraft their control systems are continuously re-designed and perfection to achieve improvements in accuracy and stabilization. A clear line in research is the improvement in the design and development of sensors and actuators to become smaller, more precise and cheap. The research line in intelligent control leads to the development of new control strategies based on new ideas and principles.

The goal of the paper is to describe the undergoing European projects to develop and achieve a fuzzy logic based technology for the control of a spacecraft. In the search for an easy, efficient, cost-effective control design and development technique, fuzzy logic seems to provide a method of reducing system complexity while increasing control performance.

First, the article analyses which the current techniques in spacecraft control systems. The emphasis is put on the analyses and design of spacecraft control systems due to its complexity.

Second, the article discusses in detail if fuzzy logic can be applied to spacecraft control systems and how can this be done easily and efficiently. Two different techniques are detailed: direct control and supervisory control. The advantage and disadvantages of each of them are carefully described.

Next, the paper details the available systems in Europe at this moment. The focus is centered around the efforts made by ESA to build three different models of spacecraft control systems based on fuzzy logic: 3-axis stabilized spacecraft model, rendezvous and docking model, and re-entry model.

After that, the paper concludes with the efforts to develop a proprietary technology to cover the existing gap in Europe. Fuzzy Logic may lead the path to new fast, robust, extensible, upgradable, and much cheaper spacecraft control systems.

1. INTRODUCTION

A spacecraft control system is the component part of a spacecraft in charge of measuring its position and attitude and producing guidance and rotation commands. It contains several blocks (figure 1): the navigation block calculates the actual state of the vehicle and predicts its immediate future state to achieve the desired trajectory (guidance); and the control part calculates the desired control torques to achieve this trajectory and attitude.

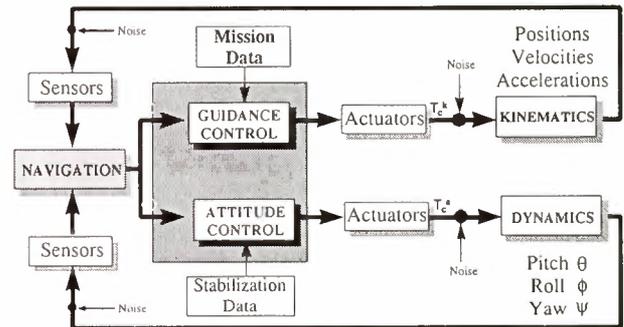


Fig. 1. Control Loop of a Spacecraft

The objectives are to maintain the vehicle within a prescribed orbit and attitude respecting the given mission constraints (fuel consumption and maneuver time minimized, heat load, etc).

The main control requirements of a spacecraft are formulated as a deviation from conditions of regular motion. In principle, this control problem could be solved in the framework of classical linear control: first defining the plant math model, second generating the laws to control it and then analyzing the robustness in conditions of abnormal operation.

The reality is that the motion equations are nonlinear [11], the performance of sensors and actuators is not totally perfect and the size of the spacecraft produces elastic modes neglected in the mathematical model of the plant. In most occasions, low and high frequencies appear with very low damping. Bode diagrams and phase plots are insufficient to forecast totally the plant behaviour in all circumstances and approximation in the discretization process must be done carefully.

2. CURRENT TECHNIQUES

This section presents a short review of the current techniques used in space control. There are several types of platforms for developing a spacecraft control system. They are classified depending on the selection of the control architecture [4]: centralized, decentralized or hierarchical. In the centralized approach all sensors provide data to the controller which on time provokes the functioning of the actuators. This model is lacking of fault tolerant features but the global control delivers performance. In the decentralized model the controller is a group of several small controllers connecting different sensors with actuators. Here fault tolerant behavior is

achieved but global coordination is difficult. The hierarchical solution is a mixture of the previous two having a coordination loop over several closed loops which control every part of the plant. In this case the design is more complex but the final system is robust to non standard situations.

Among the modern control theories developed until the present day for spacecraft systems the more widely used are the following ones:

- Multivariable robust control. Used in system with several inputs and outputs that are cross-coupled. The closed loop systems include a part for decoupling of the variables. The control engineer's goal is to stabilize the system along a series of values (a parameter). Two variants are applied in spacecraft control: H^∞ techniques and Bayesian identification techniques.
- Predictive control. It is based on the production of two models of the system: reference and predictive. The control engineer produces a mathematical reference model of the plant. At every instant the system generates some predictive models which lead to a specific end condition. Out of all these possible solutions only one will satisfy a particular restriction. The optimal model is applied as a control input to the present configuration. The complete process is repeated at regular intervals. The goal of this control is the increase in robustness and elimination of tracking errors.
- LQ (Linear Quadratic) techniques. The plant is assumed to be linear. It is described in the state space form. The control engineer creates a quadratic function using the inputs of the system. The problem is to minimize this quadratic function with respect to the control inputs subject to linear system constraints. This solution is well applied to satellites in equilibrium that must remain in equilibrium. This control is used in combination with the previous two.
- Modal control. The control engineer specifies the response time, bandwidth, damping ratio, etc. of the plant. The poles of the closed loop systems regulate the performance of the controller. The position of the poles in the Z plane *modes* are selected to fulfill a specific criterion of convergence. It is easy to apply and can be extended to more complicated models. This technique is the preamble to the applications of more deep analysis for nonlinearities.

3. FUZZY LOGIC IN SPACECRAFT CONTROL

The techniques shown in the previous section use the experience of the control engineer helped by computer design control, simulations tools and computer verification models. To apply these techniques the plant must be well understood and its reactions known in nearly all circumstances.

Can fuzzy logic be applied efficiently to spacecraft control systems? or, Is it just a good alternative to PID controllers? Can it compete with classical models?

Fuzzy logic has shown to be specially suitable in occasions when the plant is not static but changes with time (or differs slightly among very similar systems) or when the characteristics of the plant are not totally known or understood at the time when the controller was designed, or when the control actions and goals were not precisely defined. Fuzzy logic has been proven to be adequate to solve control problems not in the *best* way but just in a *suitable* way within the required limits and giving satisfactorily performance.

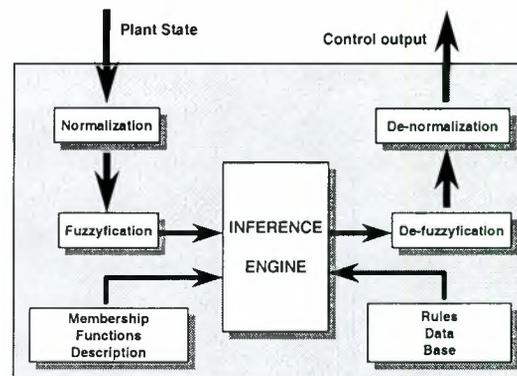


Fig. 2. Fuzzy Controller Diagram

The configuration of most spacecraft contain the following characteristics:

- The spacecraft is not a rigid body anymore but an object with multiple moving appendages.
- The final mass is not known with total precision until the complete spacecraft is finished and filled up with fuel (e.g. time close to the launch); so the control system must be designed with certain tolerances.
- A satellite thruster system can never be perfectly aligned. At the beginning of the life of the satellite every maneuver has to be carefully calibrated.
- Once in station keeping, the movement of the solar arrays provoke structural flexures to the spacecraft dynamics. As a consequence, structural resonances can occur disturbing the attitude.
- In most occasions, when thrusters are fired (re-orbit or station keeping) the satellite experiences parasitic torques along all axis different from the one containing the fired device.
- As time passes, the fuel consumption varies the total mass of the satellite and therefore the centre of gravity changes.
- The matrix of inertia is not diagonal: there are cross products of inertia.

In all the previous situations there is a significant degree of fuzziness.

Figure 2 shows a diagram of blocks of a typical fuzzy controller.

The Fuzzy Logic system represents an intelligent knowledge based controller which consists of a data base of rules and the definitions of the fuzzy sets [7], [8], [1], [3]. The plant state is normalized to be able to be fuzzified

into the appropriate fuzzy sets. The inference engine fires the rules using the membership functions over the fuzzy sets and produces a result that has to be defuzzified. Finally, the output is denormalized in order to be applicable to the control action required.

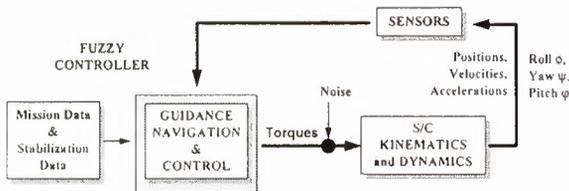


Fig. 3. Direct Fuzzy Control

Depending on the type of problem there are basically two ways to apply fuzzy logic to spacecraft control: **direct control** and **supervisory control**. In both cases the control is called *expert* because it incorporates knowledge from an expert that cannot be embedded during the design of the mathematical model.

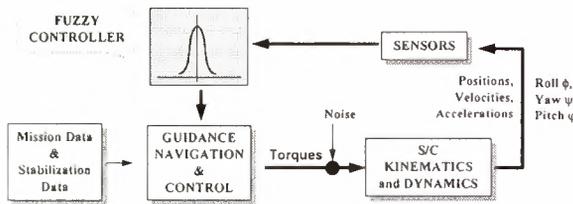


Fig. 4. Supervisory Fuzzy Control

If fuzzy logic is applied to direct control (figure 3) the fuzzy controller will replace the conventional one completely. In this case the controller replaces the role of the process operator solving the problem to produce a smooth control action in the proximities of the set point. This control reduces the errors in the process output and prevents from exceeding some predetermined value by means of adjusting the control output. In this case a typical rule of the data base looks like

*if something happens with a state variable
then produce control output*

If fuzzy logic is applied to supervisory control (figure 4) the controller acts as a supervisor of the classic control loops. The supervisor determines when and which of the classic elements will work selecting the appropriate parameters for them. Here the controller replaces the role of the control engineer tuning parameters for all the classic elements included in the complete design. The rule data base contains two kinds of rules [9]: context rules (to derive properties of close loop control from open loop) and tuning rules (to change parameters adapting them to different necessities). In this case a typical context rule of the data base looks like

if open loop process is X

then close loop is Y

and a typical tuning rule looks like

*if something happens with a control variable
then change parameter in block Z*

Basically, both types of control can be applied to spacecraft systems. Direct control is more appropriate to the centralized and decentralized types of satellite control architectures (section 2) whereas supervisory control fits perfectly in the case of a hierarchical architecture.

4. CONTROLLER CONSTRUCTION

During several years, the fuzzy logic community has developed several techniques to construct fuzzy controllers. These techniques have some commonalities. Grouped and analyzed together they form the core of a design guide for fuzzy control engineering.

The steps involved in the construction of the intelligent controller can be depicted as shown in figure 5 [12], [2].

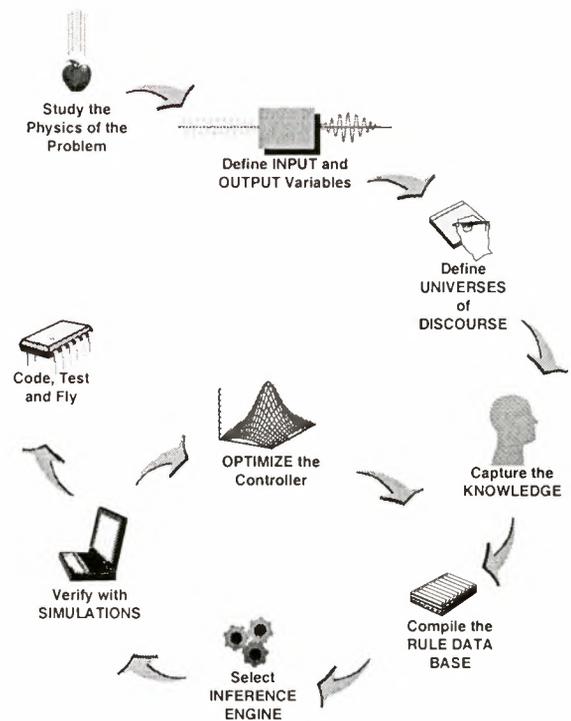


Fig. 5. Fuzzy Controller Design Spiral

Study the physics of the problem. Prior to any involvement in the design the control engineer should study the physical problem to determine which characteristics should be considered. This part is also common to the crisp approach. At this stage it is necessary to choose the type of control architecture more suitable for the problem. Several factors have to be considered: the type of satellite (science, telecommunications, Earth observation), type of orbit (circular, elliptic), etc.

The definition of input and output variables. The input variables are the sensor measurements (positions, velocities, yaw, pitch, roll, etc).

For a system with thrusters the output variables are the firing of a particular thruster (thrust position and time of fire) and the attitude angles and rates. For a system with momentum wheels the output variables can be the angular velocity of wheel rotation or the deflection angle for a gimballed momentum wheel control system. If the system includes solar arrays another output variable will be the deflection angle of the flaps to force the solar sailing navigation, etc.

Universe of discourse. The next step is the definition of the universe of discourse for all variables. For angles the universe of discourse stretches from (e.g.) $[-\pi/2, \pi/2]$. For angle rates the universe of discourse stretches between (e.g.) 0 and a maximum value governed by the actuators limits. For distances, velocities, etc. and their rates the universes of discourse belong to a particular interval.

Knowledge acquisition. An efficient method to acquire and capture the knowledge of an experienced spacecraft controller is very important. This knowledge will form the rules data base which will contain the type of control to realize.

Compilation of the rules data base. The rules data base form the kernel of the knowledge based controller [20]. Depending of the type of fuzzy control (direct or supervisory) the construction of the rules data base is significantly different. In the case of direct control the knowledge based controller implements the close loop control actions substituting completely the operator. The data base rules are grouped depending on the control action they generate. In the case of supervisory control the fuzzy device must schedule the functioning of the classic control blocks. The rules data base contains context rules and tuning rules. With the context rules the fuzzy controller classifies the satellite flying type environment. With the tuning rules the fuzzy device changes loop gains, delays, constants, etc. Thanks to the tuning rules the data base will incorporate an experience which can only be realized in the corresponding analytic model by means of manual operations.

The election of the Inference Engine. The inference engine is needed to fire the rules. There are several methods to program the engine. One of the most popular is the Mamdani's Min-Max mechanism; normally the AND operator is chosen as the minimum of two weight antecedents instead of its multiplication. For fast processing the defuzzification strategy used is often the centre of gravity computation. In general, the inference engine can be an approximate reasoning kernel based on already proposed systems.

Verification with simulations. The power of the simulations can be used to verify the convergence and stability of the controller. A fast prototype must simulate the plant and the controller as well. Most of the available packages provide with graphical tools to visualize the results of the simulations.

Optimization. The knowledge of a spacecraft controller can be captured to generate the rules data base or to determine the overlapping of the fuzzy sets. A priori, it is

difficult to evaluate if the control output produced is optimal or not. To optimize the rules data base or the fuzzy sets used by the membership functions two approaches can be followed: manual optimization using the common sense and human experience or automatic tuning (using adaptive fuzzy control or genetic algorithms tools for example).

Coding, testing and flying. The physical implementation of the controller requires to write source code that will be inserted in the computer memory of the flying processor. The final system will be mounted in the attitude and orbit control subsystem of the vehicle [21], [5]. It will determine the actual state of the spacecraft and it will generate torques to execute maneuvers to guide and position the spacecraft. Once in the final orbit the close loop operations of the intelligent controller are performed in an autonomously way replacing the usual control algorithms.

5. APPLICATIONS IN EUROPE

The European Space Agency is currently undertaking studies in the applicability of the fuzzy logic control techniques to spacecraft control.

Utilising the research made by the Technical University in Delft (The Netherlands), ESA is on the way to construct spacecraft simulators which incorporate fuzzy logic techniques in their guidance, navigation, and control systems.

Three different projects demonstrate the feasibility of the fuzzy logic control for spacecraft applications:

- 3-axis stabilised satellite control.
- Rendezvous and docking control between a resupply vehicle and a space station.
- The Earth atmospheric re-entry of a rescue vehicle which carries astronauts from an orbiting space station back to Earth.

In all cases, the control system is based on fuzzy logic, capturing the knowledge of experienced spacecraft pilots or ground operators. This knowledge is represented as a set of rules and the definitions of the fuzzy sets. The control system shall determine the present state of both vehicles, and shall generate torques to execute the maneuvers that will lead to the desired orbit and attitude.

3-axis Stabilized Satellite Control

The three-axis stabilized spacecraft case is representative of an ESA typical scientific, Earth observation, or telecommunication satellite mission. The target of this development is the ESA Infrared Space Observatory ISO. The fuzzy control for ISO shall verify the advantages of this type of control in high pointing accuracy manoeuvres (figure 6).

The demand for accuracy in pointing manoeuvres has increased during this decade and it is expected to further increase in the future.

Typically the satellite is pointed to several targets in several slots of time [19], [18], [16], [17], [14]. These operations are commanded from ground using operational procedures executed by spacecraft controllers.

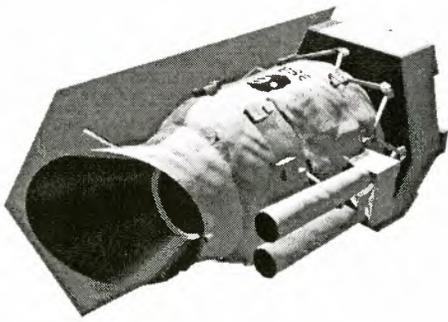


Fig. 6. The Infrared Space Observatory

A fuzzy logic based intelligent control system could measure its position and orientation in space with respect to the target and compute the torques to re-point the satellite. ISO had to be able to maneuver smoothly from one celestial source to the next, and then maintain accurate pointing on that target. The spacecraft was capable of pointing at any region of the sky that satisfies certain stray-light constraints. The slew speed between sights was set at 7 degrees/min in order to optimize observation time, and the duration of each observation could range from a few seconds to up to 10 h, depending on the type of source. In this case, a direct control type could activate the reaction control wheels and the thruster system of the satellite achieving a smooth, very fine pointing accuracy. The control effort should be minimum, having the constraints to keep fuel consumption and slewing time as a minimum.

Rendezvous and Docking Control

The second case of applicability of fuzzy intelligent control is the problem of the rendezvous and docking operations of two spacecrafts [10], [15]. The target of the investigation is here the ESA's re-supply vehicle for the International Space Station (the Automatic Transfer Vehicle ATV) [6].

As one of the European contributions to the future International Space Station (ISS), the European Space Agency is developing the Automatic Transfer Vehicle (see figure 7). ATV is an unmanned, Ariane-5 launched vehicle that will perform regular reboost and refuelling and payload supply and removal to the ISS. Other missions of ATV will comprise payload supply and payload removal from the ISS.

The ATV project was approved in October 1995 by the Council of the European Space Agency. ATV will be launched for the first time from Kourou (French Guiana) in February 2003.

ATV is basically a cylindrical shaped spacecraft containing a cargo module pressurized or un-pressurized, a docking port, and a propulsion module. ATV will dock to the service module of the Russian segment of the ISS.

The ATV rendezvous and docking mission is equivalent to the problem of the rendezvous and docking of an active servicing spacecraft into a big passive space station rotating around the Earth. In this problem, the active chaser

produces smooth control actions in the proximity of the passive target and during the structural latching to avoid disturbance torques in the final assembly orbit [13].

In this case, a supervisory control could be applicable. The reason is that fuzzy logic may be very well suited to guide the servicing vehicle during the rendezvous phases. For the fine docking and structural latching operations, the fuzzy device could command a typical PID type control block.

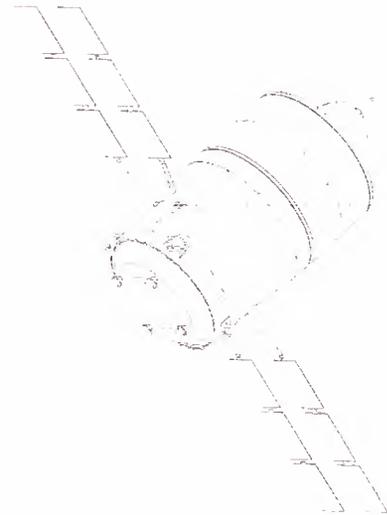


Fig. 7. The Automatic Transfer Vehicle

Atmospheric re-entry of a lifting body

The third study case is a lifting body winged type vehicle for atmospheric Earth re-entry and landing. The target of this development is the ESA-NASA Crew Rescue Vehicle (CRV).

The CRV (depicted in figure 8) is a spacecraft attached to the International Space Station which will serve as a re-entry vehicle for astronauts on-board. The CRV will depart from its docking port of ISS and will reach a particular landing site.

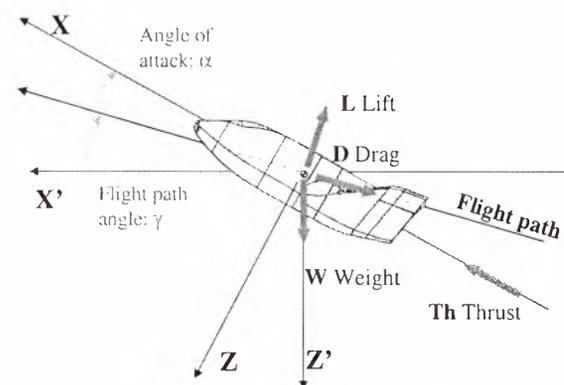


Fig. 8. The ESA-NASA Crew Rescue Vehicle

The short flight (around 40 minutes) of the CRV is catalogued in three basic phases: the de-orbiting, the re-entry, and the landing. For each of the phases, the craft behaves differently. This fact makes the flight dynamics rather complex and engineering demanding. The critical control problem is the stabilization of the spacecraft axis, and its velocity vector.

Fuzzy logic in the guidance, navigation, and control unit of the CRV shall be able to cope with a huge variety of control regimes, performances, and constraints during the complex flight of the vehicle.

6. CONCLUSIONS

From the experience of several decades and a tremendous effort employed in the optimization of a variety of control systems the engineers know that a poor identification of the plant produces good results in the robustness of the system.

Fuzzy logic deals with uncertainty in the identification of the system model. Fuzzy logic emulates the behavior of human operators for complex control tasks. A fuzzy logic controller embedded in a guidance, navigation and control system of a spacecraft can realize autonomously the close loop operations helping or replacing the conventional crisp control algorithms.

ESA is underway to build up three fuzzy logic based spacecraft control simulators for three different types of missions: a classic 3-axis stabilized satellite mission (ISO), a rendezvous and docking (ATV) and a lifting body re-entry vehicle (CRV). These shall prove the capability and adequacy of fuzzy logic in the area of spacecraft control, leading the way to new cheaper, faster, better control system for space vehicles.

Fuzzy and crisp logic will coexist in the near future to develop a new generation of spacecraft control systems of high quality, more flexible, cheaper and intelligent.

REFERENCES

- [1] O.A. Abihana. Design and implementation of fuzzy logic controllers. Master's thesis, College of Engineering & Technology, Old Dominion University, Norfolk, Virginia 23529, July 1993.
- [2] Hamid R. Berenji, Robert N. Lea, Yashvant Jani, Pratap Khedkar, Anil Malkani, and Jeffrey Hoblit. Space shuttle attitude control by reinforcement learning and fuzzy logic. In Robert J. Marks II, editor, *Fuzzy Logic Technology and Applications*, pages 363–368. IEEE Technical Activities Board, 1994, 1993.
- [3] R. Brown. *Fuzzy Logic Application for Modeling Man-In-The-Loop Space Shuttle Proximity Operations*. PhD thesis, Massachusetts Institute of Technology, 1994. CASI number: NASA-CR-188283.
- [4] C. Champetier, E. Desplats, and P. Pelipenko. Modern control techniques for spacecraft aoc/gnc systems. In *Proceedings of the First ESA International Conference of Spacecraft Guidance, Navigation, and Control Systems*, number 323 in ESA-SP, pages 483–490, 1991.
- [5] Stephen Chiu and Masaki Togai. A fuzzy logic programming environment for real-time control. In Henri Prade Didier Dubois and Ronald R. Yager, editors, *Fuzzy Sets for Intelligent Systems*, pages 361–367. Morgan Kaufmann Publishers, Inc., 1993.
- [6] M. Cislighi, M. Lellouch, and J.M. Pailot. The atv rendezvous pre-development programme. *ESA Bulletin*, 1997.
- [7] S. Daley and K.F. Gill. A justification for the wider use of fuzzy logic control algorithms. In *Proceedings of the Institute of Mechanical Engineering*, volume 199, No. C1, 1985.
- [8] S. Daley and K.F. Gill. Attitude control of a spacecraft using an extended self organizing fuzzy logic controller. In *Proceedings of the Institute of Mechanical Engineering*, volume 201, No. C2, 1987.
- [9] D. Drianov, H. Hellendoorn, and H. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Berlag, 1993.
- [10] W. Fehse. Rendezvous and docking technology development for future european missions. Technical report, European Space Agency, 1985.
- [11] G. Georgiou, D. Normand-Cyrot, S. Di Gennaro, and S. Monaco. On the non-linear adaptive control of a flexible spacecraft. In *Proceedings of the First ESA International Conference of Spacecraft Guidance, Navigation, and Control Systems*, number 323 in ESA-SP, pages 509–516, 1991.
- [12] Chuen Chien Lee. Fuzzy logic in control systems: Parts i and ii. *IEEE Transaction on Systems, Man and Cybernetics*, 20, No. 2, March 1990.
- [13] V.V. Malyshev and M.N. Krasilshikov. Modern approaches on guidance and control of russian spacecraft. In *2nd ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, pages 331–338, April 1994.
- [14] S. Nicosia and P. Tomai. Nonlinear observer and output feedback attitude control of spacecraft. *IEEE Transactions on Aerospace and Electronic Systems*, 28(4):970–977, oct 1992.
- [15] J.M. Pailot, W. Fehse, and A. Getzschmann. Development of an autonomous onboard control system for rendezvous and docking. In *Automatic Control in Aerospace 1992*, volume 12, pages 331–338, 1992.
- [16] R. Prigge, H. Klotz, and H. Strauch. Spacecraft attitude control using magnetic torques and gas jets. In *Proceedings of the First ESA International Conference of Spacecraft Guidance, Navigation, and Control Systems*, number 323 in ESA-SP, pages 491–496, 1991.
- [17] M. Shamma and T. Michaelis. Double gimbaled momentum wheel design for small satellites. In *Proceedings of the First ESA International Conference of Spacecraft Guidance, Navigation, and Control Systems*, number 323 in ESA-SP, pages 369–392, 1991.
- [18] Sahjendra N. Singh. Nonlinear adaptive attitude control of spacecraft. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(3):371–379, may 1987.
- [19] Sahjendra N. Singh. Robust linear attitude control of flexible spacecraft. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(3):380–387, may 1987.
- [20] S.K. Tso and H. Fung. Effective development of fuzzy-logic rules for real-time control of autonomous vehicles. In *IFAC workshop on Safety, Reliability and Applications of Engineering Intelligent Control Technologies*, pages 143–148, December 1994.
- [21] T. Yamaguchi, K. Goto, and T. Takagi. Intelligent control of a flying vehicle using fuzzy associative memory system. In Robert J. Marks II, editor, *Fuzzy Logic Technology and Applications*, pages 369–379. IEEE Technical Activities Board, 1994, 1992.

INTELMOD: ARTIFICIAL INTELLIGENCE IN SUPPORT OF MISSION OPERATIONS TASKS

Alessandro Donati

ESA/ESOC, Robert-Bosch-Str. 5, D-64293 Darmstadt, Germany,
tel. +49 6151 902574, fax +49 6151 902135, e-mail adonati@esoc.esa.de

Enrico Romani

Dataspazio S.p.A., Via Laurentina 756, I-00143 Rome, Italy,
tel.: +39 06 50097 24, fax: +39 06 5022331, e-mail enrico.romani@dataspazio.it

Mark Aynsley

Science Systems (Space) Ltd. 1st Floor, 12 The Parks, Newton-le-Willows, Merseyside, UK,
tel: +44 (0)1942 270 060, fax: +44 (0)1942 270 049, e-mail: mark.aynsley@scisys.co.uk

ABSTRACT

This paper presents the results of a project aimed to deliver an advanced pre-operational Artificial Intelligence (AI) system to support visual monitoring, diagnostic processing and resource management of a spacecraft during its operational lifetime. The tool can also be used during mission preparation for training of flight controllers and as an assistant during operational procedures development and validation.

The INTELLigent MODeller (INTELMOD) toolkit has been developed using a COTS knowledge based system and object-oriented techniques. Initially, it will assist the flight controllers in the generation and maintenance of spacecraft models, providing a user friendly man-machine interface to collect engineering and operational knowledge directly from the experts. The models will then be used by the INTELMOD inference engine to support the flight controllers in the following tasks: monitoring, anomaly detection and anticipation, failure detection, isolation, diagnosis and recovery, failure propagation analysis and on-board resource management.

The generic tool has been then demonstrated for the Cluster mission, with the modelling of the Cluster AOCMS and power subsystems and with customized interfaces for the Cluster mission database and SPEVAL [1] system, providing processed telemetry (and telecommand) retrieval.

1. INTRODUCTION AND BACKGROUND

The European Space Operations Centre (ESOC) of ESA, as a centre of excellence in delivering flight operations services to spacecraft user community, has often made the effort to pursue constant improvement in its own processes, methods and tools to guarantee an outstanding record of successful mission operations. In this framework INTELMOD represents a pioneering activity in understanding and exploiting, at prototype level, the potential benefits offered by mature

knowledge based system technology and applied to flight operations processes.

Previous studies [2] have already offered indications of the possible use of artificial intelligence in the spacecraft operations domain; the INTELMOD toolkit now provides a user-friendly modeling environment directly usable by operations and spacecraft experts for two purposes:

- to transfer and organize their own knowledge of the spacecraft system, mission phases, and diagnostic rules into a knowledge database;
- to use the stored "know-how" during mission operations to advice operations staff when and, possibly, just before anomaly occurrences.

2. OPERATIONAL CONCEPT

INTELMOD is not intended to replace any existing Mission Control System (MCS); it will be interfaced to existing MCS with the objectives of expanding and extending the current MCS supported functions.

To support the different stages of model development and implementation, three classes of INTELMOD user have been identified:

Spacecraft Component Developer (SCD)- responsible for the creation of spacecraft components (modules, subsystems and units) which are then inserted into a Component Library to be used later during a model definition phase. Component Developers are expected to have a high level of knowledge concerning typical spacecraft "building blocks".

Spacecraft Model Developer (SMD) - creates a mission specific model representation by selecting and configuring items created by the Component Developer. Models are progressively assembled and configured to provide a physical, functional and mission-related representation of the spacecraft in question.

INTELMOD User - interacts with the models created by the Model Developer during Mission operations / training scenarios.

The toolkit has been conceived to be used in two distinct phases of the spacecraft operations lifecycle:

- during *mission preparation*, the users will access the toolkit as SCD, to create, modify or augment the models of the "terminal" elements of the hierarchical representation of generic spacecraft, to be then stored in a library. In the very same phase, the user will access also as SMD to model selected subsystems, down to the end item, the related mission modes, the diagnostic and failure propagation rules, the contingency procedures and the trend analysis rules belonging to a specific spacecraft and mission;
- during *flight execution* phase INTELMOD will be connected to the existing Mission Control System (MCS), to receive telemetry (and telecommands). The user will access as INTELMOD User. This time the toolkit will provide operational advisory services to the flight control team.

The flight operators will be supported by INTELMOD for visual monitoring and alarming, diagnostic support, including failure detection and anticipation, failure isolation, diagnosis and recovery and failure propagation analysis, resource evaluation and assessment.

The toolkit has been developed taking into account the following requirements:

- support multi-mission environment;
- user-friendliness of the interface for the operations and spacecraft experts during modeling and flight operations phases;
- minimal software customization effort when applying the toolkit to a specific mission, limited to interface adaptation.

3. THE PROJECT

INTELMOD has been developed using a RAD-style approach, based upon the Dynamic System Development Method (DSDM), which is a non-proprietary method that is becoming a de-facto standard within the UK [3]. This approach was partially adopted in this study to help ensure that the system could be developed in a much shorter timescale, and that the final system was more closely matched to ESA's real needs.

DSDM employs an iterative approach to development with heavy emphasis on end-user involvement and a project management philosophy which focuses on *products* rather than the activities needed to achieve them. Timeboxes were used to control the development

process, whereby a fixed amount of time was allocated to complete a given area of functionality. Figure 1 below shows the DSDM development process, although this required some modification to comply with ESA's particular project control requirements and to overcome the challenges associated with the development team and end users being located in geographically distant locations.

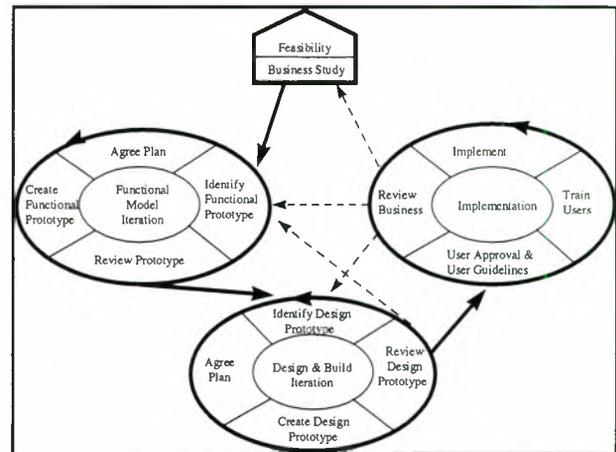


Figure 1. DSDM Development Process

INTELMOD has made extensive use of commercial off-the-shelf (COTS) software products, including:

- G2
- GDA
- G2-Weblink
- ODBC Bridge
- Space UNiT (Universal Intelligent Toolkit)

INTELMOD has primarily been developed using G2 (Gensym Corporation). This provides an object oriented environment for building and deploying mission-critical, intelligent applications. It is typically used to represent knowledge captured from operations experts performing complex tasks in real-time situations and has a broad range of potential application areas. GDA (G2 Diagnostic Assistant) is a layered application product for G2. It provides an integrated visual development and execution environment for modelling application logic / diagnostics. The major strength of GDA lies in its intuitive Graphical User Interface (GUI), allowing faster development of complex systems models required for INTELMOD.

G2-Weblink enables users to access G2-based applications via Web browsers. This allows the distribution of intelligent decision support information to intra/internet users throughout an organisation. Gensym also provides bridges for all ODBC-compliant databases. In the current INTELMOD study, a Microsoft Access copy of the CLUSTER database was used. Finally, INTELMOD also incorporates Space UNiT (Science Systems Space Ltd). This has been

developed in a partnership programme for ESA to provide a component based suite of graphical products for procedure execution, schedule execution, monitoring and event handling [4]. Space UNiT enables INTELMOD to automatically execute contingency procedures following the detection of anomalies by one or more of the functional models described in section 4.

These products were used in order to provide the rapid delivery of functionality required. In addition, the industrial partnership approach made possible for the project to remain within budget.

4. ARCHITECTURE AND MODELS

4.1 Spacecraft Systems Model

This model provides a hierarchical representation of the spacecraft, its subsystems and individual components. For example, a spacecraft may be partially represented in terms of power, thermal and AOCMS subsystems. The power subsystem in turn may be composed of a power distribution unit, batteries etc. This knowledge is entered using a *breakdown editor* to interactively gather and structure knowledge related to the physical organisation of the spacecraft. The breakdown editor configures itself according to the user currently interacting with the system (SCD, SMD or generic user). Fig. 2 contains an high-level object-oriented diagram describing breakdown editor behaviour and user relationships

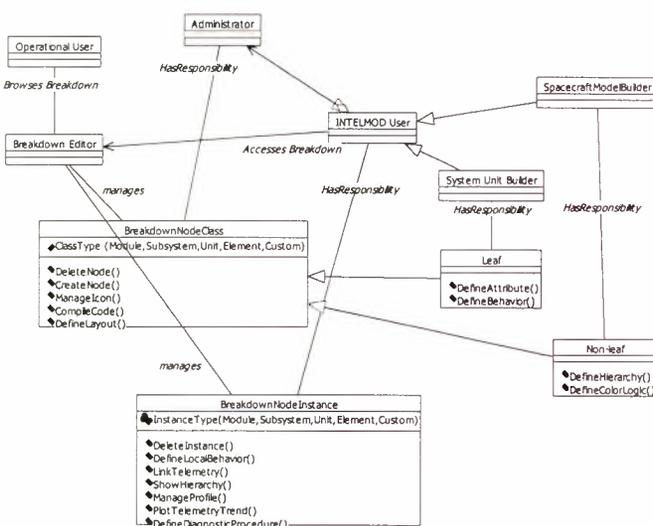


Figure 2. Breakdown Editor OO diagram

4.2 Mission Model

The Mission Model provides a hierarchical representation of the mission in terms of the various activities performed within various phases and modes, together with the expected configuration (i.e. status and resource consumption) for all the physical components defined in the physical spacecraft model. As with the physical model, a *breakdown editor* allows the Model Developer to gather and structure this knowledge.

4.3 Functional Model

The Functional Model uses a *graphical rule based language* to define knowledge related to the functions to be performed or assured during the course of a mission. This knowledge falls into the following areas:

- **Spacecraft Behavioural Knowledge:** Contains knowledge describing the behaviour of the spacecraft systems with respect to the interaction between the various components and subsystems. This knowledge enables the model to perform basic diagnostic functions including failure isolation and recovery.
- **Mission Behavioural Knowledge:** Describes the spacecraft behaviour exhibited during the execution of different mission phases and the activities performed during those phases. This model also uses the Flight Operations Plan (FOP) to enable INTELMOD to perform resource evaluation.
- **Spacecraft/Mission Relationship Knowledge:** This knowledge model captures the heuristics used by the Operations, Spacecraft and Payload engineers to identify and rectify problems that occur over the lifetime of the spacecraft. Once defined, the spacecraft/mission relationship knowledge enables INTELMOD to perform trend analysis, failure detection, diagnosis and prevention and to recommend recovery actions.
- **Spacecraft/Mission Propagation Effect Knowledge:** This model, available from the Flight Operations Plan and Mission specialists, describes the cause and effect knowledge, which relates sections of the spacecraft and mission models. A causal network allows the flight controllers to perform an analysis of process and hardware failures and predict the consequences of failure if no corrective action is taken. It also allows the controller to assess the mission impact in terms of unavailable hardware and lost functionality.

A summary of INTELMOD users and model types is shown in figure 3:

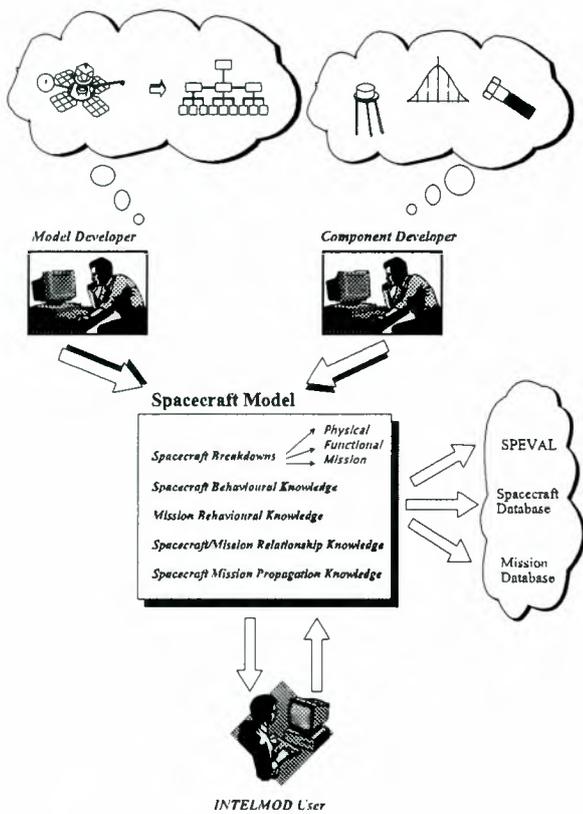


Figure 3. INTELMOD Users and Models

5. EXTERNAL INTERFACES

In order to demonstrate that INTELMOD could operate in a realistic manner and provide the support required to Mission Operations, the system must be provided with a high degree of connectivity. This was achieved using three separate interfaces:

- TM/TC bridge
- Spacecraft Database Bridge
- WWW Bridge (for message broadcast)

These interfaces are shown in figure 4 and described below.

5.1 TM and TC Bridge

The telemetry and telecommand bridge provides INTELMOD with the information required to perform its analysis. In a "live" implementation, this information would be supplied directly from the Mission Control System (MCS) software. However, in the time frame of the current study, data files were provided using SPEVAL (Spacecraft EVALuation tool), a client server application developed during a previous ESOC study [1]. SPEVAL maintains its own archive of data from the Cluster MCS, and this data can be retrieved batchwise using SPEVAL "Save Cases". The resulting data files are then read by INTELMOD, using a dedicated bridge.

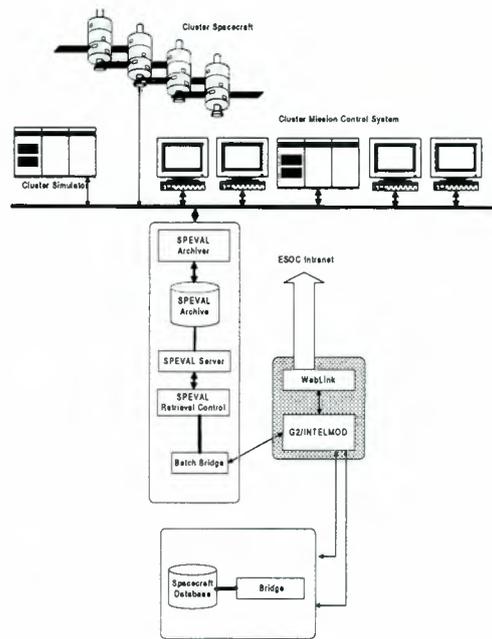


Figure 4. INTELMOD Interfaces

5.2 Database Connectivity Bridge

The purpose of the database bridge is to connect the INTELMOD modelling environment to a satellite database containing information regarding the various parameters, command definitions and scaling / limit data that are specific to the particular spacecraft under study. This bridge enables INTELMOD to support the development of the models described in Section 4.

5.3 Message/ Alarm Broadcast Bridge

To provide a method for the distribution of alarms, warning messages and general information, G2 Weblink has been incorporated into INTELMOD. Whilst it would have been possible to provide a dedicated bridge to send messages as e.g. e-mail messages within ESOC, the use of Weblink provided a more generic COTS solution, enabling information to be accessed from the widest possible range of platforms.

6. INTELMOD FOR CLUSTER II

Within the scope of the current study, INTELMOD customization has been focussed on the AOCMS and Power subsystems of Cluster [5]. Taking the Power subsystem as an example, this has been decomposed using INTELMOD's Breakdown Editors into the following units:

- power control

- power distribution
- internal power dumpers
- external power dumpers
- battery regulation
- batteries
- pyro-electronics

Once these breakdown components have been identified attributes can be added, again using a dedicated editor e.g. a battery would typically be described using properties such as voltage, temperature, charge current and discharge current. This is shown in figure 5 below.

dedicated parsers which allow defining this logic in a 'natural language' way.

Figure 6 shows a simple INTELMOD GDA-based diagnostic model (Spacecraft Behavioural). The blocks on the left of the diagram are "entry points", usually corresponding to a telemetry value which can be automatically created from the breakdown components. Signals are fed through various GDA logic blocks in an attempt to diagnose the cause of operational problems - in this case an internal power subsystem failure arising

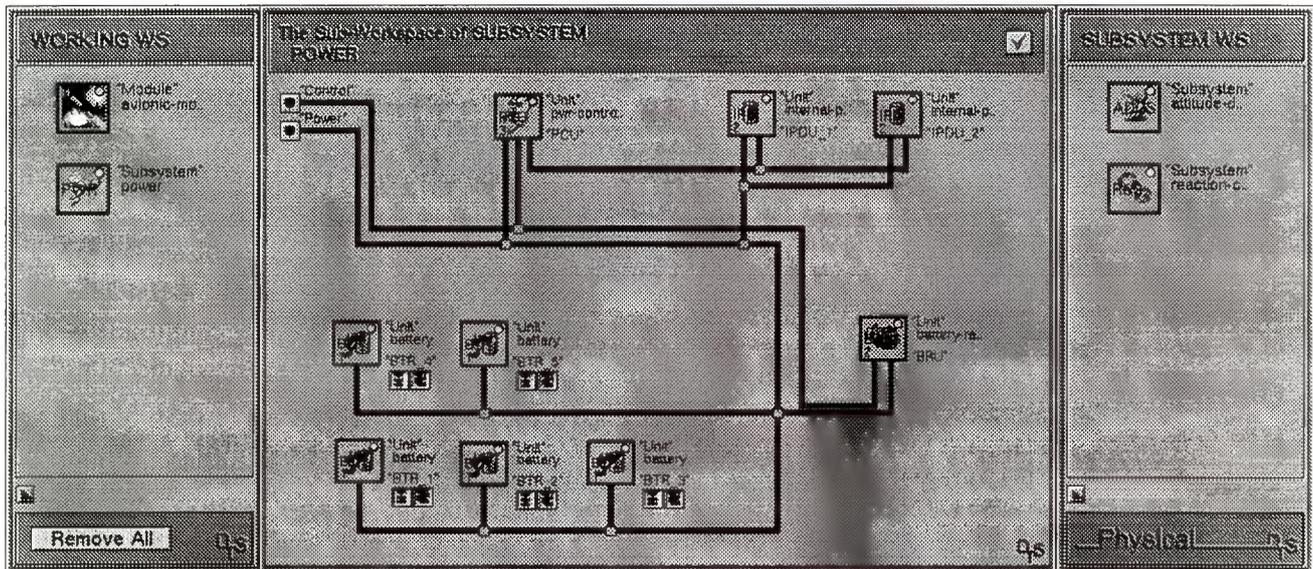


Figure 5. Screenshot of Breakdown Components

Having linked the INTELMOD breakdown components with their external counterparts (i.e. telemetries or groups of telemetries) by interactively querying the spacecraft database, it is then possible to construct the monitoring and diagnostic "rules".

Monitoring rules allows mapping status of breakdown leaves (as derived from the incoming telemetries) with a synthesis information: whenever the status of changes the color of the related component displayed in the breakdown mimics changed according to the following table:

- | | |
|------------|------------------|
| 1 - green: | OK |
| 2 - red: | fault |
| 3- yellow: | off-line |
| 4- blue: | stand-by |
| 5- orange: | redundancy lost. |

This information travels upward in the hierarchy: once colors has been computed for all the components belonging to a specific level (e.g. units), the same colors are logically combined to derive the color representing the "assembling" component at upper level (e.g. subsystem). Color propagation and telemetry association logic is entered within INTELMOD via

from a battery overdischarge. If all the logic paths entering the "AND" block on the left of the diagram are true, then a diagnosis can be made. A message will be sent to one of INTELMOD's message areas alerting operators to the cause of the problem. It should also be noted that the outputs/conclusions of one GDA diagram can pass information to other diagrams (via the connection post BTR1-CP1) and other INTELMOD model types. Customised GDA blocks are available to link diagnostic models with fault propagation models. In this way, operators are not only alerted to system failures and their potential causes, they can also be supported in assessing the likely knock-on effects, when these are expected to occur, and the impact on mission operations. In the example shown, the GDA model also incorporates a link to a UNiT procedure (shown by the block labelled "SL"). Consequently when the diagnosis is made, a contingency procedure will be automatically invoked to provide failure recovery .

All of INTELMOD's model types share a common mode of use. Model Developers use *pull-down menus and palettes* to select the building blocks that are required. These are then placed on a workspace, configured with any necessary information, then connected together. Such models are then immediately ready for use, allowing the developer to concentrate on the *expression of expert domain knowledge* rather than writing conventional programs.

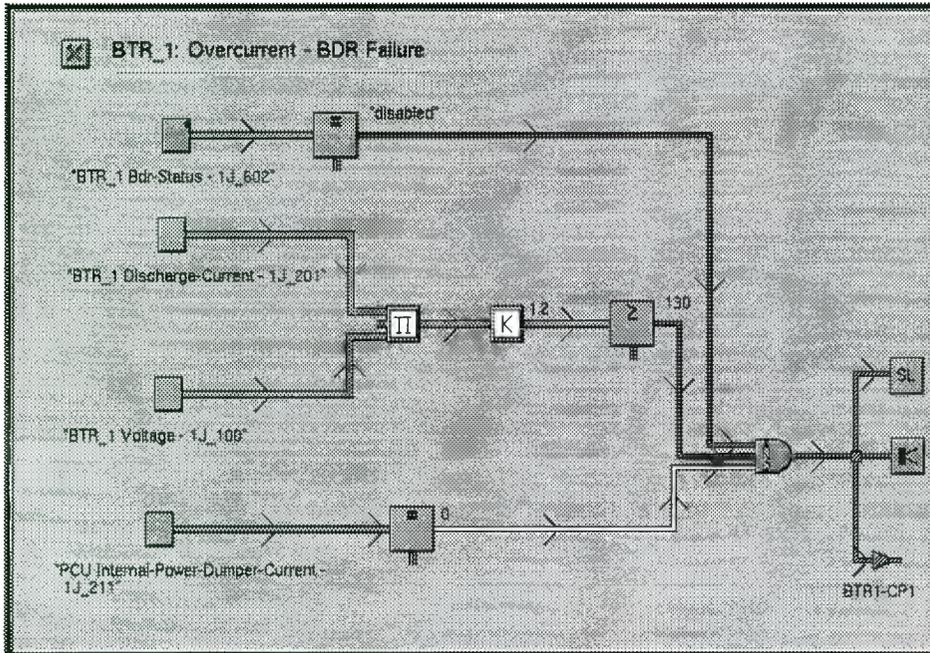


Figure 6. GDA-based diagnostic model

7. BENEFITS, LIMITS, FOLLOW-ON

INTELMOD is a generic satellite modelling toolkit that has been developed to offer faster, incremental development of spacecraft models. Thanks to its user-friendly man-machine interface, the system requires no formal programming expertise.

The use of AI techniques within INTELMOD has provided a significant enhancement in supporting operational tasks like monitoring, anomaly detection and anticipation, failure detection, isolation, diagnosis and recovery, failure propagation analysis and resource management.

The toolkit could also be exploited for assisting the training of new operations staff during simulated test case sessions, making use of the previously captured expertise.

Moreover it could potentially assist engineers during trade-off analysis, specifically to investigate alternative design solutions, alternative operational strategies and contingency recovery procedures.

The investment required in the modelling process has to have an economical return to be justified. From initial estimations, the toolkit should provide potential cost savings in the flight operations budget especially if applied to long duration missions (e.g. interplanetary missions) or recurrent/ repetitive missions (e.g. satellite constellations, meteorological satellites, etc.). Its applicability could also be of interest if focused on critical subsystem(s) of a specific spacecraft.

INTELMOD can also support the automation of routine operational activities.

The results provided by INTELMOD will pave the road for an innovative operations concept where AI-based tools, integrated in an existing mission control system, will provide more effective and efficient support to the flight controllers during safety-critical and routine operations.

8. REFERENCES

[1] ESA (1994): "SPEVAL Software Users Manual". ESA Internal Document. Ref. SUM/A18/0300. Vol. A.18, Issue 3.00

[2] Donati, A. Romani, E (1996): "A Knowledge Based System to Support the Operation in ATV Rendez-vous Missions with the International Space Station". SpaceOps '96.

[3] Stapleton, J. (1997): "DSDM: Dynamic Systems Development Method. The Method in Practice". Addison-Wesley Longman Limited.

[4] Bone, J. (1999). "Space UNiT". Journal of the British Interplanetary Society Vol. 52, No. 2.

[4] DASA-Dornier (1995). "CLUSTER Users Manual - Vols. 1,4,5".

LIST OF ACRONYMS	
AOCMS	Attitude, Orbit Control and Measurement Subsystem
COTS	Commercial Off-the-shelf Tool
DSDM	Dynamic System Development Method
ESOC	European Space Operations Agency
GDA	G2 Diagnostic Assistant
GUI	Graphical User Interface
INTELMOD	INTELLigent MODeller
MCS	Mission Control System
FOP	Flight Operations Plan
ODBC	Open DataBase Connectivity
RAD	Rapid Application Development
SCD	Spacecraft Component Developer
SMD	Spacecraft Model Developer
SPEVAL	SPAcecraft EVALuation system
UNiT	UNiversal intelligent Toolkit

ANOMALY DETECTIVE GROUND SUPPORT SYSTEM FOR MARS PROBE "NOZOMI"

Masashi Hashimoto

Institute of Space and Astronautical Science (ISAS)

3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, Japan

phone: +81 42 759 8352, fax: +81 42 759 8440, e-mail: masashi@nml.isas.ac.jp

Naomi Nishigori

Systems Section, Space Systems Dept., Systems Engineering Group, Fujitsu Limited

9-3, Nakase 1-chome, Mihama-ku, Chiba-shi, Chiba 261-0023, Japan

phone: +81 43 299 3248, fax: +81 43 299-3012, e-mail: RHE00337@nifty.ne.jp

Mitsue Mizutani

Fujitsu Social Systems Engineering Limited

1-22-1, Higashigotanda, Shinagawa-ku, Tokyo 141-0022, Japan

e-mail: KYR03543@nifty.ne.jp

ABSTRACT

Institute of Space and Astronautical Science (ISAS) was successfully launched Japan's first Mars probe "NOZOMI" on July 4, 1998. In order to enable the safe operation of "NOZOMI" under the severe demand of the cost reduction, ISAS provided an anomaly detective ground support system named ISACS-DOC for "NOZOMI" operation. This system is the second application of expert technique to daily operation of scientific spacecraft at ISAS. Many improvements are achieved based on the experiences of the first expert application to the anomaly detective ground support system for the geomagnetic observation spacecraft "GEOTAIL". Especially, the communication link levels can be evaluated within the error of ± 3.0 dBm. This performance is expected to be enough to operate "NOZOMI" safely in the Mars transfer and orbiting phase. Now ISACS-DOC diagnoses "NOZOMI" every operation, and it can warn about 450 kinds of "NOZOMI" abnormality to the spacecraft operator.

1. INTRODUCTION

ISAS operates four satellites (AKEBONO, YOHKOH, ASCA and HALCA) and two spacecraft (GEOTAIL and NOZOMI) on daily bases at the present. Scientific satellites/spacecraft are becoming more sophisticated according to more advanced scientific instruments onboard and the knowledge required to operate them is also becoming deeper and broader.

On the other hand, strong demand to suppress or decrease the operating cost exists almost everywhere these days and also in ISAS. One of the solutions for these contradictory requirements is to use expert technique in the satellites/spacecraft operation. ISAS decided to take the expert technique at first for the geomagnetic observation spacecraft "GEOTAIL" that was launched in 1992 and is still in operation at Sagamihara Spacecraft Operation Center (SSOC). This system is named "ISACS", which stands for Intelligent SAtellite Control Software. ISACS consists of two functions: one is ISACS-PLanner (ISACS-PLN) and the other is ISACS-DOctor (ISACS-DOC). ISACS-PLN is an automatic command planner with expert technique, and ISACS-DOC is an anomaly detective ground support system with diagnostic functions. ISACS-DOC for GEOTAIL has been used for 6 years in SSOC. One of the important things learned from ISACS-DOC for GEOTAIL is that collection of many kinds of information from the various kinds of ground support systems like tracking systems, trajectory determination systems and attitude/maneuver control systems is essential to improve the reliability of diagnosis. ISAS completed establishing a new ground operation system based on client/server workstation systems and started to use it from Japan's first Martian probe "NOZOMI" at SSOC. Figure 1 shows the outline of the ISAS new ground operation system. In this configuration, useful information from essential ground systems can easily be gathered through the network in real time. ISACS-DOC for "NOZOMI" was developed by taking the above advantage as well

as all other experiences obtained by ISACS-DOC operation for GEOTAIL. "NOZOMI" was successfully launched July 4, 1998 from Kagoshima Space Center (KSC) in Japan and is now flying to Mars to study the structure and dynamics of the Martian upper atmosphere with emphasis on its interaction with solar wind. This paper describes ISACS-DOC for "NOZOMI" and reports some of the actual operation results.

2. OUTLINE OF ISACS-DOC FOR "NOZOMI"

2.1 Basic Concept of System Development

The followings are basic ideas in designing ISACS-DOC for "NOZOMI". Some of them are introduced based on lessons learned from ISACS-DOC for GEOTAIL.

- (1) All information to diagnose the spacecraft is fed on-line through the network of the ISAS new ground operation system shown in Figure 1.
- (2) Whole areas of the spacecraft are repeatedly

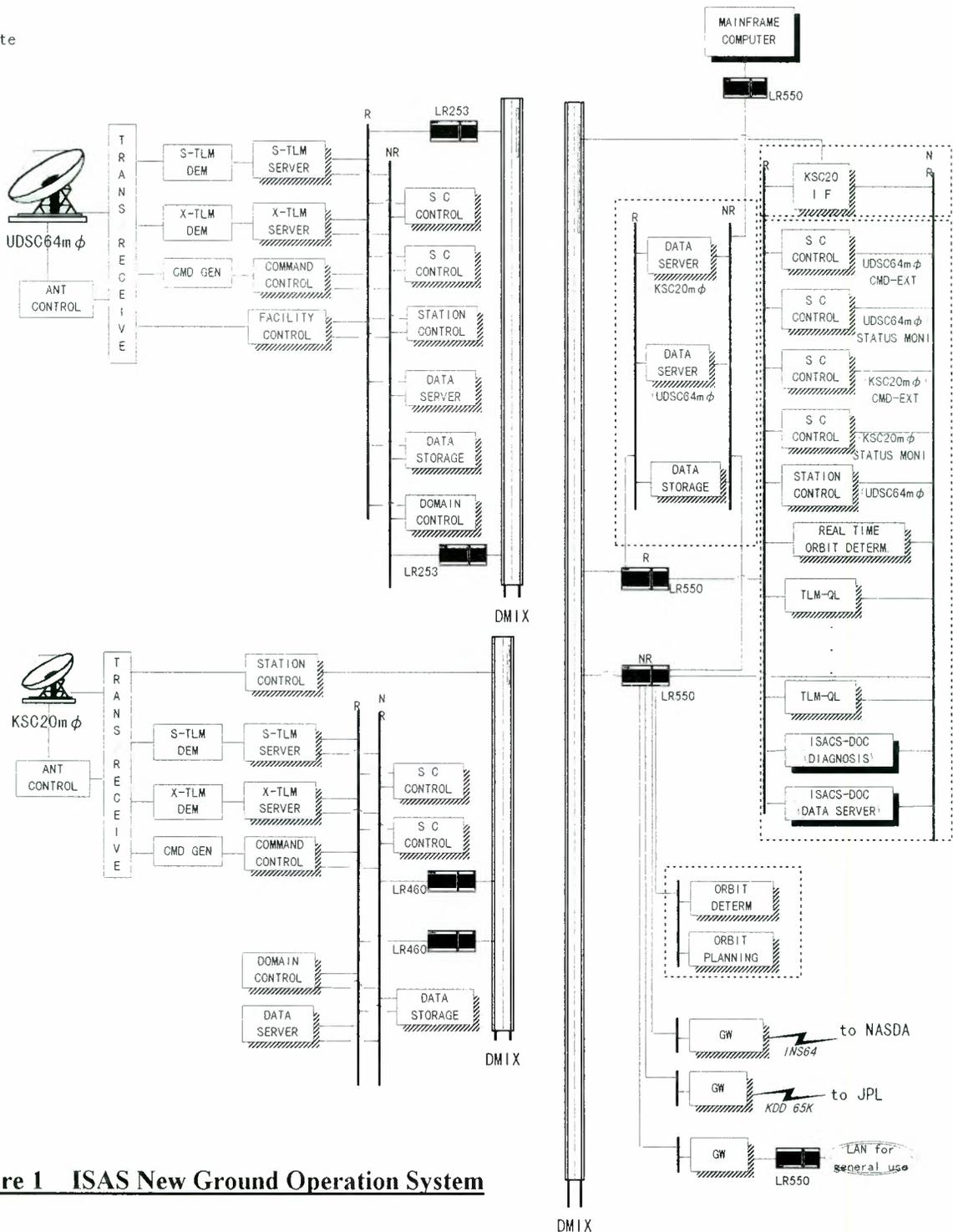


Figure 1 ISAS New Ground Operation System

diagnosed automatically during real-time operation.

- (3) The diagnosis covers almost all areas of the spacecraft's communication system, electrical power system, thermal control system, data handling system, attitude and orbit control system, and all scientific instruments onboard.
- (4) Diagnostic results are restricted to highly confident items to avoid confusing the spacecraft operators at SSOC. Facts or actual phenomena are only shown if the inference of the abnormality is vague.
- (5) Knowledge database can be easily updated according to the new situation of the spacecraft.
- (6) Communication links are carefully watched by comparing the real receiving levels of down-links and up-links with the estimated values which are calculated from antennas patterns, attitude of the spacecraft, distance between the spacecraft and the ground tracking antenna, performance of the ground station, etc.
- (7) All data used for diagnosis are saved in ISACS-DOC and can be used for checking the past status of the spacecraft.

Since "NOZOMI" is a deep space mission, communication links are evaluated with high accuracy in real time. Furthermore, electrical current monitoring for all onboard subsystems is introduced in designing "NOZOMI" to make diagnosing easier.

2.2 System Configuration

ISACS-DOC for "NOZOMI" consists of two computers: one is a SPARC workstation and the other

is a PC. The SPARC workstation handles on-line data feeding function. The on-line data feeding function automatically reads the latest real-time telemetry data, orbit and attitude data, and some of the ground operation systems' data through the network as shown in Figure 2. The system configuration of ISACS-DOC for "GEOTAIL" is also shown in Figure 2 for reference. This function also decodes the above data to physical values if it is necessary, and outputs a diagnostic information file by referring the Satellite Information Base (SIB) file which defines positions of telemetry frames and words, threshold values, and equations for decoding the data to physical values, etc. The diagnostic information file is automatically transferred to the diagnostic function on the PC. The PC handles the diagnostic function using a commercially available diagnostic expert tools package named APSHELL/DIAG 'Manadeshi-kun'; (meaning a favorite pupil) which runs in the Microsoft Windows NT environment to reduce the development cost. The APSHELL/DIAG 'Manadeshi-kun' has three standard functions: knowledge database editor, inference engine, and knowledge database. The diagnostic expert knowledge is sorted out as a tree form. Every condition to verify a tree node is determined by questions and results. The execution of diagnosis is conducted with the forward inference engine tracing the diagnosis tree by checking the conditions (truth or falsehood) of each node. The scale of knowledge database in ISACS-DOC for "NOZOMI" is shown in Table 1.

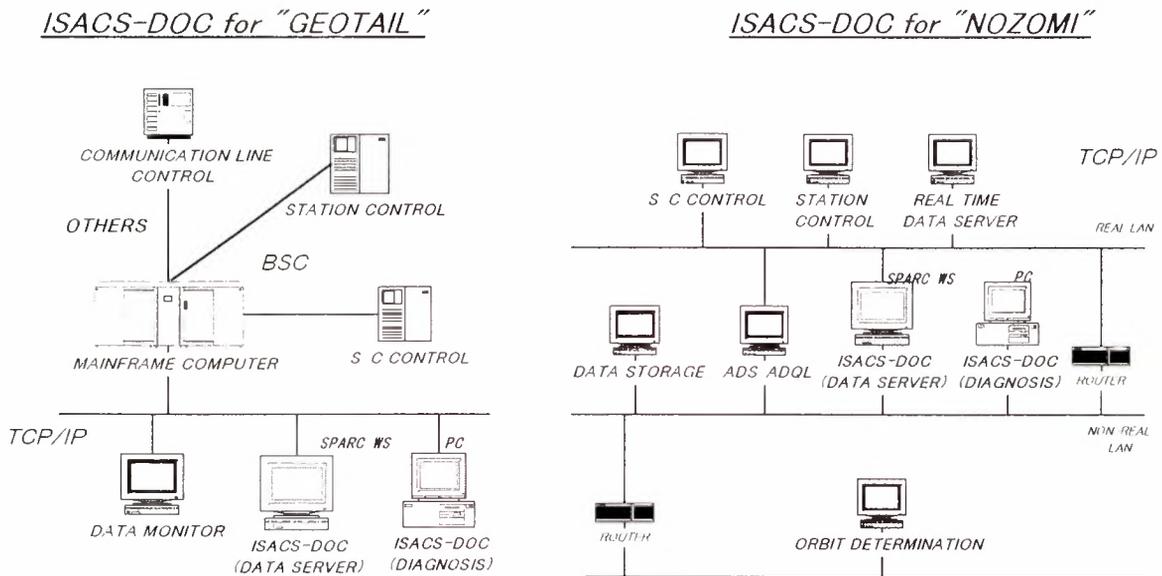


Figure 2 System Configuration of ISACS-DOC

Table 1 Scale of Knowledge Database in ISACS-DOC for "NOZOMI"

	Common Instruments					Scientific Instruments							Total	
	BCM	BPS	BAO	BHK	others	PSA	PWA	MPM	MIC	IMI	MDC	TPA		NMS
Questions	135	86	97	3	6	42	64	52	34	17	16	25	13	590
Results	61	64	101	1	-	35	64	36	29	15	14	18	12	450

BCM:	Communication System
BPS:	Power System
BAO:	Attitude and Orbit Control System
BHK:	House Keeping Items
PSA:	Particles Spectrum Analyzer
PWA:	Plasma Wave Analyzer
MPM:	Magnetic Field Measurement/ Probe for Electron Temperature/ Extendible Mast/ Ultraviolet Imaging Spectrometer
MIC:	Mars Imaging Camera
IMI:	Imaging Camera
MDC:	Mars Dust Counter
TPA:	Thermal Plasma Analyzer
NMS:	Neutral Mass Spectrometer

As mentioned above, the contents of the diagnosis knowledge database are changed according to the status of "NOZOMI". The numbers of questions and results are approximately 590 and 450 respectively at the present.

2.3 Process of Diagnosis

ISACS-DOC is a fully automatic and operator-less anomaly detective ground support system. One of the functions on the PC shown in Figure 2 refers the status of the SPARC workstation every 5 minutes regardless of the spacecraft operation time. When a set of necessary data for 10 minutes duration is prepared in the SPARC workstation, ISACS-DOC automatically diagnoses the whole areas of "NOZOMI". When the plural number of necessary data is included in 10 minutes duration, the statistical process (calculation of average, mean, maximum, minimum, or etc.) is practiced according to the data characteristics defined in the diagnostic information file. The ISACS-DOC warns the operators when it finds something abnormal. The operators can know the details of the abnormality as well as the first aid action to avoid fatal damage that may be caused by the abnormality. The diagnosis is repeatedly practiced about every 5 minutes during "NOZOMI" real-time operation excepting the duration of the recorded data reproducing. We found the diagnosis using the recorded data was very useful to know what happened while the spacecraft was invisible from SSOC. This function is strongly required to actualize more useful anomaly detective ground support system.

2.4 Display of Diagnostic Results

Five standard windows on the screen of ISACS-DOC for "NOZOMI" are prepared to display the following information:

- Abnormal items.
- Explanation of each abnormality.
- Normal and actual status/values causing the abnormality.
- Related figures/graphic data.
- Urgent level, contact information such as telephone numbers of senior engineers or scientists who can supervise the further contingency operation for the abnormality.
- Urgent commands to save the probe from catastrophe (first-aid commands), and
- Some common data like distance between the probe and the earth.

Figure 3 shows an example of the screen display, and Figure 4 is the printed results by ISACS-DOC.

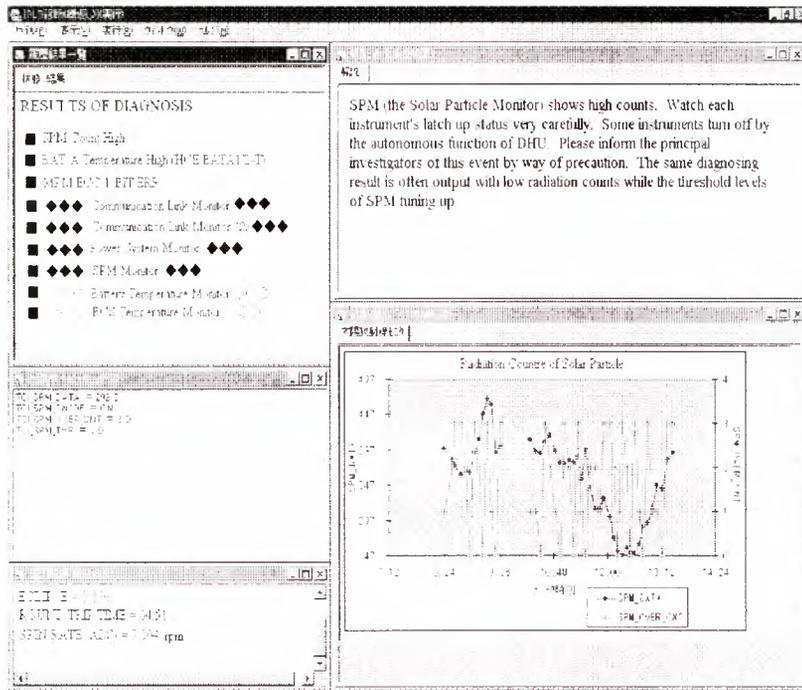
3. OPERATING STATUS OF ISACS-DOC FOR "NOZOMI"

It is very important to verify the definitions of the knowledge database in order to actualize the reliable anomaly detective system. The knowledge database has been checked and maintained by comparing the diagnosing results of ISACS-DOC with the actual spacecraft status since "NOZOMI" launching. Additionally, the "NOZOMI" initial orbit plan arriving in the Martian orbit on October 1999 was changed to arriving at the end of 2003 by a

Window 1
Results of
Diagnosis

Window 2
Actual
Status/Values

Window 3
Common
Information



Window 4
Explanation of
Abnormality

Window 5
Useful Figures/
Graphic Chart

Figure 3 Example of Screen Display

malfunction of the propulsion system in December 1998. The tuning of ISACS-DOC has been almost completed and is already used to assist the daily operation of "NOZOMI". This system will be fully operated regularly from June 1999 at SSOC. For about ten months of test-running, ISACS-DOC has been refined on the following points:

- (1) The communication link levels can be evaluated within $\pm 3.0\text{dBm}$ accuracy.
- (2) Six different kinds of trend graphs to monitor communication links, power system, propulsion system, and solar flare can be always referred regardless of the diagnosis results.
- (3) The important values/status relevant to the abnormality can be referred at a glance on ISACS-DOC screen.
- (4) The definition of diagnosis knowledge database has been modified according to the actual situation of "NOZOMI".
- (5) The explanation of the abnormality has been modified reflecting the actual satellite operation.

Table 2 shows some examples of "NOZOMI" abnormality that ISACS-DOC found in the actual operation.

4. CONCLUSION

The development and tuning of the anomaly detective ground support system for Mars probe "NOZOMI"

has almost been completed and this system will be used to keep the safer operation of "NOZOMI" on daily basis till the end of "NOZOMI" mission. The effectiveness of ISACS-DOC has already been shown by finding some abnormalities of "NOZOMI" during its test-running. Especially the communication link levels are evaluated with very high accuracy after three times revision of the calculation algorithm of the estimation. This accuracy will be sufficient enough to operate "NOZOMI" safely in the trans Mars orbit and also in the Mars orbiting phase after the end of 2003. It is strongly required to append the diagnosis function for the reproduced data in order to perform more reliable anomaly detection. We are considering the way to add this function without making much alteration to the present version of ISACS-DOC.

ACKNOWLEDGMENT

The author would like to thank all of the people for giving us precious information of each onboard instrument of "NOZOMI" and ground support system. This information is essential to construct the knowledge database of the ISACS-DOC. The authors also would like to acknowledge Professors Ichiro Nakatani, Toshifumi Mukai, and Hajime Hayakawa of ISAS for their tremendous support of this development.

DIAGNOSIS RESULT: Radiation Count of Solar Particle High (TCI_SPM_OVER_CNT)

COMMON INFORMATION:

- PASS No: 9905060100
- TIME: 1999/05/06/ 12:52:56
- EARTH-PROBE RANGE: 43797066.1 Km
- ECLIPSE RATE: 0.0 %
- TRANSMISSION DELAY: 04:51
- θ_e (ADS): +3.1374 degree
- θ_s (ADS): +45.600 degree
- SPIN RATE (ADS): 7.804 rpm

EXPLANATION OF ABNORMALITY

STATUS: CAUTION! TCI_SPM_OVER_CNT= High

SPM (Solar Particle Monitor) shows high counts. Watch latching up of each instrument carefully. Some instruments may be turned off by the autonomous function of DHU. Report to the persons in charge of today's operation. False diagnosis results may be output with low radiation level till adjustment of SPM threshold levels is completed.

REFERENCED DATA:

TCI_SPM_DATA = 1 D
 TCI_SPM_ON/OFF = ON
 TCI_SPM_OVER_CNT = 2 D
 TCI_SPM_THR = ***

REFERENCE GRAPH:

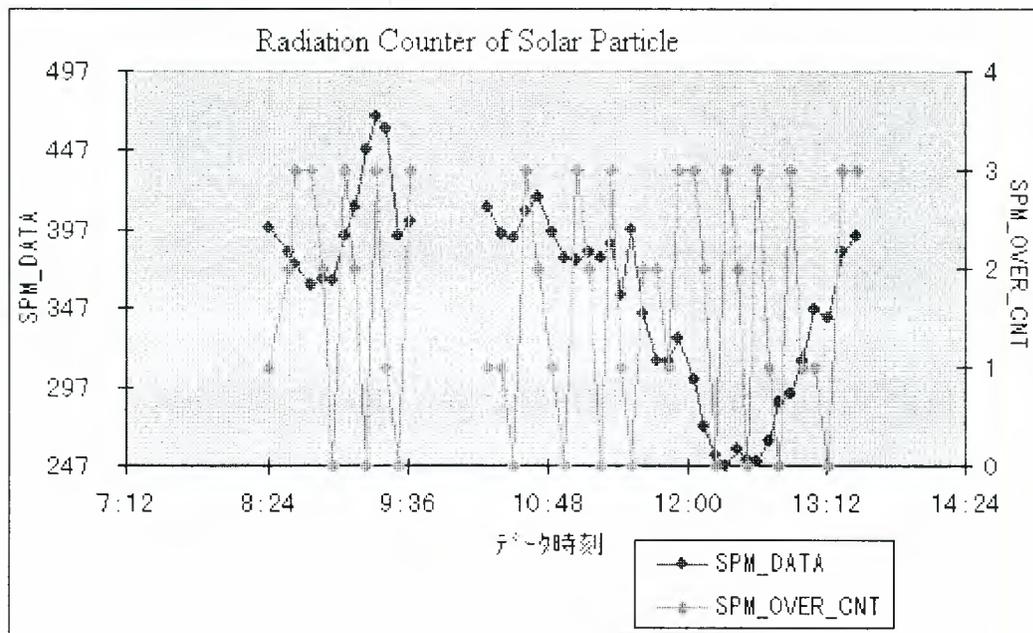


Figure 4 Example of Abnormalities Found and Printed out by ISACS-DOC

Table 2 Examples of “NOZOMI” Abnormality

1	TMS Temperature Out of Upper Limit → Required range = -20 ~ 45 C. It occurred during near-earth orbit. TMS temperature was carefully watched.	Oct., 1998
2	Latching Valve Status Error (LV-2 = CLOSE) → Malfunction of the propulsion system occurred during maneuver operation. Orbit plan of “NOZOMI” was changed.	Dec., 1998
3	MPM ECC 1-BIT Error → MPM ECC 1-bit error occurred. It was recovered by sending commands.	Mar., 1999
4	BAT-A Temperature Out of Upper Caution Limit → Required range = 5 ~ 15 C. It occurred during near-earth orbit Battery-A temperature was carefully watched.	Mar., 1999
5	MDC Impact Counter Full → Dust impact counter memory was full. Impact data were downloaded.	Apr. 1999
6	TMS-BASE_BAND S/N Out of Lower Limit → Communication link margin was small. Telemetry bit rate was lowered.	May, 1999
7	Solar-Particle-Counter High → It was caused by Solar high activity. Activity level was carefully watched.	May, 1999

REFERENCES

- (1) Hashimoto, M., Nishigori, N., and Mizutani, M.: Monitoring and Diagnostic Expert System for Scientific Spacecraft: Evolution for PLANET-B Probe. Proceedings of the twenty-first International Symposium on Space Technology and Science. OMIYA, Japan, Volume 1, ISTS98-e-12, pp.1033-1038, 1998.
- (2) Hashimoto, M., Nishigori, N., and Mizutani, M.: Operating Status of Monitoring and Diagnostic Expert System for Geomagnetic Satellite GEOTAIL. Proceedings of the 2nd International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations', Oxford, UK, RAL.GS2.73, pp.73.1-73.8, 1997.

A NEW DESIGN APPROACH OF SOFTWARE ARCHITECTURE FOR AN AUTONOMOUS OBSERVATION SATELLITE

Jérôme Gout, Sara Fleury

LAAS-CNRS

7 Av. du Colonel Roche - 31077 Toulouse Cedex 4 - France
Jerome.Gout@laas.fr Sara.Fleury@laas.fr

Hervé Schindler

Matra Marconi Space

31 Av des Cosmonautes - 31402 Toulouse Cedex 4 - France
Herve.Schindler@tls.mms.fr

Abstract

The new generation of satellites are complex autonomous systems that present similarities with autonomous robots. This paper analyses a design approach and tools taken from robotic research. It focuses on the knowledge representations handled by the different architecture components and on the problems arising from the integration of the decision capacities (incremental reactive planner). Formal models of actions and tasks from which local representations of a varied nature are automatically derived, ensure the consistency of the data.

The approach has been evaluated using a real system specification and a complete architectural instance has been implemented from the low level real-time control routines to the high level missions.

1 Introduction

The new generation of satellites have to fit constraints that have important repercussions on the design philosophy: lighter satellites, reduction of the design process duration, use of "standard" components, simplification of the on-site maintenance (in particular the heavy control from the central ground station), and so on. These constraints lead to satellite designs which manage both *reactive and decision capacities on-board* the system, that is *autonomous satellites*.

In this paper, we propose an approach along with tools to design software architectures which are inspired by work in autonomous robot. To illustrate and demonstrate the relevance of this approach, we have considered, in collaboration with Matra Marconi Space¹, the example of an autonomous observation

satellite.

The main objective of these new satellites is to allow *direct access* to end-users using the World Wide Web through ground stations situated all around the world. Thus, unlike SPOT satellites for instance, it must offer high level interactions (e.g., "Take a photo of Noordwijk between 9AM and 10AM local time") and must be able to integrate all the client requests.

Consequently, the system must be able to manage the *planning* of the actions required to accomplish the missions (maneuvers, image processing, data down-loadings, ...) and their *execution control*, including failure recovery and redundancy management, *on-board*.

In order to integrate all these capacities we propose a generic software architecture structured in two main hierarchic levels (section 2). A lower functional level which embeds all the basic capabilities of the system (device control, servo-control, monitoring, etc) is controlled by an upper decision level that plans and controls the execution of the operations required to accomplish a mission.

We consider the elaboration of a real-time, modular and controllable functional level, using tools such as the Generator of Modules GenoM, a mastered operation and we therefor focus on the decisional level. Indeed, if the organization of this level is also well defined and if different tools exist to implement its components, the actual realization of such complex systems still raises important difficulties, the major ones being:

- *the knowledge representations*: the different components of the architecture have to handle and to share data of a varied nature (static models, dynamic state vectors, numerical/symbolical data,

tributed Systems).

¹This collaboration has been supported by the Région Midi-Pyrénées (France) within the project SyDRE **S**ystèmes **D**istribués **R**éactifs **E**mbarqués (On-Board Reactive Dis-

etc). To avoid redundancies and to ensure the consistency of the system we propose a unified knowledge representation associated with an automatic synthesis of the local models (section 3).

- *the master of algorithm complexity* which requires automatic synthesis based on validated models and the use of generic tools and control algorithms (section 4.2).
- *the integration of a reactive temporal planner* in the decision level raises problems related to *incremental planning* and to the synchronization between the future plan (elaborated from a predicted state and models of actions) and the on-going execution (section 4.3).

A complete integration based on a simulated satellite will illustrate our approach (section 5).

2 Software Architecture Overview

In order to reconcile both decision and real-time capacities on board an autonomous robot, a generic software architecture composed of 2 hierarchic levels has been developed [1] :

- At the lower level a reactive distributed *functional level* embeds all the operational functions (control of devices, processing, ...).
- At the upper level, the *decisional level* decides which actions are to be executed according to the mission and the state of the system and controls their execution at the lower level.

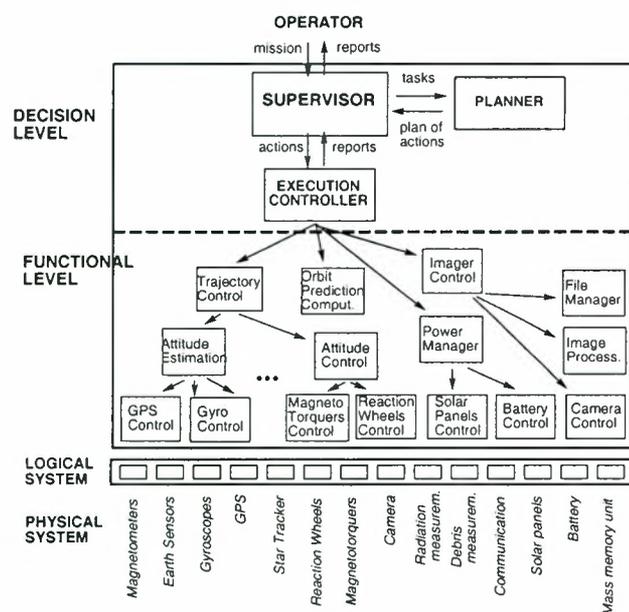


Figure 1: A generic architecture instantiated for an autonomous observation satellite. The functional level embeds about 20 modules generated by GenoM.

2.1 The Decision Level

The decisional part is responsible for mission management and for the control of the on-board system: it has to interpret the mission, to plan the adequate sequence of actions according to the current state of the system and to control on-board execution. It is composed of three entities: a supervisor, a planner and an execution controller. The planner is used as a resource by the supervisor which actually interacts with the next level, controls the execution of the plan and reacts to incoming events. An instance of this level will be presented in the following sections.

2.2 The Functional Level

The actions are executed at the functional level which embeds all the operation functions like the control of the various hardware devices (magneto-meters, earth sensors, gyroscope, gps, reaction wheels, ...), and also processing like orbit prediction or image processing. This level is organised as a network of modules: the functions are embedded in independent modules that have the responsibility of physical or logical resources.

The modules are capable of performing a number of specific services by processing inputs from, or acting on, physical robot devices and/or other modules. The services are parameterised and activated asynchronously through a non-blocking client/server protocol: a relevant *request*, that may include input parameters, applies to every service of each module. Thus requests start processings. The end of the service is marked by a reply returned to the client that includes an *execution report* and possibly data results.

For this application we have developed about 20 modules: one basic module for each hardware device (sensors, actuators, payloads, communication), one for the orbit prediction computation, and several to estimate and to servo-control the attitude using the previous basic ones (Figure 1).

Every module at the functional level is an instance of a generic model. They are automatically generated using the generator of modules GenoM which simplifies the design process and ensures a correct implementation (see [2]).

3 Knowledge Representations

A unified and consistent knowledge representation is fundamental to design and implement complex high level software architectures. Only a unified and formalized knowledge representation:

- ensures the consistency of the local representation of the different architectural components;
- allows the use of generic, and thus validated, architectural components;
- allows automatic code synthesis.

However if all the components of the architecture (i.e., the supervisor, the execution controller and the planner for the proposed architecture) *in fine* reason on, or handle, the same low level functional operators of the functional level, they do not consider the same properties of these operators.

Whereas the activities of the functional level are essentially characterized by numerical processing, the decisional level needs an abstract and symbolical representation (effects, conditions, resources used) to organize their execution.

Moreover, each component of this decision level requires specific knowledge:

- for planning purposes the planner needs to know their effects, their (pre-)conditions, their duration, their resource consumption, using or production, etc.;
- the supervisor that supervises the correct execution of the plan of actions and implements the failure recovery needs to know all potential malfunctions of every action.
- and finally the execution controller needs to know how to control (start, stop or parameterize) these actions at the lower level.

From these considerations we have elaborated *action models* that can be seen as an abstraction of module requests. Thus these actions modelise the low level operators and fill the gap between the functional and the decisional levels.

The high level missions will be realized by combining these basic actions into *tasks*. The tasks are sequences of actions that allow us to predefine complex operators. They fill the gap between the high level missions and the on-board capacities (the actions).

The actions represent the basic platform-dependent capacities of the system (control of hardware devices, servo-control, monitoring, filtering, etc), whereas the tasks represent complex application-dependent functional capacities.

Action and task representations must be formal to allow automatic synthesis and reasoning both for the planning and the supervising processes.

3.1 Action Representation

The decisional level has to decide which of the operational functions (ie, of the requests) are to be executed at the lower level. However, it can not reason directly on the request descriptions provided by GenoM. These descriptions are only functional and do not integrate information related to their conditions or effects on the state vector of the agent. Moreover this data cannot be added to the module description as they depend on the application context (eg: according to the situation, satellite maneuvers can be allowed or not during image acquisitions).

Thus, from a bottom-up view point, the requests have been enriched with the actions: actions are extensions of module requests with semantic information.

The description of an action is basically composed of two parts :

- structural and functional information (name, requests involved, all possible termination status, ...) to control the execution of the action when required;
- resource and logical information (effects on the resources, conditions and effects on agent state) which allow reasoning about the usage and consequences of the action.

Actions are defined as a list of (attribute: value) couples containing executive service name (*service*), the non-nominal possible terminations (*end_slots*), resource usage/production and consumption (*uses*, *consumes*, *effects*); the conditions and effects specifications (*assertions*, *effects*).

The following example is the action CAMERA that allows to take an image:

```

action CAMERA {
  service      : take_image;
  concurrence  : interrupt;
  end_slots    : cam_hard_failed, cam_soft_failed;
  uses         : camera(1);
  consumes    : power(20)@start,
               mmu(100)@start;
  produces    : image(1)@ok;
  assertions   : on_zone() = ?zone in [start,ok];
  effects     : image(?zone) = taken @ok;
};

```

The actions are characterized by the following properties:

- An action starts on the controllable² event *start/-*.
- The end of an action is associated with the terminal contingent event³ *-/end*.
- The terminal event is always associated with a report that characterizes how the action has ended (the default report is *ok*).
- An action may be interrupted by the controllable event *kill/-*.
- An action may produce intermediate contingent events. The default intermediate event *-/started* confirms its starting.

Actions are the smallest entity handled at the upper levels of the architecture. The decisional level can

²This event is controllable from the decisional level view point. The controllable events are noted *evt/-*.

³This event is contingent from the decisional level view point. The contingent events are noted *-/evt*.

act on the system only using the controllable events `start/-` and `kill/-`, and the evolution of the system is perceptible (measurable) only through the incoming contingent event `-/started` and `-/end` (associated with their terminal reports). The state vector of the system is the integration of all these events.

From the textual description of an action a graphical representation can be derived (Figure 2). This representation will be used to define tasks.

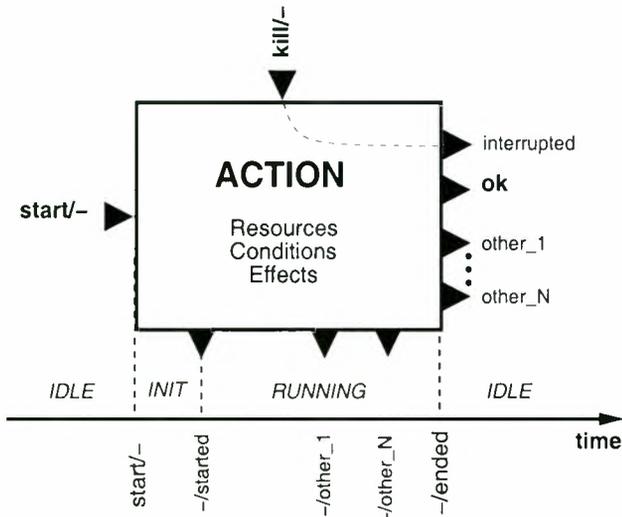


Figure 2: Graphic and synthetic view of action. The time runs from left to right. The incoming black arrows (or slots) receive the `start/-` (on the left) and the `kill /-` (on the top) events. The outgoing ones export the events produced by the action during its execution (on the bottom) or at its end (on the right). The end slots on the right are exclusive and allow expression of conditional tasks according to termination reports.

3.2 Task Representation

The task model is defined as a complex combination of actions, expressing control and ordering information of actions. Formally, a task can be defined as $\{\{A_i\}, \mathcal{R}\}$, $\{A_i\}$ being a set of actions and \mathcal{R} being a partial order relation between these actions.

The tasks are designed by the operator using an intuitive graphical tool called *TaskBuilder*. Within this environment, the actions are composed using their graphical representation: “contingent” slots (ie, intermediate or terminal slots) are linked to “controllable” ones (ie, start or kill slots). In such a way, one can express part of known plans or skeletons (partial graphs) of actions that represent complex satellite processes, including failure detections and recovery actions (using the different termination slots).

Figure 3 presents a simple example of a task with three different actions. The nominal process of this

task involves two actions: *CAMERA* to take an image and *DOWN_LOAD* to down-load it to a ground station. In case of camera failure the *CHECK_UP* action is invoked.

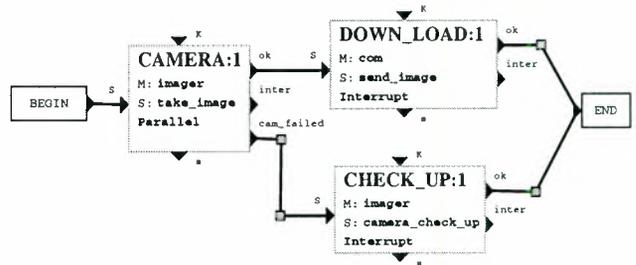


Figure 3: An example of task composed of three actions. According to the report of the *CAMERA* action, the system down-loads the image or checks the equipment.

3.3 Automatic synthesis of the models

The action and task formal models have been designed to unify the representations handled in the architecture. From the action and task descriptions *TaskBuilder* automatically produces (Figure 4):

- the upper execution controller procedures that allow execution of the different actions by sending the adequate module requests;
- the planning operators for the planner that are elaborated from both the resource and logical information on the actions involved in the task, and the partial order relation between these actions;
- the supervision procedures elaborated from the relations between the all the events and their termination or intermediate reports.

Because of the limitations of planning algorithm (*i.e.* no handling of conditional plans) the planning operators derived from the complete task model, contain only the subset of nominal actions and their relations. The supervision procedures integrate the complete description including nominal and non-nominal events and actions. The contingent events that are not explicitly considered in the task description, implicitly aim to a failure state and a survival mode of the satellite if they occur during the task execution.

4 Decision Level Integration

4.1 The execution controller

The execution controller, or executive, interfaces the functional and the decision levels. It is a purely reactive system without reasoning. It controls the module requests according to the “controllable” events coming from the supervisor, and returns “contingent” events from the execution reports. An action can involve several module requests coordinated by the executive.

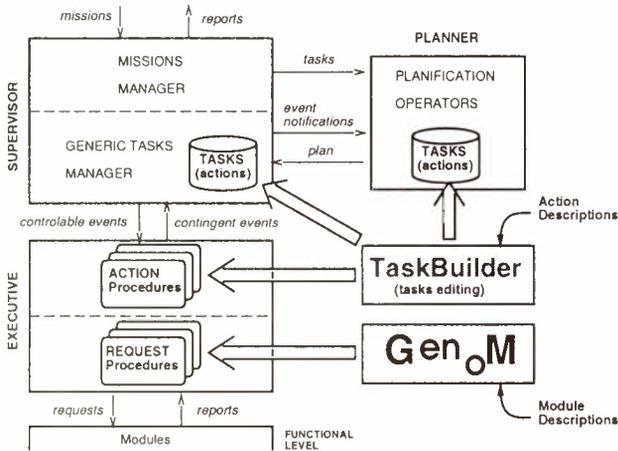


Figure 4: Overview of the origin and the distribution of the knowledge representations (in bold) and the dynamically exchange data (in italics) in the architecture.

The executive maintains a state vector of the functional level and manages the conflicts between module requests.

The executive is written with PRS and its procedures have been entirely produced by the couple of systems: TaskBuilder and GenOM (Figure 4).

4.2 The Supervisor

As presented in section 2, the supervisor is actually the conductor of the decisional level. It is in relation with:

- the users, through mission requests;
- the planner, through planning requests and the returned solution plans;
- the execution controller, through action events.

The supervision system is divided into two different subsystems (Figure 1). The upper one deals with the clients' requests. It receives mission observations, processes them taking into account client priority and flight over target area. According to these data the supervision sends a planning request to the planner. This part is application dependent as it contains all the application specificities due to the interface between users and the autonomous satellite.

The lower subsystem is in charge of the execution and the supervision of the plan produced by the planner. Unlike the upper one, this subsystem is *generic*. Following the dynamic task plan and the task models produced by TaskBuilder, it starts actions by sending the adequate events to the executive and integrates returned feedbacks to make the plan progress and to control its execution.

The plan maintained by the supervisor is composed of the 3 following parts:

- Static description of task produced from the formal model by TaskBuilder.
- Numerical temporal data: the planner returns a temporal window for each event corresponding to an action event contained in the newly inserted tasks of the plan
- Symbolic temporal relations between events of different tasks.

The two last parts of the plan are dynamically produced by the planner in response to planning requests.

The supervision of the plan is achieved by analysing received feedbacks from the executive. According to the execution report of the actions, the plan progresses following nominal or non-nominal branches of the complete static description. A fatal error occurs when there are no action operations (starting or interruption) associated with a received execution report. The satellite switches automatically to a survival mode, closing all client connections and waiting for the maintainer's intervention.

Our supervision system is implemented using a tool called PRS (for Procedural Reasoning System) ([3, 4]).

4.3 Planner

The asynchronous arrival of numerous client requests, their strong temporal constraints (ie, the communication or image acquisition temporal windows), the inaccuracy of the orbit prediction in long term, the important resource constraints (eg, shared hardware devices, images storage capacities before their download, limited energy capacities between two battery rechargings with the solar panels, etc) call for an efficient *incremental temporal planner*.

This incremental planning has been elaborated upon IxTET, a temporal planner developed in our research group ([5]).

A temporal planner. IxTET is a general and open planner (see [6] for an adapted planner). Its formalism is based on a reified temporal logic that defines several temporal predicates on both state and resource attributes. State attributes are handled with the *event* predicate to express a change in the world, whereas the persistence can be expressed by the *hold* predicate. Concerning resource management, one can express resource usage during a determined time interval (*use* predicate) or resource production (*produce*) and consumption (*consume*). IxTET's algorithms are sound and complete. Until now IxTET has not been integrated with a supervisor as an incremental reactive planner.

An incremental planner. The incremental planner, based on an IxTET's kernel, has to maintain a global historical plan of all missions sent by the clients, updating it for each new request.

Thus, the planner is a plan server running concurrently to the supervisor: the supervisor sends a plan-

ning request to the planner and get back a new temporal plan. This dynamic link between the supervisor and planner builds the planning problem online, using the planning operators synthesized by *TaskBuilder* from the task descriptions.

A reactive planner. Another problem to integrate the planner/supervisor couple is related to the synchronisation of the plans. In order to predict the future state of the system, the planner maintains a global plan elaborated from the task and action *models* that will quickly diverge from the real executed plan without synchronisation. Thus, several synchronisation operators have been added allowing the supervisor to update the planner plan:

- the planner gets the real execution date of all events as they occur,
- the resource and logic states predicted by the planner are updated by the supervisor according to the *real* feedback of the actions,
- the planner can retract tasks from its plan in case of failures or mission abortion.

In return, the planner informs the supervisor about a new plan insertion resulting from a new planning request. A translation between the planning representation and the supervision one is necessary, including completion of a nominal plan by failure recovery actions. Therefore, the supervisor plan model is dynamically updated.

Finally, the planning time process must be bounded to ensure the global system dynamics.

5 Application to an Autonomous Satellite

The presented methodology and tools to design the software architecture of autonomous systems have been evaluated on a future autonomous observation satellite project.

To run the whole system, we have implemented a simulator to emulate the physical system: the earth rotation, the orbital motion of the satellite (including noises), the energy consumption of the hardware devices, the occurrence of failures, and so on.

An example of a user mission is presented bellow. It corresponds to an image acquisition request with parameters including: the target area (Toulouse), some constraints for the image acquisition (local time interval, maximum inclination, type of camera ...) and on-board image processing, the client identification (in particular to know where to down-load the image) and the request priority (in case of resource saturation).

```
(IMAGE-RQST (ZONE TOULOUSE 43.62 1.45 30
              12:00 13:00)
 (IMAGE_HIGH NONE NONE (..)
 878 12})
```

Once selected by the supervisor, this client request is translated to a planning request and sent to the planner.

This mission uses the task *take_image* which contains 4 main steps involving 8 basic actions:

1. satellite orientation (SLEW action)
2. image acquisition (CAMERA, ZONE_IN and ZONE_OUT actions)
3. data processing (IMAGE_PROC action)
4. processed image down-loading to the client's ground station (DOWNLOAD, ZONE_IN and ZONE_OUT actions).

ZONE_IN and ZONE_OUT are monitoring actions that allow detection of the entrance or the exit of the satellite over a given area (for photographic or for communication purpose): a contingent event is returned to the decisional level once the monitored condition is satisfied.

Note that a single action may involve several module requests at the functional level. For instance, the SLEW action requires several sensors and actuators to estimate and control the attitude of the satellite.

The average time taken by the planner to find the solution plan is about 2 seconds⁴. The resultant plan is presented in figure 5. In this example the request has been integrated in a plan that already contained 2 previous user requests. Only three planning operators have been used here (the two first operations of the *take_image* task have been gathered) TAKE_IMAGE, DSP, DOWN_LOAD.

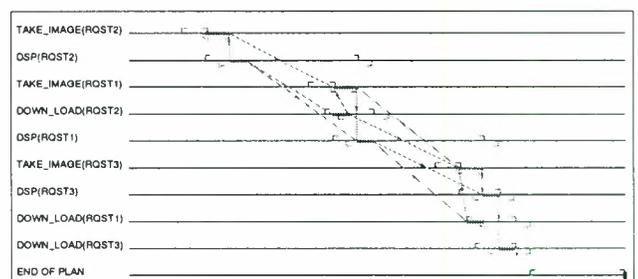


Figure 5: *An example of temporal plan. The operators (listed on the left) of three different tasks (RQST1, RQST2, RQST3) are intertwined. The arrows represent their precedence constraints.*

⁴Let us recall that the time allocated to the planner to find out a solution is bounded. The request is put back to the mission queue if no solution is found.

The planner translates the plan as a chronology of events for the supervisor. The following PRS facts give an illustration of that data. The first group shows symbolic temporal constraints between events belonging to different tasks.

```
(EPP 1 (EVENT TI END CAMERA 1) 1 (EVENT TI END ORBCP_OUT 1))
(EPP 1 (EVENT TI END CAMERA 1) 2 (EVENT TI START SLEW 1))
(EPP 1 (EVENT TI END ORBCP_IN 1) 1 (EVENT TI START ORBCP_OUT 1))
```

The next 3 facts express the temporal window of each event of the plan.

```
(ETW 1 (EVENT TI START ORBCP_IN 1) 300 3889 642)
(ETW 1 (EVENT TI START ORBCP_OUT 1) 3600 4189 -1)
(ETW 1 (EVENT TI START SLEW 1) 0 3589 235)
```

And finally, to complete this data dynamically produced by the planner, the supervisor uses PRS procedures, synthesized by TaskBuilder from the static description of the tasks (pair of condition/action-like rules).

```
(ETP (. (EVENT TI END CAMERA 1) ok .) (EVENT TI KILL ORBCP_OUT 1))
(ETP (. (EVENT TI END ORBCP_OUT 1) inter .) (EVENT TI START DSP 1))
(ETP (. (EVENT TI END DSP 1) ok .) (END))
```

Figure 6 is a screen copy of an experiment session. One can distinguish the supervisor (top left), the execution controller (top right), the user console (bottom left), the planner (bottom right) and the simulator in the center showing the satellite and the current area being flown over.

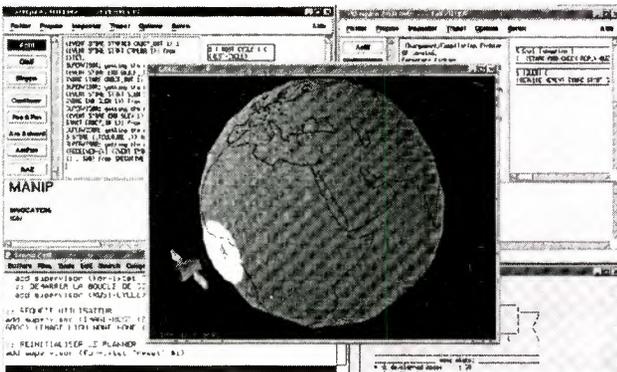


Figure 6: A snapshot of the experiment running

6 Conclusion

The objective of this study is to analyze and demonstrate concepts and tools taken from research in robotics to design the new generation of autonomous satellites. Formal hierarchic models of knowledge representations (*i.e.* **action** and **task**) have been proposed and allowed to:

- eliminate the description redundancies,

- automatically produce the derived instances handled by the different architecture components,
- ensure the consistency between model instances.

Concerning the deliberative processes, we have implemented the integration of planning within the dynamic loop of execution/supervision. This includes an incremental planning based on I^2T and an extension of the planning capabilities (necessity to extend the task insertion control).

Actions and tasks describe both nominal and non-nominal situations that are managed by the supervisor according to the feedback received from the functional level.

The executive and the lower part of the supervisor are generic and handle models automatically synthesized by *GenoM* and *TaskBuilder*. This simple the integration procedure masters the complexity of the system.

The approach has been evaluated using a real specification, and a complete architectural instance has been implemented from the low level real-time control routines to the highest level missions.

References

- [1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *Special Issue of the International Journal of Robotics Research on Integrated Architectures for Robot Control and Programming*, 17(4):315–337, April 1998. Rapport LAAS N97352, Septembre 1997, 46p.
- [2] S. Fleury, M. Herrb, and R. Chatila. *Genom*: A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *International Conference on Intelligent Robots and Systems (IROS'97)*, Grenoble, France, 1997.
- [3] F. Ingrand, M. Georgeff, and A. Rao. An architecture for real-time reasoning and system control. *IEEE Expert, Knowledge-Based Diagnosis in Process Engineering*, 7(6):p. 33–44, December 1992.
- [4] F. Ingrand, R. Chatila, R. Alami, and F. Robert. Prs: A high level supervision and control language for autonomous mobile robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA'96)*, Minneapolis (USA), June 1996.
- [5] M. Ghallab and H. Philippe. A compiler for real-time Knowledge-based Systems. In *International Workshop on Artificial Intelligence for Industrial Applications*, Hitachi City, Japan, May 1988.
- [6] P. Dago. *Extention d'algorithmes dans le cadre des problèmes de satisfaction de contraintes valuées: application à l'ordonnancement de systèmes satellitaires*. PhD thesis, Ecole National Supérieure de l'Aéronautique et de l'Espace, France, 1997.

Robot Operations Preparation and Commanding

MISSION PREPARATION AND TRAINING FACILITY FOR THE EUROPEAN ROBOTIC ARM (ERA)

Zeholij Pronk

National Aerospace Laboratory NLR, Space Division,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Phone: +31 527 8223, fax: +31 527 8210, e-mail: pronk@nlr.nl

Marcel Schoonmade, NLR,

Phone: +31 527 8219, fax: +31 527 8210, e-mail: mschoonm@nlr.nl

Waheed Baig

Spacebel Informatique, Space Business Unit
I.Vandammestraat, 5-7, 1560 Hoeilaart, Belgium
Phone: +32 2 658 2053, fax: +32 2 658 2090, e-mail: waheed@spacebel.be

ABSTRACT

In 2000 the European Robotic Arm (ERA) will be launched and attached to the Russian segment of International Space Station Alpha. The arm will initially be used to support assembly operations on the Russian segment, and will eventually be used as servicing tool for at least ten years during Space Station life cycle. The Mission Preparation and Training Equipment (MPTE) is an important part of the ERA project. ERA operations will be prepared, planned, and supported from the MPTE, and ERA operators will be trained with the MPTE. Three identical versions of the MPTE will be installed at RSC/Energia-MCC and at the Gagarin Cosmonaut Training Centre in Russia, and at ESA/ESTEC in the Netherlands. Each of the facilities has its particular function in support of ERA operations, training, and maintenance.

The design of the MPTE is based on existing tools and facilities to a maximum extent. Re-use is made of the real-time operations simulation facility EUROSIM, including the Image Generation Subsystem (IGS), and of the Columbus Ground Software system (CGS). Also, re-use is made of developments from the ERA projects, both hardware and software. Next to elegant breadboard systems of ERA flight hardware, development support simulation models from the ERA Simulations Facility (ESF) are re-used.

1. INTRODUCTION

The European Robotic Arm (ERA) will be one of the major European contributions to the operational capabilities of the Russian Segment (RS) of the International Space Station (ISS). Under contract with the European Space Agency (ESA), Fokker Space is leading a European consortium developing the arm and its ground support facilities (Ref. 1). The baseline of

ERA's missions will be to support assembly and, later on, servicing of the Russian Segment of Space Station. The arm will be controlled by cosmonauts in EVA (Extra-Vehicular Activity) or in IVA (Intra-Vehicular Activity) (See figure 1).

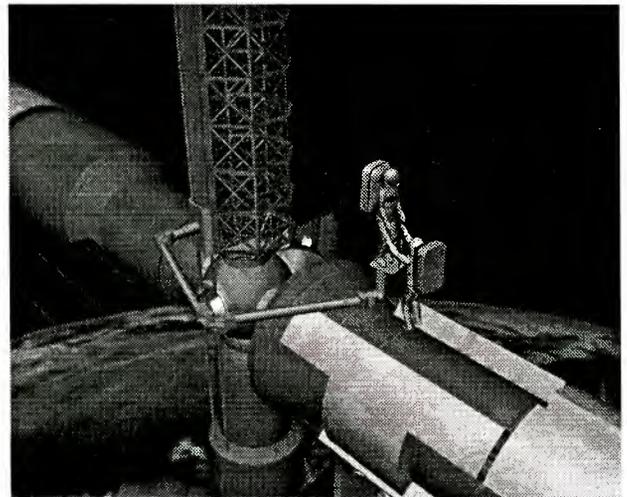


Figure 1: ERA on the Russian Segment of International Space Station, operated with two cosmonauts in EVA

The manipulator consists of seven rotational axes in an anthropomorphic configuration. In nominal operations only 6 Degrees of Freedom (DOF) will be operated, one DOF (shoulder) will be locked. The symmetric design with two 3 DOF wrists, one elbow joint, and two multifunctional end-effectors make ERA a re-locatable arm and allow ERA to move along the Space Station. A number of base-points is installed on the Russian Segment to allow ERA to reach the assembly and servicing sites of the Space Station. The ERA is capable of transfer of different kind of payload, from standard Orbit Replace-able Units (ORU) of a few hundred kilogram, to large payload of several thousand

kilogram. For this purpose the control system deals with payload classes to optimise arm and joint control characteristics. In order to move payload over more than one base-point, so-called intermediate Payload Mounting Units (PMU) are made available.

The ERA Control Computer (ECC) is integrated in the arm structure. It allows communication, power distribution, and vision processing. Therefore, ERA is rather independent of the Space Station systems. The ERA Camera and Lighting Unit (CLU) provides for visual interaction from inside the Space Station with external operations and for proximity sensing. The ERA has two cameras at the elbow and two cameras at the end-effector, a symmetric configuration. Next to the ERA cameras RS cameras and lighting units support the ERA operations, dependent on operational conditions.

The ECC is connected to the RS onboard computer, the Central Post Computer (CPC) and the RS Mass Memory Unit (MMU) via the RS Space Station bus (Mil-std-1553 bus). The CPC and MMU provide operational support at RS system level, such as data management

The arm is controlled by ground-prepared command lists under supervision from cosmonauts in EVA or IVA via the EVA Man-Machine Interface (EMMI) or the IVA MMI (IMMI) (see figure 2).

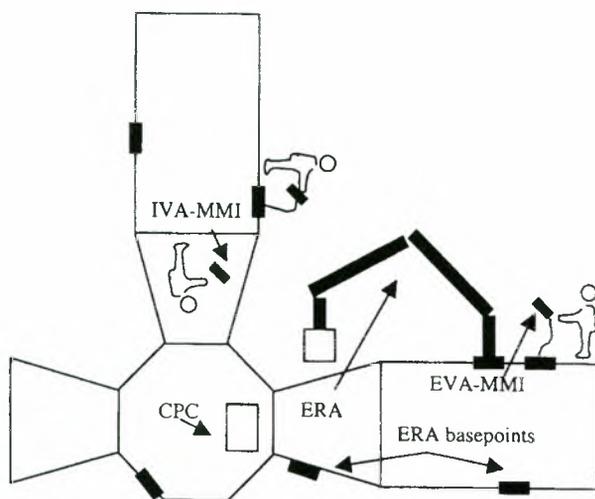


Figure 2: ERA flight segment configuration

The MPTE is under development with the National Aerospace Laboratory NLR (The Netherlands), and the Spacebel Trasys Association (Belgium).

The MPTE provides the ground support functions for ERA operations, both off-line and on-line (Ref. 2):

- Mission Preparation
- Training of ERA operators
- On-line Mission Support
- ERA Mission evaluation.

To support mission validation and training and to manage the MPTE as a stand-alone system, the MPTE also includes simulation, visualisation, and facility management.

In addition, the MPTE supports flight and ground operational software maintenance.

The MPTE will be installed at three locations:

- RSC/E-MCC, Korolev, Moscow Region, Russia, to be used for support of flight operations (mission preparation and mission support).
- GCTC, Star City, Moscow Region, Russia, to be used for training of ERA operations.
- ESTEC Noordwijk, The Netherlands, to be used for training of Russian instructors, and for maintenance of flight and ground operational software.

2. OPERATIONS

ERA Operations

In nominal situations, ERA missions will be prepared and validated on the ground, using the MPTE. An ERA mission is defined as a complete end-to-end sequence of ERA operations, between one hibernation and another. The MPTE has to provide the data-sets ready for up-linking to the Russian Segment. For this purpose, the MPTE is used for composing an ERA Operations Plan (EOP), based on a high level RS Mission Plan, which contains all information about the Russian Segment to plan a detailed ERA mission.

In general, the ERA Operation Plan contains (Ref. 3):

- ERA Actions, the lowest level element in the hierarchy of ERA operations
- ERA Tasks, a subset of the Auto Sequence composed of logical groups of individual commands or actions
- ERA Auto Sequences, part of the ERA Operations Plan which constitutes the sequence of commands which will be executed automatically by the ECC
- ERA Uplinkable Command Lists (EUCL) (ERA data-sets), consisting of one or more auto sequences, where ERA commands are the implementation of planned actions to control one of ERA's subsystems

The EUCL will be derived from the EOP. Next to the EUCL, other data-sets can be prepared to enable updating of onboard databases, upgrading of onboard software, dumping of memory and data.

The high-level chronological order of MPTE activities in supporting preparation, training, on-line mission support and evaluation of an ERA mission is assumed to be as follows.

Facility management

- The MPTE will be configured compliant with the onboard configuration
- Data and files, needed for preparation and support of the mission are transferred from the RS or other MPTE's.

Mission preparation

- Based on the RS mission plan, which includes the scope of planned ERA activities and system configuration identifiers, an ERA mission is prepared for flight, training, or software verification purposes.
- In case the Space Station configuration has changed, a new geometry model will be used. This model will be processed in such a way that it can be used for planning of detailed paths, for visualisation, and for preparation of onboard (geometry-related) data.
- The data will be used to define a corresponding sequence of ERA tasks (e.g. Attach) and ERA actions (e.g. insert, grapple), with the associated ERA task and ERA action attributes, and related tasks for cosmonauts in EVA and IVA, together constituting the ERA Operations Plan (EOP).
- From this ERA Operations Plan, data will be selected and converted to EUCL data-sets and other flight data-sets (EOP, code files, dump requests, and specific ECC database loads).
- The new ERA mission will be validated by loading of data-sets into the MPTE/ERA hardware-in-the-loop, and by simulation of the mission. Generation of simulated telemetry data will enable validation of the prepared MPTE on-line mission support configuration.

Operations training

- ERA operators and cosmonauts will be trained on generic tasks derived from ERA reference missions, using the MPTE training support facility.
- Specific ERA operations will be trained during mission-specific ERA training sessions.

Mission execution

- At the appropriate time, the prepared data-sets will be uplinked (RS responsibility), and the ERA will be controlled by the on-orbit operator (cosmonaut), supported from the ground segment.
- ERA operations are monitored, using the MPTE on-line mission support function. Flight data as well as ground configuration data will be archived for post mission analysis. In case of non-nominal behaviour, contingency actions will be initiated.
- Post-flight, the ERA operations and performance will be evaluated using the MPTE mission evaluation function, and it will be possible to replay on-line mission support.

Maintenance operations

Furthermore, the MPTE installed at ESTEC will include the Software Development Environment (SDE) for the maintenance of the ERA Ground and Flight Operational Software. The updated Flight and Ground Operational Software will be transferred to the other MPTE's.

On-line help

The MPTE is designed as a stand-alone operating facility with tools to support operations and training. Operational user support is given with on-line help functions and instructions.

3. FUNCTIONAL DESIGN DESCRIPTION

The MPTE design is based on:

- MPTE delivery: three stand-alone identical facilities
- MPTE operations, operational interfaces and dedicated hardware interfaces
- Design constraints
- Maintenance and configuration management
- System specification and functional breakdown

To get three almost identical facilities, the MPTE design is based on building blocks, each representing main MPTE functions. Ground Operations have been taken into account as main building blocks. Operational and hardware interfaces are mainly determined by external interfaces to the RS. A distinction can be made between on-line interfaces, such as Telemetry (TM), and off-line interfaces, such as the RS-Mission Plan and a copy of the geometrical configuration of the Space Station.

Design constraints are:

- ERA development standards, and requirements for availability and maintainability
- Re-use of existing software:
 - ◆ European real-time operations Simulator (EuroSim) software platform (Ref. 4) together with the Image Generation System (IGS)
 - ◆ Columbus Ground Software (CGS) (Ref. 5)
 - ◆ ERA dedicated software, such as the ERA Simulation Facility (ESF) software, and the ERA Flight Software Maintenance Facility (SMF)
- ERA mission constraints, such as cosmonauts, payloads, memory, operational conditions, etc.
- Simultaneous use of MPTE functions, such as mission preparation and on-line mission support.

The MPTE has to be designed for 10 years operational lifetime. This is rather difficult to realise, taking into account the re-use of existing software, hardware and ERA dedicated developments. A provisional maintainability and availability analysis resulted in the identification of some spare units and a provisional approach for maintenance of the system. Final conclusions are still to be taken.

Another challenge for the MPTE design is the requirement for configuration control of operational software and data over three facilities. Next to the design approach of having three identical facilities, the design is based of having a master database at one of the facilities for each MPTE software and data segment.

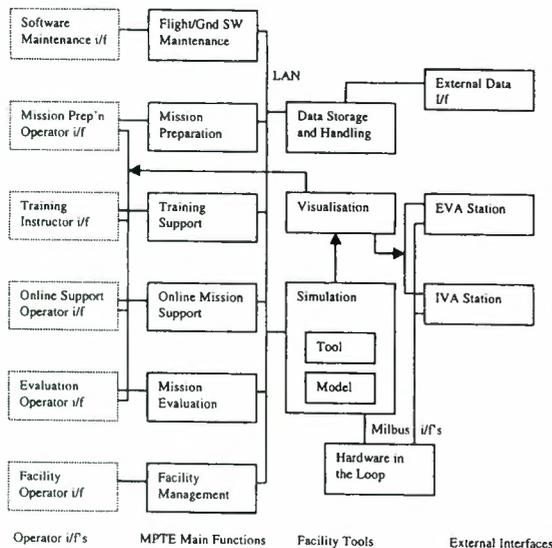


Figure 3.: MPTE Functional Breakdown

For the MPTE main functions the following design and development concept is followed (See figure 3):

ERA Mission Preparation function:

The model of the Space Station geometry (including payload and ERA) is processed for the purpose of ERA path planning, visualisation and preparation of onboard data by using the software packages ROBCAD and Multigen2.

The preparation of the EOP is implemented by MPTE specific software for both ERA missions and training missions, including a preparation function for a generic mission database, a specification function of the onboard ECC database parameters, and a verification function for the planning results. The Mission Database consists of a combination of file and data oriented storage of basic operational elements. Oracle is used as the data oriented relational database for ERA missions, tasks, actions, and commands. For planning of the EOP, a user interface supports the handling of generic operational elements. For detailed planning of ERA paths, use is made of the ERA Path-Planning module, using the ROBCAD software package.

For validation of the mission, dedicated MPTE simulators are developed, implemented on the EUROSIM/IGS simulation/visualisation software platform, with real ERA hardware in the loop (HIL). For monitoring and evaluation of the mission, simulated Telemetry (TM) data is generated and processed by the On-line Mission Support function, and later on by the Mission Evaluation function.

ERA Training Support function:

A number of MPTE simulators have been designed on the EUROSIM platform, making reuse of ESF model software, and the IGS. Different images will be generated dependent on the required views for the training configuration (EVA training, IVA training, combined EVA/IVA training).

ERA On-line Mission Support function:

The Online Mission Support function is used to monitor ERA missions and to collect and store ERA data for post-mission evaluation. Monitoring is based on ERA joint temperatures, angles etc extracted from ERA telemetry (TM) data received from the Russian Segment. The Mission Support synoptic displays show calibrated engineering values, memory dumps as well as ERA pose (the ERA pose display is identical to the display used by the on-board IVA operator).

Online Mission Support is based on CGS (Ref. 5):

- Mission Data Base (MDB) is used to maintain a definition of all ERA parameters
- Test Execution Software is used to receive and process TM (Calibration, limit monitoring, event management etc)
- Human/Computer Interface (HCI) is used for the synoptic displays
- Test Result Data Base (TRDB) is used to store all data received
- Test Evaluation Software (TEV) to perform post-mission evaluation.

ERA Mission Evaluation function:

The Mission Evaluation function can be used to perform trend analysis and raw data dumps, list events and engineering values as well as replay mission data using synoptic displays or high quality visual effects.

High quality visualisation is based on EuroSim/IGS whereas trend analysis etc is based on CGS. The fairly extensive in-built CGS evaluation functions are further extended by interfaces to:

- Microsoft Excel
- PVWAVE
- Special Application Software.

MPTE Facility Management Support function:

The MPTE facility management operator will use available tools and functions (e.g. Unix, CGS, FTP tools) to perform the tasks. The MPTE provides an MPTE dedicated installation of these tools.

ERA Flight Operational Software Maintenance function:

The design of this function is based on the integration of existing Software Development Environments (SDE) and Electrical Ground Support Equipment (EGSE) for the OBS parts. For the MPTE, the internal interfaces

between this maintenance facility and other MPTE functions are the design drivers.

ERA Ground Operational Software Maintenance function:

This function will be implemented by re-using MPTE development tools.

4. SYSTEM DESCRIPTION

The MPTE system consists of a computer system interconnected by a local area network, some specific hardware to support dedicated simulation and training functions and software. For the software, a distinction is made between COTS software, ERA dedicated software, and MPTE dedicated software. In figure 4 the hardware configuration is presented.

MPTE hardware architecture

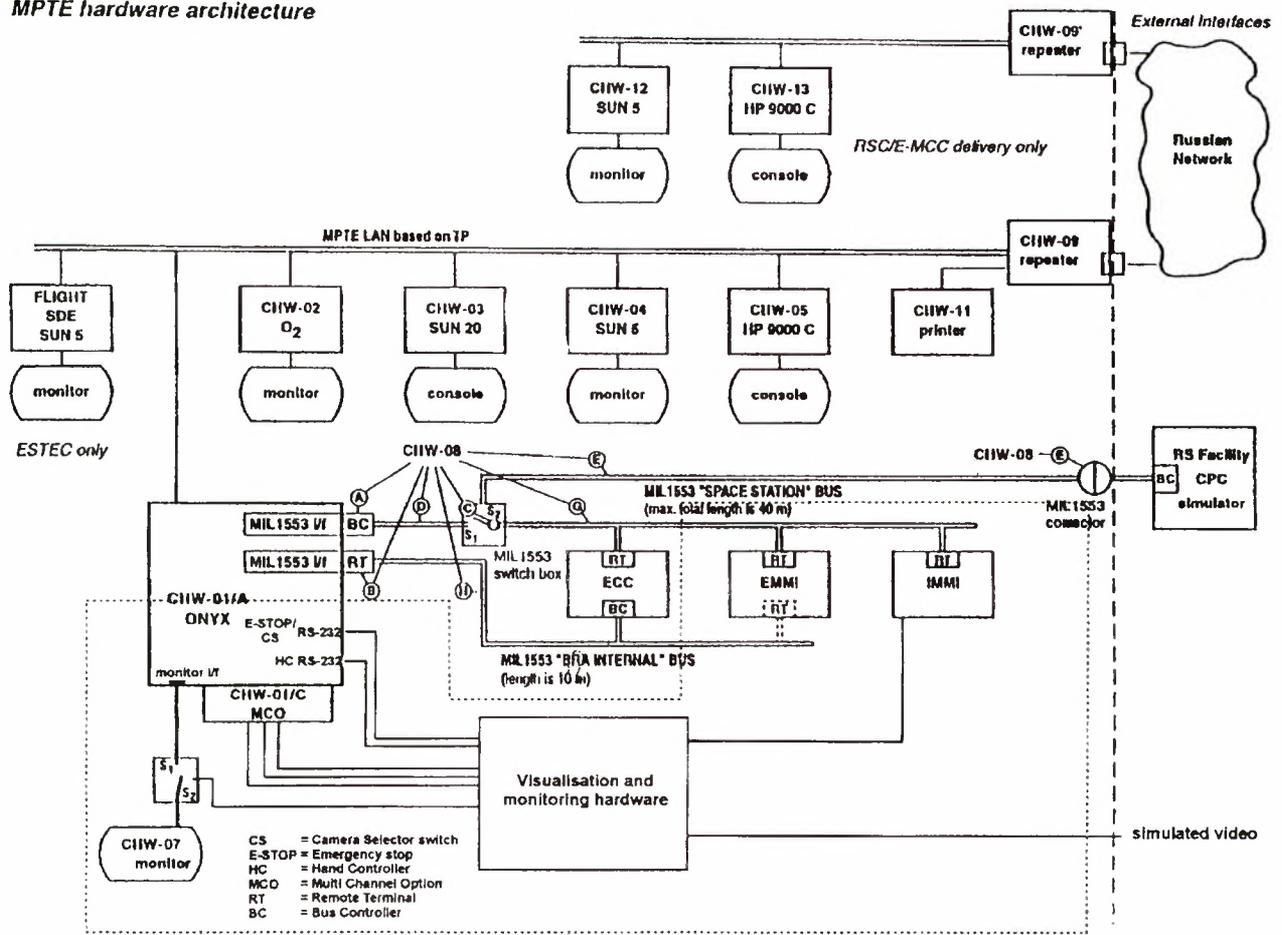


Figure 4: MPTE hardware architecture, needed for all required functions.

5. DEVELOPMENT STATUS

Presently, the design and development (including coding) of the MPTE is in the detailed design phase for the Pre-flight delivery. Although specification and architectural design phases have been closed, the interfaces to the ERA system are still changing, due to parallel development of the ERA system and its interfaces. Concurrent design and development of the MPTE is one of the major challenges of this project.

The pre-flight configuration will be based on the need for basic functions in preparation and training of ERA

operations. Very important is the use of external interfaces to the RS, the stand-alone operations of the facilities in Russia, and the training of ground operators, being the mission preparation operator, the instructor, and the mission support operators.

Testing of the MPTE is according to the ESA software Engineering standards. Unit testing has already been started on the level of software and hardware configuration items. Since the MPTE is designed for operations in stand-alone mode, an incremental integration of MPTE units is planned, starting from

'importing of data from RS' until 'extracting data ready for uplinking'.

System testing and provisional acceptance of the pre-flight configuration is planned for September/October 1999. The final configuration is planned for delivery mid 2000.

6. FUTURE EXTENSIONS

The ERA pre-flight configuration has limited operational capabilities. For instance, onboard collision detection (implemented by a dedicated Collision Avoidance database) has not been implemented. In addition, the software has not all functions to support all types of operations. The ERA final configuration will include all operational capabilities as required for the operations planned to be performed onboard the RS part of Space Station.

Extensions of operational capabilities for ERA are not planned yet. However, the ERA design is such that additional functions can be implemented by onboard software upgrades. Amongst others, it could be operated from the ground. Other operational extensions might be found in extension of onboard tools and equipment, such as ERA tools, lighting system, dextrous gripper, smart sensors (Ref. 6). For most of the extensions, extra software on both flight segment and ground segment will be required.

There are some potential extensions. The ERA and the MPTE are built for 10 years operational life time, but for reliability reasons and use of state-of-the-art technology, baseline functions may require upgrading of subsystems or components, such as the visualisation system (e.g. Ref. 7), mission preparation, automated prediction of onboard maintenance.

7. DISCUSSION AND CONCLUSION

The description of the MPTE functions and the MPTE design depicts a multi-purpose system for all ground operations needed to support ERA flight operations. The combination of all ground support operations, the re-use of existing software, and the delivery of three identical operational systems, has been found a major challenge in developing the system. In addition, concurrent engineering and development with the ERA system itself is found to be greater challenge even. By its functional design and building blocks, it was possible to adapt changes in interfaces and capabilities to a certain extent.

The flexibility in support of planning ERA operations is based on the operational elements of ERA. Future extensions can be easily implemented.

Maintenance and operational support of the MPTE during ERA lifetime is one of the critical aspects still to be analysed. A significant update of the systems might be the final conclusion of this analysis.

8. REFERENCES

1. Kampen, S., Mandersloot, W., Thirkettle, A., Bentall, R.H. The European Robotic Arm and its role as part of the Russian Segment of the International Space Station Alpha. IAF-95-T.3.03, 46th IAF Congress, Oct. 1995, Oslo, Norway.
2. Schoonmade, M., Mission Preparation & Training Equipment (MPTE) for the European Robotic Arm (ERA). ISU 1998 Alumni Conference, July 1998, Cleveland, USA
3. Aris, L. et.al. ERA Flight Operations Manual and Procedures. Issue 3, Rev. 1, Dec. 1998, Fokker Space
4. EuroSim Mk1 Software User Manual, Fokker Space BV, NLR-EFO-SUM-2, Issue 2.5, May '97
5. Columbus Ground Software (CGS). Assembly Requirements Specification, SPE 1214597
6. Heemskerk, C., Mandersloot, W., Steinmeyer, R., Putz, P. Innovative Technologies in the ERA Evolution Programme. IAF-98-U.1.07, 49th IAF Congress, Sept. 1998, Melbourne, Australia
7. Bouchon, P., Gautier, C., Heemskerk, C.J.M. The Use of VR in Cosmonaut Training for Robotic Cooperation. ESA WPP-150, SESP '98, November 1998, Noordwijk, The Netherlands

A Universal Task-Level Ground Control and Programming System for Space Robot Applications

The MARCO Concept and its Application to the ETS-VII Project

B. Brunner, K. Landzettel, G. Schreiber, B.M. Steinmetz, G. Hirzinger

DLR Oberpfaffenhofen
German Aerospace Center
Institute of Robotics and System Dynamics
D-82234 Wessling
Bernhard.Brunner@dlr.de

Abstract

The paper outlines the main features of DLR's ground control station for space robotics applications. It combines sensor-based task-level *teleprogramming* (as the basis for autonomy) with the features of *teleoperation* and shared autonomy. The hierarchical system structure is shown as well as the flexibility in programming and controlling each kind of space robotics application. The teaching by showing approach is the key to a easy-to-use programming interface at different levels of space robot controlling. This approach has led to a modular task-directed programming scheme, called *Modular A&R Controller (MARCO)*, which provides a very flexible architecture to adapt the application-specific requirements to a given controlling scheme. To demonstrate the power of MARCO, we describe the results of the *GETEX* experiment, which has been performed in April '99 at the first free-floating space robot on NASDA's ETS-VII satellite¹.

Introduction and Overview

After the success of ROTEX, the first remotely controlled robot in space, DLR has focused its work in telerobotics on the design of a high-level task-oriented robot programming system, which is characterized as *learning by showing in a virtual environment*. The goal was to develop a unified concept for a flexible, highly interactive, on-line programmable teleoperation ground station as well as an off-line programming system, which includes all the sensor-based control features already tested in ROTEX², but in addition provides the feasibility to program a robot system at an implicit, task-directed level, including a high degree of on-board autonomy.

This means that a non-specialist user like a payload expert will be able to control a remote robot system e.g. for internal servicing within a space station, i.e. in a well-known environment. This requires a sophisticated man-machine-interface, which hides the robot control details and provides an *intuitive programming interface*. For that reason, we have developed a network-transparent graphical user interface, based on the quasi-standards VRML and Java. Using a task-level protocol is the preferable method to remotely operate robots as it demands only extreme narrowband connections and does not bother about large time delays. The user interacts via the virtual view with the real environment, as (s)he has only to define, what (s)he wants to do, not how it has to be done.

Supported operations are e.g. open/close a door/drawer, pick&place an orbital replaceable unit etc.

However, for external servicing with free-flying robots, e.g. the repair of a defect satellite, high interactivity between man and machine is required, because the remote environment will be mainly unknown. All the well-known problems w.r.t. teleoperation under long time delays can only be solved by the *predictive graphics* approach. One of the main requirements is the feasibility to update the simulated world according to the real world as well as to provide *local autonomy based on intelligent sensor data processing* without large a priori knowledge.

To fulfill the requirements of both application fields, we have developed a 2in2-layer model³, which represents the programming and control structure from the executive to the planning level in a hierarchical way. According to the application requirements the user can use the necessary and sufficient level of commanding and programming or switch between the *different layers* especially in case of failure detection and recovery.

This control and programming system may be used for several applications: the task-oriented non-expert programming layer is demonstrated by the implementation of a net-browser VRML plugin⁴ to control a prototypic intravehicular environment, an extension of the ROTEX workcell, at the task level without any knowledge of robotics.

It seems straight-forward kind of work to make the MARCO programming and control environment applicable for the Technology Exposure Facility (EuTEF) of the International Space Station.

As a realistic test, the ground control facilities of our system were used in April '99 to remotely control the Japanese ETS-7 robot, the first robot in free space. The main goals of DLR's contribution within the GETEX project were the utilisation of the world model update concept using the real video images, to verify our task-level programming approach including on-board autonomy via selected image features and force-torque information as well as the verification of the dynamic simulation due to the interactions between robot and carrier.

Our cooperation with NASDA w.r.t. to the dynamics verification was one important step towards a *free-flying service satellite*. For more details see ⁵. In our lab the semi-autonomous telemanipulation feature of the ground control and programming system is used for the ESS (experimental servicing satellite) scenario, where a free-flying telerobot is supposed to approach, inspect und repair a malfunctioning satellite, e.g. the TV-Sat-1, where after launch one solar panel

had not opened. A special, in-house-developed capture tool, containing 6 laser range finders, a wrist-mounted force-torque sensor and stereo camera, allows, in combination with the dynamics behavior prediction, the fully autonomous servoing, insertion and capturing of apogee motors, which are typical for any geostationary satellite.

Furtheron, a *robonaut* system is proposed which can take on or share intravehicular payload activities, so far carried out by astronauts. Due to the fact that the payloads are optimized for human operation, the robot endeffector must be able to interact with this human-adapted environment. We have equipped our 7-axes light-weight-robot with a human-like 4-finger-hand to handle devices, which are standard in a human environment, and with a 3-axes gantry to reach all positions within an experimental spacelab setup. The control and programming system as used for the above applications is flexible enough for usage in this multi-degrees-of-freedom system. In extension to the former application a data glove is used for teleoperating and programming human-like grasp and manipulation actions.

It should be mentioned that our programming system is immediately applicable to *planetary rovers* as well as to terrestrial service robotics: instead of the gantry a mobile platform is used to implement a „butler“ robot, which will be able to perform helpful tasks in an ordinary environment.

The MARCO system

The goal for the development of our high-level programming system was to design a *unified concept* for a flexible, highly interactive, on-line programmable teleoperation station as well as an off-line programming tool, which includes all the sensor-based control features as tested already in ROTEX, but in addition provides the possibility to program a robot system on an implicit, task-directed level.

A non-specialist user – e.g. a payload expert – should be able to remotely control the robot system in case of internal servicing in a space station (i.e. in a well-known environment). However, for external servicing (e.g. the repair of a defect satellite) high interactivity between man and machine is demanded. For that reason the design of our programming system is based on a 2in2-layer-concept, which represents the *hierarchical control* structure from the planning to the executive layer:

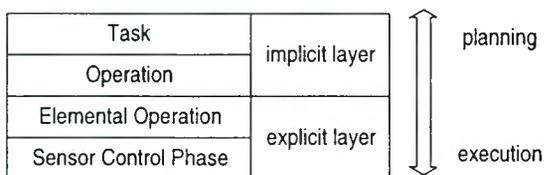


Figure 1 2in2-layer-model

On the implicit level the instruction set is reduced to **what** has to be done. No specific robot actions will be considered at this task-oriented level. On the other hand the robot system has to know **how** the task can be successfully executed, which is described in the explicit layers.

Reflex (Sensor Control Phase)

At the lowest level of the MARCO system the sensor control mechanism is active. These so-called reflexes guarantee the *local autonomy* at the remote robot's site via using sensory data processing algorithms in an extensive way. The teaching by showing paradigm is used at this layer to show the

reference situation, which the robot should reach, from the sensor's view: in the virtual environment we store the nominal sensory patterns and generate appropriate reactions (of robot movements) on deviations in the sensor space.

A reflex is described by

- A controller function, which maps the deviation in the sensor space into cartesian robot move commands
- A state recognition component, which detects the controller's end conditions (success, failure)
- The constraint frame information, which supports the controller function with the task frame data to interpret the sensory data correctly (e.g. for shared control)
- A sensor fusion algorithm, if sensor values of different types have to be transformed into a common reference system (e.g. vision and distance sensors).

Elemental Operations

The explicit programming layer is completed by the Elemental Operation (*ElemOp*) level. It integrates the sensor control facilities with position and endeffector control. According to the constraint frame concept, the non-sensor-controlled degrees of freedom (dof) of the cartesian space will be position controlled

- in case of *teleoperation* directly with a telecommand device like the SpaceMouse.
- in case of *off-line programming* by deriving the position commands from the selected task. Each object, which can be handled, includes a relative approach position, determined off-line by moving the virtual end-effector in the simulation into the desired pose w.r.t. the respective object and storing the geometrical relationship between the object's reference frame and the tool center point.

It should be mentioned that the ElemOp layer aims at a manipulator-independent programming style: if the position and sensor control function are restricted to the cartesian level, kinematical restrictions of the used manipulator system can be neglected. This implies the general reusability of so-defined ElemOps in case of changing the robot type or modifying the workcell.

A model-based on-line collision detection supervises all the robot activities. For global transfer motions a computational very fast *path planning* algorithm⁶ avoids collisions and singularities in the robot's joint space.

Operations

Whereas the Reflex and ElemOp levels require the robotics expert, the implicit, task-directed level provides a powerful man-machine-interface for the non-specialist user. We divide the implicit layer into the Operation and the Task level.

An Operation is characterized by a sequence of ElemOps, which hides the robot-dependent actions. Only for the specification of an Operation the robot expert is necessary, because (s)he is able to build the ElemOp sequence. For the user of an Operation the manipulator is fully *transparent*, i.e. not visible.

We categorize the Operation level into two classes:

- An Object-Operation is a sequence of ElemOps, which is related to a class of objects available within the workcell, e.g. GET <object>, OPEN <door>.
- A Place-Operation is related to an object, which has the function of a fixture for a handled object, e.g. INSERT <object> INTO <place>. Before an Place-Operation can be activated, the corresponding Object-Operation has to be executed. <object> is the object, known from the predecessor Object-Operation, <place> the current fixture, to which the object is related.

Each object in the workcell environment can be connected with an Object-Operation and/or an Place-Operation. Because an Operation is defined for a class of objects, the instantiation of formal parameters (e.g. the approach frame for the APPROACH-ElemOp) has been done during the connection of the Operation with the concrete object instance.

To apply the Operation level, the user only has to select the object/place, which (s)he wants to handle, and to start the Object-/Place-Operation. For that reason the programming interface is based on a virtual reality (VR) environment, which shows the workcell without the robot system. Via a 3D-interface (DataGlove or a 3D-cursor, driven by the SpaceMouse) an object can be grasped and moved to an appropriate place. If the user has moved all the objects to the places he want, the execution of the generated task can be started by doing a specific VR-hand gesture. For supervision the system shows the state of the Operation execution, i.e. the ElemOp, which is currently active. Also the position and orientation of the currently moved object is fed back.

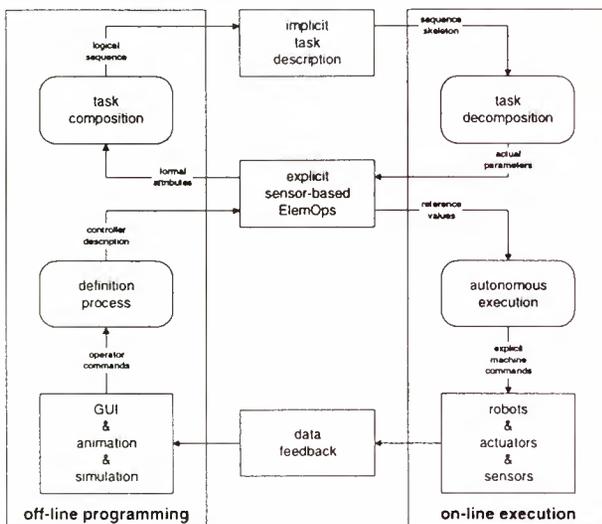


Figure 2 task-directed sensor-based programming

Tasks

Whereas the Operation level represents the subtask layer, the possibility to specify complete robot tasks must be available in a task-directed programming system. A Task is described by a consistent sequence of Operations, which are instantiated with concrete object instances (see Figure 2). To generate a Task, we use the VR-environment as described above. All the Operations, activated by selecting the desired objects or places, are recorded with the respective object or place description. An expressive example will be given in the GETEX-section.

Different graphical user interfaces

Our task-directed programming system with its VR-environment provides a man-machine-interface at a very high level, i.e. without any detailed system knowledge, especially w.r.t. the implicit layers.

To edit all four levels as well as to apply the Reflex and ElemOp level for teleoperation, a sophisticated graphical user interface based on the OSF/Motif standard has been developed (see Figure 3, screen down on the left). This GUI makes it possible to switch between the different execution levels in an easy way.

Based on the ROTEX experience we have implemented a prototypic teleoperation station, to remotely control space robotics applications by predictive graphics. Figure 3 shows different views of the simulated environment (far, near, camera view), the Motif-GUI, and the real video feedback image, superimposed with a wireframe model of the predicted state (up on the right). All the screens can be viewed in stereo mode for full immersion into the workcell environment.

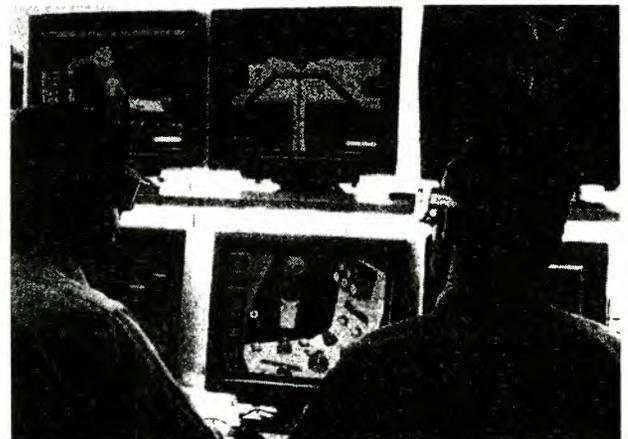


Figure 3 GUI of the universal programming and control station (MARCO)

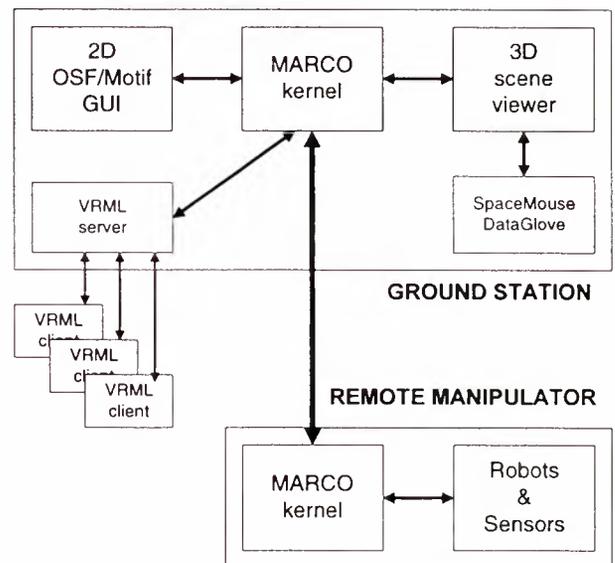


Figure 4 User Interface Structure

Java/VRML client interface

New chances towards standardization in teleprogramming arise with Java and VRML. This combination makes it possible to build easy-to-use and very cheap telerobotic stations, especially for payload users, which are not robotics experts. Via the simple Pick&Place semantics as described above, tasks can be composed and forced to a server, which will execute the desired actions. The user only clicks onto the objects, which (s)he want to handle, and starts the execution. This server also allows the cooperative work at the same environment: only one client is able to generate and start a desired task, all the other clients can view the current workcell state. After finishing the execution control is switched to the first

client, which sends the appropriate „I will do it“-command to the server.

In fact, the MARCO system acts as the server, so that the implicit layers of our telerobotic station system are fully programmable via a simple Java/VRML client, e.g. the ROTEX environment at our lab (see Figure 5).



Figure 5 Java/VRML client

General scene viewer

Besides the Motif-based GUI, which provides the 2D-interface to change parameters and compose ElemOps etc. a powerful 3D scene viewer is connected to the MARCO system, which exploits the sophisticated graphics hardware to enable real time rendering and simulation of different camera aspects. This is achieved with *IRIS Performer*, but also *Open Inventor* as 3D graphics library is utilized, especially for porting the application to different hardware like PC's.

Texture mapping and highly detailed geometries are as well supported, as all different kinds of graphic devices, like SpaceMouse, dataglove, cave-like stereo projection and head-mounted displays. It is clear, that the Viewer is open for further extensions, like new devices or different scenarios.

The GETEX experiment on ETS-VII

From April 19-21, 1999 the DLR's MARCO tele-robotic and -programming system was used to control the robot arm on the Japanese ETS-VII satellite. The main goals of this German Technology EXperiment on ETS-VII (GETEX) were

- to verify a MARCO-based telerobotic *ground control* station for remote control of a free-floating robot, in particular
- to perform a peg-in-hole experiment, using VR methods and the „*vision&force*“ control scheme, by closing sensor control loops directly on-board (force) and via the

ground track (vision), thus proving MARCO's sensor-based autonomy features,

- to conduct experiments with relevance to the *dynamic behavior* of ETS-VII in free motion mode and thus to verify the existing dynamic models.

All experiments could be performed very successfully. To implement the User Interface Structure as depicted in Figure 4, we had to add some modules for communication with the Japanese ground control system, but not to change the overall MARCO ground control structure.

To check and test our interfaces as well as our MARCO control station within the ETS-VII scenario, an on-line simulator has been developed, which emulates the remotely operated robot, its command interfaces and its environment. The simulator is able to emulate all different modes, timing, the environmental interactions, and the prediction of satellite attitude while moving the robot arm. This kind of simulation has turned up to be very useful for proving software correctness while interacting with the telerobot.

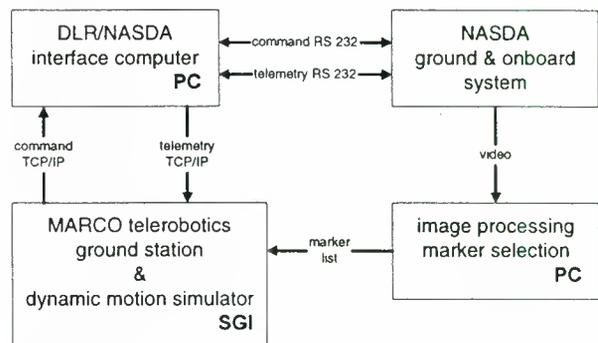


Figure 6 GETEX ground control configuration

The original MARCO kernel couldn't be implemented on-board the ETS-VII, because only the ElemOp-Layer was available on-board. All the other layers were implemented as add-ons on-ground, but this was no limitation to the verification of our task-level programming methods, because the downlink feedback data were rich enough to parametrize the next ElemOp according to the current execution state.

It should be mentioned, that the know-how, gained during the phase of adapting the MARCO system to the ETS-VII constraints, will be very useful for further space robot missions.

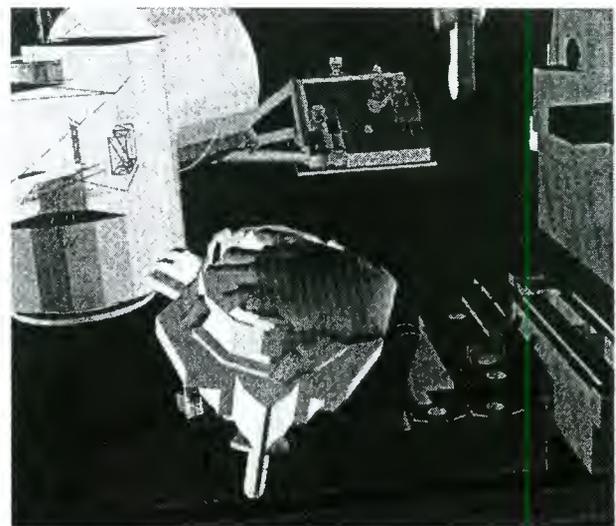


Figure 7 Pick TBTL by VR-hand

The MARCO system worked that well, that we decided, together with the Japanese partners, to execute the whole peg-in-hole experiment with the TBTL (TaskBoard Tool) in the automatic mode: after teach-in of the desired task sequence (pick TBTL, see Figure 7, and place it to HOLE A, see Figure 8) in the VR environment, the execution was started and performed fully automatically. No voice confirmation between each ElemOp was further needed, as it had to be done during the test runs.

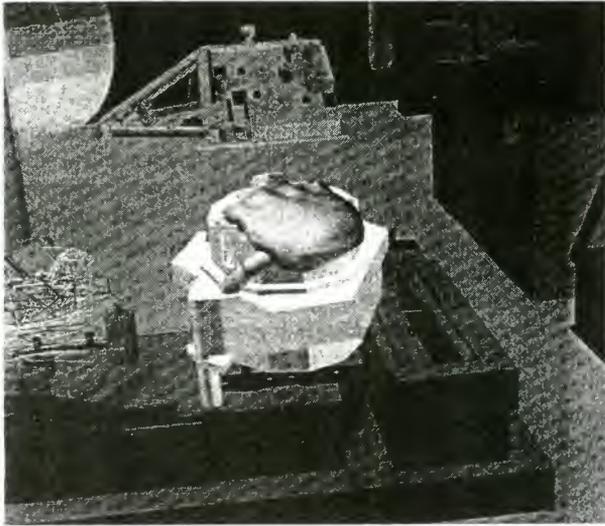


Figure 8 Place TBTL into Hole A

The real robot and object status (here the TBTL), fed back in the telemetry channel, is shown wireframed.

To get the TBTL, we first carried out a visual servoing task (at the reflex layer), which uses some marker features in the video image to control the tool center point (TCP) of the robot autonomously into the desired sensor-related pose. For that reason we have developed an approach, which doesn't need any calibration. The control law may be written as

$$v_c = \alpha C (s-s^*)$$

where $(s-s^*)$ is the vector-valued deviation between the current and the nominal sensory pattern indicating the displacement of the current robot pose x from the nominal pose x^* . v_c is the velocity command, α represents a scalar dynamic expression, at least a real constant, determining the closed control loop behavior, and C represents a projection operator used for mapping the sensor space onto the (cartesian) control space. C is determined by neural network learning or using analytical methods.

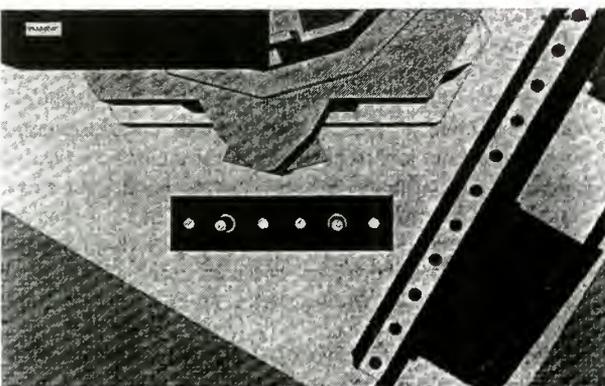


Figure 9 View out of the hand camera, showing the tracking markers for visual servoing

Here we have applied the analytical method for determination of C , which is represented by the Pseudoinverse of the Jacobian matrix of the m deviations in the sensor space w.r.t. the n deviations in the control space. For that we moved the robot's TCP a little bit around in all $n=6$ degree of freedoms, recorded the corresponding sensor values and generated the Jacobian from the resulting difference quotients.

$$J_{i,j} = \left. \frac{\partial y_i}{\partial x_j} \right|_x, \quad i = 1..m, j = 1..n$$

We performed the experimental determination of C in our simulation environment as well as in the real one. The result was nearly the same, due to the accuracy of our camera simulation. The camera parameters have been estimated using an in-house developed camera calibration tool.

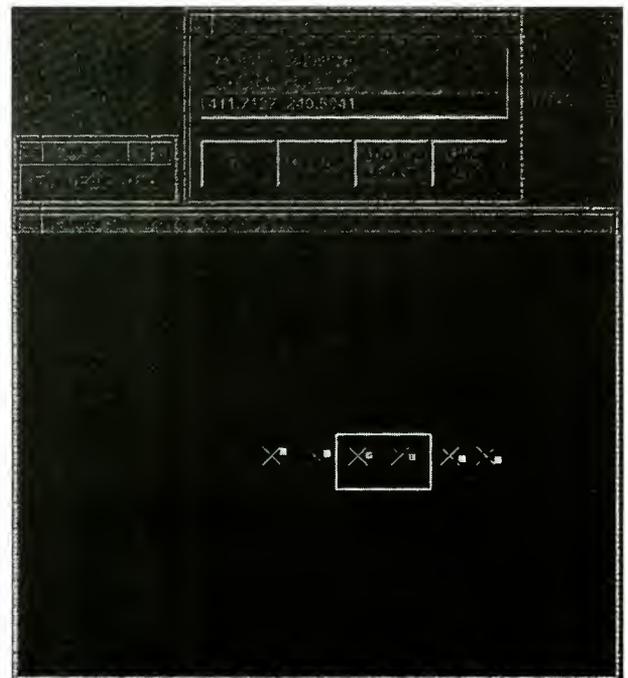


Figure 10 marker selection from real video image

For control we used the TaskBoard marker features, which were originally available to teleoperate the TCP into the right position. The goal was to find the markers in the live video image and to generate the appropriate straight path command to move the robot into the desired (sensor-defined) target pose. To verify the vision-based sensor control loop, we moved the TCP intentionally into a position different from the target pose (a few centimeters in all translational directions and about 20 degrees in z-rotation).

After 3 cycles (with $\alpha = 1$), the target pose was reached. To extract the markers from the video image we used a blob-finding algorithm supported by the MIL (Matrox Image Library) functionality. Because this algorithm delivered more „markers“ as desired, e.g. due to bad lighting conditions, we selected the markers interactively and checked the resulting control command before sending it to the real robot. Figure 10 shows the simulated (■) and the real (X) markers, with the interactive selection frame.

The differences between the ■ and X markers in Figure 10 result from a different TCP pose, to show the two representations. If real and simulated TCP are the in the same pose, the real and simulated markers have nearly the same 2D-coordinates.

A major part of the GETEX experiment time was allocated to the so-called *Dynamic Motion* experiments, which consisted of a series of manoeuvres carried out by the manipulator while the attitude control system of ETS-VII was switched off. In such a mode of operation, a space robot consisting of a manipulator and a satellite is generally considered to be free of external forces^{7, 8}. The robot therefore is assumed to have constant angular momentum, due to the law of the conservation of angular momentum, which means that if the arm moves and thus introduces angular momentum into the system, the satellite reacts with a compensating motion. The amount of satellite rotation produced depends on the mass and inertia of the bodies which constitute the system. The description of a TCP trajectory in orbit-fixed coordinates, as it is necessary e.g. for the capturing of a defect satellite, has to account for the satellite reaction. For more details see⁵.

The experiments conducted during the GETEX mission aimed at a verification of the existing models of free-floating space robots and at the identification of the dynamic model parameters such as the satellite inertia tensor. A further goal was to obtain some insight into the nature and importance of disturbances acting on a robotic satellite in low Earth orbit and to gather data for the future design of controllers which will combine the manipulator motion control with the satellite attitude control. Therefore, a variety of different manoeuvres were executed (an example of which is shown in Figure 11), which include simple point-to-point operations and closed-loop re-orientation manoeuvres, sequences during which only one joint was active at a time as well as sequences during which all joints were moving simultaneously.

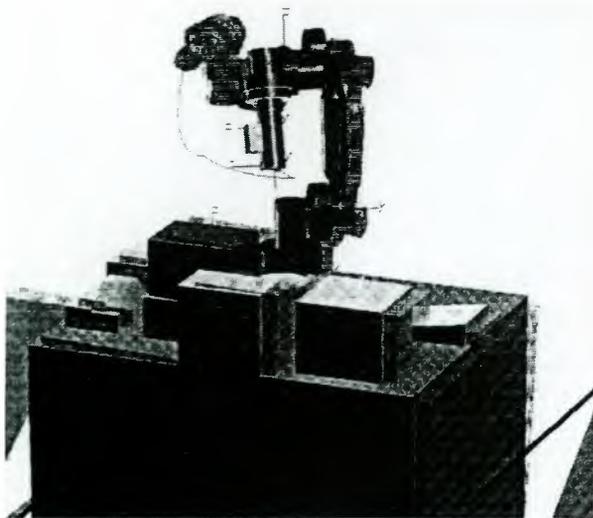


Figure 11 Example of a *Dynamic Motion* manoeuvre carried out during the GETEX mission.

The shaded robot indicates the reference position. The satellite reaction to the arm motion is scaled by a factor of 10 in this picture.

The major constraints, due to mission security aspects, were the maximum satellite attitude error allowed by NASDA which was limited to $\pm 1.0^\circ$ around each axis and the fact that the maximum tool center point velocity was limited, too. Furthermore, the reaction wheels were turning at a very low but non-zero constant velocity during the experiments, which introduced undesired torques into the system. Their effects will have to be considered during the evaluation of the mission results.

In total, over 110 minutes of dynamic motion experiments have been carried out, of which 52 minutes have been spent in

free motion mode. The remaining time was used to repeat the experiments in reaction wheel attitude control mode for verification purposes. First evaluations of the measurement data confirm the need to account for external disturbance forces acting on the satellite, such as the gravity gradient torque and magnetic torque.

The ESS scenario

For DLR the participation in the Japanese ETS-VII experiment was the first step to a very big challenge in space robotics: the capturing and repair of a failed satellite, completely controlled remotely from earth.

The technology study on the experimental servicing satellite (ESS)⁹ applies robotics to solve the problem of servicing a non-cooperative target in or near to a geostationary orbit, a region of space still out of reach to manned spaceflight. A three-month demonstration flight of ESS has been planned and all phases of its mission have been defined. These include the acquisition, inspection and servicing of an orbiting satellite through to parking it in a *graveyard*-orbit.

For that external servicing task high interactivity between man and machine is required, because the remote environment will be mainly unknown. The MARCO system¹⁰ will be used to give the system the local autonomy by intelligent sensor data processing. Because all the satellites, built so far, are not equipped for servicing, the final stages of approach and the subsequent capture of the target are the most critical phases of the mission.

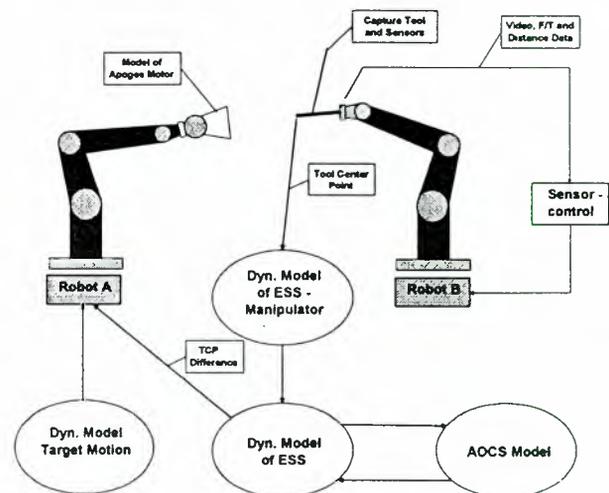


Figure 12 ESS simulation and testbed

The manipulator of ESS, equipped with a capturing tool, must follow the residual movements of a selected object on the target (e.g. the main thruster) by means of an image processing system whose data are passed through an extended Kalman filtering process. With the robot controller monitoring laser distance sensor values, force, torque and travel, the capture tool is inserted into the cone of the thruster. To simulate the dynamic behavior of the chaser during robot motions, we have arranged two KUKA robots as shown in Figure 12. Robot B is used to carry out the capturing task, Robot A emulates the entire dynamic relation between the chaser and the target satellite, where the dynamic coupling with the AOCS is included.

After capturing the target satellite, the ensemble is stabilised and reoriented. To free the manipulator for servicing activities and to provide a stiff mechanical coupling, the target satellite is grasped by means of a docking mechanism (grasping brackets in Figure 14).

To perform its servicing tasks, the robot replaces the capture tool with an appropriate servicing tool such as a scissor or a gripper. This requires that a tool adaptor, fitted with an integrated force and torque sensor and a stereo camera, is attached to the manipulator's endmost section. The tool exchange process is executed automatically, but control of the repair task itself must be shared between the machine and a human operator at the ground station. To counter the transmission time delay, a predictive graphical simulation of the robot's behaviour in its environment is used at the ground station.

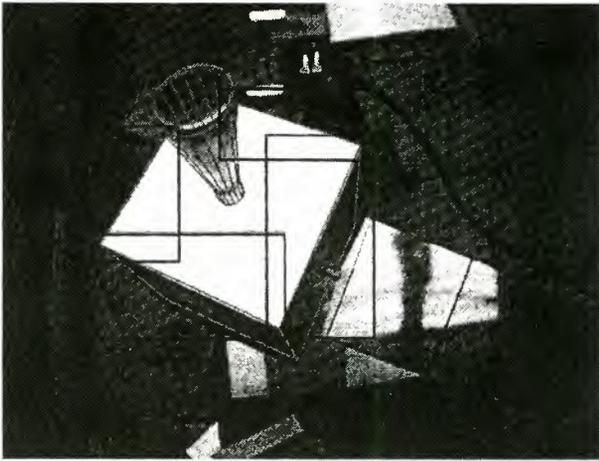


Figure 13 Tracking of target's apogee as seen from the wrist-mounted hand camera.

The wireframe model of the target is projected into the live video image at the currently estimated pose.

Although ESS is a highly complex automatic system, it is easy to maintain and its architecture is simple and extendable. This implies the use of modular hardware and software.

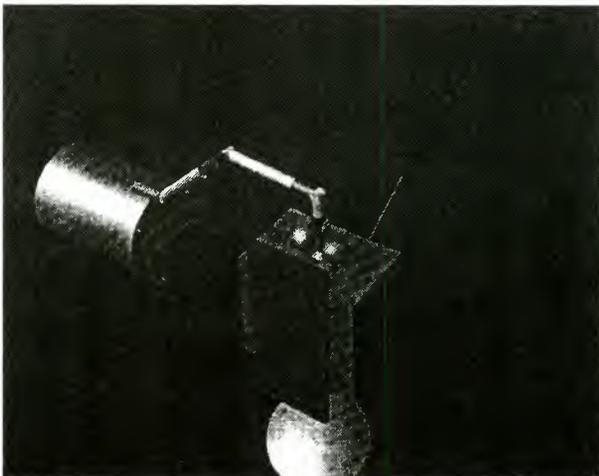


Figure 14 An artist's view of ESS, catching the apogee of TV-Sat-1

To curb costs, standardised elements are used wherever possible to realise the basic satellite functions. The satellite is now sufficiently defined to allow component procurement (in the next stage of the project) to proceed.

Technology Exposure Facility (EuTEF) at ISS

Recently we have performed extensive studies¹¹ of the European Technology Exposure Facility (EuTEF) to be robot operated at the truss structure of the International Space Station (ISS). Each payload module (PM) consists of the standard body structure (SBS) mounted on the standard grasping interface (SGI). It is placed on the express pallet (ExPA) base by means of the standard receptacle (SR). A payload module may be manipulated by the use of the basic end-effector (BEE) mounted on the robot's flange. Due to the fact that the TEF scenario will be very well-known and predictable, the implicit commanding levels can be used without any problems.

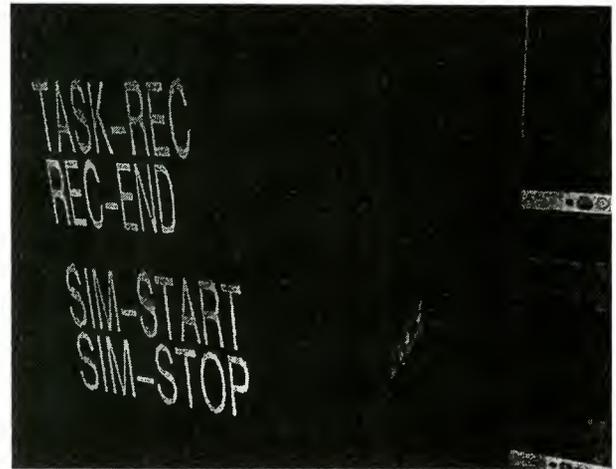


Figure 15 3D-cursor VR-interface with "function objects" and the BEE approaching a SBS



Figure 16 The BEE, approaching the SGI via visual servoing (4 markers around the middle hole)

We have the idea that the payload experts are sitting home in front of their PC's and command and supervise the TEF environment via a simple Java/VRML interface. Only in case of a failure the robot expert will take over the control and teleoperate the TEF robot into a save status or finish the desired task using the explicit MARCO levels.

We propose to apply the GETEX experience in *vision&force* control to the EuTEF scenario, e.g. to support the approach phase to a SGI by visual servoing: to align the BEE with the SGI's grasping position, the robot arm could track 4 colored markers in front of the SGI, which should be easily extracted by an image processing system, and then continue the grasping action with a well-known force/torque-control algorithm.

Multi-fingered Service Robotics

For dangerous and expensive extravehicular tasks as well as for intravehicular payload activities, which are optimized for human operation, we propose a robot system, which is able to interact with a human-adapted environment. We have equipped our 7-axes light-weight-robot with a dextrous human-like 4-finger-hand¹² to handle devices, which are standard in a human environment, and with a 3-axes gantry to reach all positions within an experimental spacelab setup (see Figure 17). The control and programming system as used for the above applications is flexible enough for usage in this multi-degrees-of-freedom system. In extension to the former application a data glove is used for teleoperating and programming human-like grasp and manipulation actions.

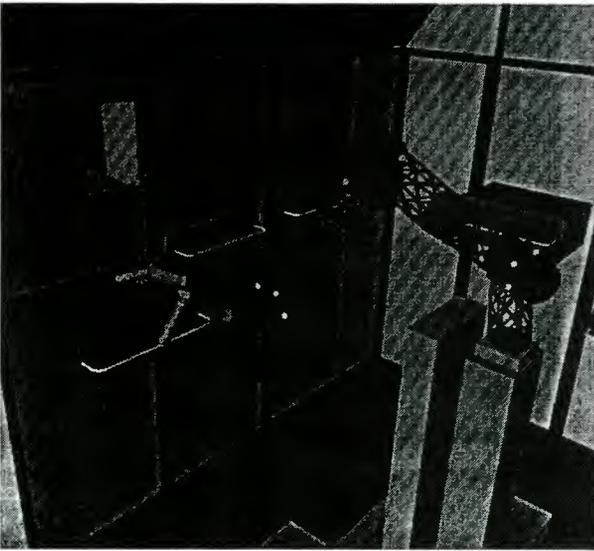


Figure 17 DLR's light-weight robot with 4-finger hand, mounted on a 3-axis-gantry

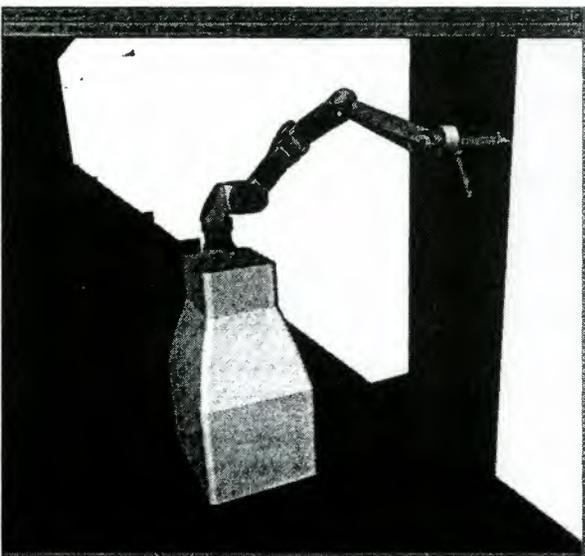


Figure 18 DLR's light-weight robot, mounted on a mobile platform for terrestrial service robot applications (e.g. opening a door)

In the spirit of the current system, an operator should not have to be a robot expert, this holds extremely for the acceptance of redundant manipulator systems which provide additional freedom. Here the future development will lead to a *task specific exploitation of Redundancy*. In the first step, we

showed, that the described system is able to handle this complex kind of kinematics. As future development, we will optimize the task specific exploitation of redundancy, so that the user will be able to easily control even such complex kinematics as 10 Axes of robotic manipulators plus additional 12 from the DLR Hand without being a robotics expert.

It should be mentioned that our programming system is immediately applicable to terrestrial service robotics: instead of the gantry a mobile platform is used to implement a „butler“ robot, which will be able to perform helpful tasks in an ordinary environment, e.g. get and bring a bottle of water from the refrigerator to a disabled person (see Figure 18).

Conclusion

We have shown the universal capabilities of DLR's MARCO system for controlling any kind of robotics applications, especially for space. Recently (April '99) we have performed the GETEX mission at ETS-VII with very successful results. Now we believe that the extensive use of robotics at the ISS must be pushed by all industrial and political partners. Furtheron, in our opinion, there are only few space technologies which promise such high terrestrial spin-off and technology transfer potentials as the development of sensor-based task-level programming tools as well as intelligent (i.e. sensor-controlled) artificial robot arms. A recent example is the very encouraging feedback of the automotive industry during the Hannover fair '99, as we have applied the „vision&force“ control paradigma to insert pistons into a rotating motorblock fully automatically.

References

- ¹ M. Oda, K. Kibe, F. Yamagata, „ETS-VII – Space Robot In-Orbit Experiment Satellite“, IEEE Conf. on Robotics and Automation, Minneapolis, April 1996
- ² G. Hirzinger, B. Brunner, J. Dietrich, J. Heindl, „ROTEX – the First Remotely Controlled Robot in Space“, IEEE Conf. on Robotics and Automation, San Diego, May 1994
- ³ B. Brunner, K. Landzettel, B.M. Steinmetz, G. Hirzinger, „TeleSensorProgramming – a task-directed programming approach for sensor-based space robots“, Int. Conf. On Advanced Robotics (ICAR'95), Sant Feliu de Guixols, Spain, Sept. 1995
- ⁴ <http://www.robotic.dlr.de/Joerg.Vogel/Vrml/Rotex/>
- ⁵ K. Landzettel, B. Brunner, G. Hirzinger et. al., „DLR's robotics lab – recent developments in space robotics“, i-SAIRAS 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1-3 June 1999, ESTEC, Noordwijk, The Netherlands
- ⁶ B. Baginski, „Efficient Dynamic Collision Detection using Expanded Geometry Models“, IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems IROS'97, Grenoble, September 7-11, 1997
- ⁷ Longman, R.W., Lindberg, R.E. and Zedd, M.F., „Satellite-mounted Robot Manipulators – New kinematic and Reaction Moment Compensation“, International Journal of Robotics Research, 3, 1987.
- ⁸ Dubowsky, S. and Papadopoulos, E., „The Kinematics, Dynamics and Control of Free-Flying and Free-Floating Space Robotic Systems“, IEEE Transactions on Robotics and Automation, 5, 1993.
- ⁹ <http://esapub.esrin.esa.it/pff/pffv7n2/settv7n2.htm>
- ¹⁰ K. Landzettel, B. Brunner, G. Hirzinger, „The Telerobotic Concepts for ESS“, IARP Workshop on Space Robotics, Montreal, July 1994
- ¹¹ K. Landzettel, B. Brunner, G. Hirzinger, I. Schäfer, M. Fischer, M. Grebenstein, N. Sporer, J. Schott, „Space Robotics – Recent Advances in DLR's Robotics Lab“, 5th Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA'98), Noordwijk, The Netherlands, Dec 1998
- ¹² M. Fischer, P. van der Smagt, G. Hirzinger, „Learning techniques in a dataglove based telemanipulation system for the DLR hand“, IEEE Conf. on Robotics and Automation, Leuven, Belgium, April 1998

PROJECTIVE VIRTUAL REALITY: A NOVEL PARADIGM FOR THE COMMANDING AND SUPERVISION OF ROBOTS AND AUTOMATION COMPONENTS IN SPACE

Eckhard Freund and Juergen Rossmann

Institute of Robotics Research (IRF)
University of Dortmund, Otto-Hahn-Str. 8, 44227 Dortmund
 Email: rossmann@irf.de
www.irf.de

Abstract

Smart man machine interfaces turn out to be a key technology for service robots, for automation applications in industrial environments as well as in future scenarios for applications in space. Experience at the Institute of Robotics Research in this field showed that intuitively operable man machine interfaces can be developed most efficiently, if a twofold approach is taken. On the one hand the capabilities of the automation system is enhanced in a way that they can act and react more autonomously, but on the other hand, the development of an intuitively operable virtual reality based man-machine interface is pushed further to be able to command and to keep control over the system. Latest results gained from the realization of the ground control station for the Japanese space robot ERA proved impressively, that the realization of a man machine interface based on modern virtual reality (VR) techniques is a promising approach for a new command and supervision interface that is intuitively operable. The general aim of the development which has been used for the ERA robot was to provide the general framework for „Projective Virtual Reality“ which allows to „project“ actions that are carried out by users in the virtual world into the real world with the help of robots and other means of automation.

1 Introduction

When autonomous systems with multiple agents are considered or when intuitive control of automation means over long distances is required, conventional control- and supervision technologies are often inadequate because the amount of information available is often presented in a way that the user is effectively overwhelmed by the displayed data. New virtual reality (VR) [1][7] techniques can help to cope with this problem, because VR offers the chance to convey information in an intuitive manner and can combine supervision capabilities and new, intuitive approaches to

the control of autonomous systems. In the approach taken, control and supervision issues were equally stressed and finally led to the new ideas and the general framework for *Projective Virtual Reality*. The key idea of this new approach for an intuitively operable man machine interface for decentrally controlled multi-agent (= different robots and automation means) systems is to let the user act in the virtual world, detect the changes and have an action planning component automatically generate task descriptions for the agents involved to *project* actions that have been carried out by users in the virtual world into the physical world, e.g. with the help of robots. Thus the Projective Virtual Reality approach is to split the job between the *task deduction in the VR* and the *task "projection" onto the physical automation components* by the automatic action planning component [4] (see chapter 4). Furthermore, the presented new approach to virtual reality based man-machine interfaces for automation applications allows to present system status- and sensor information by means of intuitively comprehensible metaphors and visualization aids (chapter 5).

2 Applications of the Virtual Reality System

Practical experiences with the control of a multi-robot system showed, that with a new *task deduction* capability of the realized Virtual Reality system and the corresponding *action planning component* a new quality of intuitive controllability, observability and system safety can be provided. The methods and techniques described in this paper have been developed and tested for two space robotics applications and several industrial applications related to flexible assembly workcells. The two space robotics applications were most challenging in the way, that the new techniques of Projective Virtual Reality as a man machine interface have been applied in the most comprehensive manner.

2.1 Commanding the Japanese ERA Robot

Already in 1996, the Japanese Space Agency NASDA and the German Space Agency DLR agreed on a Memorandum of Understanding (MOU) in the field of space robotics. A major part of this agreement was related to the Japanese ETS-VII (Engineering Test Satellite) which has been developed by NASDA to perform Rendezvous & Docking (RVD) and Space Robotics (RBT) experiments.

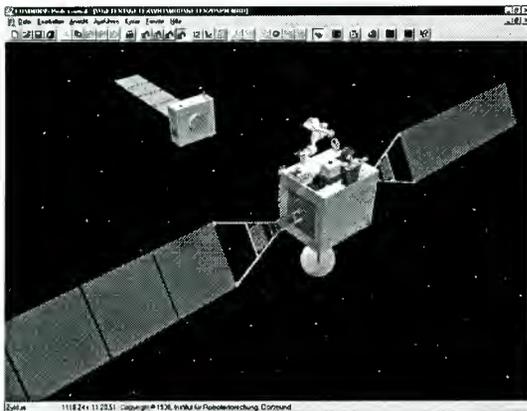


Fig. 1: Simulation of the Japanese ETS-VII Satellite

NASDA and the German Space Agency agreed on a cooperation project where the German side contributes the on-ground robot control and command station, which combines enhanced robot control and latest virtual reality techniques to provide intuitive control and supervision of the robot arm ERA onboard the ETS-VII. In April 1999, a team of the IRF traveled to Tsukuba, Japan, to install and run the ground control station based on the Projective Virtual Reality methods described in this paper.

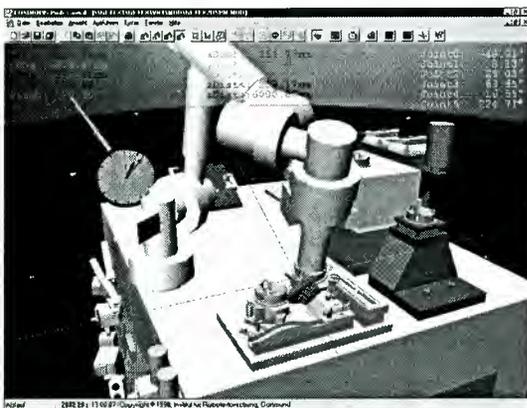


Fig. 2: View of the Projective Virtual Reality system to control and supervise the ERA robot

The mission in April 1999 was a great success. In five missions, different experiments were carried out which were safely commanded and supervised by the IRF ground station. The experiments ranged from simple routine tasks like gripper attachment and detachment over assembly benchmarks to manipulation tasks under force/torque control. (Details about the missions and

the experiments can e.g. be found under <http://www.irf.de/getex>).

In cooperation with our Japanese colleagues, we even had a *world premiere*: For the first time, a space robot was commanded and supervised on-line by means of a "immersive virtual reality interface" based on helmet and dataglove — without previous off-line preparations and detailed pre-checks of the generated robot commands to be expected. Figure 3 shows Dr. Mitsushige Oda, Principal Investigator at NASDA, controlling the ERA robot by means of the IRF Projective Virtual Reality system. Dr. Oda successfully commanded a mission of the ERA robot after only 10 minutes of introductory training.



Fig. 3: Dr. Mitsushige Oda, Principal Investigator at NASDA, controls the first IRF experiment by means of a data-helmet and a data-glove.

The basic ideas of Projective Virtual Reality comprising task deduction, task "projection" and commanding and supervision metaphors that will be discussed below could fully be applied to the ERA application; this new paradigm proved to be very successful.

2.2 Applications in "Internal Space-Laboratory Servicing"

The very first and the still the most comprehensive application of Projective Virtual Reality was to provide the man machine interface for the multi-robot system in the CIROS-(Control of Intelligent Robots in Space) testbed (fig. 4), a multi-robot system developed for space-laboratory servicing. CIROS, as a multi-robot system, is more complex than a single-robot system like ERA and thus allows to demonstrate convincingly how to exploit the inherent flexibility of a multi-agent system automatically by means of the new Projective Virtual Reality system. Thus CIROS will serve as the main example in the further discussion.



Fig. 4: The CIROS multi-robot testbed

The testbed developed in the CIROS project [3] is equipped with two redundant robots with 6 revolute and one prismatic axis each (fig. 4). The layout of the laboratory is similar to that of the Columbus Orbital Facility (COF), the European contribution to the International Space Station (ISS). Six racks, switches and other operating elements of the experiments were reproduced and arranged in order to be performing realistic operational sequences. A tool exchange capability and force/torque-sensors have been included to allow the robots to operate autonomously under the multi-robot-control IRCS developed at the Institute of Robotics Research.

The redundant two-armed robot configuration with the force/torque-sensors at the robots' wrists permit fully coordinated operation, similar to the cooperation capabilities of two human arms, as well as synchronized or independent action of the two robots, working together like a team. Furthermore, the robots are equipped with hand cameras and the whole laboratory can be supervised by a scene camera.

For this multi-robot-system, the VR-based man-machine interface allows the intuitive commanding of new tasks and permits the checking of status information and the intuitive „presentation“ of warning messages as well as messages concerning the successful completion of tasks. While designing the virtual environment for the CIROS testbed, emphasis was laid on providing „a familiar environment“ to an experimenter who conducts experiments in the space laboratory from ground with the help of the Projective VR system. In order to „immerse“ into the virtual reality, the experimenter wears a head-mounted-display (HMD) and a data-glove. Both tools are equipped with position and orientation sensors, so that the location of the HMD and the data-glove are known to a graphics workstation which generates the virtual, graphical image of the environment with respect to the operators position and viewing direction. Furthermore, a graphical image of the operators hand is shown, which allows to operate in the virtual environment (Fig. 5). For cost-sensitive applications, a desktop VR version is available that works with shutter glasses to provide a stereoscopic view into the workcell.

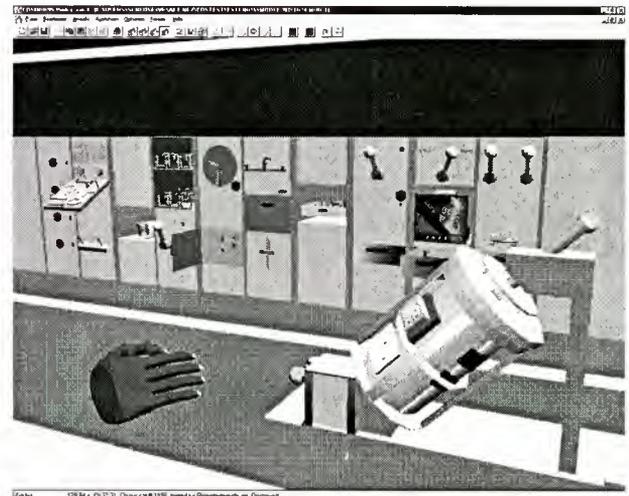


Fig. 5: The virtual laboratory

3 The Idea of Projective Virtual Reality Based Control

When we started to control the robots via VR, we immediately found that the standard teleoperation or "hand-tracking" approach would not work for most of our applications which contain assembly tasks [1][2]. The following problems arose:

- Time delays between the display of a robot's movement in the VR and its physical movements are critical for the stability of the process, because, similar to standard teleoperation approaches, the user is still "in a realtime control loop".
- The graphical model has to be very precise.
- The measurement of the position and orientation of the data-glove has to be very precise.
- Measures have to be taken to reduce "trembling" of the operators hand.
- A versatile sensor-control is necessary to compensate for unwanted tensions when objects are inserted into tight fittings.

To cope with the problems mentioned above, the previously mentioned *task deduction mode* [5] was developed. The solution was to enhance the VR-system in the way that while the user is working, the different subtasks that are carried out by him are recognized and task descriptions for the IRCS, the multi-robot control system of the CIROS environment are *deduced* (chapter 4). These task descriptions are then sent to the *action planning component* [5] of the IRCS. The action planning component can "understand" task descriptions on a high level of abstraction like "open drawer", "insert sample 1 into heater slot 1" etc. and thus is the ideal counterpart for the task deduction component of the VR-system. Using this task deduction mode is almost ideal, because:

- The required communication bandwidth is low, because only subtasks like "open flap", "move part A to location B" or "close drawer" are sent over the communication channel.

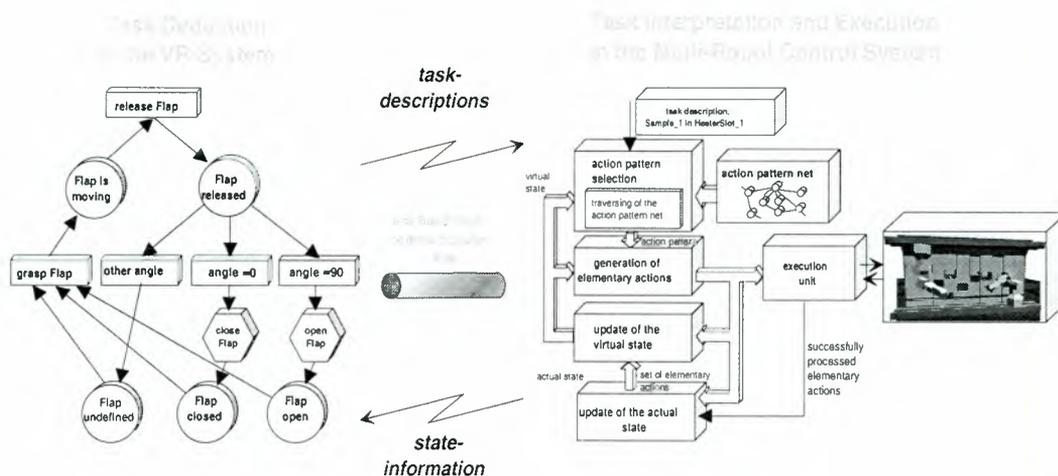


Fig. 6: Cooperation between the petri-nets for task-deduction and the action planning system

- The user is no longer in the "realtime feedback loop". Complete subtasks are recognized and carried out as a whole without the necessity for immediate feedback to the user.
- For assembly tasks, the accuracy of the environment model can be compensated for by automatic sensor-supported strategies.
- The accuracy of the data-glove tracking-device is not as important as for the direct tracking mode. The allowable tolerances when the user is gripping an object or inserting a peg into a hole can be adjusted in the VR-software.
- Different users working at different VR-systems can do different tasks that are sent to the planning component of the IRCS, which then can compute an adequate sequence of the tasks to be carried out, depending on the available resources. Thus one robotic system can serve e.g. multiple experimentators in a space laboratory environment.
- If the robot control is versatile enough, there is no longer a need to even show a robot in the virtual environment displayed to the user; so the user more and more gets the impression of carrying out a task "himself", which is the highest level of intuitivity that can be achieved.
- If the planning component is versatile enough, it cannot only control the robots, but also other kinds of automated devices. The action planning component in the CIROS environment "knows" that to open the leftmost one of the three drawers (fig. 4), it doesn't need to employ a robot. This drawer is equipped with a motor, so that it just has to control the motor to open this drawer. Robot-automated and hard-automated tasks are thus controlled under one unified framework.

In general terms, it is one of the key issues of Projective Virtual Reality is splitting the job between the *task deduction in the VR* and the task "projection" onto the *physical automation components* by the automatic action planning component. The necessary expertise to conduct an experiment in a space laboratory

environment like **CIROS** is thus *shared* between the user with the necessary knowledge about the experiment and the robot-control with the necessary "knowledge" about how to control the robots.

4 Task Deduction in the VR-Environment

The task-deduction module relies on messages from inside the VR system. Messages are generated and are sent to the task-deduction module for example when an object was gripped by the user, when an object was released or when the user's dataglove enters a certain region of the environment displayed in the VR.

These messages are interpreted by means of finite-state machines which can be visualized as petri-net structures. These structures determine whether the actions can be combined to a task description for the robotic system. Fig. 1 shows an example of such a petri-net which allows to deduce tasks like "open Flap" or "close Flap" from the actions a user is performing in the VR. Fig. 1 gives a simple example of a task-deduction network which allows to detect whether the user wants to open a flap. As a starting point, the flap shall be closed, so that we have to imagine a mark in the state "Flap closed" in the lower left part of fig. 1. During runtime, the task-deduction component is notified of different events related to user actions in the virtual environment.

For these events, different classes are distinguished, e.g. those related to interactions between the user and the environment by means of the dataglove, events related to user movements and events related to communication between the multi-robot control system and the VR-system. If the user grasps the flap, the corresponding message is evaluated, the "grasp Flap" transition fires, and a state-change in the petri-net is carried out from "Flap Closed" to "Flap is moving" (fig. 1). If the flap is released again, the state changes to "Flap released". For the next transition, the actual angle of the flap's joint has to be evaluated. If, for example, the user opened the flap, the angle is approximately 90

degrees, so that the mark is to be moved to "Flap open". On the way from "Flap released" to "Flap open" in fig. 1, we passed the six-edged "communication-symbol", which indicates, that the task description "open Flap" is to be sent to the action planning component of the robot control system at this time to have the robot perform this task physically.

5 Control and Supervision Aids in the Virtual World

The previous chapters outlined the basic ideas of controlling an automated system by means of Projective Virtual Reality: The user just acts in the virtual world, a corresponding task description is derived automatically by the task deduction component. The automatic action planning component in turn generates programs and commands for the physical automation means in order to have them carry out the corresponding action in the physical world. Besides providing this new control approach, the Projective Virtual Reality system developed at the IRF also realizes new ideas related to supervision, teleoperation and object placement aids to make the work in the VR as effective and as intuitive as possible

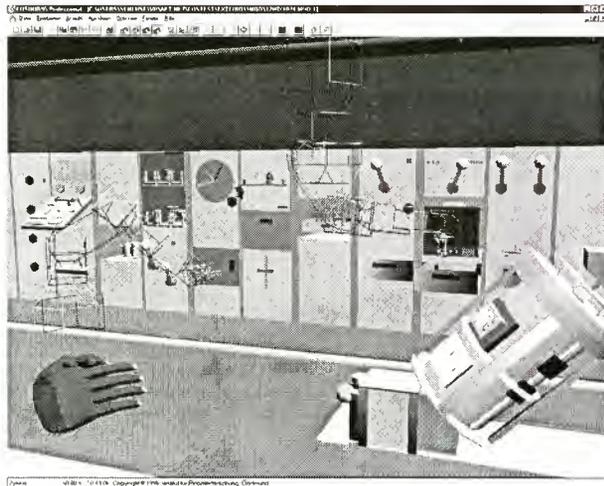


Fig. 7: The CIROS environment in supervision mode

As stated already in chapter 2.2, the virtual environment for the CIROS testbed was designed to provide „a familiar environment“ to an experimenter who conducts experiments in the space laboratory from ground with the help of the CIROS-VR-system as shown in fig.5. Please note, that in fig. 5 the robots are not shown in order to not distract the experimenters from the experiment they are conducting. But, as our VR system allows to have different views into the same virtual world, the view of the system operator might be different from that of a "plain user", because the operator usually is interested very much in what the automation means are currently doing. So most of the time he will activate at least the simplest supervision aid provided: the image of the physical robots as wireframes (fig.7). The wireframe representation of the

robots is only shown in "supervision mode", which can be switched on and off by the user through a simple gesture with the dataglove.



Fig. 8: Position and Placement Aids in the Virtual World.

Fig. 8 shows another two visual aids for the user in the virtual world; the image of the user's hand which has grasped an open sample container is displayed in the center of the left part of fig. 8. The first visual aid is the wireframe representation of the container displayed above the grasped container box. This wireframe — displayed in red on a color display — pops up in the virtual world, when the user approaches a sensible deposit for the object that is currently being grasped. The user may then release the object and it will snap to the wireframe position, so that minor trembling or errors of the dataglove position sensor can be compensated for automatically. The wireframe on the right of the container, the second type of visual aid, indicates the actual position of the physical container in the CIROS testbed which has not yet been moved by the robots. As soon as the robots start to move the physical container, this wireframe will move to the target position determined by the solid container and will vanish as soon as this position is reached thus indicating that the robots carried out the task successfully.

A last field where strong metaphors are necessary is the teleoperation and inspection support in virtual reality. Teleoperation in space applications mostly means, that a specialist at the ground station gets a video image from the robot system flying in space and he controls the robot by means of a joystick or a 6D space mouse. In order to also support teleoperation and video inspection by Projective Virtual Reality a first approach was to have a virtual camera that can be guided to the desired position. To do this, the user just grasps a virtual camera as a metaphor for teleoperation mode and positions it so that the desired object can be inspected. This action makes the action planning component switch the multi-robot control system to teleoperation mode and guide to the desired position a currently available robot that is equipped with a hand camera — as is depicted by the wireframe robot in fig. 9.



Fig. 9: Teleoperation commanded by the virtual camera

The deficiency of this idea is that the user, after having positioned the camera correctly in the virtual world, has to take off the head-mounted-display to watch the screen with the video image. This is an annoying procedure if applied practically, so that the metaphor of a "TV-View into Reality" was invented.

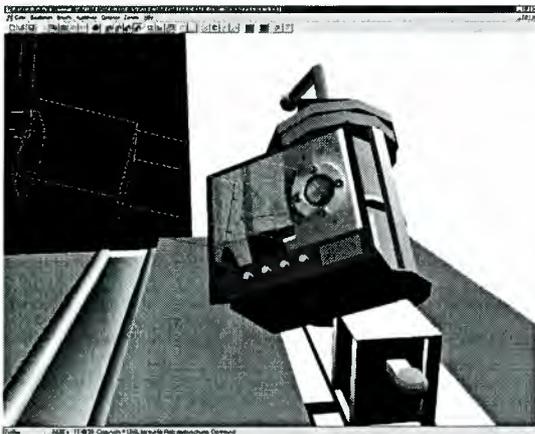


Fig. 10: Inspection with the help of the "TV-View into Reality" metaphor

Fig. 10 shows this new approach for inspection support. Instead of moving around a virtual camera to position the robot's hand camera, we replaced the virtual camera with a virtual TV and on the TV screen the actual video image is shown as a texture. Thus the user does not have to "leave" the virtual reality in order to get visual information about the physical environment. All he has to do is to "watch TV".

Apart from not having to leave the virtual world, this approach has another advantage over the first one: If the robot's hand cameras have to be turned and tilted to show a desired object, most users have difficulties imagining the current orientation of the camera, when they just look at the video screen. With the "TV-View into Reality"-approach, the user in the VR intuitively knows how the physical cameras are oriented by the orientation of the screen of the virtual TV. Most users that tested both methods preferred the virtual TV.

6 Conclusion

The given examples prove that Virtual Reality technology has developed the potential to become a key-technology for the design of modern man-machine interfaces not only for robotics in space environments. Whereas most VR-applications aim at the "plain VR", that is the improvement of the virtual worlds that are displayed to the user, this paper shows the application of new ideas related to *Projective Virtual Reality*, where the aim is to use VR-technology as an intuitively operable man-machine interface for robotic and automation systems. The presented new *task deduction approach* was developed to "project" virtual actions onto robotic systems, that is to make physical robots and other means of automation carry out the same tasks in the physical environment that have been carried out by the user in the virtual environment. Last but not least, the new metaphors to represent system states are a great help to be able to intuitively supervise complex automation systems.

This work was supported by the German Space Agency DARA/DLR under the Contracts No. 7984/40233, 50 TT 9434 and 6-056/0048

REFERENCES

- [1] .K. Bejczy: Kim W.S.: "Sensor Fusion in Telerobotic Task Controls", Pittsburgh, PA, August 1995
- [2] A.K. Bejczy: "Virtual Reality in Telerobotics", Barcelona, Spain, August 1995.
- [3] E. Freund, J. Roßmann: "Control of Multi-Robot-Systems for Autonomous Space Laboratory Servicing", Proc. of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS, Vol. 1, S. 443-453, Toulouse, France, September 1992.
- [4] E. Freund, J. Roßmann, K. Hoffmann.: " Automatic Action Planning as a Key Technique for Virtual Reality based Man Machine Interfaces", Proceedings of the Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'96), Washington D.C., USA, December 1996.
- [5] Freund. E.; Roßmann, J.: „Intuitive Control of a Multi-Robot-System by Means of Projective Virtual Reality“, Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems '96, pp. 273-280, Washington, December 1996
- [6] E. Freund, J. Roßmann, J. Uthoff, U. van der Valk: "Towards realistic Simulation of industrial Robots", Proceedings of the IEEE/RSJ/GI Intelligent Robots and Systems, IROS, Munich, Germany, September 1994.
- [7] K. Helsel (Editor): "Virtual Reality: Theory, practice and promise", Meckler, London, 1991, ISBN 0-88736-728-3.

A system integrating high and low level planning with a 3D-visualizer

Alberto Finzi, Fiora Pirri, Marco Schaerf
 Dipartimento di Informatica e Sistemistica
 via Salaria 113, 00198 Roma, ITALY
 e-mail{finzi,pirri,schaerf}@dis.uniroma1.it

Abstract

One of the most difficult problem in the designing of planning systems concerns the integration of several reasoning components like sensing, perception and high level planning together with robotics modeling techniques. In this paper we present a simulation system for a robotic arm operating on a platform of the ISS (JERICO domain). We have defined a hierarchy of planners at the task, global and local level that suitably interact to account for different levels of control of the execution of the tasks. The task planner, realized in GOLOG, utilizes a KB in the Situation Calculus to find the sequence of abstract actions necessary to reach the goal. The global and local planners expand each abstract action into a more refined sequence computing a path in the workspace (global planner) and then in the configuration space (local planner). We illustrate the 3-dimensional graphical interface and the robot simulation module and how it interacts with the planning system.

1 Introduction

One of the most difficult problem in designing a planning systems concerns the integration of several reasoning components like perception, scheduling, execution monitoring and planning together with manipulation planning, motion planning and sensing. As McDermott and Hendler have remarked in their introductory paper of the AI-journal special issue on planning [1], scaling a planning domain usually yields a set of problems that involve a lot of reasoning techniques from other fields. In this paper we present a proposal for decomposing a manipulation planning problem into a hierarchy of planners and integrate them with a 3D-visualizer. The idea of adopting a hierarchy of models has been also investigated by Cameron [5]. The novelty of our approach relies on the fact that we use a symbolic model of the domain for the task/high level planner as opposed to the geometrical model of the workspace used for the low level planners. The advantage of a hierarchy of planners is manifold. Here we quote only two aspects: computational complexity and modularity. For the first aspect consider that the domain independent planning problem is in general undecidable; however under certain restrictions like when there are no function symbols and only finitely many

constant symbols then planning is decidable and its computational complexity varies from constant time to EXPSPACE-complete [6]. On the other hand planning a manipulation path to bring the movable objects to their specified goal location is PSPACE-hard [21]. Now, for planning, most of the complexity is to be found in the way preconditions for actions are formally represented while for manipulation – and in general motion – planning it is to be found in the geometrical representation of the workspace and in its dimension. Keeping the geometrical model of the workspace separated from the symbolic model of the domain is thus necessary to avoid an increase in complexity on both the planners. Another advantage of the hierarchy is modularity. Local planners depend on a particular robot, on its linkages, joints, degree of freedom, while global and task planners can be both formulated independently from the specific structure of the manipulator. Therefore, under a suitable decomposition, the same task and global planners can be adapted to several manipulators.

The paper is organized as follows. In the next section we introduce some preliminaries just to specify the notation. Then we present the hierarchical planner decomposed into task, global and local planners and we discuss some example in the literature that use an analogous hierarchy. In Section 4 we introduce the symbolic model formalized in the Situation Calculus and the way primitive and complex actions are managed via the axiomatization and the programming language GOLOG. In Section 5 we discuss the geometrical model of the workspace in which all objects are assumed to be convex. We then introduce the global planner and its interaction with both the task and the local planner. In Section 6 we introduce the local level problem, the module taking care of the robot kinematics in the configuration space together with the graphic module that simulates the robot executing the complex tasks required to achieve a goal. Finally we address some further issues that we have not included in this presentation.

2 Preliminaries

In most part of the paper we refer to the Situation Calculus [16, 8], a first order language with sorts. The three disjoint sorts are: *action* for actions, *situation* for situations and *object* for everything else depending on the domain of application. We refer the reader to the literature, e.g. [17], for a detailed presentation of the alphabet of $\mathcal{L}_{sitcalc}$ and for the metalanguage adopted to denote terms and formulae of the language. The alphabet includes relations and functions called Fluents, because their truth value depends on the history of actions performed by an agent: a history, like

[*grasp(payload), rotate(payload), ungrasp(payload), rotate(handle)..*]

is designated by a situation s . A situation s is the last argument of Fluents, e.g. $Handle(payload5, do(pickUp(payload5, S_0)))$, where $s = do(pickUp(payload5, S_0))$. The Situation Calculus is a powerful basic axiomatization for representing dynamic domains. We define a *basic theory of actions* to be a set of axioms describing the preconditions for each action that can be dealt with by an agent, via the *Action Precondition Axioms*, the effect of each action via the *Successor State Axioms* and the initial situation that we call D_{S_0} , in which no action, relative to the current task, has been executed.

Example 1 Consider the following fluents:

inContainer(p, n, s) :
 payload p is in nest n in situation s
locked(n, s) : nest n is locked in s
position(pos, s) :
 end-effector is in position pos in s
holding(obj, s) :
 end-effector is holding the obj in s
posPlat(pos, s) : position of the mobile platform

And the following *primitive actions*:

goto(pos) : move to the pos position
lock(nest) : lock the $nest$
unlock(nest) : unlock the $nest$
extract(nest) : extract the object (if it exists) contained in $nest$
insert(nest) : insert the object (if the arms is holding something) in $nest$
move(plat, θ) : move the mobile platform to the specified orientation θ

An initial situation S_0 can be defined in this way:

$locked(nest_1, S_0)$
 $\neg \exists p(holding(p, S_0))$

Actions precondition axioms for each primitive action have the form:

$Poss(lock(nest), s) \equiv$
 $position(handle(nest), s) \wedge \neg \exists x(holding(x, s)) \wedge$
 $\neg locked(nest, s)$

Successor State Axioms for each fluent have the form:

$locked(nest, do(a, s)) \equiv a = lock(nest) \vee$
 $locked(nest, s) \wedge a \neq unlock(nest)$

Complex actions can be dealt with via the programming language GOLOG [15] (alGOL in LOGic), whose declarative semantics is given in the Situation Calculus. GOLOG is a logic-programming language which, in addition to the primitive actions axiomatized as specified above, allows the definition of complex actions using programming constructs which are like those known from conventional programming languages like conditionals, iteration, procedures.

What is special about GOLOG is that the meaning of these constructs is completely defined by sentences in the Situation Calculus. For this purpose, a macro $Do(\rho, s, s')$ is introduced whose intuitive meaning is that executing the program ρ in situation s leads to situation s' . Here we provide some sample definition needed for Do. See [15] for the complete list.

$Do(A, s, s') \doteq Poss(A, s) \wedge s' = do(A, s)$, where A is a primitive action.

$Do(\text{if } \varphi \text{ then } \rho_1 \text{ else } \rho_2 \text{ endif}, s, s') \doteq$
 $Do([\varphi?; \rho_1][\neg \varphi?; \rho_2], s, s')$

Here φ is a formula of the Situation Calculus with all situation arguments suppressed.

For specific sections of the paper we assume the reader familiar with basic robotics terminology, we refer to [10] for a full introduction. We recall that a *Configuration* is a mathematical specification of the position and orientation of every body composing a robot, relative to a coordinate system. The configuration space C is the set of all configurations of a robot. The configuration space has dimension m , where m is the number of the degree of freedom (dofs). The number of dofs of a robot arm is equal to its number of joints. We denote by C_{obj} the configuration space of the object Obj .

3 The Hierarchy

Consider a manipulator, a 6 or 7 degree of freedom robot arm working on a platform where there are payloads installed in nests and locked. An action like $pickUp(payload5)$ can be considered a primitive one, at the level of abstraction at which we are used to think about *simple actions*. On the other hand, picking up the payload may require a huge amount of simpler actions like verifying whether the payload is really reachable, where it is, and thus moving to the payload position – avoiding all the obstacles – unlocking the handle of the nest where the payload is installed, rotating the end-effector once or twice so as to rotate the payload for detaching it from the nest and finally pulling the payload out of the nest.

Now, we are still missing something: each of the more detailed actions in which $pickUp$ has been decomposed actually refers only to the end effector. In fact,

we have to consider the whole arm, which is a collection of bodies, connected by joints, having constraints. These constraints have to be satisfied in the space of all the configurations that the robot arm can assume, while *passively* following the end-effector. What we have just described is the simplest and natural hierarchy that a manipulation planning problem requires. The hierarchy we are proposing decomposes the planning problem into different abstraction levels allowing to manage challenging domains both from a conceptual (logical) point of view and from the geometrical and dynamical ones. Examples of three layered architectures can be found in [13], where successful mobots, developed with the P-SA approach, are described. Interesting examples are *RHINO* [18] and *Saphira* [14]. The upper level of these systems requires a Task Planner but the lower levels are usually reactively managed. This is possible because the robots considered are involved in tasks in which low level behavior can be driven on-line. In general, however, for manipulation planning where robots arms are involved, off-line planning at each level is required.

Task level We consider an autonomous agent potentially able to achieve any complex task like delivering hot coffee in an office, cooking pasta and serving it, moving any number of blocks on a table so as to form any sophisticated shape [17]. All these complex tasks are potentially achievable as far as we are concerned with a symbolic model of the world, taking care of the causal laws governing preconditions and postconditions of each primitive action, together with a suitable solution to the frame problem [16]. A solution to the frame problem specifies what in the domain has been changed and what remains unchanged after the execution of an action. Tasks and goals at this level are formalized using a domain theory and a basic theory of actions (see Section 2 above) that provides, for each primitive action that can be executed, i.e. whose preconditions are satisfied, a full description and formal characterization. Complex actions can be obtained by composing primitive actions in the programming language GOLOG.

At the Task level the preconditions to any control action already en-globe a solution to the problems of controlling the real forces applied to the end effector, of finding a free space for the path needed to execute the action, and of a transfer path for correctly manipulating the objects. In other words any action executed at the task level *can* be executed because all the space problems have *already* been solved. As the formalization relies on this assumption, at the task level each primitive action can be considered as an idealized representation of the physical world and the agent as a free-flying object.

The role of the task planner is to give the agent the postulates to reason about the domain and to coordinate her actions in an intelligent behaviour so as to achieve the required goals.

Example 2 Consider the domain JERICO (Joint European In-Orbit Calibration and Operation) defined for the Russian segment of the International Space Station, see Figure 1. Given an initial configuration of the payloads, locked in their nests, and an initial configuration of the exchange terminal and the pointing platform, the agent – the robotic arm – has to re-orient the pointing platform and move the payloads from their nests to other nests by turning the nest-handle to unlock the payload, transferring them to the required nest, eventually using the exchange terminal, turning the payload to insert it in the nest and finally locking the handle. Observe that to remove or install a payload into a nest the agent has to ungrasp the handle and regrasp it such that it can be rotated.

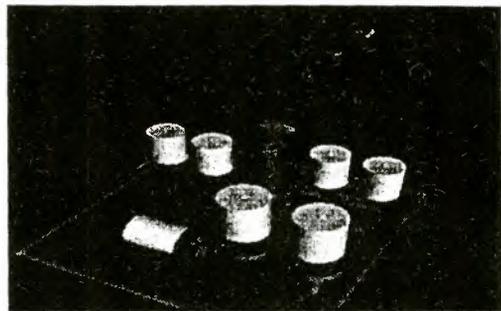


Figure 1: Jerico domain

The Task planner provides us with very interesting off-line plans that can also take into account sensing and perception.

Global level In Latombe [12] a manipulation path is defined as an alternating sequence of *transit* and *transfer* paths that connect an initial configuration q_{init} to a goal configuration q_{goal} . A *transit* path is an arm's motion that does not move any object. A *transfer* path defines an arm's motion that does move an object. In our hierarchical model, actions are executed only at the task and global level, therefore transit and transfer paths are defined at the global level. The global level is formalized within a geometrical model of the agent workspace in which both the agent and objects are assumed to be convex. Objects in the workspace, including the robot end effector are represented within particular bounding volumes called cylspheres [9], that is, cylinders with semi-spheres of the same ray of the cylinder added on top, see Figure 2.

The global level takes care of computing a manipulation path – free from collision – for the end effector from an initial situation S_0^g to a final situation S_{goal}^g that satisfies the postconditions of a given task action a . In other words, given a task level action a , the global level *expands* such a single action into a sequence of manipulation actions $\{ma_1, \dots, ma_n\}$ that satisfy the geometrical constraints of the workspace, that is, avoids the obstacles and correctly manipulate the movable objects.

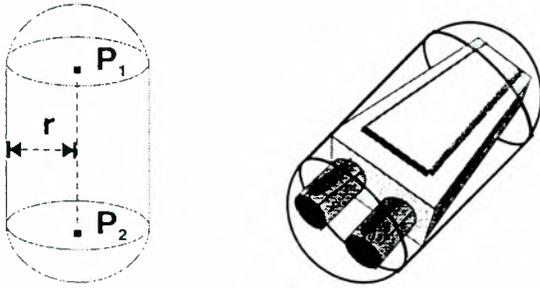


Figure 2: A cysphere bounding the end effector.

The role of the global level is to ensure that all the preconditions required to execute action a – within the workspace – are satisfied. Our strategy uses Latombe idea [11] consisting in representing the end effector as having a dynamic shape that changes together with the objects it is manipulating; see Figure 3.

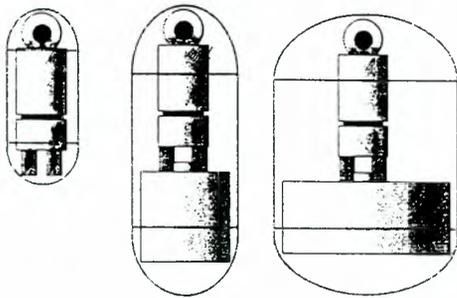


Figure 3: The end effector shape depends on the object is manipulating.

We formalize the geometrical model also in the Situation Calculus.

Example 3 Suppose the task level has delivered a sequence of actions $[a_1, \dots, a_m]$ and its present situation is s . Suppose also that at situation s the action $a = \text{pickUp}(\text{payload5})$ has to be executed, the task level queries the global level to verify whether the geometrical preconditions for a to be executed are satisfied. The current situation s is transformed into the global initial situation S_0^g , which is the start situation for the global planner. A sequence of actions $[ma_1, \dots, ma_k]$ is then computed by the global planner and are such that the situation s^g reached by the execution of these actions satisfies the postconditions of $\text{pickUp}(\text{payload5})$.

Local level The *local planning* step manages the whole structure of the arm, namely its end-effector, elbow, joints etc. The local planner makes a constrained search in order to achieve a safe path for each joint. The movements to which the arm is committed are strictly dependent on local information.

Planning at this level can be done in several ways. A variable that influences the local planner architecture is the grid step used in the global planning phase. In particular, if this step is small then a *ONE-SHOT planner* it is needed [11] that reaches directly a final position for the robot avoiding collisions. Otherwise, if the step is large, a more powerful planner [3, 5] is necessary that produces intermediate configurations for the arm.

Our local planner belongs to the category of *ONE-SHOT planners* that are based on inverse kinematics algorithms. These algorithms iteratively calculate a final configuration starting from an initial one and a final position for the end-effector. The module that calculates the inverse kinematics is the same used by the simulator.

Discussion Our architecture is close to the idea described in [5], although our Task level is far different from the one proposed by Cameron as we use a symbolic model formalized in the Situation Calculus. Cameron's view is to split between tactical knowledge (Task and Global levels) and geometric (Global and Local levels). We agree that the Cameron's structure has several advantages: the decomposition came natural in solving the manipulation and planning problem, the planner is easier to understand and to modify and eventually to adapt to a new domain or to upgrade it. Our system has developed the Task layer, that in the Cameron system is considered as a marginal aspect of the architecture, and its connection with the Global level (consistency between the two representation and communication between the two modules). In addition our Global Planner is developed as an interface between the logical representation and the geometrical one. Our representation of the world is mixed: the metrical representation is connected to a Knowledge Base that allows, when necessary, to perform some spatial reasoning. At the local level the system proposed by Cameron uses an approach based upon *virtual forces* whilst we have used a kinematic approach. The Global-Local interaction therefore is similar to the one described in [3] where Local planning consists in special inverse kinematic algorithm and Global planning is developed using RPP.

4 The Task planner

At the task level, we define a basic theory of actions representing the virtual attitude of the agent to reason about the domain. A sequence $[a_1, \dots, a_n]$ of actions that the agent executes at this level leads the agent into a situation s . In s the domain has been transformed by the actions executed by the agent. The transformation is witnessed by the truth values of the fluents. When the language is suitably restricted, the set of fluents $\langle F_1(s), \dots, F_n(s) \rangle$ which are entailed by the basic ac-

tion theory, at situation s , is a state that will be used to interact with the global and local levels.

Following Green [7], given a set of conditions on the domain that has to be satisfied and which we call a *Goal*, a *plan* is any sequence of actions $[a_1, \dots, a_m]$, whose preconditions are satisfied and are such that in the situation $s = [a_1, \dots, a_m]$ the *Goal* is verified. Formally, if \mathcal{D} is a basic theory of actions, as we defined in the preliminaries, and *Goal* is a set of conditions, we require that:

$$\mathcal{D} \models \exists s. Executable(s) \wedge Goal(s)$$

In the Situation Calculus, since the preconditions for each action are suitably axiomatized the above definition implies that s is a plan whenever $\exists s Goal(s)$ is a theorem of the basic theory of actions. In particular, given an initial domain specification \mathcal{D}_{S_0} , if \mathcal{D}_{S_0} is a complete theory about the initial situation, it is always possible to determine whether there exists a situation $s = [a_1, \dots, a_n]$ in which the *Goal* is satisfied and such a situation, if it exists, can be constructively given via any sound and complete deductive method.

The axiomatization of the situation calculus ensures that the search space is a tree rooted in S_0 . Starting from the initial situation, the Task Planner searches for a sequence of actions that leads to a situation where the goal is satisfied. The search is driven by an heuristic that can be well defined using the expressiveness of the language: the heuristic is described in the Situation Calculus as well, using the fluents introduced for the basic theory of action, that express both the knowledge and the meta-knowledge.

Following the approach of [2] we use a domain specific knowledge to control the search of a forward chaining planner. To this end we have introduced two special fluents: *badSituation(s)* [17] and *sugg(a, s')* that indicates respectively: a situation s in which it is not useful to search the goal and the action a that is suggested in the situation s' . For example:

$$badSituation(do(goto(x), do(goto(y), s)))$$

The above statement cuts out all situations in which the end-effector moves toward a position and then moves away without accomplishing any task in the situation where she arrives. On the other hand a suggestion can be defined as follows:

$$\begin{aligned} & sugg(insert(payload(y), nest(x)), s) \\ & \leftarrow goodInCont(payload(y), nest(x)) \end{aligned}$$

here, *goodInCont* is a predicate that is true iff the payload y must be in the nest x in the final configuration. This formula suggests to insert a payload in a nest (when it is possible) that must contain that payload in a final configuration.

An interesting property of this planner is that it represents a compromise between deductive planning and planning as a search process. The Task Planner is implemented as a GOLOG procedure that searches for a plan in the space of situations. If a sequence $[a_1, \dots, a_n]$ satisfies the *Goal* then it is accepted as the plan.

The Knowledge Base can be easily implemented as a PROLOG program. In the case of a complete representation of the domain, it is possible to exploit PROLOG as a theorem prover (in this case negation as failure is valid), otherwise (see Open Word Golog in [17]) it is necessary to use a theorem prover ad hoc developed for domains written in the Situation Calculus. The GOLOG interpreter is written in Prolog as well [17].

Developing a Task Planner in GOLOG has several advantages. With this language it is possible to exploit properties like: quick prototyping, expressiveness of the KB, integration between knowledge and meta-knowledge, integration between procedural and denotational way of programming using automated reasoning just when it is strictly necessary. These features are very important: GOLOG is a procedural language that can directly use the Knowledge Base to deliberate when it is needed. In this way, during the execution of the program, it is possible to access the Knowledge Base testing the validity of some property, but also to control the execution by explicit meta-level knowledge (in our case the heuristics defined by the fluents *badSituation(s)* and *sugg(a, s)*). Therefore with our GOLOG planner the trade-off between expressiveness of the Knowledge Base and computational complexity of the planning task is addressed finding a way between writing a high level control program (that is the classical GOLOG approach [15]) and developing a backward search planner.

5 The global planner and the geometric domain

Objects and the end-effectors are represented by particular bounding-volumes called cylsphere [9]. A cylsphere is just the 3D projection of a segment and its geometrical structure is defined by the centers of the two semi-spheres and by the ray common to the cylinder. A basic volume of this kind is well specified using two points and a ray. As we observed above we represent the end effector as a cylsphere of varying dimension, depending on the payload carried in the transfer part of the manipulation.

The distance between two cylsphere can be reduced to the distance between two segments. The distance point-segment is defined along the perpendicular to the line, to which the segment belongs, passing through the

point P .

$$M = P_1 + At$$

with $A = (P_2 - P_1)/\|P_2 - P_1\|$ the verse of the line and P_1, P_2 the extreme points of the segment. Once the parameter t is known, the distance d is:

$$d = \|P - M\|$$

The distance between two segments is always defined on the perpendicular to the line to which the segment belongs but one has to take care of problems like coplanarity and parallelism. The distance between two cylinders is defined accordingly. In fact it results from the distance between the segments defined by the extremes of the solids to which the value of the rays has to be subtracted. A function *Box* applied to any object in the geometrical model will give back the bounding volume of the object as a cylinder. The geometrical model is also axiomatized in the Situation Calculus, but the domain objects denotes only the reals. To capture the relations between objects we define a hierarchy similar to the one adopted for graphical applications. Each object is represented using two parameters: the distance between vertices and the ray and a transformation function involving the ancestor nodes in the hierarchy. A functional fluent $Edge(x, y, s^g)$ represents the geometric transformation of x w.r.t. y in the geometrical situation s^g .

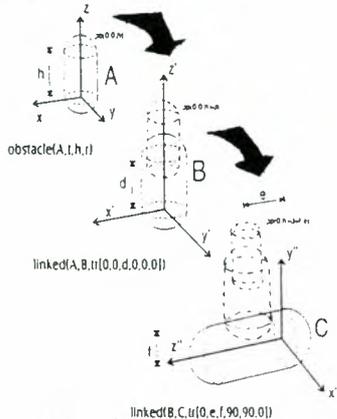


Figure 4: The object hierarchy

Movable objects are simply linked to the nests frames, according to the current task state. A subset of the objects that could be considered as obstacles for the manipulator is defined as a state, that is, at a given situation s each object is described in terms of the coordinates of its cylinder; see Figure 4). The exploration starts from the node specified and goes back towards the root applying all the transformation encountered to the cylinder-sphere contained in the starting node.

Global planning means searching for a manipulation path of the end effector as if it will be free from the rest of the body. The global planner generates a sequence of wrist positions and orientations so that the end-effector shall avoid obstacles and reach the final task.

The search algorithm proposed is a special translation of A^* driven by a heuristic that minimize the straight-line distance between the current state and the goal one. The expansion step takes care of the current arrangement of the end-effector and of the payload carried represented with their own cylinder-sphere, and avoids all states that generate a collision in the environment.

Each orientation of the end-effector comes from an interpolation between the initial and the final orientation desired, re-calculated at each iteration of the A^* algorithm. See Figure 5).

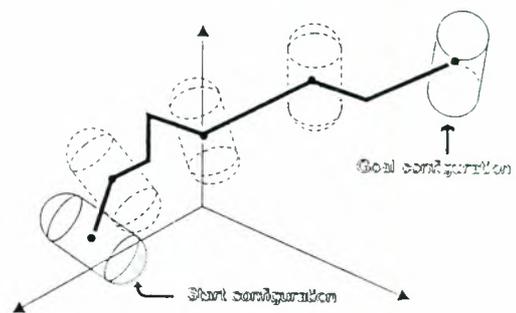


Figure 5:

To improve efficiency, we introduce a grid in the 3D space whose step-size is determined by the complexity of the world where the manipulator acts. A large step decreases the number of moves needed to reach a goal state, but could generate a non collision-free path because it doesn't take care of obstacles that lie between two adjacent positions. On the other hand a small step increases the resolution, but also the number of steps required to reach the goal. For this reason we define a variable step that can be defined at the beginning of the computation according to the complexity of the environment. The global planner has been implemented in Prolog.

6 Local Planner

The role of the local planner is to verify and refine the manipulation plan delivered by the global planner. At this point some paths may be found to be impossible; in such a case the global planner has to find alternative solutions otherwise the task planner has to re-plan.

The local planning problem, in our hierarchical structure, is defined as follows. A configuration C_{obj}^s is generated by the state of the global planner defined at situation s . A state is the vector $\langle p_1(s) \dots p_n(s) \rangle$ of all positions and orientations of the objects in the work space at the situation s^g , that is the current geometrical situation of the global planner. The configuration C_{e-f}^s is the subset of the work space occupied by the end-effector and generated by the state

$\langle q_1(s) \dots q_m(s) \rangle$, where $q_1(s) \dots q_n(s)$ are the positions and orientations of the cylindersphere bounding the end-effector and eventually the object it is manipulating.

Given C_{obj}^s and C_{e-f}^s and a sequence of actions $[ma_1 \dots ma_n]$ executable at the global level and a sequence of states associated with situations $s_1 \dots s_n$, the problem is to find configuration spaces $C_1 \dots C_n$, where C_i is the set of all configurations of the robot arm and objects at the state $\langle p_1(s) \dots p_n(s), q_1(s) \dots q_m(s) \rangle$, such that there exists a collision free path for the whole arm for executing $[ma_1 \dots ma_n]$. Observe that since $[ma_1 \dots ma_n]$ is a coarse expansion of a task action a , in order to find a manipulation sequence, collision free, for the end effector, any subset of $[ma_1 \dots ma_n]$ satisfying the preconditions and postconditions or any sequence of configurations for a , at the geometrical model, would be accepted. To solve the local planning problem we have used an inverse kinematics algorithm based on the computation of the transpose of the Jacobian matrix [19].

The method relies on the linear relationship between end-effector and joint velocities; it was early introduced by Wolovich and Elliot [20]. Sciavicco Siciliano in [19] applied the method to redundant manipulators and showed that the redundant degrees of freedom could be used to satisfy both obstacle avoidance constraints, and constraints on joint ranges of motion.

The method works as follows. Considering a composite force F applied to the end-effector, this external force will result in internal torques and forces at the joints. The relation between F and the internal forces τ can be written as:

$$\tau = J^T F \quad (1)$$

This suggests an iterative method for forcing the end-effector to track a time-varying trajectory $x_d(t)$. If the current end-effector position is $x_c(t)$, then the error measure.

$$e(t) = x_d(t) - x_c(t) \quad (2)$$

can be thought of as a force f pulling the end-effector toward the desired trajectory point $x_d(t)$. From this force we can calculate the joints velocities q' .

$$q' = J^T F \quad (3)$$

A single integration step yields a new vector q which moves the end-effector towards $x_d(t)$. This procedure repeats until the end-effector reaches the desired position, or some other stopping condition is met.

The method ensures that only forward kinematic calculation is required and in general problems with matrix singularities are avoided. Their occurrence can be overcome using an integration method with an adaptive step-size.

The local planner has been implemented in JAVA.

7 The graphic Interface

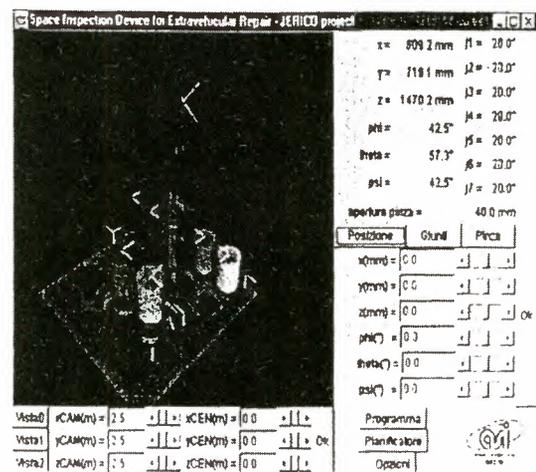


Figure 6: Graphical interface

The three-dimensional user interface allows to manage in a visual and friendly way the operations required by the robot arm during the execution of the tasks. In particular the user can perform the following operations.

1. Observe the evolution of the scene in a window visualizing the 3-D animation of both the robot and the object it is manipulating.
2. Observe the state of the robot on some panels in which there are information about end-effector position and orientation, joints angles, etc.
3. Manually interact with the robot arm specifying a final configuration for the arm: joints angles, end-effector position, hand opening etc.
4. Writing a program to accomplish a specific task.
5. Graphically define a final scene configuration in terms of payload dispositions in nests, or nests and platform orientations.

The actual scene is represented in a window and the final one is obtained modifying this scene by the mouse. The final configuration activates the planner that produces a program directly executed by the robot simulator

8 Conclusions

We have presented a modular decomposition of a planning system for a manipulator. We have integrated our system together with a simulator and a graphical interface. We have developed the system with conditional plans and perception. The modularity allows to take care of both perception and spatial reasoning. We are now concerned with the run-time behaviour of the planner and with execution monitoring.

Acknowledgments

This project has been funded by ASI (Italian Space Agency). We thank Ray Reiter, Andrea Pompili, Giuseppe Rossi and Angelo Dell'Arciprete for their help and important contributions to the realization of this work.

References

- [1] D. McDermott, J. Hendler *Planning: What it is, What it could be, An introduction to the Special Issue on Planning and Scheduling. Artificial Intelligence Journal* vol. 76(1-2), pages 1-16, 1995.
- [2] Bacchus and Kabanza. *Using Temporal Logic to Control Search in a Forward Chaining Planner. New Directions in Planning, M.Ghallab and A.Milani(Eds.)IOS Press* pages 141-153, 1996.
- [3] J.Barraquand, J.C.Latombe. *Robot Motions Planning: A Distributed Representation Approach Int. J.Rob.Research*, 10(6), 1991.
- [4] A. McLean, S.Cameron. *Effective Path Planning and Collision Avoidance for Redundant Manipulator. Int. Conf. Adv. Rob. & Comp. Vision*, 1996.
- [5] S.Cameron. *Dealing with Geometric Complexity in Motion Planning.. IEEE Conf. Rob. & Automation*, 1996.
- [6] K. Erol, D.S. Nau, V.S. Subrahmanian. 1995 *Complexity, decidability and undecidability results for domain-independent planning. Artificial Intelligence Journal* vol. 76(1-2), pages 74-88, 1995.
- [7] C.C.Green. Theorem Proving by resolution as basis for question-answering systems. In B.Meltzer and D.Michie, editors, *Machine Intelligence 4*, pages 183-205. American Elsevier, New York, 1969.
- [8] P.J.Hayes and J.McCarthy. 1969. *Some Philosophical Problems from the Standpoint of Artificial Intelligence. Machine Intelligence 4*. B.Melzer and D.Michie eds., Edinburgh University Press 463-502.
- [9] D. Henrich, X. Cheng. Fast Distance Computation for on-line Collision detection with multi-arm robots. In *IEEE International Conference on Robotics and Automation* pp. 2514-2519, 1992.
- [10] J.C. Latombe. *Robot motion planning* Kluwer Academic Publishers, 1991.
- [11] Y. Koga, K. Kondo, J. Kuffner, J.C. Latombe. *Planning Motions with Intentions* In *Proc. SIGGRAPH'94*, 394-408, 1994.
- [12] Y. Koga and J.C. Latombe. *On Multi-arm manipulation planning* In *International Conference on Robotics and Automation*. pp 945-952, 1994.
- [13] D. Kortenkamp, R. Bonasso and R. Murphy eds. *AI-based Mobile Robots: Case studies of successful robot systems* Cambridge, MA: MIT Press. 1998
- [14] K. Konolige and K. Meyer. The Saphira architecture for autonomous mobile robots. In *AI-based Mobile Robots: Case studies of successful robot systems* Cambridge, MA: MIT Press. 1998
- [15] Y. Lesperance, H.J. Levesque, F. Lin, R. Reiter and R.B. Scherl. GOLOG: a logic programming language for dynamic domains. *Journal of Logic Programming*, 31, 59-84, 1997.
- [16] R.Reiter. *The Frame Problem in the Situation Calculus: A simple solution (sometimes) and a completeness result for goal regression. Artificial Intelligence and Mathematical theory of Computation: papers in Honor of John McCarthy, Vladimir Lifschitz (ed.)*, Academic Press, San Diego, CA, 1991, pp. 359-380.
- [17] R. Reiter. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamic Systems*. MIT Press, to appear. <http://www.cs.toronto.edu/~cogrobo/>
- [18] Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., The Interactive Museum Tour-Guide Robot, *AAAI-98*.
- [19] L. Sciavicco and B. Siciliano. A dynamic solution to inverse kinematic problem of redundant manipulators. In *IEEE Int. Conference on Robotics and Automation*, pages 1081-1086, March 1987.
- [20] W.A. Wolovich and H. Elliot. A computational technique for inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 1988
- [21] G. Wilfong. Motion Planning in the presence of movable obstacles. *Proceedings of the 4th ACM Symposium on Computational Geometry*. 1988, pp 279-288.

Systems Aspects of Space Manipulation

**SAFETY APPROACH
OF JAPANESE EXPERIMENT MODULE
REMOTE MANIPULATOR SYSTEM**

Matsueda Tatsuo, Naoki Satoh, Takahisa Satoh, Yasushi Hisadome, Shinobu Doi* / NASDA
Fumihiko Kuwao⁺ / Toshiba

* National Space Development Agency of Japan, JEM Project Team,
Tsukuba Space Center, 2-1-1 Sengen, Tsukuba-City, Ibaraki, 305-8505 JAPAN
Phone: +81-298-54-3934, Fax: +81-298-50-1480, E-mail: doi.shinobu@nasda.go.jp

+ Toshiba Corporation, Komukai Works
1, Komukai-Toshiba, Saiwai-ku, Kawasaki-City, Kanagawa, 210-8581 JAPAN
Phone: +81-44-548-5125, Fax: +81-44-541-1211

ABSTRACT

The Japanese Experiment Module (JEM) Remote Manipulator System (JEMRMS) is a JEM element and will be used for exchange of Payloads (P/L) and Orbital Replacement Units (ORU).

In general a malfunction of a robotics system or improper operation by Intravehicular Activity (IVA) crew might cause catastrophic hazards for crew members or ISS itself. Very strict H/W and S/W safety design is required to prevent these hazards. Therefore, JEMRMS has many safety-related functions.

This paper summarizes the safety-related design of JEMRMS and the Manipulator Flight Demonstration (MFD) which was conducted as a flight demonstration of JEMRMS prior to JEM launch.

1. INTRODUCTION

JEM is the major Japanese contribution to the International Space Station (ISS) and consists of the Pressurized Module (PM), Experiment Logistics Module - Exposed Section (ELM-PS), Exposed Facility (EF), Experiment Logistics Module-Exposed Section (ELM-ES) and JEMRMS. The EF provides an external experiment environment that is attractive to researchers. The primary mission of JEMRMS is to replace P/Ls and exchange of ORUs on EF and ELM-ES. These tasks are performed by the Main Arm (MA) and the Small Fine Arm (SFA).

A malfunction or improper operation of the robotics system might cause a collision against JEM structures such as PM, or inadvertent release of a P/L or ORU. These hazards are identified as catastrophic hazards because they might result in loss of crew members or ISS. Very strict safety

requirements are imposed on the H/W and S/W design to prevent them. The first part of this paper summarizes how these requirements are implemented and verified in JEMRMS design.

In the current concept of JEMRMS design, on-board crew will still be needed to ensure JEMRMS safe operation by monitoring telemetry data and arm movement or sending commands. In the near future, however, the operation from a ground site is expected to reduce crew load and to conserve IVA resources. NASDA has conducted the preliminary Ground Commanding (GC) experiment, a sort of robotics operation from the ground, as a part of the MFD mission aboard the Space Shuttle. The latter part of this paper summarizes the safety concept of MFD mission and introduces unique safety implementation of GC experiment.

2. OVERVIEW OF JEMRMS

JEMRMS consists of the Main Arm (MA), Small Fine Arm (SFA) and RMS Console.

The MA has six degree of freedom and is approximately 10m long. It has an End Effector (EE) and two vision systems, one is installed on the wrist joint and the other on the elbow. The primary mission of the MA, whose base mechanism is fixed on the PM end cone, is to replace large P/Ls (1.85*1*0.8m) and handle JEM elements such as the EF to back up the Space Station RMS. The MA overview is shown in Figure2-1.

The SFA has six degrees of freedom and is approximately 1.5m long. It has the Tool as an End Effector, force torque sensor and a TV camera on the tool. The Tool has three fingers to grapple an ORU by opening fingers and a torque

drive mechanism to screw or unscrew bolts. The SFA mission, which is attached to the tip of MA, is to perform dexterous tasks such as ORU replacement. The SFA overview is shown in Figure2-2

The performance of the MA and SFA is shown in Table 2-1.

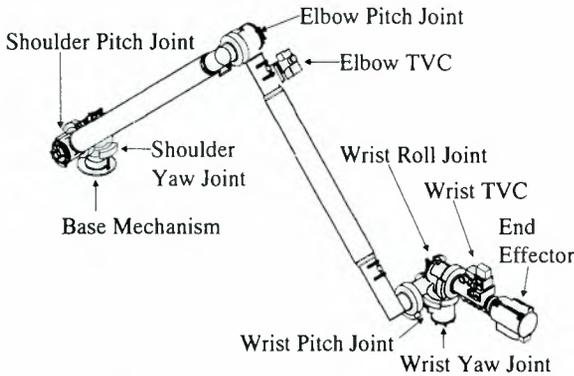


Figure 2-1. Main Arm overview

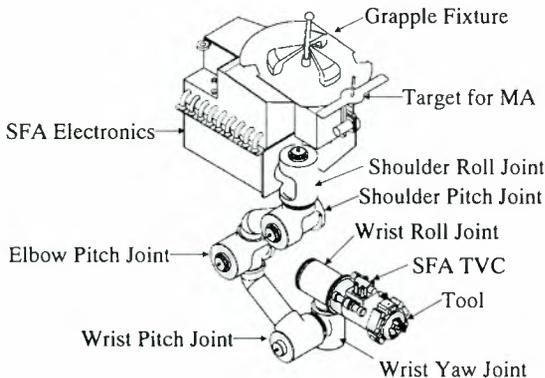


Figure2-2. Small Fine Arm overview

The arms are controlled by two computers in the RMS console. These computers communicate with six Joint Electronics Units (JEU) for MA and SFAE (SFA electronics). One of the computers is the JEMRMS main computer and is called Management Data Processor (MDP). The other computer is the Arm Control Unit (ACU) which controls arm movement while communicating with the JEU and SFAE. These computers have almost the same performance as shown in Table2-2. MDP sends an arm tip position command to the ACU. In manual mode, the ACU generates the Frame of Resolution velocity command in proportion to the voltage input from the Hand Controller by a crewmember. The ACU processes it by resolving inverse kinematics and generates each joint angle and angle rate. The ACU sends the angle commands to the JEU and then six JEUs control the servo of each joint. The ACU sends

angle rate commands to SFAE and then SFAE controls the servo of six joints.

Table2-1. Payload Handling Performance of JEMRMS

	MA	SFA
Maximum payload weigh	7000kg	300kg
Maximum payload inertia	20000kgm2	30kgm2
Maximum payload size	4.5m*6m Dia.	0.8m*0.6m Dia.
Maximum c.g.offset	2m	0.3m
Max. Translation Speed	60mm/sec	50mm/sec
Max. Rotation Speed	2.5deg/sec	7.5deg/sec
Position Error	<±50mm	<±10mm
Attitude Error	<±1.0deg	<±1.0deg

Table2-2. Computer performance of JEMRMS

	MDP/ACU
CPU	2CPU(MQ80386)
FPU	2FPU (MQ80387)
Clock	25MHz
Throughput	3.8 MIPS
System ROM	128KB
Main memory	4MB
EDAC	1bit detection/correction per every 8bits

3. JEMRMS SAFETY APPROACH

To ensure safe design, the following steps are taken:

- (1) Identify potential hazards in JEMRMS
- (2) Identify the control and verification method for each hazard
- (3) Verify each control by analysis or test

Figure 3-1 shows the safety-related schematics of JEMRMS to help readers understand the following descriptions.

3.1. HAZARDS

Typical manipulator hazards are the accidental release of objects and collision against other structures.

JEMRMS has the same potential hazards and identifies the following hazards as catastrophic, Severity I .

- (1) Inadvertent release of objects such as P/L during berthing or unberthing.
- (2) Collision against EVA crew or the structures such as the PM.

Collision against EVA crew members is not covered in detail in this section because safety is assured by established procedures such that power is never supplied to arm motors during co-operation with EVA crew members. In addition, collisions against structures are identified as catastrophic only when a structural failure could result in a floating object in space, space debris.

3.2. REQUIREMENTS

The JEMRMS must adopt a two fault tolerance (2FT) design for catastrophic hazards. Therefore, JEMRMS must remain safe after two mis-operations by crew, two failures of system, or one mis-operation and one system failure. In addition, three independent inhibit are required where inadvertent operations could result in catastrophic hazards, and at least two of the three inhibits status must be monitored by a crewmember.

The safety requirement for inadvertent release requires confirmation of three independent grapple statuses.

3.3. JEMRMS SAFETY DESIGN

The JEMRMS operation sequence is roughly divided into three phases. First is the maneuver phase in which the JEMRMS maneuvers in the Non-proximity area far enough from the EF or other structures to stop safely. Second is the approaching phase in which the JEMRMS approaches from the pause position to the final target position. Third is the berthing phase, an example is the cooperation with the Equipment Exchange Unit on EF (EEU-EF) to transfer a P/L grappled by EE to EEU. The following paragraphs describe how safety is achieved in each phase.

3.3.1. Maneuver phase in Non-Proximity Region

In this phase, collision hazards might occurred due to electrical or electromechanical failure affecting arm control, improper operation by IVA crew, mechanical failure such as galling, joint brake failure, or failure in the control path. However, at least one of the functions below works for any two combinations of failures. Basically 2FT design is assured by JEU or SFAE, ACU and MDP.

(1) Detection by JEU/SFAE

When JEU detects its own failure, it issues the brake on and servo off commands to itself. When SFAE detects one joint failure, SFAE will issue the brake on and servo off commands to all joints. JEU and SFAE report the error detection to ACU.

◆Check of Sensor data and Command

Each MA joint has two joint angle sensors (encoders) and a joint motor axis angle sensor (resolver). Each JEU Firm Ware (F/W) checks the followings using these sensor:

- Cross checks joint motor axis angle sensor and primary joint angle sensor.
- Cross checks primary and redundancy joint angle sensor.
- Checks continuity of joint angle command from ACU.
- Cross checks joint angle command from ACU and joint motor axis angle sensor.
- Cross checks joint angle command from ACU and

primary joint angle sensor.

- Checks limit of joint angle by F/W and the limit switch
- Checks motor current limit.

Each SFA joint has two joint motor axis sensors. The primary sensor is an incremental encoder, and the redundant sensor is a hall device. SFAE F/W checks the followings using these sensors.

- Checks continuity of motor axis angle rate command and joint angle.
- Cross checks primary and redundant joint angle sensors
- Checks limit joint angle, angle rate and a force torque sensor

◆Watch Dog Timer

All checks above are performed by F/W running on SIOP and the Motor Control Processor. WDT is provided to detect anomalies of processor and F/W.

(2) Detection by ACU

If the ACU detects at least one malfunction in the following safety-related functions or detects the error status from JEU or SFAE, the ACU will issue the Emergency stop (E-stop) command to the Power Distribution Box (PDB) to cut-off the power to motors. In addition, the JEMRMS adopts the negative-actuated brake mechanism for fail safe design. The ACU reports the error detection to the MDP.

◆Region check

Region check area can be set with a maximum 10 by 10 mesh to cover the whole surface of the PM end-cone and EF including P/Ls and ORUs. ACU Software (S/W) detects the invasion of the arm tip or other reference points into the region check area by calculating the motor axis sensor data from the JEU and SFAE. The example of region check area is shown in Figure3-2.

In manual mode, an additional region check area about 80mm outside of the above area will be set. If the ACU S/W detects the invasion of the arm tip or other reference points into this region, the ACU S/W restricts commands from the Hand Controller driving the arm in the direction of invasion but allows commands in the opposite direction. In this case, ACU will not send E-stop command. This additional region check function is allocated to only the ACU.

◆Check of sensor data and command

The ACU S/W cross-checks the command and the status of the arm tip position and attitude, and checks limit of the arm tip velocity and limit of the arm tip trajectory error.

In addition, the ACU S/W checks the following using sensor data from the JEU during MA operation:

- Checks limit of the expected motor axis angle data.
- Checks limit of joint angle by S/W and limit sensor.

- Cross checks motor axis angle command and motor axis angle data.
- Check limit of motor axis angle command compared with motor axis angle rate.
- Cross checks primary and redundant joint angle data.
- Cross checks of primary joint angle data and motor axis angle data.

The ACU S/W checks following using sensor data from SFAE during SFA operation:

- Checks limit of expected motor axis angle data.
- Checks limit of joint angle by S/W and limit sensor.
- Cross check motor axis angle command and motor axis angle data
- Checks limit of motor axis angle command compared with motor axis angle rates
- Cross checks of primary and redundant joint angle data.

◆Communication error check

The ACU communicates with the JEU and SFAE via the Arm bus using a MIL-STD-1553B Bus interface. The Arm bus is a redundant bus. When the ACU, Bus Controller, detects a communication error, the ACU sends E-stop command. Crewmember will be able to re-start arm operation after manually switching the bus.

◆WDT

All checks above are performed by S/W running on two CPUs. WDT is provided to detect CPU or S/W anomalies.

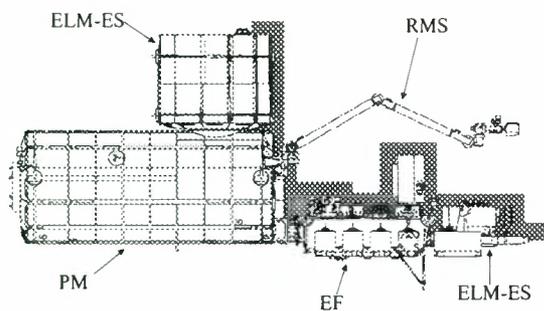


Figure3-2. Example of region check area

(3) Detection by MDP

If the MDP detects at least one malfunction in the following safety-related functions or detects the error status from ACU, the MDP will issue the E-stop command to the PDB to cut-off the power to motors.

(i) Region check

The region check area independent of ACU can be set in the same manner as ACU. MDP S/W detects the invasion of arm tip or other reference points into region check area by calculating the joint sensor data from ACU.

(ii) Check of sensor data and command

During MA operation, MDP S/W checks the following:

- Cross-checks between joint motor axis angle sensor and redundant joint angle sensor
- Checks limit of joint angle

During SFA operation, MDP checks the following:

- Cross checks between primary and redundant motor axis angle data

(iii) Communication error check

MDP communicates with the ACU via the Console Bus using a MIL-STD-1553B Bus interface. Console bus is redundant bus. When MDP, Bus Controller, detects communication error, MDP switch the bus automatically. When the communication does not recover even after switching the bus, MDP sends Emergency stop command.

(iv) WDT

This function is same as ACU.

(4) Detection by IVA crew

IVA crewmember will monitor the arm data on the RLT (RMS Laptop terminal: Thinkpad 760XD) and actual arm motion on TV Monitors to ensure safe arm operation. IVA crewmember can send brake command manually from the Remote Interface Panel (RIP) when he or she detects an anomaly. Basically, however, the JEU, ACU and MDP can control all hazards described in this section before IVA crewmember acts. Therefore anomaly detection by IVA crewmember is not identified as the control path for hazards but just as a redundant path.

3.3.2. Approach phase in Proximity region

When approaching the berthing mechanism in the proximity region, JEMRMS could collide against the berthing mechanism because there is not enough distance to stop safely even if the JEU, ACU or MDP detects the failure, which results in uncommanded motion and sends E-stop commands. The delay time for detecting an anomaly and initiating safing determines the impact energy. The worst-case impact energy is calculated based on the maximum velocity for the worst case delay time when two failures occur simultaneously. Therefore, to achieve two-fault tolerant design, the following method are adopted:

- (1) Calculate the maximum impact velocity and the impact load by worst-case (two fault case) analysis.
- (2) Confirm that the maximum impact load is within the structural allowable level and that structures will never fail.

3.3.3. Berthing phase

When MA berths a P/L to an EEU on the EF or ELM-ES,

or when SFA hands an ORU to attachment mount on EF, ELM-ES or Airlock Table, a payload may be inadvertently released by:

- False indication of grapple mechanism,
- Inadvertent actuation of release mechanism, or
- Mechanical failure of grapple or release mechanism

The JEMRMS, thus provides three independent grapple statuses and two inhibit switches on the power supply line to meet the requirements. This requirement also applies to co-operating mechanisms such as the EFU and ORU attachment mount. Example of typical MA and SFA operational cases are given.

(1) EE grapple status

Before releasing a P/L from EEU, IVA crew member must confirm the following three grapple statuses.

- EE micro switch status: Confirm the capture status on the RLT using signal from EE micro switches which indicate capture, snare-closed and rigidization.
- Side view on TVM: Confirm there are no gaps between the EE and the surface of the Grapple Fixture (GF) in the TVM images captured by one or two exposed cameras. 80% of the EE edge ring should be monitored.
- Target view: Confirms the criteria given by the overlay displayed on the images of the GF target from wrist TVC.

(2) TOOL grapple status

An ORU is fixed on the attachment mount with two bolts. Before unscrewing the second bolt, a crewmember must confirm following three grapple statuses.

- Micro switch status of Tool Latch Mechanism: Confirm the finger open status and the latch status on RLT. The finger open status is generated when three micro switches indicate "open" and two micro switches indicate "not closed".
- Two Side views on TVM: Confirm that two out of the three visual cues on Tool indicate latch completion on TVM and that there is no gap between Tool and the surface of Tool Fixture.

(3) Release command

The release command to EE and Tool adopts the same safety concept. Therefore, the following three independent actions based on two independent information sources are taken to meet the two fault tolerant requirement. Before releasing a P/L, crewmember must confirm three captured statuses from the cooperating mechanism. Crewmember, then, sets two power enable switches from the Remote Interface Panel (RIP) and monitors each status on the RLT. Crewmember, then sends the release command from the Rotational Hand Controller (RHC) after confirming that the

RHC status is "HOT" on the RLT. This active status is generated from the two power enable statuses. The release command is sent to each motor driver after the prerequisite check in the MDP S/W.

3.4. VERIFICATION OF HAZARD CONTROL

All hazard control methods are verified by analysis or PFM testing during the JEMRMS development. However the safety design in the berthing phase will be demonstrated in the JEM overall system test.

4. MFD SAFETY APPROACH

The MFD took a different safety approach from the JEMRMS to achieve two-fault tolerant design. This section gives the basic concept of the MFD safety design and then introduces the unique safety approach in GC experiments that NASDA and NASA took.

4.1. Mission overview of MFD

The mission of MFD project was to demonstrate the prototype SFA (MFD robot arm) functions and performance including the man-machine interface system in a micro-gravity environment, and to feed the results back to the SFA PFM development.

The MFD system consisted of the Shuttle onboard system (MFD payload) and the ground segment. The MFD payload was launched on board STS-85 / Discovery from NASA's John F. Kennedy Space Center on August 7, 1997. The MFD payload consisted of the Payload Bay (PLB) element (Fig.4-1) consisting of the MFD robot arm and other electronics components and the Aft Flight Deck (AFD) element consisting of two 3 degrees of freedom hand-controllers and workstation.

Following the crew-tended demonstrations, file transfer-based Ground Commanding (GC) experiments were conducted as planned using the computer network in JSC (Fig.4-2.) to obtain useful basic data for future space robot arm operations.

All the planned tests and experiments in the MFD mission including GC experiment were accomplished successfully in 12 days. Discovery landed on KSC on August 19, 1997.

4.2. GC Experiment Overview

GC was an advanced technological experiment as well as a preliminary step toward the ground control of a future space robotics system. It showed potential in assisting future Space Station crewmembers so that they could focus their time on the other more important tasks.

The GC experiments were initiated by electronic arm-trajectory file transfers from the ground facility to the Space

Shuttle. These remote control experiments were the first of their kind conducted on board a manned spacecraft with hardware exposed to the space environment. One of the experiments was to repeat a crew-operated robot arm motion on orbit by recreating the crew-operated trajectory in a digital format on the ground. From the arm motion telemetry resulting from prior crew control, an arm-trajectory file was developed on the ground. Since one digital file size was limited to 10 Kbytes due to the system design, an interpolation method was adopted to reduce and adjust the number of pathway points. The trajectory file was then up-linked and executed.

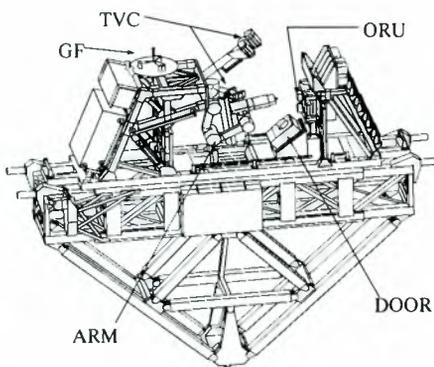


Figure 4-1. MFD PLB element

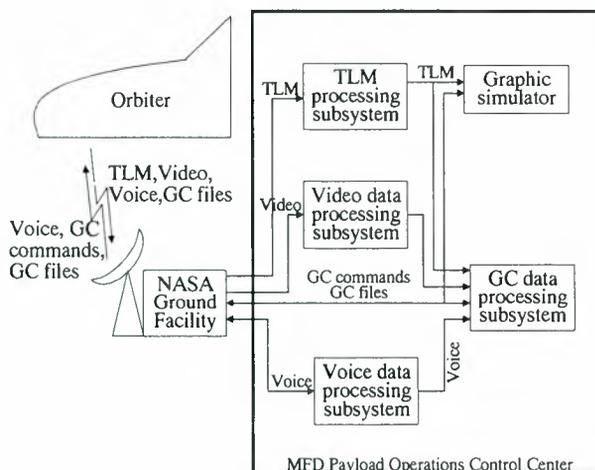


Fig.4-2. MFD GC system

4.3. MFD Mission Safety Concept

The MFD robot arm had the same potential hazards as the JEMRMS. MFD, however, took a different approach than JEMRMS because of its zero-fault-tolerant computer-system. Therefore MFD achieved two-fault tolerant design by assuring safe return of the Orbiter not collision.

(1) To control unplanned contact hazards, the robot arm

reachable envelop was restricted by software and mechanical joint stoppers.

(2) A collision tolerant design was adopted for the robot arm and the structures.

(3) Extra-Vehicular Activity (EVA) compatible design was adopted and flight operation scenarios were developed to stow the robot arm in a safe configuration if it lost the functions due to the collision.

4.4. Unique Approach of GC Experiment

In the GC experiment, the Arm Control Computer (ACC) drove the MFD robot arm based on the arm tip trajectory file. This file was up-linked from the MFD Payload Operations Control Center (POCC) in NASA JSC through NASA data network system. In addition to the major safety-related features above, a step-by-step verification approach was adopted in the GC experiment operations to prevent an arm collision.

(1) To verify the proper command (trajectory) generation, the commands were demonstrated and confirmed one by one by the ground segment prior to up-link.

(2) To detect communication error, the up-linked command was sent back to the ground for validation.

(3) The received and memorized command in ACC was checked using syntax check by ACC prior to arm movement.

In addition, only free-in-motion of an unloaded robot arm without ORU was permitted for safety.

5. CONCLUSION

This paper has presented JEMRMS design and the MFD design concept focusing on safety design. The MFD mission has completed successfully and has been feed-backed many useful techniques and experiences to JEMRMS including safety design. But in the GC experiment, however, crewmember still must monitor the operation to ensure safety. This suggested to us the theme to study how crewmember should be involved with unmanned robotics operation on a manned space facility.

The JEMRMS safety related design, that is the identification of potential hazards and the hazard control, has been approved by the ISS safety panel. JEMRMS was now completed in the design phase and is undergoing the Proto-Flight Model manufacturing, assembly and testing. The identified control will be verified in the series of PFM test. The verification results will also be reviewed and approved at the ISS safety panel in the JEM PQR phase.

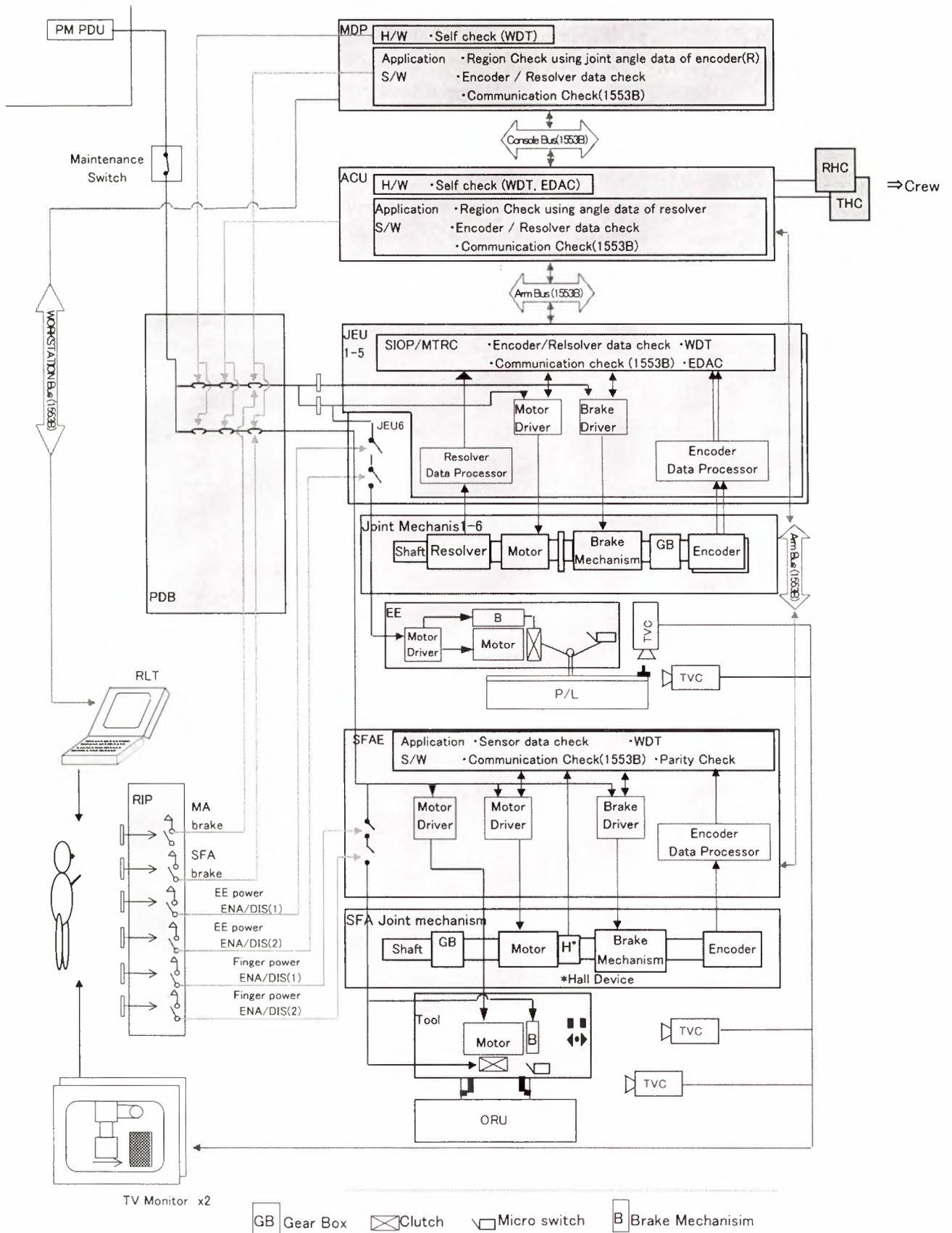


Figure 3-1. Safety-related schematics of JEMRMS

KNOWLEDGE REPRESENTATION AND REASONING FOR FAULT IDENTIFICATION IN A SPACE ROBOT ARM

Luigi Portinale

Dipartimento di Scienze e Tecnologie Avanzate, Università del Piemonte Orientale,
C.so Borsalino 54, 15100 Alessandria (ITALY)
phone: +39 0131-283815, fax: +39 0131-254410, e-mail: portinal@al.unipmn.it

Pietro Torasso, Gianluca Correndo

Dipartimento di Informatica, Università di Torino
C.so Svizzera 185, 10149 Torino (ITALY)
phone: +39 011-6706711, fax: +39 011-751603, e-mail: torasso@di.unito.it

ABSTRACT

The construction of diagnostic systems able to manage tasks like fault detection, fault localization or fault identification in autonomous spacecraft is currently considered a big challenge for Artificial Intelligence techniques. In the present paper we report on the work done inside a project sponsored by ASI (the Italian Space Agency) aimed at building an intelligent multi-agent system for the control and supervision of the SPIDER Manipulation System with some form of interaction with the human operator. In particular, we will discuss knowledge representation and reasoning issues related to the construction of a model-based diagnostic component which has to co-operate with other modules of the system. An in-depth analysis of FMECA documents has guided the modeling of the domain knowledge on the faulty behavior of SPIDER. In this paper, problems related to the choice of the suitable modeling formalism involving abstractions and interaction among components are formally addressed, as well as the definition of innovative diagnostic strategies able to deal with the huge number of possible diagnoses that may arise during the diagnostic activity. The paper reports some preliminary results of the prototypical version of the diagnostic module on simulated data.

1 INTRODUCTION

In recent years, a lot of attention has been paid to investigate perspectives and technical problems involved in the supervision of autonomous spacecraft [10]. In particular, the construction of diagnostic systems able to manage tasks like fault detection, fault localization or fault identification in such spacecraft is currently considered a big challenge for Artificial Intelligence techniques [17]. Indeed, within the mission of Deep Space 1 experiments are scheduled for

testing the functionality of Remote Agent which include planning and scheduling of mission activities as well as fault detection and reconfiguration [4]. More information about the actual experiments is reported at the web site <http://rax.arc.nasa.gov>.

Of course, providing such an autonomy is consequent to an activity aimed at studying and proposing the most suitable formalisms and techniques for solving the above problems. These problems remain very difficult, even when we take into consideration "interactive" autonomy, where some form of interaction with human operator (either on ground or on board) is required.

In the last decade several approaches based on Model-Based Reasoning techniques have been proposed for diagnostic problem solving [11]: many approaches exploit some form of behavioral models of the system under examination (see for example [7]) for detecting and identifying faults. A typical problem in such a case involves how to identify relevant components of the system and their behavior (correct and/or faulty) both in terms of behavioral modes (diagnostic hypotheses) and their observable consequences (symptoms). Another relevant problem is the development of appropriate diagnostic strategies, since it is well known that in the worst case, model-based diagnosis from a computational point of view [1].

In the present paper we report on the work done inside the project *An Intelligent System for Supervising Autonomous Space Robots* sponsored by ASI (the Italian Space Agency), aimed at building an intelligent system for the control and supervision of a spacecraft. The chosen testbed of the project is the robot arm of the SPIDER Manipulation System (SMS) developed by ASI and TecnoSpazio [12]. While other partners of the project are responsible for planning and scheduling [2], image and sensory interpretation [3], interaction with human operator and supervision [9], our group is responsible for developing a diagnostic component able to identify failures and malfunctions of the SPIDER arm. While the diagnostic agent should be autonomous in deriving possible diagnoses given a set of observations about the behavior of the

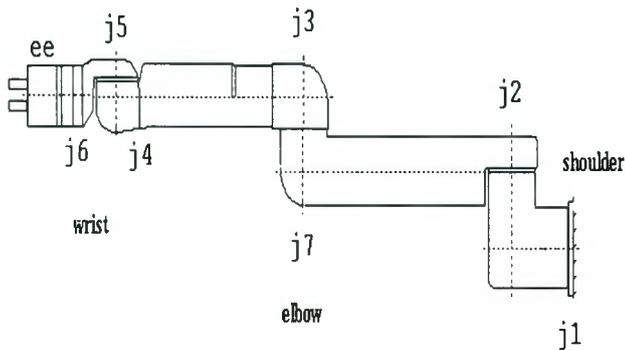


Figure 1: The Spider Arm

robotic arm, it has also to interact with the human operator, by providing him/her with the most plausible diagnoses explaining the observations.

It is worth noting that the diagnostic agent has the goal of detecting, localizing and identifying faults concerning the arm and not of dealing with failure of the plan or activity it currently performs (i.e. plan failures). For this reason, the diagnostic component we have devised is based on the information extracted from system level FMECA (Failure Mode Effects and Criticality Analysis) documents [5], where possible fault modes of SPIDER and their effects are considered and analyzed.

The paper is organized as follows: in section 2 the features of the SPIDER arm are briefly outlined, in section 3 the model of the arm is discussed and in section 4 the diagnostic strategies based on such a model are defined, while in section 5 the notion of abstract model for diagnosing particular faults is introduced; finally section 6 briefly reports about system implementation.

2 OVERVIEW OF THE SPIDER MANIPULATION SYSTEM

The SPIDER Manipulation System (SMS) is a space robot system whose main component is the SPIDER robot arm, a 7 d.o.f (degrees of freedom) robot arm developed by ASI and TecnoSpazio and designed for external space station environment, equipped with a Force/Torque sensor and two sensorized fingers on the end effector. The SMS has been completed and tested in 1998 and its use is expected in some planned missions. It can perform the following tasks: installation and removal of small payload containers on exposure attachment ports, handling of small payloads for scientific and technological investigations, close-up visual inspection of payload units through a camera installed on the arm. All the above tasks can be monitored from ground and no crew intervention is required during nominal operations.

A schematic view of the SPIDER arm is reported in figure 1; the 7 d.o.f. of the arm are obtained by means of 7 joints (j_1 through j_7 in fig. 1), each one equipped

with a position and a thermal sensor. The same kind of sensors are also present at the end effector (ee); moreover, a Force/Torque sensor is positioned in correspondence of the ee to monitor the force applied by the jaws (fingers) on the possible payload. From FMECA documents, the following components (system level items) can be identified; for each joint: the *mechanical part of the joint*, the *electrical part of the joint*, the *harness and electrical connectors for the joint* and the *joint engine*; in addition, the SMS consists of the following components: the *end effector*, the *end effector engine*, the *power supply* and several electronic components, namely the *drive electronics*, the *control electronics* and the *acquisition electronics*. Sensors are not included in the system level items; from the diagnostic point of view this means that we do not model sensor failures (i.e. sensors are reliable components).

Each one of the above components may exhibit, in addition to the normal or nominal behavior, different potential faults whose consequences in terms of observable parameters (usually sensor reports) are described in the FMECA sheets [5]. We explicitly distinguished between observables coming from sensors (*sensorial manifestations*) and other observables that may require more complex operations for getting their value. In the next section we will discuss how the nominal and faulty behavior has been modeled in the diagnostic system we have developed.

3 MODELING THE SPIDER BEHAVIOR

Following the tradition of the model-based approach to diagnosis, we have devised a component oriented model for SPIDER, where each identified component (i.e. the system level items identified from FMECA documents and described in section 2) can assume different behavioral model, one *normal* mode and several *fault* modes [8]. In particular, we adopted a logical approach where each component is identified with a particular predicate, whose admissible values are the behavioral modes of the represented component. For example, the fact that the end effector is in the nominal behavior is modeled by means of the ground atom $ee(normal)$. A set of predicates has also been devised for representing *observable parameters* as well as *contextual information*. Indeed, in the SPIDER domain, the observed behavior of the arm is strictly related to contextual information, usually represented by the particular type of command the arm is executing. For this reason, in the ontology of our model we explicitly consider the presence of *context* predicates. Relationships between components, contextual situations and observable parameters are modeled by means of *definite clauses*¹. The choice of definite clauses allows us to resort to a class of models deeply investigated and widely adopted, which are not too complex from

¹ Actually, we model observable parameters in a slightly more complex way, by associating with them internal system states having observable manifestations. For the sake of simplicity, in the present paper we will make the simplifying assumption that components and contexts are directly associated with observable parameters. This just simplifies the ontology and it is no restrictive at all.

a computational point of view, while preserving a significant modeling power.

The development of a model containing knowledge on both the nominal and the faulty behavior of the SPIDER arm gives us the possibility of solving three different kinds of problems: *fault detection*, *fault localization*, *fault identification*. It is worth noting that the above problems can be solved as far as two assumptions hold: the model is reasonably complete and the discretization of observable measurements into qualitative values captures interesting behavior.

As concerns the completeness of the model, we are confident that the set of behavioral modes for each component is almost complete, since FMECA documents contain a detailed list of faults for each component. More critical is the assumption that the relations between behavioral modes of the components and the observables are accurate. In fact, FMECA does not provide a complete list of the interactions between different faults affecting the same observable parameter, so we had to make some extra-assumptions to model these interactions.

Example 1. Two components of the SPIDER manipulator system are the *control electronics* (*ce*) and the *drive electronics* (*de*); from FMECA documents it is possible to determine that both components influence the *current level* (*curl*) at the drive electronics level. In particular, there are two faults of the above components (fault identified as *sp21060* for *de* and fault *sp23060* for *ce* respectively) that determine an overcrossing of the current limit. Since no other information about *curl* is present in the FMECA sheets, a first choice we have made has been to assume two possible values for the parameter *curl*: $\{normal, high\}$ to model the fact that an overcrossing of the current limit will result in a high current level. Since both *ce* and *de* components have several behavioral modes (one normal mode where no fault is exhibited, 6 fault modes for *de* and 5 fault modes for *ce*), a complete model relating them to the parameter *curl* should take into account all the possible interactions between behavioral modes (in this case 42 interactions).

This kind of problem is not peculiar of this part of the model, but it arises every time that more than one component influences a given parameter. To overcome this model complication, we have chosen to adopt a modeling assumption borrowed from *Bayesian Network* theory: the *noisy-max interaction* [15]. Using the noisy-max interaction, only the influence of each single component (and not of every combination of components) on the given parameter has to be specified: the assumption needed in order to apply noisy-max is that the admissible values of the involved parameter have to be ordered. Given a particular instantiation of the set of components influencing the parameter, the value assumed by the latter is the maximum among the values that are determined by each single component. Returning to our example, we can order the values of the parameters *curl* as $normal < high$. Let now $\{normal, sp21020, sp21030, sp21120, sp21140, sp21160, sp21060\}$ be the behavioral modes (one normal and 6 faulty) of the *de* component and

$\{normal, sp23010, sp23020, sp23140, sp23160, sp23060\}$ be those of the *ce* component; we can model single-component interactions as follows: for the drive electronics

$$\begin{aligned} de(normal) &\rightarrow curl(normal) \\ de(sp21020) &\rightarrow curl(normal) \\ de(sp21030) &\rightarrow curl(normal) \\ de(sp21120) &\rightarrow curl(normal) \\ de(sp21140) &\rightarrow curl(normal) \\ de(sp21160) &\rightarrow curl(normal) \\ de(sp21060) &\rightarrow curl(high) \end{aligned}$$

and for the control electronics

$$\begin{aligned} ce(normal) &\rightarrow curl(normal) \\ ce(sp23010) &\rightarrow curl(normal) \\ ce(sp23020) &\rightarrow curl(normal) \\ ce(sp23140) &\rightarrow curl(normal) \\ ce(sp23160) &\rightarrow curl(normal) \\ ce(sp23060) &\rightarrow curl(high) \end{aligned}$$

By adopting noisy-max, we implicitly model all the interactions between *de* and *ce* on *curl*; for instance the combination $\{de(sp21120), ce(sp23160)\}$ will cause *curl(normal)* since both *de(sp21120)* and *ce(sp23160)* determine this value, while the combination $\{de(sp21120), ce(sp23060)\}$ will cause *curl(high)* since *de(sp21120)* would determine *curl(normal)*, *ce(sp23060)* would determine *curl(high)* and $normal < high$.

As concerns the discretization of observable parameters, it is common for most artifacts to have a range of nominal values and to have range of slight and large deviations. When a parameter exceeds the range of nominal values, some form of alarm arises and the human (or the software) agent has to start some activity, in order to figure out whether something unexpected is actually occurring in the system. In case of SPIDER, fault detection is more complex since it does not depend only on the fact that one (or more than one) parameter has a value outside the nominal range. In particular we recognize that a fault exists (fault detection) if there is discrepancy between what we expect to observe in case all components are OK (that is each component has the normal mode) and what we actually observed. Such expectations depend not only on the mode of the components, but also on the values of the contextual information.

4 DIAGNOSTIC STRATEGIES

As said above, the diagnostic agent has to be able to perform both fault detection and fault identification. By taking into consideration that the model of SPIDER is (almost) complete and the domain theory contains also rules for describing the correct behavior, the first inference step performed by our diagnostic system is a prediction step under the assumption that all components are OK. If there is at least one discrepancy between the predictions and the observations, other reasoning steps have to be activated. It is worth noting that this prediction step is computationally cheap, since it involves deductive closure on a definite clause theory (see [13]).

If a discrepancy exists, the fault identification step has to be activated and the reasoning mechanisms involved in such a step are by far more expensive from a

computational point of view. In component-oriented model-based diagnosis, the notion of partial diagnosis (and of kernel diagnosis) has received significant attention [7] in order to concisely characterize a (potentially large) set of diagnoses. The basic idea behind kernel diagnosis is to include in the diagnosis just the assignment to the components for which the observation imposes some constraints; in other words, a component is not mentioned in the diagnosis if all the behavioral modes of the component are consistent with the observations.

The use of kernel diagnosis does not guarantee at all that the number of kernel diagnoses is small for a given diagnostic problem. In fact, this may happen in the SPIDER domain where for some diagnostic problems, hundred of kernel diagnoses may be generated. This has also the side effect that the time for generating kernel diagnoses can be too high. We noticed this problem by performing some experiments with a first prototype of the diagnostic system; as a consequence we had to move to alternative forms for representing diagnoses in a concise way and for generating them by taking into consideration computation time.

We generalized the notion of abductive diagnosis presented in [6], by introducing the notion of scenario. Let c be a predicate representing the component c having D_c possible behavioral mode (thus the ground instance $c(m)$ represent the component c being in mode m); technically, a *scenario* is represented as a particular kind of *conjunctive normal form* (CNF) formula, where each conjunct is a disjunction of (at most D_c) ground instances of the same predicate c (see [14] for more details).

The basic idea consists in building a representation centered around components, able to capture a number of different diagnoses that involve the same set of components, but which assign different behavioral mode to each mentioned component.

Example 2. Consider the model of example 1; let us suppose that an overcrossing of the current limit at de level is reported (i.e. we observe $curl(high)$). By considering ce and de components, the two following scenarios can represent a set of 14 diagnoses:

$$de(sp21060) \wedge (ce(normal) \vee ce(sp23010) \vee ce(sp23020) \vee ce(sp23140) \vee ce(sp23160) \vee ce(sp23060))$$

$$ce(sp23060) \wedge (de(normal) \vee de(sp21020) \vee de(sp21030) \vee de(sp21120) \vee de(sp21140) \vee de(sp21160)) \vee de(sp21060))$$

Indeed, it is easy to verify that there are 14 different conjunctions of ground atoms of the type $ce(a) \wedge de(b)$ represented by the two above scenarios.

The introduction of a representation based on the notion of scenario has two advantages:

- it reduces the number of diagnoses to be presented to the human operator, so that we reduce the information overflow when the diagnostic problem under examination has a very large space of solutions;
- it provides information about fault localization, since the results are centered around components so that the different faulty modes assigned to a

component are represented in just one structure; this indeterminacy in assigning a unique faulty behavior to that component is then made explicit.

The introduction of the notion of scenario does not mean that the solution of a diagnostic problem is unique, as it was apparent from the example above. Since the notion of scenario is more general than the one of diagnosis, we had to define preference criteria for ranking different scenarios. In model-based diagnosis, a number of criteria have been used for ranking solutions: minimal cardinality, minimality, kernel diagnosis, probabilistic measures. There is a large variety of different preference criteria that can be defined in order to rank scenarios: in particular, in some situations one could prefer quite specific scenarios (for most of the components the scenario indicates just a single behavioral mode), in order to reduce the effort for further discrimination; in other cases one could prefer just to look at faulty components without paying too much attention to the set of specific faults possible for that particular component.

In [14] we have defined a set of preference criteria on scenarios, based on the notion of *minimum description length* or MDL [16], a criterion widely used in Machine Learning for ranking alternative descriptions of a learned concept. The basic idea is to consider a suitable encoding of a scenario and to prefer scenarios having minimum coding length. In particular, useful results have been obtained by considering an encoding where unconstrained components (i.e. components for which every behavioral mode is still possible within the given scenario) are not weighted, and constrained components are weighted proportionally to the number of possible behavioral modes within the scenario and to the prior probability of such modes (see [14] for a more detailed discussion). The adoption of a preference criterion based on MDL is quite relevant, because it can be used not only for ranking scenarios at the end of the fault identification step (that is, for deciding which are the best ones to be presented to the human operator), but also to guide the search process to generate just the most preferred ones.

Since the diagnostic process is in general quite expensive from a computational point of view, the ability of reducing the search space is very important from a practical point of view, even if it cannot guarantee the tractability of all the diagnostic problems.

Let OBS be the set of observable parameters (manifestations or symptoms) to be explained in the current case; the diagnostic search strategy is outlined in figure 2. Some comments are worthwhile:

- the diagnostic strategy is activated by invoking `cover(initial_scenario,OBS)` where `initial_scenario` is the trivial scenario where for each components all the behavioral modes (the normal one as well the faulty ones) are considered admissible and OBS represents the set of manifestations to be explained in the specific case under examination.
- in order to solve a diagnostic problem the inference mechanism considers one observation at

```

cover(current, to_be_expl)
IF to_be_expl = empty_set
THEN
BEGIN
  print("found solution:", current);
  IF no_more_solution_needed THEN EXIT
END
ELSE
BEGIN
  O := first(to_be_expl);
  to_be_expl := to_be_expl - {O};
  expl(O) := explanations of O;
  new_scen := empty_set;
  FOREACH S in expl(O)
  BEGIN
    expl(O) := expl(O) - {S};
    new_scen := union(new_scen, merge(S, current))
  END
  new_scen := heuristic_sort(new_scen);
  FOREACH(S in new_scen) cover(S, to_be_expl)
END

```

Figure 2: Sketch of the Diagnostic Strategy

each step and for the chosen observation the inference mechanism determines all the possible ways such an observation can be explained (in terms of abductive reasoning). Instead of representing the alternative explanations in terms of partial diagnoses (usually a very large set of diagnoses) these explanations are summarized in a (relatively) small number of alternative scenarios.

- The resulting scenarios (i.e. `expl(O)`) are merged with the scenario under examination: the `merge` operation combines the restrictions of the possible behavioral modes for a component determined so far (represented by `current`) with the restriction derived by explaining the current manifestation `O` to be processed (represented by scenario `S`). It is possible that an inconsistency arises, if the behavioral modes of a given component consistent with the manifestations considered so far are not within the set of possible assignments of behavioral modes necessary for explaining manifestation `O`. When an inconsistency arises the scenario is disregarded (i.e. it is not included into `new_scen`).
- the set of scenarios generated by considering manifestations up to `O` are sorted according to the chosen preference criterion (a suitable adaptation of the MDL principle) and the diagnostic process continues by considering remaining observations (the ones not yet considered). As soon as all the manifestations have been considered, the resulting scenario is a solution to the diagnostic problem. The search continues if one is interested to consider alternative solutions to the specific diagnostic problem.

It is clear that the search strategy is essentially a hill-climbing technique and therefore it does not guarantee that solutions are generated in order of preference criterion. However, in the specific case of SPIDER, the adoption of the above strategy resulted to

EF	O1	O4	DO	TO
35.81%	247	249	0.003	0
21.06%	237	247	0.027	0

Table 1: Diagnostic Algorithm: Experimental Results

be very satisfactory as shown by results summarized on table 1. In this table we report the average results concerning two batches of experiments consisting of 250 simulated cases each. Since at this stage of the project we do not have access to real data, cases have been generated by means of a *simulator* we have developed on the behavioral model of the SPIDER arm. Each case is obtained by injecting a particular set of faults and by setting some suitable parameters like for instance the probability of non sensorial predicted manifestations to be part of the actual symptoms of the case. The first line of table 1 concerns a batch with 1 injected fault, while the second line concerns a batch with 2 injected faults. We tested the diagnostic algorithm by setting a time-out of 30 seconds on CPU time (on a Pentium II) and by measuring the following parameters reported in the table: the average expansion factor (EF) representing the percentage of the whole search space (in terms of expanded nodes) that has been visited to find the optimum, the number of times where optimum is the first solution (O1), the number of times where optimum is in the first 4 solutions (O4), the average distance of the coding length of the first solution with respect to the optimum (DO) normalized in $[0, 1]$ with respect to the maximum value, the percentage of time-outs (TO) occurred in the batch. As we can notice the performance of the algorithm appears to be very good, both in quantitative (e.g. EF) and in qualitative terms (e.g. O1, O4 and DO). In particular, it is worth noting that very often the algorithm is able to get the optimum as a first solution (or at least in the first 4); moreover even when the optimum is not obtained as a first solution, the quality of such a first solution is very high as suggested by reported values on DO. Results reported in table 1 refer to just one particular coding function for scenarios, where the contribution of components that, in the given scenario may assume all admissible modes is not weighted; work in [14] reports similar results also for alternative codings.

5 DIAGNOSIS WITH ABSTRACT MODELS

Despite the interesting results obtained by adopting the diagnostic strategy based on the notion of scenario and discussed above, the SPIDER domain has some peculiarities that require the introduction of other reasoning and representation mechanisms in order to supplement the basic mechanisms described above. In order to give a flavor of the problems to be faced, let us consider the case where all the observable parameters related to the joint positions have a qualitative value indicating a large deviation from the expected one. Each single observation can be explained by the mechanical and/or electrical faults of the joint which the parameter sensor is associated to. Since

the abnormal manifestations are related to all joints, all joints have to be assumed faulty (more precisely a huge number of scenarios have to be generated to take into account the possible combination of mechanical and electrical faults of every joint). However, according to FMECA sheets, a fault in the control electronics ce may cause deviations from the expected positions for all the arm joints. It is clear that, in order to explain the above manifestations, it is much more preferable to assume a fault in the control electronics rather than to assume that there are many simultaneous concurrent faults, each one related to a single joint.

Even if in principle such a situation can be dealt with just using a preference criterion, we have preferred to approach the problem by explicitly representing the phenomenon. In particular, we have made use of a notion of *abstraction* both at the level of manifestations that at the level of domain knowledge. As concerns manifestations, we have introduced rules for the synthesis of *abstract manifestations* which summarize the behavior of a number of observed manifestations. For example, to deal with the problem introduced above, concerning an abnormal deviation of every joint position, an abstract manifestation $all_joint_pos(abnormal)$ has been introduced, with the meaning that the position of every joint of the arm is deviating from its nominal value.

As concerns domain knowledge we have derived an abstract model relating the behavioral modes of the components with abstract manifestations. In this way the abstract model shares some portion of the detailed domain knowledge, but it includes clauses specific for the abstract model (e.g. clauses relating faults of the control electronics ce with the abstract manifestation all_joints_pos).

In case of SPIDER, the abstract model is significantly more concise (and simpler) than the detailed model. The diagnostic system is able to work with both the abstract model and the detailed one. The control strategy first tries to activate rules for inferring abstract manifestations. If this inference step succeeds (i.e. at least one abstract manifestation is inferred), the set of observations to be explained is modified by adding the abstract manifestations and by deleting the detailed manifestations subsumed by the abstract one. The fault identification process is activated and the abstract model is used for finding the explanations of the observations. If at least a solution exists (represented by one or more scenarios explaining the manifestations) the process can be considered completed and there is no need of invoking again the fault identification process on the detailed domain theory (unless the user explicitly requires this step). On the contrary, a failure in producing a solution by using the abstract model does not mean a failure in the overall diagnostic process. The diagnostic process is indeed activated for an attempt to explain the set of detailed observations, by using the detailed domain theory.

Example 3. Let us consider again the control electronics component ce ; among its faults there are 4 faults (namely $sp23010$, $sp23020$, $sp23140$, $sp23160$)

that, when present, imply a deviation of each joint position. Let j_pos_i be the predicate representing the position of joint i , with $1 \leq i \leq 7$; the detailed model concerning ce and the joint positions will have the following clauses for each one of the 7 joints:

$$\begin{aligned} ce(normal) &\rightarrow j_pos_i(normal) \\ ce(sp23010) &\rightarrow j_pos_i(abnormal) \\ ce(sp23020) &\rightarrow j_pos_i(abnormal) \\ ce(sp23140) &\rightarrow j_pos_i(abnormal) \\ ce(sp23160) &\rightarrow j_pos_i(abnormal) \end{aligned}$$

Moreover, it follows from FMECA that abnormal positions can result from specific faults (namely $sp11170$, $sp11150$, $sp11030$) of the mechanical part of a joint; let j_i be the predicate modeling this component (i.e. the mechanical part of joint i), then the detailed model will also include the following clauses for each joint²:

$$\begin{aligned} j_i(normal) &\rightarrow j_pos_i(normal) \\ j_i(sp11170) &\rightarrow j_pos_i(abnormal) \\ j_i(sp11150) &\rightarrow j_pos_i(abnormal) \\ j_i(sp11030) &\rightarrow j_pos_i(abnormal) \end{aligned}$$

Concerning this part of model, the detailed model will result in a total of 63 clauses.

If we consider now the abstract model, while no difference arises with respect to components j_i , the part relating ce and the joint positions can be abstracted by using the abstract manifestations all_joint_pos in the following way:

$$\begin{aligned} ce(normal) &\rightarrow all_joint_pos(normal) \\ ce(sp23010) &\rightarrow all_joint_pos(abnormal) \\ ce(sp23020) &\rightarrow all_joint_pos(abnormal) \\ ce(sp23140) &\rightarrow all_joint_pos(abnormal) \\ ce(sp23160) &\rightarrow all_joint_pos(abnormal) \end{aligned}$$

resulting only in 33 clauses (28 relating j_i with j_pos_i and the 5 above).

Moreover, in case we observe for each joint i the manifestation $j_pos_i(abnormal)$, we can substitute this set of manifestations by synthesizing it into the abstract manifestation $all_joint_pos(abnormal)$ and, by using the abstract model, we will avoid at this level of abstraction the generation of diagnoses involving components j_i . In fact, the diagnostic process will result in the generation of just one scenario:

$$ce(sp23010) \vee ce(sp23020) \vee ce(sp23140) \vee ce(sp23160)$$

where only the control electronics is involved. In this way, working on the abstract model, the set of diagnoses involving the fault of just one component (ce) is preferred over diagnoses that, in order to account for the observations, have to hypothesize a fault on each of the 7 arm joints.

6 IMPLEMENTATION

The diagnostic system described in this paper has been implemented as a prototypical system in Java (jdk1.2) on a Pentium II architecture running the Windows98 operating system. The system integrates

²Actually the situation is even more complex, since the other parts of a joint (for instance the electrical part) exhibit the same behavior and in the current version of the model, the abnormality of a joint position is actually modeled with two different values, representing a small and a large deviation from the nominal value respectively.

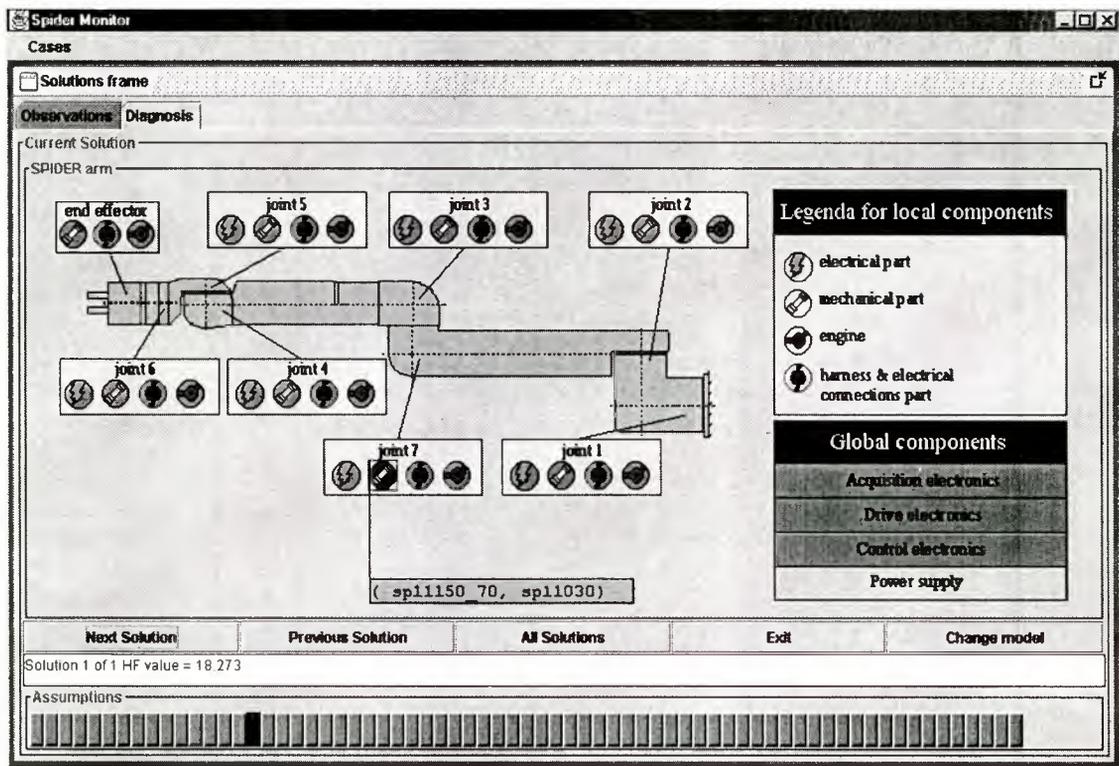


Figure 3: Diagnostic System Interface

the simulator for case generation with the diagnostic problem solver. Cases can be saved into a case library and then loaded for resolution. After a case has been loaded, a window reporting the state of the observed parameters and of contextual information can be displayed; after resolution, results of diagnostic reasoning can also be displayed on a separate window (see figure 3). The system interface allows the user to have a schematic view of the SPIDER arm, where for each joint, all the components (mechanical, electrical, connectors and engine) can be separately considered and the behavioral modes assigned by a given scenario can be displayed. In addition, the status of global components (like electronics units) can be examined by means of a separate set of buttons. The user may then control the generation of diagnoses, by requiring the computation of the next scenario, the computation of all possible scenarios or by changing the model from the abstract to the detailed one.

ACKNOWLEDGMENTS

The research described in this paper has been funded by ASI under grant n. 155/97 and is part of the project "An Intelligent System for Supervising Autonomous Space Robots". Partners in the project are IP-CNR, IRST-ITC, Universities of Genova, Palermo, Parma, Roma and Torino.

References

- [1] T. Bylander, D. Allemang, M. Tanner, and J. Josephson. The computational complexity of abduction. *Artificial Intelligence*, 49(1-3):25-60, 1991.
- [2] A. Cesta, P. Riccucci, M. Daniele, P. Traverso, E. Giunchiglia, and M. Piaggio. JERRY: A system for the automatic generation and execution of plans for robotic devices - the case study of the SPIDER arm. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, iSAIRAS99*, Noordwijk, NL, 1999.
- [3] A. Chella, S. Gaglio, D. Guarino, and I. Infantino. An artificial high-level vision agent for the interpretation of the operations of a robotic arm. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, iSAIRAS99*, Noordwijk, NL, 1999.
- [4] S. Chien, D. DeCoste, R. Doyle, and P. Stolorz. Making an impact: Artificial intelligence at the jet propulsion laboratory. *AI Magazine*, 18(1):103-122, 1997.
- [5] M. Colomba, P.G. Magnani, P. Rigatelli, E. Re, and V. Venturini. Project P0051: System level FMECA. Technical Report BMAS-SM-TS-038, TecnoSpazio SpA, 1997.
- [6] L. Console and P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 7(3):133-141, 1991.
- [7] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197-222, 1992.
- [8] J. de Kleer and B.C. Williams. Diagnosis with behavioral modes. In *Proc. 11th IJCAI*, pages 1324-1330, Detroit, 1989.
- [9] A. Dell'Arciprete, A. Finzi, F. Pirri, G. Rossi, and M. Schaerf. A system integrating high and

- low level planning of complex tasks with a three-dimensional visualiser. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, iSAIRAS99*, Noordwijk, NL, 1999.
- [10] R. Doyle. The emergence of spacecraft autonomy. In *Proc. AAAI 97*, pages 756–761, Providence, RI, 1997.
- [11] W. Hamscher, L. Console, and J. de Kleer. *Readings in Model-Based Diagnosis*. Morgan Kaufmann, 1992.
- [12] R. Mugnuolo, S. Di Pippo, P.G. Magnani, and E. Re. The SPIDER manipulation system (SMS). *Robotics and Autonomous Systems*, 23(1-2):79–88, 1998.
- [13] L. Portinale and P. Torasso. On the usefulness of re-using diagnostic solutions. In *Proc. 12th European Conf. on AI - ECAI 96*, pages 137–141, Budapest, 1996.
- [14] L. Portinale and P. Torasso. Diagnosis as a variable assignment problem: a case study in space robot fault diagnosis. In *Proc. IJCAI 99*, Stockholm, 1999.
- [15] M. Pradhan, G. Provan, B. Middleton, and M. Henrion. Knowledge engineering for large belief networks. In *Proc. 10th Conf. on Uncertainty in Artificial Intelligence*, pages 484–490, Seattle, WA, 1994.
- [16] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):416–431, 1983.
- [17] B.C. Williams and P.P. Nayak. A model-based approach to reactive self-configuring systems. In *Proc. AAAI 96*, pages 971–978, Portland, 1996.

Research and Development of Reconfigurable Brachiating Space Robots

*Yoshiaki OHKAMI, Ryoichi HAYASHI, Hiroshi YAMAMOTO and Saburo MATUNAGA
Mechano-Aerospace Engineering, Tokyo Institute of Technology, Japan*

Abstract

This paper presents a summary of the system of "Reconfigurable Brachiating Space Robot". The robot consists of a center hub and three 6 degree-of-freedom arms with an end-effector and a pivot, which has a reconfigurable mechanism, for each arm. This space robot is capable of moving over the Japanese Experimental Module of the International Space Station in a brachiating manner and also of arm reconfiguration according to the various task requirements. This paper discusses the mechanisms of the arm module, the end-effector and the docking element in detail, and also focuses on the fundamental concept of controllers' system as well as the communication system of the RBR.

1. Introduction

It is very important to develop space robots supporting space activities, especially internal and external vehicular activities for future space utilization. We have started a joint project with NASDA and other Japanese institutes to develop the Reconfigurable Brachiating Space Robot (called RBR) to be tested on the Japanese Experimental Module (JEM) of the International Space Station (ISS). The RBR is designed to make locomotion by grasping handrails and to reconfigure its structural topology in order to have various kinds of functions. This paper gives outline on current research and development status of the RBR, in particular, on the hardware design, the communication and controller design and the experimental demonstration of the RBR mainly conducted by Tokyo Institute of Technology's group.

The objectives of this research study are design, manufacture, test, and demonstrate a space robot with the following:

- i) Modular design of the joints
- ii) Simplification of the harness and the wiring by employing advanced communication method and decentralized control of the joints
- iii) Multi-functional reconfigurable end-effector

Furthermore, the following operational aspects are also considered:

- i) Execution of predetermined tasks according to well-established teaching playback modes
- ii) Improved interaction with human operators

2. System Architecture

As Shown in Figure 1, the RBR system consists of three 6 degree-of-freedom (DOF) arms with an end-effector and a pivot, and a center hub (box) to attach the arms. The hub has three ports (end-effector) to attach arms. Each arm has six revolute joints, one end-effector on its tip and one pivot on the other end. Each arm can be attached to and removed from the center hub as well as another arm and the pivot placed on any point over the space system by the combination of the end-



Figure 1 Conceptual model of Reconfigurable Brachiating Space Robot

effector and the pivot. Thereby the RBR has a variety of compositions with the center hub and the three arms. The end-effector is designed for holding handrails with three fingers with helping of a small TV camera. It also has power and communication connectors mating to the pivot connector. All of the three arms are initially attached to the hub independently; however, its arm-combination can be changed to the another adapted for the given tasks.

3. Major Subsystems

In this section, three major subsystems will be described with specifications. Included are the joint and the arms, the end-effector and pivot and embedded drivers and controller.

3.1 Joint and Arm

A unit of the Joint Module is shown in Figure 2 with specification of Table 1. As shown in Figure 3, one arm is composed of the six Joint Modules in roll or pitch configuration. The basic design criterion is “modularized unit” with integrated electronic devices into a mechanical part. It should be noted that the arm can be operated in a stand-alone mode with a communication controller located at the arm, without a central communication controller located at the center hub. Each of the six Joint Modules has the same components summarized as follows:

i) Mechanical part includes DC servomotor, harmonic drive, rotary encoder and other sensors in a compact form. The major design challenge is to secure ample the space necessary for the harness and wiring of the DC power line and the



Figure 2 Unit of Joint Module

Table 1 Specification of Joint Module

Item	Specification
Drive unit	DC Motor (217W) + harmonic drive (1:120)
Angle detector	3ch Rotary Encoder (1000 CPR)
Joint Torque [Nm]	20.06 (Max39)
Joint Speed [rpm]	21.94 (Max43.88)
Movable area [degree]	-170~+170 (Roll) -120~+120 (Pitch)
Size [mm]	$\phi 92 \times 76.5$
Weight [kg]	1.0

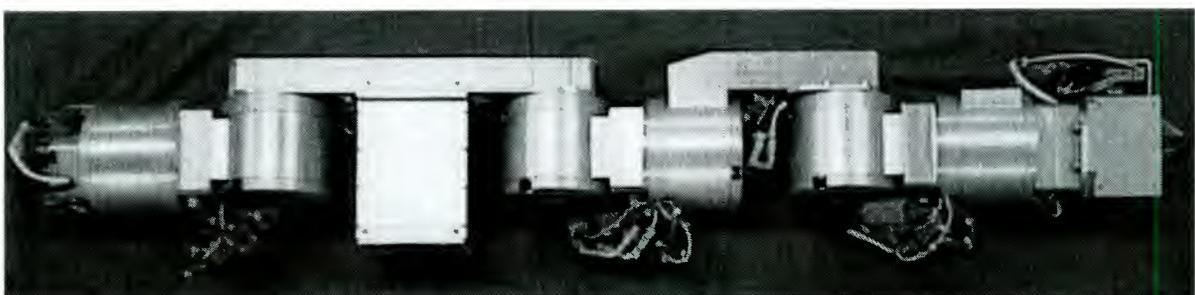
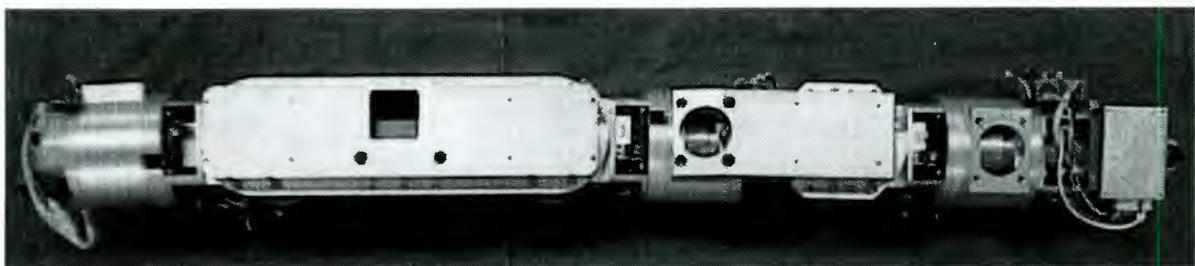


Figure 3 Unit of 6 DOF Arm, End-Effector and Pivot for RBR

communication line for control to the arm and the Joint Module. A hollow shaft of the motor has been utilized at rotating axis for the harness and wiring to pass through this point. This requirement conflicts the downsizing design requirement, both of which must be traded under the currently available state-of-the art technology.

ii) The major characteristics of the electronic part are the motor driver which is named "Device Controller", explained in section 3.3. The Device Controller controls the Joint Module by pulse width modulation (PWM). Mode under the local feedback loop with the rotary encoder and the command signals sent from the communication controller through TIA/EIA-485.

3.2 End-Effector and Pivot

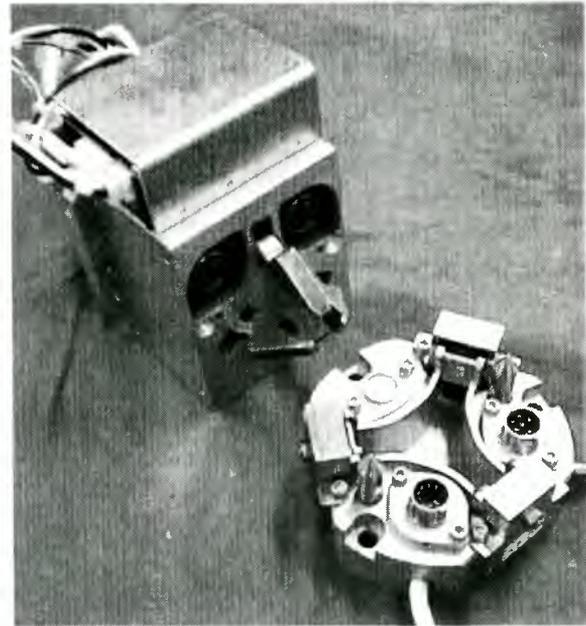
The End-Effector and the Pivot are shown in Figure 4 with the specification of Table 2. Fig.4 (a) is detached configuration and (b) is attached configuration. The pivots are normally placed on the spacecraft or the ISS/JEM Space Experimental Payload (SEP) wall and provide electric power and communication signals to the end-effector through the connector. In addition, the pivot is placed the arm of the tip for connecting to the center hub and is provided electric power and communication signals from the end-effector located the center hub. Both the end-effector and the pivot have the following interface:

- i) Mechanical interface between the end-effector and pivot is required for mating. The mating is realized by radial opening of three claws of the end-effector outward to the extent that it touches the stopper and then by translational drawing action of the claws to secured position of the both units, and both of the electronic connectors is connected.
- ii) In the brachiating mode, the claws of the end-effector are fully open first and then start closing inward and grasp the handrail with one claw at one side and the other two claws at another side. The force is controlled by current measurement on the Device Controller.
- iii) Electronic interfaces are the same as the Joint Module, because the identical Device Controller of the Joint Unit drives the servomotor of the end-effector.

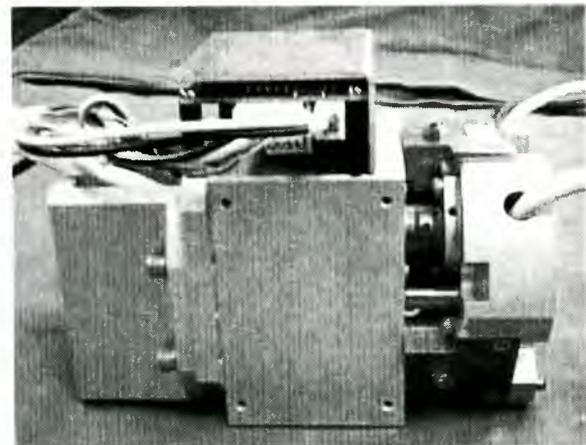
In addition, a miniature-sized CCD camera is installed inside the end-effector that can be used for the precise control of the arm.

3.3 Embedded Driver and Controller

One of the most important key technologies required for realizing reconfigurable robots is the system of distributed the controllers for the joint's



(a) Detached Configuration



(b) Attached Configuration

Figure 4 End-Effector and Pivot

Table 2 Specification of End-Effector and Pivot

Item (End-Effector)	Specification
Drive unit	DC Motor (67W) + harmonic drive (1:100)
Grip Torque [Nm]	5.0
Grip Speed [rpm]	7.0
Size [mm]	$\phi 100 \times 120$
Weight [kg]	1.0
Option	CCD Camera
Electrical Interface	Power Line Ethernet Line

Item (Pivot)	Specification
Size [mm]	$\phi 90 \times 40$
Weight [kg]	0.5
Electrical Interface	Power Line Ethernet Line

motor drivers and joint-to-joint or arm-to-arm communication. The RBR makes reconfiguration of arm compositions, so that it is required to have an autonomous control for each arm working at the moment of reconfiguration as well as any permissible configuration. It is necessary for this requirement to provide the continuous power supply and the information transmission, and be adapted to hot swapping between the distributed controllers of the each arm during attaching and removing operations.

Each arm has a hierarchical system consisting of two layers, shown in figure 5: the top layer is a main controller named Communication Controller, and the other seven sub-controllers named Device Controller for six Joint Module and an end-effector. The Communication Controllers utilize a high-end PC and communication interfaces, as shown in figure 6 with specification of Table 4. Most components used are embedded by PCMCIA, which makes extensions and/or repairing of functional parts by exchanging PCMCIA.

Figure 7 shows the Schematic of the Device Controller. The Device Controller is composed of 16 bit MPU (Hitachi H8) with a TIA/EIA-485 transceiver, PWM drivers for the joint motor and interfaces with sensors including the encoder and the area/limit sensors. Figure 8 shows the Device Controller with specification of Table 3. The size of the Device Controller is small enough to be installed in a mechanical adapter of the Joint Module. The Device Controller communicates with the Communication Controller through the TIA/EIA-485 line. Each arm has the identical

system and communicates with each other using Internet Protocol (IP) through 10BASE-T line. The Communication Controller performs a high-level control while the Device Controllers execute a low-level control of the Joint Module. The high-level control is calculation to a trajectory of a brachiating and reconfiguring motion or any tasks by inverse kinematics and analysis that the arm interference with each other. The low-level control is a local control of the Joint Module, which is an angle and/or angular velocity of the joint, a current of the motor and judgement of the area/limit sensors. The angle, the angular velocity and the current of the motor is controlled via proportional and differential control (PD Control). The control command and the parameter are set through the TIA/EIA-485.

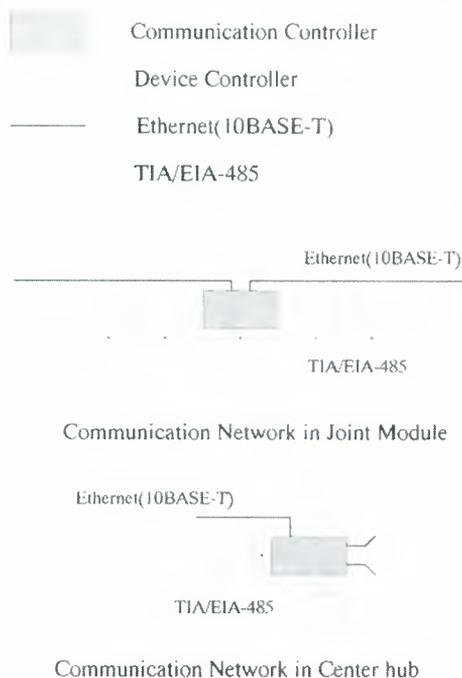


Figure 5 Schematic of Network System of RBR

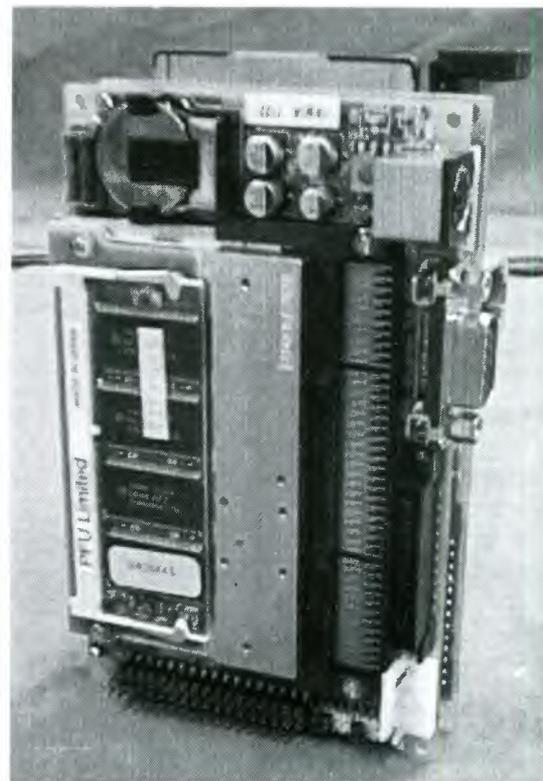


Figure 6 Communication Controller

Table 4 Specification of Communication Controller

Item	Specification
CPU	Intel Pentium MMX 200
Frequency [MHz]	200
RAM [MBytes]	64
HDD [MBytes]	Compact Flash 45
NIC	2Ethernet(10BASE-T) TIA/EIA-485
OS	Debian GNU/Linux + RT-Linux
Size[mm]	124×78×44
Weight[g]	310

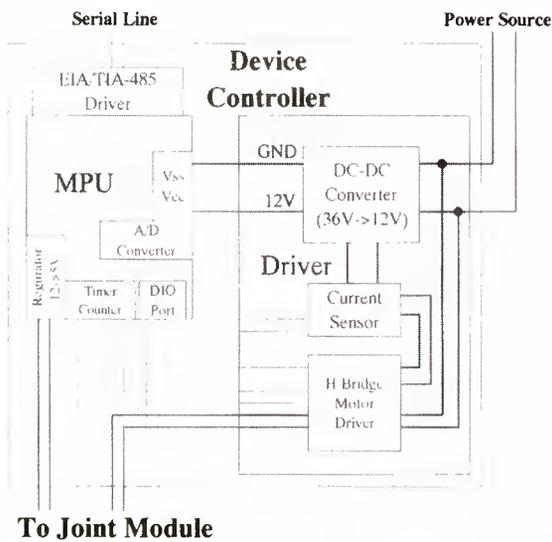


Figure 7 Schematic of Device Controller

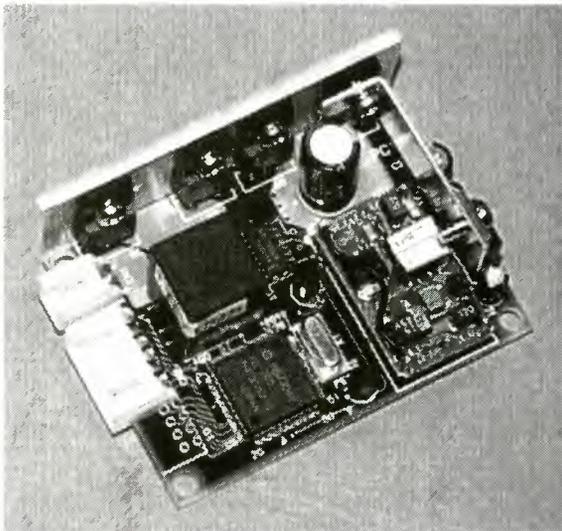


Figure 8 Device Controller

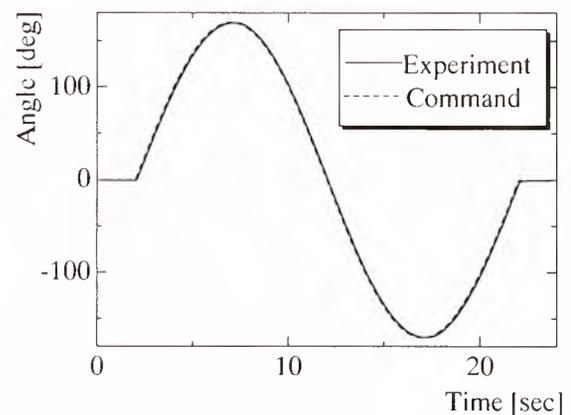
Table 3 Specification of Device Controller

Item	Specification
Motor Power [W]	Max. 288
Input Voltage [V]	18~48
Output Current [A]	2 (Peak 6)
Onboard MPU	Hitachi H8/3048F
Frequency [MHz]	16
Control Period [msec]	1
PWM Frequency [kHz]	32~192
Control Mode	Position Velocity Current
Interface	TIA/EIA-485 (RS-485)
Baud Rate [bps]	38400
Size [mm]	70x50x25
Weight [g]	50

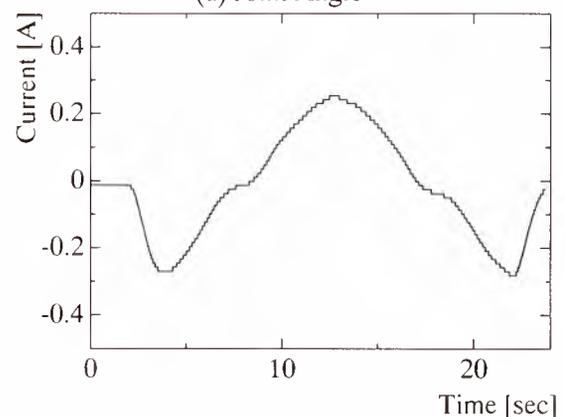
The design philosophy of this control system is modularization and usage of standard off-the-shelf-products. The major advantage of this system is that the number of the cables inside the robot arm is only fourteen: two for the power line, four for the Device Controller line (TIA/EIA-485), four for the Communication Controller line (10BASE-T) and four for spare. The spare cables are used to change the cut wires or extend the communication line to 100BASE-TX. This reduction in the wire number is indeed the key to realization of this RBR system.

4. Preliminary Experiment

We conduct preliminary experiment to verify the performance of the Device Controller. In this experiment, the PD Controller of the joint angle is tentatively installed in the Communication Controller for debugging, so the control period is 10[msec], because of the communication cycle. If PD Controller is moved to the Device Controller, then the control period will be reduced to 1[msec]. The current of the motor driver is measured and filtered by the Device Controller. A cut-off frequency is about 5[Hz]. The experiment is to let the joint angle to follow a sinusoidal command



(a) Joint Angle



(b) Current of Motor Driver

Figure 9 Experimental Data

input with period 20[sec]. Figure 9 shows the experimental data of the joint angle (a) and the current of the motor driver (b). Fig.9 (a) shows that the Device Controller can control the joint angle by PWM. Fig.9 (b) shows that the filter can remove the high frequency noise of the current sensor. In this experiment, we have confirmed the capability of the Device Controller. In immediate future, we will demonstrate the overall capability of the distributed control systems including the seven Device Controllers and one Communication Controller with emphasis on the communication performance.

5. Conclusion and Future Plan

The basic design features and the major characteristics of the Reconfigurable Brachiating space Robot (RBR) have been presented.

This R & D research project started in FY 1997 as part of the Ground Research for Space Utilization and will terminate at the end of FY 1999. During the remaining period of this fiscal year, it is planned to carry out the following experiments and demonstrations:

- i) Brachiating capability in a standard configuration using handrails installed in the 3 Dimension Testbed already installed at TIT.
- ii) Reconfiguration capability with power supply and information transmission for various configurations.
- iii) Joint performance and characteristics in quasi-microgravity condition using parabolic flight in an airplane.

Based upon the results from the experiments described above, onboard experiments will be proposed on the ISS/JEM exposed facility.

6. Acknowledgements

This research work is funded by a part of "Ground Research for Space Utilization" promoted by NASDA and Japan Space Forum. The authors also appreciate cooperation of the members of the RBR design team.

Tele-Science by Planetary Rover : Micro 5

Yasuharu KUNII , Masahiro OTSUKA

Dept. of Electrical and Electronic Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 113-8551, JAPAN.
TEL: +81-3-3817-1866, FAX: +81-3-3817-1847
E-Mail: kunii@hmsl.elect.chuo-u.ac.jp

Masaya SUHARA , Yoji KURODA ,

Dept. of Mechanical Engineering, Meiji University
1-1-1 Higashi-mita, Tama-ku, Kawasaki, Kanagawa 214-8571 JAPAN
TEL: +81-44-934-7183, FAX: +81-44-934-7183
E-Mail: ykuroda@capecod.mind.meiji.ac.jp

Takashi KUBOTA

The Institute of Space and Astronautical Science,
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, JAPAN.
TEL: +81-42-759-8304, FAX: +81-42-759-8304
E-Mail: kubota@cnenet.isas.ac.jp

Abstract

In this paper, we propose the method of constructing the operation environment with the manipulator to gather the sample for the planetary mission, and also describe the development of the 5 Degree-Of-Freedom micro manipulator for the tele-science. Command Distortion Compensation is also presented for Continuous Real-time Tele-driving with the time-delay.

1. Introduction

Recently, it has become easy to construct a cheap, high-speed computer system caused by the high-speed development of the computer technology. As a result, the human interface technology such as the VR technology has been developed rapidly[1][2]. The task operability in the remote environment on Tele-operation has improved by these technologies. However engineers, who have special skills, have



Fig. 1 Planetary Rover: Micro 5

still done almost of actual operations by themselves[3][4]. Especially, in the field of a planetary exploration such as the moon and Mars mission, the scientists have requested their own operations by tele-science equipments to the engineers because the operations require high-level skills and the mission can not be failed. In the nature of things, scientists should really operate these equipments by themselves and then it can be expected that we can obtain more good results. Therefore, there are a

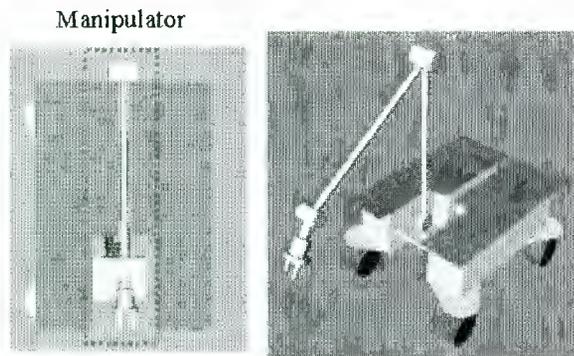


Fig. 2 Image Schematic of 5 DOF Micro Manipulator

lot of demands from the scientist in this point. However an operation environment in a present state is still complex and, moreover, an operation by an engineer is in the situation not avoided because of the existence of the time-delay in the planetary exploration.

In this paper, we propose the method of constructing the operation environment with the manipulator to gather the sample for the planetary mission, and also describe the development of the 5 Degree-Of-Freedom micro manipulator for the tele-science.

2. Planetary Rover : Micro 5

2.1. Micro 5

Developed Micro Planetary Rover : "Micro5" is shown in Fig. 1. Micro5 is driven by five wheels controlled independently. The steering is controlled by differential of left and right wheels. Those wheels are actuated by small DC motors. The velocity of the rover is about 1.5[cm/s]. It has the proposed new suspension system called *PEGASUS* (*Pentad Grade Assist SUSpension*)[5]. So the climb-able step is 0.15[m] and the climbable slope is about 40[deg]. Power is supplied by solar panel on the top of the rover. It's also driven by on-board batteries.

Two CMOS cameras are used as stereo camera for a forward terrain sensor. It also has other cameras around of the body for navigation and scientific

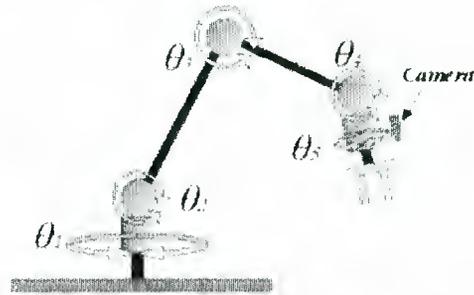


Fig. 3 Structure of Micro Manipulator

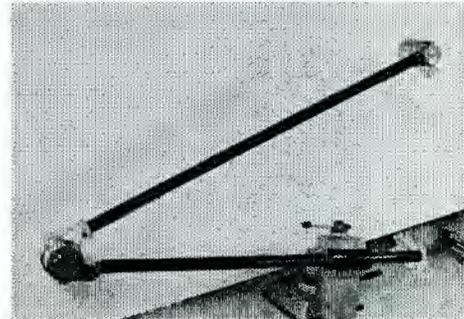


Fig. 4 5DOF Micro Manipulator (under-developing)



Fig. 5 Joint Unit and Actuator (UltraSonic Motor)

observation. The rover is equipped with pitch and roll clinometers for attitude detection and encoders for dead-reckoning. Sensor data processing and control are performed by on board computers, for example, RISC-CPU.

2.2. 5DOF Micro Manipulator

Micro Manipulator is planned to be mounted on Micro 5 (Fig. 2). It has 5 Degree-Of-Freedom (DOF) serial link structure as shown on Fig. 3. It can perform grasping samples, operating some science equipment, scratching sample surface. Moreover, endeffectors based on the mission will be

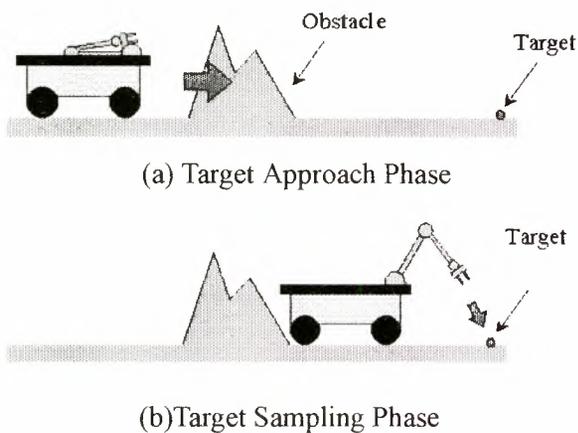


Fig. 6 Two Tele-Sampling Phase

able to be installed on the top. Here, the sample collection mission such as small stones is assumed, and a gripper will be equipped. The gripper has a piezo element as a sensor for the grasping recognition of a sample and a small type C-MOS camera is mounted on the gripper. Each joint is driven by Ultra-Sonic Motor (USM) with Harmonic Drive gear, and all of links are produced in a single structure with a carbon fiber. Here, it is forecast that the manipulator is spent much time for a command waiting state etc. In general, USM adopted on this manipulator can drive by a low electric power and has a big geostationary torque. Therefore, the conservation of electric power can be achieved on this manipulator system. In addition, we have much advantage because CFRP used for the main body has light weight and high strength.

3. Tele-science by Micro5

Let's assume tele-sampling method is composed of two phases. The first is "Target approach phase" in which the rover is going to the close area of a target (Fig. 6 (a)). That is a short range navigation. Another phase is "Target Sampling phase", which is approach to a target truly to grasp it by using an endeffector (Fig. 6 (b)).

3.1. Target approach phase

In general, it is difficult to achieve the remote control of the system with the time-delay caused by

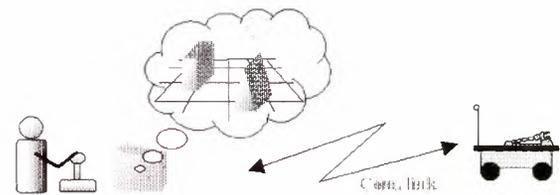


Fig. 7 Direct Tele-Driving System

the communication delay. In such a situation, it is necessary to think about some method to achieve a stable control. Supervisory Control is one of the solutions[3][4]. However, high-level Supervisory Control demands high autonomy, that is, the performance of high calculation power and the sensor. Actually, in space, it is difficult, in many cases, to install an high-performance computer and various sensors due to problems of the harsh environment, weight of equipments and so on. Therefore, high-level autonomy is not expected in the system, but we develop the system based on human direct and continuous control (Fig. 7).

The virtual rover is controlled in the virtual environment created with received data which have been sent from the real rover, and these data is used for the rover control data, as continuous command data or discrete waypoint data (Fig. 9). At this time, the data received by the real rover are generated based on the data which it measured in the past, because of a time-delay. However, the real rover is updating environmental data which are more reliable at this moment. There is a possibility that operator's environmental data and the latest environmental data on the real rover are different, and it is necessary to change data by using the latest environmental data.

3.1.1. Command Data Compensation

It seems that the difference of these data is mainly caused by the error included according to the distance from the rover. In this case, the reliability of data increases by approaching to the remote area.

In this research, this difference is assumed as a

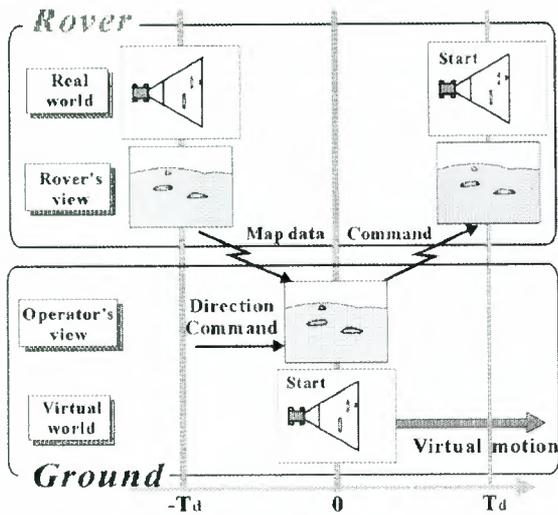


Fig. 9 Time chart of Data Transmission

distortion, and the mapping between old and new data, that is the distortion correction matrix, is acquired. Actually, environmental information is 3 dimensional data. However, it is assumed that the distortion is two dimensions and linear, in an initial stage of the research. And, the distortion compensation algorithm of the camera lens is applied as a method of the distortion compensation.

The Command Distortion Compensation (CDC) transformation is as follow.

$$X = \bar{X}A$$

Where \bar{X} is sampling data of old environmental data, X is the last environmental data, and A is the distortion compensation matrix. Actually, we need only three pairs of the sampling point (X and \bar{X}), if these measurement data have much accuracy. However, measurement data are including noise which is nonlinear. So it is better to measure a lot of point, and then we use a least mean square techniques to obtain a suitable linear solution.

In the first, we have to make Orthogonal-triangular decomposition by using Householder reflections :

$$\bar{X}P = QR$$

where R is upper triangular matrix, P is permutation matrix and Q is orthogonal matrix. Then the least squares approximate solution is given by

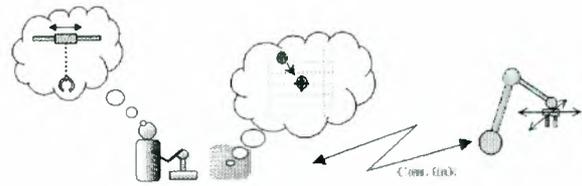


Fig. 8 Tele-Sampling

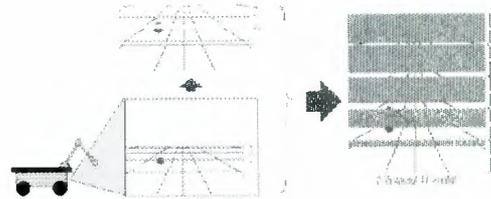


Fig. 10 View Point Transformation

$$A = P (R^{-1} (Q^T X)).$$

Therefore, the command distortion compensation is obtained from

$$Wp_{new} = AWp_{old}.$$

Where Wp_{old} is way-point data matrix sampled data created by operator and Wp_{new} is compensated way-point data matrix.

3.2. Target sampling phase

Target sampling phase is designed by using the above-mentioned manipulator. This phase is basically operated like a X-Y table with a joystick. In the first, we turn a camera and an endeffector equipped on the manipulator to ground. It can be approached to and gathered the target. This operation method looks like the crane game and can be expected that it is easy to accept in general. A detailed operation steps are described as follows.

Step.1: The side-view image which includes the target, and the distance information to the target are acquired by some sensor mounted on the rover, for example the stereo camera, the laser sensor and so on (Fig. 10). These data aren't expected much accuracy in this step.

Step.2: The upper-view image is generated from the side-view image based on distance information. Here, it is composed by using a real

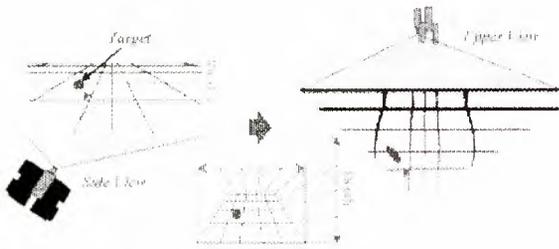


Fig. 12 Example of View Point Transformation

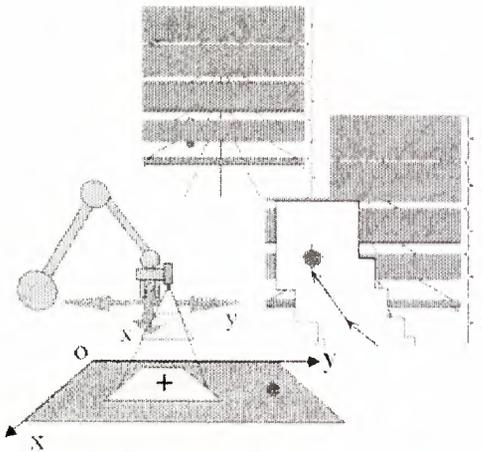


Fig. 13 Motion constrained on X-Y plane



Fig. 11 Target grasping

side-view image and CG(View Point Transformation: VPT, Fig. 10). Fig. 12 is shown a simple example of VPT.

Step.3: The movement of the manipulator is constrained on the X-Y plane, which is a parallel plane to the ground (Fig. 13). The operator is operating a manipulator watching the upper-view image acquired in above step. The generated upper-view image is updated with a real-time real image from a camera mounted on the manipulator.

Step.4: After the target is put in a real upper-view image, the manipulator is guided on the target, that is, the center of the screen (Fig. 11).

Step.5: Next, the direction, to where the manipulator is operated, is constrained to the vertical direction (Z-axis), and it approaches to the target and grasps a sample. Here, a touchdown to the ground is judged by using the shadow, which is coming close to the top of manipulator and also useful for a human-interface.

4. Simulation and Experimental Results

Fig. 15 is shown a simulation result of Command Data Compensation. Here, black boxes are obstacles measured as old environment data and gray boxes are new environment data, and, the trajectory (a gray line) in Fig. 15(a) and (b) are command data created by an operator and a black line in (b) is compensated data. Belts in Fig. 15(c) indicate the width of a rover (black belt: operator command, gray belt: compensated data). It's clearly understood that rover's trajectory is avoiding to run against into objects.

An experimental result is shown in Fig. 14. Image (a) and (b) were measured on the start point, and the point of 140cm from the start point, and are images measured by the Tricrops that has three CCD cameras and is a commercial product. The Tricrops can perform the measurement of depth data. The depth data are obtained on points in circles in Fig. 14(a). Fig. 14(c) is a result of command data compensation. The compensated trajectory could be also avoided from objects

5. Conclusion

The micro planetary rover: Micro5 and its 5 DOF Micro Manipulator were shown. These systems are expected to be launched to the moon in the near future (any mission is not authorized yet). In this paper, we proposed the method of Tele-sampling as an example of Tele-Science. This tele-sampling method is composed of two phases: the first is

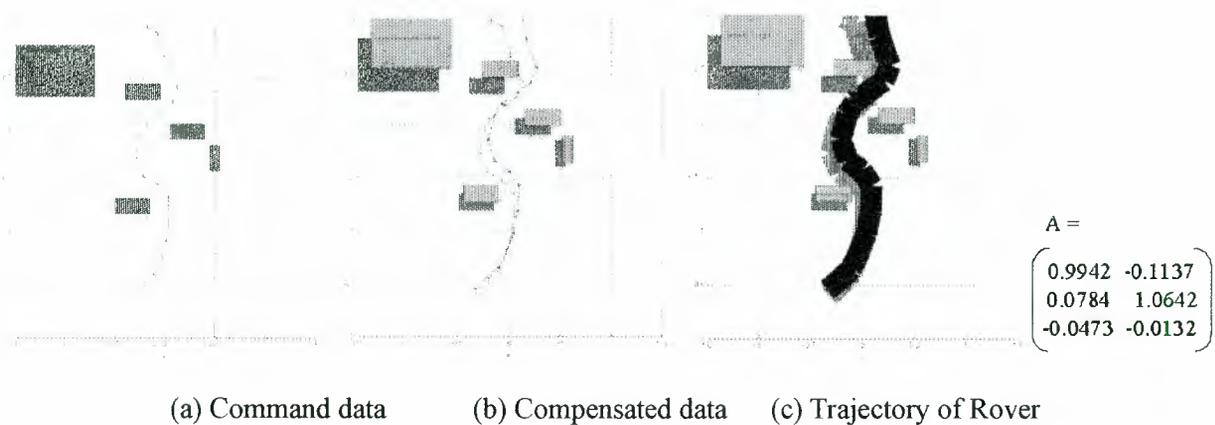


Fig. 15 Simulation Results of Distortion Compensation

"Target Approach Phase" and another is "Target Sampling Phase". In Target Approach Phase, we proposed Command Data Compensation and showed simulation and experimental results. In Target Sampling Phase, we proposed the way to operate a manipulator to grasp a target and it works like a crane game.

Finally, we have a lot of future works, for example, the achievement of the total tele-sampling system and its evaluation. Moreover, nonlinear and 3D command data distortion compensation and the use of shadow information are one of future works, too.

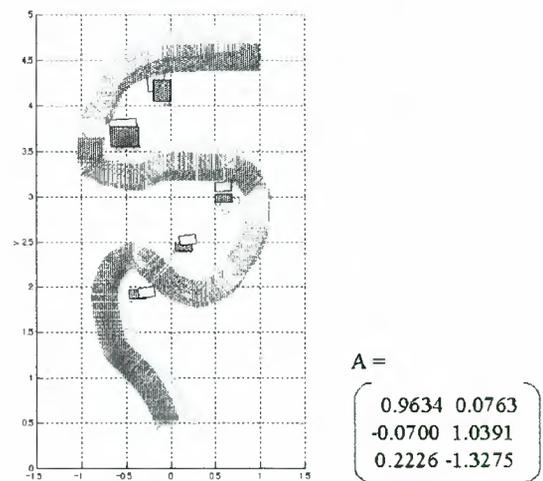
Reference

- [1] S.Tachi: "ARTIFICIAL REALITY AND TELE-EXISTENCE", Proc. of Int. Conf. on Artificial reality and Tele-existence, pp.7-16,1992.
- [2] Karun B.Shimoga: "A Survey of Perceptual Feedback Issue in Dexterous Telemanipulation:Part I. Finger Force Feedback", Proceedings of Virtual Reality Annual International Symposium VRAIS, pp.263--270 ,1993.
- [3] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin, "Mars Microrover Navigation: Performance Evaluation and Enhancement." Autonomous Robots Journal, special issue on Autonomous Vehicles for Planetary Exploration, Volume 2(4), pp. 291-312, 1995.
- [4] S. Laubach and J. Burdick, "An Autonomous



(a) Camera Image

(b) Camera Image



(b) Command Data

Fig. 14 Experimental Results of Distortion Compensation

Sensor-Based Path-Planner for Planetary Microrovers." To appear in Proceedings of the IEEE Conference on Robotics and Automation (ICRA'99), Detroit MI, May 1999.

- [5] Y.Kuroda, K.Kondo, T.Miyata,M.Makino,"The Micro5 Suspension System for Small Long-range Planetary Rover", ISAS 8th Workshop on Astrodynamics and Flight Mechanics, pp.287-292, 1998.

Planning and Scheduling of Space Operations (2)

ORGANIZATIONAL LEARNING AGENTS FOR TASK SCHEDULING IN SPACE CREW AND ROBOT OPERATIONS

○ Keiki Takadama

ATR Human Information
Processing Research Labs.
2-2 Hikaridai, Seika-cho
Soraku-gun, Kyoto
619-0288 Japan
keiki@hip.atr.co.jp
Tel: +81-774-95-1007
Fax: +81-774-95-1008

Hitomi Kasahara

Nara Institute of Science
and Technology
8916-5 Takayama-cho
Ikoma City, Nara
630-0101 Japan
hitomi-k@is.aist-nara.ac.jp
Tel: +81-743-72-5256
Fax: +81-743-72-5259

Linchun Huang

Japan Advanced Institute
of Science and Technology
1-1, Asahidai
Tatsunokuchi, Ishikawa
923-1292 Japan
huangle@jaist.ac.jp
Tel: +81-761-51-1699
Fax: +81-761-51-1116

Masakazu Watabe

ATR Human Information
Processing Research Labs.
2-2 Hikaridai, Seika-cho
Soraku-gun, Kyoto
619-0288 Japan
xmwatabe@hip.atr.co.jp
Tel: +81-774-95-2665
Fax: +81-774-95-1008

Hiromitsu Ii

Univ. of Tokyo
7-3-1 Bunkyo-ku
Tokyo 113-8656 Japan
hiromi@space.t.u-tokyo.ac.jp
Tel: +81-3-3481-4486
Fax: +81-3-3481-4585

Katsunori Shimohara

ATR Human Information
Processing Research Labs.
2-2 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288 Japan
katsu@hip.atr.co.jp
Tel: +81-774-95-1070
Fax: +81-774-95-1008

Shinichi Nakasuka

Univ. of Tokyo
7-3-1 Bunkyo-ku
Tokyo 113-8656 Japan
nakasuka@space.t.u-tokyo.ac.jp
Tel: +81-3-3481-4452
Fax: +81-3-3481-4585

Abstract

This paper explores rescheduling and reorganization abilities of our organizational learning model in the following two important applications in space: crew task scheduling in a space shuttle/station and task planning for truss construction with multiple space robots. Through intensive simulations of the above two tasks, the following experimental results have been obtained: (1) Our model provides good feasible schedules quickly in the case of rescheduling, and it keeps the computational cost for rescheduling low; (2) Plans generated by our model keep or recover efficiency in tasks when robots are added, removed, or exchanged among robot groups; and (3) The integration of (a) learning mechanisms, (b) rule based systems with evolutionary approaches, and (c) multiagent approaches is effective in rescheduling/re-planning problems.

Keywords: crew task scheduling, planning for multiple space robots, multiagent system, organizational learning, learning classifier system

1 Introduction

In space tasks, unexpected situations often occur that avoid experiments or works from going according to schedules or plans. For example, the crew task schedule on a space shuttle/station, a type of job-shop scheduling problem, is often modified due to instrument/crew anomalies, mission changes, or other

schedule change requirements. As another example, pre-determined plans for multiple space robots fail to make sense when one or more robots become failed or inoperative. In the above two cases, new acceptable schedules or plans, even if not optimum, must be obtained as quickly as possible to minimize the time loss. Thus, it is hard in this case to employ conventional methods based on operations research, expert systems, domain-specific heuristic algorithms, or meta-heuristics methods [Osman 96] such as *genetic algorithms* (GAs) [Goldberg 89] or *simulated annealing* (SA) [Aarts 89] for practical and engineering use. This is because (1) the above methods require a lot of time or high computational costs even for small modifications, (2) the methods are difficult to cover all unexpected situations, and (3) even small modifications affect whole systems.

To overcome these problems, recent research on (1) learning mechanisms, (2) rule based systems with evolutionary approaches, and (3) multiagent approaches has studied new possibilities in scheduling or planning domains. For instance, Zhang showed that a reinforcement learning approach found a good feasible schedule more quickly than Zweben's method which is based on simulated annealing [Zweben 94] in the NASA space shuttle payload processing task [Zhang 95]. Since this method can utilize results acquired through the learning, times for making a schedule or a computation costs are reduced. Tamaki showed the generality/applicability of production systems with an evolutionary approach in the case of environmental changes [Tamaki 99], which indicates the potential to cover some unexpected situations. Furthermore, Fujita and Ima

showed multiagent approaches contribute to finding good schedule in a reasonable time in rescheduling problems [Fujita 96, Iima 99].

However, research in these three areas seems to have concentrated on improvements in particular methods or techniques independently, in spite of the fact that these components complement each other. Therefore, this paper employs our model that integrates the above three components from multi-strategic standpoints [Takadama 98a, Takadama 99a] and explores this model's possibility in rescheduling and re-planning problems.

This paper is organized as follows. Section 2 starts by explaining our model, and Section 3 describes two space tasks for scheduling and planning. Section 4 presents our simulations, and the possibilities of our model is discussed in Section 5. Finally, our conclusions are given in Section 6.

2 Organizational-learning oriented Classifier System

Our Organizational-learning oriented Classifier System (OCS) [Takadama 98a, Takadama 99a] is a GBML (Genetics-Based Machine Learning) architecture. OCS is composed of many Learning Classifier Systems (LCSs) [Goldberg 89, Holland 78], which are extended to introduce the concepts of organizational learning (OL) † studied in organization and management science [Argyris 78, March 91, Cohen 95]. Since LCS is equipped with (1) an environmental adaptation function via reinforcement learning mechanisms, (2-a) a problem solving function via rule-based production systems, and (2-b) rule generation/exchange mechanisms via genetic algorithms, and (3) OCS is an extension of LCSs to multiagent environments, it is easily found that OCS includes (1) learning mechanisms, (2) rule based systems with evolutionary approaches, and (3) the multiagent approaches mentioned in the previous section.

2.1 Aim of agent and function

In OCS, agents (jobs of crews or robots in this paper) are implemented by their own LCSs, and they divide given problems by acquiring their own appropriate *functions* through interaction among agents in order to solve problems that cannot be solved at an individual level. Based on this way of problem solving, the *aim* of the agents is defined as finding appropriate *functions*. Furthermore, these functions are acquired through the change of agents' rule sets (*i.e.*, rule base), and thus a *function* is defined as a

rule set. In particular, a rule set drives a certain sequence of actions such as $ABCBC\dots$, in which the A, B and C actions are primitive actions.

Note that the learning needed to acquire appropriate functions in some agents is affected by the function acquisition of other agents. For example, some agents are affected when one of the A, B , or C actions of other agents changes through learning or when the fired order of the A, B , and C actions of other agents changes.

2.2 Architecture

As shown in Fig. 1, OCS is composed of many agents, and each agent has the same architecture, which includes the following problem solver, memory, and mechanisms. In this model, each agent can recognize its own environmental state but cannot recognize the state of the total environment. Note that the component concerning organizational knowledge is not used in this experiment because it is a different component as compared with the three components mentioned in section 1 and because the aim of this paper is to explore the possibility of the integration of these three components.

< Problem Solver >

- **Detector and Effector** change a part of an environmental state into an internal state and change an internal state into an action [Russell 95], respectively.

< Memory >

- **Individual knowledge memory** stores a rule set (a set of CFs (classifiers)) as individual knowledge. In OCS, agents independently store different CFs that are composed of *if-then* rules that have a strength factor (*i.e.*, the worth of rules). In particular, one primitive action is included in the *then* part.
- **Working memory** stores the recognition results of sub-environmental states and also stores the internal state of an action of fired rules.
- **Rule sequence memory** stores a sequence of fired rules in order to evaluate them. This memory is cleared after the evaluation.

< Mechanisms >

- **Roulette selection** probabilistically selects one rule from among plural rules that match a particular environment. In detail, one rule is selected according to the size of the strength attached to each rule. Since each rule includes one primitive action, one action is performed in each roulette selection.
- **Reinforcement learning, rule generation, rule exchange, and organizational knowledge reuse mechanisms** are reinterpreted from the four kinds of learning in OL (Details

†Detailed introduction to the concepts of OL is discussed in [Takadama 99a].

are described later except for the organizational knowledge reuse mechanism).

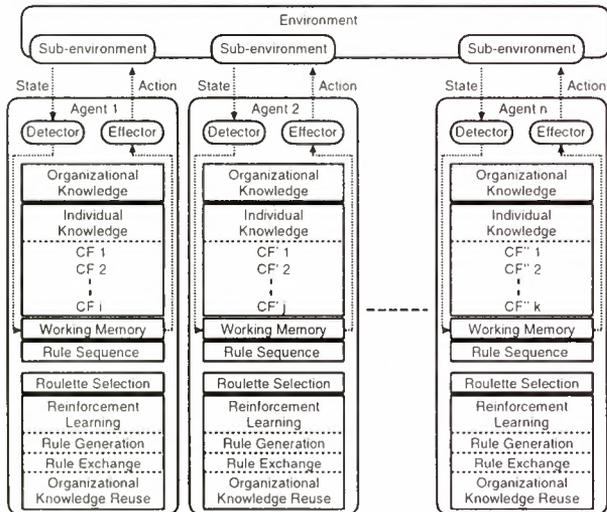


Figure 1: OCS Architecture

2.3 Learning in OCS

2.3.1 Reinforcement learning mechanism

In OCS, the reinforcement learning (RL) mechanism enables agents to acquire their own appropriate actions that are required to solve given problems. In particular, RL supports agents to learn the appropriate order of the fired rules by changing the strength of the rules. In detail, OCS employs a *profit sharing* method [Grefenstette 88], which reinforces a sequence of rules at once when agents obtain some rewards[†].

2.3.2 Rule generation mechanism

The rule generation mechanism in OCS creates new rules when none of the stored rules match the current environmental state. In particular, when the number of rules is `MAX_CF` (maximum number of rules), the rule with the lowest strength is removed and a new rule is generated. In a process of rule generation, the condition (if) part of a rule is created to reflect the current situation, the action (then) part is determined at random, and the strength value of the rule is set to the initial value. Furthermore, if the situation does not change because the same rules are repeatedly selected, the strength of the rules is temporarily decreased and these rules become candidates that may be replaced by new rules.

2.3.3 Rule exchange mechanism

In OCS, agents exchange rules with other agents at a particular time interval (`CROSSOVER_STEP`[‡]) in order to solve given problems that cannot be solved at

an individual level. In this mechanism, a particular number ((the number of rules)×`GENERATION_GAP`[†]) of rules with low strength values are replaced by rules with high strength values between two arbitrary agents. For example, when agents X and Y are selected as shown in Fig. 2, the CFs in each agent are sorted by order of their strength (upper CFs have high strength values), and $CF_{j-2} \sim CF_j$ and $CF'_{k-2} \sim CF'_k$ in this case are replaced by $CF'_1 \sim CF'_3$ and $CF_1 \sim CF_3$, respectively. However, rules that have strength higher than a particular value (`BORDER_ST`) are not replaced to avoid unnecessary crossover operations. The strength of replaced rules are reset to their initial values. This is because effective rules in some agents are not always effective for other agents in multiagent environments.

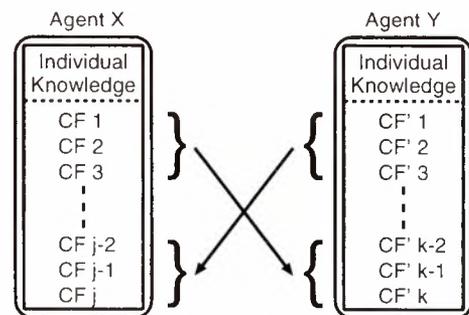


Figure 2: Rule exchange mechanism

2.4 Supplemental Setup

In addition to the above mechanisms, OCS is set up as follows: In the beginning, a particular number (`FIRST_CF`) of rules in each agent is generated at random, and the strength values of all rules are set to the same initial value.

3 Task Domain

3.1 Crew Task Scheduling

3.1.1 Problem Description

In the crew task scheduling of a space shuttle/station, many crew jobs must be scheduled under hard resource constraints. In particular, jobs in this task are components of missions, and they should be assigned while satisfying the following constraints.

1. **Power of space shuttle/station:** Each job needs a particular size of power (from 0% to 100%) in experiments, but the summation of the power of all jobs_k at each time must not be more than 100%.
2. **Link to the ground station:** Some jobs need to use a link in experiments, but only one job can use it at each time. Due to the orbit of

[†]The detail credit assignment in OCS was proposed in [Takadama 98b].

[‡]This step is defined in section 3.1.2 and 3.2.2.

[†]The ratio of operated rules.

the spacecraft, none of the jobs can use the link during a certain time.

3. **Machine A:** Some jobs need to use a machine A in experiments, but only one job can use it at each time. Examples of such machines involve computers, voice recorders, and so on.
4. **Machine B:** The condition is the same as for machine A.
5. **Priority order in jobs:** In a mission unit, jobs have their priority orders (from 1 to the total number of jobs where a smaller number means a higher priority). Jobs in a mission must be scheduled to satisfy their priority orders.
6. **Crew assignment types:** The crew is divided into the following two types: Mission Specialist (MS) and Payload Specialist (PS). The former is mainly in charge of experiments, and the latter supports experiments. In a specific assignment, "the required number of crew members," "the necessary persons," and "the necessary crew assignment types" are decided for each job. For the third element, one of the following crew assignment types must be satisfied: (a) Anybody, (b) PS only (PS is not specified), (c) One specified PS with somebody, (d) One specified MS with somebody, and (e) Combination of PS and MS (PS and MS are not specified). These types are based on the space shuttle missions.

3.1.2 Problem Setting

In this task, each job is designed as an agent in OCS, and each learns to acquire an appropriate sequence of actions that minimizes the total scheduling time. In detail, jobs have 15 primitive actions such as movements for satisfying power constraints, or movements toward an earlier time in a schedule if all constraints are satisfied. Furthermore, jobs can only recognize the situations of their neighbors.

As the concrete problem setting without anomalies, all jobs are initially placed at random without considering overlaps and the six constraints described in the previous section, and therefore a schedule at this time is not feasible. After this initial placement, the jobs start to perform some primitive actions in order to reduce the overlap or to satisfy the constraints while minimizing the total scheduling time. When the value of the total time converges with a feasible schedule, all jobs evaluate their own sequences of actions according to the value of the total time. Then, the jobs restart from the initial placement to acquire more appropriate sequences of actions which find shorter times. In this cycle, one *step* is counted when all jobs perform one primitive action, and one *iteration* is counted when the value of the total time converges with a feasible schedule.

In the case of anomalies, on the other hand, there are two ways of scheduling in OCS: (1) the same way

as a case without anomalies (reschedule from the beginning) and (2) all jobs start from the placement of a current schedule that satisfies all constraints except for the anomaly parts (reschedule from the current schedule). Especially in the latter case, only jobs that do not satisfy constraints due to anomalies change their locations in the schedule, and thus a modified schedule can be obtained quickly.

3.1.3 Index of Evaluation

In this task, the following two indexes are evaluated:

- Goodness = *total scheduling time*.
- Computational cost

$$= \sum_{i=start}^{iteration_in_convergence} step(i)$$

The first index (*goodness*) evaluates a solution of a feasible schedule, and the second index (*computational cost*) calculates the accumulated steps. In this equation, "*step(i)*," "*start*," and "*iteration_in_convergence*" respectively indicate the steps counted in *i* iterations, the start iterations, and the iterations when the value of the total scheduling time converges through repetitions that attempt to find times shorter than the initial placement. This convergence is recognized when the total time shows the same value in some particular iterations. Furthermore, computational costs for repairing anomalies can be calculated by setting *start* to the iterations when anomalies make the schedule change.

3.2 Task planning for truss construction

3.2.1 Problem Description

In the task planning for truss construction with multiple space robots, we employ a robot which has only one arm in order to reduce its weight[†]. This means that each robot can only hold either a beam or a welding tool to combine/weld beams, which are the basic components of a truss. In a concrete truss construction with these robots, an example in the first several steps is shown in Fig.3. In this figure, the black circle with the solid line, the mesh circle, the double circle, and the dashed line respectively indicate a robot with its own beam, a robot without a beam, the space station, and the location for the truss that will be constructed. Note that all robots are supposed to have their own welding tools and thus robots without beams can weld beams by holding welding tools.

- (1) Two robots hold their own beams and go to the beam constructing location.
- (2) Two robots with their own beams arrive at the beam constructing location and set the desired angle between the beams. The robot without a beam goes to the welding location.

[†]In space, it is important to reduce the weight of robots because launching costs are quite expensive.

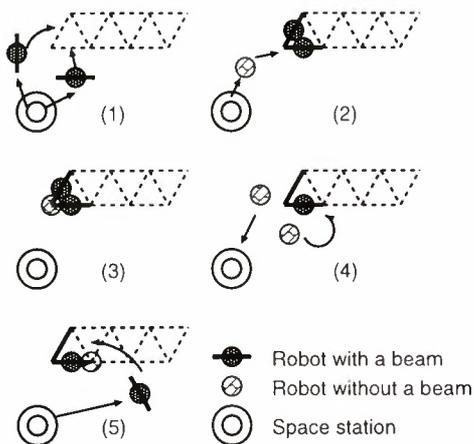


Figure 3: Truss Construction

- (3) The robot without a beam arrives at the welding location and welds the beams.
- (4) After welding, the robot with a beam on the left side returns to the station and the robot that welds the beams goes to another welding location.
- (5) Another robot that has its own beam goes to the beam constructing location.

In addition, robots get into deadlocked situations when either all or none of the robots hold their own beams, or when some robots cannot go to the beam constructing or welding locations because other robots wait in the course of the target location.

3.2.2 Problem Setting

In this task, each robot is designed as an agent in OCS, and each learns to acquire an appropriate sequence of actions that minimizes the truss construction steps. In detail, robots have 11 primitive actions such as holding a beam, or moving toward a beam constructing location. Furthermore, robots can only recognize the situations of their neighbors.

As a concrete problem setting, all robots start at the space station and learn whether they hold their own beams or not. After beams are welded, robots that hold or weld beams learn again whether to go to other welding locations to weld the next beams or whether to return to the station to get other beams. When robots complete a truss construction or get into a deadlocked situation, all robots evaluate their own sequences of actions according to the current situation (completion or failure). Then the robots restart from the space station to acquire more appropriate sequences of actions that take fewer steps. In this cycle, one *step* is counted when all robots perform one primitive action, and one *iteration* is counted when robots complete to a truss construct or get into a deadlocked situation.

3.2.3 Index of Evaluation

In this task, the following two indexes are evaluated:

- Goodness = *truss construction step*
- Task completion rate

The first index (*goodness*) evaluates a solution that is the same viewpoint in the crew task scheduling, and the second index (*task completion rate*) evaluates how robots reconfigure cooperation among robots when there are anomalies. In particular, this rate is calculated as the average of the task completion numbers in a certain range of iterations. Note that the viewpoint of the second index is similar to that of the computational costs for repairing anomalies in the crew task scheduling.

4 Simulation

4.1 Experimental Design

A simulation investigates the rescheduling and reorganization abilities of OCS when anomalies occur. In the crew task scheduling, six types of anomalies shown in table 1 are introduced into a schedule of 10 jobs, and the results of rescheduling from the current schedule are compared with the results from the beginning. Since all constraints cannot be satisfied unless anomalies are removed, feasible schedules cannot be found. For example, a job that requires a link cannot be completed as long as a link is down. From this fact, this paper supposes a certain duration of anomalies. That is, both the start and end times are decided in each anomaly.

Table 1: Type of anomalies

Type	Anomaly	Content
1	Crew sick	A crew cannot perform experiments
2	Power down	The max size of power decreases
3	Link down	A link cannot be used
4	Machine A down	Machine A cannot be used
5	Machine B down	Machine B cannot be used
6	Type 1+2+3+4+5	Integration of 5 anomalies

In the task planning for truss construction, on the other hand, the four operations shown in table 2 are performed after the two robot groups A and B acquire some division of works while keeping their division of work, and the results after operations are compared with the results before operations. In particular, two groups construct their trusses from the same space station, thus they affect each other. Furthermore, each group is composed of five robots. Finally, the robot added in the “addition” operation is a new one which has not yet learned, and the failed robot in the “failure & removal” operation is an inoperative robot that behaves at random.

Table 2: Operation

Operation	Content
Addition	One robot is added to group A
Removal	One robot is removed from group A
Exchange	One robot in group A is exchanged with one robot in group B
Failure & Removal	One robot in group A fails and is removed

4.2 Experimental Results

Table 3 shows both the total scheduling time and the accumulated steps required in rescheduling for anomalies. The values are calculated both from the beginning and from the current schedule after anomalies occur. All results are averaged from five different examples of each anomaly type [†]. For instance, the duration time and anomaly start time of “link down” are different in each example.

Table 3: Total scheduling time and accumulated steps

Type	Total scheduling time		Accumulated steps	
	From the beginning	From current schedule	From the beginning	From current schedule
1	29.4	30.2	241.2	14.0
2	32.2	33.6	557.4	11.8
3	33.6	33.8	700.6	43.4
4	34.6	32.4	1581.8	16.4
5	32.2	29.4	1116.2	13.8
6	35.8	37.2	3204.2	38.0

Next, Table 4 and Fig. 4 respectively show the truss construction steps and the task completion rate, and compare the results before and after operations. Since this paper shows the change in the task completion rate, the values of Table 4 and Fig. 4 are obtained from one result. However, we have confirmed that the tendency of results does not change drastically with other examples or different random seeds. Furthermore, all operations except for “failure & removal” are performed in 117 steps, and “failure & removal” is performed in 117 and 417 steps. As shown in Table 4, the truss construction steps before operations in group A is smaller than those in group B because the location of truss A is nearer the station than that of truss B.

5 Discussion

(1) Rescheduling ability of OCS

The following discussions based on Table 3 suggest that OCS has a rescheduling ability that provides good feasible schedules quickly.

[†]This corresponds to the average of five situations with different random seeds in one example.

Table 4: Truss construction steps

Operation	Group A		Group B	
	Before	After	Before	After
Addition		208		313
Removal		310		310
Exchange	232	235	313	315
Failure & Removal		441		311

- **Total scheduling time** from a current schedule for each anomaly type is almost the same as the scheduling time from the beginning. This tendency does not change with the number of anomalies, even if constraints in a schedule become hard as the number of anomalies increases. Based on the fact that OCS finds good feasible schedules just from the current schedule, OCS has a mechanism for providing the appropriate rules for each job. Since these rules are acquired in just 103 accumulated steps [†] in the case of without anomalies, OCS is effective for practical and engineering use.
- **Accumulated steps** from the current schedule in each anomaly type is much smaller than those from the beginning (even if 103 accumulated steps which are needed for making a schedule in advance are added to the results from a current schedule). This effectiveness increases as the number of anomalies increases. Based on this fact, OCS provides a feasible schedule quickly in the case of rescheduling. Furthermore, this schedule is easy understandable for schedulers because most parts in the original schedule remain.

(2) Reorganization ability of OCS

The following discussions based on Table 4 and Fig. 4 suggest that OCS has a reorganization ability that keeps or recovers efficiency in tasks.

- **Addition** makes the truss construction steps in group A decrease from 232 to 208 steps because OCS enables an added robot to acquire the appropriate actions that are used to cooperates with the five original robots [‡]. This keeps the same task completion rate.
- **Removal** makes the truss construction steps in group A increase from 232 to 310 steps because one robot is removed. This result can be understood by considering the fact that the truss construction steps increase as the number of robots decreases. Since an effective division of work

[†]103 accumulated steps can be calculated in about 3 seconds with a personal computer (Pentium 200MHz CPU).

[‡]Basically, the truss construction steps decrease as the number of robots increases. However, this is based on the assumption that an added robot never fails to cooperate with others appropriately.

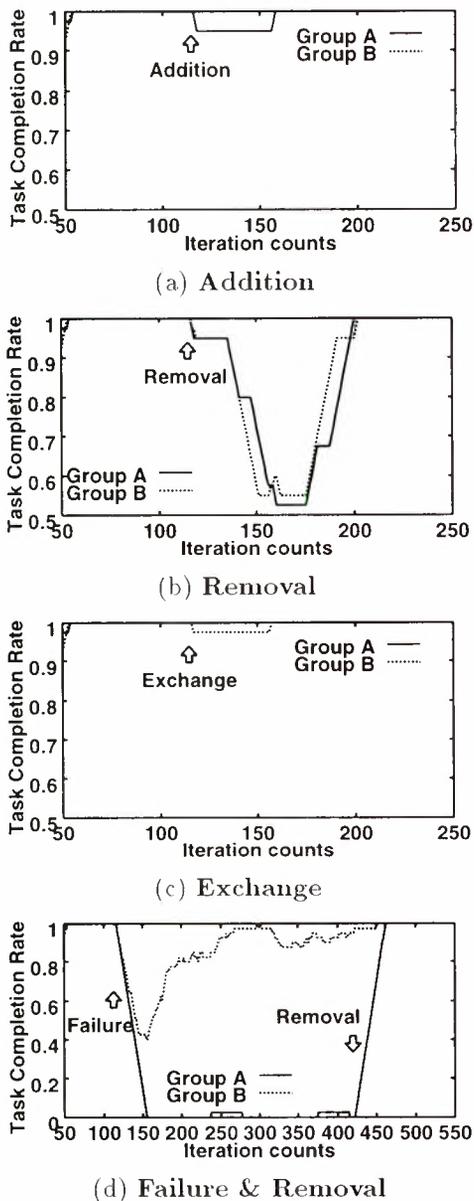


Figure 4: Four operations

in group A is broken by removing a robot, the remaining robots must re-acquire new actions (which means to shift to a new division of work) to cooperate with each other again. This not only makes the task completion rate in group A decrease but also affects behaviors of group B. Thus, the task completion rate of group B decreases and increases according to the change in group A.

- **Exchange** does not drastically change the truss construction steps in group A, because OCS enables the exchanged robots to modify their actions to cooperate with other groups. This keeps the same task completion rate.
- **Failure & Removal** make the truss construction steps in group A increase from 232 to 441 steps due to the same reason as in “removal”.

However, the task completion rate of group B recovers more quickly than that in the “removal” case. This is because group A does not shift the original division of work to a new division of work due to the unfixed actions of the failed robot. Thus, the division of work in group B is not affected much by group A.

However, one may think that OCS is not so useful in terms of fault tolerance because the rate of group A becomes almost 0 after one robot fails and does not recover until the failed robot is removed. However, this rate depends on the design of the actions of the robots. For example, robots do not get into the deadlocked situations mentioned in section 3.2.1 if we design actions that go back to the station to release a beam or other actions that move to another place while a few time. However, we cannot guarantee to design appropriate and indispensable actions in advance, especially in space tasks. Therefore, we must consider cases in which some robot groups confront unexpected situations, and we also must consider how other groups complete the tasks with being affected by those robots that confront unexpected situations. From this sense, OCS has the potential to recover the task completion rate when there are anomalies in some robot groups.

- **Truss construction steps in group B** does not change drastically because group B is not directly operated by the four operations.

(3) Possibility of OCS

In the task of space shuttles, schedulers appropriately assign jobs for each crew. However, there is a limitation to schedulers for space stations in which crews from different countries perform many experiments. This is because (1) there are more crew members on a space station than on a space shuttle and (2) experiments can be performed through 24 hours according to the time zone of each country. This situation obviously causes unexpected anomalies frequently. Even in such a case, OCS proposes good feasible schedules quickly. Furthermore, crews sometimes want to change constraints like job order because they know their jobs best, and these kinds of requirements occur asynchronously. In this case, OCS also provides this chance just by allowing crews to set their preferences for job constraints[†]. In particular, this property of OCS leads to effective coordination between crews and schedulers. At least, the hard work of schedulers is reduced to some extent.

In addition, cooperation among countries is indispensable in space stations. However, this is often

[†]Even if crews set their own preferences, OCS does not always satisfy these because the main aim of OCS is to improve total (organizational) performance according to the concept of organizational learning.

difficult because (1) jobs for each country are scheduled by each country's scheduler and (2) a sudden change in schedules affects other schedules, especially when the same instruments are used. Even in such a cases, OCS provides schedules that recover efficiency in jobs.

(4) Integration of three components

Although the effectiveness of OCS is shown through the above discussion, one may wonder if all three components ("learning mechanisms," "rule based systems with evolutionary approaches," and "multiagent approaches") are really needed to prove the effectiveness. In answer to this question, we have previously shown the effectiveness of integrating "learning mechanisms" and "rule based systems with evolutionary approaches" in OCS [Takadama 99a]. Furthermore, we have also shown the effectiveness of a "multiagent approach" integrated with the above two components. This was done by comparing the results of OCS with those of a model of the Michigan approach [Holland 78], which is one of conventional models in LCS and which employs the above two components in a centralized approach [Takadama 99b].

6 Conclusion

This paper has explored possibilities of our organizational learning model and has shown its rescheduling and reorganization abilities through examples of the crew task scheduling in a space shuttle/station and the task planning for truss construction with multiple space robots. The main results are summarized as follows: (1) Our model provides good feasible schedules quickly in the case of rescheduling, and it keeps the computational cost for rescheduling low; (2) Plans generated by our model keep or recover efficiency in tasks when robots are added, removed, or exchanged among robot groups; and (3) The integration of (a) learning mechanisms, (b) rule based systems with evolutionary approaches, and (c) multiagent approaches is effective in rescheduling/re-planning problems.

Future research will include an exploration of effective components, such as the above three properties, and will investigate their integrated effectiveness in scheduling and planning domains.

References

- [Aarts 89] E. Aarts and J. Korst: *Simulated Annealing and Boltzmann Machines*, Jon Wiley & Sons, 1989.
- [Argyris 78] C. Argyris and D.A. Schön: *Organizational Learning*, Addison-Wesley, 1978.
- [Cohen 95] M.D. Cohen and L.S. Sproull: *Organizational Learning*, SAGE Publications, 1995.
- [Fujita 96] S. Fujita and V.R. Lesser: "Centralized Task Distribution in the Presence of Uncertainty and Time Deadlines," *The Second International Conference on Multiagent Systems (ICMAS'96)*, pp. 95-102, 1996.
- [Goldberg 89] D.E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [Grefenstette 88] J.J. Grefenstette: "Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms," *Machine Learning*, Vol. 3. pp. 225-245, 1988.
- [Holland 78] J.H. Holland and J. Reitman: "Cognitive Systems Based on Adaptive Algorithms," in *Pattern Directed Inference Systems*, D.A. Waterman and F. Hayes-Roth (Eds.), Academic Press, 1978.
- [Iima 99] H. Iima, T. Hara, N. Ichimi, and N. Sonnomiya: "Autonomous Decentralized Scheduling Algorithm for a Job-Shop Scheduling Problem with Complicated Constraints," *The 4th International Symposium on Autonomous Decentralized Systems (ISADS'99)*, pp. 366-369, 1999.
- [March 91] J.G. March: "Exploration and Exploitation in Organizational Learning," *Organizational Science*, Vol. 2, No. 1, pp. 71-87, 1991.
- [Osman 96] I.H. Osman and J.P. Kerry: *Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers, 1996.
- [Russell 95] S.J. Russell and P. Norving: *Artificial Intelligence: A Modern Approach*, Prentice-Hall International, 1995.
- [Takadama 98a] K. Takadama, S. Nakasuka, and T. Terano: "Printed Circuit Board Design via Organizational-Learning Agents," *Applied Intelligence*, Vol. 9, No. 1, pp. 25-37, 1998.
- [Takadama 98b] K. Takadama, S. Nakasuka, and T. Terano: "Multiagent Reinforcement Learning with Organizational-Learning Oriented Classifier System," *The IEEE 1998 International Conference On Evolutionary Computation (ICEC'98)*, pp. 63-68, 1998.
- [Takadama 99a] K. Takadama, T. Terano, K. Shimohara, K. Hori, and S. Nakasuka: "Making Organizational Learning Operational: Implication from Learning Classifier System," *Computational and Mathematical Organization Theory (CMOT)*, 1999, to appear.
- [Takadama 99b] K. Takadama, T. Terano, K. Shimohara, K. Hori, and S. Nakasuka: "Can Multiagents Learn in Organization? ~ Analyzing Organizational-Learning Oriented Classifier System ~," *The 16th International Joint Conference on Artificial Intelligence (IJCAI'99) workshop on Agents Learning about, from and with other Agents*, 1999, to appear.
- [Tamaki 99] H. Tamaki, M. Ochi, and M. Araki: "Introduction of a State Feedback Structure for Adjusting Priority Rules in Production Scheduling," *Transaction of SICE (the Society of Instrument and Control Engineers)*, Vol. 35, No. 3, pp. 428-434, 1999, (in Japanese).
- [Zhang 95] W. Zhang and T.G. Dietterich: "A Reinforcement Learning Approach to Job-shop Scheduling," *The 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pp. 1114-1120, 1995.
- [Zweben 94] M. Zweben, B. Daun, and M. Deale: "Scheduling and Rescheduling with Iterative Reaper," In *Intelligent Scheduling*, M. Zweben and M.S. Fox (Eds.), Morgan Kaufmann Publishers, Chapter 8, pp. 241-255, 1994.

O-OSCAR: A Flexible Object-Oriented Architecture for Schedule Management in Space Applications

Amedeo Cesta, Angelo Oddi, Angelo Susi

IP-CNR, National Research Council of Italy

Viale Marx 15, I-00137 Rome, Italy

{cesta|oddi|susi}@pscs2.irmkant.rm.cnr.it

Abstract

This paper overviews the results of a project aimed at developing a state-of-the-art framework for intelligent constraint-based scheduling for activity management in space applications. The paper starts discussing the main features that an architecture for planning and scheduling should have to be actually used in a working environment. Crucial aspects are seen to be: the ability to build and dynamically maintain a representation of a certain domain; the ability to efficiently search for a solution in a space of possibilities, and the ability to effectively interact with users according to needs of different operative environments. The O-OSCAR software architecture is described that contributes to solve two classes of problems of increasing difficulty, a satellite scheduling problem and a resource constrained project scheduling problem for space missions.

1 Introduction

This paper describes the results of a project supported by the Italian Space Agency (ASI) ¹ aimed at developing a general framework for intelligent constraint-based scheduling and activity management in space applications. The initial goal of the project consists of building a reference architecture for temporal planning and scheduling that could be flexibly configured for different space applications. Although several results of the project can be independently used in both planning and scheduling

¹This paper describes research developed under a three years project titled "Stazione di lavoro per la generazione interattiva di piani per sistemi spaziali complessi" ("A workstation for the interactive generation of activity plans for complex space systems"). In November 1998 ASI has approved the continuation of the project for two further years with the title "Un toolkit per la creazione di pianificatori interattivi per sistemi spaziali complessi" ("A toolkit for the synthesis of interactive planners for complex space systems").

applications, during the project particular attention has been dedicated to scheduling problems that were relevant for the supporting agency.

Leading ideas for the project has been the following:

- to guarantee a complete approach to the resolution and management of a problem. This means being interested not only in developing a particular search algorithm for the problem but also in building up a framework able to support the "problem life-cycle" from the description of the domain knowledge to the presentation of different solution aspects to the users;
- to pay particular attention to the problem of plan/schedule maintenance. In particular we aim at supporting a rich query set to the solution and the definition of a number of update and modification commands on the current solution. We consider these aspects as basic starting points to allow the continuous use of intelligent scheduling systems in a work environment;
- to create an open representation able to support multiple approaches to the resolution of problems. In particular we have been interested to integrate multiple problem solving strategies in an uniform framework to allow comparisons but also to allow the use of the more appropriate approach according to the problem at hand;
- to design a software structure that allows to integrate different research results for the solution creation and management.

Specifically requirements and constraints come from space applications. In particular:

- since space missions span for several years since their original design, a major role assumes the possibility of modifying plans and schedules, as well as the details of the application domain, as soon as the steps of a mission become more mature. Attention towards the dynamic evolution of reality has been a peculiar aspect of our work.
- the explicit consideration given to aspects of user interaction and acceptance of the automated system in a working environment. A con-

tinuous attention has been dedicated to the investigation of human-computer interaction aspects customized to the application domain.

A final characterization concerns our own approach to the problem. Our main interest is focussed on constraint-based approaches to scheduling problems, we heavily rely on constraint satisfaction as both a representation tool, and as a mechanism for guiding problem solving, in this way being similar to approaches described in [12; 10; 11]. A more specific feature of our work has been the interest for scheduling problems with a strong temporal structure, in particular we have considered problem where quantitative temporal constraints are defined between activities to bound minimal and maximal distances among them.

The major result of our investigation has been the software architecture named O-OSCAR (Object-Oriented SCheduling ARchitecture) that represent a carefully design library of functionalities designed to support the previous requirements in an integrated way.

2 Ingredients for a Scheduling Architecture

To develop a complete solution for a planning/scheduling problem a basic step consists in identifying exactly the basic problems to be addressed, their peculiarities, and the interrelationships between them. Figure 1 sketches the results of our analysis showing four aspects that contribute to the solution.

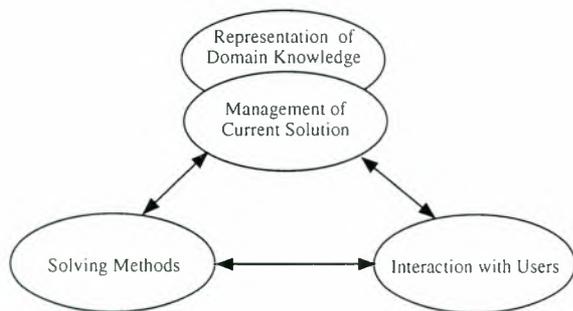


Figure 1: Functional Aspects

Two aspects are strictly interconnected:

Domain Representation Language. A key initial decision consists in defining the class of problems that is possible to address with the architecture. A Domain Representation Language allows the system developer to describe different aspects of the world that the scheduling system needs to know in order to produce a solution. Usually such languages allow the representation of classes of problems and the peculiar domain constraints.

Solution Representation and Management.

Constraint-based methods are centered on the production and maintenance of a symbolic solution that relies on a number of a specialized constraint reasoners, representing different aspects of the current context (e.g., temporal constraints, resource availability). When a change to the solution is performed by a problem solver or a user, the module taking care of solution representation checks the consistency of the change and updates its representation. The solution manager is usually endowed with a set of primitives that allow both atomic or aggregate changes, and with a set of query functionalities for knowing specific information in the solution.

It is not surprising that the basic representation language and the tools for representing the solutions represent the core part of an architecture (the part that more influences the further choices). It should be also clear that they are strictly interconnected, in fact the domain description should allow to express in a suitable way the main features of a domain but also, and more importantly, the constraints that limit finding a solution to a problem in that domain. All this features should be naturally mapped in the representation mechanism of the solution manager because the core of the constraint-based approach is an active service that automatically take care of checking/maintaining the satisfaction of the basic domain constraints.

Once done the architectural choices for realizing these two core components, a complete approach to the solution is obtained addressing the two missing aspects: adding one or more strategies to solve the problem and coping with the interaction with users. This means adding two further blocks to an architecture:

Automated Problem Solving. This is the module that makes available a portfolio of solution methods for a given class of problems (e.g., exhaustive search procedures, greedy heuristics, local search approaches). All the methods use the query and change primitives of the solution manager.

User-System Interaction. This module allows the interaction of the user with both the solution and the problem solving methods. The interaction functionalities may vary from more or less sophisticated visualization services, to a set of complex manipulation functionalities allowed to the user on the solution. A further aspect, very relevant in developing applications, consists in the possibility of adapting the interaction to the working tasks and competence of different users, in order to allow maximal productivity to each person that interacts with the scheduling system.

An advantage of having identified the basic func-

tionalities (and as a consequence the basic modules) a scheduling architecture should be endowed with stays in the possibility of focalizing the research on specific features of each part (e.g., the expressiveness for the Description-language, the efficiency and flexibility of services for the Solution-manager, the ability to controlling search for the Solver; the capability to be adaptable to different needs for the Interaction-module).

It is worth observing the key role that the solution management has in this approach (see the central placement in Figure 1). As a consequence, a major effort in our work has been dedicated to produce a flexible, configurable and efficient software system for schedule management.

3 The O-OSCAR Architecture

As said in the introduction, the project has focussed its attention on the production of an open software architecture for the solution of scheduling problems. Such a software system, named O-OSCAR (Object-Oriented Scheduling ARchitecture), is a principled kernel of functionalities that has allowed to create an open, configurable framework to be adapted to multiple contexts.

Following the distinctions introduced in Section 2, O-OSCAR mainly consists in a software system that makes available the pair $\langle \textit{Description-language}; \textit{Solution-manager} \rangle$ according to a class of problems. Such software system guarantee an amount of functionalities that joined with a problem solving algorithm and an interaction module allows for the development of a complete system to solve a class of problems.

As explicitly stressed in the name of the system, a main feature of O-OSCAR is the attention paid to the object-oriented design. Object-oriented techniques allow the stable implementation of specific modules with clear interfaces that can be composed to configure the software system according to the application. Moreover, the use of specialization techniques allows also an incremental refinement of different functionalities.

Figure 2 shows the general schema followed to create O-OSCAR versions for two different classes of problems.

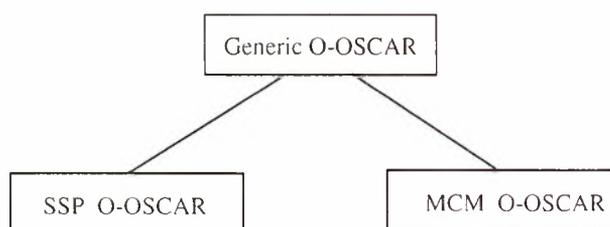


Figure 2: O-OSCAR: The Developed Software

We have designed a generic representation for schedules named *Generic O-OSCAR* that currently plays the role of Domain Description Language at the higher level of abstraction. *Generic O-OSCAR* identifies the typical aspects involved in a schedule, namely resources, activities, constraints and decisions. Having chosen a general representation allows us to interface our work directly with typical abstractions from Operations Research (see for example [13]).

The generic level has been specialized to create software architectures for two classes of scheduling problems:

- The *SSP problem* (SSP stands for Satellite Scheduling Problem). It represents scheduling domains in which resources have binary capacities, activities may have flexible temporal durations and the user may specify preferences over allocation intervals. This class of domains is quite frequent in space applications in particular in satellite allocation requests scheduling (see later the DRS request allocation problem we have studied).
- The *MCM problem* (MCM stands for Multi-Capacitated and Metric). The MCM O-OSCAR management system represents a more sophisticated problem characterized by resources whose capacities are integer numbers greater than 1 (to represent aggregate resources), and several metric temporal separation and time-window constraints may be represented. This class of problems include quite complex cases like Multi-Capacitated Metric Job-Shop [5] and Resource Constrained Project Scheduling [8]. With such an extension it is possible for example to deal with several mission planning problems having the possibility of expressing a quite realistic set of constraints over available resources.

The development of two different software systems is due to needs of the project. Focalizing on SSP has allowed us to prototype quickly a complete system to be used to make the dialogue with the supporting agency more concrete. Nevertheless SSP O-OSCAR allows to address effectively a subset of scheduling problem very frequent in space domains. The development of MCM O-OSCAR derives from the experience of SSP but has involved a major redesign to cope with more sophisticated constraints and a wider class of problems.

Both SSP and MCM O-OSCAR share the same layered software design that allows us to interface the quite general representation language with the constraint-based AI techniques we wanted to use at the lower level. In particular three layers have been defined a sequencing layer, a causal layer, a constraint layer.

The *sequencing layer* is the interface of the system with the problem solver (also called Sequencer in the

following) and the interaction module functionalities. It inherits the abstract characterization of *Generic O-OSCAR* and allow to see a schedule subdivided in resources, activities, constraints and decisions. In particular the decisions represent an association with an activity and the resources it require to be executed and it is used as an input/output parameter to return the actual solution. Of course in the SSP and MCM software systems different methods are available to allow intervention by the sequencer and the users.

The *causal layer* is the level influenced by Artificial Intelligence symbolic representation techniques. It contains a structured description of the temporal evolution of the resources and the activities (in this way it represent a "causal model" of the domain, hence the name of the layer). In particular a further internal representation entity is used, the *token*, to fully represent the association among an activity, the resources it requires, the temporal and technological constraints it should satisfy in any solution.

The *constraint layer* is the level at which both general and specialized constraint satisfaction techniques are used. This level at present contains representation capabilities for temporal constraints (in particular consistent with the quantitative time network manager described in [2; 4]), and for resource constraints (namely the possibility is given to use either the propagation algorithms described in [7] or the more procedural profile-based representations formalized in [5]). It is to be noted that this lower level is a layer that offers services to the higher levels and can be sophisticated more or less according to the requirements of the current problem.

We end this section commenting about similarities between O-OSCAR solution management capabilities and, on one side, the blackboard-based representation used in OPIS [12] and SONIA [10], and, on another side, with the temporal data-base used in HSTS [11]. Similarities with the first two systems are at the methodological level both that system being reference examples for the constraint-based approach to scheduling. The similarities with HSTS are more strict and should be more carefully analyzed. We share with that system the use of a complete temporal propagation. We differ strongly with our sequencing layer because we interface a more simple generic schedule description language (influenced by [13]) instead of the description language of HSTS [11; 3] more suitable for temporal planning problems. At the causal and constraint layers the difference starts from our attempt to deal with complex multi-capacitated problems that turned out in a representation quite different from the one currently reported for HSTS.

We continue the paper giving a short overview of the two complete systems we have built starting from SSP and MCM O-OSCAR respectively.

4 Using SSP O-OSCAR

As a first use of the SSP version of O-OSCAR we have developed a complete system to solve request allocation problem for the Data Relay Satellite (DRS) System that we had previously addressed with a more "classical" knowledge-based approach [1]. The Data Relay Satellite (DRS) System is a European Space Agency program aimed at providing a data relay service between Low Earth Orbiting (LEO) satellites and their ground terminals. Actually this program is in the last step of development, and it will be operative within 1999 (its actual name being Artemis).

The scheduling problem of DRS consists in the production of a mission plan, that allows the clients to utilize the transmission services. An high number of access requests is expected, so that their temporal extension exceeds the total transmission time available, introducing conflicts that have to be solved following some quality objectives. Given the technical characteristics of the DRS system, the crucial aspect in the production of the plan is the management of the link between the DRS and the LEO satellites, while the links between DRS and ground stations are less problematic. The first type of link imposes the satisfaction of physical constraints of the DRS's antennas, temporal constraints of the requests, and requirements of priority, commercial value and allocation preference.

An interesting aspect of the problem is represented by the requests and related constraints. All user requests specify a number of desired characteristics which include: (a) static priority associated to the request's owner; (b) technical requirements: these may include for example the band, speed of transmission and the number of channels required; (c) user flexibilities: minimum and maximum time intervals for the duration, the interval of time within which the access must be scheduled (flexibility interval) and the utility function associated with these flexibilities; (d) user preferences: preferred values for the duration and the actual access time.

Goal of the system is to generate the Detailed Assignment Plan (DAP): (a) schedules of as many access requests as possible; (b) satisfies of as many user preferences as possible; (c) gives priority to preferences of requests having a higher "relevance" coefficient. The goals are potentially conflicting: an optimization in resource use required to satisfy the first goal would imply taking full advantage of user specified flexibilities but in doing so, the preference (or utility) function given by the users may not be satisfied. The other two goals are in turn partially contrasting, since maximizing user preferences does not necessarily coincide with satisfying the requests of preferred users.

According to the technical documentation, the production of the DAP is supposed to follow an iterative process repeated three times, and that involves two

types of human operators (see the schema in Figure 3):

- *Commercial operators* at the Mission Control Center negotiate the sale of the free transmission spaces with the clients, and insert the related activities in the plan;
- *Spacecraft engineers* (called Operative users) at the Operation Control Center modify the plan inserting some special activities for the maintenance of the system operativity and requests with a special requirement of urgency.

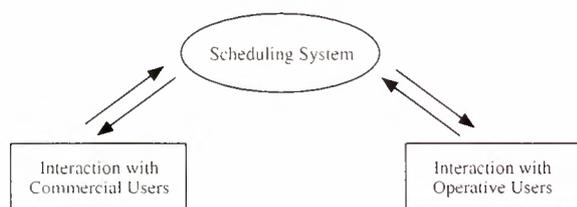


Figure 3: Users Views in DRS O-OSCAR

These two operational profiles follow different and potentially conflicting objectives (maximum satisfaction of requests vs. DRS's resources saving). Those objectives have to be integrated together in an automated scheduling system that supports decision making in this environment.

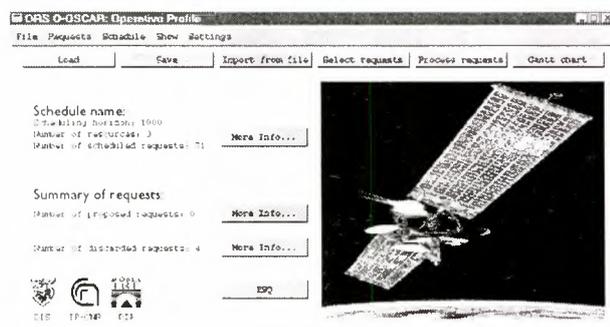


Figure 4: The interface for the operative user

The DRS Demonstrator has been built on top of the SSP O-OSCAR functionalities. According to the O-OSCAR methodology a complete system is developed starting from a core set of functionalities that are able to represent and manage a solution for a class of problems (SSP in this case). In addition to the basic functionalities for solution representation and modification, two different modules need to be built:

- A *Sequencer* that is able to produce incremental modifications on a current solution to satisfy current goals. In the DRS case a set of specialized heuristics is able to produce assignment plans in a time compatible with the duration of activities in the operative environment.

- An *Interaction Module* that allows multiple users to use the planning facilities of O-OSCAR extracting services according to the working tasks. In particular two interface profiles have been defined one for the tasks of Commercial operators and one for the tasks of Operative users (a picture is shown in Figure 4).

A peculiar characteristic of O-OSCAR is its ability to support dynamic modification to the schedule after producing a solution: it is possible to introduce a single new activity in the schedule, remove activities to serve a maximal priority one, etc.

5 Using MCM O-OSCAR

Having demonstrated the potentiality of the O-OSCAR architectural approach we have worked at producing a framework able to cope with a wider class of problems. It is worth remarking that extending O-OSCAR to cope with MCM problems allows to model temporal constraints like "observation temporal windows" very peculiar in space exploration and science, and resource constraints like "amount of energy" and "workforce" that are common in modeling the ground preparation of space missions and in the managing of space instruments.

A quite complex example of the new range of functionalities given by MCM O-OSCAR is represented by the so-called RCPSP/max problem (Resource Constrained Project Scheduling Problem with Time Windows, or with Generalized Precedence Relations). In such problem a set of activities are connected by a temporal structure that represent a project to be completely executed to solve the problem. Each activity requires different sizes of certain resources to be executed and should satisfy a number of temporal constraints with respect to other activities. The distance separating two activities may satisfy minimal and maximal duration constraints. Domain resources have a capacity greater than one.

Again, attention has been given to the possibility of incremental constructing the solution, to the ability of modifying something when a schedule exists, etc. Around the basic representation and management functionality we have built a complete system following the O-OSCAR methodology. Two modules have been added:

- A *Sequencer*. To cope with Project Scheduling problems we have built a multi-strategy module that allow the integration of several resolution procedures. In particular we have integrated state-of-art branch and bound [8] and heuristic [9] algorithms from the Operation Research community and our original constraint-based algorithm called ISES [6]. In this way we are able to test multiple approaches to the same problem but also to use the more appropriate algorithm according to the dimension of the problem.

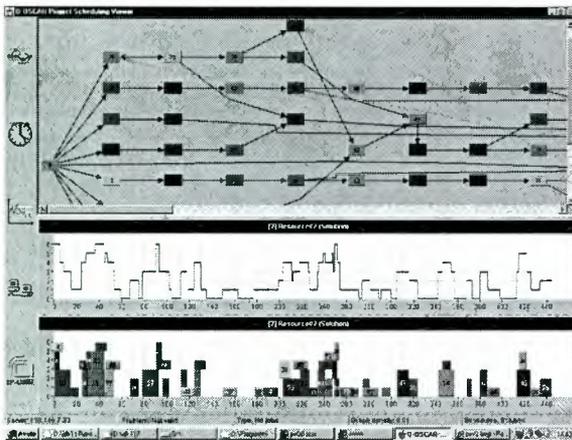


Figure 5: The PS O-OSCAR Viewer

- An *Interaction Module*. In this case a complex direction has been successfully attempted: the development of a client-server architecture and a Java client that interacts with the scheduling system through a specialized communication protocol. The result is a quite sophisticated interface a snapshot of which is shown in Figure 5.

The result of this effort is the PS O-OSCAR system that at present is able to effectively solve recognized benchmark problems and has also been officially demonstrated to ASI. It is worth remarking that also in the case of PS O-OSCAR, the functionality of dynamic modification of the schedule (that was a peculiar aspect in the DRS demonstrator) has been reproduced in this more complex scenario.

6 Conclusions

This paper has described the main aspects of O-OSCAR a scheduling architecture for plan production and management. Two different systems relative to different space applications have been successfully developed with it. Interesting features of O-OSCAR are the complete approach to the scheduling problem, the possibility of adding dynamic modifications to a current solution, the possibility of usefully integrating different solution strategies in the same system. A further aspect, very peculiar in our work, is the attention given to interaction with the user in the solution management process and the adaptation of such interaction to the user tasks and competence.

Acknowledgments

This research is supported by ASI (Italian Space Agency) and is part of a joint effort among Dipartimento di Informatica e Sistemistica dell'Universita' di Roma "La Sapienza", Dipartimento di Informatica e Automazione della Terza Universita' di Roma, and IP-CNR, Consiglio Nazionale delle Ricerche, Roma. Thanks to Luigia Carlucci Aiello (Project Coordinator), Marta Cialdea Mayer and Marco Schaerf for

creating a stimulating environment for the project. Special thanks to several people that contributed to the O-OSCAR development, namely Paolo Bazzica, Gianni Casonato, Gabriele Giuseppini, Attilio Mainetti and Fabrizio Peroni.

References

- [1] M. Adinolfi and A. Cesta. Heuristic Scheduling of the DRS Communication System. *Engineering Applications of Artificial Intelligence*, 8:147-156, 1995.
- [2] R. Cervoni, A. Cesta, and A. Oddi. Managing Dynamic Temporal Constraint Networks. In *Artificial Intelligence Planning Systems: Proceedings of the Second International Conference (AIPS94)*, 1994.
- [3] A. Cesta and A. Oddi. DDL.1: A Formal Description of a Constraint Representation Language for Physical Domains. In M. M. Ghallab and A. Milani, editors, *New Directions in AI Planning*. IOS Press, 1996.
- [4] A. Cesta and A. Oddi. Gaining Efficiency and Flexibility in the Simple Temporal Problem. In *Proceedings of the Third International Workshop on Temporal Representation and Reasoning (TIME-96)*, 1996.
- [5] A. Cesta, A. Oddi, and S. Smith. Profile Based Algorithms to Solve Multiple Capacitated Metric Scheduling Problems. In *Proceedings of the Fourth Int. Conf. on Artificial Intelligence Planning Systems (AIPS-98)*, 1998.
- [6] A. Cesta, A. Oddi, and S. Smith. An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows. In *Proceedings of the 16th Int. Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.
- [7] A. Cesta and C. Stella. A Time and Resource Problem for Planning Architectures. In *Proceedings of the Fourth European Conference on Planning (ECP 97)*, 1997.
- [8] B. De Reyck and W. Herroelen. A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations. *European Journal on Operations Research*, 1998. to appear.
- [9] B. Franck and K. Neumann. Resource Constrained Project Scheduling Problems with Time Windows - Structural Questions and Priority-Rule Methods. Technical Report WIOR-492, Universität Karlsruhe, 1998. (Revised November 1998).
- [10] C. Le Pape. Scheduling as Intelligent Control of Decision-Making and Constraint Propagation. In M. Zweben and S. M. Fox, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.
- [11] N. Muscettola. HSTS: Integrating Planning and Scheduling. In M. Zweben and S. M. Fox, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.
- [12] S. Smith. OPIS: A Methodology and Architecture for Reactive Scheduling. In M. Zweben and S. M. Fox, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.
- [13] G. Wolf. Schedule Management: An Object-Oriented Approach. *Decision Support Systems*, 11:373-388, 1994.

Distributed Planning in Constellations of Autonomous Spacecraft

Dr. T. J. Grant and Dr. A. H. Broeils

Origin Nederland B.V., Technical Automation/Command and Control,

P.O. Box 1444, 3430 BK Nieuwegein, The Netherlands

Tel: +31(0)30 608 88 88

Fax: +31(0)30 606 05 77

E-mail: {Tim.Grant, Adrick.Broeils} @nl.origin-it.com

Url: <http://www.origin-it.com>

1. ABSTRACT

This paper describes the application of Origin's generic simulation architecture and complementary toolset to a demonstrator facility for planning in satellite constellations. The architecture was designed to support widespread applicability of real-time simulation technology on low-cost platforms like PCs with Windows NT. The goal of the facility is to simulate and analyse planning problems that will occur in constellations of autonomous satellites. We present an innovative concept for mission planning based on a distributed planning algorithm. We find that Origin's generic simulation architecture is well suited for building a distributed planning tool.

2. INTRODUCTION

The purpose of this paper is to describe an innovative concept and implementation for mission planning and scheduling in constellations of autonomous spacecraft. The need for an innovative concept follows from the observed market trends that spacecraft will become smaller, highly autonomous, and will often operate in constellations, gathering and returning massive volumes of data to Earth. We foresee tuning problems both for downlinking data of several spacecraft down to one ground station as for uplinking detailed schedules (in the form of Tele-Command (TC)-sequences) to all satellites in a constellation. Additionally, the planning effort for constellations with tens or hundreds of satellites will be large.

Section 3 provides more information on the background of the tuning problems for satellite constellations. In Section 4 we describe various concepts for mission planning and argue that decentralised planning is the most promising concept to solve these problems. Under this concept, a plan is negotiated by the satellites in the constellation, rather than detailed in the Mission Control Centre (MCC). A possible negotiation protocol is introduced. Section 5 illustrates how Origin's generic simulation architecture is used to build an analysis tool for planning

in satellite constellations. Finally, in Section 6 we draw conclusions.

3. PLANNING IN SATELLITE CONSTELLATIONS

The current generation of (remote-sensing) satellites is designed around individual, large satellites carrying many instruments. Space exploration and applications in the 21st century will be significantly different from what we see today.

To save development time and cost, increasing use will be made of small satellites, typically equipped with only one or two instruments. In case of failure these *smallsats* can be replaced for a much lower price than the present day large satellites. Smallsats will have relatively cheap designs, and can be "mass" produced. The low cost of individual smallsats allows the use of spare satellites as part of the mission plan (as is the case in present-day telecommunication constellations systems, like Iridium [1]).

Smallsat will be able to operate on an individual basis, but a considerable portion will also cooperate in constellations, formations and networks to fulfil certain missions. Constellations will provide the coverage needed in many missions, like the Global Positioning System (GPS) for navigation. Satellite networks are already used in telecommunication, like the Iridium system. Formations consist of (generally) non-identical, but complementary spacecraft and together provide new products. For instance, satellite formation can be used to observe simultaneously the same target under different geometry's, with various instruments. In this paper we will use the word *constellations* for constellations, networks and formations of satellites.

Smallsats will become increasingly more autonomous. Autonomy will decrease the operating cost, since satellites will operate more independently from ground control.

Presently, nearly all planning for individual satellites and satellite constellations is done centrally in the

MCC. In the MCC the end-user goal is decomposed into sub-goals and allocated to the appropriate smallsats. Each smallsat's subgoal-set is decomposed into TC-sequences and uplinked by a ground station to the appropriate smallsat and executed.

This approach will become increasingly difficult for large constellations of satellites, because of computational complexity. Conflict resolution, scheduling and uplinking will become a bottleneck. Especially in cases where a number of satellites have to cooperate to perform a certain task, the planning and detailed control of the individual satellites from the ground becomes nearly impossible. Eventually, the mission planning process will no longer be able to react sufficiently swiftly to new user requests and to unforeseen situations happening in the constellation.

In addition, new resource bottlenecks arise in constellations. Spacecraft will compete in using the limited number of ground stations to download their data. The situation will be aggravated for smallsats, which only may carry one or two payloads. Besides, smallsats may have to cooperate in groups for limited period of time to perform some operations.

Therefore, we foresee two tuning problems:

1) planning and controlling the individual satellites for co-ordinated actions from the ground will be nearly impossible, and 2) downlinking of data by multiple satellites after a co-ordinated action to a single ground station will lead to data congestion. Clearly, the existing off-line, ground-based planning facilities will be unable to cope with generating fully-detailed TC-sequences for constellations containing up to tens or even as many as a few hundred satellites (as proposed in the Teledesic system[2]).

4. SOLUTIONS

4.1 Planning concepts

Origin initiated a project to investigate the tuning problems mentioned in the previous section. The end-goal of the project is to develop of a planning facility for satellite constellations. The planning facility can analyse planning concepts by translating a high-level user request into a detailed planning of required actions for the individual satellites in the constellation.

Planning can be done in various ways and at different locations. To create a plan the following approaches can be used:

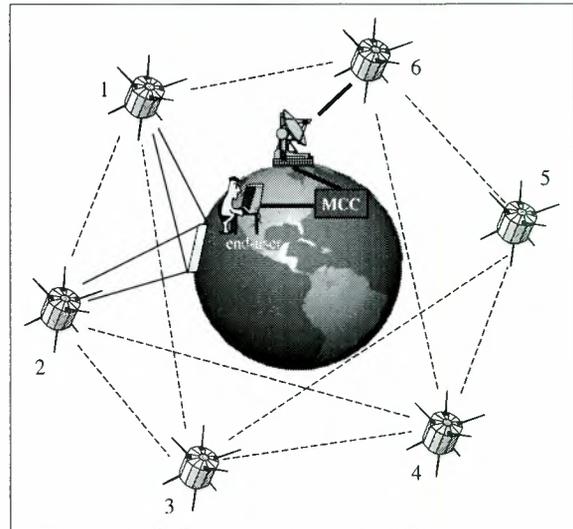


Figure 1 Decentralised planning approach

- *Centralised planning:* In the traditional approach a planning for individual satellites and satellite constellations is done centrally in the MCC. In the MCC the end-user goal is decomposed into sub-goals and allocated to the appropriate satellite. Each satellite's subgoal-set is decomposed into TC-sequences and uplinked by a ground station to the appropriate satellite, and executed.
- *Decentralised planning:* In a decentralised planning scenario the MCC translates the end-user request into a high-level command (HLC) that is uplinked to one of the satellites in the constellation. The satellite decomposes the high-level command into sub-goals and distributes these among the other satellites (through Inter-Satellite Links or ISLs) in order to achieve an internal planning of the required actions. Once the sub-goals are allocated to specific satellites, every satellite will generate its own TC-sequences and will execute them. Decentralised planning seems to be the most promising approach, given the increase in the number of small autonomous satellites foreseen in the future.

In addition there is an intermediate approach between these two extremes:

- *Planning by distributed simulation:* Planning is done in the MCC through distributed simulation of the constellation. The MCC sends a user request in the form of a HLC to the simulated constellation. The simulated constellation negotiates a plan, detailing this fully as TC-sequences. The TC-sequences are extracted from the distributed simulation and uplinked to the appropriate satellites in the real constellation.

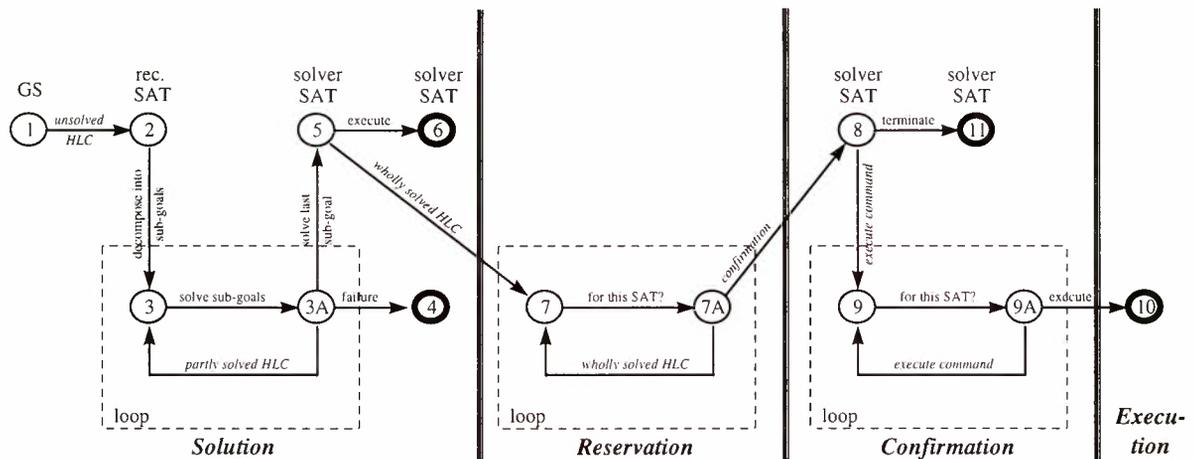


Figure 2 Negotiation steps

We argue that decentralised planning is the most promising solution for the tuning problems mentioned above. To show that decentralised planning is feasible we use distributed simulation with satellites modelled as autonomous agents as the foundation for the planning facility. With this approach, two separate solutions are still possible, depending on the capabilities of the satellites in the constellation. If the real satellites are capable to communicate with each other, the planning facility could serve as a validation facility which validates the feasibility of the request faster than real-time. If the real satellites do not have ISLs, the detailed planning produced by the facility can be sent to the satellites in the traditional approach. At this stage of the project we see the planning tool as a useful tool for mission engineers and specialist to be used for feasibility studies.

A schematic picture of the mentioned decentralised approach is shown in Figure 1. Here the high-level request is (for example) a request for an observation of a specific target on Earth at a specific time and using two specific instrument. The ground station uplinks the HLC to the first satellite that comes within reach (nr. 6). The satellites in the constellation start negotiating until it is determined that satellites 1 and 2 have the required instruments and are above the target at the specified time. If satellites 1 and 2 accept the command (i.e. allocate time and resources in their timelines), they decompose their portion of the HLC into TC-sequences and perform the observations.

The appropriate technology to implement a decentralised planning scenario exists in Distributed Artificial Intelligence (DAI), where it is known as Multi-Agent Systems (MAS). For our facility it means that each satellite, MCC, and ground station can be modelled as an agent, with the satellites having intelligent planning and negotiation capabilities. Since we

chose to build a facility for a remote-sensing mission, we also need to simulate the observable targets.

4.2 Negotiation and planning process

In the decentralised planning scenario, the planning is done through negotiating between satellites in the constellation. The negotiating process goes through a number of steps¹, each step involving a different set of possible messages. Consequently, several processes must occur in a satellite, depending on the type of message it receives.

We see a HLC as a combination of several sub-commands or sub-goals that need to be “solved”. The first satellite that receives the HLC (uplinked from the ground) will split it into several sub-goals. We assume for simplicity that a satellite can solve a sub-goal independently from any other satellite. During the planning process the complete HLC command is passed on from one satellite to another, with solutions to the sub-goals being progressively added by each satellite.

We identify four steps in the negotiation process:

- *Solution.* The HLC is decomposed into sub-goals and actions by the satellite that initially receives it. The input is an unsolved HLC. After decomposition, the set of sub-goals is broadcast to the other satellites in the constellation for solution. One or more satellites co-operate in proposing solutions to the sub-goals until the complete set is solved or failure. Output is a wholly solved HLC or a message informing the MCC of failure. If the output is a wholly-solved HLC, then the satellite achieving the solution becomes the “Solution Master” (we will call it the solverSAT).

¹ Like, for example, Tender, Bid, and Award in the Contract Net Protocol [3].

- *Reservation.* The co-operating satellites assign the actions by making resource reservations in their timelines. The inputs are one or more wholly solved HLCs, the outputs are confirmations to the solverSAT.
- *Confirmation.* The solverSAT determines which potential solution will be adopted. As there will be no optimisation in this project, the first potential solution obtained will be adopted. The input is a confirmation message and the output is an execution message to the satellites involved.
- *Execution.* The satellites involved in the confirmed solution execute the actions using the resources reserved in their timelines. The input is an "execute" command, and the output is a message to the MCC to report execution.

The four major steps in the negotiation process are schematically depicted in Figure 2. The numbers model states in the negotiation process, and the arrows indicate transitions between these states. The uplink of the unsolved HLC by the ground station is the starting point of the whole process (State 1). End-states are indicated by the "fat" circles (states 4, 6, 10, 11). Roughly half of the transitions correspond with message exchange *between* satellites (indicated by italic type-face in Figure 2), the rest are transitions *within* a satellite. States in the top row of the diagram occur in objects that (temporarily) have a special function in the constellation, i.e. the satellite that receives the unsolved HLC from the ground station, the solverSAT (the satellite that solves the last sub-goal), and the ground station. States in the bottom row occur in some or all of the other satellites in the constellation. There are three loops in the negotiation process. In these loops messages are spread through the constellation until all sub-goals are solved (in "solution" loop) or until message reaches the correct satellite (in "reservation" and "confirmation" loops).

5. TECHNOLOGY

5.1 Architecture

Origin has defined a generic architecture to support a wide spread applicability of low-cost real-time simulation technology. The architecture (described in [4]) is designed to support re-usability, extensibility, and scalability. Two types of simulations are recognised, the virtual world (consisting of a number of autonomous virtual objects) and the simulator (consisting of several components that have detailed knowledge of each other).

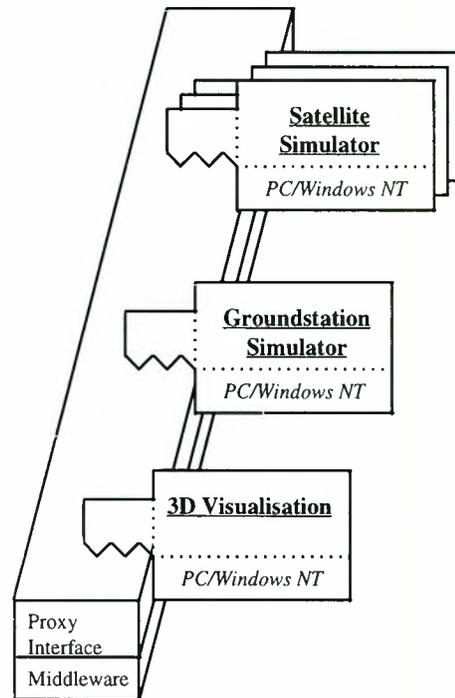


Figure 3 Virtual world simulation architecture

For the constellation planning simulations the virtual world part of the generic architecture is used (Figure 3). It contains a middleware layer based on a proxy design pattern. The proxy interface simplifies interfacing with the simulated world, by automatic creation of representatives of all other objects within the context of an application. This middleware approach abstracts the application from the communication infrastructure as it maps the simulated world directly into the context of the application. At the virtual world three categories of applications are found, all connected to a proxy interface layer:

- *Simulator Applications* are applications that simulate the behaviour of an object within the virtual world, regularly updating their state, monitoring the state of other objects and interacting with other objects. In the planning facility satellites, ground stations, MCC, targets, and even the Earth are examples of simulator applications. Their internal construction are hidden from other objects, but objects publish information when interacting with other objects.
- *Command & Control (C²) Applications* are applications that do not represent an object in the simulated world but can interact with objects. A typical example is a telecommand and telemetry station, which receives information from a satellite and can uplink new commands, but is not an object with a representation in the simulated world. In the planning facility the C² and simulator applications are actually combined in the MCC. It receives a request from the user, and translates this to a HLC for the constellation.

- *Generic Applications* are tools that can be used within every distributed simulation. Typically these respond to control and monitoring interfaces. An example is a 3D visualisation tool, which can visualise the state and interactions of objects within the virtual world.

For these type of simulations the coherence is low, and a variable number of objects is supported which can even be varied during a simulation. This last feature is useful for a satellite constellation simulation if the need arises to evaluate plans in case of launching new satellites or error situations.

5.2 Implementation

The virtual world part of the architecture is traditionally called a distributed simulation architecture. For this part we use the High-Level Architecture (HLA) as the middleware layer. HLA was defined by the US Department of Defense (DoD). The DoD declared that HLA was to be the standard technical architecture for all DoD simulations. As such it is anticipated that HLA will also quickly become a standard architecture in the space industry. HLA describes a standard interface for simulators, which allows them to interact over a network, using a global notion of *time*, i.e. a global ordering is given to all events in the simulation.

During a simulation each “physical” object (e.g. satellite, ground station, planet etc.) publishes its position, velocity and other information through the HLA services to all other simulators (or federates in HLA terminology) in the federation (A federation in our case is a collection of all federates in the simulation) at regular time intervals. This information can be used by a federate, for example, to see if communication between two federates is possible.

Origin developed the proxy interface ADS on top of the HLA middleware layer. It creates a representative of each instance of the classes derived from the “physical” objects” class within the local environment of each application (e.g. satellite, visualisation tool etc.). This is illustrated in Figure 4. Each simulator obtains status updates (position, orientation) of other objects through the proxy interface.

The planning processes as described in Section 4.2 are implemented as member functions of the physical object classes. Message exchange between satellites and between satellites and ground stations are implemented as HLA interactions and exchanges of object ownership. Interactions are short-duration events that can be discovered by all other simulators.

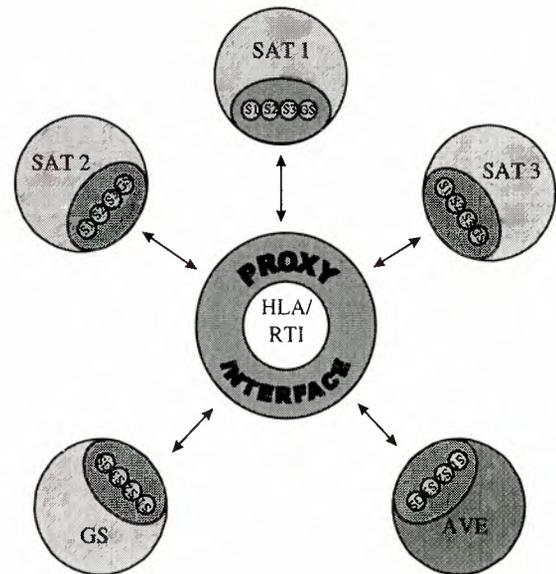


Figure 4 ADS Proxy Interface to HLA services

5.3 AVE: A 3D stealth application

For most distributed simulation, a general 3D view of the virtual world is required, which allows inspection of the virtual world without influencing it. Such a tool is often referred to as a ‘stealth’.

Based on the object-oriented proxy interface and an onward development in creating an object-oriented Virtual Environment, a low-cost 3D stealth (named AVE) has been created. The stealth supports the dynamic character of distributed simulations, where multiple instance of a type of object can exist. Each “physical” object automatically receives a connected camera, orientation axes, head-up display, trail facility, interaction display and message display facility. In our architecture (Figure 3 and 4), the VE tool is also a federate receiving the attribute updates and interaction of all other federates in the federation, enabling it to project satellites and message between satellites and other objects on the display. Figure 5 shows a screendump from the implemented Planning in Autonomous Constellations Tool (PACT).

6. CONCLUSIONS

With the growth of the number of autonomous smallsats operating in constellations we foresee severe planning problems if one tries to perform all planning actions centrally in the MCC. We propose a distributed planning concept where the planning is autonomously negotiated by the spacecraft in the constellation. We show that this concept can be analysed well using a distributed simulation facility that we implemented using Origin’s generic architecture for low-cost simulations. Especially, the scalability issue is of importance here: to simulate a

much larger system one can simply add a few low-cost PCs to the simulation environment.

7. REFERENCES

1. See <http://www.iridium.com>
2. See <http://www.teledesic.com>
3. R.G. Smith, R. Davis, *Proc. 1979 Int. Conf Cybern Soc*, Oct. 1979
4. J. Keijzer, L. Bremer, R. Trompert, *A Satellite Constellation Planning Tool: Application of a low-cost real-time simulation architecture*, Proceedings, 1999 European Simulation Multiconference, Warsaw, Poland, 1-3 June 1999.

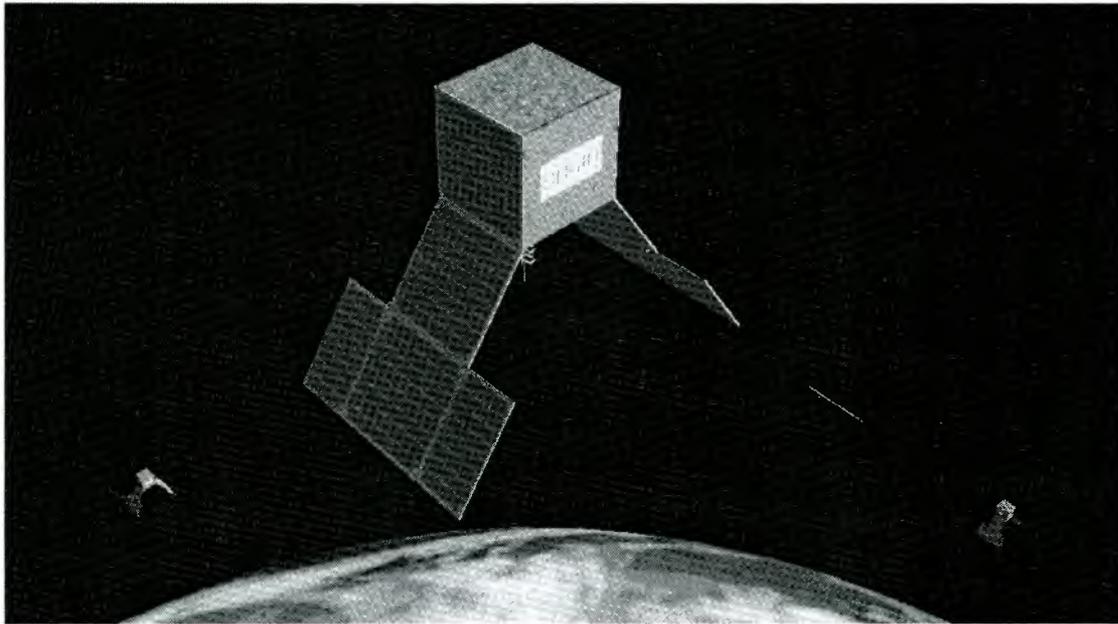


Figure 5. Screenshot from Planning in Autonomous Constellations Tool.

Space Robot System Verification

Hardware-in-the-Loop Simulation of Robots Performing Contact Tasks

Farhad Aghili Erick Dupuis Jean-Claude Piedboeuf
Jean de Carufel

Space Technologies, Canadian Space Agency,
6767 route de l'Aéroport, St-Hubert, Québec, Canada
tel: (450) 926-4688, fax: (450) 926-4695 email: Jean-Claude.Piedboeuf@space.gc.ca

Abstract

The SPDM Task Verification Facility is being developed to verify all SPDM tasks on the ground prior to their execution in space. The operations requiring a contact between the end effector and the work-site will be verified using a hardware-in-the-loop simulator (HLS). Two algorithms for the control of the HLS are proposed: a position control algorithm inspired from a force reflecting master-slave control architecture and Cartesian linearisation scheme. The performance of each scheme is analysed in terms of its ability to faithfully reproduce the dynamic behaviour of SPDM using the HLS. Conditions are given on the position control scheme to ensure adequate performance and its limitations are shown. The Cartesian feedback linearisation scheme is shown to give good fidelity emulation without the strict restrictions of the first control algorithm. A linearisation error compensator is proposed for the second scheme and is shown to ensure boundedness of the error in the presence of linearisation errors. Experimental results showing the application of the two control schemes to a single link robot are given.

1 Introduction

The International Space Station (ISS) will be assembled in space by collaboration of United States, Canada, Japan, Russia and European countries. The Special Purpose Dexterous Manipulator (SPDM) will play a key role in the assembly and maintenance of ISS which involves execution of numerous tasks. The cost and risk associated with execution of robotic tasks in space require that all procedure be verified on Earth prior to their execution in space. The Canadian Space Agency is currently developing the SPDM Task Verification Facility (STVF) that will be used to verify all SPDM tasks before their occurrence in Space. The SPDM contact tasks will be verified using a hardware-in-loop simulation (HLS) [1, 2].

The main requirement for the HLS is the fidelity of the simulator dynamics with respect to that of

SPDM. The pose of the robot end-effector and the contact forces generated throughout the task must accurately represent those of SPDM in free and constrained motion. Two algorithms are proposed for the control of the HLS component of STVF. The first method is based on a position control scheme whereas the second one uses a Cartesian feedback linearisation scheme. The performance and limitations of each algorithm are discussed in detail.

2 Position Control Algorithm

The first control algorithm proposed is inspired from a force-reflecting master-slave control architecture where the master is the dynamic simulation of the space robot. The operator enters velocity commands into the dynamic simulator and the ground robot is controlled to track the simulated end-effector position of the space robot. Contact forces are fed back to the dynamic simulator.

2.1 Linear Model of HLS

To analyse the closed-loop dynamics of the hardware-in-the-loop simulation, a linear model (shown in Figure 1) is developed using input-output transfer function in the Laplace domain. The robots are treated as linear systems with two inputs and a single output. The output of the space robot simulator is the end-effector position and its inputs are a velocity command and a force perturbation. The quantities and variables associated with the space robot simulator and ground robot are depicted by subscripts s and r respectively. By applying the principle of input superposition for linear system, the simulator and robot tip positions, x_s and x_r , can be calculated by

$$X_s = G(s)\dot{X}_{des} + Z_s^{-1}(s)F, \quad (1)$$

$$X_r = T(s)X_s + Z_r^{-1}(s)F. \quad (2)$$

In the above \dot{X}_{des} is the real-time control command to the simulator and F is the force perturbation resulting from the contact between the robot

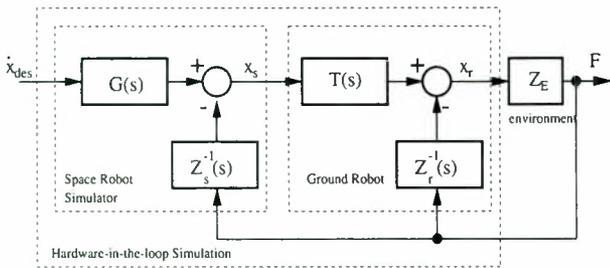


Figure 1: Linear Model of HLS with Contact

and the environment. The closed-loop dynamics of the robot position controller is represented by $T(s)$, the complementary sensitivity function of the ground robot, while $G(s)$ is the dynamics of the space robot simulator in free-space. Moreover, transfer functions $Z_s(s)$ and $Z_r(s)$ represent the impedances of the space robot simulator and of the ground robot respectively. The impedance transfer functions, $Z_s(s)$ and $Z_r(s)$, dictates the dynamical response of the robot endpoint to an external force, F .

By rewriting equations (1) and (2), the linear model of the hardware-in-the-loop simulation can be rewritten in matrix form.

$$\begin{bmatrix} X_s \\ X_r \end{bmatrix} = \begin{bmatrix} G & Z_s^{-1} \\ TG & Z_r^{-1} + TZ_s^{-1} \end{bmatrix} \begin{bmatrix} \dot{X}_{des} \\ F \end{bmatrix} \quad (3)$$

The equivalent impedance of the hardware-in-the-loop simulator is expressed as:

$$Z_{hls} = (Z_r^{-1} + TZ_s^{-1})^{-1} \quad (4)$$

Equation (3) characterise the input-output dynamics of the simulator and the robot prototype with the simulator in the loop. This model will be used to address performance and robustness issues of hardware-in-the-loop simulation using a position control algorithm.

2.2 Performance Analysis

In the case where a position control algorithm inspired from the force reflecting master-slave configuration is used, the ground robot is inserted between the space robot simulator and the environment. The ability of the hardware-in-the-loop simulator to reproduce faithfully the dynamic behaviour of the space robot in terms of end-effector position and contact force is limited by the ability of the ground robot to track the simulator both in free-motion and during contact tasks.

2.2.1 Free-Motion

In the free-space, the ability of the ground robot to track the position of the space robot simulator is completely determined by the closed-loop performance of the position control system.

$$X_r(s) = T(s)X_s(s) \quad (5)$$

To ensure perfect tracking of the space robot position by the ground robot, it is sufficient to design the ground robot position controller such that $T(s) \approx 1$ within the bandwidth of interest.

2.2.2 Contact Tasks

Let us define the environment impedance as follows:

$$Z_e = \frac{F}{\dot{X}} \quad (6)$$

In contact the closed-loop response of the HLS is given by

$$\frac{X_r}{\dot{X}_{des}} = \frac{TG}{1 + [TZ_s^{-1} + Z_r^{-1}] Z_e} \quad (7)$$

If the space robot were to contact the surface directly, its closed loop response would be

$$\frac{X_s}{\dot{X}_{des}} = \frac{G}{1 + Z_s^{-1} Z_e} \quad (8)$$

Therefore, an extra condition for obtaining the same response in contact is given by

$$Z_r^{-1} = (1 - T) Z_s^{-1} \quad (9)$$

This is equivalent to setting the condition that:

$$Z_{hls} = Z_s \quad (10)$$

Designing the force feedback law of the rigid robot to meet this condition is referred to as *impedance matching*. For linear systems, this may seem practical. For nonlinear systems such as robots, however, the design of the law requires the linearisation of the dynamics around the operating point and the online redesign of the control law.

The impedance Z_s^{-1} is a complex transfer function representing the modelled flexible robot. Therefore, the roots of $1 + [TZ_s^{-1}] Z_e$ for various gains of $Z_e(s)$ and $T(s)$ (changing the stiffness of the environment or/and the bandwidth of the position controller) may be very different than the roots of $1 + Z_s^{-1} Z_e$. In fact, figure 2.2.2 shows an example of the root locus plot for a flexible beam contacting the environment and the one of the HLS counterpart where a rigid beam is used to follow the flexible beam tip motion. In the HLS case, a fixed linear controller resulting in a fixed transmissibility $T(s)$ of the rigid beam is used. The environment stiffness is the varying parameter. The results clearly shows that HLS is conditionally stable while the real system is unconditionally stable. This represents a main limitation for the master-slave approach.

3 Cartesian Feedback Linearisation

In the light of the tracking problem of the position control approach, a new control algorithm is proposed. In this scheme, the dynamics of the ground

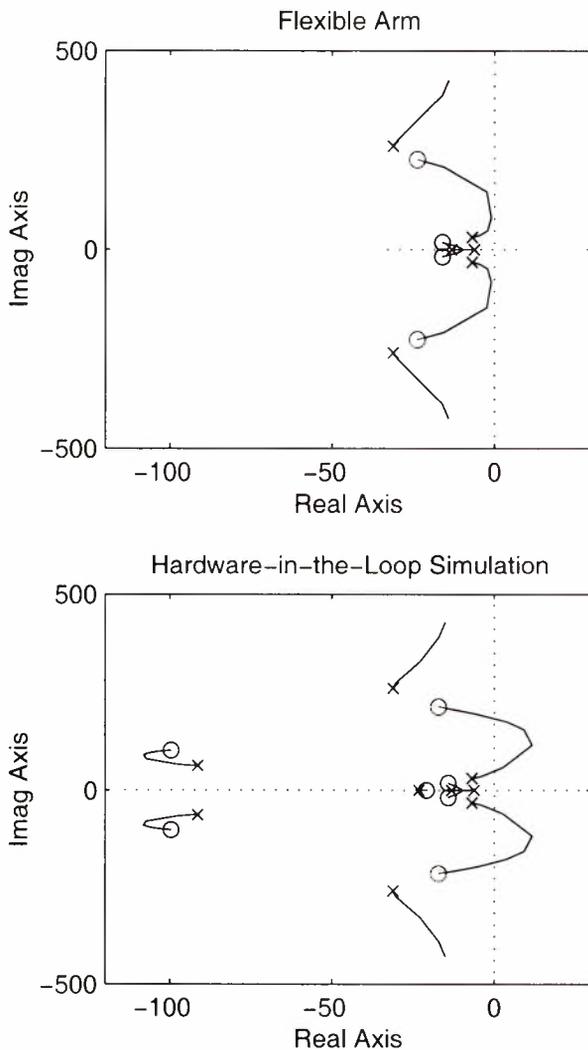


Figure 2: Effect of HLS on Stability for Master-Slave Approach

robot is linearised and the ground robot is controlled in cartesian acceleration [3]. A linear compensator is added to handle errors in the cartesian linearisation of the ground robot dynamics.

3.1 Dynamic Modelling in Cartesian Space

Let an n -DOF manipulator operates in the 6-dimensional Cartesian coordinates. Its kinematics is described by the following equations.

$$\begin{aligned} \mathbf{x} &= \Lambda(\mathbf{q}) \\ \dot{\mathbf{x}} &= \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \\ \ddot{\mathbf{x}} &= \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} \end{aligned} \quad (11)$$

Where $\mathbf{x} \in \mathbb{R}^6$ is vector of tip position/orientation, $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$ is the generalised joint angle and velocity, $\Lambda(\cdot)$ is forward kinematics and \mathbf{J} denotes the manipulator Jacobian. From (11) the joint accelerations

can be expressed as

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \quad (12)$$

The manipulator dynamics can be modeled by,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} - \mathbf{J}(\mathbf{q})^T \mathbf{f}, \quad (13)$$

where $\mathbf{M}(\mathbf{q})$ is the manipulator inertia, $\mathbf{h}(\mathbf{q})$ represents the vector on nonlinear terms including Coriolis, centrifugal, gravity, and friction torques, \mathbf{u} is the vector of joint torque and \mathbf{f} is the generalised force perturbation acting on the end effector. Substituting $\ddot{\mathbf{q}}$ into (13) yields

$$\bar{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{x}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{u} - \mathbf{J}^T \mathbf{f}, \quad (14)$$

where

$$\begin{aligned} \bar{\mathbf{M}} &\triangleq \mathbf{M}\mathbf{J}^{-1} \\ \bar{\mathbf{h}} &\triangleq \mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{h} \end{aligned} \quad (15)$$

3.2 Linearisation Control Law

Let us define the following dynamic equations for the ground robot dynamics and the space robot dynamic simulator.

$$\bar{\mathbf{M}}_r(\mathbf{q}_r)\ddot{\mathbf{x}}_r + \bar{\mathbf{h}}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) = \mathbf{u}_r - \mathbf{J}_r^T(\mathbf{q}_r)\mathbf{f}, \quad (16)$$

$$\bar{\mathbf{M}}_s(\mathbf{q}_s)\ddot{\mathbf{x}}_s + \bar{\mathbf{h}}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s) = \mathbf{u}_s - \mathbf{J}_s^T(\mathbf{q}_s)\mathbf{f}. \quad (17)$$

The following control law is applied to the ground robot,

$$\mathbf{u}_r = \alpha \mathbf{u}_s + \beta, \quad (18)$$

so that the ground robot and the space robot simulator robots have the same dynamic behaviour in Cartesian space, i.e. $\ddot{\mathbf{x}}_s = \ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}$.

Substituting the control law from (18) into equation (16) yields

$$\alpha^{-1}\bar{\mathbf{M}}_r\ddot{\mathbf{x}}_r + \alpha^{-1}\bar{\mathbf{h}}_r = \mathbf{u}_s + \alpha^{-1}\beta - \alpha^{-1}\mathbf{J}_r^T \mathbf{f} \quad (19)$$

The nonlinear feedback gain, $\alpha(\mathbf{q}_r, \dot{\mathbf{q}}_r)$, and offset, $\beta(\mathbf{q}_r, \dot{\mathbf{q}}_r)$, can be found by equating the terms in equations (17) and (19). Therefore,

$$\alpha = \bar{\mathbf{M}}_r \bar{\mathbf{M}}_s^{-1}, \quad (20)$$

and,

$$\beta = \bar{\mathbf{h}}_r - \bar{\mathbf{M}}_r \bar{\mathbf{M}}_s^{-1} \bar{\mathbf{h}}_s + (\mathbf{J}_r^T - \bar{\mathbf{M}}_r \bar{\mathbf{M}}_s^{-1} \mathbf{J}_s^T) \mathbf{f} \quad (21)$$

Figure 3 shows the realisation of the control law given by (18). By defining a new control input \mathbf{u}'_r , one can partition the given control law in two parts. The first part, depicted by the dashed box entitled Space Robot Simulator in Figure 3, is the calculation of the forward dynamics of the space robot simulator,

$$\mathbf{u}'_r = \ddot{\mathbf{x}}_s = \bar{\mathbf{M}}_s^{-1} [\mathbf{u}_s - \bar{\mathbf{h}}_s] - \bar{\mathbf{M}}_s^{-1} \mathbf{J}_s^T \mathbf{f}. \quad (22)$$

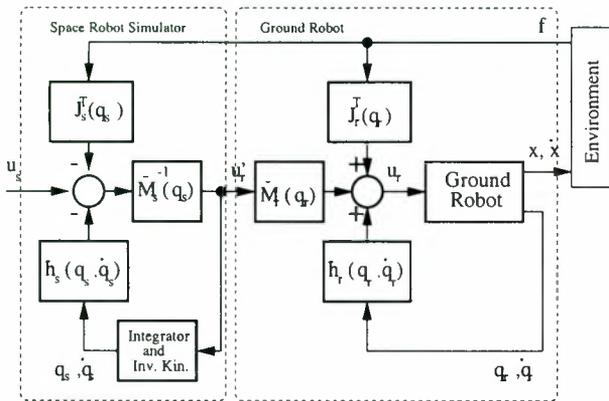


Figure 3: Schematic of the nonlinear control.

The second part of the control law is the linearisation of the ground robot dynamics,

$$\mathbf{u}_r = \bar{\mathbf{M}}_r \mathbf{u}'_r + \bar{\mathbf{h}}_r + \mathbf{J}_r^T \mathbf{f} \quad (23)$$

Combining equations (16), (17), (23) and (22), it is easily shown that the ground robot has exactly the same dynamic behaviour as the space robot simulator in the absence of linearisation errors.

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_s \quad (24)$$

3.3 Linearisation Error Compensator

To compensate for errors in the linearisation of the ground robot, a proportional-derivative compensator is added to ensure that the ground robot tracks the motion of the space robot dynamic simulator.

Presuming the presence of errors in the linearisation of the ground robot dynamics, its resulting motion is then described by:

$$\ddot{\mathbf{x}}_r = \mathbf{u}'_r + \Delta = \ddot{\mathbf{x}}_s + \Delta \quad (25)$$

To alleviate the linearisation errors represented by Δ , a compensator is added to the Cartesian feedback linearisation controller. This can be expressed as a modification to the reference signal fed to the Cartesian feedback linearisation controller as follows:

$$\mathbf{u}'_r = \ddot{\mathbf{x}}_s + \mathbf{K}_v(\dot{\mathbf{x}}_s - \dot{\mathbf{x}}_r) + \mathbf{K}_p(\mathbf{x}_s - \mathbf{x}_r) \quad (26)$$

Substituting (26) into (25), the following closed-loop behaviour is obtained for the error.

$$(\ddot{\mathbf{x}}_s - \ddot{\mathbf{x}}_r) + \mathbf{K}_v(\dot{\mathbf{x}}_s - \dot{\mathbf{x}}_r) + \mathbf{K}_p(\mathbf{x}_s - \mathbf{x}_r) = \Delta \quad (27)$$

This equation shows that for bounded errors in linearisation, it is possible to design a linear compensator such that tracking errors between the space robot simulation and the ground robot are bounded.

4 Experimental Results

Both algorithms presented in this paper were verified experimentally using a one degree of freedom system.

The results obtained supports very well the analysis presented.

4.1 Experimental Setup

The experimental setup is schematically represented in Figure 4. A DC motor coupled to an harmonic drive is used to control the position of a rigid beam in the horizontal plane. The position of the motor is measured using a relative encoder located on the motor side and the contact force using a one dof force sensor (FMS) located on the environment side. The joint velocity commands are generated by an operator through a hand controller. The operator commands, as well as the measured contact loads, are supplied directly to a controller driving a simulated flexible manipulator. The controller of the rigid beam receives the simulated tip acceleration, velocity and position, and uses them for defining the desired motion for the rigid beam.

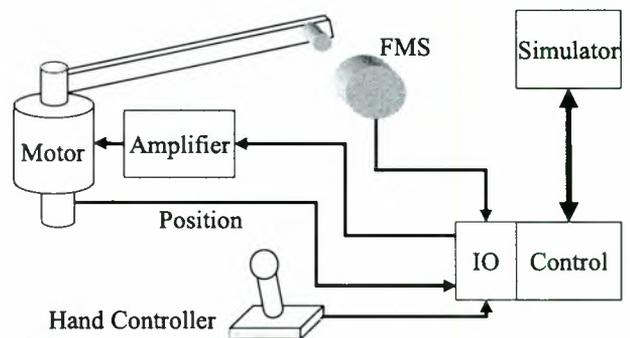


Figure 4: One DOF Prototype Setup

To track the tip motion of the simulated flexible beam, a 3 DOF hardware would be required. Since the rigid beam system has only one DOF, the tracking capabilities are limited. In the experiment presented here, the coordinate θ_{eq} defined in Figure 5 is used as the common coordinate between the simulator and the hardware. It is defined as the angle made by the line passing through the motor axis of rotation and the point on the tip that can make contact with the flat wall.

4.2 Results with Position Control

For the first control algorithm, experimental results were obtained with and without impedance matching.

4.2.1 No Impedance Matching

The results without impedance matching are shown in Figures 6 and 7. During the five seconds of non-contact motion, the tracking error between the hardware and the simulator remains small. In contact,

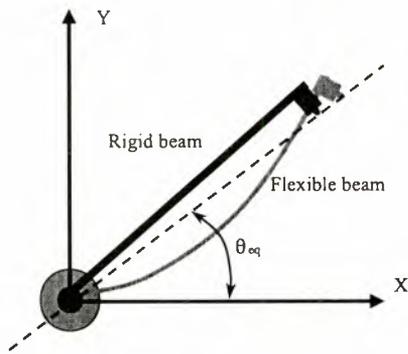


Figure 5: Definition of the tracking angle

however, the tracking error becomes large and constant. This steady-state error, predicted using equation 7 and 8, results from the difference in impedance between the HLS and the simulated system.

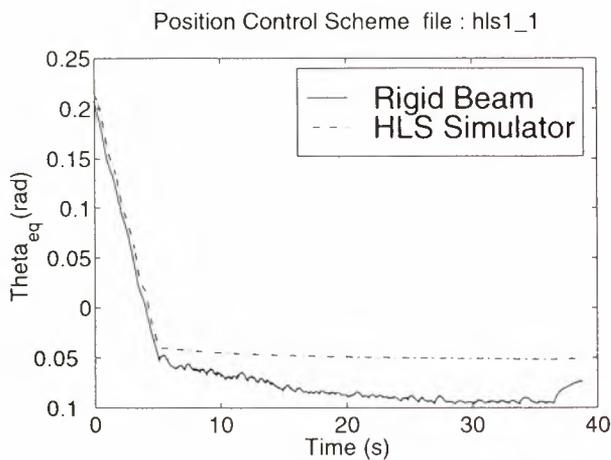


Figure 6: No impedance Matching - position vs time

4.2.2 With Impedance Matching

For the matched impedance case, a force feedback filter was designed and added to the rigid beam control scheme to shape the close-loop impedance and match the flexible beam simulator impedance. This design task requires the complete knowledge of both the master (simulated flexible beam with its control laws) and the slave (rigid beam with its position controller) systems. The results are shown in Figure 8 and 9. Tracking performance in free motion is not affected, but it is improved substantially in contact.

The overall results basically demonstrate that the control algorithm based on the force-reflecting master-slave approach requires impedance matching. Since the matching condition requires a-priori knowledge of the simulated system, the approach is limited in applications.

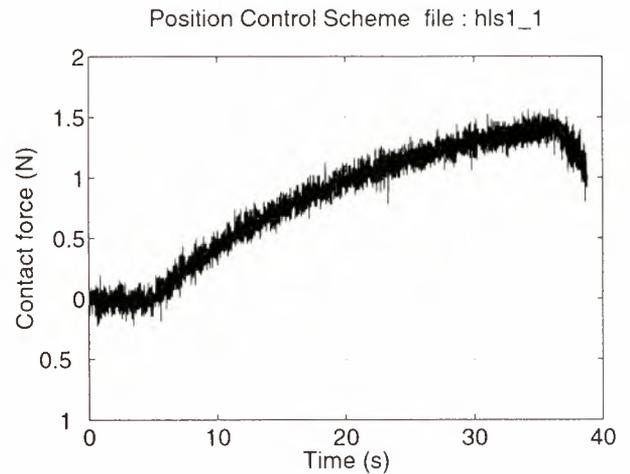


Figure 7: No impedance Matching - contact force vs time

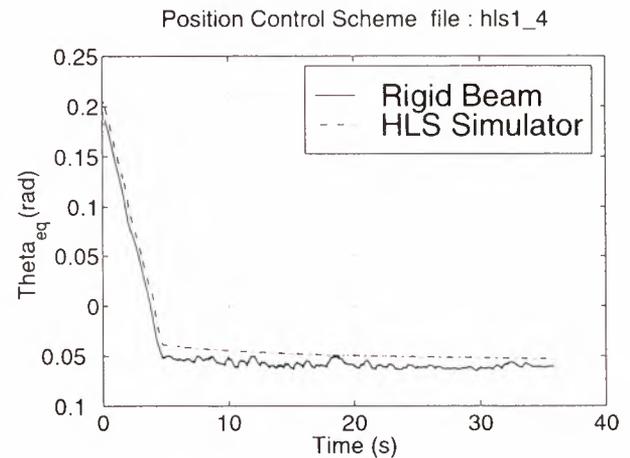


Figure 8: With impedance Matching - position vs time

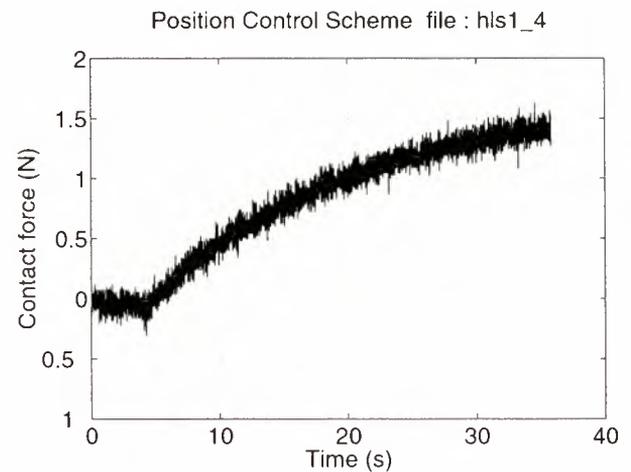


Figure 9: With impedance Matching - contact force vs time

4.3 Feedback Linearisation

The second approach was also demonstrated using the same experimental setup. In addition, pure simulation results were obtained replacing the hardware by a simulation model of the contact dynamics. The results are shown in Figure 10 and 11. The tracking results are good and the control scheme for the rigid beam does not imply any a-priori knowledge of the flexible beam simulator. The experimental contact force also compares very well with the simulated one. The error in tracking is primarily the result of the errors in compensating friction in the motor/drive system. The same observation explains the difference between the contact times in the hardware and in the simulation. These preliminary results suggest that the second scheme is more appropriate for controlling the hardware for a hardware-in-the-loop simulator, and that the performance is basically linked to the knowledge of the hardware to be linearised.

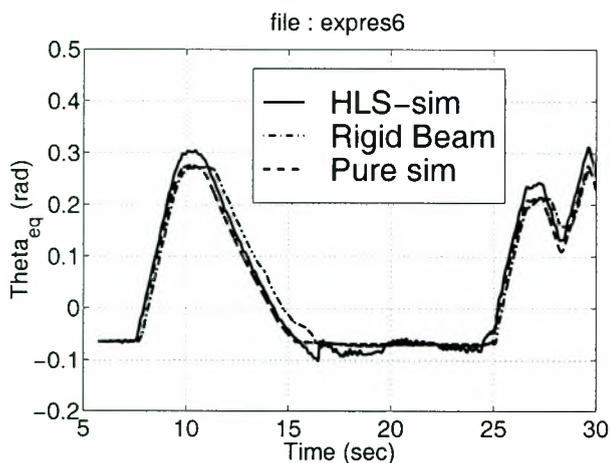


Figure 10: Cartesian Feedback Linearisation - position vs time

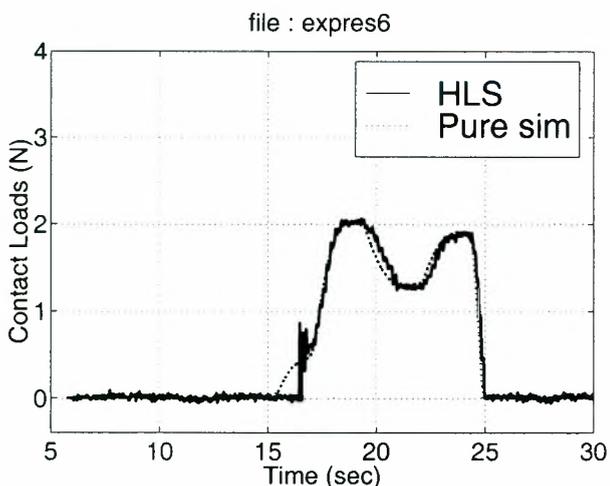


Figure 11: Cartesian Feedback Linearisation - contact force vs time

5 Conclusion

Hardware-in-the-loop simulation includes a wide variety of applications. In the case of the SPDM Task Verification Facility, a non-representative hardware is used to emulate the behaviour of a simulated system. The control system design problem generated by this application is conceptually very similar to the typical master-slave control problem. A main difference, however, lies in the objectives of the design. While for the master-slave system, the objective is to obtain stable close-loop response meeting a given performance criteria, the objective for the hardware-in-the-loop simulation is the complete transparency of the hardware. This paper has demonstrated that in the standard force-reflecting master-slave control, the complete transparency is achieved by matching the impedances between the HLS and the simulated system. This impedance matching condition is realised by properly shaping the force feedback loop. The concept was demonstrated successfully through analysis and experiments.

To alleviate the problems associated with shaping force feedback loop, this paper contributed another approach to obtain complete transparency. Using feedback linearisation, the dynamics of the hardware is linearised and decoupled in Cartesian space, providing complete transparency for cartesian motion assuming perfect linearisation. This approach has the advantage that transparency is achieved independently from the simulated system. It was also shown that for bounded linearisation errors, the tracking error is bounded as well. Experiments results demonstrated the potential of the method.

References

- [1] CSA. Special purpose dextrous manipulator (spdm) task verification facility (stvf) concept document. Technical Report CSA-SS-CD-0016, Canadian Space Agency, 1998.
- [2] E. Dupuis, M. Doyon, F. Aghili, and J. Piedboeuf. Control system prototyping of the stvf manipulator test-bed. In *1st IFAC workshop on Space Robotics*, Montreal, Canada, October 1998.
- [3] N. Hogan. Stable execution of contact tasks using impedance control. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1047 – 1054, Raleigh, N.C., 1987.

EXPERIMENTAL VALIDATION OF CONTACT DYNAMICS SIMULATION OF CONSTRAINED ROBOTIC TASKS

J. Van Vliet and I. Sharf

Department of Mechanical Engineering
University of Victoria, British Columbia, Canada V8W 3P6
phone: (250) 721-6035, e-mail: isharf@uvic.ca

O. Ma

Macdonald Dettwiler Space and Advanced Robotics Limited
9445 Airport Road, Brampton, Ontario, Canada L6S 4J3
phone: (905)790-2800 ext.4535, e-mail: oma@spar.ca

ABSTRACT

Dynamics simulation plays a key role in the design, verification, and operation planning of the International Space Station manipulator systems because of the difficulties of ground-based physical tests with large flexible robotic systems. Modeling of contact dynamics has become essential to dynamics simulation of space station robotic operations (such as the assembly and maintenance of the station). To meet its mandate, the modeling and simulation tool must be of very high fidelity. This paper describes a research project aimed at experimentally validating a general contact dynamics simulation software developed by Macdonald Dettwiler Space & Advanced Robotics Ltd. (previously Spar Aerospace Ltd.). The experimental tests were carried out in the robotics laboratory at the University of Victoria. The validation results demonstrated that the software is capable of predicting realistic contact behavior during constrained robotic operations.

1. INTRODUCTION

Future applications of space manipulators will require execution of complex robotic operations involving constrained motions, such as Orbital Replaceable Unit (ORU) exchange and the assembly of the International Space Station. The contact objects may have complicated interface geometries, various physical properties, and arbitrary operational maneuvers. Ground-based physical testing of these operations with the entire robotic system will be extremely difficult. As a result, validated simulations become a prerequisite for the development of the corresponding control systems and the study of the operation missions.

Modeling contact dynamics is one of the most difficult aspects of developing a generic dynamics simulator for simulating robotic operations. Over the last few years, Macdonald Dettwiler Space & Advanced Robotics

Limited (MDSAR) has developed a contact dynamics modeling and simulation tool as part of the Manipulator Development and Simulation Facility (MDSF). MDSF is a large software package developed by MDSAR for simulating general flexible multibody systems [1]. The package is currently employed for design, verification, operation planning, and engineering analysis of the International Space Station robotic systems: the Space Station Remote Manipulator System (SSRMS) and the Special Purpose Dexterous Manipulator (SPDM). Both robots are being built by MDSAR as a prime contractor to the Canadian Space Agency (CSA).

MDSF has undergone several validation exercises in the past and its simulation models of the SSRMS and SPDM have been accepted as truth models for the International Space Station Program. One previous validation of the MDSF's *contact dynamics* capability used the experimental results available in the literature [2]. Responses simulated by MDSF were compared with the peg-in-hole experiment where a cylindrical peg is slowly inserted in a hole with a milling machine. A detailed description of this validation has been reported in [3] where excellent agreement between MDSF and experiments is demonstrated.

In this paper, we report the results of a collaborative research project between University of Victoria and MDSAR on further experimental validations of MDSF's contact dynamics capability. This research was motivated by the need to expand the previous experimental validation to more complicated and diverse contact scenarios carried out with a robotic arm. The particular objectives of this research are stated below:

1. validate the contact dynamics methodology implemented in MDSF for robotic insertion of pegs of different geometries, materials and different insertion trajectories;
2. investigate the modeling effects of varying contact and manipulator parameters.

The experimental work to meet the objectives of the research has been conducted on the planar robotics facility at the University of Victoria [4]. The arm employed for the peg-insertion experiments has three degrees of freedom and is actuated by Harmonic Drive motors. An instrumented contact interface was added to the facility, as described in Section 2 of the paper. To simulate the peg insertion experiments, a model of the robotic arm, the actuators and contact interface was created in MDSF (see Section 3). In Section 4, we briefly discuss the validation of the manipulator model (arm and actuators) with unconstrained maneuvers under closed-loop and open-loop control. The key part of the paper, Section 5, contains experimental and simulation results for tasks with contact--the impact and peg insertion maneuvers. In the former, the peg is commanded to strike the side walls of the hole, thus allowing validation of the *impact* modeling capability of MDSF. A summary of all peg insertion experiments carried out with the test-bed is presented with results shown for a particular test-case. The plots illustrate the contact forces from simulation and experiment as well as the response of the arm. The paper concludes with comments on MDSF's overall capability and fidelity to simulate diverse contact scenarios for constrained robotic tasks.

2. EXPERIMENTAL TEST-BED

The experiments were conducted on the University of Victoria's robotics test facility which houses three robotic arms on a glass-topped table [4, 6]. The arm employed for the current work was configured with three Harmonic Drive actuators joined by aluminum square-section links. A contact dynamics interface was designed and consists of the special-purpose payload and an instrumented hole fixture. The payload, shown in Fig. 1, includes a Remote Center of Compliance device (RCC) that adds desired passive compliance about the peg tip. This compliance is necessary to facilitate the insertion operations without the involvement of a force control strategy. The end-effector is also instrumented with a planar force sensor to measure the forces and moment exerted on the peg. The range and bandwidth of the force sensor is ± 222 N and 220 Hz. The peg, shown in Fig. 1 (also visible in Fig. 2), was designed to mimic the contact interface of RPCM which is one of the typical ORUs to be handled by SPDM.

The hole fixture, shown in Fig. 2, houses the 10 cm hole walls, hole force sensors and depth-of-insertion sensor. The hole is defined by two interchangeable hole profile pieces which are bolted to the inside faces of the load beams (flexures). Hole width is adjusted with spacers to enable insertions at different clearance ratios. On the outside faces, the load beams rest against compression load cells. Combined with the peg force sensor

measurements, these allow determination of all contact forces when the peg is in contact with both sides of the hole (two-point contact). The depth of insertion sensor is a linear potentiometer which measures the deepest distance of any point on the peg. The sensor resolution is 0.1 mm in the measurement range of 100 mm.

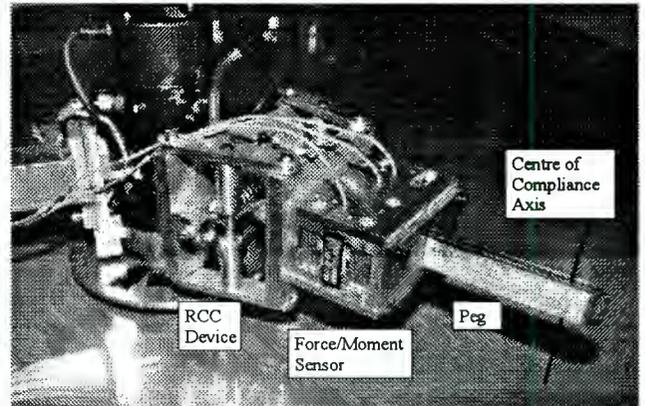


FIG. 1: Wrist joint, RCC, force/moment sensor, and peg

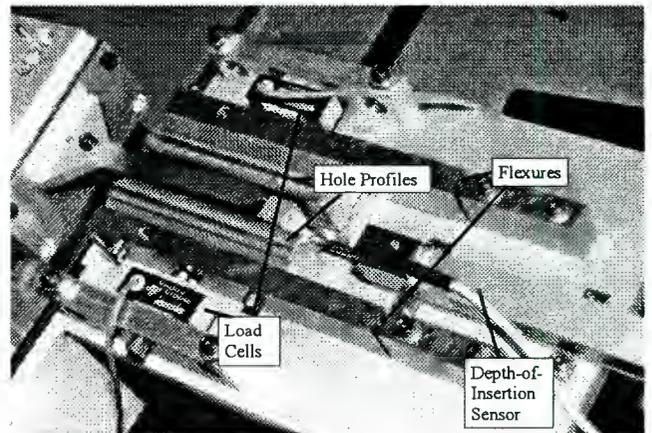


FIG. 2: Hole fixture assembly and an inserted peg

3. TEST-BED MODEL

To simulate the contact experiments with the facility described in Section 2, a model of the test-bed has been implemented in MDSF. As well, control code was developed to mimic the control module driving the physical test-bed. The MDSF model of the test-bed includes three major components: the robotic arm, the actuator dynamics and contact interface.

3.1 Arm model

The arm modeling involved defining the geometric, inertial and kinematic parameters for the component bodies of the robot. A schematic drawing of the arm subdivided into 22 component bodies is shown in Figure

3. It is noted that all component bodies were modeled as rigid with the exception of the rotational compliance of the RCC. Joint compliance was handled separately in the actuator dynamics code (see section 3.2).

The 22 component bodies were synthesized into five manipulator bodies, with two additional bodies describing the table and the hole fixture. These are shown in Figure 4 where we also annotated the articulations with $n-p$, where n is the body number and p is the node number. The articulation type is indicated by one of the following: a solid square (locked joint), empty circle (transient articulation joint), or solid circle (1-dof revolute joint).

3.2 Actuator Model

The actuator model was based on that in [5], combined with the results from joint identification experiments conducted in-house and manufacturer's specifications. Several experiments were carried out with individual joints to quantify the following actuator characteristics:

1. conversion from commanded rotor torque (in counts) to actual torque at the rotor;
2. startup (Coulomb) friction for gearbox;
3. load dependent Coulomb friction;
4. rate dependent (viscous) friction;
5. gearbox stiffness.

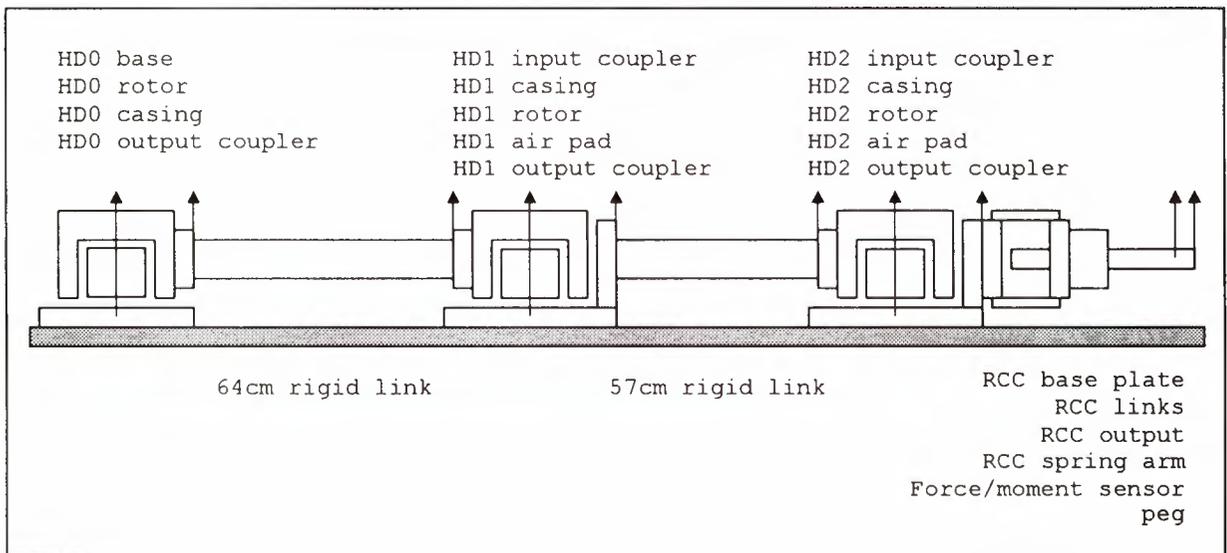


FIG. 3: Component bodies of the manipulator model

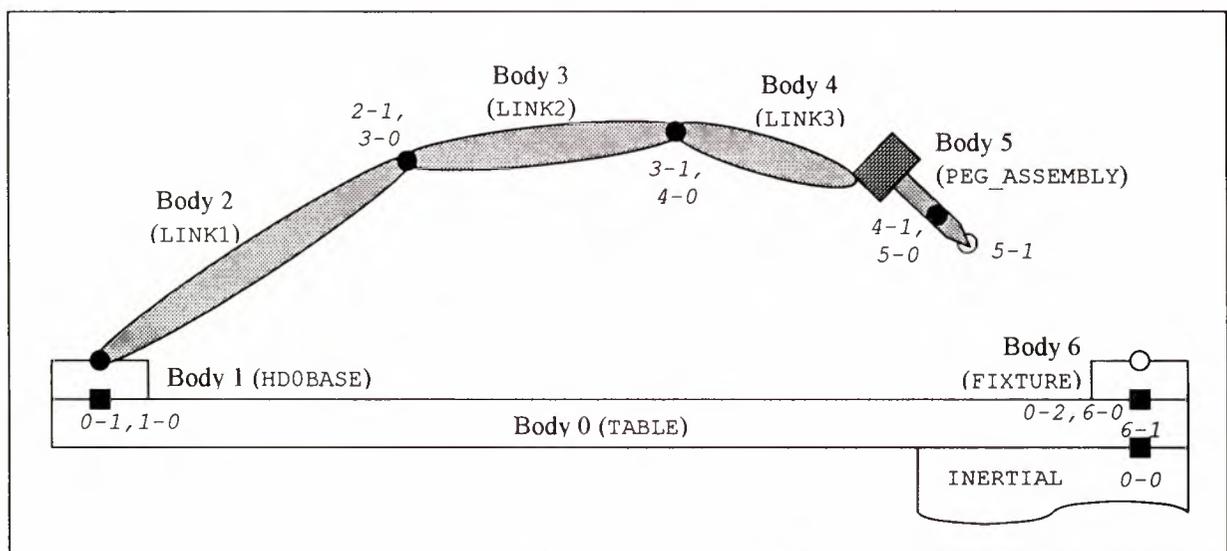


FIG. 4: Topology of the assembled manipulator model

The resulting actuator model includes the rotor inertia, gear ratio, gearbox flexibility, and friction. In accordance with manufacturer's specifications, the harmonic gearbox compliance is modeled as a three stage piece-wise linear spring with experimentally identified stiffness values. The static friction model employed here includes the load-dependent term, although it is nominally set to zero in the simulations conducted with MDSF. One unique aspect of the friction model is the *distributed* static friction in the gearbox. This was necessary to match the rotor angle vs. commanded torque curves observed when each joint was loaded and unloaded.

It is noted that due to the complexity of the identified actuator dynamics, the code implementing the actuator model was written separately and integrated with the rest of MDSF code, similarly to the control code.

3.3 Contact Model

The contact model created in MDSF defines the geometry of the contact bodies (peg and hole walls) as well as contact parameters. The latter represent the stiffness, friction, and damping properties of the contact surfaces. For our model, these were estimated or calculated as described below.

The contact stiffness is dominated by the compliance of the peg force sensor and the hole load cells since it is not included in the model of the manipulator system. The load cell stiffness values are directly available from manufacturer's specifications. These were used to estimate a range of the effective stiffness along the contact normal for peg inserted in the hole. For the majority of the simulations, the value of $0.91e6$ N/m was used. This value was varied by an order of magnitude for MDSF sensitivity studies.

Peg insertion experiments with one-point sliding contact were performed to determine the coefficient of kinetic friction, μ_k , for aluminum and steel. The value of μ_k is employed by MDSF to evaluate the bristle friction model for contact friction [1]. The model also uses the coefficient of static friction which is set to 1.2 times the corresponding kinetic friction. It is widely acknowledged that friction depends on surface properties, such as roughness, cleanliness, and other factors, for example, speed of sliding, temperature and humidity. Special care was taken to ensure consistency of surfaces prior to conducting experiments. Under these conditions, the estimated values for steel and aluminum were: $\mu_{k,steel} = 0.22$ and $\mu_{k,Al} = 0.65$. As with contact stiffness, MDSF sensitivity simulations were conducted where μ_k was varied by $\pm 20\%$.

Finally, the contact damping was set to 0.2 for all cases as this parameter was deemed less important for slow peg insertion maneuvers.

4. VALIDATION OF MANIPULATOR DYNAMICS MODEL

Prior to validating the contact dynamics model, it is necessary to validate the manipulator simulation model. To this end, a set of experiments and simulations was conducted with the manipulator moving free of contact. These constituted the *unconstrained* validation tests and allowed us to gain confidence in the modeling of the multibody arm, actuators, and control code. Two types of unconstrained experiments were conducted as follows.

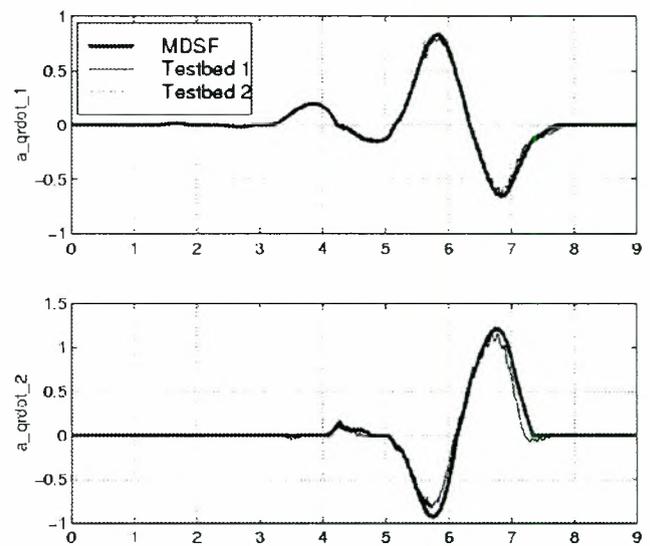


FIG. 5: Joint rates (rad/s) vs. time (sec) from unconstrained open-loop torque test

The first set of unconstrained maneuvers consisted of open-loop tests and was used to assess the accuracy of the inertial and friction elements of the MDSF model. The test involved commanding a smooth torque at a single joint, while other joints remained unactuated. The torque profile for each experiment included three cycles of increasing magnitude with the first peak typically below the static friction torque for the joint. Figure 5 presents the base and elbow joint rates obtained with the torque commanded at the base joint. The wrist is barely excited in this test-case and the corresponding results are not shown. The plots contain two sets of experimental results ('Testbed 1' and 'Testbed 2') and the simulated responses. As can be seen, the experimental curves show excellent repeatability and moreover, are in very good agreement with simulated profiles. There is a consistent pattern where the MDSF rates slightly overestimate the experimental rates. This is likely a result of unmodeled inertia in the system (such as signal and power cables

along the links), as well as small amount of friction between the joint air pads and the table.

In the second set of unconstrained maneuvers, closed-loop PD control is used to move the manipulator through the desired joint-level trajectory. In addition to providing further validation of the arm model, these tests act as a check on the control system implementation. The closed-loop experiments were conducted for a demanding fast maneuver with three sets of proportional feedback gains. Figure 6 displays the commanded torque and joint rate error for the elbow joint, obtained with the full gains. Note again the good agreement between the experimental and the simulated profiles. The visible disagreement between 5 and 6 seconds is immediately after a cusp in the desired joint rates. The agreement for the base joint has similar results, while that for the wrist joint is even better.

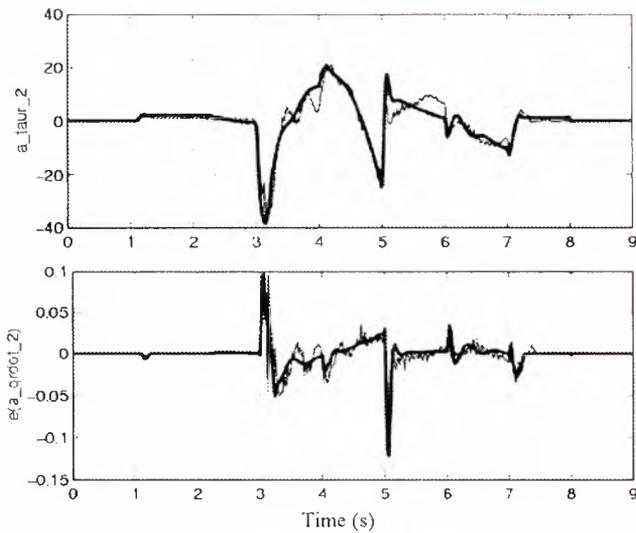


FIG. 6: Elbow joint torque (Nm) and rate error (rad/s) from unconstrained closed-loop torque test

For the unconstrained closed-loop maneuvers, there is little difference in the quality of the agreement between experiment and MDSF's predictions for different proportional gains. This is in contrast to constrained (contact) maneuvers where the agreement improves as proportional gains decrease to quarter of full gain values. This fact was attributed to the compliance in the links which is not modeled in MDSF, but becomes significant when manipulator arm is in contact with its environment (the hole fixture). Because the links are quite stiff, the effect of link flexibility is mitigated by lower joint stiffness caused by reducing the proportional gains of joint servos. This in turn leads to convergence of simulated and experimental results for constrained maneuvers.

5. VALIDATION OF CONTACT DYNAMICS

5.1 Impact Experiments

We begin the contact dynamics validation by presenting results for an impact maneuver where the peg strikes the sides of the hole. The desired Cartesian motion of the peg tip is shown in Figure 7. This maneuver, as well as the peg insertion experiments, is executed under PD joint control where the joint errors are generated from Cartesian motion errors. Since the facility is not instrumented with absolute end-effector position sensing, the 'actual' Cartesian motion is estimated from joint encoder measurements and forward kinematics.

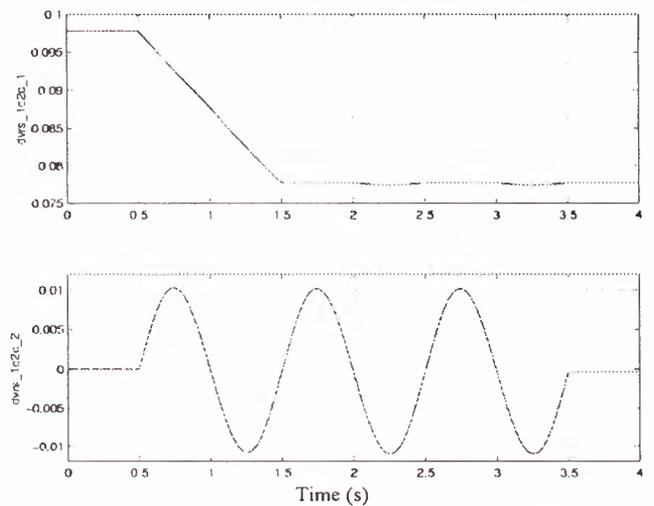


FIG. 7: Cartesian axial and lateral positions (m) of peg tip during impact test

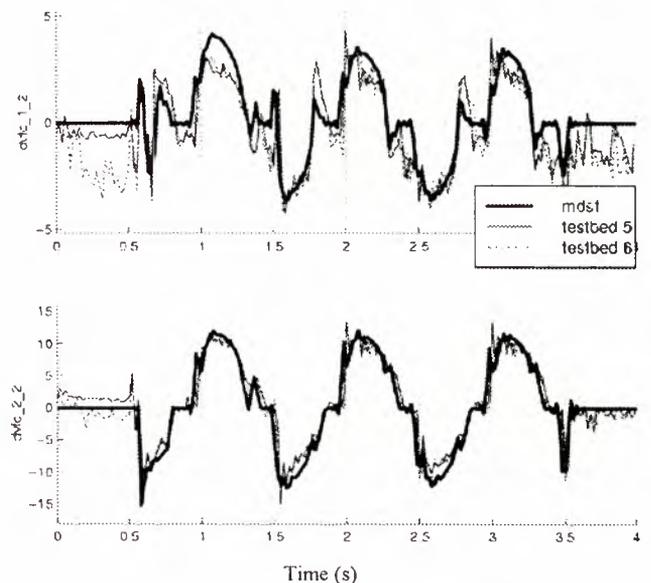


FIG. 8: Axial and lateral forces (N) of the impact test

As in the unconstrained experiments, the constrained tests were conducted with different proportional gains. We also note that all contact experiments were carried out under 'position' control described above, but without force control. It was deemed that the use of force control would mask the contact dynamics which was to be validated. On the other hand, without position feedback, as for instance in feedforward control, it would be difficult to ensure consistent peg insertions. Moreover, in such a situation, the contact dynamics would be contaminated by errors resulting from the inevitable inaccuracies in the dynamics model of the manipulator arm.

The axial and lateral forces on the peg are shown in Figure 8 for the impact test with the soft controller. Two sets of experimental results are presented ('Testbed 5' and 'Testbed 6') to illustrate the repeatability. As can be seen from Figure 8, the peg forces are in excellent agreement between experiment and simulation. The MDSF results are somewhat overdamped which indicates that a lower value for contact damping is more appropriate for the impact maneuver.

5.2 Peg Insertion Experiments

The full spectrum of the peg insertion experiments conducted in the scope of the present project covers the following scenarios:

1. three peg-hole configurations: steel peg-hole (one peg with chamfer and another without chamfer), aluminum peg-hole (one peg without chamfer);
2. three different hole widths;
3. four insertion trajectories distinguished by the initial lateral and angular misalignments of the peg with respect to the hole axis;
4. three different insertion speeds from quasi-static to relatively fast;
5. RCC active or locked where the Remote Center of Compliance was either active or inactive.

In addition, as before, experiments were conducted with different controller gains.

A typical peg insertion experiment consisted of three stages. The first involved moving the peg out of the hole from its home position flush against the right wall. At the end of this stage, the peg had the desired initial misalignments with respect to the hole. The next stage, which represents the beginning of the experiment, involved inserting the peg into the hole, at constant speed. Most experiments were executed at 1 cm/sec insertion speed and this stage took 10 seconds. After a 1 second hold at the end of the insertion, a removal stage was initiated to withdraw the peg from the hole. At the end of the experiment (typically 21 seconds), the arm is relaxed by commanding zero joint torques.

With the parameter variations listed above, we were able to produce a range of contact situations from one-point contact insertion and removal, to two-point contact occurring during insertion and removal, to peg jamming. At the inception of jams, the contact forces increase rapidly which often resulted in load cell overload condition and premature termination of the experiment. In the following, we present results for a insertion and removal maneuver and a jamming situation.

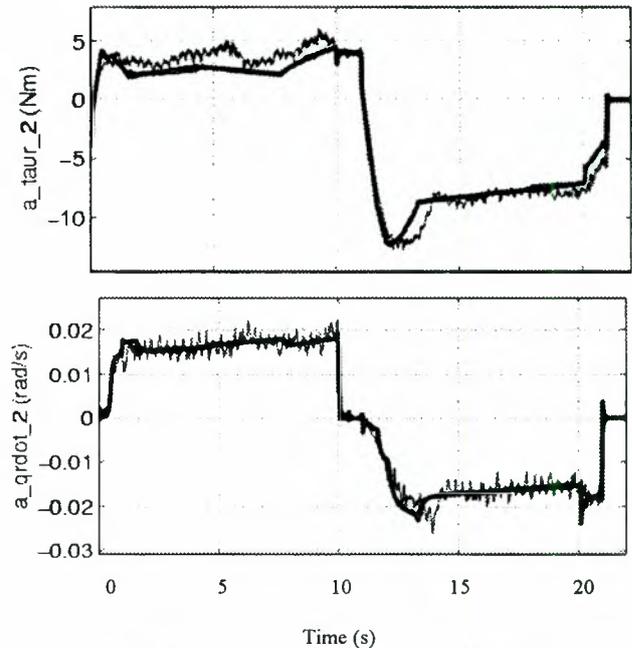


FIG. 9: Elbow Joint Torque and Rate for Peg Insertion Test (RCC active)

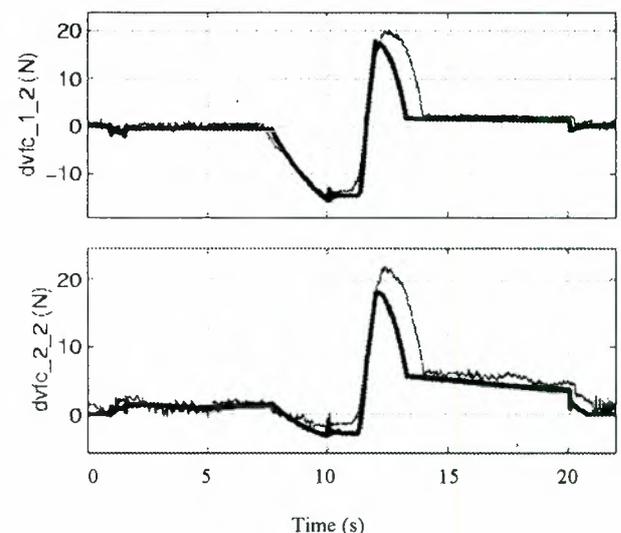


FIG. 10: Axial and lateral forces applied to peg tip during peg insertion (RCC active)

In the peg insertion test considered here, a square steel peg, 12.45mm (0.49") wide, is inserted at 10 mm/sec into a hole of 15.49mm (0.61") wide. The initial misalignment of the peg is nominally a -5 mm lateral offset and 0.12 rad peg angle. The same experiment was conducted with RCC active or inactive, the latter resulting in a jamming situation. Figure 9 shows the commanded joint torque and joint rate response for the elbow joint, while Figure 10 shows the forces on the peg, with RCC active. In this insertion experiment, the peg makes a transition from one-point contact to two-point contact at 7.3 seconds during the insertion stage, and the reverse happens at 14.0 seconds during the removal. The results illustrate very good agreement between simulation and experiment. The somewhat significant discrepancy is observed during the two-point contact stage of the removal (11-14 seconds). This is typical of many test-cases investigated experimentally and in simulation. Our parameter sensitivity studies also indicate that the two-point contact phase is particularly affected by contact stiffness and friction values.

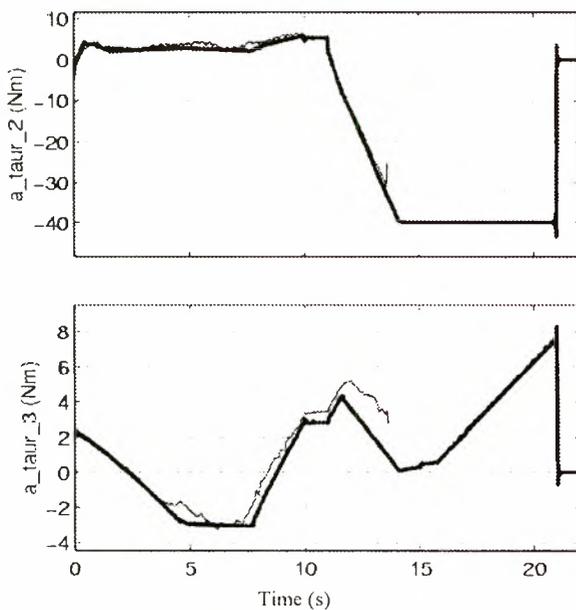


FIG. 11: Elbow and wrist joint torques for peg insertion test (RCC inactive)

The results for the jamming case with RCC inactive are shown in Figure 11, which displays the commanded elbow and wrist torques and Figure 12, exhibiting axial and lateral forces on the peg, as well as the depth of insertion. In this case, the experiment was terminated prematurely (at about 13.3 seconds) because of the aforementioned overload protection measure, which in turn indicates the inception of a jam. The simulation results clearly predict a jamming condition as the peg remains stuck until the nominal end of the maneuver (21

seconds). As can be seen from Fig.12, up until the experiment was terminated, the agreement between the experimental data and simulation results for both tip position and contact forces is very good.

As noted earlier, many peg insertion experiments and simulations with various different conditions have been carried out in this investigation. In addition to demonstrating very good quantitative agreement, our results led to a number of intuitive qualitative observations which hold true for both experiment and MDSF simulation:

- (1) jamming situations are more likely to occur with a stiffer arm (such as with locked RCC);
- (2) occurrence of jams increases when the clearance between the peg and hole decreases or the misalignment between the peg and hole increases.
- (3) jams are more probable during the removal operation, and in fact, at the initiation of a removal;
- (4) for a particular insertion/removal maneuver, removal forces tend to be larger than insertion forces;
- (5) results during two-point contact are much more sensitive to contact stiffness and friction than during one-point contact. As a consequence, similar observations hold for jams.

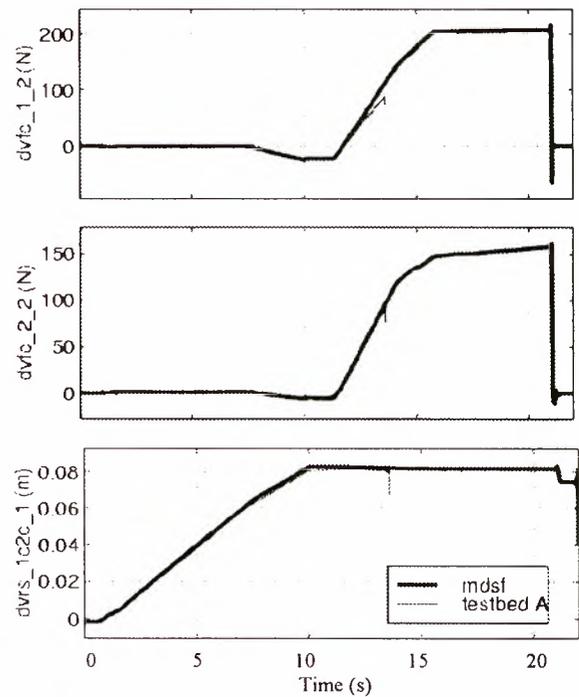


FIG. 12: Top two plots: axial and lateral forces applied to peg tip. Bottom plot: depth of insertion of the peg tip during peg insertion (RCC inactive). The test data ends at about 13.3 s.

6. CONCLUSIONS

In this paper, we presented a selection of the results from the research project on the experimental validation of the general contact dynamics modeling and simulation software developed by Macdonald Dettwiler Space and Advanced Ltd. The venue for the experiments was the University of Victoria robotics facility retrofitted with a specially designed peg-and-hole interface. A model of the manipulator arm was created in MDSF and validated before integrating it with the contact dynamics model. The simulation model was then employed to simulate the experiments conducted with the facility. In general, the agreement between simulation results and experiments is very good. This holds true for a variety of contact geometries, materials, and insertion speeds investigated as part of the objectives of the project. Based on the results of the parameter sensitivity studies conducted with MDSF simulation, we conclude that the quantitative agreement between simulations and experiments could be further improved by tuning parameters of the contact dynamics model and adding manipulator link compliance to the arm dynamics model.

Acknowledgements

Funding support from the National Science and Engineering Research Council (NSERC) and Canadian Space Agency (CSA), through research grant #210114, is acknowledged. Funding and technical support from Macdonald Dettwiler Space & Advanced Limited is also acknowledged.

References

- [1] Ma, O., Buhariwala, K., Roger, N., MacLean, J. & R. Carr, (1997). "MDSF---A Generic Development and Simulation Facility for Flexible, Complex Robotic Systems," *Robotica*, Vol. 15, pp. 49-62.
- [2] Whitney, D.E., (1982). "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, pp. 65-77.
- [3] Ma, O., (1995). "Validation of MDSF Contact Dynamics Simulation against Whitney's Analytical & Experimental Results", *Spar Technical Report CSS-TN.206*.
- [4] Nahon, M., Damaren, C., Goncalves, J. and A. Bergen, [1995]. "Test Facility for Multi-Armed Space-Based Manipulators," *Canadian Aeronautics and Space Journal*, Vol. 41, No. 4, pp. 150-162.
- [5] Bridges, N.M. & D.M. Dawson, [1993]. "Redesign of Robust Controller for RLFJ Robotic Manipulators Actuated with Harmonic Drives," *Proc. American Control Conference*, San Francisco, pp. 2507-2510.
- [6] Van Vliet, J., and I. Sharf, [1998]. "Development of a Planar Macro-Micro Manipulator Facility: From Design through Model Validation," *Canadian Aero. and Space Journal*, Vol. 44, pp. 40-50.

AN END TO END SOLUTION FOR ROBOTIC WORKCELL CALIBRATION AND PERFORMANCES ASSESSMENT

Frédéric Didot¹
Daniele Galardini²
Filip Geuens³

¹ Automation & Robotics Section (MMA), ESA / ESTEC
P.O. Box 299, 2200 AG Noordwijk, The Netherlands
phone: +31-71-565-4403, fax: +31-71-565-5419
email: fdidot@estec.esa.nl

² TRASYS Space, Zaventem, Belgium

³ Krypton NV, Leuven, Belgium

ABSTRACT

This paper describes an integrated approach, including performance assessment, calibration and task parameter updating, for robot calibration technology as a support to predictable, reliable and safe autonomous robotics operations in space.

The paper will discuss (1) performance assessment, i.e. to characterise a system via a well defined set of standardised engineering values (ANSI-RIA, ISO), and (2) calibration, i.e. system modelling and computing the model parameters, of both the environment (called workcell calibration) and the robotics device. Both will make use of hardware and software tailored for either on-ground or in-orbit use as the available resources are quite different.

The paper will also describe the developed H/W and S/W tools which support the calibration technology implementation. The results of the calibration are finally fed back to the pre-programming via an update of the model of the system in the off-line programming system.

1. BACKGROUND

In order to be able to operate Automation and Robotics (A&R) devices in a predictable, safe and cost efficient way, the use of off-line programming techniques is preferred for the mission preparation. In addition, significant communication delays are often an important constraint for motion execution via a closed loop control over the ground segment. Therefore, the European Space Agency (ESA) has been defining the Interactive Autonomy mode of operation, see figure 1, to make the flight segment control loop independent from the communication delay. The interactive mode of

operations consists of a high rate positioning feedback loop closed at the flight segment and a low rate position commanding loop over the ground segment. This mode of operation is insensitive to communication delays, but it relies on an accurate knowledge of the kinematics and dynamics of the A&R devices to guarantee a successful execution. The commanding is based on pre-programming and optimisation of the motion based on a prediction of the behaviour of the flight system.

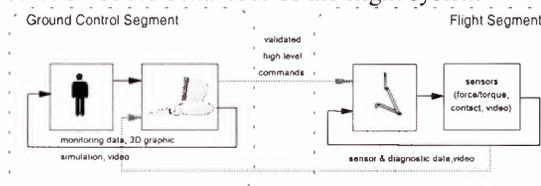


Figure 1: Control loops Interactive Autonomy

As a result effort has been spent in the last few years on robot calibration technology to set-up an integrated approach aiming at modelling A&R devices and determining the model parameters via on-ground and in-orbit performance assessment and on-ground and in-orbit calibration [1,2,3].

2. PERFORMANCE ASSESSMENT

Performance assessment aims at characterising a system via a well defined set of standardised engineering values computed from a standardised measurement set. These values are given by norms such as ANSI-RIA and ISO9283 and additional values are added for application dependent purposes or on the basis of industrial experience. A short list of engineering values is pose accuracy and repeatability, path accuracy and repeatability, cornering overshoot and round-off errors, hysteresis, minimum positioning time (arm oscillations), etc.

To summarise the performance assessment:

- quantifies the system performance characteristics;
- enables to determine whether the requirements for an operation can be met with the chosen system;
- and indicates the need for calibration .

3. CALIBRATION

"Calibration" is the whole process which aims at improving the knowledge of the system; it involves modelling, measuring, identification and model implementation. In general, any calibration strategy consists of four steps:

MODELLING aims at determining how many and which parameters should be used in the robot and world model.

MEASURING aims at measuring robot poses by an external measurement system such that sufficient information for identification is gathered or the system is persistently excited to identify the influence of error sources.

IDENTIFICATION step aims at updating the parameters of the robot model (see figure 2). Corrections on the model parameters are calculated to explain the measurement data. Different techniques can be worked out, such as global identification and the independent axes method.

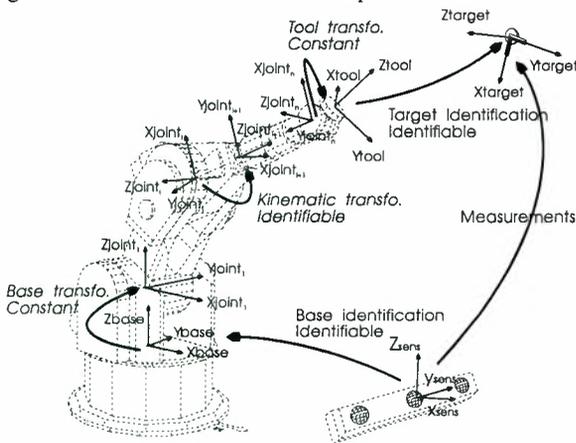


Figure 2: Philosophy of identification

MODEL IMPLEMENTATION will compensate the task specification making use of the improved robot model.

3.1 Robot Calibration

The inaccuracy of a default, manufacturer model of the system explains the difference between the simulated system response and the measured system response. In order to obtain good calibration results, it is necessary:

- to analyse the error sources and to create an appropriate model of the system,
- to assure a persistent excitation of these errors,
- and third to perform an accurate acquisition of the measurements.

The calibration step uses the measurement set in order to update the parameters of the default model and to compute the additional model parameters based. A "user defined" objective function governs the way the parameters are changed. Special attention has to be paid to model the robotic device. The robot model of the system comprises a geometrical model, either expressed in Denavit-Hartenberg [4,6] or Hyrati-Mirmirani [5] coordinates, an actuator model, and a compliance model based on a set of selected primitives.

3.2 Workcell calibration

Besides a well calibrated robot, two other important conditions have to be fulfilled in order to guarantee an accurate relative positioning of the tool w.r.t. the workpiece.

- Firstly, the position and orientation of the workpiece should be accurately known w.r.t. the robot base frame.
- Secondly, the tooltip (TCPF) should be accurately known w.r.t. the robot toolframe.

Workcell calibration aims to update a nominal workcell model in order to fit the actual workcell more closely. The workcell model is composed of the position of the robot base and its tool, the measurement equipment or supporting equipment, the workpiece, and a set of programmed target positions in the robot controller.

The workcell model represents the nominal environment, i.e. the environment 'such as it is supposed to be'. However, the relative position of objects or points and (a) robot(s) in the actual workcell is likely to differ from the nominal workcell model.

Therefore, the parameters of the nominal model in the off-line programming system (OPS) should be updated such that it corresponds to the real environment. For this purpose, not only the link from measurement system to OPS is required, but also the inverse link: the nominal information of the objects to be identified can be extracted out of the CAD. This results in the system layout presented in figure 3.

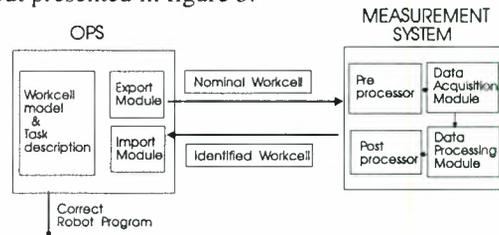


Figure 3: Workcell calibration system layout

This concept is worked out for ROBCAD as the OPS and ENVCAL as the measurement system software.

4. COMPENSATION: UPDATE OF MOTION PLANNING

Finally, the model implementation or compensation (see figure 5) enables to incorporate the calibration results into the motion programming to obtain an enhanced performance. Since most robot controllers do not allow to introduce the identified parameters and since easy, closed form equations for the identified robot will not hold, on-line calculations are not desired. Instead of changing the controller parameters, the setpoints of the operations are transformed to new setpoints, called "fake poses", in order take into account the difference between the identified model and the model on which the controller of the system is based.

Two steps are required: forward and inverse model as shown in figure 5. The inverse of the identified model enables to compute corrected joint values for the robot to reach the desired cartesian pose. The robot model in the controller, i.e. the nominal model, is used to compute the modified cartesian poses, called fake poses, to be send to the controller such that the robot reaches the desired pose.

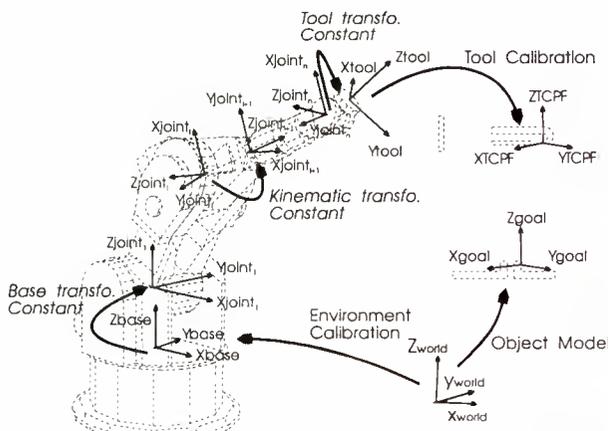


Figure 4: Philosophy of compensation

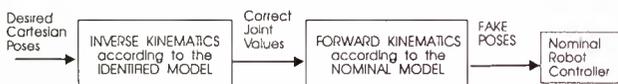


Figure 5: Principle of the Fake Pose method

5. THE RESULTS

5.1 Robot Performances Assessment

S/W has been developed to post-process automatically robot performances data such as pose accuracy & repeatability, path tracking accuracy, pose stabilisation and overshoot, cornering round-off errors. Different robot performance tests were conducted on the JERICO Evaluation Testbed (JET). Three diodes were mounted on the JET tool flange and measurements were acquired

autonomously using the 3D RODYN measurement system (see figure 7).

The results were processed. They showed that the initial JET accuracy was 21.6mm, while its repeatability was 0.5mm. The accuracy figure indicates the need for calibration while the repeatability figure gives a flavour on how "calibratable" was the JET system.

5.2 Robot Calibration

The robot calibration s/w has been developed to calibrate any open kinematic structure, including

Dynamic positioning results	
Mean position:	
X	-0,096 mm
Y	0,203 mm
Z	1,000 mm
Results:	
Mean distance	0,017 mm
Max distance	0,028 mm
1 Sigma	0,006 mm
Repeatability	0,035 mm

Figure 6: robot performance test results example

flexibility's effects. JET which consists of an eight axes robot was a perfect show case to asses the calibration s/w capabilities.

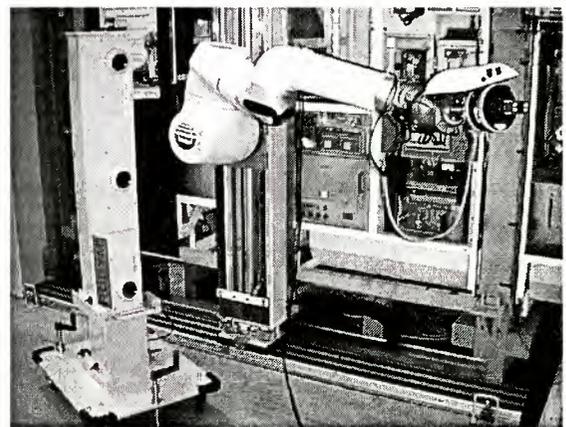


Figure 7: the JET robot and its workcell. On the left side, the RODYN6D measurement device

From the JET performance evaluation, the need for the JET robot calibration was clearly identified. The robot accuracy was 21.6mm while an accuracy of +/-2mm w.r.t the environment was required to perform off-line programming of the JET system. The JET robot calibration was performed in two steps. In the first step, the joint zero offset of each of the eight axis robot were identified. Once identified, the joint zero offsets were up-loaded in the COMAU 3G robot controller. An accuracy performance test was run again, showing an improvement from 21.6mm down to 3.6mm. Despite the robot accuracy was much better than initially, more error

sources identification was required in order to reach the final accuracy goal.

In a second step, a new calibration session was run with more parameters to be identified. Not only different kinematic error sources were introduced such as parallelism, orthogonality or link length errors, but also flexibility's effects were considered. As a matter of fact, using the built-in library of flexible models from ROCAT, three different flexibility's sources contributed significantly to the arm accuracy. These were the elbow and wrist joints flexibility and the robot mounting plate flexibility. Taking into account all the various error sources, a final accuracy of 0.6mm was reached. This final accuracy was cross-checked from an independent set of measurement. Now that the JET robot arm was calibrated, its workcell needed to be calibrated.

6.3. Workcell Calibration

In order to perform "easily" the JET workcell calibration, a so-called space probe was designed. That space probe is a single point measurement device that enables an operator to digitise point in the environment. The space probe is used together with the RODYN 6D measurement device and turns RODYN6D into a portable 3D Coordinated Measurement Machine.

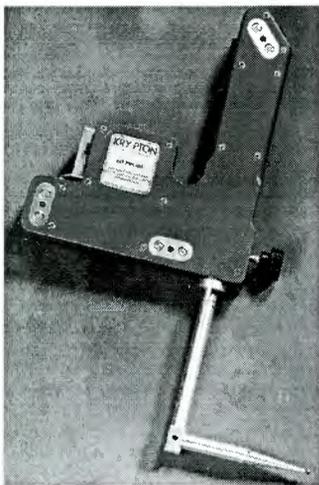


Figure 8: The space probe

The first step is for the operator to define in a script the measurement session procedure. This script is an important step as the object location that needs to be identified shall have a hierarchical structure in the workcell. For instance the sample is in a drawer, the drawer in a rack and the rack within the workcell. It is important to maintain the hierarchical structure to enable e.g. easier database maintenance or exchange of rack. Once the measurement script is written, the operator moves the space probe in the workcell to identify all interesting actuation interface location. Once completed, the measurements are post processed, and a new workcell model is generated in Robcad compatible

format. Robcad is a commercial off-the-shelf off-line programming system used for JET off-line programming. The operator loads the calibrated workcell model into the nominal workcell model, The differences between the model may then be compared. Large differences were noticed between the nominal workcell geometric model of Robcad and the real workcell geometry (up to 5cm in some cases).

6.4. End to End Test

Using the calibrated Robcad model and the calibrated robot model, some poses to be reached by the real JET robot were off-line define, the associated fake pose derived and down loaded to the real robot controller.

The robot controller commanded the end effector to the actual goal pose with an accuracy of about +/-1.5mm. The end-to-end demonstration was successful.

7. CONCLUSION

Performance assessment and calibration enables to describe the system characteristics based on off-line processing techniques and to enhance the system performance by updating the setpoints which drive the motion controlled system. As such these techniques contribute to achieve a safe and reliable autonomous operation execution which is quite valuable especially for, but not limiting to, space applications.

Moreover, future space programs will need highly intelligent and highly autonomous systems due to the limited communication & power resources. Therefore, an accurate knowledge of the system is really necessary to pre-program & validate the operations, to guarantee a predictable execution of them, to optimise the available on-board resources and to limit operator interventions. As such performance assessment and calibration is to be considered as a supporting technology for future space application.

The necessary tools to support performance assessment and robot+workcell calibration have developed. This encompasses the necessary measurement systems, the s/w to process measurement data, the necessary drivers to up-load the robot and workcell calibration results in an Off-Line Programming system.

Experiments of a Space Robot in the Free-Fall Environment

Yasuyuki Watanabe

National Space Development Agency of Japan

Tsukuba Space Center, 1-1, Sengen 2-chome, Tsukuba-city, Ibaraki 305-8505 Japan
phone: +81-298-59-2967, fax: +81-298-52-2410, e-mail: Watanabe.Yasuyuki@nasda.go.jp

Yoshihiko Nakamura

Department of Mechano-Informatics, University of Tokyo

3-1, Hongo 7-chome, Bunkyo-ku, Tokyo 113-8656 Japan

phone: +81-3-5804-6256, fax: +81-3-3818-0835, e-mail: nakamura@ynl.t.u-kyo.ac.jp

Abstract

We propose an experimental system of a space robot in the microgravity environment at the Japan Microgravity Center (JAMIC), where the microgravity—kept as accurate as $10^{-5}G$ for 10 seconds—is generated by 490m-depth free-fall. We developed a system for experiments of free-flyers at the JAMIC. In this paper, we take up two important topics to perform the experiments. The first one is to release a robot stably in the microgravity, and the other is to measure position and orientation of a robot by using cameras set outside of it. Using the experimental system, we performed several experiments of a space robot capturing a target.

1 Introduction

A lot of studies of motion and control of space robots have been carried out theoretically, and at present, realization and operation of space robots in space are investigated by many research groups. Though it is necessary to perform experiments on the ground, the influence of the gravity troubles researchers about credible experiments. One of the methods to simulate a microgravity environment on the ground is using a drop-shaft such as the Japan Microgravity Center (JAMIC). At the center, free-fall of the drop capsule with experimental setups on board generates the microgravity environment as accurate as $10^{-5}G$ for about 10 seconds. This method has the disadvantage of limited space and time for an experiment and the advantage of a robot free from physical contacts with the surroundings. The former is dominant and therefore there are few studies on experiments in the free-fall environment. Our survey found only the study on nonholonomic motion by Iwata et al.[1]. In the study, the robot is autonomous and has sensors and controllers on board, and therefore it is difficult to miniaturize it or to apply it to other practical experiments such as capturing a target.

The purpose of our research is to establish an experimental system of a space robot in the free-fall

environment at the JAMIC. We developed an experimental setup for visual feedback control with two outside cameras and radio remote control. To measure motion of a robot, images of two CCD cameras are processed by the tracking vision system (TRV). In this paper, after describing the outline of the experimental system, we discuss the problem on release of the robot in the microgravity environment, and establish two computational methods to identify the 3D position and orientation of the robot base. The experiments and their results follow to evaluate the developed system.

2 Outline of Microgravity Experiment at the JAMIC

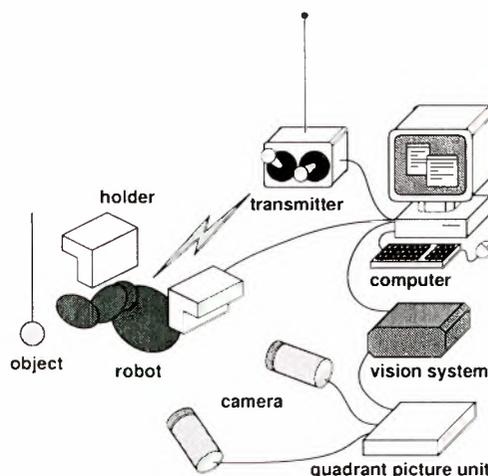


Fig. 1: Experimental System of Visual Feedback Control

The JAMIC has a 710m-depth drop-shaft of which 490m is for free-fall. The drop capsule consists of the inner capsule and the outer one which has a thruster to cancel air drag in falling. The space between them is made a vacuum and the inner capsule falls in the vacuum. Experimental setups are mounted into a

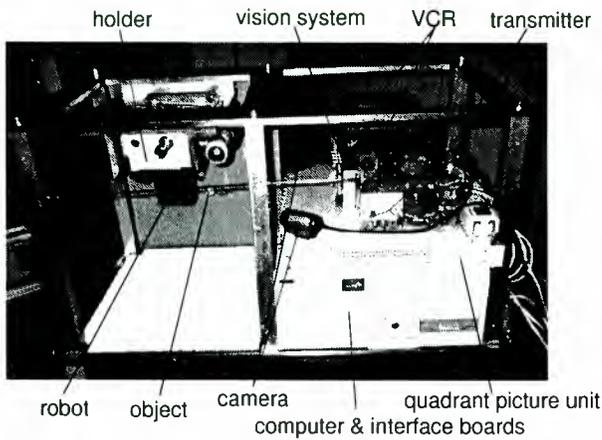


Fig. 2: Assembled Experimental Setup

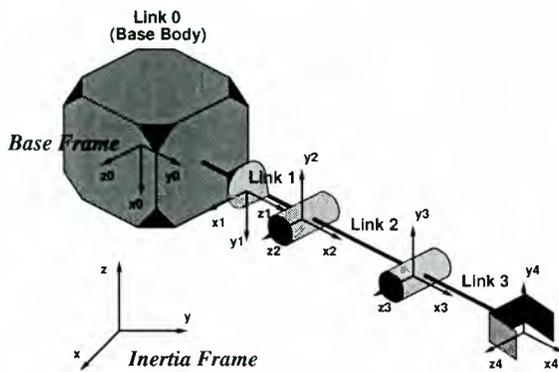


Fig. 3: Experimental Robot with a 3DOF Manipulator

rack which is loaded in the inner capsule. With this system, we can perform experiments in the microgravity environment as accurate as $10^{-5}G$ for about 10 seconds.

Figure 1 describes our developed setup and it is assembled as shown in Fig.2 to be mounted into the rack. The robot has a 3DOF manipulator which is composed by servo-motors, a receiver, and a battery for radio control. Its mechanism is shown in Fig.3 and its total length is approximately 20cm. The sensors and the controllers are put in the outside of the robot. The two CCD cameras are used for three-dimensional measurement. The images from the cameras enter the 'quadrant picture unit' which is used to arrange the two images into one. The output image of the apparatus is processed by the TRV to compute the position of the target and the position and orientation of the robot. The desired joint angles are designed from these data and transmitted to the robot by radio.

Figure 4 describes the sequence of the experiment. Before drop, the robot is held by the supporting mechanism 'holder' under the gravity. When the



Fig. 4: Experimental Sequence of Capturing a Target

controller receives the drop-start signal from the capsule, the robot is released and starts capturing motion. The sequence is finished when the robot accomplishes capturing the target.

3 Releasing a Robot in Microgravity

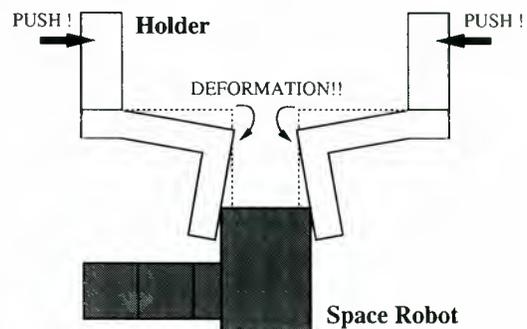


Fig. 5: Deformation of the Supporting Mechanism

One of the critical problems in the microgravity experiment is that the robot suffers force and torque at the contact with the supporting mechanism at the moment of releasing. Large momentum and angular momentum of the robot cause the robot to be

out of control, and therefore it is necessary to keep them small for a stable experiment. In our experiment with the holder shown in Fig.5 which is made of aluminum, we observed the robot moving upward as shown in Fig.6. The cause of the motion is the elastic energy stored in the the holder deformed by the force to support the robot (see Fig.5). To reduce the influence of the elastic energy, we need to choose the shape and material of the holder with high stiffness because elastic energy is inversely proportional to stiffness if holding force is constant. When we designed a sturdy-shaped iron holder, we could release the robot calmly.

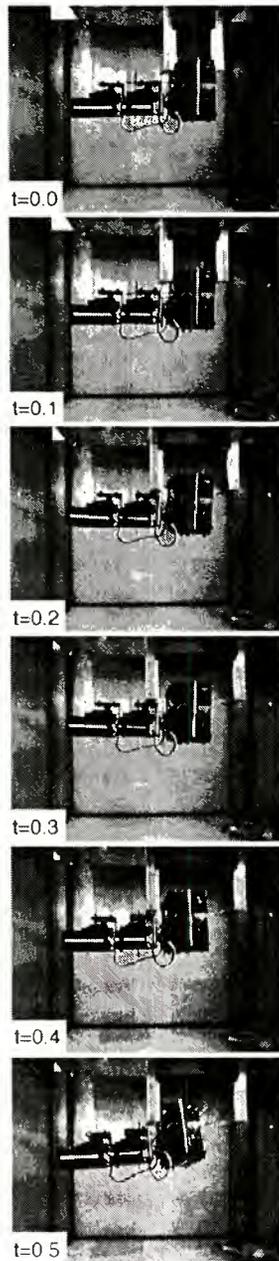


Fig. 6: Upward Motion of the Robot after Released

4 Measurement Methods

The position and orientation of the robot are computed from the data obtained by the TRV processing the images of the two CCD cameras. In this section, we propose two computational methods: (1) stereo-vision method —the 3D positions of plural points marked on the robot base are measured by using the stereo vision and the position and orientation are computed from them—, and (2) non-stereo-vision method —the two cameras observe the different points marked on two surfaces of the robot base respectively and the position and orientation are computed from their images directly, without stereo vision measurement—.

4.1 Stereo-vision Method

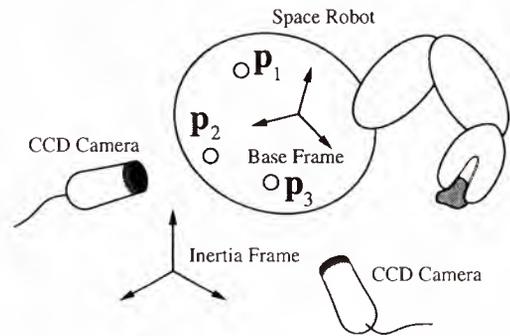


Fig. 7: Two Cameras Looking at the Same Marks

The position and orientation of the robot are computed from the 3D positions of several marks on the base body. The positions of the marks are measured by the stereo vision. The principle of this method is illustrated as follows.

As shown in Fig.7, three points are marked on the surface of the robot base such that they make a triangle. The position of each point in the inertial frame is expressed as follows:

$${}^I\mathbf{p}_i = \mathbf{r}_0 + \mathbf{R}_0 {}^b\mathbf{a}_i \quad (i = 1, 2, 3) \quad (1)$$

where \mathbf{r}_0 and \mathbf{R}_0 denote the position and orientation of the robot base respectively, and ${}^b\mathbf{a}_i$ the position of each point in the base frame. ${}^I\mathbf{p}_i$'s are measured by using the stereo vision (see Appendix A). Equation (1) yields the following equations:

$${}^I\mathbf{p}_2 - {}^I\mathbf{p}_1 = \mathbf{R}_0 ({}^b\mathbf{a}_2 - {}^b\mathbf{a}_1) \quad (2)$$

$${}^I\mathbf{p}_3 - {}^I\mathbf{p}_1 = \mathbf{R}_0 ({}^b\mathbf{a}_3 - {}^b\mathbf{a}_1) \quad (3)$$

\mathbf{R}_0 is orthogonal and therefore the following equation is obtained from Eq.(2) and (3).

$$({}^I\mathbf{p}_2 - {}^I\mathbf{p}_1) \times ({}^I\mathbf{p}_3 - {}^I\mathbf{p}_1) = \mathbf{R}_0 \{ ({}^b\mathbf{a}_2 - {}^b\mathbf{a}_1) \times ({}^b\mathbf{a}_3 - {}^b\mathbf{a}_1) \} \quad (4)$$

\mathbf{R}_0 is computed from Eqs. (2)-(4) which are linearly independent, and \mathbf{r}_0 is computed by substituting the obtained \mathbf{R}_0 into Eq.(1).

4.2 Non-stereo-vision Method

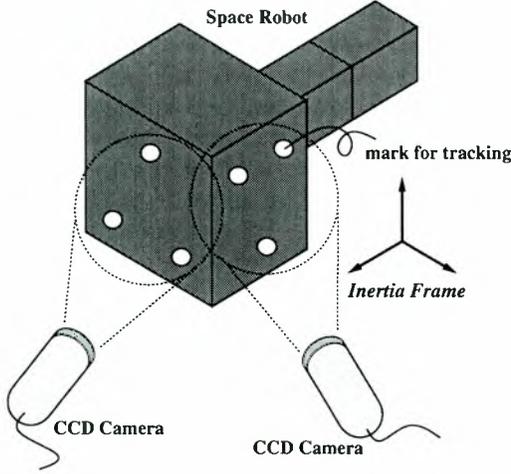


Fig. 8: Two Cameras Looking at the Different Marks

As discussed in the previous section, there is the problem that the holder gives the robot some momentum and angular momentum at the moment of releasing the robot. If the robot rotates and the surface with the marks turns off from the camera vision, it becomes impossible to measure the position and orientation of the robot. In the experiment with the stereo-vision method, the impossibility of measurement occurred as shown in the next section. This problem is inevitable when using such cameras as are set outside of the robot. To expand the measurable region of the robot attitude, we propose the non-stereo-vision method. In the stereo-vision method, the two cameras observe the same marks on the surface of the robot base and therefore the region of the attitude where the marks are simultaneously in the visions of the two cameras is small. In the non-stereo-vision method as shown in Fig.8, the two cameras observe the different points on the two surfaces respectively and therefore the measurable region of the attitude is larger than that of the stereo-vision method.

Three marks for each cameras, altogether six marks, are made on the respective surfaces of the robot base. We call them point- P_{*i} ($* = l, r; i = 1, 2, 3$). The subscript $*$ represents the left camera or the right one observing the mark. The position of each mark ${}^I p_{*i}$ satisfies the following equation (see Appendix A).

$$\mathbf{A}_{*i} {}^I p_{*i} = \mathbf{b}_{*i} \quad (5)$$

where

$$\mathbf{A}_{*i} = \begin{bmatrix} S_{7*}H_{*i} + S_{1*} & S_{8*}H_{*i} + S_{2*} & S_{9*}H_{*i} + S_{3*} \\ S_{7*}V_{*i} + S_{4*} & S_{8*}V_{*i} + S_{5*} & S_{9*}V_{*i} + S_{6*} \end{bmatrix} \quad (6)$$

$$\mathbf{b}_{*i} = \begin{bmatrix} H_{*i} - U_{x*} \\ V_{*i} - U_{y*} \end{bmatrix} \quad (7)$$

and $[H_{*i} \ V_{*i}]^T$ denotes the position of the point- P_{*i} in the image. The position of each point is expressed as follows:

$${}^I p_{*i} = \mathbf{r}_0 + \mathbf{R}_0 {}^b a_{*i} \quad (8)$$

where ${}^b a_{*i}$ denotes the position of the point- P_{*i} with respect to the base frame. Substituting Eq.(8) into Eq.(5), we have

$$\mathbf{A}_{*i} (\mathbf{r}_0 + \mathbf{R}_0 {}^b a_{*i}) = \mathbf{b}_{*i} \quad (9)$$

where

$$\mathbf{r}_0 = [r_{01} \ r_{02} \ r_{03}]^T \quad (10)$$

$$\mathbf{R}_0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (11)$$

$${}^b a_{*i} = [a_{*ix} \ a_{*iy} \ a_{*iz}]^T \quad (12)$$

Since Eq.(9) is a linear equation of the components of \mathbf{r}_0 and \mathbf{R}_0 , it is transformed into the following equation:

$$\tilde{\mathbf{A}} \tilde{\mathbf{r}} = \tilde{\mathbf{b}} \quad (13)$$

where

$$\tilde{\mathbf{r}} = [r_{01} \ r_{02} \ r_{03} \ r_{11} \ r_{12} \ r_{13} \ r_{21} \ r_{22} \ r_{23} \ r_{31} \ r_{32} \ r_{33}]^T \quad (14)$$

$$\tilde{\mathbf{b}} = [b_{l1}^T \ b_{l2}^T \ b_{l3}^T \ b_{r1}^T \ b_{r2}^T \ b_{r3}^T]^T \quad (15)$$

and $\tilde{\mathbf{A}} (\in R^{12 \times 12})$ is easily obtained from Eqs. (6),(7),(9)-(12). From Eq.(13), we can obtain $\tilde{\mathbf{r}}$, that is the components of \mathbf{r}_0 and \mathbf{R}_0 .

5 Microgravity Experiment

We performed experiments of a space robot capturing a target by using visual feedback control and evaluated our developed experimental system. The results of the experiments with the stereo-vision method and with the non-stereo-vision method are shown in the following.

5.1 With the Stereo-vision Method

Figure 9 shows the images processed by the TRV in the experiment with the stereo-vision method. The left column and the right one show the images of the left camera and the right one respectively. In the

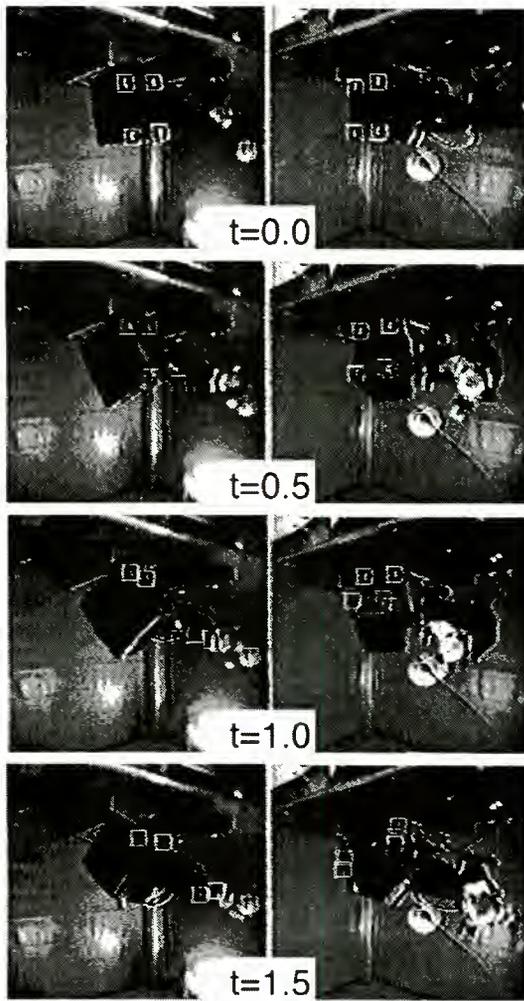


Fig. 9: Motion of the Robot in the Experiment with the Stereo-vision Method

figure, the TRV is tracking the images surrounded by the small white squares. After released by the holder, the robot started motion of capturing the target. As mentioned in the previous section, the robot base rotated and the TRV failed tracking the marks just before the robot would accomplish capturing the target. The tip of the arm approached the target steadily and reached just near it.

5.2 With the Non-stereo-vision Method

Figure 10 shows the images processed by the TRV in the experiment with the non-stereo-vision method. the problem of missing the marks didn't occur because of the large measurable region of the robot attitude. The robot accomplished capturing the target as shown in Fig.10.

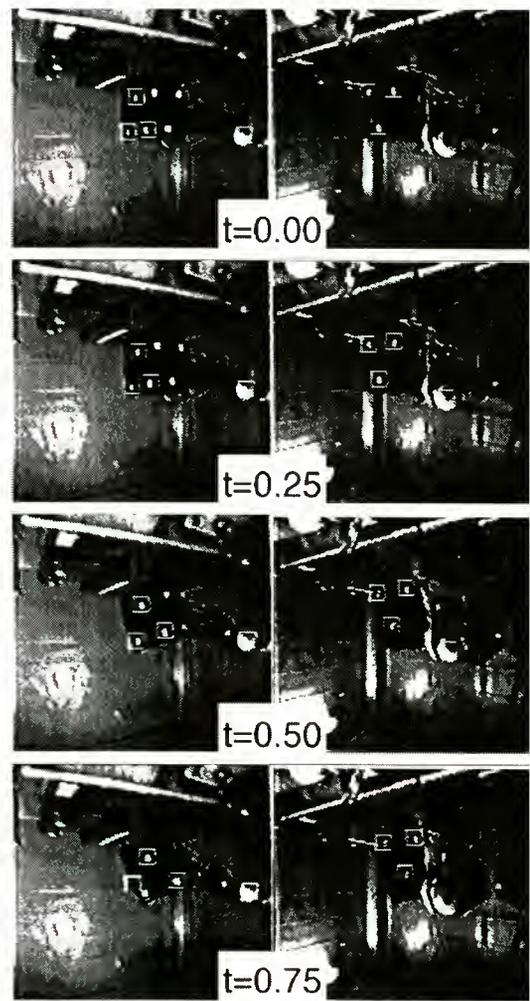


Fig. 10: Motion of the Robot in the Experiment with the Non-stereo-vision Method

6 Conclusion

In this paper, we proposed an experimental system in the microgravity environment generated by free-fall, and showed the experiments with the system at the Japan Microgravity Center. We summarize this paper as follows:

1. The microgravity experiment at the JAMIC was explained and the outline of our experimental system was introduced.
2. The problem in releasing the robot stably in the microgravity environment was discussed, and the importance of stiffness of the mechanism supporting the robot was pointed out.
3. We proposed the stereo-vision method and the non-stereo-vision method to measure the position and orientation of the robot.
4. The experimental results with the two proposed measurement methods were shown, and

the problem in measuring motion of the robot by using cameras set outside of the robot was discussed.

References

- [1] T.Iwata, K.Kodama, F.Numajiri, and H.Murakami: "Experiment on Robotic Motion Using Drop Shaft," Strengthen Cooperation in the 21 Century (6th IS-COPS) Vol.91, Advances in the Astronautical Society by Univelt, 1996.
- [2] Nakamura,Y. and Mukherjee,R., "Formulation and Efficient Computation of Inverse Dynamics of Space Robots", IEEE Trans. on Robotics and Automation, vol.8, no.3, pp.400-406, 1992.
- [3] H.Zuang, K.Wang, and Z.S.Roth: Simultaneous Calibration of a Robot and a Hand-Mounded Camera, IEEE Trans. on Robotics and Automation, vol.11, no.5, pp.649-660, 1995.

A 3D Position Measurement

We illustrate the parameters and the computational method used in this paper for measuring the 3D position of a point-P by using two camera, which are based on Ref.[3].

The relation between the position of the point-P in the inertial frame ${}^I\mathbf{p}$ and the one in the camera frame ${}^c\mathbf{p}$ is expressed as follows:

$$\begin{bmatrix} {}^c\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{O}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^I\mathbf{p} \\ 1 \end{bmatrix} \quad (16)$$

where

$${}^I\mathbf{p} = [x_I \ y_I \ z_I]^T \quad (17)$$

$${}^c\mathbf{p} = [x_c \ y_c \ z_c]^T \quad (18)$$

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (19)$$

$$\mathbf{t} = [t_x \ t_y \ t_z]^T \quad (20)$$

${}^c\mathbf{p}$ is transformed to the two-dimensional coordinates in the image plane by perspective as follows:

$$\frac{x_v}{f} = \frac{x_c}{z_c}, \quad \frac{y_v}{f} = \frac{y_c}{z_c} \quad (21)$$

where $[x_v \ y_v]^T$ denote the position of the point-P in the image plane and f the focal distance of the camera. The transformation from the pixel unit to the unit of the inertia coordinates is expressed as:

$$s_x x_v = H - c_x, \quad s_y y_v = c_y - V \quad (22)$$

where $[H \ V]^T$ denotes the pixel-unit position of the point-P on the image, $[c_x \ c_y]^T$ that of the image

center, and s_x, s_y the camera scale factors. From Eqs.(16)-(22), we obtain

$$\mathbf{C}\boldsymbol{\phi} = 0 \quad (23)$$

where

$$\mathbf{C} = \begin{bmatrix} x_I & 0 & x_I H & y_I & 0 & y_I H \\ 0 & x_I & x_I V & 0 & y_I & y_I V \\ & & & z_I & 0 & z_I H & 1 & 0 & H \\ & & & 0 & z_I & z_I V & 0 & 1 & V \end{bmatrix} \quad (24)$$

$$\boldsymbol{\phi} = [S_1 \ S_4 \ S_7 \ S_2 \ S_5 \ S_8 \\ S_3 \ S_6 \ S_9 \ U_x \ U_y \ -1]^T \quad (25)$$

$$\left. \begin{aligned} S_1 &= \frac{s_x f r_1 + c_x r_7}{t_z} & S_2 &= \frac{s_x f r_2 + c_x r_8}{t_z} \\ S_3 &= \frac{s_x f r_3 + c_x r_9}{t_z} & S_4 &= -\frac{s_y f r_4 - c_y r_7}{t_z} \\ S_5 &= -\frac{s_y f r_5 - c_y r_8}{t_z} & S_6 &= -\frac{s_y f r_6 - c_y r_9}{t_z} \\ S_7 &= -\frac{r_7}{t_z} & S_8 &= -\frac{r_8}{t_z} & S_9 &= -\frac{r_9}{t_z} \\ U_x &= \frac{s_x f t_x + c_x t_z}{t_z} & U_y &= -\frac{s_y f t_y - c_y t_z}{t_z} \end{aligned} \right\} \quad (26)$$

$\boldsymbol{\phi}$ consists of the unknown camera parameters and therefore is decided by the camera calibration. We define the parameters for the two cameras as follows:

$$\boldsymbol{\phi}_* = [S_{1*} \ S_{4*} \ S_{7*} \ S_{2*} \ S_{5*} \ S_{8*} \\ S_{3*} \ S_{6*} \ S_{9*} \ U_{x*} \ U_{y*} \ -1]^T \quad (27)$$

($*$ = l, r)

The pixel unit positions of the point-P on the two images are represented by $[H_l \ V_l]^T$ and $[H_r \ V_r]^T$ respectively. Then, ${}^I\mathbf{p}$ is computed from the following equation.

$$\mathbf{A}{}^I\mathbf{p} = \mathbf{b} \quad (28)$$

where

$$\mathbf{A} = \begin{bmatrix} S_{7l}H_l + S_{1l} & S_{8l}H_l + S_{2l} & S_{9l}H_l + S_{3l} \\ S_{7l}V_l + S_{4l} & S_{8l}V_l + S_{5l} & S_{9l}V_l + S_{6l} \\ S_{7r}H_r + S_{1r} & S_{8r}H_r + S_{2r} & S_{9r}H_r + S_{3r} \\ S_{7r}V_r + S_{4r} & S_{8r}V_r + S_{5r} & S_{9r}V_r + S_{6r} \end{bmatrix} \quad (29)$$

$$\mathbf{b} = \begin{bmatrix} H_l - U_{xl} \\ V_l - U_{yl} \\ H_r - U_{xr} \\ V_r - U_{yr} \end{bmatrix} \quad (30)$$

Robotics for Small Body Exploration

Autonomous Landing and Smart Anchoring for In-Situ Exploration of Small Bodies

Ali R. Ghavimi, Frederick Serricchio, Ben Dolgin, Fred Y. Hadaegh

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California, USA
Email: ali.r.ghavimi@jpl.nasa.gov
Tel: (818) 354-0470 Fax: (818) 393-4440

Abstract. Future NASA missions include in-situ scientific explorations of small interplanetary objects like comets and asteroids. Sample acquisition systems are envisioned to operate directly from the landers that are anchored to the surface. Landing and anchoring proves to be challenging in the absence of an attitude control system and in the presence of nearly zero-gravity environments with uncertain surface terrain and unknown mechanical properties. This paper presents recent advancements in developing a novel landing and anchoring control system for the exploration of small bodies.

1. Introduction

Smart Landing systems are essential for in-situ scientific investigation of small interplanetary objects like comets and asteroids. The ability to land on small bodies is a challenging problem mainly due to nearly zero-gravity environments, intrinsic properties of small bodies. The complexity arises as target bodies pose uncertain characteristics in their associated spin axes, rotation rates, geometric orientation, material properties, and surface terrain. In the presence of these limiting constraints and in the absence of an attitude control system, a smart landing strategy is needed that is capable of providing a reasonable impact absorption, a stable landing configuration, and an anchoring system for retaining the lander to the surface of the target body. Science objectives require that

sample acquisition systems and other science instruments operate directly from the lander. Therefore, the anchoring system must be actively controlled to maintain surface retention to facilitate sampling operation and minimize the effects of reaction forces and torques induced by science instruments. The following provides a brief summary of the proposed landing concept, hardware development, and control system design at NASA's Jet Propulsion Laboratory (JPL).

2. Landing System Requirements

The main requirements for the small body landing are based on the earlier version of the Space Technology 4 / Champollion Lander system to be launched in 2003. The target body assumptions are as follows: the gravitational acceleration is nearly zero; the surface terrain has uncertain properties and includes obstacles of up to 0.5 meters in height; the surface has unknown mechanical characteristics whose hardness can range from concrete to fluffy snow.

The lander system does not include an active descent mechanism. The baseline requirements consider impact trajectories with vertical landing velocity of up to 4 meters per second and horizontal landing velocity of up to 1 meter per second, relative to the local surface. Regardless of the large impact velocity and uncertain angle of attack, the rebound velocity must be controlled to less than 1 m/s.

3. Lander Configuration and Design

The proposed architecture for the landing mechanism is a three-legged lander system, where each leg is composed of a tripod of damping struts and an articulated footpad with an integrated anchor and winch mechanism. The damping struts must provide maximum energy absorption to ensure low rebound upon impact at cryogenic temperatures. The articulated footpad conforms to the surface on impact and the pyro fires a tethered anchor. Upon rebound, the winch mechanism brings the footpad back to the surface and maintains surface contact with a prescribed tether tension.

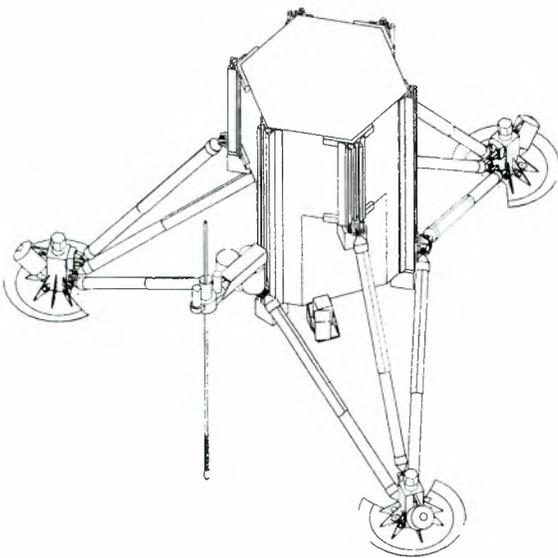


Figure 1. Lander Configuration

The anchoring system in each foot contains a compact pyro device, a tethered anchor, and a winch mechanism. The compact pyro device launches a tethered anchor into the surface. Each anchor is designed for a tethered high-speed deployment and is accelerated to speeds ranging from 80 to 120 meters per second. The anchor design parameters include mass, shape, material, and impact velocity. These parameters are optimized for minimum momentum transfer

to the lander foot and maximum surface penetration of the anchor. The later capabilities are essential for stability during surface landing, anchoring, rebound, and retention.

The tether in each anchor is connected to a footpad-mounted winch mechanism. The winch motor spools up the tether and provides the necessary means to retain the lander to the surface. The tension in the tether is actively controlled to bring the lander back to the surface upon rebound, maintain surface retention during sampling, and avoid anchor displacement at all times.

The proposed landing concept was implemented in two stages. The first phase of the hardware development involved the implementation and testing of a one-dimensional landing system referred to as *The Sled Mechanism*. The completion of this stage played a key role in the development of the six degree-of-freedom lander design referred to as *The ESB Lander*.

4. The Sled Mechanism

The Sled Mechanism is representative of a special case of one-dimensional landing in a low-gravity environment. The sled platform slides on two nearly frictionless rails. A hanging counterweight and pulley system is used to obtain a realistic simulation of a low-gravity test environment by overcoming the effects of the residual friction. The mass of the sled platform is about 45 Kilograms to closely approximate the actual lander mass as defined by the requirements. The full travel length of the rails is about 3 meters and is sufficient to simulate the one-legged landing scenario over the prescribed range of impact velocities. Further, the amount of travel during rebound is representative of the amount of tip over that is seen by the three-legged lander system. The setup provides an ideal testbed for the evaluation of the control system performance of the sled platform.

Moreover, performance analysis can be conducted based on variation of rebound velocity that is controlled by scaling the impact velocity.

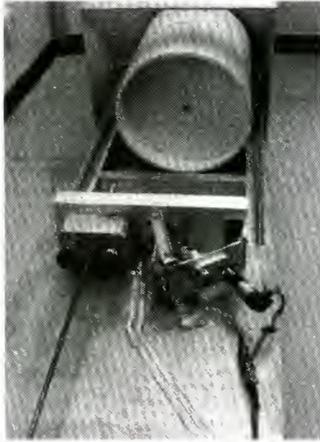


Figure 2. Sled Mechanism

The sled lander setup consists of a winch mechanism, tethered anchor, and compact pyro device. The winch mechanism includes a motor, encoder, tether spool, force sensor, and accelerometer. The sled platform accelerates towards a cylindrical target surface to achieve a prescribed 5 meters per second impact velocity. Upon impact sensed by the accelerometer, the compact pyro device launches the tethered anchor into the target surface. The motor, encoder, tether spool, and force sensor provide the means for the sled control system to bring the sled platform back to the target surface after rebound and maintain surface retention with a prescribed contact force.



Figure 3. Integrated Foot, Tethered Anchor, Spool, Compact Pyro, and Winch Mechanism

The tethered anchor embedded in the target surface has limited force retention capabilities. Note that independent tests were conducted to measure the force threshold of the anchor in various cometary simulant materials. During rebound, bounce off, surface return, and retention, the control system must control the tether tension to within thresholds imposed by the anchoring force constraint. This means that the motor may pay out the spool before it acts to wind up the tether spool to bring the sled platform back to the surface. This is particularly important during active sampling operation, when drilling forces continuously behave as disturbances to the lander control system

5. The ESB Lander

The ESB Lander is representative of a six degree-of-freedom landing in a low-gravity environment. The system consists of a central body and three landing legs. Each leg includes a tripod of damping struts connected to an integrated footpad described in Figure 3. A sampling system is attached to the central body for drilling purposes as shown in Figure 4.



Figure 4. ESB Lander

The lander is hung from a long tether and is released from an off vertical position to accelerate into the target surface. At impact,

damping struts are compressed, the legs are conformed to the surface, and pyros are fired.

JPL has developed a novel damping strut design that utilizes the shearing action of plunging cutters into vacuum-rated polyurethane foam designed to operate at cryogenic temperatures. This strut design is capable of providing passive damping under both tension and compression loads. Another significant property of the damping strut is the capability of retaining tension loads once the struts have been collapsed. This is an essential requirement during any sample drilling operation, as reaction forces tend to apply tension load to the damping struts.

The force sensors trigger the pyros at impact and the landing control system brings the feet back to the surface. The dynamic response of each individual foot in the ESB Lander is similar to that in the case of the Sled Mechanism. A smart control system must be devised, however, to initiate firing pyros in a proper sequential manner after the first foot has made contact with the surface. In other words, the second and third foot should fire their associated pyros based on actual surface impact as opposed to false triggering on the dynamic responses of the first foot. This is, of course, true for the third foot once the first and second feet have made contact with the surface.

5. Control System Design

The primary objective of the landing control system is to enable safe landing and secure anchoring to the surface. A successful landing is defined as: landing within the envelope of specified initial conditions, having all three feet on the surface, and maintaining the desired contact force. This means that the lander control system must provide autonomous actions to firing individual pyros in a proper sequential

manner, preventing the lander from tipping over, minimizing the tether tension, and keeping the anchor embedded in the surface. A robust implementation of a successful landing scenario requires full knowledge of the lander attitude. However, detailed analyses and simulation results show that the proposed three-legged lander design is able to meet the successful landing requirements within the envelope of initial conditions in the absence of a full attitude control system.

The overall control system design is fully autonomous and incorporates a hybrid of appropriate controller design methodologies. The control system is composed of three levels of decentralized controller design, where an executive controller commands the individual local servo and accordingly an associated low-level servo in each foot. The executive controller responds to event-based scenarios: impact, surface contact identification, and enabling surface retention. The local and low-level servos perform continuous control of the lander dynamics: tether tension control and surface retention.

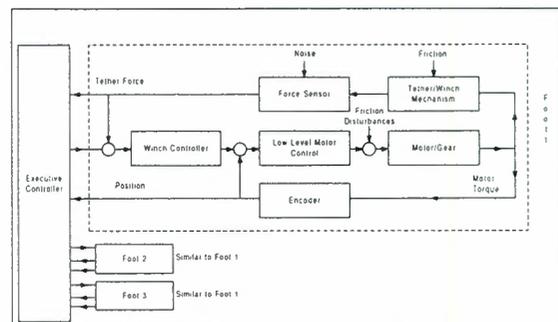


Figure 5. Control System Block Diagram

The first function of the executive controller is to ensure landing stability upon impact. A smart controller scheme is designed to determine when true surface contact is made at each foot by exploiting the dynamic coupling of the legs and processing the force sensor and accelerometer data. The

identification of the actual surface contact of each foot is crucial for proper sequential firing of the pyros. The second function of the executive controller is to enable the tether tension control servo to bring the feet back to the surface after surface contact of all three feet are made and the anchors have penetrated into the surface. This action of the landing scenario is performed to avoid pulling the anchors from the surface. Another function of the executive controller is to maintain surface retention in the presence of disturbances from science instruments

The local servos initiate motion on impact, absorb bounce off energy, bring the feet back to the surface, and retain them to the surface all under tether tension control. The operating plant in each local servo is an independent low-level motor/encoder servo loop. The controller in each local servo tracks the force input commanded by the executive controller. This is accomplished by issuing appropriate position input to the associated low-level servo subsystem.

The low-level servos initiate the physical motions of the tether spools. The active elements in the low-level servo plant are a motor and an encoder. The low-level servo controllers are highly optimized for tracking the input commands while minimizing the transient effects and providing fast reaction responses.

6. Simulation Models

A detailed simulation model of the ESB Lander is developed using ADAMS (Automatic Dynamic Analysis of Mechanical Systems). The model includes the inertia, mass, stiffness and damping characteristics of each component. The model incorporates internal dynamic coupling of the lander states, as well as, dynamic interactions with the external surface terrain. The terrain is also modeled

to represent topography and surface strength. The ADAMS lander model contains 46 rigid bodies connected by interface constraints and forces to yield a system with 141 degrees of freedom. A complete sensitivity analysis of the ADAMS lander model is performed by variation of important simulation parameters such as impact velocity, angle-of-attack, surface properties, anchor penetration depth, and anchor retention force. The results provide a baseline for choosing a mechanical design approach, as well as, a controller design strategy.

7. Test Results and Demonstrations

The end-to-end operation of both the Sled Mechanism and the ESB Lander was demonstrated under numerous operating conditions. The landing systems were tested for a wide range of impact velocities to assess the control system performance. The ESB Lander was tested for various angles of attack to evaluate the capability of the landing and anchoring system. The anchoring system was tested in various comet simulant materials including foam, plaster, limestone, bishop tuff, and sandstone. The penetration depth and anchor retention force were determined in each case to classify the strength of the anchoring system.

Sample drilling operations were performed from both the Sled Mechanism and the ESB Lander. The lander control systems successfully maintained the surface retention force and limited the tether tension during the drilling operation. The overall landing control system design and implementation met the objectives to demonstrate landing, anchoring, and sampling.

8. Conclusions

JPL has developed a unique landing concept, together with the proposed landing control strategy, that has potential for

meeting requirements for autonomous in-situ scientific exploration of small interplanetary objects.

9. Acknowledgements

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

10. References

- [1] D. Sevilla, A. Ghavimi, F. Serricchio, *Drilling and Sampling Technology for the Exploration of Small Bodies*, International Symposium on Artificial Intelligence, Robotics and Automation in Space, Japan, 1997.
- [2] A. Ghavimi, F. Serricchio, F. Hadaegh, *Autonomous Sample Acquisition for Planetary and Small Body Explorations*, International Symposium on Artificial Intelligence, Robotics and Automation in Space, The Nether Lands, 1999.

Autonomous Landing System for MUSES-C Sample Return Mission

Takashi Kubota, Shujiro Sawai, Toshihiko Misu,
Tatsuaki Hashimoto, Jun'ichiro Kawaguchi, Akira Fujiwara

The Institute of Space and Astronautical Science,
3-1-1 Yoshinodai, Sagamihara, Kanagawa 229-8510, JAPAN.
TEL: +81-42-759-8304, FAX: +81-42-759-8304
E-Mail: kubota@cnenet.isas.ac.jp

Abstract

This paper describes an autonomous landing system for MUSES-C sample return mission. In deep space, it is difficult to navigate, guide, and control a spacecraft on a real-time basis remotely from the earth mainly due to the communication delay. So autonomous navigation and guidance is required for final approach and landing to an unknown body. It is important to navigate and guide a spacecraft to the landing point without hitting rocks or big stones. In the final descent phase, cancellation of the horizontal speed relative to the surface of the landing site is essential. This paper proposes an autonomous landing method based on optical sensors. The validity of the proposed method is confirmed by graphical simulations. This paper also proposes a sample collector method to collect the surface materials.

1. Introduction

As increasingly many missions are developed to explore the moon or planets, the navigation technology of a spacecraft in deep space is getting more important than ever. In recent years, the probe and sample return of asteroid has received much attention in Japan as well as in Europe and the United States. The Institute of Space and Astronautical Science (ISAS) will launch the engineering test spacecraft, MUSES-C[1] toward a near earth asteroid NEREUS in 2002.

In deep space, it is difficult to navigate, guide, and control a spacecraft on a real-time basis remotely from the earth mainly due to the communication

delay. So autonomous navigation and guidance is required for final approach to an unknown body. Many researchers have studied rendezvous methods in interplanetary and approach phases. However, there are few publications on autonomous landing methods in the final approach phase. For landing on an unknown body safely, it is necessary to obtain the terrain information of a planetary surface around a landing point. It is also important to navigate and guide a spacecraft to the landing point without hitting rocks or big stones. In the final descent phase, cancellation of the horizontal speed relative to the surface of the landing site is essential. This paper proposes a method for a spacecraft to land on the asteroid surface autonomously. This paper presents an autonomous landing scheme by integrating several navigation sensors. For the purpose of guiding a spacecraft to the landing point, the MUSES-C spacecraft is supposed to drop a visual target marker that can play a navigation aid as an artificial landmark on the surface. This landmark drastically reduces the computer burdens. This paper also proposes a sampling mechanism to collect the surface materials.

This paper is structured as follows. Section 2 describes the purpose and the mission scenario of MUSES-C mission. In Section 3, the strategy for autonomous approach and landing is proposed. Section 4 describes navigation sensors used in MUSES-C mission. In Section 5, a navigation method using visual sensor is proposed. A method to extract visual feature is explained Section 6 presents a sampling mechanism. Finally, Section 7 is for discussions, conclusions, and future work of the research.

2. MUSES-C Mission

ISAS will launch the spacecraft, MUSES-C[2] toward the asteroid NEREUS in 2002. This project is aiming at demonstrating four key technologies required for the future sample and return missions from extra-terrestrial bodies. Those technologies are : 1) solar electrical propulsion with ion thrusters in an interplanetary space, as a primary propulsive means, 2) autonomous optical guidance and navigation, 3) automated sampling mechanism, and 4) direct hyperbolic reentry of the recovery capsule to the ground.

The nominal target of the MUSES-C[3] spacecraft is a near earth asteroid NEREUS(4660). The launch is scheduled in January of 2002 and the arrival at Nereus at the beginning of April of 2003. Leaving the asteroid at the end of May of 2003, the spacecraft returns to the Earth in January of 2006. The mission duration from launch to the Earth return is about four years. In this nominal plan, the MUSES-C spacecraft stays for about two months at the asteroid and both mapping and sampling operations have to be carried out during that short period. The project also has a backup target 1989ML(10302) for which the launch and recovery take place half a year later respectively. In the backup plan, the mission period is about six months.

The spacecraft is launched via the ISAS medium class launch vehicle M-V. The mass of the spacecraft is about 500[kg] including chemical and ion engine propellant of 130[kg]. The planned solar cell is a tri-junction cell and the solar panel generates approximately 1.8[kW]. During the flight, the distance from the earth is shorter than 2 AU.

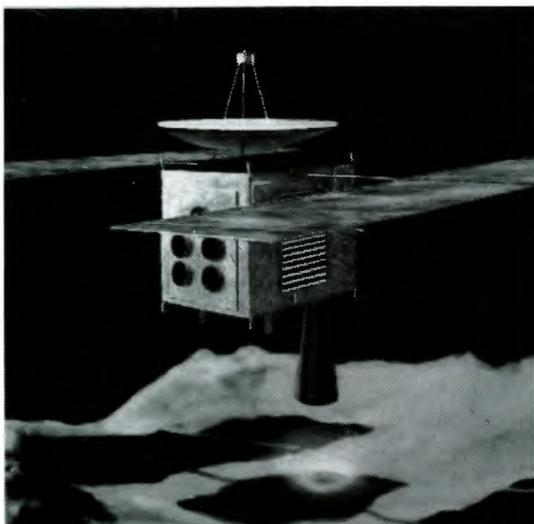


Fig.1 MUSES-C Mission

3. Autonomous Landing System

The MUSES-C spacecraft can rendezvous the asteroid by range and range rate method and conventional optical navigation method[4]. However long communication delay prevents ground based remote control in the proximity region around the asteroid. So autonomous landing system is important for mission success. At a range 20[km] from the asteroid, global mapping of the asteroid is performed to determine the topography of the surface and to search candidates of landing sites of scientific interest. The precise spin axis orientation and rotation rate and phase are also determined. The three-dimensional shape model of the asteroid is constructed for the approach and descent phase. Global mapping[5][6] from the Sun side and terminator side are scheduled in MUSES-C mission. The spacecraft keeps the home position before descent for sampling.

The strategy for autonomous landing consists of the following phases as shown in Fig.4.

1. Descent Phase

Optical navigation is used when the whole of the asteroid can be visible on the image. However difficulty shows up, when the spacecraft approaches close to the surface and the image spreads over the field of view. Feature areas are extracted and tracked on the images. If some of feature areas are unsuitable, new appropriate feature areas are extracted automatically.

2. Final Descent Phase[7]

In the event of sampling, cancellation of the relative horizontal speed is essential to the touchdown. For the purpose of securing this highlighted event, the spacecraft is supposed to drop a visual target marker that can act as a navigation aid. The position of a spacecraft with respect to the target marker is estimated by processing both flash-on and flash-off image data. The spacecraft is navigated and guided to the landing point based on landmark image. Introducing artificial landmarks drastically reduces the computer burdens.

3. Touch Down Phase[8]

As the spacecraft descends, there are some possibilities to collide with the surface. So it is needed to keep the attitude of the spacecraft parallel to the touch-down surface, while hovering at some altitude. Final Go or NO-Go decision for sampling is made at that time. Then the spacecraft starts the free-fall and touch down the asteroid surface to collect samples. During the free-fall of the spacecraft, some potential obstacles are checked.

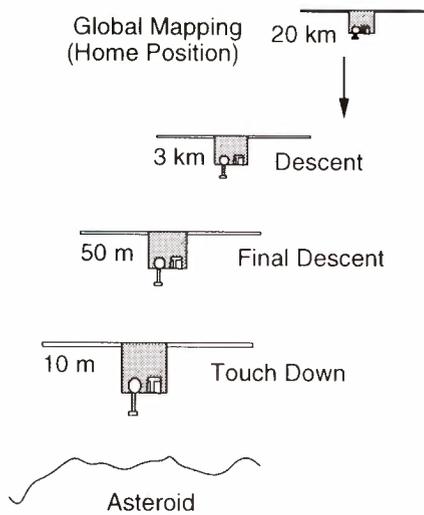


Fig.2 Autonomous Landing Strategy

4. Navigation Sensors

Rendezvous and touch down for the asteroid, whose size, shape, surface condition are unknown, requires intelligent and advanced navigation, guidance and control. A method is proposed to combine several navigation sensors, which makes it possible for a spacecraft to approach and touch down on the asteroid surface safely.

Figure 3 shows the current developing navigation sensors for MUSES-C mission. These instruments are used for the purpose of not only the navigation of the spacecraft but also scientific observation. The spacecraft has two kinds of optical navigation cameras. The narrow angle camera (ONC-T) is used for mapping and multiple scientific observations. The wide angle camera (ONC-W) is used for mapping and regional safety monitoring of surface obstacles. Measurement of the altitude is performed with LIDER (Light radio Detecting And Ranging). LIDAR covers the measurement range from 50[m] to 50[km].

In the final approach phase to the asteroid, the spacecraft orbit motion is synchronized with respect to the surface using image data. For sampling the surface materials, cancellation of the relative horizontal speed is essential to the touch down. To accomplish this, the spacecraft will drop a Target Marker that can act as a navigation aid by posing as an artificial landmark on the surface. The position of the target marker is estimated by ONC-W.

Laser Range Finder (LRF) is used at a lower altitude. LRF provides the height and attitude information with respect to the surface. A method is proposed, to estimate the height and attitude information of a spacecraft relative to the landing surface based on the range data. LRF has four beams that can measure the range from 7[m] to 120[m].

Fan Beam Sensor (FBS) is onboard as an alarm sensor to detect some potential obstacles that may hit the solar cell panels. This paper presents an autonomous landing scheme by integrating the visual information and the range information. The effectiveness and the validity of the proposed landing method are confirmed by graphical simulations.

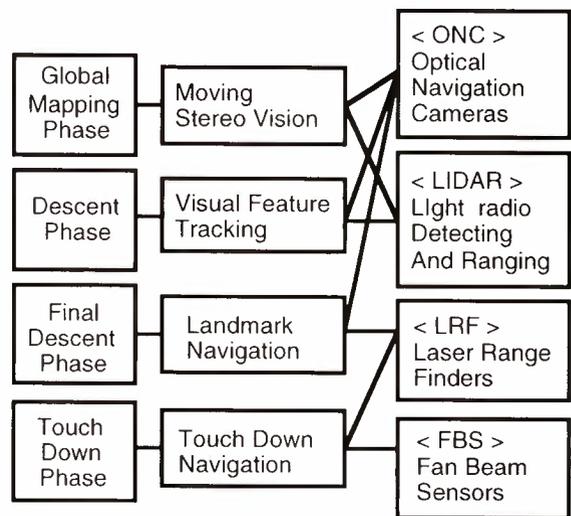


Fig.3 Navigation Sensors

5. Image Based Navigation

In descent phase, geographic features such as craters would be visible in images and be very useful in optical navigation. This paper proposes a new optical GNC system[9][10] which tracks small features such as craters, rocks valleys etc. on the target surface as feature points (FPs). In case that the landing site has visually characteristic features, it would be easy to guide the spacecraft to the landing site. However, not always exist such features on the landing site. In many cases, the landing site is chosen from flat areas so as to have the spacecraft touch down and land safely. So unluckily such an area would have almost no distinctive features. The proposed method uses distinctive features, which are not identical with those of the landing sites as FPs in order to avoid this problem.

The proposed scheme consists of two modes : tracking mode and updating mode. A landing point is designated on the camera image from the earth. Then visual feature areas are extracted as FP on an image and range to the landing site is measured by LIDAR. The location of the landing site with respect to the spacecraft is calculated as shown in Fig.4. In the tracking mode, the spacecraft tracks the FP on images based on template matching algorithms. The merit of the proposed scheme is that the landing site itself does not need to have any characteristic features. Therefore the spacecraft can land flat and safe area. As the spacecraft descends, however, some of FP would become unsuitable for tracking because of split from the field of view, extension etc. If such a situation occurs, new appropriate FPs are extracted automatically to reconstruct a renewed coordinate.

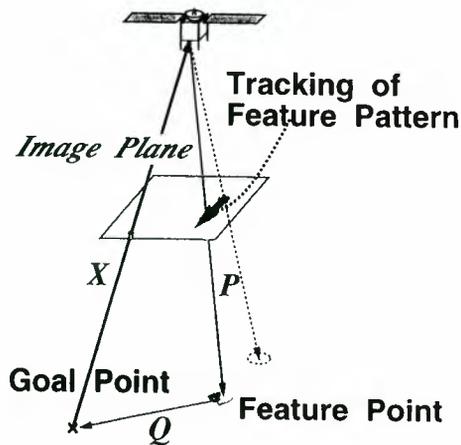


Fig.4 Image Based Navigation

5.2 Feature Extraction Scheme

As the spacecraft is at the mercy of the visual tracking, the block matching should be robust enough not to be defeated by noise. In addition, matching templates have to be chosen automatically, because real-time support from the earth is hardly available in deep space mission. The authors propose a simple and fast on-board scheme for extracting templates[11][12].

To make the tracking robust and accurate, each template used for the matching should have the following features :

1. shading pattern the wavelength of which is comparable with pre-selected size of template.
2. distinctive in the sense of contrast.

Such templates are extracted by the following procedure :

1. Enhance specific spatial wavelength of the original image by 2D band-pass filter.
2. Calculate local variance of filtered image to evaluate contrast.
3. Extract high local-variance areas as templates.

(1) Band pass filter

In the proposed scheme, BPF consists of the three image processing techniques : averaging, sub-sampling, and Laplacian filtering.

(2) Variance map

To evaluate roughness of the BPF image, the distribution of local variance is calculated. The procedure is as follows.

1. Calculate statistical variance within a window the size of which is equal to the template that is to be extracted.
2. Scan the window so as to cover the entire image.

(3) Templates extraction

Templates used for the tracking are extracted in order of local variance. Since the high local-variance points cluster in many cases, the extraction of another template is inhibited near to the already extracted ones.

5.3 Simulation Results

Figure 5 and Figure 6 show an example of original image and template extraction respectively. Four 32×32 [pel] windows are extracted from 256×256 [pel] original image in this case. The sub-sampling interval is chosen 4×4 . Fig.6(a) is the smoothed and sub-sampled image of the original one. It is shown that detailed structures are omitted. In Fig.6(b), smooth shading of Fig.6(a) is suppressed by Laplacian filter to enhance comparable features with the templates (32×32). The local variance is calculated as shown in Fig.6(c). Bright areas have high-variance region as shown in Fig.6(d). This simulation result shows that the templates have appropriate features.



Fig.5 Original Image

(a) Smooth and sub-sample



(b) Laplacian filtering



(c) Variance map



(d) Extracted areas on the original image

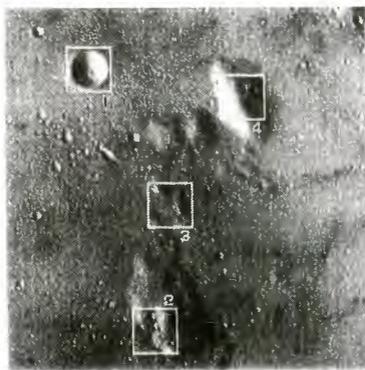


Fig.6 Simulation Results

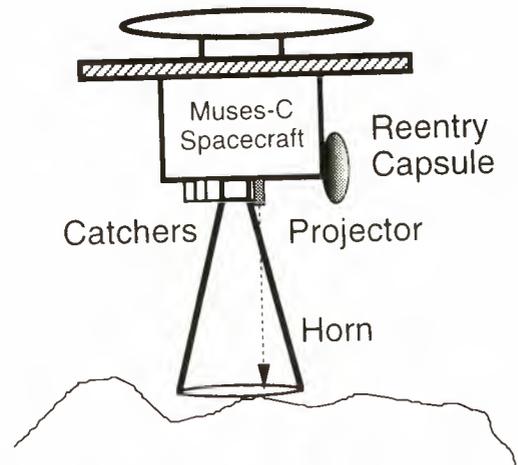


Fig.7 Sample Collector System

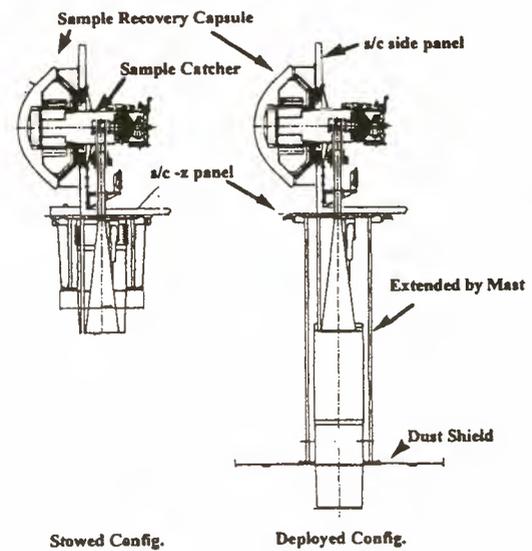


Fig.8 Sampler Horn

6. Sample Collector System

A sample collection technique is what the MUSES-C spacecraft demonstrates first in the world. Different from the large planets, the asteroid is a very small object whose gravity field is too little for any sampler to dig and drill the surface. Nevertheless, the spacecraft has to cope even with the hard surface such as rocks, while it is requested to function for soft surface like sands as well. Therefore, the authors propose a sample collection scheme[13] by the combination of the Shooting Projectile and the Fragment Catcher as shown in Fig.7. The basic idea is retrieving fragments from surface ejected by the projectile shot. The key in the mechanism is the use of the catcher whose inlet surface covers the shot area that is concealed from the spacecraft main body, so that the fragments and dusts cannot hit the spacecraft at all. The

spacecraft extends a mast whose tip end is equipped with a gun shooting a projectile of 10 grams at the speed of 300m/sec. Fig.8 draws how the extensible horn is stowed and deployed. Some low-gravity experimental results shows that several grams of fragments were successfully captured.

7. Conclusions

This paper has presented an autonomous navigation method to land on the asteroid surface in MUSES-C mission. A landing scheme by integrating several navigation sensors has been proposed. In descent phase, image based navigation has been also proposed. A method to extract visual feature areas has been explained. The validity of the proposed method has been verified by computer graphical simulations. This paper also has proposed a sampling mechanism to collect the surface materials. The validity of the proposed method has been confirmed by some experiments under the low gravity environment.

Reference

- [1] <http://www.muses-c.isas.ac.jp/>
- [2] T.Kubota, B.Wilcox, H.Saito, J.Kawaguchi, R.Jones, A.Fujiwara and J.Veverka, "A Collaborative Micro-Rover Exploration Plan on the Asteroid Nereus in MUSES-C Mission," IAF-97-Q.5.06, 1997.
- [3] J.Kawaguchi, Y.Morita, T.Hashimoto, T.Kubota, H.Yamakawa, H.Saito, "Nereus Sample Return Mission." 45th Congress of the International Astronautical Federation IAF-94-Q.5.354, 1994.
- [4] I.Nakatani, T.Kubota, T.Isaka, "Navigation for Small Body Rendezvous by Visual Active Sensing," Int. Symposium on Space Dynamics, pp.853-858, 1995
- [5] S.Nakamura, T.Kubota, T.Hashimoto, K.Ninomiya, "Motion Estimation of Unknown Body based on Optical Information," Proc. of 6th Workshop on Astrodynamics and Flight Mechanics ISAS, pp.54-59, 1996.
- [6] M.Maruya, S.Sawai, T.Kubota, T.Hashimoto, K.Ninomiya, "Estimation of Motion and Shape of Asteroid Based on Image Sequences," 14th International Symposium on Space Flight Dynamics, 1999.
- [7] J.Kawaguchi, T.Hashimoto, T.Kubota, S.Sawai and G.Fujii, "An Autonomous Optical Guidance and Navigation Strategy around a Small Body," Journal of Guidance, Control, and Dynamics, AIAA, vol. 20, no.5, pp. 1010-1017, 1997.
- [8] T.Kubota, I.Nakatani, T.Yamaizumi, H.Hayashida, "Navigation for Autonomous Landing on Unknown Planetary Surface Using Conical Laser Scanner," 3rd Int. Conf. on Spacecraft Guidance, Navigation and Control Systems, No.13.3, 1996.
- [9] T.Hashimoto, T.Kubota, N.Murata, M.Uo, M.Ogasawara, N.Kato, "Approach Navigation for Asteroid Sample Return Mission Utilizing Image and Range Measurements," Proc. of 5th Workshop on Astrodynamics and Flight Mechanics ISAS, pp.308-313, 1995.
- [10] J.Kawaguchi, T.Hashimoto, T.Misu, S.Sawai, "An Autonomous Optical Guidance and Navigation around the Asteroid," 47th Congress of the International Astronautical Federation IAF-96-A.4.03, 1996.
- [11] H.Osanai, T.Hashimoto, T.Kubota, S.Sawai, K.Baba, M.Ogasawara, M.Uo, K.Oda, "Experimental Study on Image Processing of Asteroid Surface by Using the TRANslational Motion Simulator (TRAM)," Proc. of 6th Workshop on Astrodynamics and Flight Mechanics ISAS, pp.31-36, 1997.
- [12] T.Misu, T.Hashimoto, K.Ninomiya, "Autonomous Guidance of Spacecraft based on Visual Feedback and Laser Ranging," Second IAA Int. Conf. On Low-Cost Planetary Missions, 1996.
- [13] J.Kawaguchi, K.Uesugi, A.Fujiwara, H.Saitoh, "The MUSES-C. Mission Description and its Status," Third IAA Int. Conf. on Low-Cost Planetary Missions, IAA-L98-0505, 1998.

Design of Contact Compliance and Simulation of Touch-Down Sequence of MUSES-C Spacecraft for Asteroid Sampling

Kazuya Yoshida

Department of Aeronautics and Space Engineering, Tohoku University
Aoba 01, Sendai 980-8579, Japan
yoshida@astro.mech.tohoku.ac.jp

Abstract

Small planetary objects such as asteroids and comets receive increasing attention for near-future exploration of the solar system. Some pioneering probes have already sent and returned remarkable findings, and others are being planned and developed to follow them. In Japan, the Institute of Space and Astronautical Science (ISAS), is now organizing the development of a sample-return mission to an asteroid. A spacecraft named MUSES-C is targeting one of near-Earth asteroids, in order to obtain samples and return to Earth. Robotics technologies are applied to the guidance and control of the landing and contact. Since the gravity of the asteroid is very small, the spacecraft will not be able to stand on its surface, but make dynamic touch in a free-flying situation. In this paper, the free-flying and contact dynamics are investigated to study the touch-down sequence for sample acquisition. The contact with mechanical compliance is modeled and dynamics simulations are carried out for feasible touch-down conditions.

1 Introduction

Asteroids are small particles of rocky bodies orbiting the Sun, a concentration of which bodies form an asteroid belt between Mars and Jupiter. The investigation into the astronomical questions on where these bodies come from, why they concentrate there, and what materials they are composed of, brings us significant knowledge on the origin and history of our solar system. The most informative way to answer these questions is to obtain samples from these planetary bodies themselves.

The Institute of Space and Astronautical Science (ISAS), Japan has a plan to launch an exploration robotic spacecraft, named MUSES-C, which can touch down on a surface of an asteroid and acquire samples off its surface, then take them back to the Earth [1]. In a tentative mission scenario, MUSES-C, a 400 [kg] spacecraft, will target one of

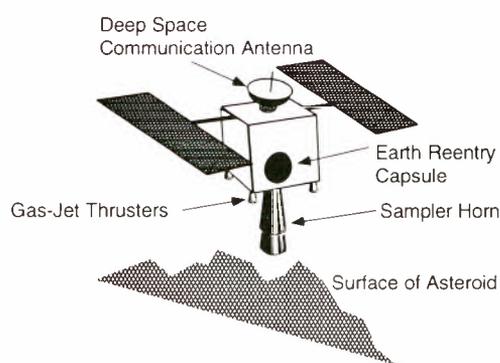


Figure 1: The asteroid sample-return spacecraft, MUSES-C

the near earth asteroids, estimated less than 1 [km] diameter rocky object. Candidates for such asteroids are "NEREUS" and "1989ML." The MUSES-C approaches to the asteroid's surface with small relative velocity controlled by a vision-based guiding system and makes contact by a horn-like sampling probe. Inside the probe, a projectile is projected toward the planet with some high velocity to crash the surface, then rebounding particles, ejecta, will be collected at the top corner of the horn.

There are many technical challenges in this mission. Particularly contact with the surface of asteroid is one of the most critical challenges. In order to make sure the safety, in terms of the strength of the structure against the impulsive force and the attitude maintenance against the impulsive moment, we need to carefully design the mechanisms and control systems, and simulate their dynamic behavior with making use of our maximum knowledge on the free-flying and contact dynamics. Only limited experiments are possible to test micro-gravity environment on Earth, thus hardware verification with a full-scale model is usually very difficult. Computer simulations are therefore a significant approach to study this design problem.

This paper discusses the dynamic simulation of

the touch-down sequence with the development of free-flying and contact models then, using tentative design parameters of MUSES-C, illustrates the dynamic motion after the contact.

2 MUSES-C Mission Scenario

2.1 MUSES-C

MUSES-C is a spacecraft for the asteroid sampling-return mission which is planned by ISAS to launch in 2002. Figure 1 depicts a basic configuration of the spacecraft, with 400 [kg] total mass and the dimension of the main body: 1.6[m] × 1.0[m] × 1.0[m]. The spacecraft has the following subsystems: an ion engine system for interplanetary voyage, a high-gain antenna for deep space communication, solar paddles, thruster propulsion systems, a sampling mechanism called "sampler horn" and a reentry capsule back to Earth. After the sampling action, sample particles are collected and packed into the reentry capsule, then its door is latched and sealed carefully to avoid contamination. When the spacecraft returns to Earth, only the reentry capsule, which we hope filled with a lot of informative samples, parachutes down to the Earth's surface.

2.2 Asteroids

Asteroids are small particles of rocky bodies orbiting the Sun. Up to now, we have a very limited information about these small planets through telescopes, and analysis of meteors. Recently, impressive pictures of some asteroids are taken by deep space explorers, such as Galileo. Those pictures show that an asteroid is not a spherical planet but a very oblique and rugged rock with craters. Generally speaking, these images agree our scientific expectation in the point that, for example, the gravity is not strong enough to form a spherical planet in these size of objects. But specific information such that, if an asteroid is a huge monolith or a cluster of soft soils, if the surface is rocky, sandy or dusty, and what materials it is composed of... all these are open questions, and the answers depend on the history of each asteroid.

In a tentative mission scenario, MUSES-C will target the asteroid NEREUS, or 1989ML. So far our knowledge is very limited on these asteroids, particularly its gravity and the surface condition (hardness, roughness), and right answers should be given only when the MUSES-C makes a physical contact with it. For the purpose of the simulation study, we assume the gravity on surface 9.8×10^{-4} [m/s²].

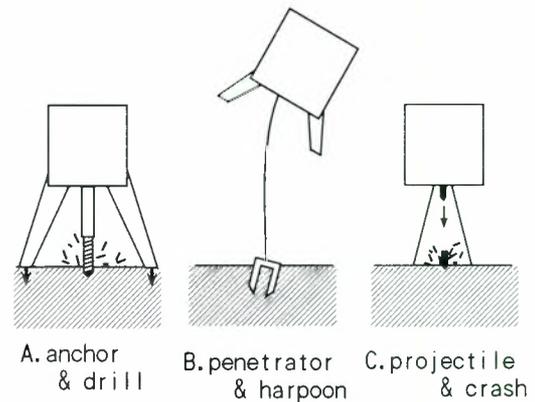


Figure 2: Sampling Methods

2.3 Sampling Strategy

In order to obtain samples from a small planet, which does not have enough gravity to firmly fix the explorer on its surface, the following strategies depicted in Figure 2 have been discussed.

Drilling technology (A) may work effectively for the sampling from a comet, which is considered to be composed of relatively *soft* materials such as dusty, icy, and snowy compounds. However an asteroid is considered as a more rocky or stony object covered with relatively *hard* surface, then we need more high-energy methods to crash the surface.

One of such methods is with a penetrator harpoon (B). If a penetrator capsule is projected down to a planet, it will be packed with crashed surface materials. In this method, however, the issue will be how to pull the capsule off the surface and retrieve it safely.

Currently a group of people are developing a projector method (C) and its possible designs. The basic idea is to project a 5-10 grams projectile toward the asteroid's surface with several hundreds [m/s] velocity inside the sampler horn. We expect this provides enough energy to crash the surface and the rebounding particles, or ejecta will be collected at the top corner of the horn. Since the sampling by the projector system will complete very quickly, the spacecraft is required to maintain the sampler contact for very short while, say 2-3 seconds, on the surface of the asteroid. This point is very favorable to our touch-down scenario.

2.4 Sampling Sequence

The gravitational force of the asteroid is very weak, estimated as one ten-thousandth of the earth gravity or less, the situation is therefore not that the explorer makes "landing" or "standing" on its surface but it does "rendezvous" and "berthing" in the free-flying environment. Assuming that MUSES-C takes the projection & crash method (Figure 2(C)) for sampling, we can summarize the sampling scenario as

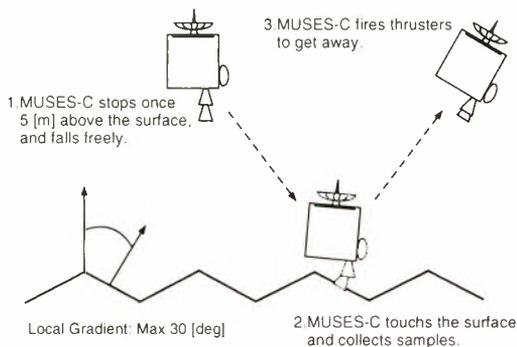


Figure 3: Sampling Sequence of MUSES-C

follows (see Figure 3):

1. The MUSES-C spacecraft makes rendezvous with the asteroid and descends to a point of interest.
2. About 5 [m] distance from the surface, MUSES-C is controlled its descending velocity to zero (hovering by thrusters,) then the thrusters are turned off to freely fall down on the surface. This will result 0.1 [m/s] vertical velocity at the surface contact.
3. The sampler horn, a contact probe is compliantly mounted on the spacecraft main body. The compliance works to reduce the contact impulse and extend the contact period.
4. While the endtip of the sampler horn stays on the surface, a projector is triggered and samples (ejecta) are collected inside the horn.
5. Thruster propulsion will follow immediately after the sampling, to get the spacecraft away from the surface. Note that thruster propulsion should turn on only after the sampling to prevent the contamination of the samples.

3 Modeling

3.1 Free-Flying Dynamics

To discuss flying or floating robot dynamics, we consider a general model that a robotic spacecraft has plural arms including solar paddles, reaction wheels or other appendages. Such a spacecraft is modeled by a chain of free-floating links in a tree configuration consisting of $n + 1$ rigid bodies, connected with n articulated joints. Assume that ℓ pieces of arms are mounted on the main body, and the arm k has n_k pieces of links, then $n = \sum_{k=1}^{\ell} n_k$. An example with a single arm is depicted in Figure 4.

Flexible arms or solar paddles can be treated as segmented virtual rigid links connected with elastic

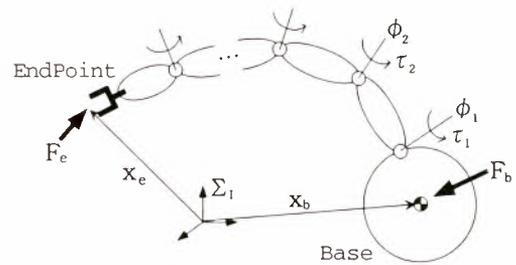


Figure 4: Free-flying robot system with a single arm

hinges. The flexibility yields elastic forces on the virtual hinges according to their virtual deformation. In this paper, we do not discuss the compliance and deformation of the sampler horn, but do not discuss the flexibility of solar paddles or other appendages to avoid complexity.

We assume that the system freely floats in the inertial space, and no orbital motion is considered.

Let us define the following coordinates and driving forces applying on the system.

- $\mathbf{x}_b \in R^6$: position/orientation of the base
- $\phi \in R^n$: joint angle of the arm
- $\mathbf{x}_e \in R^6$: position/orientation of the endpoint
- $\mathcal{F}_b \in R^6$: thruster force/moment on the base
- $\tau \in R^n$: joint torque of the arm
- $\mathcal{F}_e \in R^6$: external force/moment on endpoint

Here we can obtain the equation of motion in the following form [2, 3, 4, 5]:

$$\begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{bm} \\ \mathbf{H}_{bm}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_b \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_m \end{bmatrix} = \begin{bmatrix} \mathcal{F}_b \\ \tau \end{bmatrix} + \begin{bmatrix} \mathbf{J}_b^T \\ \mathbf{J}_m^T \end{bmatrix} \mathcal{F}_e \quad (1)$$

where

$$\mathbf{H}_b \in R^{6 \times 6} \equiv \begin{bmatrix} w\mathbf{E} & w\tilde{\mathbf{r}}_{0g}^T \\ w\tilde{\mathbf{r}}_{0g} & \mathbf{H}_\omega \end{bmatrix} \quad (2)$$

$$\mathbf{H}_\omega \in R^{3 \times 3} \equiv \sum_{k=1}^{\ell} \sum_{i=1}^{n_k} (\mathbf{I}_i^k + m_i^k \tilde{\mathbf{r}}_{0i}^k \tilde{\mathbf{r}}_{0i}^{kT}) + \mathbf{I}_0 \quad (3)$$

$$\mathbf{H}_m \in R^{n \times n} \equiv \sum_{k=1}^{\ell} \sum_{i=1}^{n_k} (\mathbf{J}_{Ri}^{kT} \mathbf{I}_i^k \mathbf{J}_{Ri}^k + m_i^k \mathbf{J}_{Ti}^{kT} \mathbf{J}_{Ti}^k) \quad (4)$$

$$\mathbf{H}_{bm} \in R^{6 \times n} \equiv \begin{bmatrix} \mathbf{H}_{v\phi} \\ \mathbf{H}_{\omega\phi} \end{bmatrix} \quad (5)$$

$$\mathbf{H}_{v\phi} \in R^{3 \times n} \equiv \sum_{k=1}^{\ell} \sum_{i=1}^{n_k} m_i^k \mathbf{J}_{Ti}^k \quad (6)$$

$$\mathbf{H}_{\omega\phi} \in R^{3 \times n} \equiv \sum_{k=1}^{\ell} \sum_{i=1}^{n_k} (\mathbf{I}_i^k \mathbf{J}_{Ri}^k + m_i^k \tilde{\mathbf{r}}_{0i}^k \mathbf{J}_{Ti}^k) \quad (7)$$

$$\mathbf{J}_{Ti}^k \in R^{3 \times n} \equiv [\mathbf{k}_1^k \times (\mathbf{r}_i^k - \mathbf{p}_1^k), \mathbf{k}_2^k \times (\mathbf{r}_i^k - \mathbf{p}_2^k), \dots, \dots, \mathbf{k}_i^k \times (\mathbf{r}_i^k - \mathbf{p}_i^k), \mathbf{o}, \dots, \mathbf{o}] \quad (8)$$

$$\mathbf{J}_{Ri}^k \in R^{3 \times n} \equiv [\mathbf{k}_1^k, \mathbf{k}_2^k, \dots, \mathbf{k}_i^k, \mathbf{o}, \dots, \mathbf{o}] \quad (9)$$

$$\mathbf{r}_{0g} \in R^3 \equiv \mathbf{r}_g - \mathbf{r}_0 \quad (10)$$

$$\mathbf{r}_{0i}^k \in R^3 \equiv \mathbf{r}_i^k - \mathbf{r}_0 \quad (11)$$

m_i^k : mass of link i of arm k

w : total mass of the system ($w = \sum_{k=1}^{\ell} \sum_{i=1}^{n_k} m_i$)

\mathbf{r}_i^k : position vector of centroid of link i of arm k

\mathbf{p}_i^k : position vector of joint i of arm k

\mathbf{k}_i^k : unit vector indicating joint axis direction of link i of arm k

\mathbf{r}_0 : position vector of centroid of spacecraft base body

\mathbf{r}_g : position vector of a total centroid of the system

$\mathbf{c}_b, \mathbf{c}_m$: velocity dependent non-linear terms

\mathbf{E} : 3×3 identity matrix

and a tilde operator stands for a cross product such that $\tilde{\mathbf{r}}\mathbf{a} \equiv \mathbf{r} \times \mathbf{a}$. All position and velocity vectors are defined with respect to the inertial reference frame.

3.2 Contact Dynamics

We assume the contact happens only at defined end-points. Note that MUSES-C does not have what is called manipulator arm, but the endpoint of the sampler horn, which is modeled as an articulated compliant arm, makes contact with an asteroid. The following discussion is on how to determine the contact force \mathcal{F}_e .

In literature, there are a few papers to deal with a dynamic model of rigid body collision with friction. A paper by Keller [6] and a book by Brach [7] are good references. Most of literature including the above deal with the relationship of momentum exchange and force-time product under the assumption of infinitesimal impact. However, the infinitesimal impact between two of single rigid bodies is a very idealized, special case. Eventually if the colliding body has elasticity, there occurs non-zero, finite-time period of contact. Or if the system is articulated and the connecting joints are compliant, the methods discussed for infinitesimal impact of a single rigid body cannot be applied. We may call such finite-time contact as *soft* contact against the infinitesimal impact as *hard* contact.

On MUSES-C we put a spring between the main body and the sampler horn. The spring is used to deploy the horn to stretch out from the the launch configuration, and it is more important to fit the front end of the horn to the uneven surface and absorb energy at the time of contact. Therefore we need to treat the "soft" contact problem to simulate the contact behavior of the spacecraft.

The dynamic motion of the free-flying multibody system is described by Equation (1) with the presence of the external forces \mathcal{F}_e . The magnitude of the forces is determined by the compliant deformation and friction of the contact surface.

Let us assume a point contact, then the contact moment is zero and the translational contact force \mathbf{f} should be discussed. If we assume a model of elastic-plastic deformation in the normal (z) direction of the contact point, and Coulomb friction in the tangential directions (x and y), we have the following general expressions:

$${}^c\mathbf{f}_z = K(\mathbf{d})^r + D(\dot{\mathbf{d}})^s, \quad (12)$$

$${}^c\mathbf{f}_x \leq \mu \cos \eta {}^c\mathbf{f}_z, \quad (13)$$

$${}^c\mathbf{f}_y \leq \mu \sin \eta {}^c\mathbf{f}_z, \quad (14)$$

where \mathbf{d} is the depth of penetration and $\dot{\mathbf{d}}$ is its velocity. The left-superscript $\{ {}^c \}$ indicates the local coordinate frame located on the contact point. Also, μ is the coefficient of friction and η is the angle defined by

$$\tan \eta = \frac{v_{cy}}{v_{cx}}. \quad (15)$$

There are number of discussions and still open questions on the above equations in the points that what numbers should be used for K, D, r and s , and how to find a consistent solution from inequality of the friction model. Here in this paper, we take an approach featured by a) a linear spring-damper model for the deformation mechanics, say $r = s = 1$, b) experimental estimation of K, D as is reported in [8]. And c) we take a special care on the treatment of frictional force, which may easily yield physically impossible solutions that is called *negative energy loss* by Brach [7].

4 Simulation

4.1 Model Parameters

Figure 5 depicts a drawing of MUSES-C used for the simulation. The kinematic and dynamic parameters of main components are listed in Table 1.

The sampler horn is assumed compliant in vertical (longitudinal) direction, but constraint in other directions.

The attaching point of the sampler horn is far away from the centroid, or the inertial principle axis of the main body. This off-axial attachment yields significant moment then angular motion to the main body due to the contact impulse, as will be seen in the simulation results later. However, a connector interface with a launching rocket booster takes place in the center of this surface, then there is no room to mount the sampling horn on axis.

The surface of the asteroid is assumed with same or similar hardness and damping of firebricks. The parameters of the firebrick we identified are used in the simulation. The surface is assumed flat and horizontal.

Table 2 Simulation results: contact force, time and rebounding velocities

	With horn compliance			Without horn compliance
	$v_x = 0.08$ [m/s]	$v_x = 0.0$ [m/s]	$v_x = -0.08$ [m/s]	$v_x = 0.0$ [m/s]
V_x [m/s]	0.146	0.086	0.018	-0.010
V_y [m/s]	-0.004	-0.006	-0.007	-0.004
V_z [m/s]	0.032	0.066	0.093	0.064
ω_x [deg/s]	0.069	0.077	0.086	0.135
ω_y [deg/s]	1.934	1.169	0.359	1.599
ω_z [deg/s]	0.0123	-0.005	-0.021	0.017
f_{max} [N]	15.304	17.962	20.571	136.360
t_c [sec]	6.690	6.365	6.150	0.785

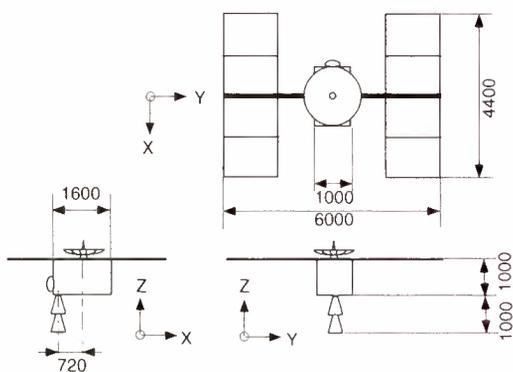


Figure 5: A drawing of MUSES-C used for simulation. Note that this is not a final configuration, which is currently under discussion as of June 1998.

Table 1 Simulation Parameters

mass of main body [kg]	m_0	409
moment of inertia [kg·m ²]	$I_{0,xx}$	300
	$I_{0,yy}$	230
	$I_{0,zz}$	430
attaching point of the sampler horn from the centroid of main body [m]	d_x	-0.72
	d_y	0.02
	d_z	-0.50
mass of sampler horn [kg]	m_1	1.0
moment of inertia [kg·m ²]	$I_{1,xx}$	1.0
	$I_{1,yy}$	1.0
	$I_{1,zz}$	1.0
compliance of the horn [N/m]	K_s	100
damping of the horn [Ns/m]	D_s	4.3
compliance of the asteroid [N/m]	K_w	10000
damping of the asteroid [Ns/m]	D_w	17.0
friction coefficient	μ	0.5
inclination of the surface [deg]	θ	0

4.2 Reaction of the Projector and Thrusters

The reaction of the projector and the gas-jet thrusters are other sources of external force on the

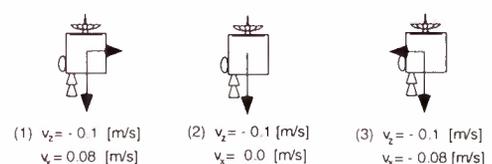


Figure 6: Three cases of contact velocity

main body than the contact impulse. The reaction of the projector is estimated to yield 3 [Nms]. Twelve of 22 [N] thrusters are mounted on the main body and four of them can be used to lift-off from the asteroid. However in the following simulation, these forces are not accounted, in order to see the nature of the physical contact and rebound.

4.3 Contact Velocity

The nominal contact velocity in vertical (z) direction is -0.1 [m/s]. The horizontal velocity, however, may be more difficult to control. This is because the height (vertical distance) can be measured by a ranging sensor, but there is not an easy way to measure the horizontal distance. Then we set a design interface for horizontal motion control to allow plus-minus 0.08 [m/s]. In the simulation, we evaluate three cases of contact velocity as shown in Figure 6.

4.4 Contact Force and Rebounding Motion

Figures 7 show an example of the simulation, where the contact forces, the horn-tip positions, and the attitude of spacecraft are displayed. As a parametric study, the rebounding (lift-off) velocities, the maximum contact force f_{max} , and the contact duration time t_c are compared as listed in Table 2.

The right column is the result without horn compliance to be compared with other three. It is clearly shown that the vertical (longitudinal) compliance in the sampler horn is very effective to reduce the con-

tact impulse and extend the contact duration.

All results show significant rotation around y axis. This is due to the moment of the off-axial horn attachment. This rotation is very serious especially when the spacecraft has horizontal velocity in $+x$ direction before the contact, because this horizontal

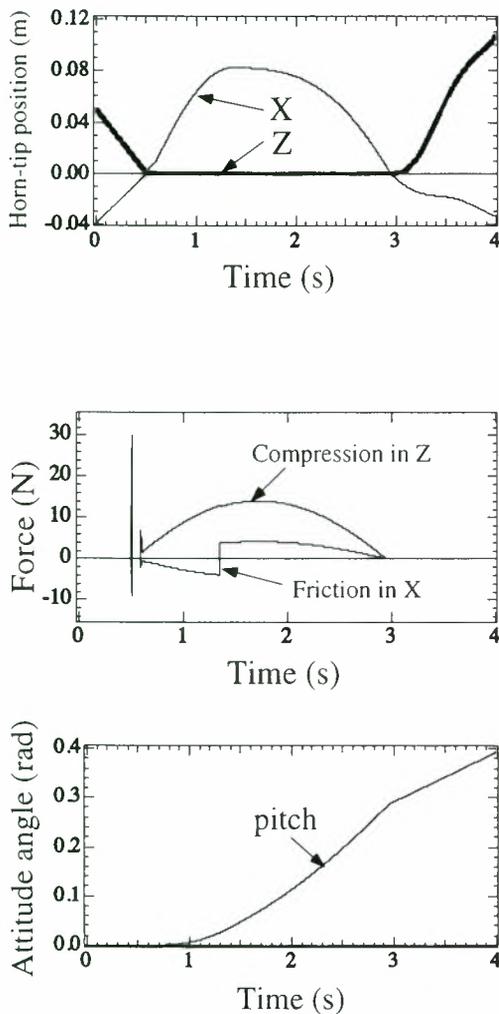


Figure 7: Simulation: a set of force/motion profile

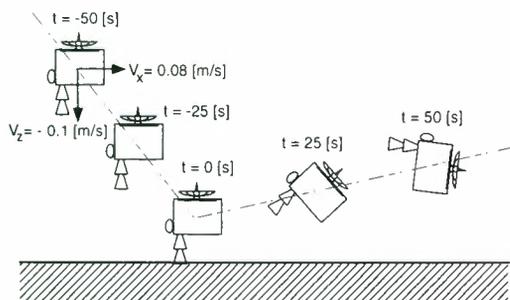


Figure 8: Simulation results of a critical situation

motion accelerates the pitch rotation. Figure 8 depicts an animated motion in such a critical case. We should carefully consider solutions to avoid this case. On the other hand, if the spacecraft has horizontal velocity in $-x$ direction before the contact, the moment by the off-axial horn and the moment by the horizontal velocity will cancel each others, thus yield smaller rotation.

5 Conclusion

In this paper, we discussed the dynamics simulation of the MUSES-C spacecraft for asteroid sampling, from the free-flying and contact dynamics point of view. A mathematical model to deal with free-flying and contact dynamics is developed. Then the dynamics simulations are carried out for feasible touchdown conditions. As a result of the simulation, we find the longitudinal compliance in the sampler horn is effective, and point out a critical situation due to the off-axial attachment of the horn. We need to carefully design the sampling sequence and control procedure to avoid such hazardous, and clarify safety margins by further simulations.

References

- [1] *A Proposal of the Asteroid Sample Return Mission: MUSES-C*, The Institute of Space and Astronautical Science, 1996. (in Japanese)
- [2] *Space Robotics: Dynamics and Control*, edited by Xu and Kanade, Kluwer Academic Publishers, 1993.
- [3] R. Mukherjee and Y. Nakamura: "Formulation and Efficient Computation of Inverse Dynamics of Space Robots," *IEEE Trans. on Robotics and Automation*, vol.8, no.3, pp.400-406, 1992.
- [4] Y. Yokokoji, T. Toyoshima, and T. Yoshikawa: "Efficient Computational Algorithms for Trajectory Control of Free-Flying Space Robots with Multiple Arms," *IEEE Trans. on Robotics and Automation*, vol.9, no.5, pp.571-580, 1993.
- [5] K. Yoshida: "A General Formulation for Under-Actuated Manipulators," *Proc. 1997 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1651-1957, Grenoble, France, 1997.
- [6] J. B. Keller: "Impact With Friction," *ASME J. of Applied Mechanics*, vol.53, no.1, pp.1-4, 1986.
- [7] R. M. Brach: *Mechanical Impact Dynamics: Rigid Body Collisions*, John Wiley & Sons, 1991.
- [8] K. Yoshida and T. Hiraoka: "Touch Down Simulation of the MUSES-C Satellite for Asteroid Sampling," *1998 AIAA Modeling and Simulation Technologies Conference*, AIAA 98-4376, pp.1-7, Boston, MA, 1998.

Precise Image-Based Motion Estimation for Autonomous Small Body Exploration

Andrew E. Johnson and Larry H. Matthies
 Autonomy and Control Section
 Jet Propulsion Laboratory
 California Institute of Technology
 Pasadena, CA 91109

Abstract

Space science and solar system exploration are driving NASA to develop an array of small body missions ranging in scope from near body flybys to complete sample return. This paper presents an algorithm for onboard motion estimation that will enable the precision guidance necessary for autonomous small body landing. Our techniques are based on automatic feature tracking between a pair of descent camera images followed by two frame motion estimation and scale recovery using laser altimetry data. The output of our algorithm is an estimate of rigid motion (attitude and position) and motion covariance between frames. This motion estimate can be passed directly to the spacecraft guidance and control system to enable rapid execution of safe and precise trajectories.

1 Introduction

Due to the small size, irregular shape and variable surface properties of small bodies, accurate motion estimation is needed for safe and precise small body exploration. Because of the communication delay induced by the large distances between the earth and targeted small bodies, landing on small bodies must be done autonomously using on-board sensors and algorithms. Current navigation technology does not provide the precision necessary to accurately land on a small bodies, so novel motion estimation techniques must be developed. Computer vision offers a possible solution to precise motion estimation.

Historically, optical navigation has been used for orbit determination and instrument pointing during close fly-bys of small bodies and moons of the outer planets. Generally, this has been implemented by ground-based image processing to extract centroids of small reference targets like asteroids and moons from which target relative spacecraft attitude and position are computed.

The Near Earth Asteroid Rendezvous (NEAR), a current mission that will rendezvous with asteroid Eros 433 in February 2000, uses optical navigation extensively for orbit determination and small body 3-D modeling [5]. The base-

lined navigation technique will combine manually designated landmarks from imagery of Eros and radiometric data to compute and control the trajectory of the orbiter. The NEAR mission will clearly demonstrate the effectiveness of optical navigation. However, this ground-based paradigm will not map to missions involving small body exploration and landing.

Small body exploration requires multiple precise target relative maneuvers during a brief descent to the surface. The round trip light time prohibits the determination of the necessary trajectory control maneuvers on the ground. Furthermore, typical onboard position sensors do not have the accuracy needed for small body landing (e.g., during a small body descent taking a few hours accelerometer errors will grow to the kilometer level). However, the required positional accuracies can be obtained if autonomous real-time optical navigation methods are developed.

The Deep Space 1 mission as part of the New Millennium Program is flying an autonomous optical navigation technology demonstration. The DS-1 AutoOpNav system will use onboard centroiding of reference asteroids for autonomous navigation during small body fly-bys [6]. They expect to obtain automatic position estimates with accuracies on order of 100 kilometers. For scientific instrument pointing purposes, this accuracy is sufficient. Controlled small body landing will require much better position and motion estimation accuracies. Furthermore, since the appearance of the small body is variable, small body landing cannot always rely on reference landmarks for navigation. The DS-1 AutoOpNav system will demonstrate autonomy and computer vision in space, however for small body landing a more versatile and accurate system is required.

This paper describes a fully autonomous and onboard solution for accurate and robust motion estimation near a proximal small body. Our techniques are based on automatic feature tracking between a pair of images followed by two frame motion estimation and scale recovery using laser altimetry data. The output of our algorithm is an estimate of rigid motion (attitude and position) and motion covariance between frames. This motion estimate can be passed directly to the spacecraft guidance navigation and control system to enable rapid execution of safe and precise trajectories.

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract from the National Aeronautics and Space Administration.

2 Motion Estimation

Motion estimation from images has a long history in the machine vision literature. The algorithm presented in this paper falls in the category of two-frame feature-based motion estimation algorithms. Once the spacecraft sensors are pointed at the small body surface, our algorithm works as follows. At one time instant a descent camera image and a laser altimeter reading are taken. A short time later, another image and altimeter reading are taken. Our algorithm then processes these pairs of measurements to estimate the rigid motion between readings. There are multiple steps in our algorithm. First, distinct features, which are pixels that can be tracked well across multiple images, are detected in the first image. Next, these features are located in the second image by feature tracking. Given these feature matches, the motion state and covariance of the spacecraft, up to a scale on translation, are computed using a two stage motion estimation algorithm. Finally the scale of translation is computed by combining altimetry with the motion estimates using one of two methods which depend on the descent angle. The block diagram for motion estimation is shown in Figure 1.

2.1 Feature Detection

The first step in two-frame motion estimation is the extraction of features from the first image. Features are pixel locations and the surrounding image intensity neighborhood (call this a feature window) that can be tracked well across multiple images that may undergo arbitrary, but small, changes in illumination or viewing direction. A qualitative definition of a good feature is a feature window that has strong texture variations in all directions.

Feature detection has been studied extensively and

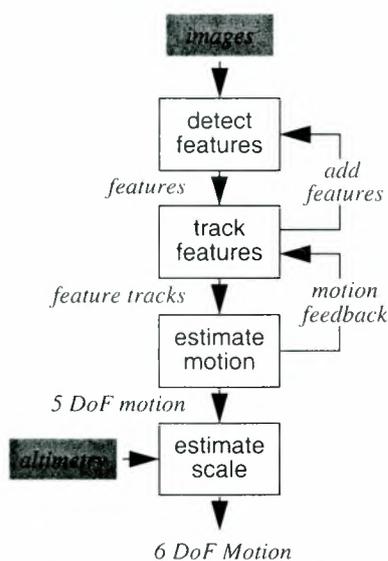


Figure 1: Block diagram for motion estimation.

multiple proven feature detection methods exist. Consequently, we elected to implement a proven feature detection method instead of redesigning our own. Since processing speed is a very important design constraint for our application, we selected the state of the art feature detection algorithm of Benedetti and Perona [2]. This algorithm is an implementation of the well known Shi-Tomasi feature detector and tracker [7] modified to eliminate transcendental arithmetic.

Surfaces of small bodies generally appear highly textured, so good features to track are expected to be plentiful. Usually feature detection algorithms exhaustively search the image for every distinct feature. However, when the goal is motion estimation, only a relatively small number of features need to be tracked (~ 100). The speed of feature tracking can be increased up to two orders of magnitude by using a random search strategy, instead of an exhaustive search for all good features, while still guaranteeing that the required number of features are detected. Suppose that N features are needed for motion estimation. Our detection algorithm selects a pixel at random from the image. If the randomly selected pixel has an interest value greater than a predetermined threshold, it is selected as a feature. This procedure is repeated until N features are detected.

2.2 Feature Tracking

The next step in motion estimation is to locate the features detected in the first frame in the second frame. This procedure is called feature tracking. As with feature detection, there exist multiple methods for feature tracking in the machine vision literature. Feature tracking can be split into two groups of algorithms: correlation based methods and optical flow based methods [7]. Correlation based methods are appropriate when the motion of features in the image is expected to be large. For small motions, optical flow based methods are more appropriate because in general they require less computation than correlation methods. We use the Shi-Tomasi feature tracker an optical flow based method for feature tracking, because in our application of precision landing, we know a-priori that the motion between image frames will be small. Our

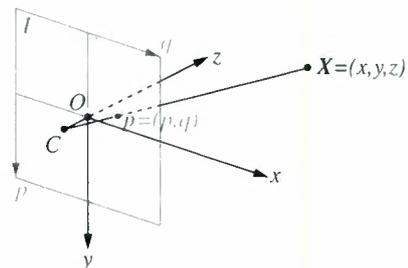


Figure 2: Unit focal length imaging geometry. World coordinate origin O is on image plane and optical center C is 1 unit behind image plane.

implementation of feature tracking follows that in [7] for 2-D (not affine) feature motion.

2.3 Two Frame Motion Estimation

The motion between two camera views can be described by a rigid transformation (R, T) where R encodes the rotation between views and T encodes the translation between views. Once features are tracked between images, the motion of the camera can be estimated by solving for the motion parameters that, when applied to the features in the first image, bring them close to the corresponding features in the second image.

In our algorithm, motion estimation is a two stage process. First an initial estimate of the motion is computed using a linear algorithm. This algorithm is applied multiple times using different sets of features to eliminate feature track outliers and determine a robust LMedS estimate of motion. The result of this algorithm is then used as input to a more accurate nonlinear algorithm that solves for the motion parameters directly. Since a good initial estimate is needed to initialize any nonlinear feature-based motion estimation algorithm, this two stage approach is common [11]. Output from the nonlinear algorithm is the estimate of the five motion parameters and their covariance. Our algorithm assumes that the camera taking the images has been intrinsically calibrated (i.e., focal length, radial distortion, optical center, skew and aspect are all known).

A fundamental short coming of all image-based motion estimation algorithms is the inability to solve for the magnitude of translational motion. Intuitively the reason for this is that the algorithms cannot differentiate between a very large object that is far from the camera or a small object that is close to the camera; the camera does not convey information about scene scale. Consequently, the output of motion estimation is a 5 DoF motion composed of the a unit vector $T_v = T/\|T\|$ describing the direction of heading and the rotation matrix R between views. As is shown in the next section, laser altimetry can be combined with 5 DoF motion estimation to compute the complete 6 DoF motion of the camera.

2.3.1. Robust Linear Motion Estimation

The first stage of motion estimation uses a linear algorithm to compute the motion between views [4]. Since the linear algorithm has a closed form solution, motion can be computed quickly. However, the linear algorithm does not solve for the motion parameters directly, so its results will not be as accurate as those obtained using the nonlinear algorithm. Our linear algorithm is an implementation of the algorithm presented in [10] augmented by normalization presented in [3] for better numerical conditioning. To filter out possible outliers in feature detection, we use a robust linear motion estimation algorithm based on least median of

squares[12].

2.3.2. Nonlinear Motion Estimation

Robust linear motion estimation serves two purposes: it provides an initial estimate of the 5 DoF motion between views and it detects and eliminates feature track outliers. The nonlinear algorithm takes the initial linear estimate of the motion and refines it by minimizing an error term that is a function of the motion parameters and the outlier-free feature tracks. There exists many nonlinear motion estimation algorithms in the vision literature. Instead of starting from scratch, the nonlinear algorithm we have developed combines the attractive elements of multiple algorithms to produce an algorithm that is computationally efficient, numerically stable and accurate. For numerical stability, we use the camera model parameterization of Azarbajani and Pentland[1]. For highly accurate motion parameter estimation we use the Levenberg-Marquardt algorithm as proposed by Szeliski and Kang[8]. Finally, for computational efficiency, we remove the scene structure from the nonlinear minimization as suggested by Weng et al. in [11].

First, the homogenous coordinates of each feature are determined by projecting them onto the unit focal plane. This projection will depend on the lens, imager, and camera model used. A simple model for the transformation of a feature at pixel location (p_i, q_i) to its homogenous coordinates u_i is

$$u_i = \begin{bmatrix} a_i & v_i & 1 \end{bmatrix}^T = \begin{bmatrix} p_i - C_p & q_i - C_q \\ f & sf \end{bmatrix}^T \quad (1)$$

where (C_p, C_q) is the center of the camera in pixel units, f is the focal length of the camera in pixel units and s is the aspect ratio of the pixels. This model assumes no radial distortion in the camera. More sophisticated models that include radial distortion are used when necessary [9].

Before we can express the error function, we need to detail the motion parameters over which the minimization will take place. First of all, the motion between frames is presented as a translation and rotation pair (R, T) . To simplify the parameter estimation, we represent the rotation

with a unit quaternion $q = [q_0, q_1, q_2, q_3]^T$ where the rotation matrix in terms of a unit quaternion is

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (2)$$

The translation is represented by a unit vector

$$T = [T_x, T_y, T_z]^T. \text{ Together the unit quaternion and unit}$$

translation comprise the parameter state vector \mathbf{a} .

$$\mathbf{a} = [q_0 \ q_1 \ q_2 \ q_3 \ T_x \ T_y \ T_z]^T \quad (3)$$

Nonlinear motion estimation attempts to minimize the image plane error between the features in the second view and the projection of the features in the first view into the second view given the motion between frames.

If the unit focal coordinates (defined by Equation 1) of the features in image I are $\mathbf{u}_i = [u_i \ v_i]^T$ and $\mathbf{u}'_i = [u'_i \ v'_i]^T$ in image J , then the image plane error is

$$C(\mathbf{a}) = \sum_i \|\mathbf{u}'_i - f(\mathbf{u}_i, \mathbf{a})\|^2 \quad (4)$$

where f represents the projection of the features \mathbf{u}'_i into image J given the motion \mathbf{a} . Correct image projection requires knowledge of the depth to a feature and a perspective camera model. Using the model of Azarbayejani and Pentland [1], if the (unknown) feature depths from the image plane are α_i , then the relation between unit focal feature coordinates and 3-D feature coordinates is

$$\mathbf{X}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} u_i(1 + \alpha_i) \\ v_i(1 + \alpha_i) \\ \alpha_i \end{bmatrix} \quad (5)$$

The features in image I are transformed into image J according to

$$\mathbf{X}'_i = [x'_i \ y'_i \ z'_i]^T = R(q)\mathbf{X}_i + \mathbf{T}. \quad (6)$$

By combining Equation 5 and Equation 6, the feature depths $[\alpha_i \ \alpha'_i]^T$ can be computed through triangulation by solving

$$\begin{bmatrix} -R\mathbf{u}_i & \mathbf{u}'_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha'_i \end{bmatrix} = \mathbf{T} \quad (7)$$

assuming that the translation between views is nonzero [10].

The camera model given the imaging geometry, shown in Figure 2, is

$$f(\mathbf{u}'_i, \mathbf{a}) = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \frac{1}{1 + z'_i} \quad (8)$$

Combining Equation 5, Equation 6, and Equation 8 results in a complete definition of Equation 4.

To estimate the motion parameters, we minimize Equation 4 using the Levenberg-Marquardt algorithm for nonlinear minimization. This approach was also used by Szeliski and Kang [8], however, unlike in their approach, we do not include the feature depths in the minimization. Inclusion of the feature depths would increase the length of the parameter vector from 7 to 7+N. Since the minimization relies on an inversion of a square matrix of rank equal to the

length of the parameter vector, a computationally expensive matrix inversion would result. Since feature depths can be computed directly from the motion between views, it is not necessary to include them in the parameter vector. Instead, at each iteration, the feature depths are updated using the current motion estimate. The result is a computationally efficient and accurate motion estimation algorithm.

Since we are solving for a rotation represented by a unit quaternion and also a unit length translation, these constraints need to be enforced during minimization. We enforce these constraints by setting $\|q + \delta q\| = 1$ and $\|\mathbf{T} + \delta \mathbf{T}\| = 1$ during the update of the parameter vector at each iteration of the Levenberg-Marquardt algorithm. Consequently, these constraints are enforced while not complicating the minimization by including the constraints explicitly in the minimization function.

The output of nonlinear motion estimation is an estimate of the 5 DoF motion between views. In addition, the covariance Σ of the motion parameters \mathbf{a} can be extracted directly from the quantities computed during minimization using

$$\Sigma(\mathbf{a}) = A^{-1}. \quad (9)$$

2.4 Scale Computation Using Altimeter

The final stage of motion estimation computes the remaining motion parameter, magnitude of translation, from laser altimetry data. Depending on descent angle and surface relief, one of two complimentary methods is used.

2.4.1. Motivation

Motion estimation using monocular imagery cannot solve directly for the magnitude of translation, so an external means must be used to recover this parameter. For a spacecraft in orbit about a small body, there exist multiple possible solutions.

One solution is to integrate the accelerometer measurement in the spacecraft inertial reference unit to determine position. The advantage of accelerometers is that they present a completely onboard solution. Unfortunately, because that come from integration of noisy acceleration measurements, position measurements from accelerometers may be too inaccurate for precision landing.

The traditional approach is to use radiometric tracking measurements from earth. This approach has the advantage that it is well understood and uses equipment already on board the spacecraft. However, radiometric tracking has many disadvantages. First, it requires dedicated Deep Space Network tracking which is expensive and difficult to schedule. Second, round trip light time for tracking from earth induces a large latency in any position measurements (approximately 24 minutes for comet Tempel 1).

Multiple missions have or are using laser altimeters for

science return and navigation. As shown below, laser altimeters can also be used as a navigation sensor by aiding the determination of the position of the spacecraft. Laser altimeters give accurate range estimates and, when combined with a descent imager, present a complete on-board solution to 6-D body relative motion estimation. A disadvantage of the laser altimeter approach is that they have limited range (50 km for the NEAR laser altimeter). However, near body operations is precisely when accurate position estimation is needed the most, so this is not a major issue. A laser altimeter is an additional sensor; however, science return combined with navigational use justify the addition. Based on the disadvantages of the other available options, we determined that the use of a laser altimeter was the most promising solution for scale estimation.

2.4.2. Difference Scale Estimation

If images are taken as the spacecraft descends vertically to the surface, or the surface has very little surface relief, computation of translation magnitude is straightforward. Laser altimeter readings A_I and A_J are acquired simultaneously with each image. As shown in Figure 3, the difference in altimeter readings is equal to the translation of the spacecraft along the z-axis between images. Consequently, the magnitude of translation is

$$\|T\| = \frac{(A_I - A_J)}{t_z} \quad (10)$$

For motion approaching horizontal, t_z approaches zero, Equation 10 becomes ill conditioned and difference scale estimation will not work. Furthermore, if the spacecraft is not descending vertically and the surface topography is rough on order of the scale of translation then the difference of altimeter readings will not accurately reflect the z component of the translation. Once again, difference scale estimation will not work. Fortunately a different, albeit more complicated, procedure exists for computing scale in these cases.

2.4.3. Structure-Based Scale Estimation

From the feature-based motion estimate, the scaled depths α_i (Equation 7) to features in the scene can be computed. Assuming, without loss of generality, that the laser altimeter is aligned with the camera optical axis, features in the optical center will be at a depth equivalent to the laser altimeter reading. Consequently, the ratio of the laser altimeter reading to the scaled feature range will be the magnitude of translation. This approach requires only one altimeter reading, so it is not susceptible to errors from changing surface relief. Furthermore, it does not depend on nonzero translation along the z-axis. In fact, structure-based scale estimation works better when the spacecraft is descending at an angle with respect to the surface because

in this case, scene structure can be estimated more accurately than for pure descent.

The procedure for structure-based scale estimation is to first compute the feature based motion between images along with the depth of the features in the image. Assuming alignment of laser altimeter with the optical axis, the features near the center of the image will be geometrically close to the surface patch that supplies the reading for the laser altimeter (see Figure 3). Since it is unlikely that a feature will correspond exactly to the image center, a few (3-5) features closest to the image center are selected and weighted interpolation is used to determine the scene depth at the image center α_c . The image-based scene depth at the image center has the same depth as the altimeter reading taken when the first image was acquired, so the magnitude of translation is

$$\|T\| = \frac{A_I}{\alpha_c} \quad (11)$$

A number of observations can be made about structure based scale estimation. First, As the translation between images approaches vertical, the structure estimates degrade, especially near the optical axis (i.e., on the optical axis, the displacement between features will be zero for vertical descent - structure from triangulation cannot be computed). Fortunately, vertical descent is precisely the motion where difference scale estimation works best. Second, for the altimeter reading to be related to scene structure, a feature must be located near the optical axis in the first frame, so structure-based scale estimation will work better when more features are tracked.

The magnitude of translation from laser altimetry when combined with feature-based motion completes the 6 DoF motion estimation of the spacecraft.

3 Results on Real Imagery

To test our motion estimation algorithm, we generated two sequences of real imagery. First a comet nucleus analog was created by a comet scientist at JPL. This analog is rough at all scales and matte black, the expected characteristics of comet nuclei. The analog has an approximate diameter of 25 cm. We placed the analog on a rigid stand and took two sequences of images as the camera moved toward the comet analog. The first sequence which

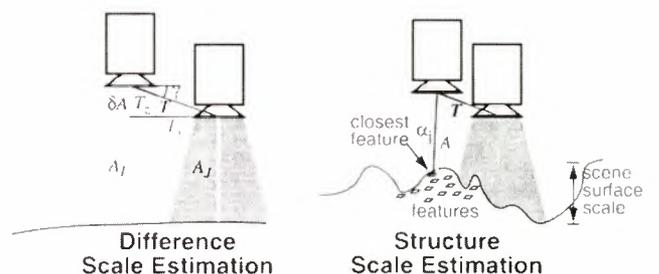


Figure 3: Methods for estimating translation magnitude.

we call *descent* was with a 640x480 CCD imager, a 15 degree field of view lens. The second sequence called *approach* was taken with a 1024x1024 CCD imager and a 25 degree field of view lens. Both sequences were acquired with the camera starting 80 cm from the comet analog; the camera moved 1.00 cm toward the analog between each image.

Ground truth for the image sequence motions were obtained through camera calibration [9]. Each camera was calibrated using a calibration target and as a by product of the calibration procedure, the direction of translation was computed. For the descent sequence, the true translation direction is (0,0,-1), and for the approach sequence, the true translation direction is (0.0096, -0.0033, -0.9999). Since the cameras were rigidly fixed, there was no rotation in the motion.

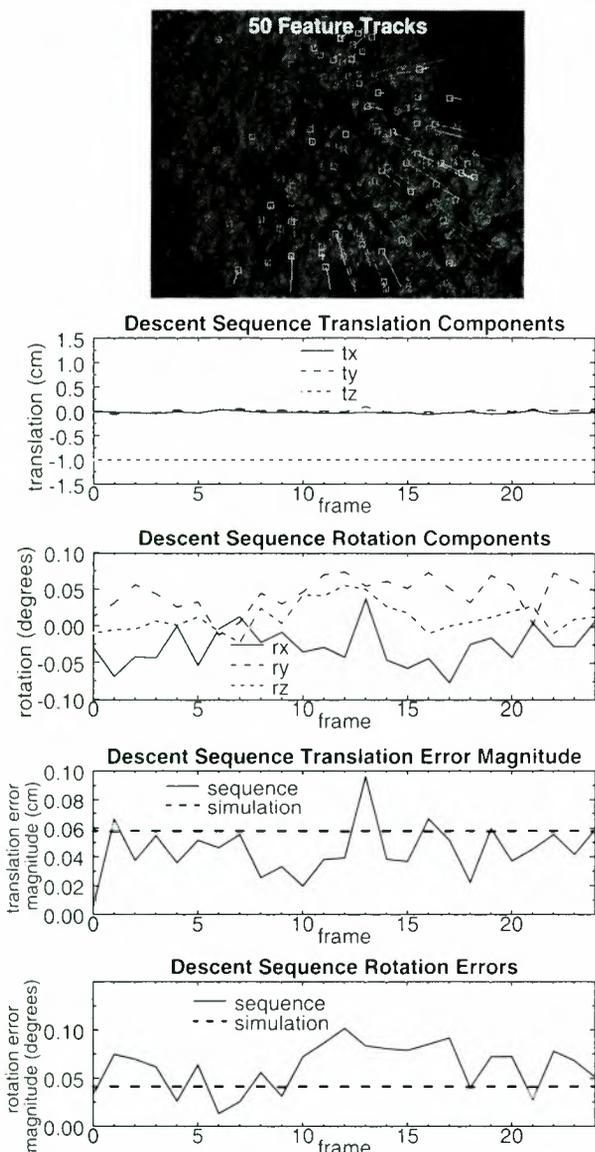


Figure 4: Motion Estimation for the Descent Sequence with 50 features tracked.

An altimeter reading was simulated for each image by using the translation stage reading as the altimeter reading. Using this data type, the scale of translation is known to the accuracy of the translation stage, so no scale estimation method is needed.

The motion estimation results for 50 features and the descent sequence are shown in Figure 4. At the top is shown the feature tracks for the entire sequence. Different shaded tracks correspond to the different key frames when the features were added to the sequence: a key frame occurred every 4 frames. Next are shown the computed translation (t_x, t_y, t_z) and rotation angles (r_x, r_y, r_z) of the motion computed for each frame using the two stage motion estimation algorithm. Following these is a plot showing the translation error magnitude (vector distance between the true and estimated translations) for each frame in the sequence. On this plot, the dashed line corresponds to the expected performance of the algorithm established using Monte Carlo simulation (assuming perfect feature tracking) for the imaging parameters and motion (See Section 4). Finally, the rotation error magnitude (vector difference between estimated and true rotation angles) is shown for each frame. Again, the dashed line corresponds to the expected performance of the algorithm established using Monte Carlo simulation.

Table 1 summarizes the additional motion estimation results obtained from processing the approach and descent sequences obtained using 50 or 500 features and linear or linear+nonlinear motion estimation

For the 50 feature descent sequence and the linear motion estimation algorithm, the average translation error is 0.045 cm or 4.5% of the distance traveled. The average rotation error is 0.063 degrees from no rotation. These error values are similar to the expected motion errors (0.057 cm and 0.04 degrees) from Monte Carlo simulation given the parameters of the image sequence. The frame rate for this sequence is 4.01 Hz on a 174 Mhz R10000 SGI O².

For the 50 feature approach sequence and the linear motion estimation algorithm, the average translation error is 0.028 cm or 2.8% of the distance traveled. The approach sequence results are more accurate because the resolution of the imager is greater. The frame rate for this sequence is 2.91 Hz on a 174 Mhz R10000 SGI O². The approach sequence takes slightly longer to process because the larger image requires more time to detect features.

The results in Table 1, show that in general the addition of the nonlinear motion estimation algorithm does not improve the results of motion estimation all that much. This is because for vertical descent, the motion computed using the linear algorithm is very constrained, so the results are very close to those obtained using the nonlinear algorithm. Including the nonlinear algorithm in general doubles the running time of the algorithm, so for the vertical descent, it is probably a good idea to remove this stage from the

algorithm if running time is important. However, for other motions (e.g., orbital motion) the nonlinear algorithm will result in improved motion estimation and should be used.

Table I also shows that adding features (50 vs. 500) does not improve motion estimation all that much. Since adding features increases the processing time of each frame, using 50 features is recommended for estimating descent motion.

4 Performance Testing

Using Monte Carlo testing, the effect of sensor parameters (e.g., field of view, resolution), spacecraft trajectory (e.g., motion, altitude) and scene characteristics (e.g., surface scale) on the accuracy of body relative motion estimation can be determined empirically. We used these tests to search for the “best” sensor parameters for precise motion estimation and to predict the performance of the algorithm given a predetermined set of sensor parameters.

4.1 Monte Carlo Simulation

The procedure for a single Monte Carlo trial is as follows: First a synthetic terrain map is generated to represent the surface of the small body. Next, a feature position in the first image is generated by randomly selecting a pixel in the image (feature position in first image). The 3-D position of the feature is found by intersecting its line of sight ray with the synthetic surface. Since the position of the camera for the second view is a known input, the 3-D point can be projected into the second view to determine its pixel position in the second image. Gaussian noise is then added to this feature pixel position to simulate feature tracking errors. This is repeated for however many features are requested. Altimeter readings are computed by intersecting the line of sight for the altimeter (the camera optical axis) with the synthetic terrain, and computing distance between the sensor origin and the surface intersection. Gaussian noise is then added to the range value to simulate measurement noise in the altimeter. Using simulated feature tracks and altimeter readings, the complete 6 DoF motion is estimated.

For these tests some of the motion estimation parameters

were fixed: imager resolution was fixed at 1024, field of view was set to at 30 degrees, spacecraft altitude was set to 1000 m, altimeter range accuracy was set to 0.2 m, feature tracking error was set at 0.17 pixels, average feature tracking disparity was set at 20 pixels, scene surface scale was set to 200 m., and number of tracks was set at 500. The remaining parameters to investigate are spacecraft motion and the scale estimation mode used in the algorithm.

4.2 Effect of Motion on Motion Accuracy

This investigation was performed to determine the effect of different spacecraft motions on motion estimation accuracies. To simplify this investigation, the space of possible motions was broken into two groups: descent (pure translational motion) and pointing (pure rotational motion).

Descent can be parameterized by descent angle γ (See Figure 3), the angle between horizontal and the translation direction of the spacecraft. Given the above parameters, simulations showed that a translational motion accuracy of 0.22 m is expected independent of scale estimation mode and descent angle. At a fixed pixel disparity, the distance traveled between frames varies depending on the magnitude of translation. For a horizontal motion ($\gamma=90^\circ$), a 20 pixel disparity and 30° field of view corresponds to a motion of 12 m. The motion error is then 0.22 m over 12 m or 1.8%. For a descent angle of $\gamma=45^\circ$ and a 30° field of view, a 20 pixel disparity corresponds to a motion of 17 m resulting in a motion error of 0.22 m over 17 m or 1.3%. Finally for vertical descent ($\gamma=0^\circ$) and a field of view of 30° , a 20 pixel disparity corresponds to a 65 m motion. Thus the error is 0.22 m over 65 m or 0.34%.

By integrating this motion accuracy estimate from multiple frames as the spacecraft descends to the surface an upper bound on the expected horizontal landing position accuracy can be obtained. Simulations showed that the most accurate landing position occurs for the vertical descent with a 10 degree field of view. In this case the landing position accuracy is 3.6 meters. From a height of 1000 meters, this is an accuracy of 0.36% of the starting altitude.

To determine pointing accuracy we only investigated

Table 1: Motion estimation results.

sequence	number of features	motion estimation stages	δT_{seq} (cm)	δR_{seq} (degrees)	processing time (seconds)	number of frames	frame rate (Hz)	δT_{sim} (cm)	δR_{sim} (degrees)
descent	50	linear	0.044927	0.06376	6.24	25	4.01		
descent	50	nonlinear	0.044966	0.0662209	13.1	25	1.90	0.0579763	0.0411912
descent	500	linear	0.033483	0.056666	31.61	25	0.79		
descent	500	nonlinear	0.033615	0.056834	82.33	25	0.30	0.0169	0.0120
approach	50	linear	0.028092	0.024439	2.4	7	2.91		
approach	50	nonlinear	0.023936	0.021443	3.94	7	1.77	0.0659696	0.0505746
approach	500	linear	0.01861	0.017992	13.42	7	0.52		
approach	500	nonlinear	0.018938	0.15937	24.05	7	0.29	0.0221996	0.169442

rotations with axes perpendicular to the camera Z-axis since rotations about the camera Z axis are unnecessary for pointing to surface targets. For a 30° field of view, a 20 pixel average disparity corresponds to a rotation of 0.6° away from the optical axis. Simulations showed that given these parameters, a rotational motion estimation accuracy of 0.006 degrees or 1% of the rotational motion is expected.

4.3 Scale Estimation Mode

Descent angle and scene surface scale dictates which scale estimation mode to use during descent. Simulations were performed to determine at which descent angle the transition between scale estimation modes should occur. This angle is dependent on scene scale and is defined as the angle where translation magnitude errors of the two modes cross over.

The results of the simulation are shown in Figure 5. Inspection of the graph reveals that structure scale estimation should be used except when the surface is very flat (scale < 25 m at 1000 m altitude or 0.25% of altitude) or descent is very close to vertical ($\gamma > 88^\circ$). Using this plot, it is possible to determine which scale estimation mode to use before scale estimation is performed. Descent angle is fully determined from 5 DoF image-based motion estimation. The scene scale can be determined before descent then though 3-D modeling or analysis of laser altimeter readings. Given this descent angle/scene scale data point, the scale estimation mode can be looked up using Figure 5.

5 Conclusion

We have developed and tested a software algorithm that enables onboard autonomous motion estimation near small bodies using descent camera imagery and laser altimetry. Through simulation and testing on real data, we have shown that image-based motion estimation can decrease uncertainty in spacecraft motion to a level that makes landing on small, irregularly shaped, bodies feasible. Possible future work will include qualification of the algorithm as a flight experiment for the ST4/Champollion comet lander mission currently under study at the Jet Propulsion Laboratory. Current research is investigating the use of this algorithm to aid 3-D modeling of small bodies

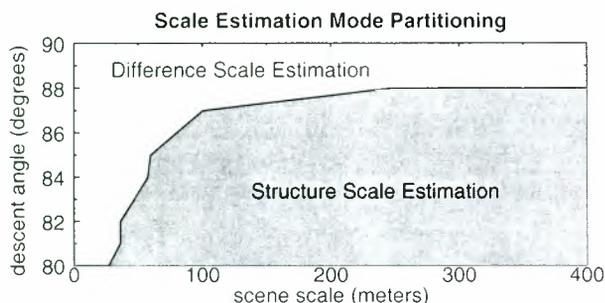


Figure 5: Scale Estimation Mode Partitioning from Monte Carlo Simulation.

for terrain hazard assessment and comet absolute position estimation.

The algorithm we have presented can be used to estimate motion with respect to any proximal surface. Consequently, it can be used for precision landing on comet nuclei, asteroids and small moons. It can also be used for proximity operations during rendezvous and docking between two spacecraft. Another application is estimating the altitudinal motion of an orbiter or satellite during precision pointing to surface targets. Rotational motion is completely determined from image-based motion estimation, so a laser altimeter is unnecessary for this application.

References

- [1] A. Azarbayejani and A. Pentland. Recursive estimation of motion structure and focal length. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562-575, June 1995.
- [2] A. Benedetti and P. Perona. "Real-time 2-D feature detection on a reconfigurable computer." *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 586-593, 1998.
- [3] R. Hartley. "In Defence of the 8-point algorithm." *5th Int'l Conf. on Computer Vision (ICCV'95)*, pp. 1064-1070, 1995.
- [4] H. Longuet-Higgins. "A computer algorithm for reconstructing a scene from two projections." *Nature*, vol. 293, pp. 133-135, September 1981.
- [5] J.K. Miller et al. "Navigation analysis for Eros rendezvous and orbital phases." *Journal Astronautical Sciences*, vol. 43, no. 4, pp. 453-476, 1995.
- [6] J.E. Reidel et al. "An autonomous optical navigation and control system for interplanetary exploration missions." *2nd IAA Int'l Conf. on Low-Cost Planetary Missions*, Laurel MD, IAA-L-506, 1996.
- [7] J. Shi and C. Tomasi. "Good Features to Track." *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 593-600, 1994.
- [8] R. Szeliski and S.B. Kang. Recovering 3-D shape and motion from image streams using non-linear least squares. *Journal Visual Communication and Image Representation*, vol. 5, no. 1, pp 10-28, March 1994.
- [9] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal Robotics and Automation*, Vol. RA-3, No. 4, pp. 323-344, August 1987.
- [10] J. Weng, T. Huang and N. Ahuja. "Motion and structure from two perspective views: algorithms, error analysis and error estimation." *IEEE Pattern Analysis and Machine Intelligence*, vol 11, no. 5, pp. 451-476, 1989.
- [11] J. Weng, N. Ahuja and T. Huang. "Optimal Motion and Structure Estimation." *IEEE Pattern Analysis and Machine Intelligence*, vol 15, no. 9, pp. 864-884, 1993.
- [12] Z. Zhang. "Determining the epipolar geometry and its uncertainty: a review." *Int'l Jour. Computer Vision*, 1997.

We would like to thank Jean-Yves Bouguet for discussions on motion estimation. We would also like to thank Jackie Green for providing the comet analog.

Advanced Space Robot Technologies

A DEXTEROUS GRIPPER FOR SPACE ROBOTICS

Claudio Bonivento, Claudio Melchiorri, Gabriele Vassura*,
DEIS, DIEM* - Università di Bologna, Viale Risorgimento, 2, 40136 Bologna, Italy
phone +39 051 6443034, fax +39 051 6443073, e-mail: cmelchiorri@deis.unibo.it

Gianni Ferretti, Claudio Maffezzoni, GianAntonio Magnani
DEI - Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy
phone +39 2 2399 3682, fax +39 2399 3412, e-mail: ferretti@elet.polimi.it

Giuseppe Beccari, Stefano Caselli, Francesco Zanichelli
DII - Università di Parma, Viale delle Scienze 181A, 43100 Parma, Italy
phone +39 521 905724, fax +39 521 905723, e-mail: caselli@ce.unipr.it

ABSTRACT

The paper summarises the work done so far by four University groups involved in a joint project for the development of a medium-complexity robotic gripper, respectively developing the mechanical design (DIEM-Bologna), the sensory and control system (DEIS-Bologna), the dynamic simulation (DEI-Milano) and the sensor fusion (DII-Parma). The gripper has been designed in order to perform low- and medium-complexity space-lab manipulation tasks, aiming to achieve a trade-off between simple twin-jawed grippers and highly sophisticated multi-degree-of-freedom hands. It presents a three-finger, three-degree-of-freedom architecture and is capable of synchronous application of the grasping contacts, so that force-closure grasps can be achieved on irregularly shaped objects even floating in micro-gravity conditions. Proximity sensors and intrinsic contact force sensors installed on each finger can allow object shape recognition together with control of approaching and grasping procedures. The capability of being tele-operated is addressed as well as the possibility of accomplishing elementary tasks in autonomous way.

1. INTRODUCTION

The convenience to accomplish simple and routine space-lab activities by artificial facilities and not by astronauts is being currently acknowledged for many reasons, not least the cost of astronauts' labour. Robotic arms equipped with suitable end-effectors could substitute human intervention in many activities, operating autonomously or being tele-operated from less expensive earth-based staff.

Purposely designed facilities, e.g. pay-load tutors [1], should provide for a structured environment where the robotic accomplishment of experiments would result greatly helped.

Due to the fact that the experiments could involve the manipulation of objects of quite different nature, (not only

"technical" objects but also biological or natural items) the determination of a suitable gripper architecture must be inspired to very particular issues, including the capability of shape adaptation with fine control of grasping forces (soft grasping).

To this purpose, dextrous articulated hands can be considered a suitable solution for future space application and great effort is still given to their development [2,3]. As a matter of fact, anthropomorphic hands seem to be highly performing solutions for generalised skilled tasks, both for their intrinsic versatility and for being the easiest man-interfaceable end-effectors for tele-operation.

However, in a wide class of space manipulation tasks, a high kinematic complexity of the gripper could not be necessary or, when available, be very partially exploited.

A family of intermediate configuration grippers, not so elementary as twin jaw grippers but not so complex as multi-fingered articulated hands, could therefore represent a valid trade-off between the achievable functional capability and the overall complexity, bulk and cost. Efforts in this direction have been proposed even in recent years (see for example [4-6]).

The activity reported in this paper, jointly performed by four University research groups with financial contribution by A.S.I. (Italian Space Agency), was specifically addressed to define a medium-complexity gripper and to test its actual manipulation capability in the perspective of use in intra-vehicular experiments in micro-gravity conditions, with levels of autonomous operation as well as remote operation capability.

2. THE GRIPPER ARCHITECTURE

The choice of the kinematic architecture of the gripper has been conditioned by the assumption that three-point adaptable grasps should be the main design goal and that no more than three actuators should be used. Another mandatory goal was to get the possibility of simultaneous

application of the contacts on the object to be grasped: as widely demonstrated by simulation, a synchronous application of the constraints can help reducing uncontrolled movements of objects freely floating in micro-gravity space during the approach phase.

The mechanical design has been developed, so far, assuming size and weight compatibility with the ASI Spider Arm [7].

A detailed discussion about the choice of the kinematic configuration has been reported in [8-10]. We present here the final features that are being implemented in the gripper prototype, that has been designed for laboratory evaluation and does not cope yet with space application specifications as to materials, sub-components and processes.

The general architecture of the gripper is shown in Fig. 1.

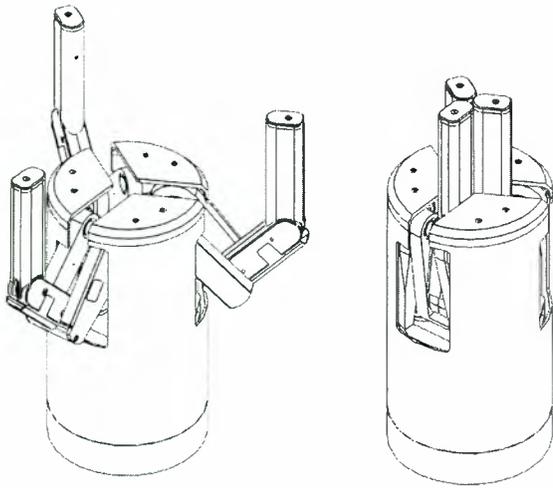


Fig. 1 A general view of the gripper

Three articulated fingers are equally spaced and contacts can occur along three intersecting coplanar lines. Each articulated finger has a distal phalanx that gets in touch with the object and two intermediate phalanxes, coupled by means of cable transmissions, that allow translation of the distal link.

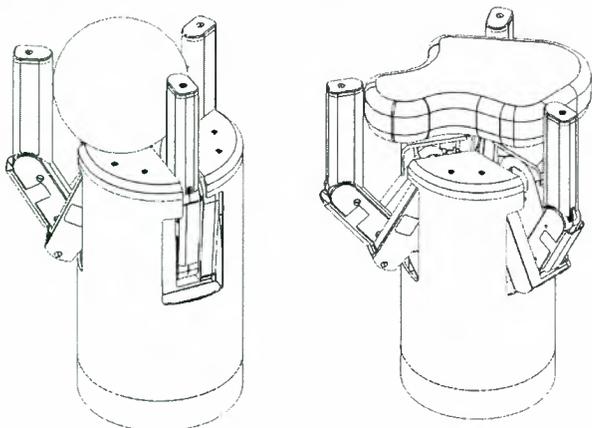


Fig. 2 Two examples of possible grasp configurations

In addition to the advantage of using only revolute pairs, this kind of kinematic structure presents a high ratio between maximum and minimum extension, obtaining a

very large workspace with respect to the size of the gripper body.

Because the three fingers can move independently, the grasping configuration may be any triangle having vertices on the approach trajectory segments. Two examples of possible grasp configurations are shown in Fig. 2. The gripper adopts a modular architecture. A view of each finger module is shown in Fig. 3, together with a scheme of the internal cable transmission. The actuation of fingers is provided by three linear actuators manufactured by Wittenstein GmbH according to the model of the artificial muscle developed at DLR [2]. A purposely designed sensor based on a Hall effect transducer is connected to the rod for position measurement. Further details about the mechanical transmission can be found in [8-10].

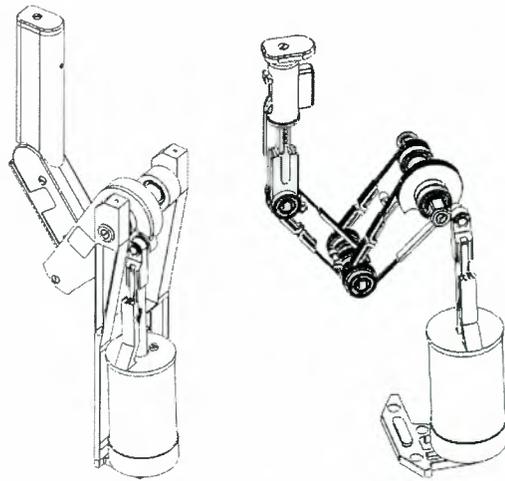


Fig. 3 The single finger module and its internal cable transmission

The sensory equipment installed in each finger is visible in Fig. 4, and consists of an optical proximity sensor and a miniaturised intrinsic tactile sensor (based on a multi-component force/torque sensor) [11]. This basic equipment allows the control of approach movements of each finger, with simultaneous reach of contact, and the control of grasping forces once the contacts have been applied.

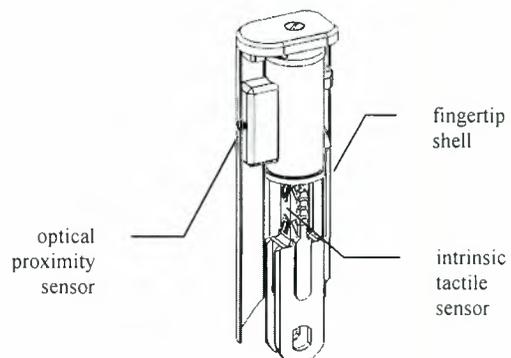


Fig. 4 The sensory equipment placed on each fingertip
Being capable of detecting not only the intensity of contact

force components but also the position of the contact centroid on the external surface of the finger, the intrinsic tactile sensors allow for efficient recognition and control of contact conditions, included incipient sliding. This basic arrangement does not exclude the possibility of further integration with additional sensory equipment, like distributed tactile sensors, stereo vision or more sophisticated scanning devices.

3. COMMENTS ON THE MECHANICAL DESIGN

A potential drawback of the adopted kinematic configuration is that the achievable three-point force-closure (precision grasp), even if adaptable and synchronous, may be not sufficient to satisfy the demand of grasp robustness in space manipulation. This could be better guaranteed by form-closure configurations (encompassing grasps). Without back-drawing from the initial choice of using the few available degrees of freedom in order to get, first, synchronous precision grasp (useful in micro-gravity operations), some solutions are under evaluation in order to add some level of form-closure capability. A simple solution might consist in purposely shaping one fingertip in order to allow multiple contacts along the same finger.

Other solutions, requiring at least an additional degree of freedom, could be obtained by changing the reference angle of the internal fixed pulleys that determine the posture of the distal phalanx or the direction of its approach linear trajectory. In both cases the front surface of the gripper body should act as a palm surface (Fig.5).

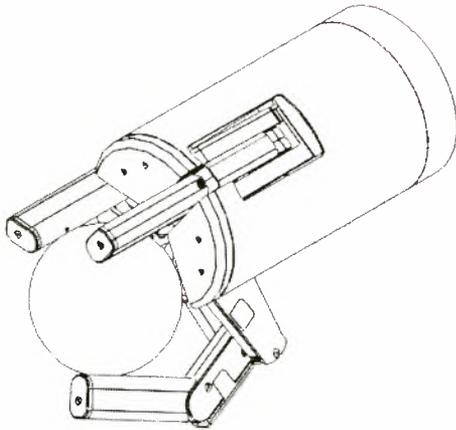


Fig. 5 Form closure by changing the approach trajectory of one finger

As to the prototype design, it can be observed that its final size has been conditioned by the use of off-the-shelf actuators that were oversized with respect to the actual need of thrust and stroke. A purposely design of the actuators could further reduce the size and the bulk of the gripper body, particularly its length.

In conclusion, the proposed robotic gripper exhibits some interesting features that can make it suitable for application

whenever the operating capability of a parallel jaw gripper is not adequate and, at the same time, the complexity of a multi-fingered articulated hand is not acceptable. The main advantages of such a device are:

- it is not very complicated as to kinematics, actuation and control, using only three actuators and three degrees of freedom;
- it can provide adaptable and synchronous application of contacts to objects of any shape, thus allowing to grasp objects not centred with respect to the gripper axis of symmetry, without disturbing their initial posture;
- it presents a very large workspace with respect to its body size, and is capable of operation both on small and on large objects;
- force-closure grasps can be integrated by some capability of form-closure grasp;
- it adopts a sensory equipment suitable for allowing both autonomous and tele-operated procedures by means of a three-finger interface.

4. OPERATION AND CONTROL

The gripper will be tested both in autonomous operation and as tele-operated system. For the former goal, several strategies are under evaluation and will exploit the available sensory capability and the possibility of independent controlled motion for each finger.

A typical autonomous task could be articulated in the following phases:

- approach motion in fully open configuration to the space region where the object is expected to be;
- object surface scanning by means of the proximity sensors mounted on the fingers: this operation should be aided by combined movements of the fingers along their approach direction and of the robotic arm and wrist; the object should not be touched in this phase;
- choice of the optimal grasp configuration, to be computed by means of proper algorithms for the optimisation of the three-point grasp;
- synchronous application of contacts and control of the grasping forces during the manipulation of the object.

Concerning tele-operation tasks, the definition of a suitable interface with the operator is a major issue. A solution currently under evaluation consists in a set of three wire-driven fingertip interfaces [12]. Haptic sensations connected to the three contact grasp on a virtual object can be reproduced by proper control of wire tension and elongation. Due to the different kinematics of the gripper and of the human fingertips, the virtual object should be properly scaled and modified with respect to the real object. Work in this direction is at a very early stage.

As to the set-up of the gripper prototype, in this initial phase of activity it has been decided to use, as long as possible, standard hardware/software components for controlling the gripper and evaluating its capabilities. The adopted architecture consists in a PC equipped with a DSP (TMS320C32) board and connected with the motor drives

and to an input board for the sensors. This board has been purposely designed because of the relatively high number of signals (30) to be acquired in real-time. From the software point of view, besides a real-time kernel on the DSP board, an interface between the DSP and the PC has been developed, allowing to use in an integrated fashion both real-time software and high-level environments for user interface.

5. DYNAMIC MODELLING, AND SIMULATION

Dynamic simulation is based on MOSES, a Modular Object-oriented Software Environment for Simulation developed at DEI. In MOSES, dynamic models of multi-body systems are built by assembling basic models (modules) and their aggregates through a graphic interface. Modules are defined through a Model Definition Language (MDL), which is "natural" for the analyst and fully "declarative". This is opposed to a "procedural" form, where a relation of causality between inputs and outputs is made explicit. Furthermore, standard module interfaces are defined to avoid the replication of models of the same physical component in case of different boundary conditions or different sets of exported variables. The complexity and variety of the data defining a complex technological are managed in MOSES by structuring modelling data in an Object-Oriented database.

Since for complex systems the raw assembly of declarative equations results in a largely redundant DAE system, a symbolic manipulation software has been developed in order to gain computational efficiency [13], [14]. The symbolic manipulation essentially aims at splitting the global DAE system into almost independent subsystems and at minimising the order of the implicit system to be solved.

On the other hand, when the modelling of mechanical systems is of concern, tools for 3D solid modelling turn out to be essential for both model building and 3D rendering of motion. To this aim, a 3D solid modelling interface, ROSE (Robot Solid modelling Environment) has been developed. It allows the geometric and kinematic modelling of a robotic system and generates the topological data for the automatic generation of the MOSES dynamic model. ROSE was designed to strictly match the modular approach of MOSES, in terms of a one-to-one correspondence of the graphical entities with their MOSES dynamic modules (or sub-models) counterparts and in terms of a correspondence between the data structures of both environments.

The dynamic model of the mechanical structure (plant) is then automatically generated in the MOSES environment, where a control system can be also attached to the plant. The ultimate output of the simulation environment is the 3D rendering of motion, implemented in VRML (Virtual Reality Modelling Language). This has been made possible by the adoption of a solid modelling library based on a

boundary representation of solids which matches the VRML format.

The Gripper Model and the simulation of grasp

Each finger of the 3 dof gripper is a serial chain of three links connected by rotary joints and additionally constrained by the tendons, which maintain the last phalanx parallel to the wrist (approach) axis while moving. Actually, the tendons have been modelled as extensible, taking into account their elastic stress/elongation characteristics.

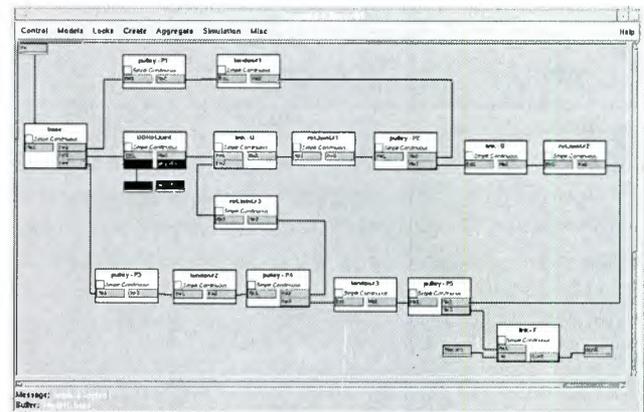


Fig. 6. The block scheme of the finger model

The chain structure is reflected in Fig. 6, where the MOSES block scheme of the model finger is shown. Each module in Fig. 6 defines a simple model and represents a well identified part: rigid body (links and pulleys), rotational joints, tendons. Aggregation is performed by establishing a one-to-one connection between two identical ports: the mechanical terminals.

Note that three mechanical terminals are not connected to any other terminal and can be reported at the upper level. These free terminals of the finger aggregate model are available for connecting the finger to the wrist sub-model and for representing the interaction of the finger with the grasped object sub-model, respectively.

As an example of achievable simulation results, let us consider the grasp of a cylinder in the absence of gravity, shown in Fig. 7. As it is clear from the Figure, which shows the co-ordinate of the centre of mass of the cylinder along the axis, immediately before $t=5$ s fingers 2 and 3 hit the cylinder, which starts moving towards finger 1. Then, after some bounces, all the fingers come in contact and the cylinder stops.

6. SENSOR FUSION TECHNIQUES

The control and tele-programming architecture of the gripper incorporates a technique for fusing information provided by tactile and IR proximity sensors. The technique is based on the construction of volumetric approximations of the explored object and enables

approximate shape reconstruction, recognition of convex subparts, and generation of efficient exploration strategies [15], [16]. Multiple polyhedral representations of the object are exploited to deal with different types of sensory information.

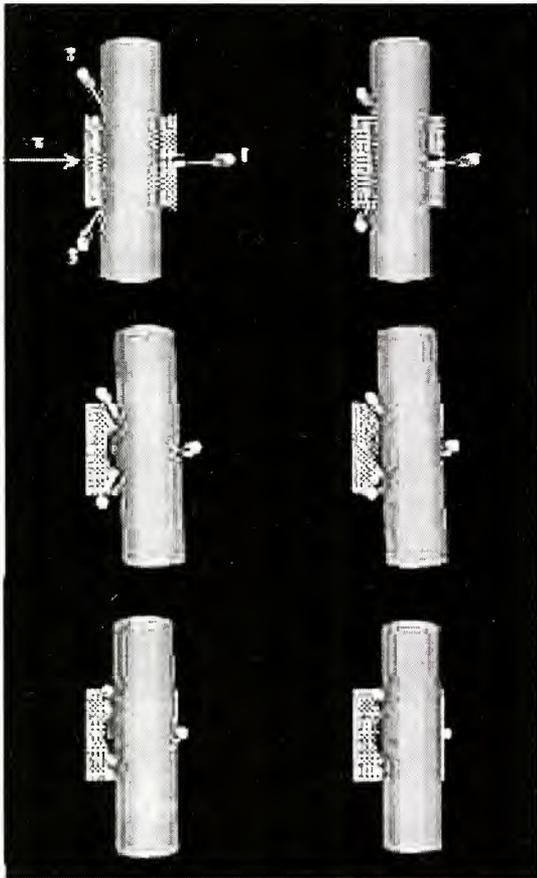


Fig 7 Animation of a grasp simulation in absence of gravity

The sensor fusion technique can cope with arbitrary shape classes, but may require prior volumetric segmentation to enable direct shape reconstruction and recognition of convex subparts. The *Enveloping Polyhedral Model* (EPM) is an upper approximation of the object exploiting perceived contact locations and normal directions at the contact points provided by IT sensors. At the beginning of the exploration an EPM equals the whole workspace. As contact points are accumulated, the EPM volume decreases, because of the "slicing" effect of contact planes on the workspace. The shape approximation returned by the EPM can be further refined by taking into account the partial containment of the object inside the volume spanned by the gripper's palm and fingers. The *Inscribed Polyhedral Model* (IPM) is built from proximity information provided by IR sensors and contact locations returned by IT sensors, but does not require contact normals (which could be noisy or unavailable). Under the hypothesis of part convexity, this representation is computed as the convex hull of the available data points. Since an IPM is always contained in the corresponding

EPM, their joint availability enables efficient recognition and exploration strategies [16]. Figure 8 shows the concurrent refinement of EPM and IPM on increasingly larger data sets.

7. TELE-PROGRAMMING ARCHITECTURE

The reference architecture for tele-programming and monitoring of the gripper consists of a *client-server architecture* where the server and one or more remote client systems are interconnected using TCP/IP. Use of a standard IP-based protocol enables direct exploitation of any enhancement in quality of service brought by Internet-related technologies. Recently, a number of tele-robotics projects have been based upon Internet infrastructures, including projects aiming at monitoring of space-robot operations, e.g. [17].

The *server* system application can operate directly on the same PC hosting the DSP for gripper control, or on a separate, locally-connected workstation providing a single access point to both the gripper and the carrying arm. The server has been implemented as a multithreaded C++ application based on the OmniORB2 multi-platform library and is fully portable across most standard operating systems. The server supports activation of gripper motion tasks with local control or under tele-operation, remote system supervision during task execution, and visualisation of the current operating environment, i.e., live feed of raw or processed sensory data. The server architecture comprises a number of concurrent threads providing system supervision, network interfacing, authentication and synchronisation of incoming client requests, and interaction with the local control system of the robotic device, while actively managing Quality of Service [18].

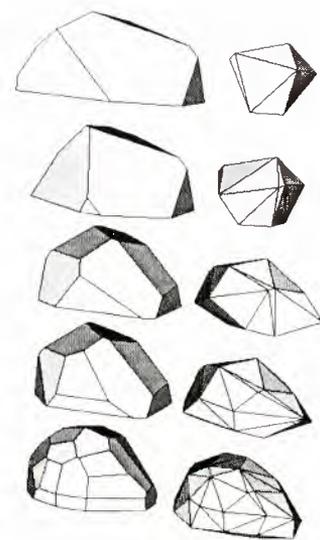


Fig.8 Refinement of EPM and IPM

A Java-based *client* is under development to assist users in their remote interaction with the gripper by means of the

server system application. The client program offers a main set of functional modules including parameterisation and execution of dynamic simulations, task programming, monitoring of task progress, and 3D visualisation of sensor fusion. Task programming is currently supported only in terms of specification of elementary finger and gripper motions, possibly terminated by a sensory condition. Task progress is shown by providing the user with a VRML animation of the scene fed with the current pose of the gripper as obtained from the server. Raw and filtered sensory data from the robotic system are also available and can be parametrised by the user, including an optional live feed from a video camera. As an additional feature, the haptic perception of the gripper can be shown in terms of the current EPM and IPM built from the fused sensory information in the same VRML scene including the gripper. The adopted client implementation technologies, namely Java and VRML 2.0, are *de facto* standards oriented to mobility and interoperability.

7. CONCLUSIONS

The current development stage of a project for the design and implementation of a dextrous gripper oriented to space applications has been described. The gripper presents some interesting features that can make it suitable for precision grasp on known-geometry objects as well as for adaptable synchronous grasp on irregularly shaped objects. Together with the development of a prototype, tools for dynamic simulation, sensor fusion and tele-programming have been defined.

The results obtained so far encourage the prosecution of the programme.

After the completion of the prototype, expected at the end of June 99, activity will mainly be devoted to laboratory experiments in autonomous grasping operations, with the gripper mounted on a Comau SMART 3S arm.

In parallel, a three-fingered wire-driven interface will be developed in order to perform tele-operation experiments.

Acknowledgements

Funds for this research have been provided by ASI (Italian Space Agency). The contributions of F. Fabbri, S. Roviero, S. Monti and G. Nannoni to the development of the prototype are gratefully acknowledged.

References

- [1] S. Di Pippo, R. Mugnolo, W.B. Williams, G. Colombina, E. Pozzi, "Payload Tutor (PAT) a Relocatable Payload Robot for ISS Internal Automation System", ASTRA'98, ESTEC, NL, Dec. 1-3, 1998.
- [2] M. Fischer, G. Hirzinger, "Fast Planning of Precision Grasps for 3D Objects" IROS 97 Int. Conf. on Intelligent Robots and Systems, Grenoble, France, 1997
- [3] B.M.Jau, "Dexterous Telemanipulation with Four Fingered Hand", IEEE Int. Conf. on Robotics and Automation, May 1995, Nagoya, Japan
- [4] [3] K. Machida, Y. Toda, Y. Murase, S. Komada, "Precise Space Telerobotics System Using 3-Finger Multisensory Hand", IEEE Int. Conf. on Robotics and Automation, May 1995, Nagoya, Japan
- [5] Barrett Technologies, Inc.
<http://www.barrett.com/robot/products/hand/handspfr.htm>
- [6] Mitsushige Oda, "Space robot experiments on NASDA's ETS-VII satellite - Preliminary overview of the experiment results-" IEEE Int. Conf. on Robotics and Automation, Detroit, MI, May 10-16, 1999
- [7] R. Mugnolo, S. Di Pippo, P.G. Magnani, E. Re, "The SPIDER Manipulation System (SMS): the Italian Approach to Space Automation", Int. J. on Robotics and Autonomous Syst., Vol. 23, No. 1-2, 1998.
- [8] G. Vassura, C. Melchiorri, "A Three-Contact, Three Degree-of-Freedom Gripper for Robotic Manipulation in Micro-Gravity Conditions", 6th IASTED Int. Conf. on Rob. & Man., Banff, CA, July 26-31, 1998.
- [9] C. Melchiorri, G. Vassura, "Design of a Three-Finger Gripper for Intra-Vehicular Robotic Manipulation", 1st IFAC Workshop on Space Robotics, SPRO98, Montreal, CA, Oct. 19-22, 1998.
- [10] C. Melchiorri, G. Vassura, G. Ferretti, C. Maffezzoni, G. Magnani, P. Rocco, "A Three Finger, Three Degree-of-Freedom Gripper for Space Robotic Manipulation", 5th ESA Work. on Adv. Space Technologies for Rob. and Aut., ASTRA 98, ESTEC, NL, Dec. 1-3, 1998.
- [11] A. Cicchetti, A. Eusebi, C. Melchiorri, G. Vassura, "An Intrinsic Tactile Force Sensor for Robotic Manipulation", Proc. 7th. Int. Conf. On Adv. Robotics, ICAR'95, Spain, Sept. 20-22, 1995.
- [12] C. Melchiorri, G. Vassura, P. Arcara, "What Kind of Haptic Perception Can We Get with a One-Wire Interface?", IEEE Int. Conf. on Robotics and Automation, ICRA 99, Detroit, MI, May 10-16, 1999
- [13] Carpanzano, E. and Maffezzoni, C. "Symbolic manipulation techniques for model simplification in Object-Oriented modelling of large scale continuous systems". *Mathematics and Computers in Simulation*, 48, 1998, pp. 133-150.
- [14] Maffezzoni, C., Girelli, R. and Lluca, P., "Generating efficient computational procedures from declarative models", *Simulation Practice and Theory*, 4, (1996), pp. 303-317.
- [15] C. Caselli, C. Magnanini, F. Zanichelli, E. Caraffi, "Efficient Exploration and Recognition of Convex Objects Based on Haptic Perception," IEEE Int. Conf. on Robotics and Automation (ICRA'96), Minneapolis, MN, April 1996
- [16] G. Beccari, S. Caselli, F. Zanichelli, "Pose-Independent Recognition of Convex Objects from Sparse Tactile Data," IEEE Int. Conf. on Robotics and Automation (ICRA'97), Albuquerque, NM, May 1997.
- [17] P.G. Backes, et al., "Mars Pathfinder mission Internet-based operation using WITS," IEEE Int. Conf. on Robotics and Automation (ICRA'98), Leuven, Belgium, May 1998.
- [18] G. Beccari, S. Caselli, M. Reggiani, F. Zanichelli, "A Real-Time Library for the Design of Hybrid Robot Control Architectures," IEEE-RSJ IROS'98, Victoria, CAN, 1998.

ON-BOARD PERCEPTION PROCESSING FOR SPACE ROBOTS

Yasufumi Wakabayashi*, Makoto Miyata*, Takeshi Nishimaki**

*National Space Development Agency of Japan(NASDA), **AES Corporation

2-1-1, Sengen, Tsukuba, Ibaraki, Japan

E-mail : Wakabayashi.Yasufumi@nasda.go.jp

Phone : +81-298-52-2369 Fax : +81-298-52-2415

ABSTRACT

The progress of embedded system technologies in multi-media consumer electronics is likely to advance the perception processing capability in space robots. Small size boards, on which 1M-gate system-ASICs with several 10MIPS MPU core and multiple communication bus controller, and several 10MB memory with various middlewares, is ready for their use in space applications. In this paper, advantages of these technologies and some results of flight and laboratory experiments are discussed.

1. INTRODUCTION

Implementation of perception functions in a robot is divided into two categories; the one is multi purpose type such as eyes (cameras) and ears (pickups), and the dedicated type such as force sensors, rendezvous & docking sensors, 3D sensors. The latter is required in the realtime controlled, robust and heavy-duty applications and is also effective to enhance the robot function in particular applications, though it costs in many points of view. On the other hand, the former is useful to acquire general perceptual information. This paper describes the applications of highly processed perceptual data in section 2, onboard processing visual data in section 3, onboard processing vibrational data in section 4 and onboard processing force data in section 5.

2. APPLICATIONS OF PERCEPTUAL PROCESSING

Highly processed perceptual data offers robots many advantages in various situations such as measurement, data transfer, inspection / observation and supervision.

(1)Measurement

The usefulness of onboard visual measurement is apparent. For example in ORU exchange tasks, marker-base operation is used with hand-eye / wrist-eye against a dedicated small 3D-marker on the object. In target

oriented attitude control in near field rendezvous, tracking-vision technique can be applied. Furthermore, for non-marker objects, pattern-matching technique and stereo-measurement are considered.

(2)Data transfer

Enabling low telemetry rate operation is important, because the necessity of large communication capacity brings considerable penalties to the system. Several 10Kbps rate is a current target for the study, and middlewares such as JPEG, JBIG, MPEG, etc., instead of dedicated LSIs, enable to use selectable image-data-compression depending on particular operation. For realtime applications, the onboard processing is a priority and high quality data is dispensable for operators; therefore, large data such as a fine imagery has to be transferred to the ground during non-realtime operation. Eventually the reduced data rate operation could be realized by adequate compression technique to each task. Furthermore, the compression technique also can be implemented in the internal data transfer in a robot.

(3)Inspection / Observation

Onboard inspection of satellites and/or equipment in orbit and precise observation of specimen in planetary explorations, are key tasks in space robot application. The onboard perception processing can be applied in these tele-science regions. Though the sensors would collect enormous data for high-resolution and sampling terms, the scientists or engineers on ground may need only the remarkable data for them. Eventually pre-processing must be done by on-board for low telemetry rate operation and effective execution of such tasks.

(4)Supervision

NASDA, office of R&D, has been investigating for years the onboard supervision technique. A space robot

has to execute a task or an action without any damages. Even without any malfunction in hardware and software system, an action could cause damage when the total situation/condition is misunderstood by commanders, operators and/or computers. The move-and-wait strategy and very low speed operation might be the answer; however the independent onboard supervisor by highly processed perception data could reduce the difficulty of overall recognition of situation and realize more efficient operational scheme.

3. ONBOARD PROCESSING OF VISUAL DATA

A camera system with middleware JPEG is developed and introduced here as onboard processing of visual data [2]. COMETS (COMmunication Engineering Test Satellite : Data Relay Satellite of Japan), launched on February 1998, has camera system for monitoring the deployment of the large scale deployment structures; the Solar Paddle and the Antenna. This system was developed based on the camera developed for the future use in space robots. The specifications of this camera are shown in TABLE.1. RISC (Reduced Instruction Set Computer) microprocessor of commercial parts is adopted and placed in this control unit. This processor executes the commands from the ground station, creates the telemetries including the compressed imagery data and transfers them to the ground station.

To decrease the amount of telemetry, the imagery data is compressed by JPEG and compression rate can be changed by the command to adjust the quality of the image.

Within the camera head is a 1/3 inch color CCD which has 320 thousand pixels, which horizontal resolution is 480TV lines at non-compression mode. The camera is equipped with many functions such as auto / manual of the electrical shutter, gamma compensation and so on. The size of the camera head is 88(mm)×81.5(mm)×141.7(mm) and that of the camera control unit is 210(mm)×245(mm)×113(mm). The weight of the camera head is about 0.9(kg) and that of the camera control unit is about 5.3(kg).

During this mission, a bit-change occurred a few times in a portion of the image data. This anomaly is thought to be SEU (Single Event Upset) which damaged JPEG's restart marker. The camera was operational during the entire mission. The actual imaged picture on the orbit is shown in Fig.1. Development at the embedded processing of visual data is underway adopting pattern-matching for non-marker objects and stereo-measurement.

TABLE.1 Specifications of COMETS camera system

Item	Specification
Form	Total Pixels : 320 thousand pixels CCD size : 1/3 inch Color CCD
Horizontal Resolution	480TV lines
Image Data Interface	Serial Digital Telemetry Data Rate : 200bps
Image Buffering Capacity	16 Frame (MAX : 1.2M byte)
Compression Method	JPEG
Control Function	Camera ON / OFF Compression Rate : 8 levels Electric Shutter : Auto/Manual (1/30~1/10000 second)
Size	Camera Head : (4 head) 88×81.5×141.7 (mm) Camera Control Unit : 210×245×113 (mm)
Weight	Camera Head : 0.9 (kg) ± 10 % Camera Control Unit : 5.3 (kg) ± 7 %
Power Consumption	Camera Head : 1 (W) Camera Control Unit : 12 (W)

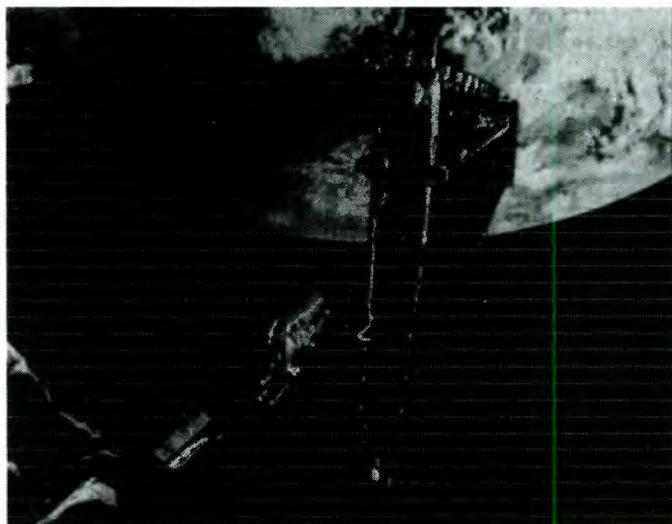


FIG.: 1 The actual imaged pictures on the orbit

4. ONBOARD PROCESSING OF VIBRATIONAL DATA

For the purpose of detecting unusual events (for example, an unexpected collision) and others, the effectiveness of the onboard signal processing, such as Wavelet analysis and STFT (Short Time Fourier Transform). If unusual events can be detected automatically, the method can be embedded into the onboard processing unit as the onboard supervisor. The FFT can find out the frequency information of the signal

but it cannot detect the specific time it occurred. On the other hand, the STFT and Wavelet analysis can detect the frequency information and the event time simultaneously. The STFT maps a signal into a two-dimensional function of time and frequency at equal intervals (Fig 2). Wavelet analysis allows the use of long time intervals where more precise low frequency information can be extracted, and shorter regions where high frequency information can be extracted (Fig.3). The vertical axis represents frequency and the horizontal axis represents time in both figures.

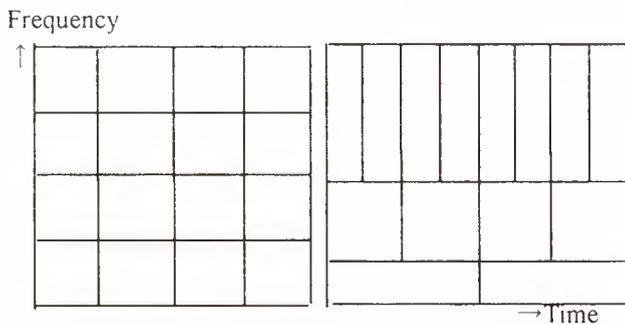


FIG.:2 STFT FIG.:3 Wavelet Analysis

The validity of these signal analysis techniques was tested with the 2-dimensional model of the multi-manipulator system [1] with 4-DOF (Degrees Of Freedom) arm (Fig.4). Accelerometers were fitted at the tip of the arm's 1-joint and received the signal in the PC with 4096Hz sampling frequency. The arm was manipulated by joysticks with about 10mm/sec speed to collide with an object (weighting about 1.0kg). After confirming the signal, the joint's motor was turned active at 1.0sec. Then the arm collided with the object at 2.5sec and collided several times again.



FIG.:4 The configuration of 2-dimensional multi-manipulator system

Fig.5 shows the result of Wavelet analysis (symlet8), Fig.6 shows that of STFT and Fig.7 shows that of FFT. The original signal is displayed at the top of the signal in Fig.5. The bottom signal, d1 has the lower half frequency information of the original. d2 has the lower half of d1, d3 has the lower half of d2, d4 has the lower half of d3, d5 has the lower half of d4 and d6 has the lower half of d5. Since the sampling frequency is 4096Hz, the signal under 256Hz appears in d4 and the signal under 64Hz appears in d6. In the d4 data, a large peak is recognized at 2.5sec (at 1×10^4 th point of data) and two small peaks at 4.0sec (at 1.6×10^4 th) and at 4.2sec (at 1.8×10^4 th). In the d6 data, the small continuous signals can be recognized from 1.0sec (at 0.4×10^4 th) to 2.7sec (at 1.1×10^4 th). The peak in d4 represents the arm's collision with the object, and the continuous signal in d6 shows the fact that the joint's motor was active in the duration.

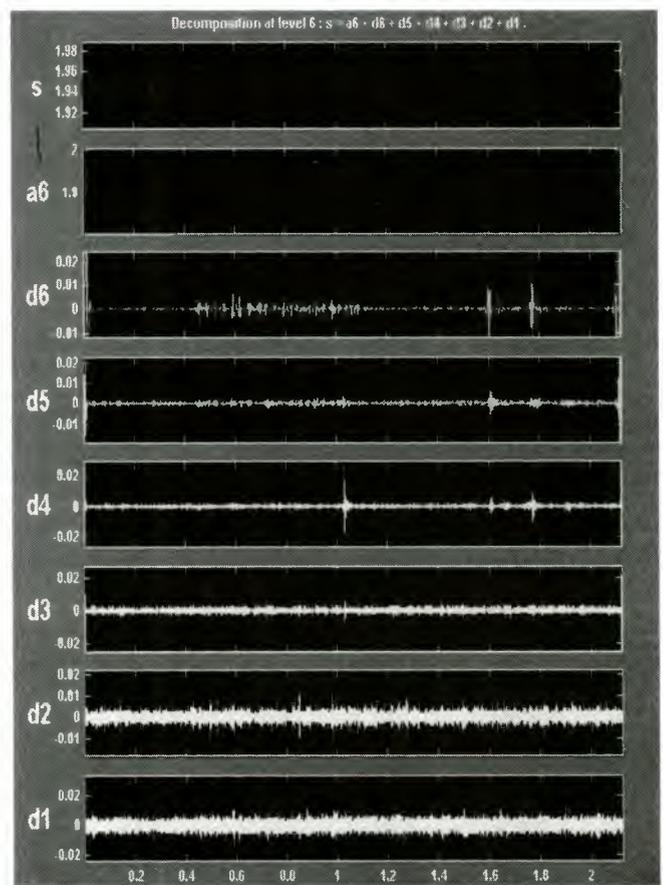


FIG.:5 The result of Wavelet analysis (horizontal axis : the number of data $\times 10^4$)

In the STFT analysis (Fig.6) example, there is a remarkable line at 2.5 seconds with a frequency of 100 ~250Hz, which represents the arm's collision with the object. And also there are two lines at 3.8 and 4.2 seconds, which frequencies distribute up to 250Hz.

These lines represent the arm's collision with the object again. This figure also shows another event at 1~2.5 seconds with 20~60Hz frequency, showing that the arm's motor was active.

Fig.7 (FFT) shows there are narrow spectrums under 20Hz and 25~65Hz. These are considered to be related to those phenomena described above, but do not show any information in time domain.

Eventually the validity of STFT and Wavelet processing of vibration data was confirmed to detect the frequency and the time on which the arm's collision occurred and the arm's motor was active / inactive.

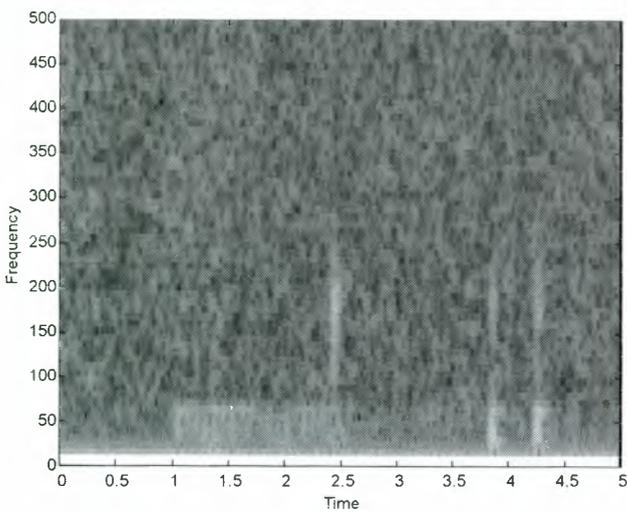


FIG.6 STFT analysis

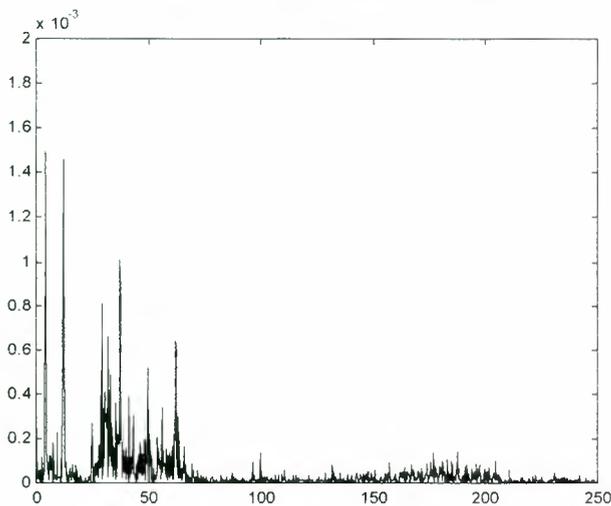


FIG.7 FFT analysis

Further offline analysis is executed in our experiment to study the possibility of the onboard perception by signal processing.

We measured the processing time of these data on the PC. The processing speed of the microprocessor is 300MHz (with DSP) and the Operating System is

WindowsNT4.0. Table.1 shows the processing time of Wavelet analysis. Each processing time was within 0.3sec. Table.2 shows the processing time of STFT. Each processing time was within 0.2sec.

TABLE.1 The processing time of Wavelet (sec.)

	Haar	Daubechies6	Symlets8
Level 3	0.13	0.16	0.18
Level 6	0.17	0.22	0.23
Level 9	0.21	0.26	0.27

TABLE.2 The processing time of STFT (sec.)

Window length	64	128	256
Overlapped			
Half	0.19	0.19	0.18

From the above results, 300MHz microprocessor (with DSP) could process 21248 data using Wavelet analysis or STFT within 0.3 seconds. If we can use the onboard perception processor of 30MHz microprocessor (with DSP), over 2000 data would be analyzed in a second. Meantime, the effectiveness of this continuous processing is evaluated further.

5. ONBOARD PROCESSING OF FORCE DATA

Various experiments were executed with ETS-VII (Engineering Test Satellite VII) and still underway. FTS (Force and Torque Sensor) is equipped with the manipulator on ETS-VII and is used for the compliance control, etc. The FTS data was analyzed to evaluate the effectiveness of the onboard measurement and supervision. The FTS data of ETS-VII exist in the 10Hz telemetry format. Fig.8 shows the force data in x-direction when the manipulator grasped the TBTL (TaskBoard ToolHead) and Fig.9 shows the torque data respectively. Various phenomena were recognized from these data.

Time	Events
1) 0--300	change of arm's activation of the motor
2) 346-> 3)	tool contact TBTL and trace on it
3) 460	finish tracing
4) 460-660	the arm control with x-directional force 20N
5) 617	latch of tool finger
6) 662	change of operation
7) 688	change of compliance mode
8) 740-800	torque of tool torquer
9) 795	latch of tool torquer
10) 880	change of motion (stop->move)

(from torque data)

- 617 latch of tool finger
- 662 change of operation
- 688 change of compliance mode
- 740-800 torque of tool torquer

Fig.8 shows the manipulator moved to x-direction 1mm/sec speed from 350 to 460sec and was controlled with x-directional force of 20N from 460 to 660sec for the grasping operation. The force increases lineally up to 24N at 460sec by the term of viscosity, equal to 3.75N. There are peaks at every 40 seconds from 460 to 660sec by the reset function of friction compensator. Fig.10 shows the x-directional data of the similar experiment on different occasion. These data have similar trends with respect to each action, such as the tool's contact to TBTL, the tool's trace on it, the tool finger's latch/unlatch, the change of manipulator's activation of the motor and the change of compliance mode etc. This analysis suggests that the FTS data changes nominally in particular pattern, which precisely reflects the context of robot action. Fig.11 shows the result of the wavelet analysis in certain frequency areas. The peak at 500sec (5000thdata) represents that the tool captured TBTL, and another peak at 700sec (7000thdata) represents the change of the arm's control mode.

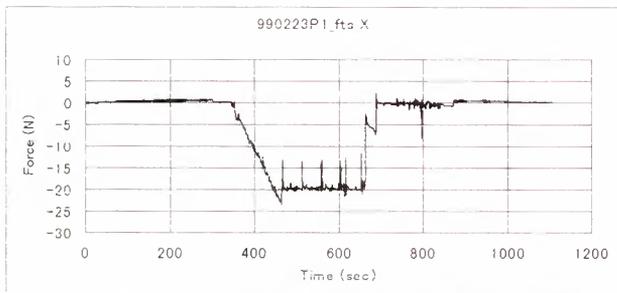


FIG.: 8 the force data of x-direction

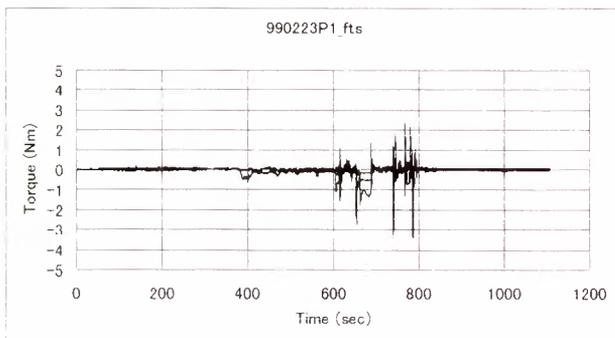


FIG.: 9 the torque data

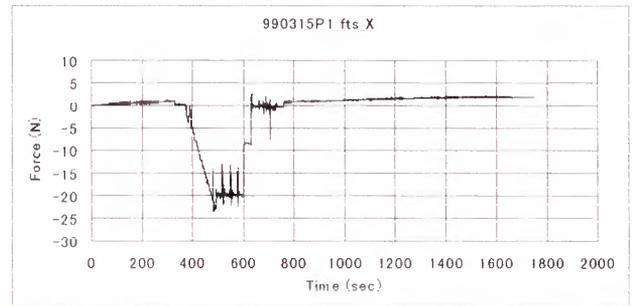


FIG.: 10 force data of x-direction of another day

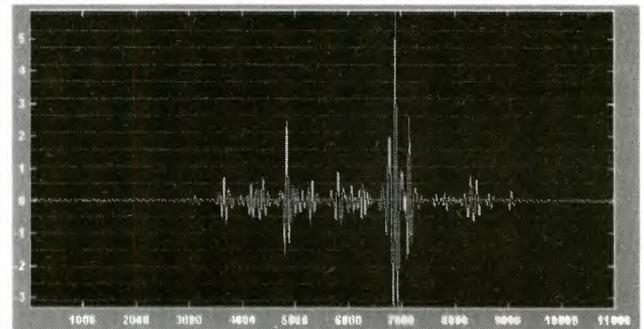


FIG.: 11 the result of the Wavelet analysis (sec.)

The FTS data from the target satellite berthing experiments was also analyzed. Fig.12 and Fig.13 show the forces and torques data of the experiments. In this experiment, at first the manipulator grasped the target satellite, then the docking mechanism(DM) was opened and finally the target satellite was separated from the chaser satellite. The target satellite was manipulated 600mm and next it moved to the original position and the DM was latched. The followings are the observed events from Fig.12 and Fig.13.

Time	Events
1) 280-420	opening of the DM
2) 700-1250	joint activation and arm's vibration
3) 1130-1180	arm's vibration
4) 1750-1900	closing of the DM

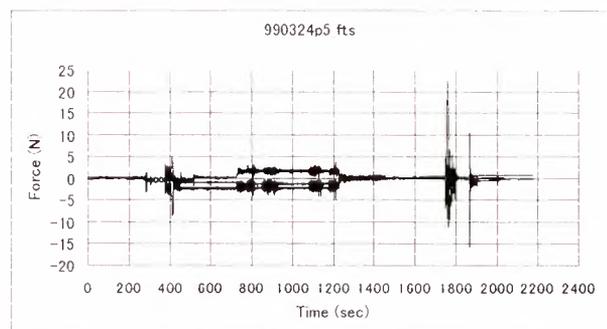


FIG.: 12 Forces data of the target berthing experiment

Fig.14 shows the expanded portion of Fig.12 (from 1100 to 1200sec). The result shows the arm's vibration after joint activation was finished (from 1140 to 1180sec). The vibration is about 0.11Hz which matches the result from the computer simulation.

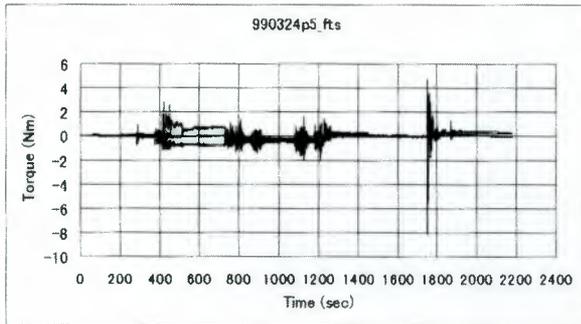


FIG. 13 Torque data of the target berthing experiment

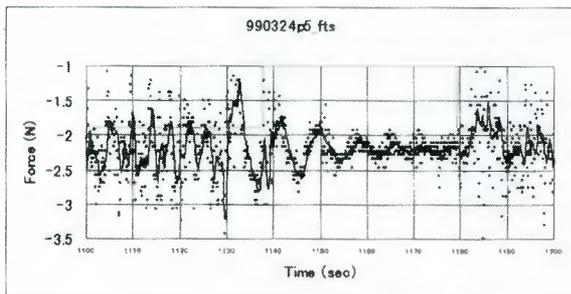


FIG. 14 Expanded figure of Fig.12 (1100sec~1200sec)

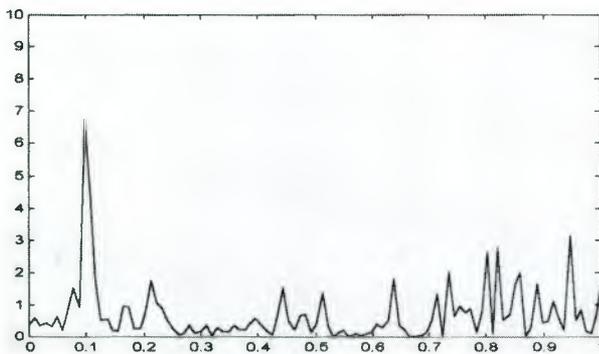


FIG. 15 FFT analysis on the data of Fig.14 (1100sec~1200sec)

Fig.15 shows FFT analysis on the data of Fig.14. There is the largest peak nearly 0.11Hz which is the frequency of the arm's vibration. Fig.17 shows the wavelet analysis of the force data of the x-direction which is shown in Fig.12. The same data is shown in Fig.16 for comparison. In Fig.17, as white is brighter, the frequency signal becomes stronger. As the position is upper, the component of the frequency becomes lower. There are the low frequency signal at near 400 and 1800 seconds. These signals reflect the DM latching and unlatching. This analysis suggests that the FTS data is

also effective for the detection of the vibrational characterizations.

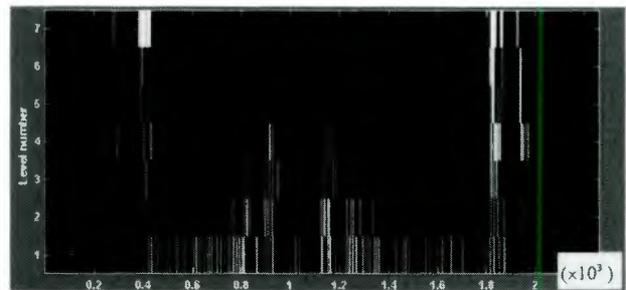
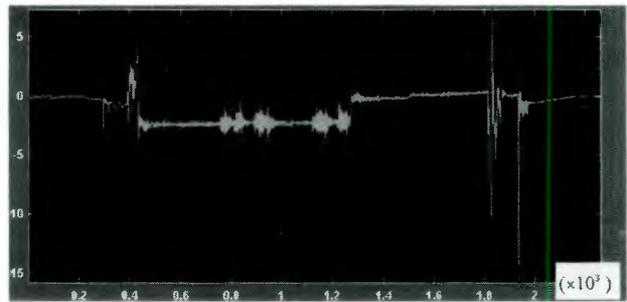


FIG. 17 Result of Wavelet analysis

6. CONCLUSIONS AND ACKNOWLEDGEMENTS

The study confirmed the middlewared JPEG is effective for onboard visual processing, and signal processing with Wavelet analysis and STFT is effective to detect the unusual collision. And the study also recognized the acoustic vibration and force data are effective for the detection of the nominal status, such as event and status, as well as for the detection of the vibrational characterizations.

Onboard perception processing is expected in future space robots, and some applications and laboratory analysis are introduced in this paper. We would like to recognize various supports from members of the related projects.

REFERENCES

- [1] Y.OHKAMI et al. "Research and Development of Reconfigurable Brachiating Space Robots", 8TH i-SAIRAS, 1999
- [2] K.Ozawa et al. "On-board monitor camera for COMETS", 42ND Space Sciences and Technology Conference of Japan, 1998
- [3] T.KASAI et al. "Results of the ETS-7 Mission-Rendezvous Docking and Space Robotics Experiments", 8TH i-SAIRAS, 1999
- [4] Y.WAKABAYASHI, et al, "A Design of Lunar Rovers for High Work Performance", 8TH i-SAIRAS, 1999

A Buyer's Guide to Forward Intersection for Binocular Robot Vision

Stephan Grünfelder

AUSTRIAN AEROSPACE GmbH, Department of Electrical Design
Stachegasse 16, A-1120 Wien, Austria
phone: +43 1 80199-0, fax: +43 1 80199-5577, e-mail: stephan.gruenfelder@space.at

Reinhard Krickl

Vienna University of Technology, Institute of Flexible Automation
Gusshausstrasse 27-29/361, A-1040 Wien, Austria
phone: +43 1 504 14 46-12, fax: +43 1 505 59 83, e-mail: rk@flexaut.tuwien.ac.at

Abstract

Binocular Robot Vision, the extraction of 3d-information from images from two distinct digital cameras, is subject to intensive research for space and terrestrial applications.

In space applications we mainly find tracking tasks (e.g. [1]) and explorative vision tasks (e.g. [2, 3]). Our paper focuses on reconstruction which is indispensable for the latter type and might be used for tracking as well. The paper presents the results of a basic research exercise.

Four different algorithms are presented that allow the reconstruction of an object point given its projections in two distinct digital cameras. The algorithms are compared with respect to absolute accuracy, relative precision, and computational requirements by means of simulations. From the simulation results we derive a rule of thumb that tells which algorithm to take for a given problem. Furthermore, the implementation on a Digital Signal Processor (DSP) is discussed and results using experimental data are given.

key words: robot vision, stereo reconstruction, DSP application.

1 INTRODUCTION

Many 3d vision tasks with stereo cameras need image feature matching and 3d reconstruction. If time and computational complexity are a minor issue, photogrammetrists will rely on extensive camera calibra-

tion and "bundle adjustment", a process which is mostly semi-automatic. It yields the highest precision rating by simultaneously performing object reconstruction, image feature matching, and, if desired, the computation of selected calibration parameters [4]. In space applications often the computational capability is very restricted with respect to memory usage and processing power. Furthermore, many applications require real-time performance and full automatic processes (e.g. visual servoing, scene perception for autonomous vehicles).

The first step to accelerate 3d reconstruction from stereo images is the gradual decomposition of the process into calibration, image feature matching, and pure geometrical reconstruction itself. Ideally calibration is needed only once before the images of the object of interest are captured but can be performed to some extent on the basis of the captured images, as well [3, 6]. Image feature matching heavily depends on the actual task and will not be examined in this paper. Methods vary from image patch distortion approaches via hierarchical procedures to knowledge based versions.

The geometrical reconstruction of an object point, given the two 2d image locations of its projections, in two distinct cameras is known as *forward intersection*. This process does *not* change with the application; only timing and accuracy requirements may vary. Forward intersection is needed when no a priori object model is at hand or when object/model matching is performed in 3d space. In reality the thus reconstructed 3d point won't match the exact object point because of imperfections of perception of the projection of the object

point and imperfect calibration.

Section 2 presents the mathematical formulation for the projection – the camera model – that has been used in our investigations. Section 3 describes the four investigated algorithms that solve the inverse problem, the forward intersection.

The quality of a reconstruction can be judged in two different ways. One is the deviation of the reconstructed point to the real object point, referred to as *absolute deviation*. The second is the divergence of a reconstructed point cloud to the real object point cloud, the *consistency deviation*. In fact, an object is reconstructed as a point cloud and the point cloud is rotated and translated, the consistency deviation remains constant but the absolute deviation changes. The absolute deviation includes the relationship between the object coordinate system and the camera coordinate system – needed for example for autonomous navigation tasks.

The accuracy of the forward intersection is influenced by 2×10 different camera parameters – such as the angle of intersection of the optical axes – by the quality of image point perception (2×2 error parameters), and the quality of the calibration result of the 2×10 camera parameters. Because of this multi dimensional variety it is impossible to base the decision which algorithm is most suitable on a comprehensive error analysis. Instead it must be done with the help of realistic parameters of real world applications, where the parameters are slightly altered and the effects on the reconstruction quality are measured. This can be done by error propagation analysis, as has been done in [10], or by means of simulations. For our investigations we have chosen the latter method.

Section 4.1 depicts a subset of the simulations performed [11] to derive the results presented in section 4.2 and verified by experiments in section 5.

Section 6 discusses the computational requirements of the algorithms and the performance on a space qualified floating point DSP. A short summary is given in section 7.

2 CAMERA MODEL

The pin hole camera model we use does not contain distortion parameters. If needed, distortion parameters can be identified and measured image points corrected according to the distortion model. In this way the described forward intersection algorithms for the pin hole camera model can be used even if distortion needs to be taken into consideration [7].

Given a 3D point $A = (X, Y, Z)^T$ in a world coordinate system (WCS) the metric coordinates (x_{a1}, y_{a1})

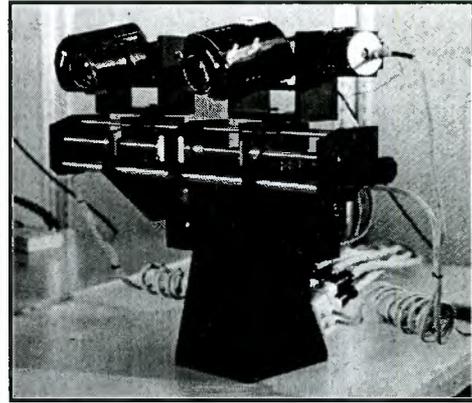


Figure 1: The KTH Robot Head.

and (x_{a2}, y_{a2}) of its two projections measured in pixels is obtained with the help of the so called collinearity equations (1, 2) and an equation that allows the transformation of the metric coordinates $(x_a, y_a)^T$ into pixel coordinates $(x_p, y_p)^T$.

$$x_a = -c \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (1)$$

$$y_a = -c \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \quad (2)$$

$$x_p = x_a / s_x + x_0 \quad (3)$$

$$y_p = y_a / s_y + y_0 \quad (4)$$

The parameter c is the distance from the optical centre of the lens to the projection plane of the camera and s_x, s_y are the lengths of the pixels in the x and y directions. The coordinates of the principal point are (x_0, y_0) . These intrinsic parameters do not change, if the cameras are moved and can be identified by calibration¹ [7, 8]. The other (extrinsic) parameters determine the translation (X_0, Y_0, Z_0) and the rotation to the WCS. The values r_{ij} are the matrix components of a rotation matrix that depends on three rotation parameters of the camera [5].

The extrinsic parameters can be identified by calibration but vary with camera movement. If the cameras are mounted on a robot head, see fig. 1, the kinematics of the robot head can be included in a more comprehensive model. Such a model reveals the extrinsic parameters with respect to the commanded poses of the cameras of the robot head [9].

3 THE FOUR ALGORITHMS

This section presents the four algorithms that are subject to our investigation. They are presented in the order of their computational complexity.

¹The values of intrinsic parameters may vary if a zoom lens is used. In this case a lookup table can be established.

APP: The simplest algorithm expresses the X and Y coordinate of A out of eq. 1, 2 as follows [5].

$$X = X_0 + (Z - Z_0) \cdot \frac{r_{11}x_a + r_{12}y_a - r_{13}C}{r_{31}x_a + r_{32}y_a - r_{33}C} \quad (5)$$

$$Y = Y_0 + (Z - Z_0) \cdot \frac{r_{21}x_a + r_{22}y_a - r_{23}C}{r_{31}x_a + r_{32}y_a - r_{33}C} \quad (6)$$

These equations appear two times, once for the left and once for the right camera. From the two equations 5 we can express Z but from the two equations 6, as well. The mean value is taken and X and Y computed [5].

VECTOR: The computationally second cheapest algorithm reconstructs the beams of sight from the known camera parameters and the measured image points. These will not intersect – due to the imperfections mentioned. The object point A is assumed to lie in the middle of the shortest distance between the two beams [4], see fig. 2.

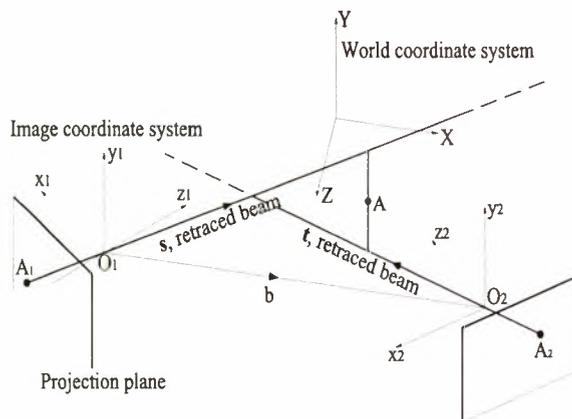


Figure 2: Point reconstruction with vector analysis.

LIN: Let's rewrite the 2×2 collinearity equations as shown for one camera [8, 11]

$$(r_{13}x_a + r_{11}) \cdot X + (r_{23}x_a + r_{21}) \cdot Y + (r_{33}x_a + r_{31}) \cdot Z = (r_{13}x_a + r_{11})X_0 + (r_{23}x_a + r_{21})Y_0 + (r_{33}x_a + r_{31})Z_0 \quad (7)$$

$$(r_{13}y_a + r_{12}) \cdot X + (r_{23}y_a + r_{22}) \cdot Y + (r_{33}y_a + r_{32}) \cdot Z = (r_{13}y_a + r_{12})X_0 + (r_{23}y_a + r_{22})Y_0 + (r_{33}y_a + r_{32})Z_0 \quad (8)$$

We interpret this as a linear system of equations in the variables X, Y and Z . The left hand side can be rewritten as a matrix S multiplied with the coordinates of A . The right hand side is a four dimensional column vector b . A least squares estimation for A is given by

$$A = (S^T S)^{-1} S^T b \quad (9)$$

NONLIN: Given a start estimation A_0 for the coordinates of A – based on one of the three algorithms described previously – we can compute the theoretical image points $(x_a^{mod}, y_a^{mod})^T$ in each camera by inserting

A_0 into equations 1 and 2. In general, these theoretical values will deviate by some few pixels from the measured image points (x_a^{meas}, y_a^{meas}) . We may tune the values for the coordinates of A_0 , obtaining a new guess A_1 such that the deviation

$$((x_a^{mod}, y_a^{mod})^T - (x_a^{meas}, y_a^{meas})^T)^2 \quad (10)$$

becomes smaller. This process is reiterated until a threshold is reached. If we take several points simultaneously, we can also include camera parameters in this adjustment and overcome imperfections of the calibration. The resulting minimisation problem is best solved with a Gauss-Newton method. Photogrammetric literature contains the Jacobian Matrix needed for a numerically efficient implementation in symbolic form [5]. The computational costs of the process described increase with the square of the number of points used.

In photogrammetric applications it is common to add control points to the fields of view of the cameras. Control points are points with known coordinates in the WCS. This is not relevant for space applications and therefore omitted in this article.

4 SYNTHETIC DATA EXPERIMENTS

4.1 Description of the Simulations

In the simulations the camera set-up always consists of two identical cameras in different poses. The working space is defined as the area which is covered by both camera views. Within this working space a cloud of random object points is generated. The projection of every single object point onto the projection planes of both cameras is computed with the model given in section 2. To model erroneously evaluated image point locations, random pixel noise is added. The evaluation of the four forward intersection procedures has been performed on a basis of 10000 random points per camera set-up and has been performed for different scales and applications [11]. This section presents results from a possible mobile robot application where we place the WCS into the object centre, for simplicity, without loss of generality.

Let's warm up with simulations where we consider a camera set-up of two cameras facing the origin of the WCS with a constant distance to the origin of the WCS. The cameras are moved along a circular course with its centre in the origin of the WCS. Their current pose is described by the angle between the cameras. For a mean absolute pixel noise set to 0.2 pixels and perfectly calibrated cameras (i.e. the exact intrinsic and extrinsic camera parameters are known) we notice similar deviations of the reconstruction results for all four algorithms. The accuracy of the reconstructions only differs in the range between 150° and 180° .

Large deviations are observed in the regions of very small and very large angles between the cameras (especially for APP), caused by the glancing intersection of the viewing beams at these angles. The *absolute deviations* reach their minimum at an angle of 90° between the cameras, of course.

We will now discuss more realistic scenarios: noisy data and imperfectly calibrated cameras. The camera set-up complies with a typical set-up of a mobile robot. Both cameras are at a distance of 4 m to the origin of the WCS, the angle between the cameras is 10° and the working space is constrained by a cylinder with a diameter of 3 m, see fig. 3. To estimate the influence

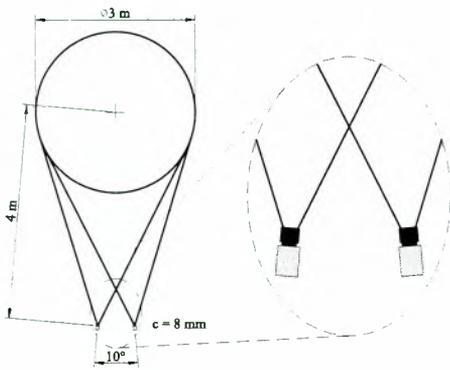


Figure 3: Camera set-up of a mobile robot.

of the calibration error of each of the three angles describing the rotation of the left camera, we set the mean absolute pixel noise to 0.5 pixels and vary the calibration error. Figures 4 to 6 demonstrate the simulation results. The unit of the ordinates are the *absolute deviations*, normalised to the range of the working space. Note the large influence of an erroneously calibrated vergence angle on the *absolute deviation* of the reconstruction. NONLIN may converge to a wrong minimum, because of the missing reference to the WCS.

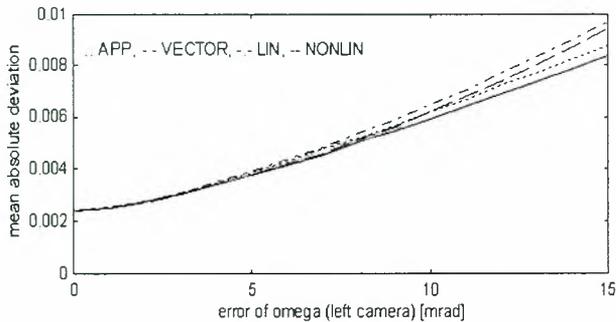


Figure 4: Imperfectly calibrated elevation.

Assuming that all 3 angles, that describe the rotation of one camera, are calibrated erroneously, we can

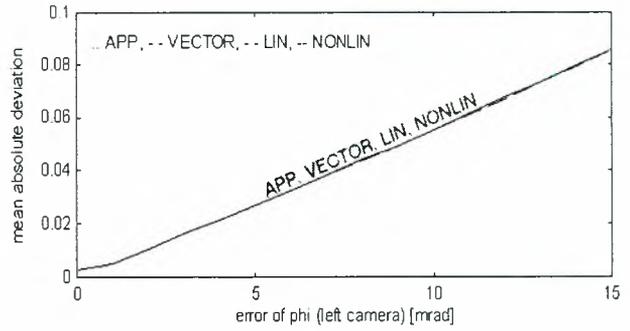


Figure 5: Imperfectly calibrated vergence.

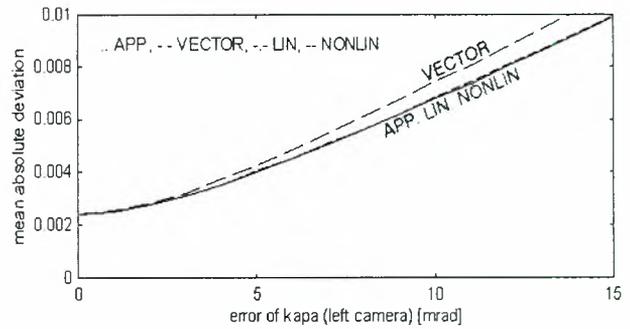


Figure 6: Imperfectly calibrated roll.

observe the *absolute deviations* of the reconstructions illustrated in fig. 7. A random noise of 0.5 pixels is added to the projection points. In the subsequent sim-

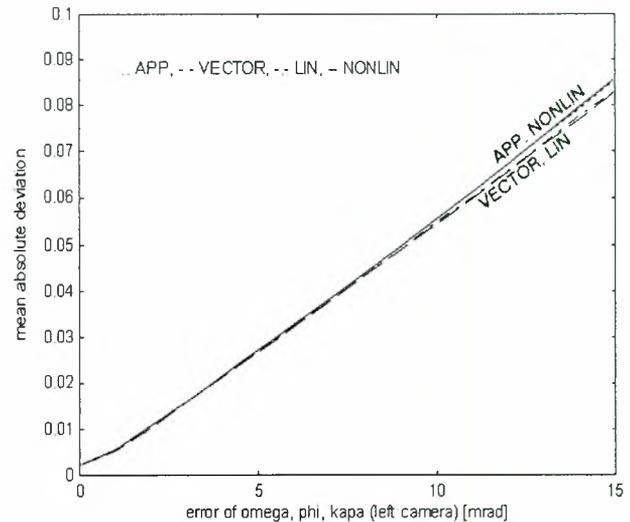


Figure 7: Imperfect calibration of one camera, *absolute deviation*.

ulation NONLIN regards the camera calibration parameters. That means the set of unknowns is extended with the erroneously calibrated angles of one camera. NONLIN uses 10 object points simultaneously in its minimisation procedure. Thus, the overdetermined system of equations consists of 40 equations derived from the 10 points to evaluate 33 unknowns (10×3

object coordinates, 3 angles). Fig. 8 presents the *consistency deviations* for the same set-up. We see that this is the domain of NONLIN, because it can correct the defective calibration parameters at the cost of computational complexity and the need for *several* points.

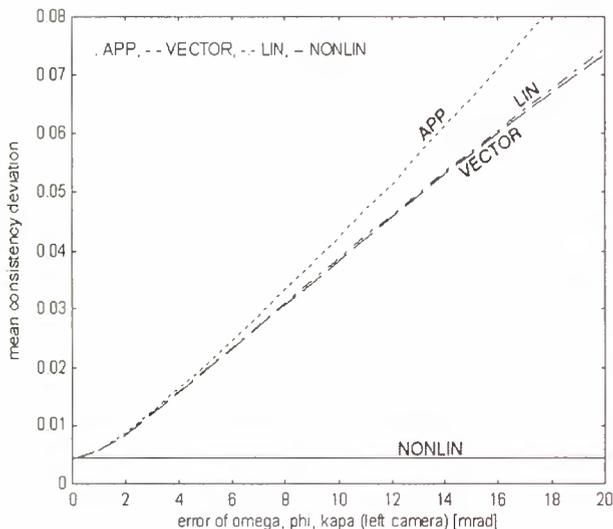


Figure 8: Imperfect calibration of one camera, *consistency deviation*.

4.2 Simulation Results

For the *absolute deviations* we can nominate a clear winner of the contest: VECTOR, followed closely by LIN. This was the case for all simulations we have performed. APP is third, NONLIN can not compete – it might converge to a wrong minimum, because its minimisation does not regard the WCS.

The *consistency deviations* are minimal when NONLIN is used. However, if the computational power is restricted VECTOR is the second best – with one exception. A badly calibrated roll error of one of the cameras can irritate VECTOR considerably and make LIN the winner. The roll of a camera is defined as a rotation about its optical axis. Robot heads mostly have a rigid construction that only allow pan and tilt movements, i.e. no rolling, and thus it is very unlikely that VECTOR is outperformed by LIN in most robot applications. The last – APP – lags more behind the others, when consistency deviations are regarded, compared to the results for absolute deviations.

5 REAL WORLD EXPERIMENTS

The implemented algorithms have been tested with real data provided by off-the-shelf cameras in a stereo set-up. Two JAI-235 industrial CCD cameras were mounted at a distance of 1000 mm to the object coordinate system, the angle between the cameras was fixed at 15° and thus the working space was constrained by

a cylinder with a diameter of 377 mm. In the working space an ellipsoid solid was situated, see fig. 9 for the images of the left and right camera. The object coordinates of marked points on the ellipsoid have been determined previously with the help of a high precision bundle adjustment together with extensive camera calibration and the image coordinates of the respective projections have been identified by ellipse fitting.

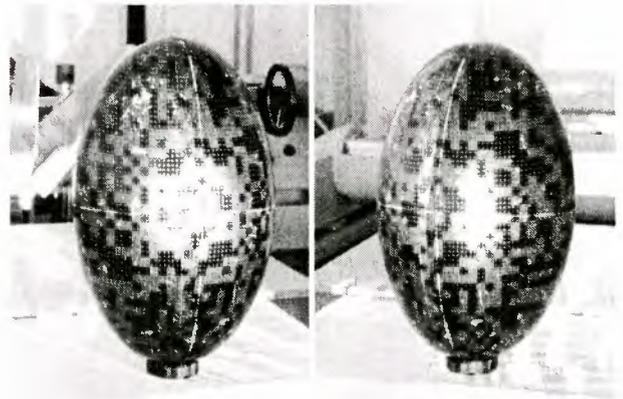


Figure 9: Images of the binocular camera set-up.

The following table lists the mean *absolute deviations* and the mean *consistency deviations* of the reconstructed object points normalised to the range of the working space. In this first approach NONLIN does not include calibration parameters in the minimisation procedure.

algorithm	deviation [10^{-3}]	
	absolute	consistency
APP	15.2	15.6
VECTOR	3.4	2.3
LIN	3.4	2.3
NONLIN	3.3	0.6

If NONLIN regards the camera calibration parameters of one camera and computes the object coordinates of 40 points simultaneously its mean *consistency deviation* reaches $0.2 \cdot 10^{-3}$, but note the quadratically increasing computational costs.

The results of these practical experiments underline the simulation results. Due to the comparatively large deviations of APP this algorithm is recommended as a start estimation, only.

6 DSP IMPLEMENTATION

The presented algorithms require the following number of floating point operations (FLOPS) for the reconstruction of one object point. The MATLAB im-

plementation used was without sophisticated optimisations and can be retrieved from [11].

algorithm	FLOP count
APP	70
VECTOR	130
LIN	363
NONLIN	> 1600

For the given comparison, the input to each algorithm comprises the two image points and the two camera coordinate systems given as homogeneous matrices. For a DSP Implementation the FLOP count is an inadequate measurement to estimate processing requirements. Thus we present the cycle count for each algorithm on a typical floating point signal processor.

The TSC21020E is a radiation hardened version of the ADSP 21020. It exhibits an enhanced Harvard Architecture that allows loading or storing data in the program memory and data memory simultaneously, if the instruction is found in an instruction cache. Since forward intersection will normally be performed for more than a single point we assume that the respective algorithm is performed in a loop and makes use of a 2×32 LRU instruction cache. Furthermore this processor can perform a multiplication simultaneously with an addition or subtraction.

For optimised implementations the following table shows the execution times in processor cycles and – for convenience – in micro seconds on a DSP operating at 20Mhz. The division is implemented for the full 32 bit floating point resolution and requires 8 cycles.

algorithm	cycles	time (μ s)
APP	56	2.8
VECTOR	152	7.6
LIN	213	10.65
NONLIN	> 800	> 40

As can be seen, the difference of computation costs between LIN and VECTOR gets smaller, but VECTOR remains the winner.

7 SUMMARY

In our paper we have addressed the problem of three dimensional point reconstruction given the projections of a point in two digital cameras with known extrinsic and intrinsic parameters. We have compared four different algorithms and have shown in which situation which algorithm is the best choice with respect to two different measures of precision. Finally we have given figures for processor loads for implementations. The

MATLAB implementations of the algorithms can be obtained in electronic form [11].

References

- [1] P. Wunsch, G. Hirzinger, "Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion". *Proc. Int. Conf. on Robotics and Automation*, Albuquerque, New Mexico - April 1997, pp. 2868 – 2873.
- [2] G. Paar, W. Poelzleitner, "Stereovision and 3d Terrain Modeling for Planetary Exploration". *1st ESA Workshop on Comp. Vision and Image Processing for Spaceborne Applications*, Noordwijk, NL, June 1991.
- [3] M. Ulm, G. Paar, "Relative Camera Calibration from Stereo Disparities". *Optical 3D-Measurement Techniques III* edited by Gruen/Kahmen, Wichmann, Heidelberg, 1995, pp. 526 – 533.
- [4] K.B. Atkinson (Ed), *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Caithness, 1996.
- [5] K. Kraus: *Photogrammetry*. Dümmler Verlag Bonn, 1993.
- [6] R.I. Hartley, "In defence of the 8-point algorithm". *IEEE Trans. PAMI*, vol. 19(1997), no. 6, pp. 580 – 593.
- [7] J. Weng, P. Cohen, M. Herniou: "Camera Calibration with Distortion Models and Accuracy Evaluation." *IEEE Trans. PAMI*, vol. 14 (1992), no. 10, pp. 965 – 980.
- [8] Mengxiang Li, "Camera Calibration of a Head-Eye System for Active Vision". *TR 147*, Computer Vision and Active Perception Lab, Royal Inst. of Technology (KTH), Stockholm, 1994.
- [9] S. Spiess, M. Li, "Kinematic calibration of an active binocular head for online computation of the epipolar geometry". *TR 205*, Computer Vision and Active Perception Lab, Royal Inst. of Technology (KTH), Stockholm, 1996. See <http://www.bion.kth.se>.
- [10] N. Georgis, M. Petrou, J. Kittler, "Error Guided Design of a 3D Vision System". *IEEE Trans. PAMI*, vol. 20 (1998), no. 4, pp. 366 – 379.
- [11] R. Krickl, *A comparison of four forward intersection methods for binocular computer vision*. Master Thesis, Inst. of Flexible Automation, Vienna Univ. of Technology, 1998. For electronic copies contact authors.

A TRAJECTORY AND FORCE CONTROL OF A MANIPULATOR WITH ELASTIC LINKS

Katsuyoshi Tsujita, Kazuo Tsuchiya and Yousuke Kawano

Dept. of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University

Yoshida-Hon-Machi, Sakyo-ku, Kyoto 606-01, Japan

E-mail : tsujita@space.kuaero.kyoto-u.ac.jp

Phone: +81-774-38-3961 Facsimile: +81-774-38-3962

ABSTRACT

When the velocities of the motion required for a manipulator become faster, and the values of forces acting on a subject are large, the elastic deformations of the manipulator cannot be ignored in the trajectory and force control. To develop a method of the trajectory and force control of a flexible manipulator is an important problem. In this paper, a trajectory and force control of a flexible manipulator based on inverse kinematics and inverse dynamics is proposed. First, inverse kinematics and inverse dynamics of a flexible manipulator are investigated in detail, and then, a hierarchical controller based on inverse kinematics and inverse dynamics is proposed. The performances of the proposed controller are verified by numerical simulations.

1. INTRODUCTION

A manipulator is a mechanical system whose links are connected through translational or rotational joints. One of the tasks for a manipulator is to control forces acting on a subject along to a given trajectory. In order to establish the task, a force and trajectory control is necessary. In the space engineering, this class of technology is required for assembly of space structures on an orbit or for release and retrieval of an artificial satellite.

When the velocities of the motion required for a manipulator become faster, and the values of forces acting on a subject are large, the elastic deformations of the manipulator cannot be ignored in the trajectory and force control. A manipulator whose links has elastic deformations which cannot be ignored is called a flexible manipulator, while a manipulator

composed of links whose elastic deformations can be ignored is called a rigid manipulator. It is necessary for a space manipulator to be considered as a flexible manipulator because the structural rigidity of the links becomes lower through lightening its weight. Then, to develop a method of the trajectory and force control of a flexible manipulator is an important problem in order to use a space manipulator well.

In this paper, a trajectory and force control of a flexible manipulator is proposed. Generally, one of the basic methods of motion control of a manipulator is as follows; First, by using control inputs, the state equations are linearized. And then, based on the linearized state equations, linear feedback control is adopted. A method in which the state equations are linearized by compensating the nonlinear terms with the use of the measured values of the state variables is called feedback linearization method. In this method, a feedback control is executed based on the linearized state equations. On the other hand, a method in which the state equations are linearized by inputting the control force which realizes the desired motion is called feedforward linearization method. In this method, the linearized state equations are derived around the desired values of the state variables, then a feedback control is executed based on the linearized state equations.

In the feedforward linearization method, inverse kinematics and inverse dynamics are important. When the desired motion and the desired force acting on the surface of an object through the manipulator are given, inverse kinematics is to calculate all the state variables according to the motion and the force are

calculated, and inverse dynamics is to calculate the desired input forces or torques to realize the motion derived through inverse kinematics.

For a rigid manipulator, inverse kinematics is derived using kinematic relations between the state variables and inverse dynamics is derived by using the state equations. And then, the input forces or torques can be realized by the actuators of the manipulator.

But, there are some difficulties in inverse kinematics and inverse dynamics of a flexible manipulator; Inverse kinematics cannot be derived by only the kinematic relations between the state variables. The input forces or torques derived by inverse dynamics cannot be realized by the actuators of the manipulator either.

In this paper, a trajectory and force control of a flexible manipulator based on the feedforward linearization method. Inverse kinematics and inverse dynamics of a flexible manipulator are investigated in detail, and a hierarchical controller based on inverse kinematics and inverse dynamics is proposed.

This paper is composed as follows; First, in section 2, the model of the manipulator system dealt with in this paper is mentioned and the equations of motion are derived. In section 3, the methods of inverse kinematics and inverse dynamics are derived. In section 4, the method of design of the controller is mentioned and in section 5, the performances of the proposed controller are verified by numerical simulations.

2. FORWARD MODEL

Consider a manipulator composed of two bodies, body 1 and body 2 (FIG. 1). Body 1 is put on a base with a rotary joint (joint 1) and body 2 is connected to body 1 with a rotary joint (joint 2). Motors are installed at the rotary joints, the axes of which are perpendicular to a vertical plane. Body 1 is a rigid rod and body 2 is an elastic beam, elastic deformations of which occur in a plane perpendicular to the axis of rotation. Introduce a set of unit vectors $\{\mathbf{a}^{(0)}\} = \{\mathbf{a}_1^{(0)}, \mathbf{a}_2^{(0)}, \mathbf{a}_3^{(0)}\}$ fixed in an inertia space, the origin of which coincides with joint 1. Vector $\mathbf{a}_3^{(0)}$ coincides with the axis of rotation and vector $\mathbf{a}_2^{(0)}$ is set downward. A set of unit vectors $\{\mathbf{a}^{(i)}\} = \{\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \mathbf{a}_3^{(i)}\}$ is introduced, the origin of which coincides with joint i . Vector $\mathbf{a}_3^{(i)}$ coincides

with the axis of rotation of joint i and vector $\mathbf{a}_1^{(i)}$ is set toward the axis of body i . Using a set of unit vectors $\{\mathbf{a}^{(i)}\}$, a column matrix is introduced,

$$[\mathbf{a}^{(i)}]^T = [\mathbf{a}_1^{(i)}, \mathbf{a}_2^{(i)}, \mathbf{a}_3^{(i)}] \quad (1)$$

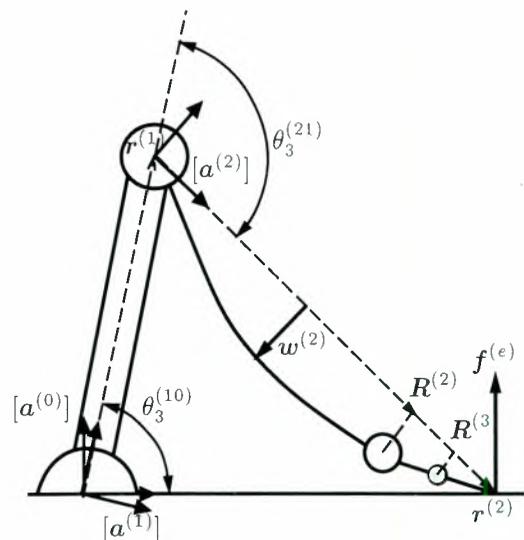


FIG. 1 Two body manipulator system

By introducing the angles of rotation from $\{\mathbf{a}^{(j)}\}$ to $\{\mathbf{a}^{(i)}\}$ about $\mathbf{a}_3^{(j)}$ axis as $\theta_3^{(ij)}$, transformation matrices from $\{\mathbf{a}^{(j)}\}$ to $\{\mathbf{a}^{(i)}\}$ are defined by $A^{(ij)}$.

The angular velocity vector of $\{\mathbf{a}^{(i)}\}$ to $\{\mathbf{a}^{(j)}\}$ is defined by $\omega^{(ij)}$

$$\begin{aligned} \omega^{(ij)} &= [\mathbf{a}^{(i)}]^T \omega^{(ij)} \\ \omega^{(ij)T} &= [0, 0, \dot{\theta}_3^{(ij)}] \end{aligned} \quad (2)$$

The following quantities are introduced,

$\mathbf{r}^{(1)} = [\mathbf{a}^{(1)}]^T r^{(1)}$; a distance vector from joint 1 to joint 2.

$\mathbf{r}^{(2)} = [\mathbf{a}^{(2)}]^T r^{(2)}$; a distance vector from joint 2 to the end effector.

$\rho^{(i)} = [\mathbf{a}^{(i)}]^T \rho^{(i)}$; a distance vector from joint i to any position in body i .

The elastic deformation of body 2 is denoted by $\mathbf{w}^{(2)}$

$$\begin{aligned} \mathbf{w}^{(2)} &= [\mathbf{a}^{(2)}]^T w^{(2)} \\ \mathbf{w}^{(2)T} &= [0, w_2^{(2)}(t, \rho_1^{(2)}), 0] \end{aligned} \quad (3)$$

By using the finite element method, the elastic deformation $w_2^{(2)}(t, \rho_1^{(2)})$ is expressed as

(Appendix)

$$w_2^{(2)}(t, \rho_1^{(2)}) = B^{(2)}(\rho_1^{(2)}) \hat{w}_2^{(2)}(t) \quad (4)$$

A distance vector $\mathbf{x}^{(1)}$ from joint 1 to any point in body 1 is expressed as

$$\begin{aligned}\mathbf{x}^{(1)} &= [\mathbf{a}^{(1)}]^T x^{(1)} \\ x^{(1)} &= \rho^{(1)}\end{aligned}$$

and the velocity vector $\mathbf{v}^{(1)}$ is expressed as

$$\begin{aligned}\mathbf{v}^{(1)} &= [\mathbf{a}^{(1)}]^T v^{(1)} \\ v^{(1)} &= \tilde{\rho}^{(1)} \omega^{(10)}\end{aligned}\quad (5)$$

where, $\tilde{\rho}^{(1)T}$ makes a cross product in $[\mathbf{a}^{(1)}]$.

On the other hand, a distance vector $\mathbf{x}^{(2)}$ from joint 1 to any point in body 2 is expressed as

$$\begin{aligned}\mathbf{x}^{(2)} &= [\mathbf{a}^{(2)}]^T x^{(2)} \\ x^{(2)} &= A^{(21)} r^{(1)} + (\rho^{(2)} + w^{(2)})\end{aligned}$$

and the velocity vector $\mathbf{v}^{(2)}$ is expressed as

$$\begin{aligned}\mathbf{v}^{(2)} &= [\mathbf{a}^{(2)}]^T v^{(2)} \\ v^{(2)} &= A^{(21)} \tilde{\rho}^{(1)} \omega^{(10)} \\ &\quad + \{\tilde{\rho}^{(2)} \omega^{(20)} + B^{(2)} (\rho_1^{(2)}) \dot{\hat{w}}^{(2)}(t)\}\end{aligned}\quad (6)$$

A state variables z of the system are set to be

$$z^T = [\theta_3^{(10)}, \theta_3^{(21)}, \hat{w}_2^{(2)T}] \quad (7)$$

The equation of motion for stable variables z are derived as follows; The equations for variables $\theta_3^{(10)}$ and $\theta_3^{(21)}$ are derived from the equations of the angular momenta of body 1 and body 2 about joint 1 and joint 2, respectively.

$$\begin{aligned}\frac{d}{dt} \langle \tilde{\rho}_c^{(1)T} v^{(1)} \rangle^{(1)} + \langle \tilde{\rho}_c^{(1)T} \tilde{\omega}^{(10)T} v^{(1)} \rangle^{(1)} \\ = -\tilde{r}_c^{(1)T} f^{(1)} - (\tilde{r}^{(1)T} - \tilde{r}_c^{(1)T}) A^{(12)} f^{(2)} \\ + \tau^{(1)} - A^{(12)} \tau^{(2)}\end{aligned}\quad (8)$$

$$\begin{aligned}\frac{d}{dt} \langle \tilde{\rho}_c^{(2)T} v^{(2)} \rangle^{(2)} + \langle \tilde{\rho}_c^{(2)T} \tilde{\omega}^{(20)T} v^{(2)} \rangle^{(2)} \\ = -\tilde{r}_c^{(2)T} f^{(2)} - (\tilde{r}^{(2)T} - \tilde{r}_c^{(2)T}) A^{(20)} f^{(e)} + \tau^{(2)}\end{aligned}\quad (9)$$

where,

$$\begin{aligned}\rho_c^{(i)} &= \rho^{(i)} - r_c^{(i)}, \quad r_c^{(i)} = \frac{1}{m^{(i)}} \int \rho^{(i)} dm^{(i)} \\ \langle * \rangle^{(i)} &= \int * dm^{(i)}\end{aligned}$$

$m^{(i)}$ is mass of body i , $\mathbf{f}^{(i)} = [\mathbf{a}^{(i)}]^T f^{(i)}$ and $\boldsymbol{\tau}^{(i)} = [\mathbf{a}^{(i)}]^T \tau^{(i)}$ are a force and a torque acting on body i at joint i , and $\mathbf{f}^{(e)} = [\mathbf{a}^{(0)}]^T f^{(e)}$ is a force acting on the surface of the object through the end effector.

The forces $f^{(1)}$ and $f^{(2)}$ are expressed as

$$\begin{aligned}f^{(1)} &= \frac{d}{dt} \langle v^{(1)} \rangle^{(1)} + \tilde{\omega}^{(10)T} \langle v^{(1)} \rangle^{(1)} \\ &\quad - m^{(1)} A^{(10)} g + A^{(12)} f^{(2)} \\ f^{(2)} &= \frac{d}{dt} \langle v^{(2)} \rangle^{(2)} + \tilde{\omega}^{(20)T} \langle v^{(2)} \rangle^{(2)} \\ &\quad - m^{(2)} A^{(20)} g + A^{(20)} f^{(e)}\end{aligned}\quad (10)$$

where g is a gravitational constant. The equations of motion for variables $\hat{w}_2^{(2)}$ are derived from the equations of elastic vibrations of body 2

$$\begin{aligned}\frac{d}{dt} \langle B^{(2)T} v^{(2)} \rangle^{(2)} + \langle B^{(2)T} \tilde{\omega}^{(20)T} v^{(2)} \rangle^{(2)} \\ = \langle B^{(2)T} \rangle^{(2)} A^{(20)} g - K^{(2)} \hat{w}^{(2)} - D^{(2)} \dot{\hat{w}}^{(2)} \\ - B^{(2)T} (r^{(2)}) A^{(20)} f^{(e)} + E^{(2)T} \tau_3^{(2)}\end{aligned}\quad (11)$$

where, the second and third terms in the right hand side of Eq. (11) express an elastic restoring force and a structural damping force, respectively.

$$E^{(2)} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

3. INVERSE MODEL

It is assumed that a desired trajectory $\mathbf{x}_d^{(e)}(t) = [\mathbf{a}^{(0)}]^T x_d^{(e)}$ of the end effector and a desired force $\mathbf{f}_d^{(e)}(t) = [\mathbf{a}^{(0)}]^T f_d^{(e)}$ acting on the surface of an object through the end effector are given. Inverse kinematics is to calculate the angles of rotation $\theta_{d3}^{(10)}(t)$ and $\theta_{d3}^{(21)}(t)$ and the elastic deformations $\hat{w}_{d2}^{(2)}(t)$ corresponding to the desired trajectory and the desired force. First, the distance vector $\mathbf{x}^{(e)}$ of the end effector from joint 1 is expressed as

$$\begin{aligned}\mathbf{x}^{(e)} &= [\mathbf{a}^{(0)}]^T x^{(e)} \\ x^{(e)} &= A^{(01)} r^{(1)} + A^{(02)} r^{(2)}\end{aligned}\quad (12)$$

Substituting the desired trajectory $x_d^{(e)}$ into Eq. (12), we obtain the equation to determine the angles of rotation $\theta_{d3}^{(10)}, \theta_{d3}^{(21)}$.

$$\begin{aligned}x_{d1}^{(e)} &= r^{(1)} \cos \theta_{d3}^{(10)} + r^{(2)} \cos \theta_{d3}^{(20)} \\ x_{d2}^{(e)} &= r^{(1)} \sin \theta_{d3}^{(10)} + r^{(2)} \sin \theta_{d3}^{(20)}\end{aligned}\quad (13)$$

Next, substituting the desired force $f_d^{(e)}$ and the angles of rotation $\theta_{d3}^{(10)}, \theta_{d3}^{(21)}$ into Eq. (11), we obtain the equation to determine the elastic deformations

$\widehat{w}_{d2}^{(2)}(t)$.

$$\begin{aligned}
& M^{(2)} \ddot{\widehat{w}}_{d2} + D^{(2)} \dot{\widehat{w}}_{d2}^{(2)} + \left\{ K^{(2)} - \omega_{d3}^{(20)2} M^{(2)} \right\} \widehat{w}_{d2}^{(2)} \\
&= \left(-g_2 \cos \theta_{d3}^{(20)} - r_1^{(1)} \omega_{d3}^{(10)2} \sin \theta_{d3}^{(21)} \right. \\
&\quad \left. - r_1^{(1)} \dot{\omega}_{d3}^{(10)} \cos \theta_{d3}^{(21)} \right) < B_2^{(2)T} >^{(2)} \\
&\quad - \dot{\omega}_{d3}^{(20)} < B_2^{(2)T} \rho_1^{(2)} >^{(2)} \\
&\quad + \left\{ (J^{(2)} + m^{(2)} R^{(2)2}) \dot{\omega}_{d3}^{(20)} \right. \\
&\quad \quad + m^{(2)} r^{(1)} R^{(2)} \cos \theta_{d3}^{(21)} \dot{\omega}_{d3}^{(10)} \\
&\quad \quad m^{(2)} r^{(1)} R^{(2)} \sin \theta_{d3}^{(21)} \omega_{d3}^{(10)2} \\
&\quad \quad + r^{(2)} (-f_1^{(e)} \sin \theta_{d3}^{(20)} + f_2^{(e)} \cos \theta_{d3}^{(20)}) \\
&\quad \quad \left. + m^{(2)} R^{(2)} g_2 \cos \theta_{d3}^{(20)} \right\} E^{(2)T} \quad (14)
\end{aligned}$$

$$M^{(2)} = < B_2^{(2)T} B_2^{(2)} >^{(2)} - < E^{(2)T} C^{(2)} >^{(2)}$$

where, $C^{(2)} = [< \rho_1^{(2)} B_2^{(2)} >^{(2)}, 0, \dots, 0]$.

Equation (14) is a set of second order ordinary differential equations and are appropriate to be formulated as initial value problems with the initial conditions

$$t = 0, \quad \widehat{w}_2^{(2)} = \dot{\widehat{w}}_2^{(2)} = 0$$

However, since the coefficient matrix of $\ddot{\widehat{w}}_{d2}^{(2)}$ is not positive definite, Eq. (14) is not well posed as an initial value problem. Here, Eq. (14) is formulated as boundary value problem with the boundary conditions

$$\begin{aligned}
t = 0 \quad \widehat{w}_2^{(2)} &= \alpha \\
t = t_f \quad \widehat{w}_2^{(2)} &= \beta
\end{aligned} \quad (15)$$

where, t_f is a time interval of manipulation, and α, β are values of elastic deformations in a steady state. As boundary value problems, we can obtain stable solutions numerically, but the elastic deformations $\widehat{w}_{d2}^{(2)}$ obtained have certain velocities at the beginning of manipulation.

On the other hand, when the angle of rotation $\theta_{d3}^{(10)}(t)$ and $\theta_{d3}^{(21)}(t)$ and elastic deformations $\widehat{w}_{d2}^{(2)}(t)$ are obtained, inverse dynamics is to calculate the torques $\tau_d^{(1)}$ and $\tau_d^{(2)}$ which realize the desired motions. First, the forces $f_d^{(1)}(t)$ and $f_d^{(2)}(t)$ are calculated by Eq. (10) with the variables $\theta_{d3}^{(10)}, \theta_{d3}^{(21)}$ and $\widehat{w}_{d2}^{(2)}$.

$$\begin{aligned}
f_d^{(2)} &= \frac{d}{dt} \left\langle v_d^{(2)} \right\rangle^{(2)} + \tilde{\omega}_d^{(20)T} \left\langle v_d^{(2)} \right\rangle^{(2)} \\
&\quad - m^{(2)} A^{(20)} g + A^{(20)} f_d^{(e)}
\end{aligned} \quad (16)$$

$$\begin{aligned}
f_d^{(1)} &= \frac{d}{dt} \left\langle v_d^{(1)} \right\rangle^{(1)} + \tilde{\omega}_d^{(20)T} \left\langle v_d^{(1)} \right\rangle^{(1)} \\
&\quad - m^{(1)} A^{(10)} g + A^{(12)} f_d^{(2)}
\end{aligned}$$

The torques $\tau_d^{(1)}(t)$ and $\tau_d^{(2)}(t)$ are calculated by using Eqs. (8),(9),(11). But, when we calculate $\tau_d^{(1)}(t)$ and $\tau_d^{(2)}(t)$ from Eqs. (8),(9),(11), it is inconsistent. In this paper, the least square solutions are used for $\tau_d^{(1)}(t), \tau_d^{(2)}(t)$.

$$\begin{aligned}
\tau_d^{(2)} &= \frac{d}{dt} \left\langle \tilde{\rho}_c^{(2)T} v_c^{(2)} \right\rangle^{(2)} + \left\langle \tilde{\rho}_c^{(2)T} \tilde{\omega}_d^{(20)T} v_d^{(2)} \right\rangle^{(2)} \\
&\quad + \tilde{r}_c^{(2)T} f_d^{(2)} + \left(\tilde{r}^{(2)T} - \tilde{r}_c^{(2)T} \right) A^{(20)} f_d^{(e)} \\
\tau_d^{(1)} &= \frac{d}{dt} \left\langle \tilde{\rho}_c^{(1)T} v_c^{(1)} \right\rangle^{(1)} + \left\langle \tilde{\rho}_c^{(1)T} \tilde{\omega}_d^{(10)T} v_d^{(1)} \right\rangle^{(1)} \\
&\quad + \tilde{r}_c^{(1)T} f_d^{(1)} + \left(\tilde{r}^{(1)T} - \tilde{r}_c^{(1)T} \right) A^{(12)} f_d^{(2)} \\
&\quad + A^{(12)} \tau_d^{(2)}
\end{aligned} \quad (17)$$

4. DESIGN OF A CONTROLLER

Consider a trajectory and force control of a manipulator. A trajectory and force control is a manipulation for the end effector of the manipulator to track a certain trajectory, that is, to follow a surface of an object and also to act a desired pushing force on the surface of it during the manipulation.

The proposed controller is composed of two parts; Feedforward and feedback terms. The feedforward term is to compensate nonlinearity of the system dynamics and is calculated based on inverse kinematics and inverse dynamics; Equation (17) is used.

Equation (17) includes some errors due to two factors. One factor is that Eq. (17) includes impulsive forces at the moments of the beginning and the end of manipulation because $\dot{\widehat{w}}_2^{(2)} \neq 0$ as the solution of Eq. (14). But when Eq. (17) is applied to the feedforward controller, the components of impulsive torques are neglected. Therefore, the motion of the manipulator have some errors at the beginning and the end of manipulation.

On the other hand, Eq. (17) is the least square solution, therefore it is not the exact solution. The errors of the solution cause some vibration modes and degrade the performances of the controller.

The feedback term compensates the errors included in the feedforward term with the model errors due to neglect of vibration modes and that caused by disturbances. A flexible manipulator become often to be a non minimum phase system. As a result, the feedback loop causes the system unstable. To pre-

vent it and to insure the robustness of the controller, a direct feedback controller is an effective one. In this paper, the feedback term of the proposed controller is designed as the direct feedback one.

The input commands to the motors at the joints are designed as follows:

$$\tau_c^{(1)} = \tau_{d3}^{(1)} \quad (18)$$

$$+s_1 \left\{ K_{DF} \dot{e}_f + K_{PF} e_f + K_{IF} \int e_f dt \right\} \quad (19)$$

$$\tau_c^{(2)} = \tau_{d3}^{(2)} - K_w \dot{\hat{w}}_{2,0}$$

$$+s_2 \left\{ K_{DF} \dot{e}_f + K_{PF} e_f + K_{IF} \int e_f dt \right\} \quad (20)$$

$$e_f = f_{2d}^{(e)} - f_2^{(e)}$$

$$s_1 = r^{(1)} \cos \theta_3^{(10)} + r^{(2)} \cos \theta_3^{(20)}$$

$$s_2 = r^{(1)} \cos \theta_3^{(10)}$$

Feedback gains are determined as follows; First, consider deviations of variables $f_2^{(e)}$, $\hat{w}_{2,n}$, $\dot{\hat{w}}_{2,n}$, and the derivatives from the nominal values.

$$\begin{aligned} \hat{w}_{2,n} &= \hat{w}_{2,n}^* + \Delta \hat{w}_{2,n} \\ \dot{\hat{w}}_{2,n} &= \dot{\hat{w}}_{2,n}^* + \Delta \dot{\hat{w}}_{2,n} \\ \ddot{\hat{w}}_{2,n} &= \ddot{\hat{w}}_{2,n}^* + \Delta \ddot{\hat{w}}_{2,n} \\ e_f &= f_{2d}^{(e)} - f_2^{(e)} \end{aligned} \quad (21)$$

Following vector is defined.

$$X = \left[\Delta \hat{w}^{(2)T} \quad \Delta \dot{\hat{w}}^{(2)T} \quad e_f \quad \int e_f dt \right]^T \quad (22)$$

Substituting Eq. (21) into Eqs. (8), (9), (11), and linearizing them following equations are obtained.

$$H_1 \dot{X} + H_2 X = 0 \quad (23)$$

where,

$$H_1 = \begin{bmatrix} \langle B^{(2)T} B^{(2)} \rangle^{(2)} & D^{(2)} & aE^{(2)T} & bE^{(2)T} \\ O & I & 0 & 0 \\ R & S & a & b \\ \langle \rho_1^{(2)} B^{(2)} \rangle^{(2)} & 0 & e & f \end{bmatrix}$$

$$H_2 = \begin{bmatrix} O & K^{(2)} & 0 & cE^{(2)T} \\ -I & O & 0 & 0 \\ 0 & d & 0 & c \\ 0 & \omega_3^{(20)} \langle \rho_1^{(2)} B^{(2)} \rangle^{(2)} & 0 & g \end{bmatrix}$$

$$a = -s_1 K_{DF}, \quad b = -s_1 (1 + K_{PF})$$

$$c = -s_2 K_{IF}$$

$$d = - \left[r^{(1)} \sin \theta_3^{(21)} \dot{\omega}_3^{(20)} \langle B \rangle^{(2)} \right]$$

$$+ r^{(1)} \omega_3^{(20)2} \langle \rho_1^{(2)} B \rangle^{(2)} \Big]$$

$$e = -s_2 K_{DF}, \quad f = -s_2 (1 + K_{PF})$$

$$g = -s_2 K_{IF}$$

$$R = \langle \rho_1^{(2)} B \rangle^{(2)} + r^{(1)} \cos \theta^{(21)} \langle B^{(2)} \rangle^{(2)}$$

$$S = -2r^{(1)} \omega^{(20)} \sin \theta^{(21)} \langle B^{(2)} \rangle^{(2)}$$

$$D^{(2)} = D + K_w E^{(2)T} E^{(2)}$$

$$K^{(2)} = K^{(2)} - \omega_3^{(20)2} \langle B^{(2)T} B^{(2)} \rangle^{(2)}$$

Based on Eq. (23), feedback gains K_{DF} , K_{PF} , K_{IF} and K_w are determined appropriately.

5. NUMERICAL SIMULATION

Here, the controller proposed in section 4 are verified numerically; The desired trajectory of the end effector $x_d^{(e)}$ and the desired force $f_d^{(e)}$ acting on the surface of the object are given and the torques $\tau_c^{(1)}$, $\tau_c^{(2)}$ which realize the desired motion are calculated on the basis of the inverse models proposed. Then, the equations of motion of the manipulator are solved numerically where the torques obtained are used as the input torques, and the force $f^{(e)}$ acting on the surface are compared with the desired force acting on the surface. The values of parameters of the manipulator are listed in TABLE 1.

TABLE 1

	Link 1	Link 2
Length [m]	0.500	0.550
Mass [kg]	8.00	0.240
Bending Stiffness[Nm ²]	-	0.480
Damping Ratio	-	5.00 E-02
Natural Frequency [Hz]		
1st mode	-	4.8004
2nd mode	-	19.8850
3rd mode	-	49.0634
4th mode	-	102.8703
5th mode	-	174.6451
6th mode	-	271.9208

Body 2 is modeled as four finite elements for the inverse models ($N = 4$). The desired trajectory of the end effector and the desired force acting on the surface are given as follows,

$$x_{d2}^{(e)} = 0.05 \text{ [m]}$$

$$f_{d2}^{(e)} = 2.0 \text{ [N]}$$

$$x_{d1}^{(e)} = 0.7 - 0.5(-252\hat{t}^{11} + 1386\hat{t}^{10} - 3080\hat{t}^9 + 3465\hat{t}^8 - 1980\hat{t}^7 + 462\hat{t}^6) \text{ [m]}$$

where, $\hat{t} = t/t_f, t_f = 2.0$ [sec].

Figure 2 shows stick diagram of the manipulator. Figures 3 ~ 6 show the forces acting on the surface of the object through the end effector.

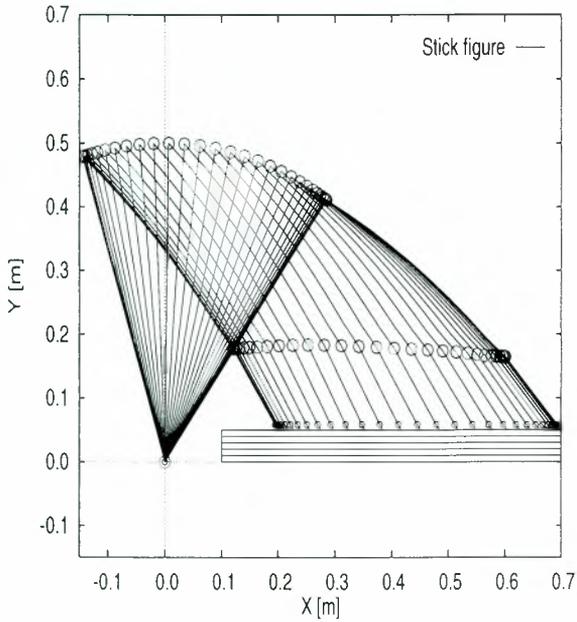


FIG. 2 Stick figure of the manipulator

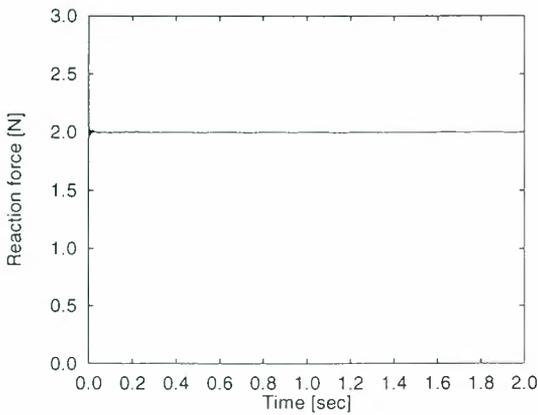


FIG. 3 The force $f^{(e)}$ acting on the surface (with initial elastic deformation velocity, feedforward only)

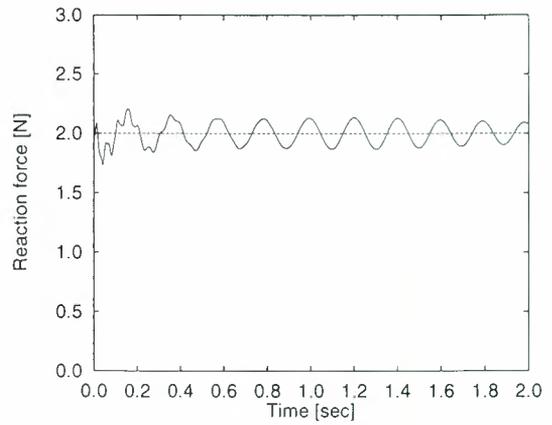


FIG. 4 The force $f^{(e)}$ acting on the surface (without initial elastic deformation velocity, feedforward only)

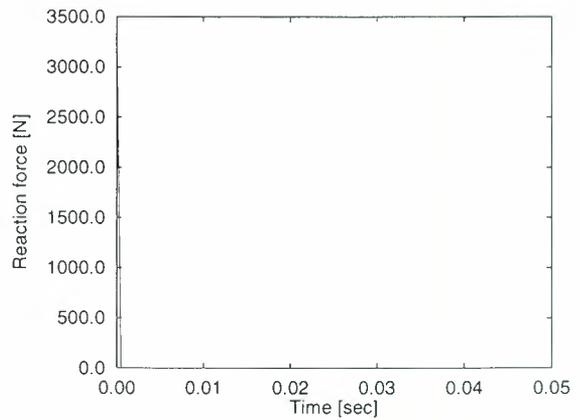


FIG. 5 The force $f^{(e)}$ acting on the surface (without initial elastic deformation velocity, feedback only)

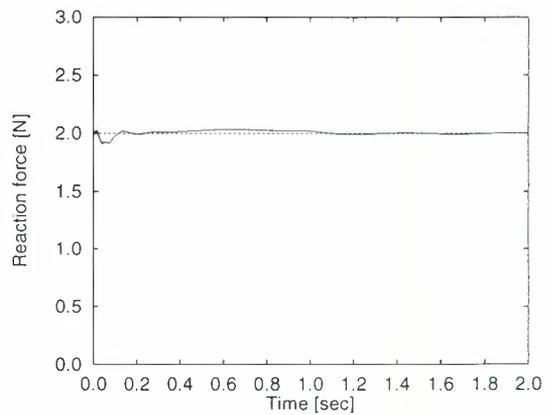


FIG. 6 The force $f^{(e)}$ acting on the surface (without initial elastic deformation velocity, proposed controller)

Figure 3 indicates the feedforward torque can control the reaction force accurately if the initial elastic deformation velocities are given. But from figure 4, we can find that if the initial elastic deformation velocities are not given(it has more reality),

the feed forward controller only causes some vibrations at the beginning of the manipulation. On the other hand, from figure 5, we can find that feedback controller only cannot control motion of the manipulator. Finally, from figure 6, the proposed controller can suppress the excitation of vibration and has a good performance for a trajectory and force control of a manipulator.

6. CONCLUSION

In order to establish a trajectory and force control of a manipulator, the controllers have to generate input torque commands to the motors at the joints for the end effector to realize the desired trajectory and pushing force. In such cases, we have to consider three difficulties to design the controller.

The first one is to deal with the nonlinearity of dynamics of the manipulator. The second one is excitation of vibration in the transition period. The last difficulty is to cancel the disturbance during the manipulation and to control the force acting on the object. To deal with these difficulties, we proposed a hybrid controller composed of feedforward and feedback controllers. Feedforward commands are generated by inverse kinematics and inverse dynamics. The performances of the proposed controller are verified by numerical simulations.

REFERENCES

- [1] Asada, H., Z.-D. Ma, and H. Tokumaru, "Inverse Dynamics of Flexible Robot Arms: Modeling and Computation for Trajectory Control," *ASME Journal of Dynamic Systems, Measurement, and Control*, 112, pp.177-185, 1990
- [2] Bayo, E., P.Papadopoulos, J.Stubbe, and M.A.Serna, "Inverse Dynamics and Kinematics of Multi-link Elastic Robots: An Iterative Frequency Domain Approach," *International Journal of Robotics Research*, 8(6), pp.49-62, 1989
- [3] Dai, Y.Q.,K.Usui, and M.Uchiyama, "A New Iterative Approach Solving Inverse Flexible-Link Manipulator Kinematics," *Proceedings of the 35th IEEE Conference on Decision and Control*, pp.2493-2494, 1996
- [4] R.E.Goddard, Y.F.Zheng and H.Hemami, "Dynamic Hybrid Velocity/Force Control of Robot Compliant Motion over Globally Unknown Objects," *IEEE Trans. on Robotics and Automation*, Vol.8, No.1, pp. 132-138, 1992
- [5] T.Yoshikawa and A.Sudou, "Dynamic Hybrid Position/Force Control of Robot Manipulators -On-Line Estimation of Unknown Constraint," *IEEE Trans. on Robotics and Automation*, Vol.9, No.2, pp.220-227, 1993
- [6] D.Wang and N.H.McClamroch, "Position and Force Control for Constrained Manipulator Motion: Lyapunov's Direct Method," *IEEE Trans. on Robotics and Automation*, Vol.9, No.3, pp.308-313, 1993
- [7] D.Jeon and M.Tomizuka, "Learning Hybrid Force and Position Control of Robot Manipulators," *IEEE Trans. on Robotics and Automation*, Vol.9, No.4, pp.423-431, 1993

A Appendix

Body 2 is divided into N finite elements where are numbered as $1, 2, \dots, N$ from joint 2 to the end effector. The nodes are also numbered as $0, 1, \dots, N$ from joint 2 to the end effector. An elastic deformation in element $n, w_{2,n}^{(2)}$ is expressed as

Element 1

$$w_{2,1}^{(2)}(t, \rho_1^{(2)}) = \begin{bmatrix} x_n - \frac{2}{l}x_n^2 + \frac{1}{l^2}x_n^3, \\ \frac{3}{l^2}x_n^2 - \frac{2}{l^3}x_n^3, -\frac{1}{l}x_n^2 + \frac{1}{l^2}x_n^3 \end{bmatrix} \begin{bmatrix} w_{2,0}^{(2)}(t) \\ w_{2,1}^{(2)}(t) \\ w_{2,1}^{(2)'}(t) \end{bmatrix} \quad (24)$$

where,

$$x_1 = \begin{cases} \rho_1^{(2)} & ; 0 < \rho_1^{(2)} < l \\ 0 & ; l < \rho_1^{(2)} \end{cases}$$

$$l = \frac{r_1^{(2)}}{N-2}$$

Element 2 ~ $N-2$

$$w_{2,n}^{(2)}(t, \rho_1^{(2)}) = \begin{bmatrix} x_n^0 - \frac{3}{l^2}x_n^2 + \frac{2}{l^3}x_n^3 \\ x_n - \frac{2}{l}x_n^2 + \frac{1}{l^2}x_n^3 \\ \frac{3}{l^2}x_n^2 - \frac{2}{l^3}x_n^3 \\ -\frac{1}{l}x_n^2 + \frac{1}{l^2}x_n^3 \end{bmatrix}^T \begin{bmatrix} w_{2,n-1}^{(2)}(t) \\ w_{2,n-1}^{(2)'}(t) \\ w_{2,n}^{(2)}(t) \\ w_{2,n}^{(2)'}(t) \end{bmatrix} \quad (25)$$

where,

$$x_n = \begin{cases} 0 & ; 0 < \rho_1^{(2)} < (n-1)l \\ \rho_1^{(2)} - (n-1)l & ; (n-1)l < \rho_1^{(2)} < nl \\ 0 & ; nl < \rho_1^{(2)} \end{cases}$$

Element $N-1$

$$w_{2,n}^{(2)}(t, \rho_1^{(2)}) = [x_{N-1}^0, x_{N-1}] \begin{bmatrix} w_{2,N-2}^{(2)}(t) \\ w_{2,N-1}^{(2)}(t) \end{bmatrix} \quad (26)$$

where,

$$x_{N-1} = \begin{cases} 0 & ; 0 < \rho_1^{(2)} < (N-2)l \\ \rho_1^{(2)} - (N-2)l & ; (N-2)l < \rho_1^{(2)} < R^{(3)} \\ 0 & ; R^{(3)} < \rho_1^{(2)} \end{cases}$$

$\widehat{w}_{2,n}^{(2)}$: elastic deformation of element n at node n

$\widehat{w}'_{2,n}^{(2)}$: angle of rotation of element n at node n

At nodes 0 and N , we may set the condition that

$$\widehat{w}_0^{(2)} = \widehat{w}_N^{(2)} = 0$$

Then, an elastic deformation in body 2 is expressed

as

$$w_2^{(2)}(t, \rho_1^{(2)}) = B^{(2)}(\rho_1^{(2)}) \widehat{w}^{(2)}(t) \quad (27)$$

$$B^{(2)} = \begin{bmatrix} 0 \\ B_2^{(2)} \\ 0 \end{bmatrix}$$

$$B_2^{(2)} = [\widehat{b}'_0, \widehat{b}_1, \widehat{b}'_1, \dots, \widehat{b}_{N-2}, \widehat{b}'_{N-2}, \widehat{b}'_{N-1}]$$

$$\widehat{w}^{(2)T} = [\widehat{w}'_{2,0}, \widehat{w}_{2,1}, \widehat{w}'_{2,1}, \dots, \widehat{w}_{2,N-2}, \widehat{w}'_{2,N-2}, \widehat{w}'_{2,N-1}]$$

SPACECRAFT 3-AXIS ATTITUDE CONTROL BY SPACE ROBOT MOTION

Shin-ichi Tsuda*
and
Hirotaka Aoki**

*Professor, **Graduate Student

Department of Aeronautics and Astronautics
School of Engineering
Tokai University

1117 Kitakaname Hiratsuka, Kanagawa 259-1292 Japan

Phone:+81-463-58-1211 Fax:+81-463-50-2060, e-mail:stsuda@keyaki.cc.u-tokai.ac.jp

ABSTRACT

This paper is concerned with a spacecraft attitude control by a space robot motion. The space robot motion is characterized by the nonholonomic nature, in which a closed trajectory in the robot joint space yields an attitude change of the spacecraft. Two dimensional, i.e., planar, robot motion has been extensively discussed by many authors. Especially, an approach using Green's Theorem has been widely applied since the attitude change of a spacecraft is given by line integrals along each joint trajectory. A robot motion with closed joint trajectories can be transferred to a surface integral. Based on the above considerations, a simple and approximate method for a spacecraft 3-axis attitude control is proposed in this paper. A space robot is mounted on a spacecraft and is assumed to have three rotary joints, yaw, pitch and roll one. The spacecraft attitude will be controlled by the space robot motion where closed trajectory, cubic-like, in three dimensional joint space will be repeated cyclically. This corresponds to the closed trajectory motion on reduced three-two dimensional joint spaces and approximate method of two dimensional, planar, operation. A simulation result is also presented to show the validity of this approach.

I. INTRODUCTION

The nonholonomy is essential nature of free-flying space robotics and space articulated mechanisms. That is, depending on the trajectory of joint space, the attitude change of the body, which mounts robot and/or mechanism, will differ. This also shows the capability of attitude control by robot arm motion without using other devices, such as reaction wheels. Especially, free-flying space structure, if articulated mechanism like space

robot is mounted there, no other device is required to control the space structure.

In this paper an application of the nonholonomic nature to three-axis attitude control of the spacecraft is discussed utilizing motion of three degrees joint freedom, which is generally seen in usual space robot configuration. In nonholonomic case, attitude change of the spacecraft is given by line integrals along each joint trajectory. Cyclic closed joint operations can yield required attitude change. In two dimensional case it is easily shown that closed trajectory operation can be transferred to surface integral using Green's theorem. This paper applies the same idea as above to three dimensional case. Closed trajectory in three dimensional joint space will be reduced to three-two dimensional joint spaces by mapping. Obtaining closed trajectory on each two dimensional joint space, cubic-like closed trajectory will be generated and operated cyclically until desired attitude change is achieved. Also a numerical simulation result is given to validate our discussions. The approach shown in this paper is widely applicable to general free-flying space structure and mechanism.

II. ANALYTICAL DEVELOPMENT

Before proceeding to the general three dimensional case, we briefly discuss simple two dimensional case, i.e., planar motion of robot arm. Fig.1 illustrates a spacecraft with robot arm. There are one degree of freedom for the spacecraft attitude, θ , and two degrees of freedom for the robot joint, ϕ_1, ϕ_2 . As shown by many authors, c.g., reference [1], the attitude change,

$\Delta\theta$, due to the robot arm motion is represented by the line integrals along the arm joint trajectory.

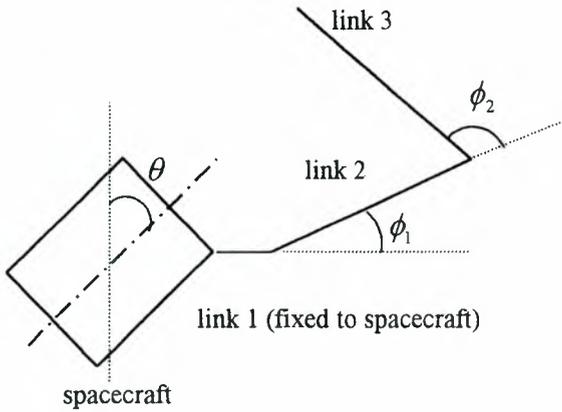


FIG.1 Definition of Planer Space Robot

$$\Delta\theta = \int_{\phi_{1i}}^{\phi_{1f}} A(\phi_1, \phi_2) d\phi_1 + \int_{\phi_{2i}}^{\phi_{2f}} B(\phi_1, \phi_2) d\phi_2 \quad (1)$$

where ϕ_{1i}, ϕ_{2i} are initial values, and ϕ_{1f}, ϕ_{2f} are final values of integration intervals. If the trajectory in the joint space is closed as seen in Fig.2, the attitude change of the spacecraft, along the trajectories I and II is given below.

$$\Delta\theta = \Delta\theta_1 - \Delta\theta_2 = \oint_{S(\Sigma)} [A(\phi_1, \phi_2) d\phi_1 + B(\phi_1, \phi_2) d\phi_2] \quad (2)$$

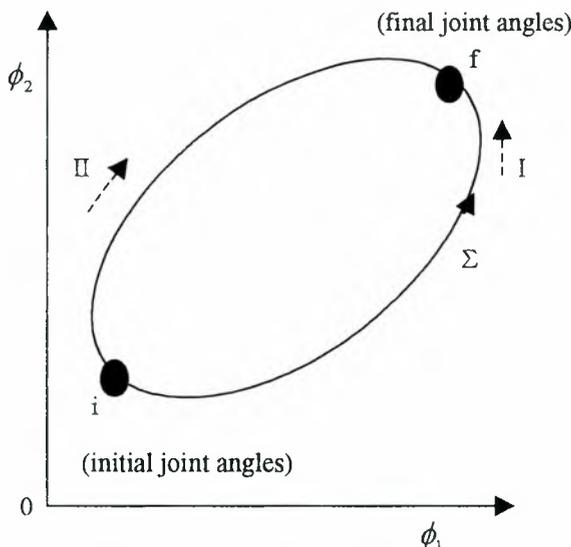


FIG.2 Closed Trajectory in Two Dimensional Joint Space

The above integral, can be transferred to the surface integral using Green's Theorem. That is,

$$\Delta\theta = \iint_{S(\Sigma)} \left[\left(\frac{\partial B(\phi_1, \phi_2)}{\partial \phi_1} \right) - \left(\frac{\partial A(\phi_1, \phi_2)}{\partial \phi_2} \right) \right] d\phi_1 d\phi_2 \quad (3)$$

where $S(\Sigma)$ equals to the area by the closed trajectory.

At this point if we define the configuration and mass properties of the robot arm, we will be able to obtain information on the integrand in eq.(3) over the entire operational region of the joint space. Fig.3 illustrates an example of contour map of the integrand.

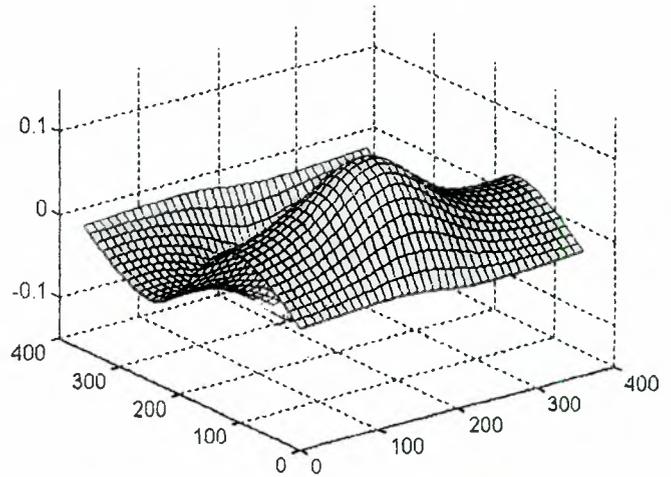


FIG.3 Illustrated Contour Map of Integrand

When we specify the closed trajectory in the joint space, the attitude change is available by calculating the eq.(3). An systematic way to generate attitude change of the spacecraft is to cyclically use an specific area on which the sign of the integrand is constant. This means a small closed trajectory robot arm motion will be repeated until enough attitude change is reached. The details of this feedback approach are given in reference [2].

Then, Let us consider a general space robot configuration, including a spacecraft, and joint allocation as shown in Fig.4. All the joints are assumed to be rotary without loss of generality. θ_x, θ_y and θ_z denote the spacecraft attitude angles around each axis. And ϕ_1 gives the shoulder yaw joint angle.

ϕ_2 and ϕ_3 represent the shoulder pitch and the elbow pitch joint angles.

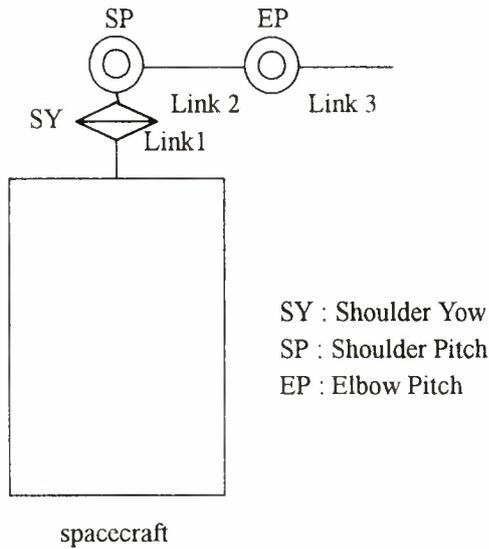


FIG.4 Definition of Three Dimensional Space Robot

As shown in previous two dimensional case, the spacecraft attitude changes, $\Delta\theta_x$, $\Delta\theta_y$ and $\Delta\theta_z$, are given by the following sum of line integrals:

$$\begin{aligned} \Delta\theta_x = & \int_{\phi_{1i}}^{\phi_{1f}} A_x(\phi_1, \phi_2, \phi_3) d\phi_1 \\ & + \int_{\phi_{2i}}^{\phi_{2f}} B_x(\phi_1, \phi_2, \phi_3) d\phi_2 \\ & + \int_{\phi_{3i}}^{\phi_{3f}} C_x(\phi_1, \phi_2, \phi_3) d\phi_3 \end{aligned} \quad (4)$$

$$\begin{aligned} \Delta\theta_y = & \int_{\phi_{1i}}^{\phi_{1f}} A_y(\phi_1, \phi_2, \phi_3) d\phi_1 \\ & + \int_{\phi_{2i}}^{\phi_{2f}} B_y(\phi_1, \phi_2, \phi_3) d\phi_2 \\ & + \int_{\phi_{3i}}^{\phi_{3f}} C_y(\phi_1, \phi_2, \phi_3) d\phi_3 \end{aligned} \quad (5)$$

$$\begin{aligned} \Delta\theta_z = & \int_{\phi_{1i}}^{\phi_{1f}} A_z(\phi_1, \phi_2, \phi_3) d\phi_1 \\ & + \int_{\phi_{2i}}^{\phi_{2f}} B_z(\phi_1, \phi_2, \phi_3) d\phi_2 \\ & + \int_{\phi_{3i}}^{\phi_{3f}} C_z(\phi_1, \phi_2, \phi_3) d\phi_3 \end{aligned} \quad (6)$$

And if we assume those are closed trajectories, then, applying Green's Theorem, we can rewrite above line integrals to the following sum of surface integrals.

$$\begin{aligned} \Delta\theta_i = & \frac{1}{2} \iint \left(\frac{\partial B_i}{\partial \phi_1} - \frac{\partial A_i}{\partial \phi_2} \right) d\phi_1 d\phi_2 \\ & + \frac{1}{2} \iint \left(\frac{\partial C_i}{\partial \phi_2} - \frac{\partial B_i}{\partial \phi_3} \right) d\phi_2 d\phi_3 \\ & + \frac{1}{2} \iint \left(\frac{\partial A_i}{\partial \phi_3} - \frac{\partial C_i}{\partial \phi_1} \right) d\phi_3 d\phi_1 \end{aligned} \quad (7)$$

where $i = x, y, z$.

Let us assume small closed trajectory in each two dimensional joint space, mapped from three dimensional joint space. Then, we can take a linear approximation in the following vector and matrix form:

$$\begin{aligned} \begin{pmatrix} \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{pmatrix} &= \begin{bmatrix} M_{xAB} & M_{xBC} & M_{xCA} \\ M_{yAB} & M_{yBC} & M_{yCA} \\ M_{zAB} & M_{zBC} & M_{zCA} \end{bmatrix} \\ &\times \begin{pmatrix} \Delta\phi_1 \Delta\phi_2 \\ \Delta\phi_2 \Delta\phi_3 \\ \Delta\phi_3 \Delta\phi_1 \end{pmatrix} = [M] \begin{pmatrix} \Delta\phi_1 \Delta\phi_2 \\ \Delta\phi_2 \Delta\phi_3 \\ \Delta\phi_3 \Delta\phi_1 \end{pmatrix} \end{aligned} \quad (8)$$

where abbreviations are $M_{iAB} = \frac{\partial B_i}{\partial \phi_1} - \frac{\partial A_i}{\partial \phi_2}$

($i = x, y, z$) and so forth. The left hand side of the eq.(8) means small three-axis attitude change vector of the spacecraft, and The vector in the right hand side represents a mapped closed area from three dimensional to two dimensional joint space. As far as such small closed trajectory operations are concerned, linearity is almost preserved between those spacecraft attitude changes and closed area. This is a basic idea to derive a corrective approach of spacecraft attitude using small closed trajectory. And if we can specify nominal joint

angles around which each joint is operated, we will be able to obtain the information on elements of matrix M by calculating integrands in eq.(7).

$$\begin{pmatrix} \Delta\phi_1 & \Delta\phi_2 \\ \Delta\phi_2 & \Delta\phi_3 \\ \Delta\phi_3 & \Delta\phi_1 \end{pmatrix} = [M]^{-1} \begin{pmatrix} \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{pmatrix} \quad (9)$$

Of course, we have to assure the invertibility of the matrix. This will depend on joints freedom allocation and arm configuration. And then, we can get information on three-two dimensional areas on each reduced, two dimensional joint space. Each joint travel is illustrated in Fig.5, where closed joint trajectory, a-b-c-d-e-f-a, will be projected onto each two dimensional joint space. In order to keep the linear approximation and achieve the convergence to the desired spacecraft attitude angle, we have to limit the area of each joint closed trajectory within small enough, therefore, a small constant gain k is introduced:

$$\begin{pmatrix} \Delta\phi_1 & \Delta\phi_2 \\ \Delta\phi_2 & \Delta\phi_3 \\ \Delta\phi_3 & \Delta\phi_1 \end{pmatrix} = k[M]^{-1} \begin{pmatrix} \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{pmatrix} \quad (10)$$

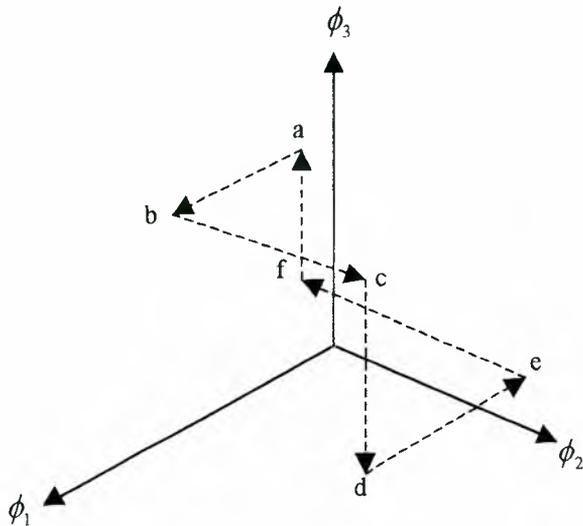


FIG.5 Closed Trajectory in 3 dimensional Joint Space

III. SIMULATION RESULT

In the preceding section, we have derived robot arm motion to achieve three axis attitude change of a spacecraft. A small closed trajectory, cubic-like, in the three dimensional joint space will be operated repeatedly to give small attitude change of the spacecraft. This is an approximate and extended method of the two dimensional ,planer, robot motion. In this

section we will show a numerical simulation result. Table 1 gives numerical data for physical properties of spacecraft and robot. Our simulation was to obtain 5 degree attitude change about each spacecraft body axis by space robot arm motion.

TABLE 1

	Spacecraft	Link 1	Link 2	Link 3
mass(kg)	2000.0	20.0	50.0	50.0
l (m)	3.5	0.25	2.5	2.5
I_x (kgm ²)	1400.0	0.1	0.25	0.25
I_y (kgm ²)	1400.0	0.1	26.0	26.0
I_z (kgm ²)	2040.0	10.0	26.0	26.0

Figs.6-8 illustrate calculated integrands in eq.(7) around specific joint angles. These values on the nominal joint angles represented those elements of the matrix in the right hand side of eq.(8) and its inversion in eq.(10). A small gain k in eq.(10) is assigned to be 0.15. In every cycle required attitude change vector, after each attitude correction, was given and, using eq.(10), three dimensional closed trajectory was generated and operated as shown in Fig.9. A history of attitude change is illustrated in Fig.10. As easily seen, a smooth attitude history has been obtained. This gives one aspect of the validity of our approach.

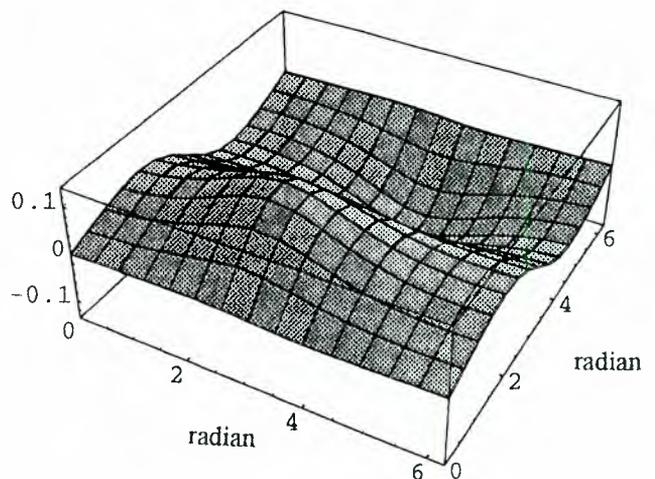


FIG.6 Contour Map for ϕ_2 and ϕ_3

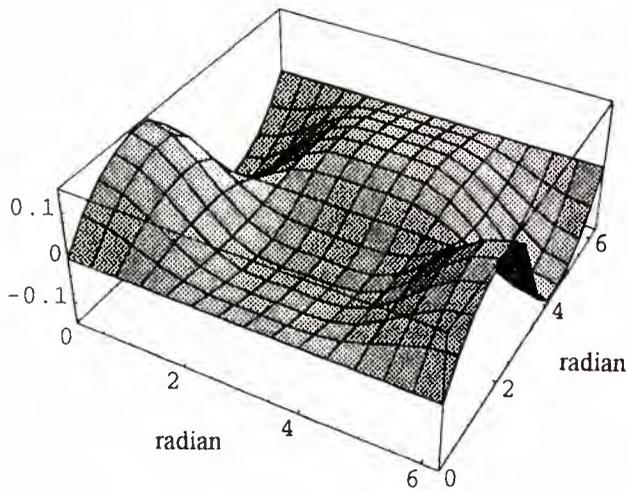


FIG.7 Contour Map for ϕ_1 and ϕ_3

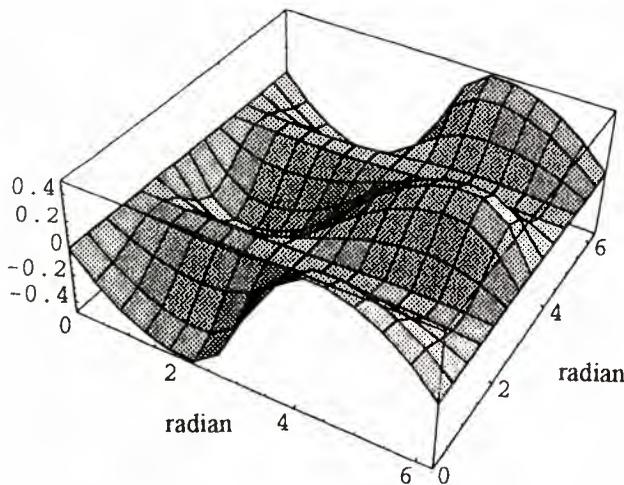


FIG.8 Contour Map for ϕ_1 and ϕ_2

IV. CONCLUSION

This paper is concerned with three axis attitude control of a spacecraft by a space robot motion. Three joint freedom is assumed to operate the space robot. Basically, an extension of two dimensional, planer, space robot operation is applied to three dimensional space robot motion strategy. Small closed trajectories in the three dimensional joint space is repeated to obtain small spacecraft attitude change. Cyclic motions of space robot eventually achieve required attitude change of the spacecraft.

Analytically, the above means three dimensional closed trajectory motion is projected onto two dimensional

joint space. And the approach derived from Green's Theorem is applied to transfer line integrals to surface integrals. Integrands of surface integrals are calculated around nominal joint angles and used to obtain an approximate closed area of each joint trajectory.

A numerical simulation was conducted to demonstrate the validity of our approach. Cubic trajectories in three dimensional joint space were generated and required attitude change of the spacecraft was achieved without unstable behavior.

Although the control method in this paper is based upon the linear approximation, three axis attitude of a spacecraft is well controlled by three joint freedom. This approach will be also applicable to the control of large space structure with articulated mechanism, so that no other device like momentum wheel will not be required.

REFERENCES

- [1]Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulator with Generalized Jacobian Matrix" IEEE Trans. ROBOTICS AND AUTOMATION, Vol.5, No.3, pp303-314
- [2]S. Tsuda et. al., "Nonholonomic Space Robot Path Planning for Spacecraft Attitude Stabilization and Control", The Institute of Space and Astronautical Science Report No. 670, September 1997

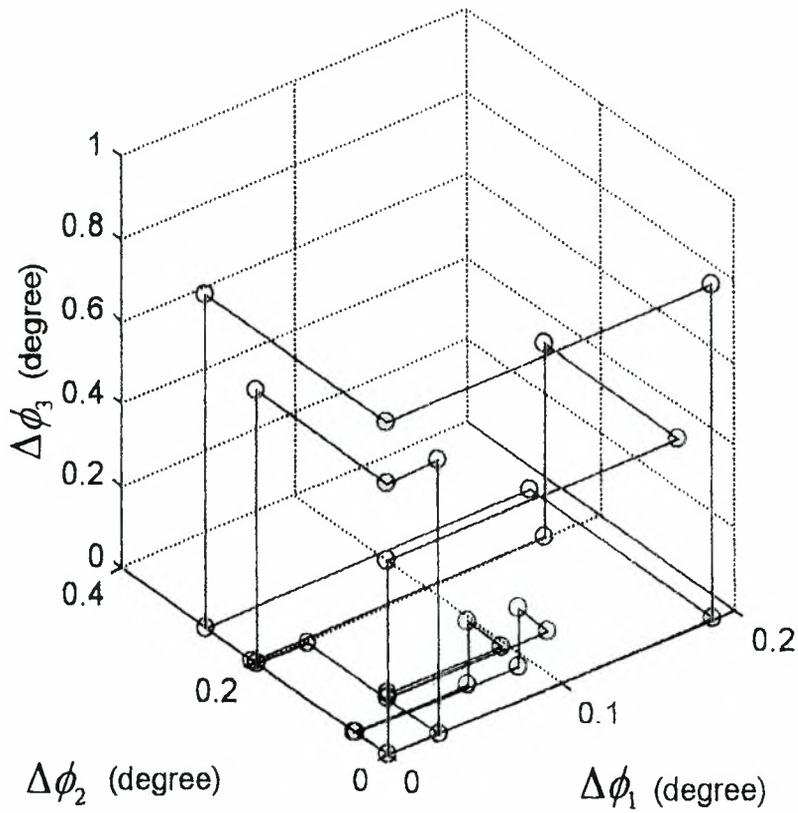


FIG.9 Closed Trajectory in Three Dimensional Joint Space for Numerical Simulation Case
 ($\Delta\phi_1, \Delta\phi_2, \Delta\phi_3$ is given around nominal joint angles)

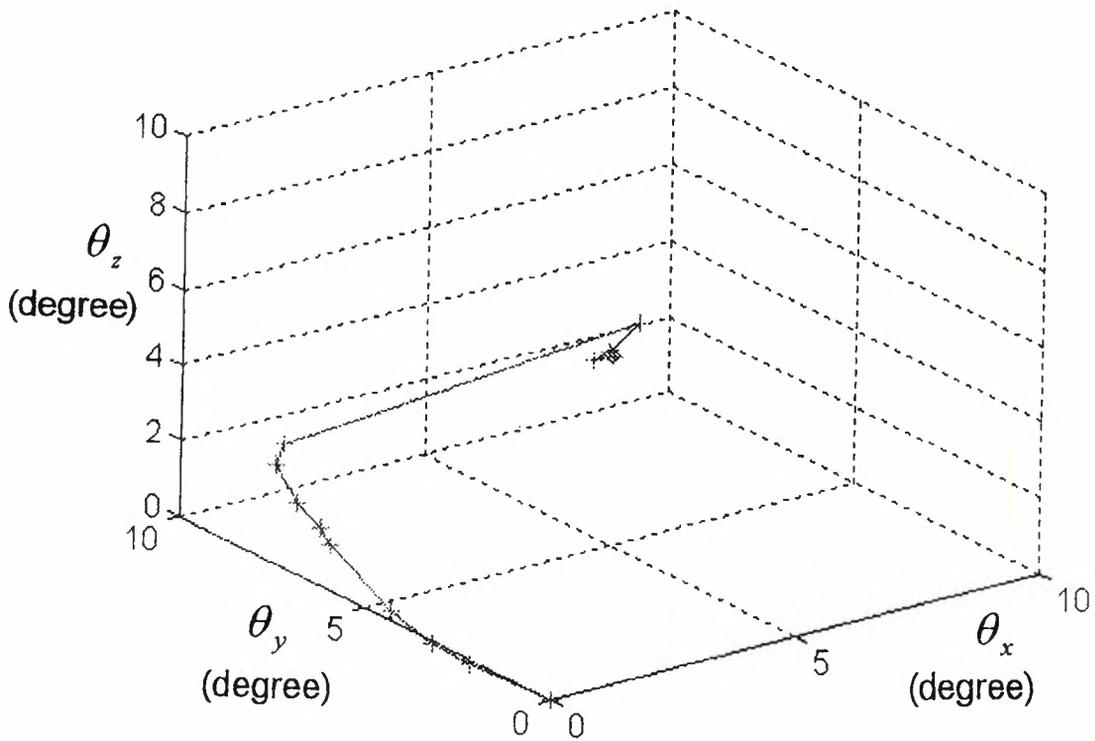


FIG.10 Spacecraft Attitude Change Profile by Repeated Closed Trajectory Arm Motion

Exhibits

Genetic algorithms used to determine WSB trajectories for the LunarSat mission

Wubbo J. Ockels^{1,2}, Robin Biesbroek³

¹ European Space Agency, office for education outreach project activities, PO Box 299 2200 AG Noordwijk

² Faculty of Aerospace engineering of the Delft University of Technology

³ GA Consultant, p/a European Space agency, PO Box 299 2200 AG Noordwijk

Abstract

This paper presents a novel way of constructing so-called Weak Stability Boundary (WSB) trajectories from GTO to the Moon, using genetic algorithms (GA). The work is part of the LunarSat project. LunarSat is an educational satellite sponsored by the new office for Education Outreach Activities and will function as the focus for a variety of related education activities. LunarSat (Lunar Academic and Research Satellite) is a 100 kg micro satellite, designed by young engineers and scientists and students from across Europe. It will be launched by Ariane 5 (auxiliary payload) into a GTO orbit around the millennium change. Once inserted into a 4hr Lunar polar orbit of perilune 100km above the south pole, it will observe the suitability of this area for locating the first extraterrestrial human outpost, i.e. measuring the solar illumination (peak of eternal light?) and the water ice distribution. Next to the cost, the main driver is mass. To maintain the 6-9 kg payload it is crucial to find the most economic trajectories from GTO to the Lunar polar orbit insertion. Being the auxiliary payload the launch date cannot be chosen. In general such trajectories make use of fly-by 's of the Moon and Earth as well gravity assists at the Weak Stability Boundaries of the Sun-Earth and Earth-Moon system. No systematic method is published for constructing weak stability trajectories from GTO to the Moon, although an ITT has been recently submitted by the Agency.

While the recognition of a suitable trajectory is easy, i.e. observing the total delta V, the elements for obtaining such a trajectory are very difficult to find. This type of problem,

where the optimum is easy recognisable, but where its relationship with the various parameters is extremely complex and perhaps chaotic, has been solved in nature by the 'survival of the fittest' evolutionary process (Darwin). The use of genetic algorithm, mimicking such evolution, is presented here, demonstrating that this is a powerful tool for finding creative solutions for trajectories with very low delta V's.

For this specific application of GA bonuses were applied to favour orbits close to the WSB .

A range of launch dates and times were investigated, where the leading parameter seems to be the angle between GTO apsidal axes and the Sun direction. This angle ranges from +6 deg to +20 deg for the standard Ariane 5 launch window. It is shown that the GA is capable of finding optimum trajectories for the range of -20 to +20 deg, with a delta V's of 1185 to 1250 m/s, thereby providing savings of 200-260 m/s with respect to a bi-elliptical trajectory. Next to these savings, the sun's gravity provided also the necessary out-off plane manoeuvre to reach the Moon, of which the orbit is inclined at 23 degrees at the end of the year 2000. The resulting trajectories have been confirmed using standard trajectory propagators (USOC, ESOC and GEODYN, Delft University of Technology).

Background

This work was part of the LunarSat study, of which phase A & B were sponsored by ESA's new office for Educational Project Outreach Activities. LunarSat's technical objectives are:

- Reach Lunar orbit from GTO via advanced, propellant saving transfer methods
- Orbit around the Moon for 6 months
- Investigate the Moon and particular the South polar region for its suitability for a first Lunar outpost

The space-craft data is:

- Mass: 100kg
- Dry mass: 59.74kg
- Payload mass: minimal 6kg
- Launcher: Ariane 5 ASAP → GTO orbit

Figure 1 shows the LunarSat satellite.

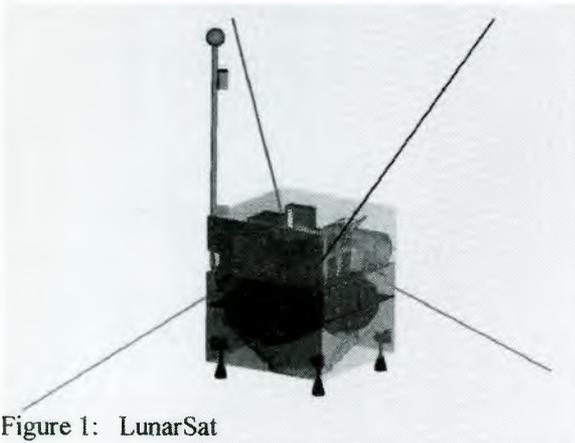


Figure 1: LunarSat

Problem definition

40% of space-craft total mass is fuel. Therefore, the total ΔV available is:

$$\Delta V = 2835 \cdot \ln\left(\frac{1}{1-0.4}\right) = 1460 \text{ m/s}$$

of which 10m/s is reserved for attitude & orbit control. Studies have proved that Lunar orbits exist requiring no orbit maintenance and therefore, the baseline ΔV available for the transfer orbit is 1450m/s. However, for contingency reasons, 100m/s is preferred for orbit maintenance. Therefore, the ΔV wanted for orbit transfer is 1350m/s.

Three Lunar transfer trajectory types exist:

1. Hohmann transfer (see figure 2).

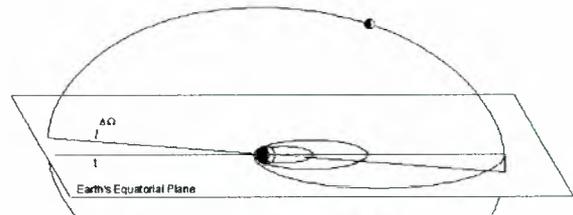


Figure 2: Hohmann transfer [RD1]

ΔV ranges from 1270 to 1770m/s. However, a Hohmann orbit is only possible when the GTO & Moon node are close together; a possibility which occurs only twice a year.

2. Bi-elliptic transfer (see figure 3)

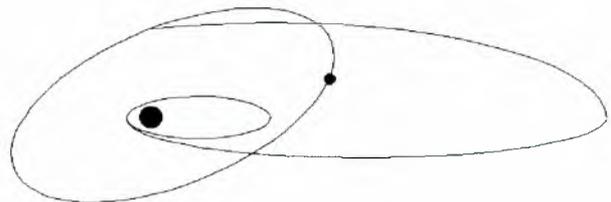


Figure 3: Bi-elliptic transfer [RD1]

ΔV ranges from 1380 to 1490m/s. With the available ΔV for LunarSat (1450m/s), the bi-elliptic orbit was chosen as the baseline for LunarSat: 1450m/s (no orbit maintenance.)

3. WSB transfer (see figure 4)

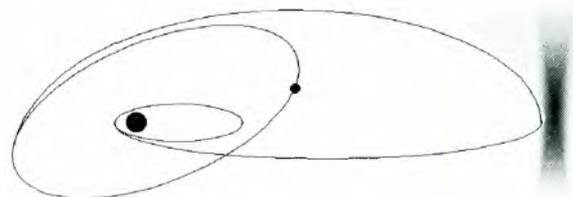


Figure 4: WSB transfer [RD1]

ΔV ranges from 1160 to 1300m/s and therefore, the WSB transfer increases payload mass by at least 4.5 kg compared to baseline bi-elliptic

This paper introduces a method to construct WSB transfers using genetic algorithms.

Genetic algorithms

A Genetic algorithm is an optimisation technique based on the mechanics of natural selection and genetics. GA's require the parameter set of the optimisation problem to

be coded as a finite-length string containing elements (such as 0, 1). A population of individuals is created which goes through a process of evolution made up of the principles of combination (cross-overs, mutation and selection)

Genetic Algorithms (GA's) are powerful when the optimum is easily recognisable but its relationship with the various parameters is complex. In this case: the optimum is the lowest total ΔV trajectory, which is indeed easy to 'detect'. However, the relationship with parameters (Sun & Moon positions, time, orbit parameters of transfer orbit) is very complex

Some GA techniques are described:

- Chromosome = string of bits
- Every chromosome represents a combination of parameters used to calculate the 'fitness' function (function of merit)
- population = set of 200 chromosomes
- Initial population is created randomly (every bit of every chromosome is randomly set to 0 or 1)
- Create new population / 'generation' using:
 - Selection (pairs of chromosomes ('parents') are selected according to their 'fitness': a higher value of the fitness function assures a higher chance for selection)
 - Cross-over (chunks of bits are swapped between two parents with probability 0.6)
 - Mutation (a bit is randomly changed (1→0) or (0→1). Probability depends on similarity)
- This process is repeated until the fitness is sufficient (typically 500 generations).

Optimisation method

The optimisation method used can be described as follows (see figure 5):

For a given launch date/hour:

- Wait 'Tphase' days in phasing orbit
- Give $\Delta V1$ in perigee to increase the apogee to reach WSB region
- In the WSB region, give a very small $\Delta V2$ to change the inclination and perigee
- Upon arrival at the Moon, calculate how much $\Delta V3$ is required to be captured into 4hr polar orbit.
- Minimise the total $\Delta V = \Delta V1 + \Delta V2 + \Delta V3$

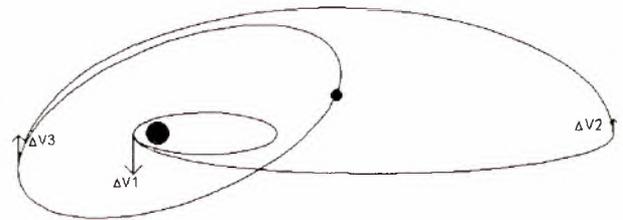


Figure 5: Overview of ΔV 's given

If the space-craft doesn't arrive at the Moon after 150 days, stop the integration.

Implementation into GA

Five parameters were chosen to be optimised:

1. Tphase: time spent in GTO or phasing orbit
2. $\Delta V1$: amount of ΔV given at GTO perigee
3. $\Delta V2$: amount of ΔV given at WSB region
4. α : azimuth of $\Delta V2$
5. δ : declination of $\Delta V2$

One chromosome represents all parameters in concatenated order:

Bit: 1 13 36 61 70 77

Tphase	$\Delta V1$	$\Delta V2$	α	δ
--------	-------------	-------------	----------	----------

The fitness function is defined as:

$$f \cong -\Delta V_{\text{total}} \text{ [km/s]}$$

A 'bonus' is added when the WSB region is reached (maximum distance Earth-satellite is between 1.32 and 2 million km).

Results

The GA was able to find WSB transfers for each day in a year. Figure 6 shows a typical resulting WSB transfer.

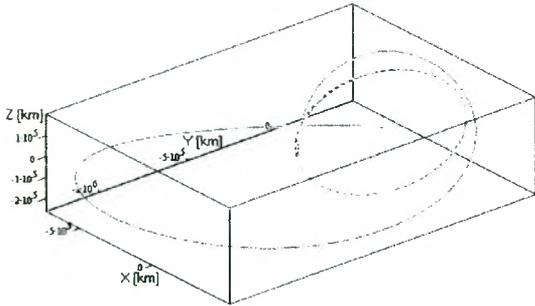


Figure 6: Typical result of GA

The ΔV_{total} is always below 1232 m/s for standard Ariane 5 launch window (compare to 1450m/s base-line), see figure 7.

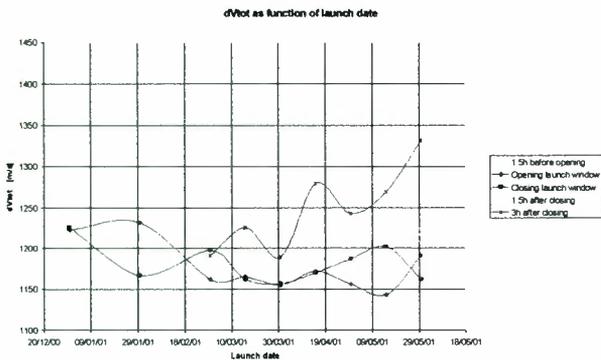


Figure 7: ΔV for all cases treated.

Cases treated outside standard launch window below 1331m/s, see figure 8.

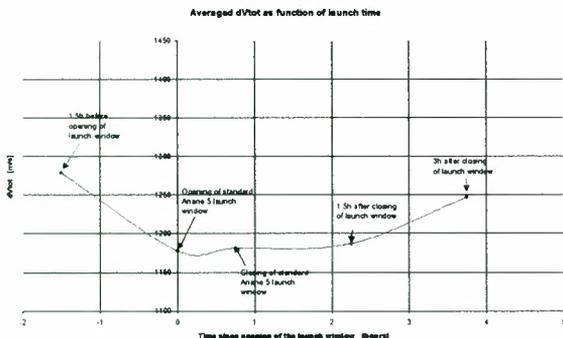


Figure 8: Average ΔV per launch hour

Conclusions

The genetic algorithm was able to find WSB transfers compatible to the constraints given, for all cases treated.

Some creative solutions were found using swing-by's, see figure 9:

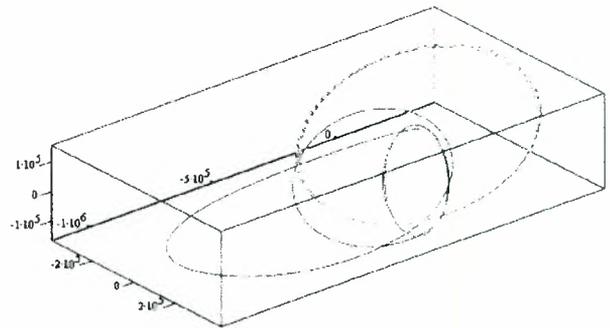


Figure 9: Solution with swing-by's

Savings in ΔV range from 218 to 265m/s corresponding to 4.7 to 5.8kg increase in payload mass over 6kg baseline as seen in figure 10.

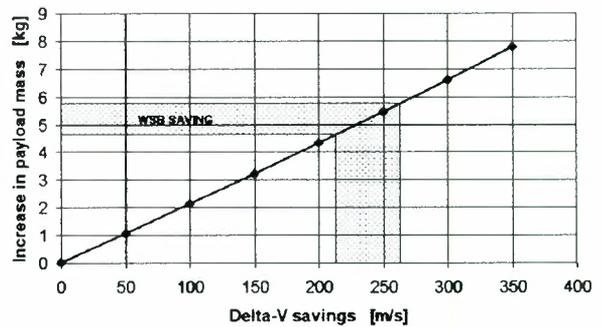


Figure 10: Payload mass increase

A detailed report on this work can be found in [RD2].

References

[RD1] *Lunar Mission From GTO*, W. Seefelder, RT-DA 98/05 Muenchen.
 [RD2] *Study on Lunar Trajectories from GTO by Using Weak Stability Boundary Transfers and Swing-by's*, R. Biesbroek, EWP 2014, ESTEC, Noordwijk.

Force Simulation in Telerobotic System with Large Time Delay

Zhuang Jun

Room 223, Blvd. 1, Tsinghua University
Beijing, 100084, P.R.China
email: zjun@th-dascom.com.cn
Tel: 86-10-62781479

Sun Zengqi

Dept. of Computer Science and Technology
State Key Laboratory of Intelligent Technology and Systems
Tsinghua University, Beijing, 100084, P.R.China
email: szq-dcs@mail.tsinghua.edu.cn
Tel: 86-10-62785118

Cheng Peng

Dept. Of Electrical Engineering, Tsinghua University
Beijing, 100084, P.R.China
email: chp@psai.eea.tsinghua.edu.cn
Tel: 86-10-62778854

ABSTRACT

This paper introduces a telerobotic system. In this system, a distributed delay-compensating 3-D stereographic simulator is implemented in SGI ONYX/4 RE² with SPACEMOUSE, HMD devices, Sirius Video. Estimated Force Emulation can protect the real robot in time from being damaged in collision. A two hand six DOF master arm with force feedback interface is used in the controller. The simulation of free-floating manipulator is achieved in this system. The command sequences are generated at the same time with the movement of the simulating robot and are sent to the real robot after the simulating time delay. The images gotten from the camera are sent back to make overlapping to the simulating robot with time delay. Virtual reality technology and shared control are supported in this system. Some basic tasks are accomplished by controlling PUMA560 robot.

1. INTRODUCTION

For humans to expand permanently into space, teleoperated robots can function indefinitely in space environment. Teleoperation allows humans to participate intimately yet safely in taming the space environment. Technologies for ground-remote telerobotics have been developed to support ground-based control of space-based robots. Communicate time delay between the local site operator control station, where the operator resides, and the remote site robot control system is a significant factor for ground control telerobotics. A round-trip delay of several seconds or more is expected. The time delay precludes control mode which require closed loop control between the local and remote sites. The operator can not operate the remote site robot in time according to the video image. An alternative method

is to generate command sequences through interaction in a local graphical environment. By simulating the real remote site robot and environment, we can control remote robot easily and efficiently. We introduce distributed system in Section 2.

The Estimated Force Emulation, which is used to protect the real robot in time from being damaged in collision, is introduced in Section 3.

In section 4, a simulation of free floating manipulator is presented.

To verify the control effect, a method is needed to resolve the mapping problem between the simulating robot and the real robot. And the graphical environment is connected to the PUMA560 robot. One camera are placed in front of the real robot to take the images and send them back to make overlapping with the graphical robot. This part will be described in Section 5.

In Section 6, a two hand six DOF master arm with force feedback interface is introduced, two robots can be controlled by this machine.

This system was used to control PUMA560 robot to perform some basic tasks, this will be introduced in Section 7.

2. Distributed Simulation System

As a trend, the calculation occupies more and more resources in emulation systems. In order to take full advantage of 3D Graphics Systems' preponderance on direct graphic interface, we designed the emulation system as distributed system. The functions that needs a lot of calculations include robot kernel emulation, collision detection and estimated force emulation. By assigning these functions into different high-end PCs, we can ensure the quality of 3D graphs. We chose TCP/IP as the network protocol because the system may be

heterogeneous. We use a central computer to synchronize all the computers and I/O devices in the system.

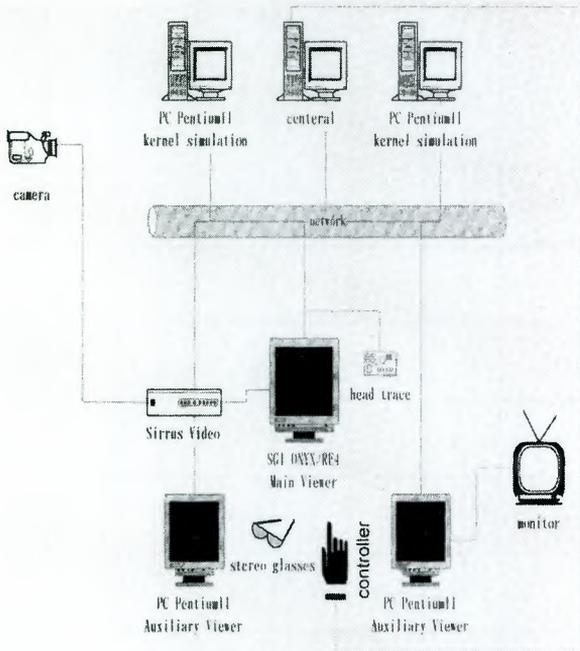


Fig. 1 System Framework

The central computer is in charge of following service:

1. Declaration Management. Allow members to specify the types of data they will send or receive by object class and attribute name.
2. Object Management. It supports creation, modification, and deletion of objects.
3. Time Management. It control advancement of system members along with central time.
4. Ownership Management. Allow system members to transfer ownership of object attributes.
5. Data Distribution Management. Allow system members to specify the distribution conditions for the specific data they send or expect to receive.

Virtual Reality technology is supported. WTK is used to deal with the VR programming. Three-dimensional stereo glasses are used to get 3-D stereograph. The SPACE MOUSE is used to control the simulating robot in the environment.

3. Estimated Force Emulation

The main purpose of Estimated force emulation is to protect the real robot when collision occurs. When the robot collides with some obstacles, operator should be able to detect the right direction of this collision from the hand controller with force feedback so the further damage to the robot can be avoided. When the system detects collision, it calculates the mutual force between these two objects. Then the system will calculate the equivalent force on the tip of the robot from the forces we just got, and finally, it will reflect the size and

direction of the force to the hand controller with force feedback.

According to the robot's DH parameters, the transformation matrix from the coordinate which fixed on the joint $i-1$ to the coordinate which fixed on the joint i is:

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta & -\cos\alpha*\sin\theta & \sin\alpha*\sin\theta & a*\cos\theta \\ \sin\theta & \cos\alpha*\cos\theta & -\sin\alpha*\cos\theta & a*\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Because of these matrixes, the accurate position of every point in every joint can be calculated. To improve the effect of simulation, the columns which are embodied to the joint are used for collision detect. Only if the column hits the obstacle, the accuracy plane collision detection is used to get the collision information and the load node. A ball obstacle is put into the test system.

We assume the manipulator is rigid, and the obstacles are linear elasticity.

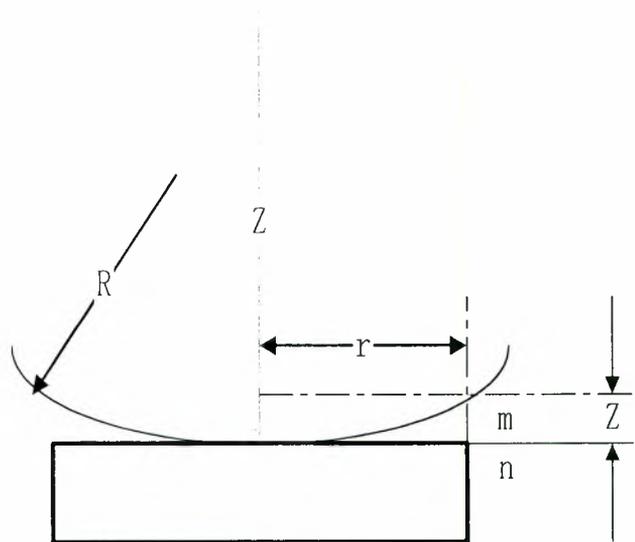


Fig 2 Mutual Force Analysis

$$z = \frac{r^2}{2R}$$

z is very small.

When the plane move toward the ball, the distance between plane and point M will become closer and closer. w is the displacement of the point M, which is caused by the distortion of the ball. The max displacement of the peak point in ball is d ,

$$d - w = \frac{1}{2R}r^2$$

$$w = d - \frac{1}{2R}r^2$$

The interface between the plane and the ball is circle. The pressure density of the interface q is symmetry to the center point O of the interface.

$$w = \frac{(1-\nu^2)}{\pi E} \iint q ds d\psi$$

$$k_1 = \frac{1-\nu^2}{\pi E}$$

The radius of interface between plane and ball is a, and the press density in the center of the interface is q_0
 $q_0 = ka$

The distribution of the press density is according to the hemisphere. The radius of the hemisphere is a.

$$k \iint q ds d\psi$$

$$= k \int_0^\pi \frac{q_0 \pi}{2a} (a^2 - r^2 \sin^2 \psi) d\psi$$

$$\frac{kq_0\pi}{2a} \int_0^\pi (a^2 - r^2 \sin^2 \psi) d\psi = d - \frac{1}{2R} r^2$$

$$\frac{kq_0\pi^2}{4a} (2a^2 - r^2) = d - \frac{1}{2R} r^2$$

For every r, the equations are true. Then a and d must suit for following equations.

$$\begin{cases} d = \frac{kq_0\pi^2 a}{2} \\ a = \frac{kq_0\pi^2 R}{2} \end{cases}$$

$$a = \sqrt{dR}$$

$$q_0 = \frac{2\sqrt{d}}{k\pi^2 \sqrt{R}}$$

$$\frac{q_0}{a} = \frac{2}{kR\pi^2}$$
 is the scale constant of the press

density distribution.

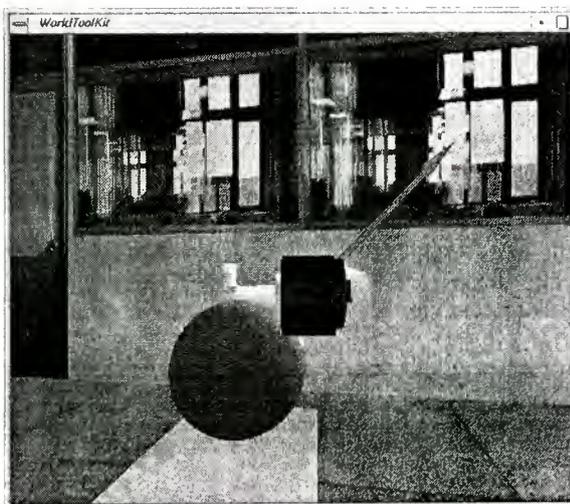


Fig3 Estimated Force Emulation

According to those formulas, the direction and the value

of the mutual force can be determined. It shows in Fig 3. The arrow in the fig 3, identifies the direction, value of the mutual force and the load node.

4. FREE FLOATING MANIPULATOR SIMULATION

When the quality of base is close to the quality of manipulator, the movement of manipulator will cause the movement of base in aerospace. According to the convert from free-floating manipulator to manipulator with fixed base, the coupling between base and joint can be resolved. The convert method is standard model transition method, which is called as dynamics equal manipulator (DEM). The kinematics of free-floating manipulator can be achieved. The simulation images are followed.

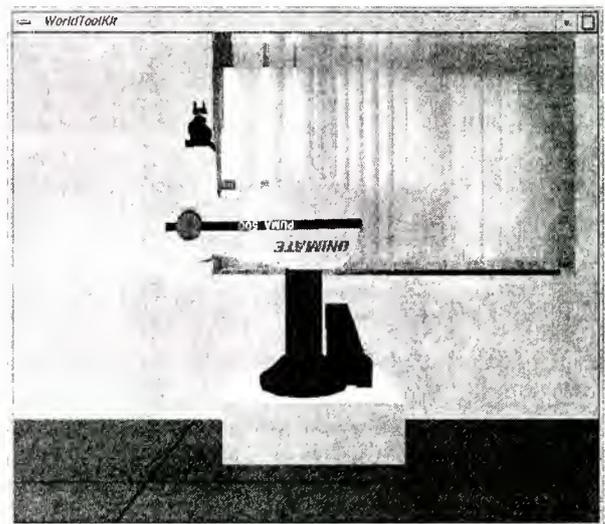


Fig4 Free Floating Manipulator 1

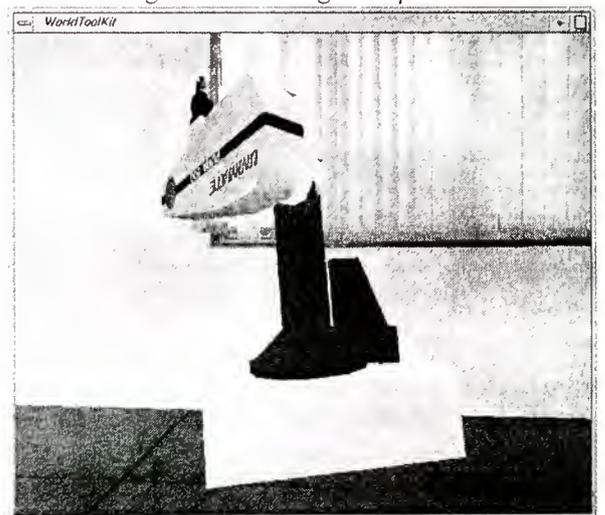


Fig5 Free Floating Manipulator 2

5. TO VERIFY THE CONTROL EFFECT

In Space Remote Operating Robot System, the transmission delays between space robot and ground station are always changing according to the variation of the environment. This inconsistency forbids us from using a constant to make the one-to-one mapping between emulated robot data and actual robot data. To solve this problem we attached a timestamp T to both upper-bound and lower-bound data. In upper-bound data, this parameter indicates in T seconds after the space robot received the first operating code, its position and attitude must fulfill the requirement that is contained in the code. In lower-bound data, this parameter means in T seconds after the space robot received the first operating code, its position and attitude is kept in this group of data. We now can get the one-to-one mapping of emulated robot data and actual robot data by this parameter T .

The PUMA560 robot was connected to the graphical environment. The graphical robot and the real robot have the same structure, so the joint-to-joint control is used to control the real robot. When the simulating robot move, the command was sent to the real robot after the simulating time delay to control the real robot to move as the simulating robot.



Fig. 6 Overlapping between image and graph

To do overlapping between the image and the graph, we should first calibrate the camera and demarcate the viewpoint and then to overlap the graphical robot on the image. In the Fig. 6, the background is the image gotten from the camera in real time. The frame drawn by line is the graph of the simulating robot created by the computer. If the frame can overlap the image of the real robot all the time, we can say the control is successful.

6. SIX DOF MASTER ARM WITH FORCE FEEDBACK INTERFACE

The structure of the master arm is STANFORD

structure, and the type is RRPRRR. It is universal hand controller. To compare with PPPRRR master arm, its structure is quite simple, but the control system must calculate kinematics. Because the structure of the master arm is different to the structure of PUMA 560, then we control the end of the PUMA 560 to trace the movement of the master arm. And it can accept the Estimated Force Emulation information through force feedback interface.

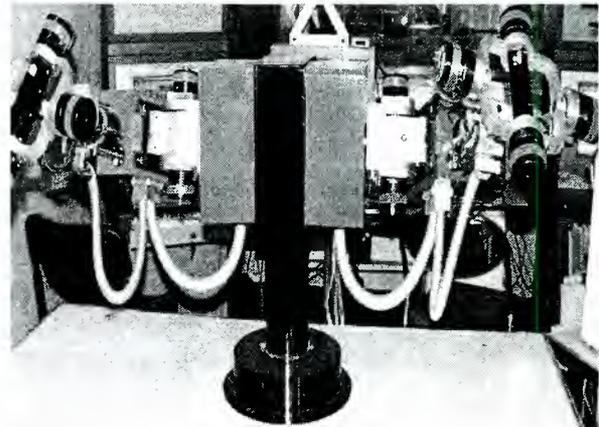


Fig 7 Six DOF Master Arm

7. THE BASIC TASKS

The system was used to perform the following tasks:

1. plug a pole into a hole
2. assemble a structure from several identical cube parts
3. two arms grasp an object together
4. two arms to do plugging operation coordinately

In the tasks, the real robot could repeat the graphical robot's movement after the simulating time delay. The frame of the graphical robot can cover the image of the real robot all the time. The result is successful.

8. CONCLUSION

Many aspects of robotics were integrated into the system. Key technologies of the system have been:

1. High quality graphical simulation, the model was displayed with texture.
2. Distributed Simulation System.
3. Estimated Force Emulation.
4. Free Floating Manipulator Simulation
5. Six DOF master arm with force feedback interface
6. By adjusting and overlapping of the camera images and the computer graphs, the effect of the control can be judged.

To improve the performance of the system, many technologies and methods will also be used into this

system such as share control, etc.

REFERENCES

- [1] Schrader, C.B.;Fisher, F.E. "Optimized Telerobot With Real-Time Kinesthetic Feedback", Biomedical Engineering Conference, 1998. Proceedings of the 17th Southern
- [2] Bon, B.; Seraji, H.. "Real-Time Model-Based Obstacle Detection for the NASA Ranger Telerobot". IEEE International Conference on Robotics and Automation, (2). 1997, pp.1580-1587
- [3] Cha, D. H.; Cho, H. S.; Kim, S.. "Design of a Force Reflection Controller for Telerobot Systems using Neural Network and Fuzzy Logic". Journal of Intelligent and Robotic Systems, 16(1), 1996, pp1-24
- [4] G.Hirzinger, K.Landzettel, Ch.Fagerer. "Telerobotics with Large Time Delays - the ROTEX Experience". Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp571-575
- [5] C. Weisbin, D. Perillard, "Jet Propulsion Laboratory Robotic Facilities and Associated Research". Robotica (1991) volume 9, pp 7-21
- [6] P.Michelman, P.Allen. "Shared Autonomy in a Robot Hand Teleoperation System", Proc. IEEE Int. Conf. Intelligent Robots and Systems 1994, Munich, Germany, September 12-16, 1994, pp253-259
- [7] Tsumaki, Y., Uchiyama, M.. "Predictive display of virtual beam for space teleoperation". Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 96. Robotic Intelligence Interacting with Dynamic Worlds, vol.3 pp.1544-1549

LIMITATIONS OF HARDWARE-IN-THE-LOOP SIMULATIONS OF SPACE ROBOTICS DYNAMICS USING INDUSTRIAL ROBOTS

Rainer Krenn, Bernd Schäfer

Deutsches Zentrum für Luft- und Raumfahrt e.V.

Institute of Robotics and System Dynamics

82234 Wessling, Germany

phone: +49 / 8153 –1463, fax: +49 / 8153 – 1450, e-mail: rainer.krenn@dlr.de

phone: +49 / 8153 –1191, fax: +49 / 8153 – 1450, e-mail: bernd.schaefer@dlr.de

ABSTRACT

The utilization of space robotics components in operational missions requires intensive pre-mission simulations, especially if dynamic influences will occur in the robotic manipulator, in its platform and the handled objects. Forces and torques are caused by manipulator activities and influences of the attitude control system of the satellite platform and lead to deformations and oscillations of elastic system components.

To cope with these problems in the first step software simulations of the kinematic and dynamic system behaviour have to be performed. Due to uncertainties in the modelling of tools, sensors and contact mechanics it would be advantageous to extend the simulation to a hybrid one integrating available flight hardware in the simulation loop. To simulate the manipulator's endeffector motion an industrial robot is used.

However integrating hardware requires real-time conditions of the simulation. Due to limitations of the sampling rate and dead times in the signal flow instabilities of the simulations may occur depending of the dynamics of the handled object.

1. INTRODUCTION

The efficient use of robotic manipulators for space applications requires intensive simulations on ground in order to guarantee for mission success, especially for mission critical manipulator operations such as deployment of the manipulator system, berthing of heavy loads and the use of tools under uncertain conditions. But also influences of the attitude control system during attitude correction manoeuvres of the satellite platform have to be considered. In all these situations strong dynamic effects occur. The forces and torques on the manipulator system lead to deformation of elastic mechanical components within the manipulator like joints and structural parts of the arm limbs.

The consequences are:

- failure of endeffector fine positioning tasks
- oscillation of the manipulator around an ideal trajectory
- undesired movement of handled components
- influences in the function of tools and working procedures.

To cope with these problems simulations on ground of the kinematic and the dynamic behaviour of the complete system during action are required. In the first step simulations can be performed in software. They may be sufficient in case of contact-free manipulator motions or on component level. But difficulties arise if the robotic tool itself and the contact mechanics between the tool and the handled object during operation have to be modelled. A similar problem is to model the behaviour of sensor systems and video based operations.

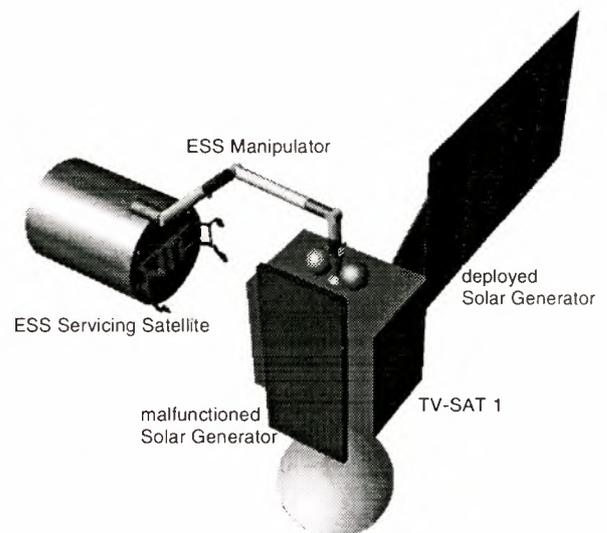


Figure 1: ESS Berthing Manoeuvre

A typical example of space robotic based activities has been recently given by the German Experimental Servicing Satellite (ESS) study (Figure 1). Here, two relevant problems were encountered: First, the capturing and berthing of a non-cooperating target satellite (TV-SAT 1), and second, the use of a repair tool for cutting a clamping bolt that still prevents one of the solar generators from proper deployment and therefore the satellite from operation.

In such a case the modelling effort to implement a valid software simulation would increase tremendously. On the other hand, model simplifications are expected to falsificate the simulation results remarkably.

A very efficient solution to overcome these modelling problems is to make use of a hybrid simulation technique, where hardware parts, such as electro-mechanical tools are incorporated in the simulation

environment. Often, these parts are already available as breadboard models, prototypes or even as a space applicable version. Then, an industrial robot plays an important role as a generic motion system. Its only task is to carry hardware equipment of the overall simulation and to follow a trajectory calculated in the software part of the hybrid simulation loop.

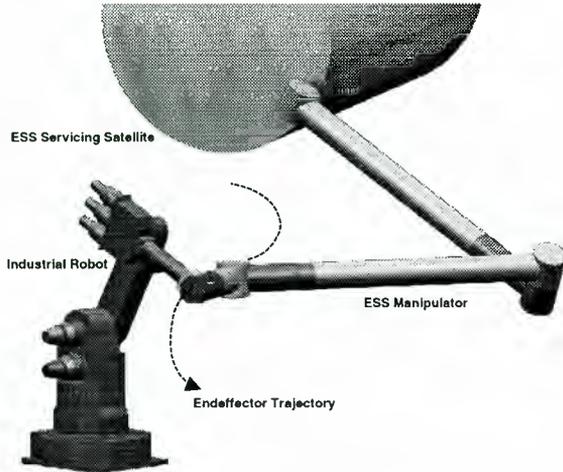


Figure 2: Physical Trajectory Display

Usually, the trajectory simulated by software is the one of the space manipulator endeffector, but every trajectory of interest can be commanded to the industrial robot equally. Figure 2 shows the industrial robot physically displaying an endeffector motion of the ESS manipulator by driving its own endeffector on

the simulated trajectory. It includes all dynamic effects such as joint oscillations, structural arm vibrations, interactions between the satellite base system and the moving manipulator and disturbances by the attitude control system of the satellite.

During contact phases, e.g. during the operation of endeffector tools, the equivalent forces and torques are measured by a force/torque sensor connected to the tool. The sensor signals are fed back to the software simulation and a new trajectory state vector is calculated which the industrial robot has to follow again. Figure 3 shows the signal flow inside the hybrid simulation set-up for an operator commanded system.

It has to be noticed that all software components of the simulation loop must be processable in real time, due to the fact that dynamic effects cannot be stretched in the time space. So it has to be carefully proven, that the calculation accuracy is high enough even if the simulation time steps are coarser compared with off-line, non real time simulations.

2. CONCEPT TESTS

In the very first steps of the realization of the hybrid simulation concept, a large manipulator with elastic joints was modelled and its motion was calculated in software. Its endeffector was driven physically displayed by the endeffector motion of the industrial robot in the lab. The task was to touch and to push the endeffector against elastic material like rubber foam or a sheet metal of copper (Figure 4).

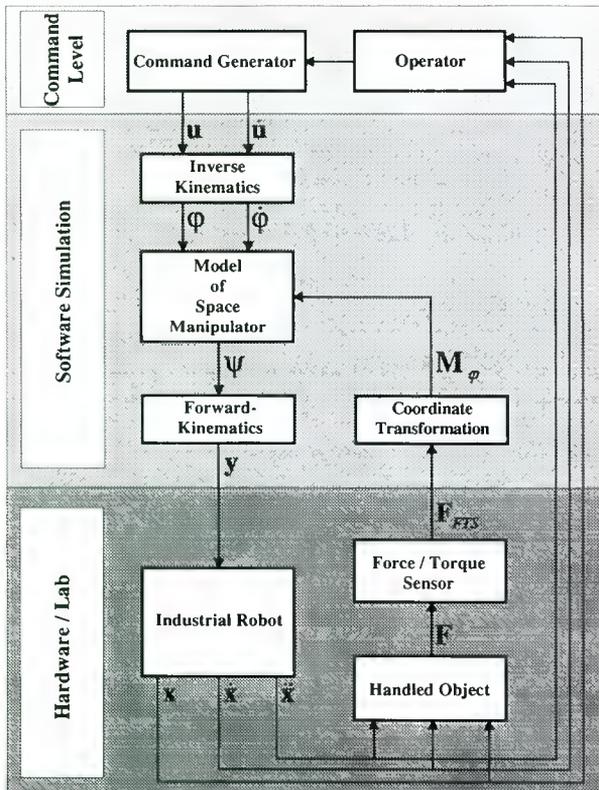


Figure 3: Signal Flow in the Simulation Loop

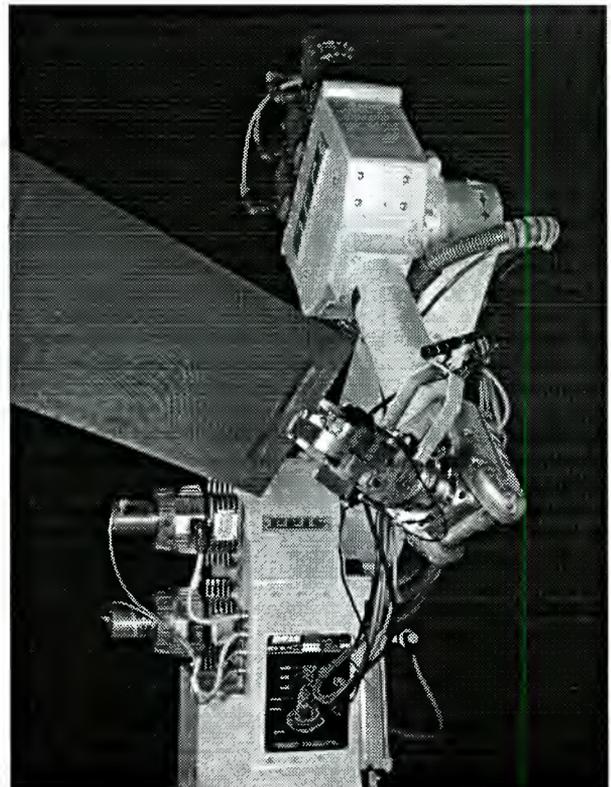


Figure 4: Industrial Robot Touching Elastic Material

In this conceptual phase it was found that in some situations the simulation becomes unstable depending on the dynamic properties of the handled object. This was surprising because the simulated manipulator system was a passive one without any controller influences.

To find the reasons for this problems the dynamics of the industrial robot was analyzed. The most important features are

- the dynamic transfer behaviour of the whole robot system and
- the time delay in executing commands.

Concerning the amplitude transfer function it was found that the robot has an almost ideal behaviour in the working volume of relevant kinematic configurations (Figure 5). In the frequency range up to 5 Hz the robot shows a small linear increase of the amplitude ratio. But this deviation is easy to filter so that we can consider the robot as an ideal transmitter in the range less than 5 Hz.

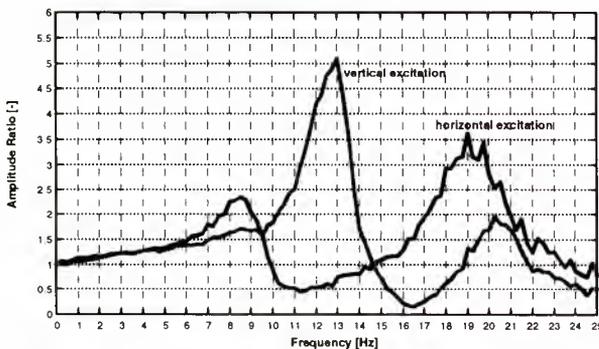


Figure 5: Transfer Function of the Industrial Robot

The time delay between sending the command and executing the motion depends on the interface between the software simulation and the robot controller. Using a high level interface with a command rate of 50 Hz the time delay is 40 ms. In case of removing control functions from the robot controller to the simulation set-up a command rate of 400 Hz is required. In this mode the dead time can be reduced down to 17.5 ms (Figure 6).

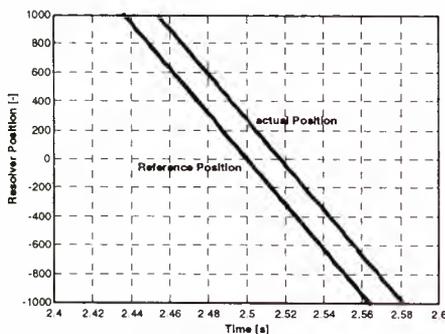


Figure 6: Time Delay Command – Motion

3. MODELLING OF THE SIMULATION SET-UP

To verify the effects of unstable simulations theoretical investigations were performed. In this step the complete hybrid simulation was modelled in software only, including all parts originally integrated in hardware. This provides an easy way to perform parameter variations of all interesting aspects of the simulation set-up.

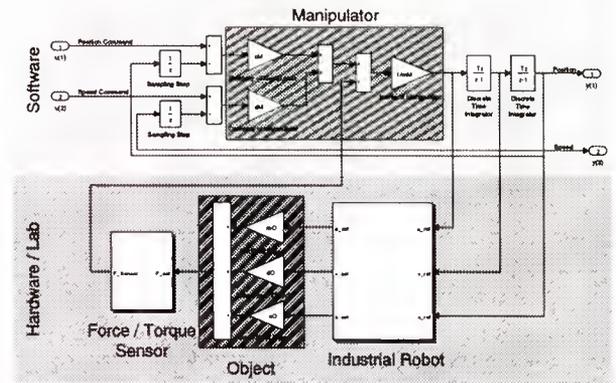


Figure 7: Block Diagram of the Hybrid Simulation

Figure 7 shows the block diagram of the software representation of the hybrid simulation for 1 degree of freedom. In the software layer the dynamic properties of the simulated manipulator and the time integration for the motion calculation can be found. The hardware layer includes all parts usually realized in hardware. In this example these are the industrial robot with its transfer behaviour, the dynamic properties of an object to be gripped and a force / torque sensor. The simulation loop is closed via sensor data feed back to the manipulator motion calculation.

To perform the investigation with a realistic data base the scenario of the ESS study (Figure 1) was used. The defined task to be simulated was to grip the deployed solar generator. There were two steps to reduce the dynamic model. First, the ESS manipulator was reduced to a two link manipulator (2 pitch links) with elastic joints. It includes the geometrical properties as well as the dynamic properties like masses, elasticity and damping.

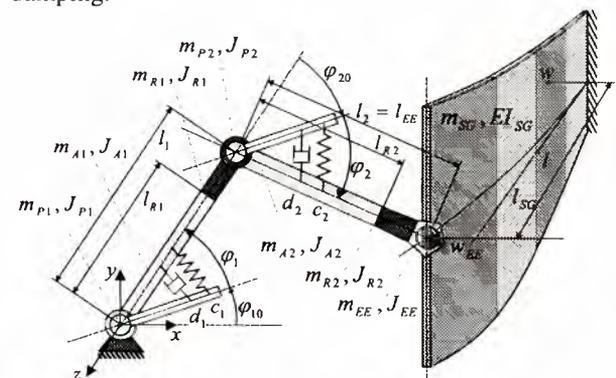


Figure 8: Reduced ESS Scenario Model

The model of the solar generator is a flexible beam with dynamic properties of the original one (Figure 8). In the second step of model complexity reduction it is assumed that endeffector motions appear only in one direction, namely the deflection direction of the solar generator. In this case the joint positions φ_1 and φ_2 are not independently to be chosen. Thus the manipulator is described as a dynamic system with 1 degree of freedom. Assuming that for simulation aspects only the first oscillation mode of the solar generator is relevant, it can be reduced also to a system with 1 degree of freedom. The result of model complexity reduction is shown in Figure 9. In the following text the handled system, e.g. the solar generator, is more generally called object.

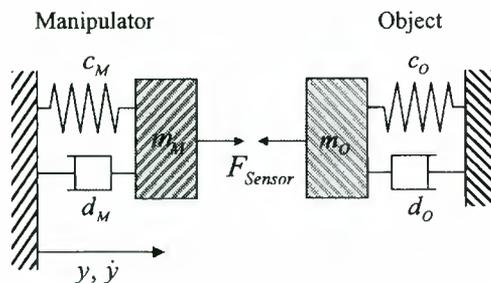


Figure 9: System with 1 Degree of Freedom

The parameters of the dynamics of the 1 DOF system (Figure 9) can then be transferred to the according blocks in the block diagram of Figure 7. The internal forces between the manipulator and the object are not theoretically calculated. They are measured and depend therefore on the dynamic behaviour of the force / torque sensor and the industrial robot which follows the manipulator trajectory. The estimation of the system behaviour is based on calculations of the system eigenvalues. The operating point is defined by both position 0 and speed equal to zero.

4. SIMULATION RESULTS

The interest in the simulation results is focused on the stability of the simulation. Therefore the main aspect of result processing is to define ranges of dynamic properties of the handled objects which fit to the simulated manipulator so that the complete system can be simulated without getting unstable. Figure 10 shows the eigenvalues of the system in the z -plane which is commonly used to display results of time discrete systems. For a stable simulation all eigenvalues have to be inside the unit circle. As a reference marked with stars the Figure shows the pairs of complex eigenvalues of the simulated manipulator itself and the coupled system manipulator-object as found for a time-continuous, dead time free simulation. The influence of the object in the system dynamics can be derived from the distance between the eigenvalues of the manipulator and the coupled system. Being marked with dots the eigenvalues of the coupled system are shown using a sampling rate of 50 Hz and a dead time

of 60 ms (40 ms dead time of the industrial robot, 20 ms for sensor data feed back). For each additional sampling time step of dead time a new eigenvalue (or a pair of complex eigenvalues) appears.

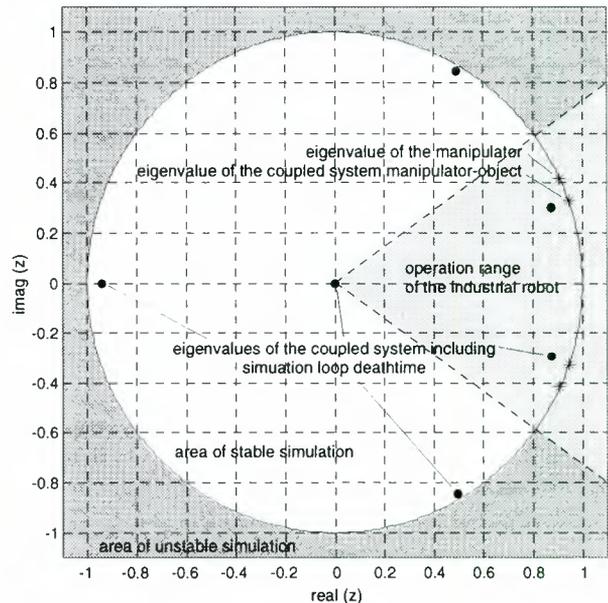


Figure 10: Eigenvalues using 50 Hz Sampling Rate

The corresponding results applying a sampling rate of 400 Hz with a dead time of 20 ms (17.5 ms dead time of the industrial robot, 2.5 ms for sensor data feedback) are shown in Figure 11.

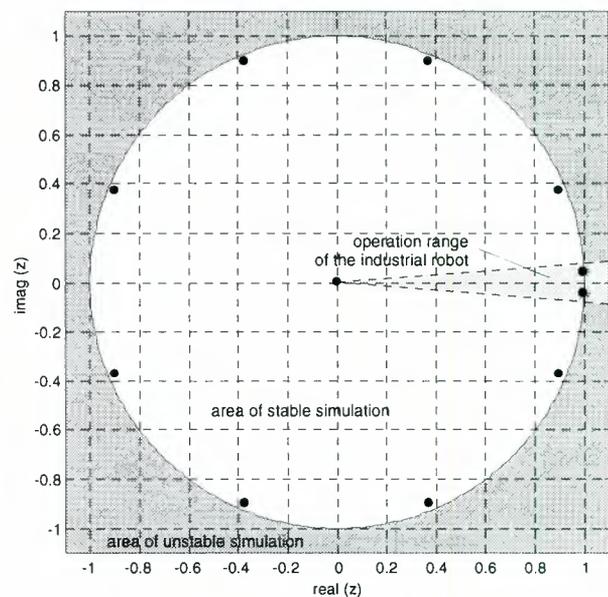


Figure 11: Eigenvalues using 400 Hz Sampling Rate

It was found that in both cases of sampling rates the simulation could be performed without instabilities. But it has to be considered, that these results are based on an extreme reduction of the model complexity and therefore are to be regarded with some uncertainties. A proper way to find a satisfactory operation range for a simulation is to vary the dynamic properties of the

handled object and to recalculate the eigenvalues. A way to display this kind of parameter variation in an easy to interpret manner is to show the absolute values of the eigenvalues ($abs(z)$) over, first, the ratio of eigenfrequency of the coupled system to the eigenfrequency of the manipulator (ω/ω_M), and second, over the non-dimensional damping rate of the coupled system. The system eigenfrequencies were varied either by changing the object elasticity or the object mass.

The vertical wall in Figure 12, Figure 14 and Figure 16 shows the operation limit (damped eigenfrequency of 5 Hz) of the industrial robot for an application as a physical motion generator inside the hybrid simulation. All systems which can be described by dynamic parameters being located on the left hand side of the limit wall could be displayed by the industrial robot.

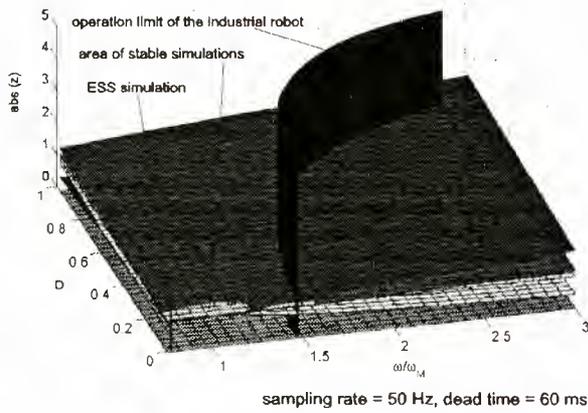


Figure 12: Area of Stable Simulations using a Sampling Rate of 50 Hz (Elasticity Variation)

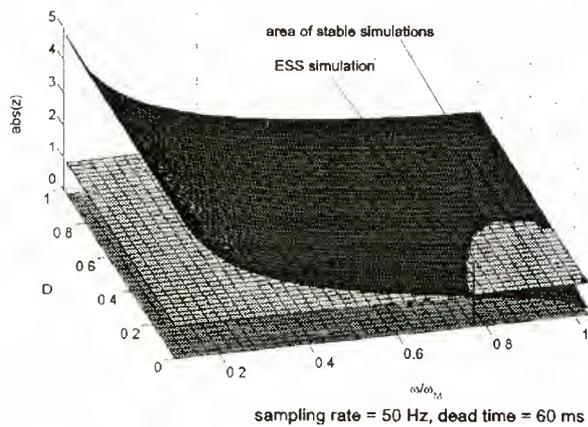


Figure 13: Area of Stable Simulations using a Sampling Rate of 50 Hz (Mass Variation)

Figure 12 and Figure 13 show that for a sampling rate of 50 Hz with a simulation facility depending dead time of 60 ms inside the simulation loop only a very small area of system dynamics variation is left to perform stable simulation within. Only there the plane

$abs(z) = 1$ is higher than all planes of the absolute values of the eigenvalues. It is just a good luck that the ESS-Simulation fits the dynamic properties to stay inside the valid area. Generally, it has to be remarked that in the operation mode mentioned above a closed loop hybrid simulation is not possible.

A much better situation is found in the second operation mode of the simulation facility (Figure 14 and Figure 15). In this case a sampling rate of 400 Hz is used in all parts of the simulation loop. The dead time within the loop is 20 ms. It is shown that in a wide range of dynamic system properties the simulation stays in a valid area. This area partly extends the frequency bandwidth that is able to be physically displayed by the industrial robot. However, it has to be noticed that for the example of the ESS scenario the simulation is also not far away from the limit of stability. Especially this is to be seen in Figure 15, where the object mass was varied to change the system eigenfrequencies.

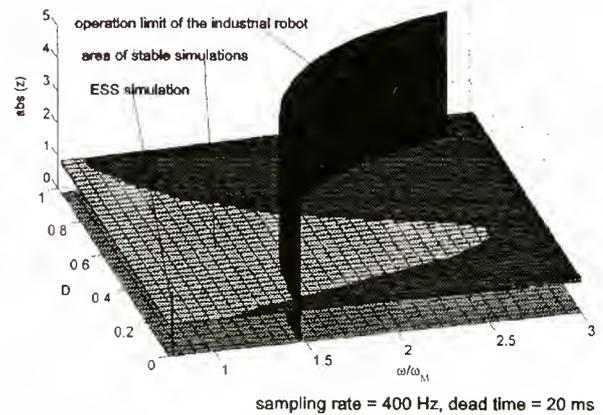


Figure 14: Area of Stable Simulations using a Sampling Rate of 400 Hz (Elasticity Variation)

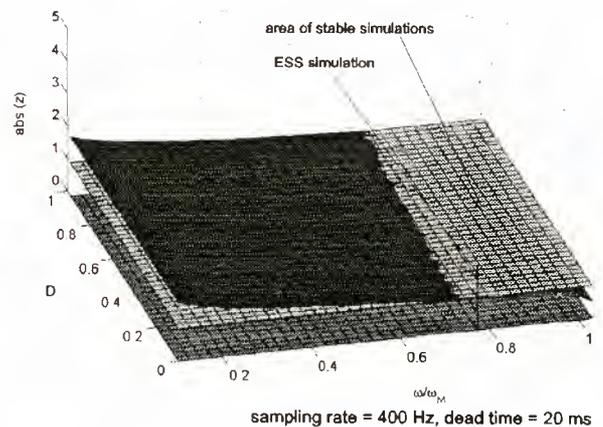


Figure 15: Area of Stable Simulations using a Sampling Rate of 400 Hz (Mass Variation)

A further aspect which has to be mentioned is the dead time inside the simulation loop. Mostly this time delay depends on hardware equipment of the simulation

facility. In this case the main reason for dead times in the loop is the control of the industrial robot. So reductions of dead times would require to impact into the robot's drives, its internal electronics and control strategies. But this cannot be performed during the development of simulation concepts.

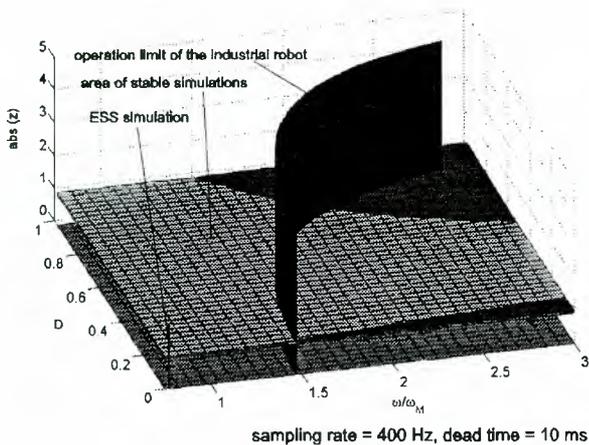


Figure 16: Area of Stable Simulations with Reduced Time Delays (Elasticity Variation)

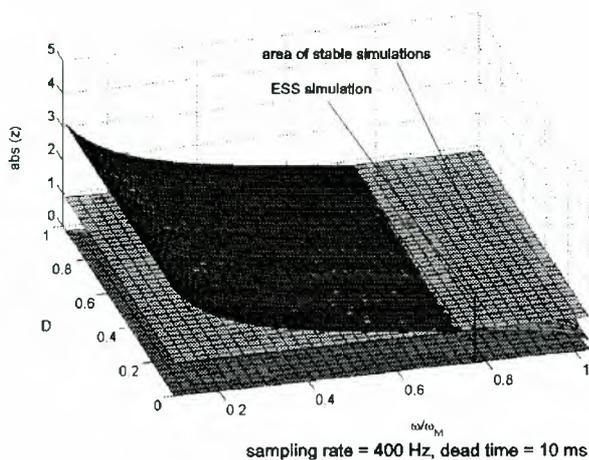


Figure 17: Area of Stable Simulations with Reduced Time Delays (Mass Variation)

However, theoretical investigations were performed to demonstrate the effect of reducing dead times in the loop. Figure 16 and Figure 17 show the results for half a dead time of the original one. The range of stable simulations can therefore be further extended but it has to be noticed, that not the complete operational range of the industrial robot is inside the valid area.

Assuming realistic values for sampling rates and dead times reachable for space robotics simulation the stable area defined above cannot be extended essentially. There are still limits concerning computer power and the dynamics of hardware equipment. If the hybrid simulations condition do not match the dynamic conditions for stable simulations, additional signal processing inside the simulation loop has to be performed. For the system example mentioned in this

paper, methods of software energy dissipation during the simulation run have to be developed. This could be performed by appropriately processing the sensor data fed back into the software simulation.

5. CONCLUSION

In this paper a method was presented to simulate space robotics operations. The simulation method is a hybrid one which combines the advantages of software simulations and hardware test set-ups. Due to the integration of an industrial robot it is possible to display simulated motions physically and to move tools, sensors and other hardware equipment. An important feature is the sensor data feedback into the software simulation to realize a closed loop simulation. However limitations in the sampling rate and dead times in the loop in some cases cause instabilities of the simulation. In opposite to off-line simulations, real-time conditions have to be maintained and a problem solution by a refinement of the simulation step size is not valid. The appearance of instabilities depends on the dynamic properties of the simulated system. Generally, an increasing sampling rate and a reduced dead time in the simulation loop extend the dynamic range of stable simulations. However, assuming realistic values for sampling rates and dead times there are still areas of instability wherein the hybrid simulation method is not suitable.

6. LITERATURE

- [1] B. Schäfer, R. Krenn; *Force/Torque Sensor Feedback Measuring Contact Dynamics in Space Manipulator Hybrid Simulation*; AIAA Guidance, Navigation and Control Conference, Portland, Oregon, USA, 9-11 August 1999.
- [2] B. Schäfer, M. Lösch, K. Landzettel; *Simulation of Manipulator Deployment and Satellite Capturing Dynamics*; Astra Workshop, ESTEC, Noordwijk, The Netherlands, 1-3 December 1998.
- [3] R. Krenn, B. Schäfer; *Instabilitäten in Simulationen bei Rückführung von Kraft-Momenten-Signalen*; GAMM annual meeting, University of Bremen, Germany 6-9 April 1998.
- [4] R. Krenn, M. Lösch, B. Schäfer; *Industrieroboter als flexibles Werkzeug in der Echtzeit-Simulation von Raumfahrtmanipulatoren*; VDI Berichte 1315 (1997), Page 217-231.

JERRY - A System for the Automatic Generation and Execution of Plans for Robotic Devices: The Case Study of the SPIDER Arm *

A. Cesta, P. Riccucci, IP-CNR, National Research Council, Rome, Italy
 M. Daniele, P. Traverso, IRST, Trento, Italy
 E. Giunchiglia, M. Piaggio, DIST, University of Genoa, Italy
 M. Schaerf, DIS, University of Rome "La Sapienza", Italy

Abstract

This paper describes JERRY, a system which supports the interactive design, planning, control and supervision of the operations of autonomous systems in a space environment. The aim of JERRY is to provide a high level of autonomy still retaining the possibility for the user to monitor, control and override potentially autonomous operations in a flexible way. JERRY is composed by a set of tightly integrated specialized sub-systems, which have been designed to perform effectively and efficiently their specific tasks, and, at the same time, to be open to the interaction with the user and among each other. This results in a system with a potential high level degree of autonomy, but which can still be controlled and guided through interaction.

JERRY's architecture and underlying ideas have been tested and made operational for monitoring and controlling a SPIDER robotic arm operating in an indoor environment very close to the payload tutor experiment described in [5].

1 Introduction

The recent development of space autonomy results in a set of novel problems related to the integrated work of human beings and robotic devices in space missions. From the one side, the increasing complexity of the services requested to robotic devices results in a need for more and more sophisticated and autonomous systems. From the other side, a relevant aspect of space missions involving humans is their possibility to maintain a level of control over autonomous robotic devices (see for example [7; 8]). This is due to quite a number of factors, such as the possibility of a completely unexpected event

that may require the humans to override the robotic autonomy; or the long run psychological effects for the human of living in an environment in which everything is "externally directed and operated".

In this paper we describe JERRY, a system which supports the interactive design, planning, control and supervision of the operations of autonomous systems in a space environment. The aim of JERRY is to provide a high level of autonomy still retaining the possibility for the user to monitor, control and override potentially autonomous operations in a flexible way. JERRY is composed by a set of tightly integrated specialized sub-systems, which have been designed to perform effectively and efficiently their specific tasks, and, at the same time, to be open to the interaction among each other. This results in a system with a potential high level degree of autonomy, but which can still be controlled and guided through interaction. The set of tools provided by JERRY, all together, have been developed to provide the following main features:

Modularity: different kinds of tasks, which are intrinsically complex and require special purpose capabilities, are handled by independent and highly specialized sub-systems.

Autonomy: the system is able to carry out tasks without a continuous and detailed user supervision, by enabling the specialized sub-systems to exchange data autonomously and to perform their own tasks automatically.

Interactivity: the system provides the user with the ability to inspect and direct every step of a system operation, via specialized sub-systems designed as "open systems" which can satisfy different kinds of user's requests.

Flexibility: the system can be reconfigured to fit several different robotic systems and environments of interest, by allowing for the possibility to flexibly specify the application domain, to require different kinds of services to the specialized subsystems, and to exchange data with different modalities.

Adaptability: the system can be adapted to work

* Corresponding author: Amedeo Cesta, IP-CNR, Viale Marx 15, I-00137 Rome, Italy, fax: +39-06-824737, e-mail: cesta@ip.rm.cnr.it.

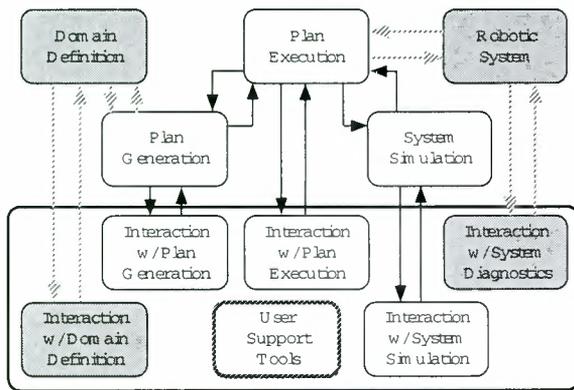


Figure 1: Structure of the System

at various levels of task specification detail and can support different user expertise.

JERRY is composed by four inter-connected modules, called the User-System Interaction module (developed at IP-CNR), the Planning module (IRST), the Execution module (DIST) and the Simulator Module (DIS). The Planning, Execution and Simulator modules are highly autonomous but open sub-systems, which can work at various levels of interaction. The planning module generates high level plans of actions to be executed; the execution module translates them into lower level programs and monitors executions; the simulator module provides a graphical and interactive simulation environment.

A main characteristic of JERRY is the provision of a flexible interface, through the User-System Interaction Module, which allows for different levels of interaction. It allows to access data and control the behavior of highly automatic systems by providing either high level specifications of what has to be achieved or detailed constraints on how the task should be performed. For instance, the user can request the planning module to generate automatically a high-level plan which achieves a high-level specified goal, or can direct the planner by imposing constraints on how to generate the plan. Analogously, the user can request the execution module to generate automatically the low level program corresponding to a plan, or can direct the execution module by imposing constraints on how the low-level robotic plan has to be generated. Finally, the user can directly monitor the execution of the program by looking at the simulation, or can directly interact with the simulator and specify the final destination to be reached. Such a high-level degree of interactivity between the user and the robotic device has been obtained via a "client-server configuration" in which the User-System Interaction Module is central to the system and can request different services from the other modules.

The structure of the system is represented in Fig-

ure 1. In this figure, the planning, execution and simulator modules are visible in the top part, while the interaction module (with the sub-modules acting as interfaces with one of the other modules) is the "big box" at the bottom. The "Domain Definition" box represents a module that allows the user to specify the domain considered, and is currently part of the simulator. The "Robotic System" box represents the real robotic device. The solid arrows represent a flow of information, while the dotted arrows represent a still missing connection. For example, the dashed arrow between "Domain Definition" box and the interface, means that currently the user can specify a domain not through the interface, but only interacting directly with this module.

Finally, JERRY has been developed as part of an ongoing and more ambitious project funded by ASI, the Italian Space Agency. In this application, JERRY provides its functionality to different kinds of users which have to design, control and monitor a SPIDER Robot Arm performing quite complex tasks, e.g., the set up of several kinds of experiments in a space workcell. Even though the project is still running, a first prototype is already working and available for experimentation. The prototype produces plans for problem in a scenario which is quite close to the payload tutor experiment described in [5]. In this scenario, e.g., the SPIDER arm is supposed to extract a tray from a shelf, fix it to one out of two tables and then automatically perform experiments moving objects contained in the tray. As far as the whole project is concerned, the functionalities of the whole system will be those of JERRY, integrated by the services provided by a module for diagnosis [9], a module for the visual interpretation of arm's activities [1], and a module (see [4]) responsible for supervising the arm in a outdoor environment similar to that described in [6]. See the corresponding papers (in this volume) for more information on any of these additional modules.

In this paper, we first provide a global overview of JERRY by describing its high level architecture (Section 2). We then describe the main features of each subsystem: the user interface (Section 3), the planning module (Section 4), the execution module (Section 5), and the simulation module (Section 6). Some conclusions end the paper.

2 JERRY's Architecture

JERRY can work at two levels of interactions that are targeted to two typical users of space robotic devices: the "programmer-level" contains functionalities offered to the robotic system operator; the "user-level" deals with activities performed by on-ground scientists or payload operators. At the programmer-level, the user can program the behavior of the device using its typically low-level interface language,

e.g. the language (called PDL2) currently used to control the SPIDER arm. A typical PDL2 instruction is "MOVE LINEAR TO point-in-space", where point-in-space is a 6-tuple of real values. This level of interaction is adequate for an experienced user. Nevertheless, programming complex tasks at this level may be very difficult for a user which has no experience with the programming language, e.g. PDL2. Moreover, low-level programs can be hard to maintain and re-use. For this reason, interaction at the user-level provides also non experts (e.g. scientists) with the ability to specify robotic tasks. Such users do not need any knowledge of the underlying physical structure of the robotic device (e.g. of the degrees of freedom of the arm) or of the physical scenario (e.g. of the exact position in space of the objects). A typical high-level instruction is "GET OBJECT object-name".

Operationally, the two interaction levels reflect two working modalities:

user-drives-system-supervises: in this modality an expert, knowledgeable of the underlying robotic device and mission interacts with the system by describing the mission in the robotic device interface language. The mission is encoded as a low level plan which is directly executable by the execution module.

system-drives-user-supervises: in this modality the user (even a non expert, e.g. a scientist) fixes the goal in a high level specification language. The high level specification cannot be executed directly. The system generates automatically executable low level programs. This is achieved in two steps. First, the planning module generates a set of high level actions which have to be executed in different situations and which are guaranteed to achieve the goal. Then the execution module, for each high level action, generates a corresponding sequence of low-level actions in the robotic device interface language (e.g. PDL2). Independently from how the low-level plan is generated, the execution module is responsible for its execution, and for the monitoring of the behavior of the robotic system. At each step of the execution process, the user can be prompted for validating the high-level action to be executed, or, if required, the current low-level program.

The resulting architecture is highly modular and configurable: the system can be configured to work at different levels of automation (e.g. depending on the activity performed by the planning module) and the user has the possibility to flexibly access data manipulated at different levels of detail (e.g. data at the execution or at the planning level). The interface can be set to be used by users with different experience (programmers or scientists) and can also be adapted

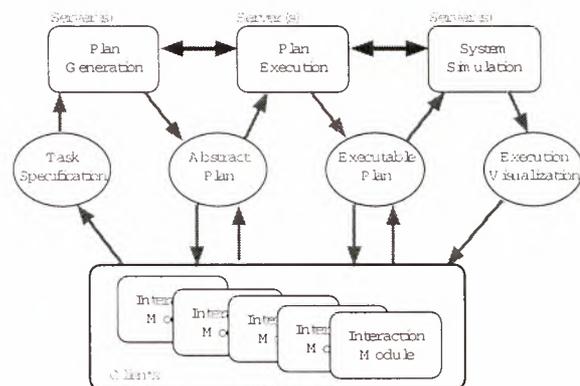


Figure 2: JERRY's Current Architecture

to different input devices (e.g., driven entirely from mouse or touchpad, entirely from keyboard, or, possibly, from custom input devices).

A first version of the demonstrator has been fully implemented, is available for inspection, and is currently under development to improve its general performance and to enrich the services offered to the user. This demonstrator (whose architecture is represented in Figure 2) is based on a client/server architecture in which a client interface service is able to continuously interact with the planning, execution and simulator modules. This has involved the development of specialized protocols that allow each interaction module to safely exchange data with the three servers through point-to-point communication. Current protocols are deliberately designed to be very simple to minimize the overhead of communication between modules and to quickly arrive to a first integration.

3 Interaction Module

The role of software systems like JERRY is to allow different users to employ complex robotic devices while preserving the levels of responsibility that users have in their working contexts. Both the user-level and the programmer-level preserve the usual working activity, but offer a number of additional functionalities that allow the users to focus on strategic and decisional tasks and to delegate repetitive or very difficult tasks to the interactive planning software.

The JERRY Interaction Module consists of a Graphical User Interface endowed with the following functionalities:

- Task oriented help.
- Problem specification targeted to the planner domain representation language.
- Inspection of high-level plans: a rather simple representation of the plan returned by the planner is shown and the possibility of inspecting the representation of single plan states is given.

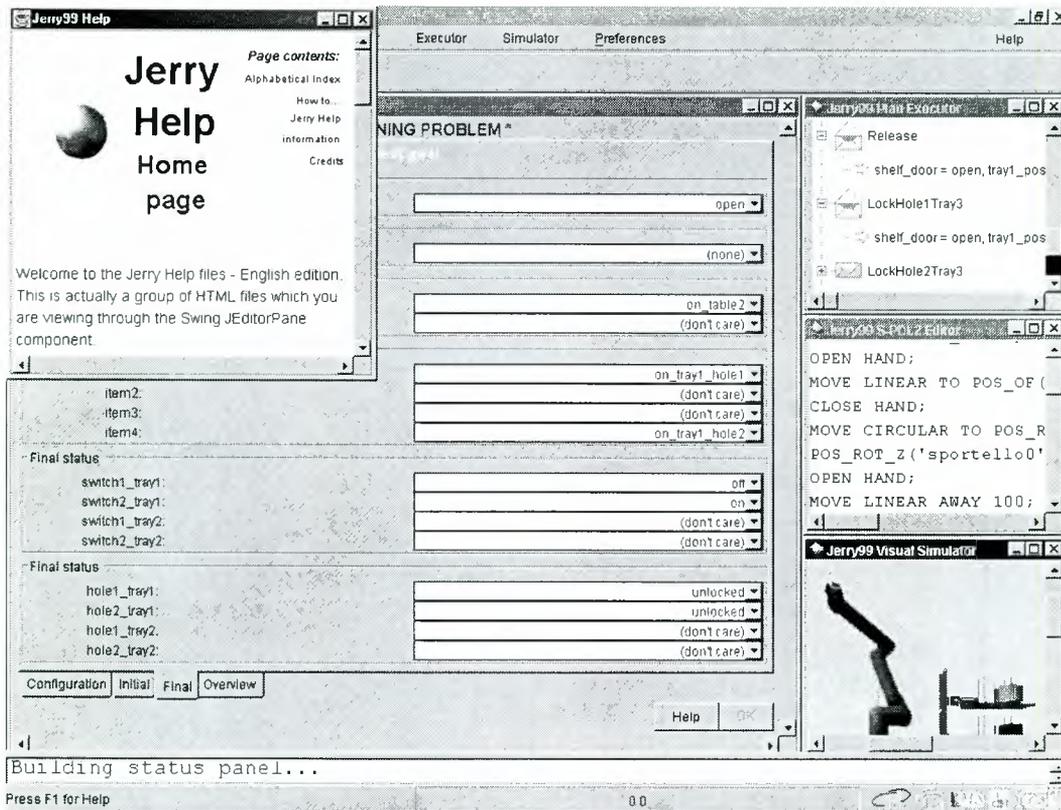


Figure 3: JERRY Interactive Module

- Inspection of plan compilation: the low-level code produced by the plan compilation and execution module is shown to the user.
- Robotic device simulator visualization.

The current look of the Interactive Module is shown in Figure 3. In the Figure we can see (i) the Help window (top-left) that is designed as a separate entity; (ii) the planning problem specification window (main window below the Help window); (iii) the plan current in execution (top-right); (iv) the PDL2 code corresponding to the action being executed (middle-right); and (v) the execution of the plan coming from the simulator (bottom-right). The size of the 4 windows corresponding to point from (ii) to (v) are interconnected and vary according to the user current focus of attention that is always contained in the main window.

According to the subdivision made between the “programmer-level” user and the “scientist-level” user, the tasks allowed to each level have been defined. In the current implementation of the “user-level interaction” the users can: (i) get acquainted with an operating environment; (ii) define specific parameters of the scenario (e.g., decide the number of trays in an experiment); (iii) specify the goal he want to achieve and the constraints to satisfy in achieving it; (iv) ask the planning module to determine the set of actions (the plan) that achieves the

goal; (v) display and comment the resulting plan; (vi) activate plan execution. Special attention has been dedicated to automatically checking the consistency of commands selected by the user and in offering explanation facilities for non-expert users. The “programmer-level interaction” offers: (i) the possibility of creating robot programs directly using the robot language, (ii) the choice of having the planning and execution mechanisms that work as background help of the programmer; (iii) the possibility of experimenting different operational situation offering a choice among alternative input modalities. The possibility of customizing the interaction modality is relevant for experimenting on-flight use of the programming ability. In is worth observing that being the Interaction Module configured as a client it is possible to serve multiple users at the same time each of them interacting with personalized functionalities.

An implementation in Java (compatible with JDK 1.2) has been realized and is currently tested for improvements.

4 Plan Generation

The Planning Module developed on top of the MBP system (Model Based Planner) [2; 3], receives in input from the Interaction Module a high level specification of the task to be performed (called goal).

The goal is a high level description of what has to be achieved. It does not detail how the task should be performed. The Planning Module generates automatically a plan of actions which achieves the task specified by the goal. The plan of actions is the output which can be passed, through the Interaction Module and possibly under control of the user, to the Execution Module. A typical plan synthesized by MBP looks like the following:

```
Get object Y.
if this action succeeds,
  then put Y on experiment tray Z,
  otherwise get object Y1;
...
```

Both the goal (the high level specification of the task to be performed) and the plan of actions (the sequence of operations to be executed to achieve the goal) can be specified and inspected by the user interface. The user-level specification of the problem is translated into the representation language of the planning module. The planning module returns to the user-interface a representation of the plan which associates to each operation in the plan a description of the situation (the state) which should be reached after executing the operation.

A main characteristic of the planning module is that it is an open system, i.e. each of its operations (e.g. plan search) can be inspected, controlled and guided by the user. This fact opens up the possibility to provide a planning functionality which supports a "user-centered operation mode" for JERRY, in which the planner interacts flexibly with the user interface module. The user, beyond asking for a goal to be satisfied, can ask the planner for different services, e.g. show all the plans which satisfy a goal, select one of them, query the planner about the possible effects of the execution of plans, re-use existing plans, ask the planner to validate a user defined plan, inhibit some plans, query the planner about the current state of the execution in terms of high-level actions. This "user-centered" modality requires a design of the planning module which is different in philosophy wrt current state of the art planners. The planner is no longer the automatic generator of solutions, it becomes a system which exploits its automatic generation capabilities to support the user to find the right solution and is flexible enough to adjust its plan generation activity to different user requirements.

Another characteristic of the planning module is that MBP returns "safe plans", i.e. plans which are guaranteed to achieve the goal in spite of non-determinism. For example, MBP is able to find a safe plan (assuming that one such a plan exists) even in the case in which some actions may fail (e.g. because of some malfunctioning of the devices) or in the case some action is no longer executable (e.g. because

some of the actuators is broken).

For efficiency reasons, the planning module has been written in C. To improve its portability, the standard ANSI has been followed.

5 Plan Compilation/Execution

The Plan Compilation/Execution module is responsible for transforming a high-level, user-oriented abstract plan into a sequence of low-level, machine-oriented execution plan. In more detail, the Plan Compilation/Execution module receives in input from the interface an arbitrarily long sequence of actions to be performed, and generates a sequence of actions (a "program") that the robot can directly execute. For example, in the case of a robotic arm, the program corresponding to a *move(o,l)* ("move object *o* to location *l*") looks like the following sequence of instructions

```
move_near <pos_o> by 800;
open_hand;
move_linear <pos_o>;
close_hand;
move_near <pos_l> BY 800;
move_linear <pos_l>;
open_hand;
move_away 1200;
```

where *<pos_o>* and *<pos_l>* are six tuples of real numbers specifying the positions of the object and of the location respectively.

In any case, the sequence of actions given to the execution module does not need to correspond to a complete plan. Instead, the user can (i) break a plan as given by the planning module into blocks of planning actions, (ii) require the compilation of all or some of the blocks, (iii) validate the execution program corresponding to a program, or (iv) ask for an execution program differing from the proposed one.

As for the planning module, the execution module is an open system in which the parameters affecting its behavior (e.g. the availability of a given low-level action) can be inspected, controlled and eventually modified by the user. For example, the user can inhibit the execution module from using a certain low-level action because it involves some dangerous or unavailable move for some joint. As above, this fact opens up the possibility to provide a "user-centered operation mode" for JERRY, in which the execution module interacts flexibly with the user interface module.

A Java (compatible with JDK 1.2) implementation of the execution module has been realized, and is currently tested for improvements.

6 Robotic System Simulator

The simulator allows for a 3D representation of a robotic arm in a given working environment. The

simulator is composed by three parts:

- a user interface which allows the user to examine the scene and to interact with it by suitable commands;
- an interface that allows the user to define all the objects in the scenario;
- and interface that allows the user to define the robot employed to manipulate the objects in the scene.

Currently, the simulator has been specialized with knowledge of two domains: the first is close to the external robotic experiment described in [6]; the second domain resembles the internal payload tending described in [5]. In both cases, the robotic device is the SPIDER arm.

About the user interface, the operator may:

- observe the evolution of the scene on a screen, both by looking at the arm's movements and information of the specific values of the various variables controlling the arm,
- interact with the robot, e.g. by specifying a position to be reached, or
- control the robot, by writing a PDL2 program which can be executed.

Finally, the simulator has been written using the Java language, the Java 3D library, while some VRML files specify the geometry of the objects in the scenario. The simulator is therefore a Java application that does not depend on the particular external browser used.

7 Conclusions

This paper describes JERRY, a system for the automatic generation and execution of plans for robotic devices, and briefly reports about the case study of the SPIDER arm. The main feature of the system is the high-level of interaction that the user can decide to have with the system. This level of interaction is critical in the context of spatial missions, where (i) unforeseen emergencies can happen, and (ii) still the mission has to proceed, possibly under the humans' supervision.

JERRY has been designed to be a flexible, open architecture. Care has been taken in order to distinguish the domain-dependent from the domain-independent tasks in order to minimize the customization efforts. JERRY's architecture and underlying ideas have been tested and made operational for monitoring and controlling a SPIDER robotic arm operating in an indoor environment very close to the payload tutor experiment described in [5].

Acknowledgments

This research is supported by ASI (Italian Space Agency) as part of the project "Un Sistema Inteligente per la Supervisione di Robot Autonomi nello Spazio". Special thanks to the other participants in the project for the many fruitful discussions and the technical help.

References

- [1] A. Chella, S. Gaglio, D. Guarino, and I. Infantino. An Artificial High-Level Vision Agent for the interpretation of the Operations of a Robotic Arm. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS99*, ESTEC, Noordwijk, NL, 1999.
- [2] A. Cimatti, E. Giunchiglia, F. Giunchiglia, and P. Traverso. Planning via Model Checking: A Decision Procedure for AR. In *Lecture Notes in Computer Science*, volume 1348, 1997.
- [3] A. Cimatti, M. Roveri, and P. Traverso. Automatic OBDD-based Generation of Universal Plans in Non-Deterministic Domains. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998.
- [4] A. Dell'Arciprete, G. Rossi, A. Finzi, F. Pirri, and M. Schaerf. A System Integrating High and Low Level Planning of Complex Tasks with a 3 Dimensional Visualizer. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS99*, ESTEC, Noordwijk, NL, 1999.
- [5] S. Di Pippo, G. Colombina, R. Boumans, and P. Putz. Future Potential Applications of Robotics for the International Space Station, Robotics and Autonomous Systems. *Robotics and Autonomous Systems*, 23:37-43, 1998.
- [6] F. Didot, J. Dettmann, S. Losito, D. Torfs, and G. Colombina. JERICO: A Demonstration of Autonomous Robotic Servicing on the MIR Space Station. *Robotics and Autonomous Systems*, 23:29-36, 1998.
- [7] G. Dorais, P. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost. Adjustable Autonomy for Human-Centered Autonomous Systems on Mars. In *Proc. Mars Society Conference*, 1998.
- [8] JERICO Project Description. ASI Internal Documentation., 1998.
- [9] L. Portinale, P. Torasso, and G. Correndo. Knowledge Representation and Reasoning for Fault Identification in a Space Robot Arm. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS99*, ESTEC, Noordwijk, NL, 1999.

AN ARTIFICIAL HIGH-LEVEL VISION AGENT FOR THE INTERPRETATION OF THE OPERATIONS OF A ROBOTIC ARM

Antonio Chella, Salvatore Gaglio, M. Donatella Guarino, Ignazio Infantino*

Dipartimento di Ingegneria Elettrica, Università di Palermo and
Centro di Studi sulle Reti di Elaboratori, CNR
Viale delle Scienze, I-90128 Palermo, Italy,
phone: +39.091.238111, fax: +39.091.6529124, e-mail: chella@unipa.it

* Dipartimento di Ingegneria Elettrica, Università di Palermo
Viale delle Scienze, I-90128 Palermo, Italy,
phone: +39.091.6566273, fax: +39.091.488452

ABSTRACT

We describe an artificial high-level vision agent for the symbolic and graphic interpretation of data coming from a video camera that acquires the image sequences of the SPIDER robot arm of the EUROPA system during its operations. The agent generates the perception grounded predicates obtained by image sequences and it provides a 3D estimation of the arm movements, thus allowing the scientist user of SPIDER to receive meaningful feedback of his operations on the arm during a scientific experiment.

1. INTRODUCTION

We describe an artificial high-level vision agent for the interpretation of data coming from a video camera that acquires the image sequences of the SPIDER robot arm of the EUROPA system [7,11,15] during its operations (see Fig. 1).

The described software module is related to the interpretation of sensory data in the framework of an ASI project aiming at the application of AI techniques to the design and realization of an effective and flexible system for the supervision of the SPIDER arm. The arm will work on board of the International Space Station (ISS) [8].

The framework project is an Italian three years research project [1,6,17] sponsored by the Italian Space Agency (ASI) involving AI researchers from the Universities of Rome, Turin, Genoa, Palermo, Parma, from the IP-CNR of Rome and from the IRST of Trento.

The main aim of the vision agent is the advancement of the state of art in the field of artificial vision for spatial robotics by introducing and integrating artificial vision

techniques that offer a unique opportunity for providing the SPIDER arm operations with effective greater degrees of autonomy [2,3].



Fig.1. The SPIDER arm of the EUROPA system.

The valuable capabilities of the vision agent are:

- to individuate and segment the SPIDER arm also in contrasted and irregular backgrounds;
- to perform a 3D estimation of the position of the arm by camera images;
- to interpret complex movements of the arm acquired by a camera in terms of symbolic descriptions.

The implemented computer vision agent is based on three main components:

- (i) the perception component;
- (ii) the scene description component;
- (iii) the visualization component.

In the following, Sect. 2 describes the perception component of the system, i.e., how the system perform the low-level image processing in order to individuate and segment the SPIDER arm. Sect. 3 describes the scene description component, in which the acquired image is interpreted both in terms of 3D parameters and in terms of generated symbolic assertions. Sect. 4 describes the visualization component, in which the user may interact with the agent components, and Sect. 5 describes the implementation details of the system. Finally, Sect. 6 outlines some conclusions and future developments.

2. THE PERCEPTION COMPONENT

The perception component of the agent processes the image data coming from a video camera that acquires the operations of the SPIDER arm.

The main task of this component is to estimate the positions of the arm in the acquired image. It should be noted that the estimation, which is generated solely by the visual data, may be useful also for fault identifications of the position sensors placed on the joints of the arm.

The images acquired by the camera are processed by the contour module that extracts the arm contours by a suitable algorithm based on *snakes* [5,9,12].

The snake is a deformable curve that moves in the image under the influence of forces related to the local distribution of the gray levels. When the snake reaches an object contour, it is adapted to its shape. In this way it is possible to extract the object shape of the image view.

The snake as an open or closed contour is described in a parametric form by:

$$v(s) = (x(s), y(s))$$

where $x(s)$, $y(s)$ are x,y co-ordinates along the contour and s is the normalized arc length:

$$s \in [0,1]$$

The snake model adopted is based on circles and squares, in order to better extract the arm components (see Fig. 2). The snake model defines the energy of a contour, named the snake energy, E_{snake} to be:

$$E_{snake}(v(s)) = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s))) ds$$

The energy integral is a functional since its independent variable is a function.

The internal energy, E_{int} is formed from a Tikhonov stabilizer and is defined:

$$E_{int}(v(s)) = a(s) \left| \frac{dv(s)}{ds} \right|^2 + b(s) \left| \frac{d^2v(s)}{ds^2} \right|^2$$

where $||$ is the Euclidean norm.

The first order continuity term, weighted by $a(s)$, makes the contours behave elastically, whilst the second order curvature term, weighted by $b(s)$, makes it resistant to bending. For example, setting $b(s) = 0$ at point s , allows the snake to become second-order discontinuous at point and develop a corner.

The image functional determines the features which will have a low image energy and hence the features that attract the contours. In general this functional made up of three terms:

$$E_{image} = w_{line} T_{line} + w_{edge} E_{edge} + w_{term} E_{term}$$

where w denotes a weighting constant. Each of w and E correspond to lines, edges and termination respectively.

The snake used in this framework has only edge functional which attracts the snake to point at high gradient:

$$E_{image} = E_{edge} = - (G_{\sigma} * \nabla^2 I(x,y))^2$$



Fig. 2. Contour module extraction by the snake technique.

This is the image functional proposed by Kass [12]. It is a scale based edge operator that increases the locus of

attraction of energy minimum. G_σ is a Gaussian of standard deviation sigma which controls the smoothing process prior to edge operator. Minima of E_{edge} lies on zero-crossing of $G_\sigma * \nabla^2 I(x,y)$ which defines edges in Marr-Hildreth [9,10] theory.

Scale space filtering is employed, which allows the snake to come into equilibrium on a heavily filtered image, and then the level of filtering is reduced, increasing the locus of attraction of a minimum.

The implemented snake allows to extract the arm shape in a simple way and in short time. Fig. 2 shows the results of the contour module.

From the extracted arm snake it is possible to estimate the position of the links of the arm in the image plane, i.e., without the depth information, which is recovered by the *scene description component*.

Let us consider a generic link i of the arm at time t ; the link is characterized by its 3D coordinates:

$$(x_i(t), y_i(t), z_i(t))$$

A generic posture of the SPIDER arm at time t is characterized by the vector $\mathbf{x}(t)$ which individuates the seven links of the arm:

$$\mathbf{x}(t) = \begin{bmatrix} (x_1(t), y_1(t), z_1(t)) \\ (x_2(t), y_2(t), z_2(t)) \\ \vdots \\ (x_7(t), y_7(t), z_7(t)) \end{bmatrix}$$

The snake information allows us to estimate the first coordinates of each link, i.e., their projection in the image plane:

$$\mathbf{x}'(t) = \begin{bmatrix} (x_1(t), y_1(t), \dots) \\ (x_2(t), y_2(t), \dots) \\ \vdots \\ (x_7(t), y_7(t), \dots) \end{bmatrix}$$

3. THE SCENE DESCRIPTION COMPONENT

The scene description component receives as input the data coming from the perception component and it generates a symbolic description of the arm operations. This component is based on a self-organizing neural network with a suitable explicit representation of time sequences [4,14].

Each unit of the ARSOM is an autoregressive (AR) filter, able to classify and recognize variable inputs. The map auto-organizes during an unsupervised learning phase. Each unit of the map characterizes a sequence of movements of the SPIDER arm.

Let us consider a generic movement associated with the SPIDER arm. The movement is characterized by a sequence of n postures:

$$\mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-(n-1))$$

The AR model associated with this movement is:

$$\begin{aligned} \mathbf{x}(t+1) = & \mathbf{A}_0 \mathbf{x}(t) + \mathbf{A}_1 \mathbf{x}(t-1) + \dots \\ & \dots + \mathbf{A}_{n-1} \mathbf{x}(t-(n-1)) + \mathbf{e}(t) \end{aligned}$$

The order of the model is n , the $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{n-1}$ matrices are the weights of the model, and $\mathbf{e}(t)$ is the error matrix. Let us denote \mathbf{B} the global matrix related to the weight matrices:

$$\mathbf{B} = [\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{n-1}]^T$$

and with $\mathbf{X}(t)$ the global matrix related to the postures. We may write the previous equation in a more compact form:

$$\mathbf{x}(t+1) = \mathbf{X}^T(t) \mathbf{B} + \mathbf{e}(t)$$

The optimal weights matrices are found by minimizing the error matrix $\mathbf{e}(t)$. We have adopted the *alms* iterative method, that is:

$$\mathbf{B}' = \mathbf{B} + h_{ci} \mathbf{e}(t) \mathbf{X}(t)$$

where h_{ci} is the neighborhood kernel:

$$h_{ci} = \begin{cases} \frac{1}{2r^2} & \text{if } i \in N_c \\ 0 & \text{if } i \notin N_c \end{cases}$$

In this equation, r is a suitable parameter and N_c is the learning window.

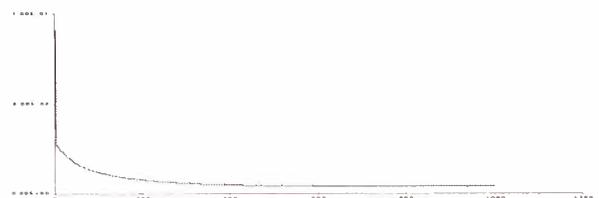


Fig.3. Error diagram vs training epochs.

The neural network, after a careful training phase, is able to classify the temporal sequences of movements of the arm into meaningful prototypical predicates.

Fig. 3 shows the diagram of the error of the neural network during the training phase. It should be noted that, after a few hundred learning steps, the error of the network is near zero value.

When the estimation of the coordinates of the link in the image plane are presented to the network:

$$\mathbf{x}'(t), \mathbf{x}'(t-1), \dots, \mathbf{x}'(t-(n-1))$$

the network is able to predict the full vector $\mathbf{x}(t+1)$, i.e., the vector with all the three coordinates of the posture of the arm links.

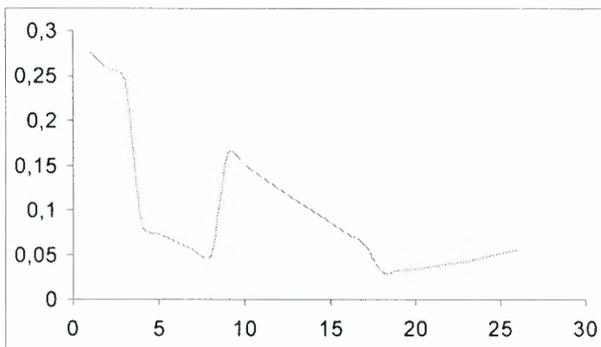


Fig. 4. Prediction error of the network.

Fig. 4 shows the prediction error of the network during its operations. It should be noted that the error, while is variable, it maintains in a reasonable limit.

Furthermore, the network is also able to perform a classification of the global arm movement and to present as output a symbolic predicate describing the movement itself.

Examples of the learned predicates describing the operations of the arm are: *Stretching_up*, *Stretching_down*, *Seizing*, *Grasping*.

The neural network approach presents the main advantage that it avoids an explicit description of the discrimination functions for the arm operations, as this function is learned during the training phase.

Furthermore, the neural network is robust with respect to the noise, as it is able to correctly classify the arm operations also when the movements estimations of some links are missing or corrupted.

In the operation tests performed, the network has been able to perform the 100% success on the classification task. To analyze the operation of the network, tests are performed on the recognition task when some links information is missed. Table 1 reports the obtained results. It should be noted that in the worst case, when the two links 1 and 3 are missing, the network is able to perform 51% of success recognition.

Missing links	Recognition %
0	100
1	75
2	74
3	62
1,3	51

Table 1. Recognition % with respect to the missing links.

4. THE VISUALIZATION COMPONENT

The scene description component receives as input the data coming from the perception component, in the same way of the scene description component, and it generates a graphic 3D representation of the arm movements.



Fig. 5. The visualization of the robot arm.

This component provides an immediate, visual feedback of the arm operations that complements the symbolic description coming from the previous component. The visualization component provides also the graphic interface for the whole agent.

Fig. 5 shows the results of the visualization component of the agent. The scientist user of the agent may view the arm operations from different point of views and he may navigate in the reconstructed environment.

He may also supervise and intervene in all the processing steps occurring in the agent itself: e.g., he may change the parameters of the perception component modules or he may tune the learning phase of the neural network in the scene description component.

The interface of the system presents several windows in order to provide the user scientist with a full control of the system.

The “camera” window shows the output image sequences of the video camera acquiring the real robot arm operations along with superimposition of the *snake* representing the output of the contour extraction module.

The 3D window shows the images representing the 3D reconstruction of the arm during its operations, and the “description” window shows the symbolic descriptions generated by the scene description component in terms of symbolic predicates.

A simple user interface based on buttons allows the scientist to modify the inner parameters of the agent in order to tailor the agent processing steps.



Fig. 6. The visualization environment of the vision agent.

5. IMPLEMENTATION DETAILS

The described artificial vision agent has been implemented in C under the Linux Operating System. The whole system currently runs on a Pentium II 400MHz and on a Apple iMac 266MHz.

The graphical interface has been realized by using the OpenGL [16] and the GLUT library [13].

6. CONCLUSIONS

The research demonstrated how the implemented artificial high-level vision agent may be an effective tool that helps the user scientist of the SPIDER arm to monitor his own operations by providing high-level feedback descriptions of the arm movements during the scientific experiments.

The described activity is aimed to the realization of a research product, which is innovative and complementary to the research activities of the ASI. It provides an effective scientific support with important effects of the development of new technologies within the ASI programs related to the utilization of the International Space Station.

The product of the described activity may be employed in all the fields in which the interactive autonomy of the space robotic systems is a mandatory requirement, as the exploration of the Moon and Mars.

The product of the present project will also give a valuable contribution to the use of the expensive and state of the art equipment related to space robotics owned by the ASI.

Of great importance are the possible industrial application of the product of the project. The system software could be employed in all the applications that require high automatic tasks in interactive autonomy, as the submarine robots and autonomous systems acting in nuclear plants.

ACKNOWLEDGEMENTS

Authors would like to thank Dr. V. Venturini of TecnoSpazio SpA and Dr. R. Mugnuolo of ASI for their kind collaboration on the described research.

This work has been supported by the ASI project “An Intelligent System for Supervision of Autonomous Robots in Space”.

REFERENCES

- [1] A. Cesta, M. Daniele, E. Giunchiglia, M. Piaggio, P. Riccucci, M. Schaerf, P. Traverso: JERRY: a system for the automatic generation and execution of plans for robotic devices -- the case study of the SPIDER arm, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).
- [2] A. Chella, M. Frixione, S. Gaglio: A Cognitive Architecture for Artificial Vision, *Artificial Intelligence*, 89, No. 1-2, pp. 73-111, 1997.
- [3] A. Chella, M. Frixione, S. Gaglio: An Architecture for Autonomous Agents Exploiting Conceptual Representations, Robotics and Autonomous Systems, 25, No. 3-4, pp. 231-240, 1998.
- [4] A. Chella, S. Gaglio, V. Mulia, G. Sajeve: An ASSOM Neural Network for Description of Autonomous Agent Actions, in: W. Gerstner, A. Germond, M. Hasler, J.D. Nicoud (eds.): *Artificial Neural Networks - ICANN '97*, pp. 799-804, Springer-Verlag, Berlin, 1997.
- [5] A. Chella, V. Di Gesù, I. Infantino, D. Intravaia, C. Valenti: A Cooperating Strategy for Object Recognition, in: R. Cipolla, D. Forsyth (eds.): *Proc. of the Int. Workshop on Shape, Contour and Grouping in Computer Vision (invited paper)*, Lecture Notes in Computer Science, Springer-Verlag, Berlin 1998.
- [6] A. Dell'Arciprete, G. Rossi, A. Finzi, F. Pirri, M. Schaerf: A system integrating high and low level planning of complex tasks with a 3 dimensional visualizer, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).
- [7] F. Didot, J. Dettmann, S. Losito, D. Torfs, G. Colombina: JERICO, Robotics and Autonomous Systems, 23, No. 1-2, pp. 29-36, 1998.
- [8] S. Di Pippo, G. Colombina, R. Boumans, P. Putz: Future Potential Applications of Robotics for the International Space Station, Robotics and Autonomous Systems, 23, No. 1-2, pp. 37-43, 1998.
- [9] O. Faugeras: *Three-Dimensional Computer Vision*, MIT Press, Cambridge, MA, 1993.
- [10] B.P.K. Horn: *Robot Vision*. MIT Press, Cambridge, MA. 1986.
- [11] JERICO System Description Tech. Rep. JER-005-10/97 ESA, November 1997, Noordwijk, The Netherlands.
- [12] M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active Contour Models, in: Proc. of First Intern. Conf. On Computer Vision, pp. 259-268, Springer-Verlag, Berlin, 1987.
- [13] M.J. Kilgard: *OpenGL Programming for the X Window System*, Addison-Wesley, Reading, MA, 1996.
- [14] J. Lampinen, E. Oja: Self-Organizing maps for Spatial and Temporal AR Models, Proc. of 6th Scandinavian Conf. On Image Analysis, Oulu, Finland, pp. 120-127, 1989.
- [15] R. Mugnolo, S. Di Pippo, P.G. Magnani, E. Re: The SPIDER Manipulation System (SMS) The Italian Approach to Space Automation, Robotics and Autonomous Systems, 23, No. 1-2, pp. 79-88, 1998.
- [16] J. Neider, T. Davis, M. Woo: *OpenGL Programming Guide*, Addison-Wesley, Reading, MA, 1996.
- [17] L. Portinale, P. Torasso, G. Correndo: Knowledge representation and reasoning for fault identification in a space robot arm, Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, Noordwijk, 1999 (this volume).

Smart, Simple and Low-Cost Control of planetary exploration rovers

Robin G.J. Biesbroek and Arne Y.J. Matthyssen

JAQAR Space Engineering
 Hector Berliozkade 68, 2551 XR Den Haag, The Netherlands
 Telephone: +31.70.3978665, Fax: +31.70.3978665
 General Information: info@jaqar.demon.nl
<http://www.jaqar.demon.nl>

1. ABSTRACT

This paper focuses on a control architecture for (micro)-rovers that can be used for planetary or small body exploration; an architecture of which the design is entirely based on the objective of creating a system that is robust, simple and cheap. This robustness and simplicity is achieved by effectively managing and allocating the control tasks to either the on-ground part or the on-board part of the system. Important features of this system are the level of rover autonomy, the use of classifier systems to generate commands (based on the knowledge of some basic rules, the rover and its environment), a contingency list. The system is foreseen to be implemented on a PC using off the shelf software. The concept incorporates human decision making and control together with a learning system and a rover with autonomous decision making capabilities.

2. INTRODUCTION

In the attempt to provide more human-like intelligence to the exploration rovers the stress most of the time lies on the enhancement of the "tele-presence" through virtual reality, stereo vision, sound, smell sensors etc. This paper presents a concept that gives the possibility to the system to learn from the decision of the operator and the interaction with the environment and change its decisions if necessary, as a human would do.

The concept was conceived with the knowledge of the reserved attitude of the planetary exploration society, regarding allowing too much artificial intelligence in expensive and complex missions. It is our feeling that the proposed concept does not allow any uncontrollable intelligence in the system that would block out human authority and provide additional risks for the mission.

3. CONTROL CONCEPT

The control concept is based on effectively managing and allocating the control tasks to either the ground segment or the rover segment of the system, together with the use of classifier systems [RD 1] to generate commands and incorporate the ability to learn.

Two different types of control are defined:

- human interactive
- autonomous control.

During human interactive control the operator commands the rover which executes the command as good as possible (although it was not the scope of this

work, the human interaction control mode can be an interactive autonomy mode, in which high level commands are send and the rover has some kind of autonomy, e.g. based on behaviours, to execute the command). This control-type is the normal operations mode and allows a man-in-the-loop at all times. The learning system based on classifier systems can be toggled on or off.

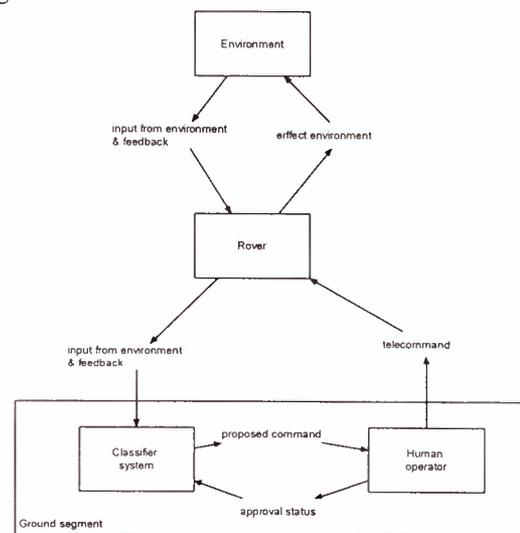


Figure 3.1: Control concept.

In the autonomous mode of operation the system is actually reduced to only a rover part. This phase only occurs when the human control has stopped e.g. due to unexpected loss of contact (LOC), in this case the rover becomes autonomous and uses the contingency list to diagnose the reason for the contingency, and take the appropriate actions to secure survival.

Note that autonomous control does not start immediately after LOC, but after a period of time (for example 1 hour); this allows the user or control centre a period of time to try and re-establish contact. This is incorporated using a trigger function built in the rover. Learning or not during autonomous operations is an issue that is briefly addressed in §5.

To provide the system with the proposed control several items are needed:

1. Simple commands
2. Classifier systems
3. Contingency list
4. Environment maps

3.1 Simple commands

The use of simple, parameter-less commands allow the use of classifier systems (see §3) to be implemented in the system. When stored in binary form, a set of simple TC's for the 4 legged PROLERO micro-rover [RD2] could look like:

Move front left leg		
Forward:	00000001	
Backward:	00000010	
Move front right leg		
Forward:	00000100	
Backward:	00001000	
Move back left leg		
Forward:	00010000	
Backward:	00100000	
Move back right leg		
Forward:	01000000	
Backward:	10000000	

These commands are of course not the commands one wants to sent to the rover. One of the objectives of the work was to evaluate the possibility to use classifier systems to generate based on these "leg" commands, more complex "steering" commands. For example:

Normal forward:	01010101 001
Normal backward	10101010 001
Climbing forward:	01010101 010
Climbing backward	10101010 010
Descending forward:	01010101 100
Descending backward	10101010 100
Turn left:	01100110 000
Turn right	10011001 000

as well as commands like "Turn over"; "Climb step"; "Take monitoring position" etc.

To get to the "steering" commands the system has to go through a learning phase, using the response of the operator to the proposed command (combination of the 8 leg commands), in order to come to a "steering" command that is appropriate in the current situation (defined by the telemetry (TM)).

If the system has enough time, the system allows, based on its "leg" & "steering" commands, the generation of new walking methods (see §5).

During non-learning mode the system will, depending on the telemetry and knowledge of the environment, propose one of these commands to the operator.

Apart from simple TC's, the classifier system requires input from the environment (telemetry (TM)) as simple string rules. If the sensor output can have many different values, the range will be divided into different classes e.g. low, medium, high in order to have a limited number of messages, that can be binary represented (see §3.4).

3.2 Classifier systems

[RD1]

Classifier systems give the system the possibility to propose commands and learn from the responds of the operator and the TM. A classifier system (CS) is a machine learning system that learns syntactically simple string rules, called classifiers which guide the system's performance in an arbitrary environment. The CS has developed out of the merging of expert systems and genetic algorithms.

Figure 3.1 shows the components of a CS and its interaction with the environment. The CS receives information about the environment, performs internal processing and then effects the environment. In learning mode, the CS uses feedback about the effect on the environment to learn from the experience. If no feedback is provided, the CS is in application mode (non-learning).

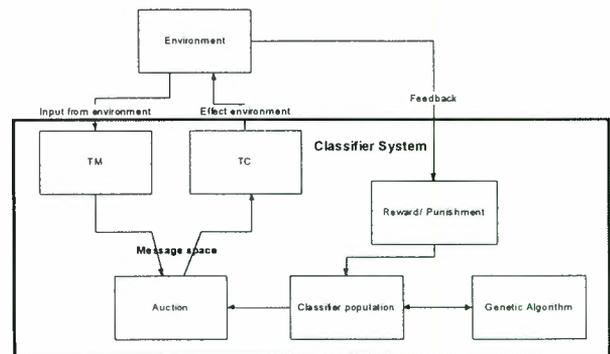


Figure 3.2: Classifier system components and interaction with the environment.

A classifier system has 3 major components:

1. Rule & message sub-system.
2. Apportionment of credit sub-system.
3. Classifier discovery mechanisms.

Detailed information on how a CS is used for rover control is given in chapter 4.

3.3 Contingency list

A contingency is a rover life threatening situation, e.g. sudden LOC, power too low, temperature too high/low etc. This means rover specific "housekeeping" parameters define the contingency. In this work the fact that e.g. the rover is turned upside down is not considered as a contingency, the system can react with its normal classifiers to this situation, or learn to get out. The contingency list is only used to detect when to go into autonomous mode. The list consists of several conditions and the appropriate action to be taken, needed for the rover to survive, in case of such a contingency condition.

3.4 Environment maps

To allow the system to propose a command which is not out of the blue, some knowledge of the environment which the rover will encounter has to be available a priori. To stay within the goal of the concept, (robust,

simple & cheap) no fancy environment mapping techniques can be used. The information about the environment can e.g. be represented as fuzzy maps [RD3]. Fuzzy map representation incorporates and allows handling of the lack of information about the environment and the in-accuracy that comes with it, and such maps do not require much disk-space to store.

The environment representation can be seen as follows. A map, divided in cells, consisting of useful parameters, e.g. temperature, height, slope, soil type, danger etc., can exist. The parameters will not consist of values, but binary re-presentable values like e.g.:

```
00000001 Danger zone
00000010 Hard ground
00000100 Medium-hard ground
00001000 Soft ground
00010000 Steep slope
00100000 Link status OK
01000000 Temperature low
10000000 Power low
etc.
```

Three types of maps can be distinguished “terrain maps”, “housekeeping maps” and “direction map”. Information like danger-zones, slopes, soil characteristics etc. are defined before operations, it can be envisaged that the rover sensors provide enough information during operations to update these “terrain maps”. “Housekeeping maps” are generated, during operations, on-ground using the rover TM (link status, power, internal temperature etc.), and regularly uploaded to the rover. This to update or replace its current maps, to insure that the rover in case of contingency has the most up to date maps, to find a safe place.

The “direction map” consists of the direction from each cell to the target position. This map can be generated manually by the human operator or force field can be envisaged.

In creating a command the system looks at the TM to be expected (included in the maps) for the next surrounding cells; going if possible to the next cell with the direction (direction map) pointing to the mission target.

4. GROUND CONTROL

The functionalities of the ground system are threefold:

- Creation and uploading of TC’s
- Storage of the rule-message system
- Receiving of TM

The creation of the TC’s is performed using a CS as described in the previous section. The TM is received in a simple string-format such as the example given in section 3.1, and is regarded as the ‘message’. The CS seeks the classifiers that match the message, and holds an auction: all the matching classifiers submit a bid, and

the classifier with the highest bid is proposed as a TC to the user.

1. Rule & message sub-system.

Each classifier consists of a rule in the form of:

```
IF (<condition1>&<condition2>&...&<conditionN>)
THEN (<action>)
```

where

<condition> is encoded as a string from the alphabet {0, 1, #}

<action> is encoded as a string from the alphabet {0, 1} and forms a TC.

The ‘#’ symbol acts as a ‘don’t care’ in the condition, matching either a 0 or 1, and allows for more general rules.

Each rule has an associated strength giving measure to the rule’s past performance in the environment in which it is learning.

The messages are generated from the environment: the TM in a simple string-format, and match the condition part of the classifier rule.

For example, the following classifier could exist (with ‘:’ denoting the break between the conditions and the action):

```
1#####01:10101010 001
```

which would, using the example codes of the previous section mean: if there is an obstacle in front of the rover (danger zone), and the rover is low on power, but the rover is not on a hard ground, then walk backwards normally. The classifier doesn’t care what inclination the rover has, or if the rover is on medium-hard ground or soft ground.

2. Apportionment of credit sub-system.

This sub-system deals with the modifications in strength of classifiers as the CS learns. These modifications occur via three mechanisms:

- Auction
- Reinforcement & punishment
- Taxation

When the CS receives messages from the environment, all the classifiers which match one or more of the messages compete, by submitting a ‘bid’ in an ‘auction’ to determine a victorious classifier that will effect the environment i.e. propose its TC to the user. The bid is a function of the classifier’s strength and specificity (number of non-‘#’ symbols). Only the bid of the victorious classifier is paid, and therefore the victorious classifier has its strength decreased by the amount of its winning bid.

The user will then approve or reject the proposed TC. If approved, the TC will be send to the rover. If the CS is

in 'learning' mode (allow feedback), the strength of the victorious classifier is increased when the TC is approved, or decreased when the TC is rejected. If the TC is rejected, the auction could be repeated until the user approves one. This has the advantage for the learning system to learn faster, since it receives more feedback, however it may take many proposals until the user approves a TC, if the TC the user has in mind is not obvious to the system. In that case, the user could choose the TC by hand. The CS could look up a matching classifier and regard it as a bid from that classifier, therefore increasing its strength.

Taxation occurs to prevent the classifiers from being cluttered with artificial high strength classifiers of little or no utility. Taxation is levied on each classifier per iteration (life tax) and on each classifier that submits a bid during an auction (bid tax).

3. Classifier discovery mechanisms.

The classifier discovery mechanisms consists of a genetic algorithm and a triggered cover detector operator.

A Genetic Algorithm (GA) [RD4] is an optimisation technique based on the mechanics of natural selection and genetics. GA's require the parameter set of the optimisation problem to be coded as a finite-length string containing elements (such as 0, 1, #). A population of individuals is created which goes through a process of evolution made up of the principles of combination (cross-overs: swapping chunks of elements between individuals), mutation (changing an element at random) and selection (creating new generations by selection individuals in proportion to their fitness, or 'strength').

The GA is applied after an epoch (of iterations), mating the classifiers and creating new ones. However, "steering" commands are not to be replaced. A solution to this problem is to divide the population of classifiers into a 'replaceable population' and a 'non-replaceable population'. The non-replaceable population is formed by the user or designer (possibly through learning, see §6), consists of classifiers made up by the user or designer and will not be replaced by the GA, although they can be selected for mating. The replaceable population is formed by the genetic algorithm, and is subject to change.

The triggered cover detector operator (TCDO) is activated whenever the CS does not have a classifier which matches a message. It responds by creating a new classifier that covers the message. The action is randomly copied from another action.

In order to keep the rover contingency control up to date with the (new) rule & message system, the entire classifier population should be uploaded after a number of TC's.

5. ON-BOARD ROVER CONTROL

In the normal interactive mode the rover is a simple slave of the human operator. During this interactive

mode of operations a command is given and the rover will execute it as good as possible and provide the operator with telemetry. It can be envisaged that the telemetry will be used on-ground to check the effectiveness of the last proposed (by the classifier system) command and so let the classifier system learn from its mistakes and/or achievements.

In a contingency phase the rover autonomy is activated. The activation is to be done by an on-board timer when an unexpected time of non-activity (no contact with the ground) is detected. This timer gives the operators on ground the time to solve the problems before the rover takes control. Of course if the problem is solved after the rover switched to autonomous mode the operator can regain control at all times.

Activation of the rover autonomy means activation of the classifier system with the contingency list and latest rules & messages, strengths and environment maps.

During contingency the rover priorities change. It is no longer its priority to reach the target defined by the operator, but it is to survive. The rover will now take into account its housekeeping data to decide what its next move will be. Depending on the contingency the rover will replace its target to a healthy environment, e.g. sudden LOC will provoke the robot to place the target on a cell in the area with positive link status.

The rover assesses what the problem is and proposes a command, no decision can be made, by a human, so the rover will execute the proposed command.

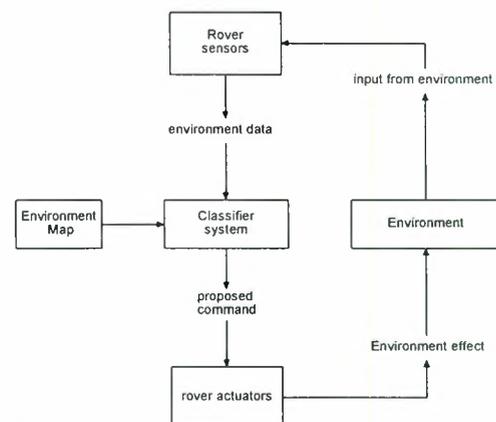


Figure 5.1: Rover autonomy

A choice has to be made to let the rover learn and change the classifiers or not. The classifier system gives the possibility to create new commands in response of TM. By evaluating its decision made earlier, using the current telemetry, the rover could change the strength and classifier rules.

If the rovers learning capacity is unrestricted during a contingency phase, the operator could, if he/she regained control, download the new classifiers to study the newly gained knowledge of the rover. The decision is now up to the operator to let the rover continue with

these classifiers or that the old ones (maybe changed by the operator) will be used.

It has to be studied if the autonomous generated classifiers do not import any uncontrollable rules in the system. For example the rover proposes: 00111010, what will the rover do, move his front left leg backward, move its front right backward, his back left back and forward at the same time and the back right leg does not move?

For this reason it is considered here that the rover does not learn during autonomous mode.

Investigations for a solution to allow generation of combination commands is ongoing, e.g. a first step can be to eliminate command combinations like "normal forward" (01010101 001) and "turn right" (10011001 000) that gives 11011101 001.

The classifier system and in-situ learning (taking into account the result of a command) however can allow the rover to use combinations of the "leg" commands (maybe together with the "steering") to generate new walking methods, e.g. the telemetry indicates the next cell is a slope with loose sand the system will after some re-occurrence of this TM combination generate not a climb command, but a different leg movement, resulting in a better command execution.

6. SIMULATION & RESULTS

The simulation that is done in the framework of this work simulates the learning process of the rover to walk. Starting from simple "leg" movements (see §3.1) and specific situations (defined by the TM) the system proposes a command. The operator evaluates the command and this result is used by the system to give a certain strength to a classifier (in order to get the correct command for that situation, i.e. the correct classifier), finally resulting in "steering" commands (see §3.1).

7. CONCLUSIONS & KEY FEATURES

It can be concluded that the clear separate control during the different modes of operations is simple, robust and safe. The autonomy of the rover is only used when normally the rover system would be lost anyway, so it poses no threat to the planned mission, but it gives extra possibilities to regain the mission.

The concept is flexible enough to allow or not to generate new classifiers when in autonomous mode, no conflict of human vs. rover intelligent system is expected.

The same smart system (classifiers, maps etc.) can be implemented on the rover as on ground.

With the use of the PROLERO rover as a testbed the problem was not approached from the simplest way, a simple 4 wheel rover would probably have made the commands simpler, but this relative complexity in possible commands did not pose any major problems. Although more investigations in un-comprehensible command generation has to be performed.

The use of separate maps containing very limited TM in a binary form together with the concept of target and

directions in a map is in line with the goal of simple robust and cheap.

The key features of this paper are: smart & simple control architecture, command generation using classifier systems, learning systems, autonomous rovers.

8. FUTURE DEVELOPMENT

During the realisation of this concept JAQAR engineers discovered some major issues to be resolved, investigations are ongoing in the field of:

- Controlling the generation of non-comprehensible commands during autonomous operations .
- Using the command generation for path-planning and rover operation scheduling purposes.

REFERENCES

- [RD1] SPHINcsX, <http://www.stanford.edu/~buc/SPHINcsX>
- [RD2] Legged Locomotion For Planetary Exploration, MSc. Thesis 1st edition Delft University of technology, April 1996, Jasper T. Hillebrand
- [RD3] Advanced teleoperation techniques for planetary mobile robots based on fuzzy models, MSc. Thesis 1st edition Delft University of technology, August 1996, Arne Y.J. Matthyssen, and published as ESA WP-1900.
- [RD4] 'Evolution in a Microchip' A brief study into Genetic Algorithms; ESTEC ESA-WP 1870, , 1996, Robin Biesbroek.
- [RD5] Genetic algorithms in search, optimization and machine learning, Goldberg, Addison-Wesley, 1989.
- [RD6] Walking Robots for Planetary Exploration Missions, Martin-Alvares, De Peuter, Hillebrand, Putz, Matthyssen, De Weerd, ISRAM, Montpellier, 1996.

The Jumping Tortoise: A Robot Design for Locomotion on Micro Gravity Surface

Kazuya Yoshida

Department of Aeronautics and Space Engineering, Tohoku University
Aoba 01, Sendai 980-8579, Japan
yoshida@astro.mech.tohoku.ac.jp

Abstract

This paper propose a novel design for a possible micro robot for exploration of a small object such as an asteroid. Micro robots are discussed and developed for the MUSES-C asteroid mission in JPL as Nano Rover and ISAS as Minerva. Both designs are nice and interesting, however we propose an alternative smart design. The proposed design uses four sets of 1-DOF legs to hit and jump like an flea and a specially designed neck to turn over like a tortoise. The design is promising for limited weight and power budget and the motion performance is evaluated by computer simulations with a proper dynamic model.

1 Introduction

This paper propose a novel design for a possible micro robot for exploration of a small object such as an asteroid. Micro robots are discussed and developed for the MUSES-C asteroid mission in JPL as Nano Rover and ISAS as Minerva. Both designs are nice and interesting, however we propose an alternative smart design.

A schematic illustration of the NANO Rover is depicted in Figure 1 (a). Nano Rover uses wheel system mounted on a swingable leg. However a wheel may not work on the micro-gravity surface. The traction force of wheel T is given by $T = \mu N$ where μ is friction coefficient and N is the normal force, which is usually equal to mg on Earth but here g is almost zero.

A schematic illustration of Minerva is depicted in Figure 1 (b). Minerva uses reaction torque generated by a reaction wheel inside the body to turn over the surface. If the body is sphere it may not move. High friction with the surface is essential for it to move. And it may be difficult for miniaturization because the reaction torque is inertia dependent and if the length becomes one-tenth the inertia becomes one-thousandth.

Here, we propose an alternative smart design that will not meet the above problems, and be promising for limited weight and power budget, The motion performance is evaluated by computer simulations with a proper dynamic model.

2 Robot Design

As the design criteria, we assume the followings: the dimension is almost 0.1 meter cube or less, weight less than 0.5 [kg], the gravity $g = 0.01[m/s^2]$. Those criteria are the same as the NANO Rover and Minerva.

The propose design of our rover, the Jumping Tortoise, is depicted in Figure 2. The design uses four sets of 1-DOF legs to hit and jump like an flea and a specially designed neck to turn over like a tortoise. A comb on the neck works as an antenna, and the neck motion provides a camera-pan function as well in upright position.

By controlling the phase and torque of four legs, the direction of the jump and the orientation of the rover may be controlled. A visual camera is considered a primary mission, but other sensors can be mounted if they are small and light enough.

The specification and weight budget are listed in Table 1 and 2. Expected motion of the robot are depicted in Figures 3 and 4.

3 Leg Mechanics

The leg mechanics is depicted in Figure 5.

Let us denote mass by m , length of the leg L , and the gravity acceleration on the surface g . When torque τ is applied to turn the leg, the force to put the ground F is expressed by

$$F = \tau/L.$$

The normal force of the contact point N is summa-

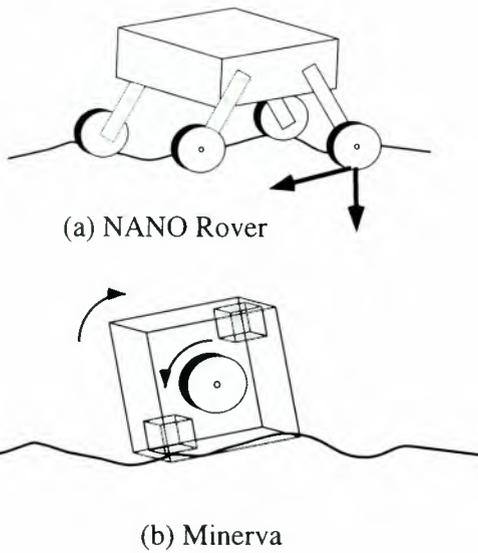


Figure 1: Different designs of rovers for MUSES-C asteroid mission

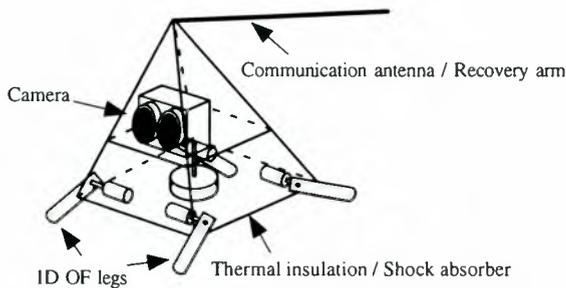


Figure 2: Proposing Robot Design

tion of mg and the vertical component of F .

$$N = mg + F \cos \theta$$

The thrust force in the horizontal direction T is generated by

$$T = F \sin \theta$$

when no slip, or

$$T = \mu(mg + F \cos \theta)$$

when the Coulomb's friction model is effective.

The above equation suggests that the trust force T is not zero even if the gravity is completely zero.

4 Friction Mechanics

Let us model the ground by a visco-elastic material with the stiffness K_w and the dumping D_w . The contact point is represented by (x, y, z) , where z is normal to the surface and x and y are tangent of

Table 1:Rover Specification

dimension	120[mm]×120[mm]×90[mm]
mass	533[g]
power generation	solar cells (effective 324[cm ²])
power consumption	max. 2880[mW]

Table 2: Weight Budget

group	part	mass [g]
structure	panels and frames	158
	legs	4
	motors · gears	49
	encoders	25
	motor control circuits	50
control	computer and controllers	60
	Sun sensor	10
	wire harness	35
power	solar cells	20
	control unit	10
communication	transmitter/reciver	50
	anntena	1
mission	camera	30
	G sensor	30
total		533

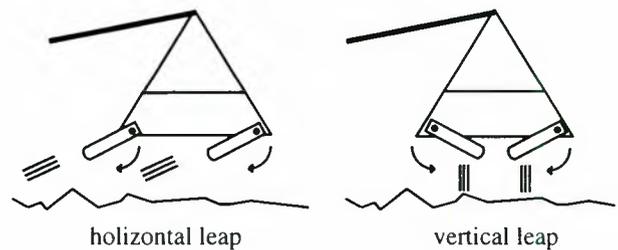


Figure 3: Jumping motion

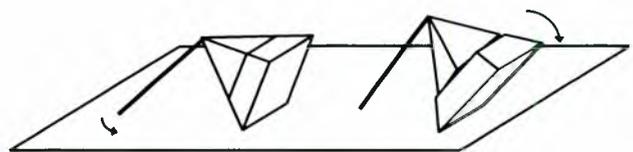


Figure 4: Turn-over motion

surface, perpendicular each others. The forces given from the ground are expressed as follows:

$$F_z = \begin{cases} -K_w z - D_w \dot{z} & (\dot{z} < 0) \\ -K_w z & (\dot{z} \geq 0) \end{cases}$$

$$F_x = -\text{sign}(\dot{x})\mu F_z$$

$$F_y = -\text{sign}(\dot{y})\mu F_z$$

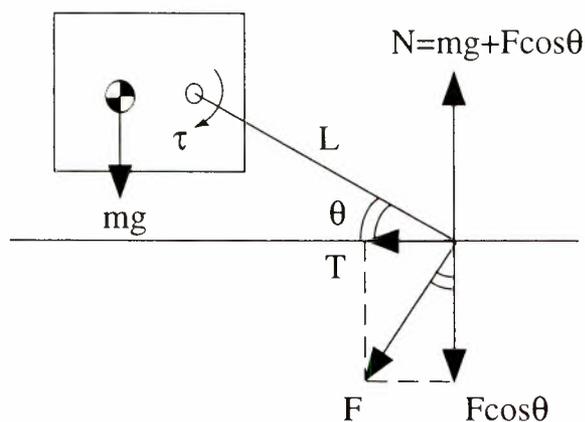


Figure 5: Leg Mechanics

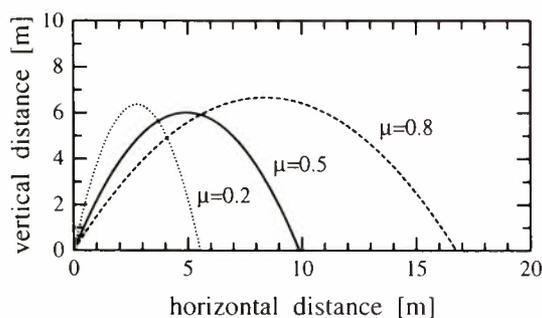


Figure 6: Jumping Trajectories

where the direction of the friction forces F_x, F_y faces to reduce the tangent velocity of the colliding part. A numerical computation algorithm is developed to avoid an energy gain due to improper modeling of friction.

5 Dynamic Simulation

Dynamic simulations are carried out with the above models of the leg mechanics and the friction mechanics. The motion trajectories of the jumping motion under $g = 0.01m/s^2$, with different friction coefficients, are depicted in Figure 6.

6 Conclusions

In this paper, we propose a possible design of a micro rover for an asteroid mission, particularly looking at the MUSES-C mission. The design is simple and promising. The weight and power budget are estimated. The motion performance is evaluated by computer simulations with a proper dynamic model.

We are looking for research collaborators and industrial sponsor who are attracted by the proposed design, and also an alternative launch opportunities

for asteroid or comet exploration, in which the proposed tortoise performs a giant leap.

SYMOFROS: A Flexible Dynamics Modeling Software

J.-C. Piedbœuf M. Doyon P. Langlois
R. L'Archevêque

Space Technologies, Canadian Space Agency,
6767 route de l'Aéroport, St-Hubert, Quebec, Canada
tel: (514) 926-4688, fax: (514) 926-4695 email: Jean-Claude.Piedboeuf@space.gc.ca

ABSTRACT

This paper describes the modeling of mechanisms in tree topology with closed kinematic loops and non-holonomic constraints. The dynamics equations are built using Jourdain's principle. The kinematics and the dynamics are developed recursively to optimize the model. The method is implemented in the modeling software SYMOFROS using the symbolic language Maple. The recursive procedures of Maple are used to obtain an efficient model generation. The model generated is totally symbolic. From that model, a C model compatible with Matlab/Simulink is generated. A graphical user interface has been developed to simplify the data input by the user. The objects are chosen from a library and the mechanism is build by linking the different objects together. Many system parameters can be fixed interactively.

1 INTRODUCTION

Canada is currently developing the Mobile Servicing System (MSS) that will be used to build and maintain the International Space Station (ISS). The MSS consists of a mobile base on which is mounted a large manipulator with seven actuators called the space station remote manipulator system (SSRMS). At the tip of the SSRMS, two smaller arms are attached on a rotating joint. This second assembly is called the special purpose dextrous manipulator (SPDM). Each arm has seven actuators. The SSRMS is 17 meters long and has flexible joints and links. The SPDM is 3.4 meter long and has flexible joints. A simplified model of the complete system includes 22 rigid degrees of freedom (dof) and more than 30 flexible ones.

In a typical maintenance task, one of the SPDM arm will grasp a stabilization point creating a closed kinematic loop. The other will be used to remove and replace a part on the station. Therefore, the contact dynamics of the system must be understood.

The MSS system is quite complex but this complexity is typical of many existing mechanisms. In order to improve a design, to develop control, or to sim-

ulate a system, dynamic models are required. These models can be obtained through a variety of methods: Newton-Euler, Lagrange, d'Alembert, Kane. These methods can be applied using purely numerical approaches or using symbolic computation. In the second case, the model is generated symbolically and can be used for simulation or control.

In the past twenty years, the Canadian Space Agency (CSA) has developed several modeling and simulation tools for off-line and real-time simulation of space manipulators. These modeling programs are based on a recursive Newton-Euler approach implemented numerically [1]. The real-time version is currently used for astronaut training for the future missions related to the ISS.

In the last few years, CSA has explored symbolic computation to model flexible manipulators. Symbolical programs such as Maple or Mathematica permit manipulation of symbols. Therefore, the dynamic model can be generated prior to the simulation and symbolic approaches should be more efficient for simulation. By contrast, in a purely numerical approach, the dynamic model must be re-created at each integration step¹.

In this regard, we have developed a general purpose program based on Maple: SYMOFROS [2]. The current version is able to model manipulators in tree topology with flexible links and joints and with closed kinematic loops. The model is developed using a recursive Jourdain approach and the foreshortening of the flexible link is included [3]. SYMOFROS has been used extensively to simulate and control experimental robots with flexible links and joints. It has also been used to develop simulation models of more industrial robotic applications. It is available on a multitude of platforms and is suited to real-time applications.

In this paper, we will go over SYMOFROS, starting with the graphical user interface, the model generation, the C implementation and the modeling done in the Matlab-Simulink environment. SYMOFROS (Fig.1) is a modeling and simulation tool based on Maple for the symbolic model genera-

¹For real-time applications, parts of the model are assumed constant for a few steps.

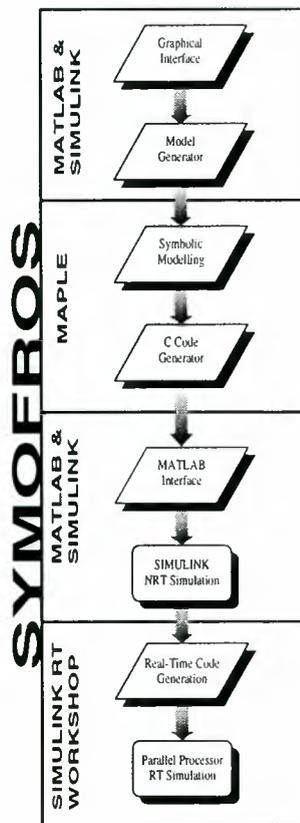


Figure 1: From a System to a Real-Time Simulation

tion and on Matlab/Simulink for the graphical user interface (GUI), the simulation and the real-time implementation.

2 SYSTEM DESCRIPTION

One of the main difficulties in the development of a general purpose program for modeling is the representation of a system. This representation should be flexible enough to allow modeling of different systems easily. It should allow the addition or removal of an object without having to redefine all the structure. It should also allow the creation of a library of manipulators or parts of manipulators that can be re-used in creating new models. The development of a good GUI is closely linked to an adequate description of the system topology. The main difficulty is the processing of the different branches and the closed kinematic loops.

In SYMOFROS, we choose an object-oriented approach to describe a mechanism. The two main objects are: generalized body and closure. In Figure 2 a general system is described using these two objects. A generalized body is composed of a body (rigid or flexible) and a joint as illustrated.

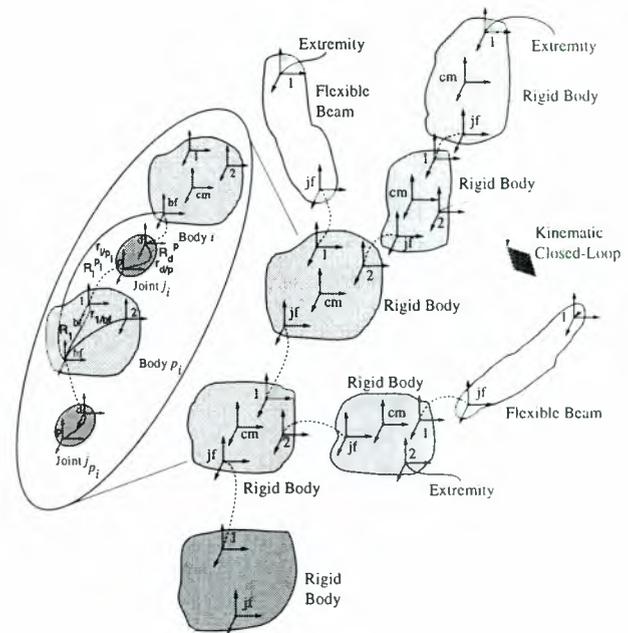


Figure 2: A Tree Structure with Closed-Loop

2.1 Body: Rigid and Flexible

The geometric properties of a rigid body are defined by giving the relationships between the extremity frames and the body frame (Fig. 2). Any number of extremity frames can be defined. The required informations are the rotation matrices R_i^{bf} (from frame i to frame bf) and the position vectors $r_{i/bf}$ (origin of frame i with respect to the origin of frame bf).

For a rigid body, the center of mass frame is added to define the inertia parameters. This frame is defined with respect to the body frame by a rotation matrix R_{cm}^{bf} and a position vector $r_{cm/bf}$. The inertia parameters are the body mass and the inertia matrix. This matrix can be defined either in the body frame or in the center of mass frame.

For a rigid or a massless body, external forces and torques acting either on the body frame or on the center of mass frame (for a rigid body) can be specified. For the rigid body, it is also possible to specify a reduction ratio to represent the gyroscopic effect of the rotor of an electrical motor connected through a reducer.

A flexible body is defined as a flexible beam. Only one extremity frame can be defined because an ideal beam is slender. The only geometric information required is the beam length. The rotation matrix and the position vector between the body frame and the extremity frame are computed by the program. The beam foreshortening is taken into account so beam stiffening is included in the model.

A flexible beam can have deformations in bend-

ing in two perpendicular directions, and in torsion around the longitudinal axis. The beam deformations are represented using an extended assumed mode method [4]. The user needs only to supply the number of modes that are used for each direction. Zero modes in a given direction is equivalent to assume a rigid beam in that direction. The default shape functions for the assumed modes are spline functions but any other assumed modes can be chosen from a library of functions. The beam's internal damping is represented using a Voigt-Kelvin model.

External forces and torques can also be specified for a flexible beam. The external forces are applied on the centroid of the beam sections. They are integrated along the beam axis during the construction of the equations of motion.

The rigid bodies do not have any internal degrees of freedom (dof). Flexible beams have internal dofs to represent the beam's flexibility. The relative motion between bodies is represented through the joints. They contain all the rigid body motion dof.

2.2 Joint

A joint is characterized by the relationship between the proximal frame and the distal frame (Fig. 2). The rotation matrix \mathbf{R}_d^p and the position vector $\mathbf{r}_{d/p}$ must be provided by the user. A joint can have from zero (constant rotation matrix and position vector) to six dof. This implies that all the different joint types can be represented.

The internal forces and torques between the bodies are specified in the joints. These forces can represent the motor torque, the elastic torque of an elastic joint or the damping force. The internal forces are represented by giving their work function. From that work, the program can compute the generalized forces associated with each generalized coordinates.

2.3 Closed Kinematic Loops

If a closed kinematic loop exists, the closure conditions must be specified. The closures are applied by connecting two extremity frames. One of the two frames is chosen as the reference frame to specify the closure conditions. The user indicates the directions in translation and in rotation along which the motion is not permitted. Closure equations are generated by SYMOFROS, along with the constraints' jacobian matrix and the non-linear terms of the constraints' second time derivative. This allows easy implementation of kinematic constraints through Lagrange multipliers. Since this approach imposes the constraints at the acceleration level, Baumgarte stabilisation is possible for enhanced stability at the position and velocity level.

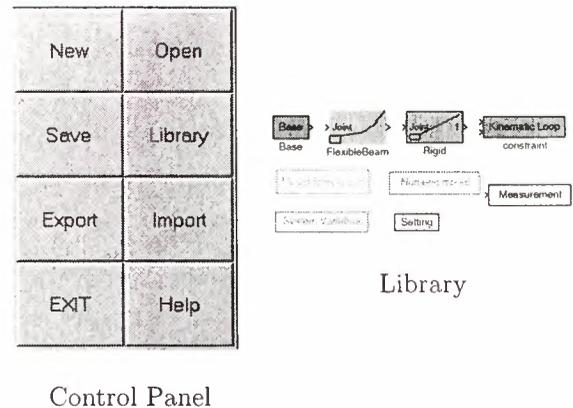


Figure 3: SYMOFROS Interface

2.4 System Parameters

In addition to the body, joint and closure descriptions, the system parameters, the generalized coordinates and the input variables must be specified. The generalized coordinates are specified only for rigid body motions, i.e., the joint variables. The coordinates associated to the beam flexibilities are defined by the program using the number of modes fixed by the user.

The user has also the possibility to give some general flags to determine how the model will be evaluated. For example, it is possible to linearize the model around a point or generate the equations required to compute the energy. It is also possible to specify non-holonomic constraints.

The gravity can be specified for the complete system by defining it on the base body. Only one base body can be specified for a system. The body frame of the base body is equivalent to the inertial reference frame.

3 THE GRAPHICAL USER INTERFACE

SYMOFROS is based both on Maple for the code generation and on Matlab-Simulink for the simulation. The code generation requires four text files that contain the complete information describing the robotic system. This information includes the model's topology, the bodies' symbolic description, the joints' symbolic description, the numerical values and the different paths. These files, being read by a program, must have a specific format and must be flawless. Although some expert SYMOFROS users are able to type in the information on their own, most beginning users are not and rely on the graphical user interface (GUI).

The GUI is based on Matlab-Simulink and uses the Simulink block diagram approach to describe the system's topology. Blocks from a library are dragged

and dropped, then linked together using the Simulink arrows. These blocks represent either a rigid body, a flexible beam, a kinematic loop, or model parameters. Each block can be double-clicked to display a window that contains its relevant information. The information is mainly entered as variables which are later assigned a value. This enables Maple to generate the model in a symbolic form. Different numerical applications of the same model can then easily be produced. The numerical values are assigned either as constants to be hard coded in a C program or as parameters that will be given as an input to the model at run-time.

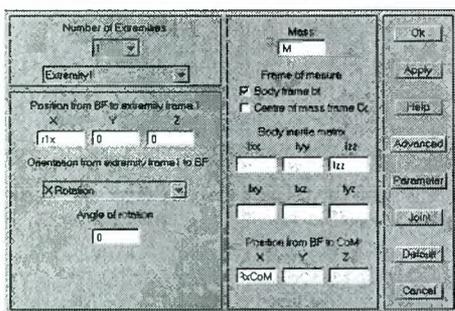


Figure 4: Description Window of a Rigid Block.

The two main building blocks for robots are rigid bodies and flexible beams. The kinematic and dynamic properties of both blocks are entered through a window-type interface, by typing in the names of the variables representing the parameters that are to be considered by the model. Beam flexibility is modeled using assumed modes. The user can enter the number of modes and rigidity to be used to describe the vibrations in the XY-plane, the XZ-plane and torsion around the local X axis.

While bodies are mostly static entities, joints describe rotations and/or translations between bodies. To keep the graphical model concise, all joints are included in the bodies and are always preceding them. Torques, damping, elasticity can all be entered into the model using a virtual power formulation. The virtual power is an expression of the type :

$$\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \delta \left(\frac{d\mathbf{q}}{dt} \right) \quad (1)$$

with which many are unfamiliar but the interface provides a more convivial way to define it for simple joints.

SYMOFROS, along with the interface, allows a user to start a medium-sized model from scratch and have a compiled executable within the hour. This executable can then be interrogated to give the mass matrix and non-linear vector, or any of the matrices that are functions of the model's states and inputs.

4 MODELING

SYMOFROS obtains the symbolic model of a system using Jourdain's principle[5], which is a variation of the generalized d'Alembert's principle. Since it is a variational method, the constraint forces are eliminated. A more complete description of the methodology can be found in Piedbœuf[3].

The kinematics are obtained recursively using Maple. A system of temporary variables is used to avoid an exponential increase in memory requirement for an increasing system complexity[2].

The flexible beams are modeled using the Euler-Bernoulli approximations. The foreshortening is included by considering second-order strain-displacement relationships. Using a consistent elimination of higher order terms, the resulting equations of motion are exact to the first order in terms of the flexible coordinates.

A symbolic linearization of the model is done and a number of C functions required for the simulation and control are generated. The C code is optimized using the Maple optimization. The generated code is ready to be compiled, then used with Matlab/Simulink and the Simulink Real-Time Workshop (RTW).

5 SIMULINK INTERFACE

Once Maple has processed the information related to the model's description, it is able to generate C code. The generated C model is to be used at the Matlab Prompt, in a Simulink simulation or with RTW. In addition, SYMOFROS supports the use of multiple models at the same time. The C model is used in the Matlab/Simulink Environment (essentially using mxArray² data structures for storage) but can very easily be adapted to pure C. Finally, SYMOFROS is a multi platform supported package (NT, Win95, SunOS, QNX) giving more flexibility and robustness for the user.

5.1 Accessing the model

The approach taken is similar to the idea exploited by Matlab with the *SimStruct* where the *SimStruct* contains all the information related to an S-Function.

In each MODEL, a C structure (ModelStruct) contains all the relevant information related to it. The ModelStruct contains a set of pointers to functions (see table 1) that, once initialized, point to all the available SYMOFROS functions; the ModelStruct contains pointers to the data storage arrays for the calculation results and a set of informative structures representing the dimensions and configuration parameters. The important issue here is that

²A Simulink data structure

each generated MODEL has its own static ModelStruct variable and static functions (initialized in the ModelStruct pointer-to-function section) that can be accessed externally via an initialization function.

Type of Functions
Model Dynamics
Model Kinematics
Holonomic Constraints
Non Holonomic Constraints
Energy

Table 1: SYMOFROS Functions

5.2 Interface with Matlab

The generated model is interfaced with Matlab, Simulink and RTW. It is always very useful to be able to examine the response of a MODEL in the Simulink/RTW environment and at the Matlab command prompt as well. The mechanisms involved in Matlab and Simulink/RTW are different (mexFunction vs SFunction). Since in the end, the same C code is used for the model, a simple interface file has been designed to support both, and properly allocate and free the memory. In Simulink, each MODEL is called once at initialization using *mexCallMATLAB* to get the pointer to the model. Once the pointer to the desired model is obtained, it is accessed directly instead of using *mexCallMATLAB*. The *mexCallMATLAB* is only used in the Simulink environment and not in the RTW. To overcome the problem, a compilation flag is used to determine if Simulink or RTW is to be used. In the later, a direct call to the initialization function is made.

5.3 Generic and Reusable

As will be described in the next section, a set of operators on the model has been developed. These operators are generic enough that the same Simulink diagram can be reused with multiple different SYMOFROS generated models. For example, one could perform a first serie of tests with a rigid model and then study another model with flexible or elastic parameters always using the same Simulink diagram (probably with a different initialisation file).

6 SYMOFROS LIBRARY

By accessing the models' functions with Simulink, it's possible to create a complete real-time simulation within a short development time. SYMOFROS provides the "symoSFunction" block, a Simulink block used to query the model in real-time. By using multiple instances of "symoSFunction", it's possible

Parameters Name	Description
ModelName	Name of the model used to perform the query
ModelFunction	Model function executed

Table 2: Parameters of the block "symoSFunction"

to build a complete simulation that interacts with one or many models. As shown in table 2, this SYMOFROS query block has two parameters.

6.1 Library Description

SYMOFROS provides a set of Simulink blocks that allow the execution of model queries, and some standard operations used in robotics. These standard operations are divided in 8 categories.

Initialisation The two blocks defined in this category are used to setup a simulation environment and the model parameters. For multi-model simulations, each model must have an associated "Model Initialisation block".

Dynamics blocks provide the functionality to apply commands to the model. Through these blocks, one can apply torques and trajectories to the different model's joints. It also allows the application of perturbations (external forces and torques) to the model. As results, we obtain the updated states, the joints' accelerations, the joints' dynamic friction forces and constraint forces. Moreover, an inverse dynamics block computes the joints torque from the joints' trajectory (position, velocity and acceleration).

Kinematics are implemented by numerical methods. The direct kinematics outputs the positions, velocities and accelerations of the model extremities. An inverse kinematics block computes the joints trajectory from a cartesian trajectory of a model's extremity. The translational and rotational jacobians and their time derivatives can also be accessed.

Inputs blocks specify predetermined joint and cartesian trajectories. It also allows to predefine a sequence of joint torque, trajectory and perturbation to the model. Generally, the inputs are specified in an independent file.

Controller blocks provide friction compensation, cartesian linear control and cartesian feedback control.

Graphical blocks deal with plotting the simulation results. Data such as the model states, model joints acceleration and extremity behavior are stored during the simulation. Some blocks are

then used to display the results in graph format. Moreover some development is being made to SYMOFROS to get a 3D visual feedback in real-time.

Network & Communication Some blocks have been developed for the support of communication links. Mainly, SYMOFROS has a transmitter and a receiver that transfer data through an internet protocol socket.

Finally, SYMOFROS provides some blocks for generic tasks such as frame transformations, orientation type conversion, orientation error computation, etc.

6.2 Simulation Block Use

To use a particular SYMOFROS block, the user only needs to drag and drop the desired block from the SYMOFROS blockset to his simulink sheet. By double-clicking on the block, a menu appears with the block's settings (figure 5). A short block description and a help button are accessible from that menu.

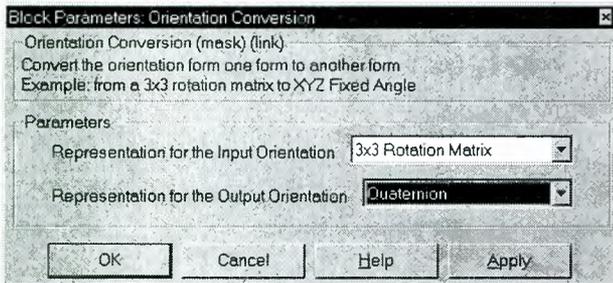


Figure 5: Block Settings

6.3 Simulation Block Implementation

The SYMOFROS blockset is implemented in 3 ways:

Simulink + Toolboxes (.mdl) Use of the standard Simulink blockset and of the Toolboxes blockset (like Digital Signal Processing). Most of the SYMOFROS Blockset has been developed with pre-built Simulink blocks. This method accelerates the development and maintenance processes. Since the simulink diagrams are portable, it is easy to generate real-time code with this method.

SFunctions (.c) C Source code embedded in a Simulink C source file template (through the use of specific macros). This method is used for the functionalities that are not supported by the Simulink blocksets. For example, the SYMOFROS network blocks are implemented

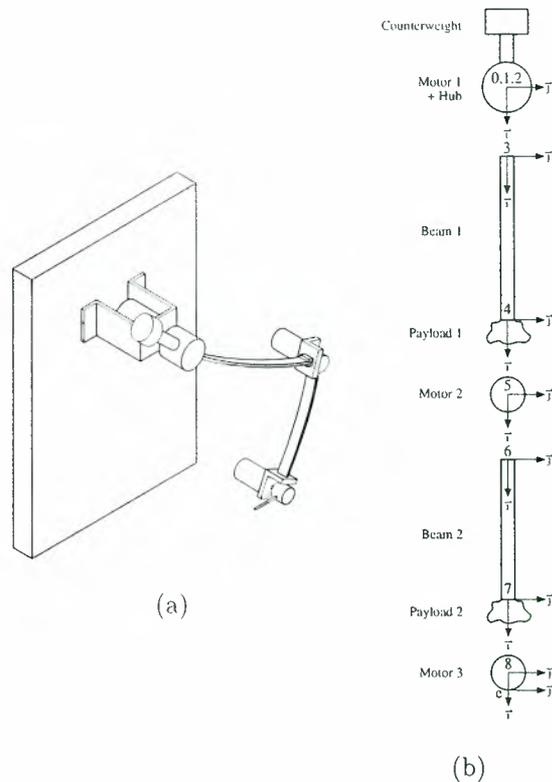


Figure 6: Planar Flexible Robot

with this method in order to support the socket communication. To obtain efficient code, the Target Language Compiler (TLC) is used to define the rules for the code generation.

Matlab Scripts (.m) This non real-time method is generally used for the initialization processes (simulation + models). Thus, the definition of trajectories and inputs is developed with Matlab scripts.

7 EXAMPLE OF A MODEL

A planar robot with three harmonic drive motors and two flexible links as shown in Figure 6(a), is used to illustrate the capability of the GUI. This robot was built at École Polytechnique and is used to study the modeling and control of flexible robots. Figure 6(b) shows a simplified model using 9 bodies (the counterweight is combined with joint 1) with their associated reference frames. Figure 7 shows the GUI representation of the robot. As indicated on Figure 7, joint elasticity is taken into account in the modeling of the first motor (motor1) while the two others are supposed to be rigid. Beams 1 and 2 are flexible while all other bodies are assumed rigid.

minor changes brought to the initialization file.

The example used for the simulation is the following. The robot starts with motor angles at -0.1 , 0.1 and 0 radian for motors 1, 2 and 3 respectively. The goal is to go to point $(0.3, 0.3)$ and from that point, trace a square with 0.6 m sides. the simulation diagram is shown in Figure 9. This shows that we can accurately simulate the behavior of a flexible robot, and the fact that the actual trajectory is not exactly the desired one only means that the controller used is not optimal.

In this example, the desired trajectory corresponded to a square. The figure below shows the superimposed desired and actual trajectories of the robot.

9 CONCLUSION

This paper described the development of a program to model flexible mechanisms in tree-topology with closed kinematic loops and non-holonomic constraints. The description of systems using an object oriented approach have been described. Three main objects are used: body, joint and closure. The dynamic equations are developed using Jourdain's principle with recursive kinematics. The flexible links are modeled as Euler-Bernoulli beams with the inclusion of the foreshortening effect. The method has been implemented in the program SYMOFROS. This program is based on the symbolic language Maple. The graphical user interface developed for SYMOFROS facilitates the input of a model, especially for user with little dynamics experience. The result of the symbolic modeling is a optimized C model. This model is fully compatible with Matlab/Simulink and can be run in non-real-time or in real-time on parallel processors. The SYMOFROS program is an appropriate tool for modeling and simulation of medium complexity mechanisms such as robots. It has been used to develop real-time control and hardware-in-the-loop simulation for robots. A copy of the program can be obtained from the first author.

References

- [1] O. Ma, K. Buhariwala, N. Roger, J. MacLean, and R. Carr, "Mdsf- a generic development and simulation facility for flexible, complex robotic systems," *Robotica*, vol. 15, pp. 49-67, 1997.
- [2] J.-C. Piedbœuf, "Modelling flexible robots with Maple," *MapleTech: The Maple Technical Newsletter*, vol. 3, no. 1, pp. 38-47, 1996.
- [3] J.-C. Piedbœuf, "Recursive modelling of flexible manipulators," *The Journal of Astronautical Sciences*, vol. 46, January-March 1998.
- [4] M. Saad, J.-C. Piedbœuf, O. Akhrif, and L. Saydy, "A comparison of different shape function in assumed-mode model of a flexible slewing beam," tech. rep., Canadian Space Agency, St-Hubert, QC, Canada, 1997.
- [5] P. E. B. Jourdain, "Note on an analogue of Gauss' principle of least constraint," *The Quarterly Journal of Pure and Applied Mathematics*, vol. XL, pp. 153-157, 1909.

List of Participants

LIST of PARTICIPANTS

Mr. A. Abramovici,
MacDonald Dettwiler
Space and Advanced Robotics Ltd.
9445 Airport Road
Brampton, Ontario
Canada
Tel: +1 905 790 2800
Fax: +1 905 790 4420
aabramov@mdrobotics.ca

Mr. D.J. Atkinson,
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 393 2769
Fax: +1 818 393 2454
David.J.Atkinson@jpl.nasa.gov

Mr. M.R. Barry,
United Space Alliance
600 Gemini Avenue
Houston
Tx 77058
USA
Tel: +1 281 282 3960
Fax: +1 281 282 3587
mrb@rice.edu

Mr. N.. Bataille,
CNES
18, Avenue Edouard Belin
F-31401 Toulouse Cedex 4
France
Tel: +33 5 6128 1805
Fax: +33 5 6128 1855
nicolas.bataille@cnes.fr

Mr. R. Bertrand
Von Hoerner & Sulger
Schlossplatz 8
D68732 Schwetzingen
Germany
Tel: +49 6202 21091
Fax: +49 6202 24303
bertrand@vhs.de

Mr., R.G.J. Biesbroek,
JAQAR Space Engineering
Hector Berliozkade 68
NL-2551 XR Den Haag
The Netherlands
Tel: +31 70 397 8665
Fax: +31 70 397 8665
rb@jaqar.demon.nl

Mr., T.T. Blackmon,
NASA Ames Research Center
Intelligent Mechanisms Group
Moffett Field, Ca 94035
USA
Tel: +1 650 604 4710
Fax: +1 650 604 4036
blackmon@mail.arc.nasa.gov

Mr. J.P.J. Bol,
Hogeschool Holland
The Netherlands
Tel: +31 71 4074696
212027@hsholland.nl

Mr. G. Borghi,
Carlo Gavazzi Space
Via Gallarate 139
20151 Milano
Italy
Tel: +39 02 3804 8261
Fax: +39 02 308 6458
gborghi@cgspace.it

Mr E. Bornschlegl,
European Space Agency (TOS-WSD)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3487
Fax: +31 71 565 4295
eric@wd.estec.esa.nl

Mr. J.L. Bresina,
NASA Ames Research Center
MS 2692
Moffett Field,
CA 94035
USA
Tel: +1 650 604 3365
Fax: +1 650 604 3594
bresina@arc.nasa.gov

Mr. T.B. Brooks,
Century Computing Division,
AppNet. Inc.
1100 West Street
Laurel,
MD 20707
USA

Mr W. Baarls,
Kerkstraat 75
NL-2211 RD Noordwijkerhout
The Netherlands
Tel: +31 252 376280
wcaarls@wins.uva.nl

Mr. G. Capuano,
Techno System Developments
Zona Industriale S. Martino 27
I-80078 Pozzuoli Naples
Italy
Tel: +39 081 526 3475
Fax: +39 081 526 2701
TSdev@tin.it

Mr. A. Cesta,
IP-CNR
Italian Nat. Research Council
Viale Marx 15
I00137 Rome
Italy
Tel: +39 06 8609 0209
Fax: 39 06 824737
cesta@ip.rm.cnr.it

Ms. M.C. Charmeau,
CNES
18, Avenue Edouard Belin
F-31401 Toulouse Cedex 4
France
Tel: +33 5 6127 4184
Fax: +33 5 6128 1996
Marie-Claire.Charmeau@cnes.fr

Mr. A. Chella,
University of Palermo
Dept. of Electrical Eng.
Viale Delle Scienze
90128 Palermo
Italy
Tel: +39 091 6566111
Fax: +39 091 488452
chella@unipa.it

Mr. Y. Coene,
Spacebel Informatique SA
I. Vandammestraat 7 bus 1
B1560 Hoeilaart
Belgium
Tel: +32 2 658 2025
Fax: +32 2 658 2090
Yves.Coene@spacebel.be

Mr. Y. Culyer,
Defence Evaluation and Research Agency
Farnborough
Hampshire GU14 0LX
United Kingdom
Tel: +44 1252 393052
Fax: +44 1252 396310
djculyer@scs.dra.hmg.gb

Mr. S.K. Das,
Charles River Analytics, Inc.
725 Concord Avenue
Cambridge,
MA 02138
USA
Tel: +1 617 491 3474
Fax: 1 617 868 0780
sdas@cra.com

Ms. F. De Hilster,
Fokker Space B.V.
Newtonweg 1
Postbus 32070
NL-2303 DB Leiden
The Netherlands
Tel: +31 71 5245109
f.de.hilster@fokkerspace.nl

Mr. D. De Hoop,
NIVR
Kluyverweg 1
P.O. Box 35
NL-2600 AA Delft
The Netherlands
Tel: +31 15 278 7340
Fax: +31 15 2623096
d.dehoop@nivr.nl

Ms. C.M. De Vos,
81 Rijnsborgerweg
NL-2334 BJ Leiden
The Netherlands
Tel: +31 71 517 3757
cdevos@estec.esa.nl

Mr. V. Dhiri,
European Space Agency (MSM-HF)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5861
vdhiri@esa.estec.nl

Mr. F. Didot,
European Space Agency (TOSMMA)
Postbus 299
2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 4403
Fax: +31 71 565 5419
fdidot@estec.esa.nl

Mr. S. Doi,
NASDA Tsukuba Space Center
211 Sengen,
Tsukuba, Ibaraki 305
Japan
Tel: +81 298 54 3934
Fax: 81 298 50 1480
doi.shinobu@nasda.go.jp

Mr. A. Donati,
European Space Agency (TOS-OS)
Robert Bosch Strasse 5
D-64293 Darmstadt
Germany
Tel: +49 6151 902.574
Fax: +49 6151 902.561
adonati@esoc.esa.de

Mr. R. Doyle,
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 354 9894
Fax: +1 818 354 2454
Richard.J.Doyle@jpl.nasa.gov

Mr. J.J. Fernandez Lozano,
European Space Agency (TOS-MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
jflozano@ctima.uma.es

Mr. Fernando Correia Silva,
Uninova Centre for Intelligent Robotics
Quinta da Torre
2825 Monte de Caparica
Portugal
Tel: +351 931 9141339
nfcs@uninova.pt

Mr. A. Finzi,
Università di Roma La Sapienza
Via Salaria 113
I-00186 Roma
Italy

Mr. A. Flatscher,
Dornier Satellitensysteme
P.O. Box 1420
D88039 Friedrichshafen
Germany
Tel: +49 7545 8 4979
Fax: +49 7545 8 4105
reinhold.flatscher@dss.dornier.dasa.de

Mr. W. Freihoefer,
European Space Agency
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31.71.565.6565
wfreihoe@estec.esa.nl

Mr. J. Fuchs,
European Space Agency (APP-JPP)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5296
Fax: +31 71 5654696
jfuchs@estec.esa.nl

Mr. Y. Fukushima,
NASDA Tsukuba Space Center
211 Sengen
TsukubaShi
Ibaraki, 305
Japan
Tel: +81 298 52 2852
Fax: 81 298 60 6718
fukushima.yohsuke@nasda.go.jp

Mr. D. Galardini,
Trasys Space
Leuvensesteenweg 510/43
B1930 Zaventem
Belgium
Tel: +32 2 773 9444
Fax: +32 2 773 7910
daniele.galardini@trasys.be

Mr. A.R. Ghavimi,
 Jet Propulsion Laboratory
 4800 Oak Grove Drive
 Pasadena,
 CA 91109-8099
 USA
 Tel: +1 818 354 0470
 Fax: +1 818 393 4440
 ali.r.ghavimi@jpl.nasa.gov

Mr. G. Goelz,
 German Aerospace Center, DLR
 Koenigswintererstr. 522524
 Postfach 300364
 D-53227 Bonn
 Germany
 Tel: +49 228 447 522
 Fax: +49 228 447 718
 Gerd.Goelz@dlr.de

Mr. R.G. Gosine,
 CCORE
 Memorial Univ. of Newfoundland
 POB 90
 St John's,
 NF A1B 3X5
 Canada
 Tel: +1 709 737 8139
 Fax: 1 709 737 4706
 rgosine@enr.mun.ca

Mr. T.J. Grant,
 Origin Nederland B.V.
 P.O. Box 1444
 NL-3430 BK Nieuwegein
 The Netherlands
 Tel: +31 30 608 8888
 Fax: +31 30 606 0577
 tim.grant@nl.originit.com

Mr. S. Gruenfelder,
 Austrian Aerospace
 Stachegasse 16
 A1120 Wien
 Austria
 Tel: +43 1 80199 5786
 Fax: +43 1 80199 5577
 stephan.gruenfelder@space.at

Mr. M. Hashimoto,
 ISAS
 311 Yoshinodai
 Sagamihara
 Kanagawa 229
 Japan
 Tel: +81 42 759 8352
 Fax: +81 42 759 8440
 masashi@nsl.isas.ac.jp

Mr. R. Hayashi,
 Tokyo Institute of Technology
 21210Okayama,
 Meguroku
 Tokyo 152
 Japan
 Tel: +81 3 5734 2644
 Fax: +81 3 5734 3982
 ryo@mes.titech.ac.jp

Mr. S. Hayati,
 Jet Propulsion Laboratory
 Mail Code 198219
 4800 Oak Grove Drive
 Pasadena,
 CA 91109-8099
 USA
 Tel: +1 818 354 8273
 Fax: +1 818 354 7354
 Samad.a.Hayati@jpl.nasa.gov

Mr. C.J.M. Heemskerk,
 Fokker Space B.V.
 Newtonweg 1
 Postbus 32070
 NL-2303 DB Leiden
 The Netherlands
 Tel: +31 71 5245427
 Fax: 31 71 5245290
 C.Heemskerk@fokkerspace.nl

Mr. A. Hipp,
 KayserThrede GmbH
 Wolfratshausener Str. 4448
 D-81379 Muenchen
 Germany
 Tel: +49 89 7249 5402
 Fax: +49 89 7249 5104
 ah@kaysertrede.de

Mr. G. Hirzinger,
 DLR Inst. fuer Rob. & Syst. Dynamic
 Postfach 1116
 D82230 WesslingOberpfaffenhofen
 Germany
 Tel: +49 8153 28 2401
 Fax: +49 8153 28 1134
 Gerd.Hirzinger@dlr.de

Mr. T. Horn,
 European Space Agency (TOS-MMO)
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 4552
 thorn@estec.esa.nl

Mr. I. Horvath,
 Central Research Institute for Physics
 P.O. Box 49
 1525 Budapest
 Hungary
 Tel: +36 1 395 9220
 Fax: +36 1 395 9151
 horvatic@rmki.kfki.hu

Mr. M.S. Hutchins,
 Defence Evaluation and Research Agency
 Farnborough
 Hampshire GU14 0LX
 United Kingdom
 Tel: +44 1252 392.534
 Fax: +44 1252 396.310
 mshutchins@scs.dera.gov.uk

Mr. A.E. Johnson,
 Jet Propulsion Laboratory
 4800 Oak Grove Drive
 Pasadena,
 CA 91109-8099
 USA
 Tel: +1 818 354 0357
 Fax: +1 818 393 4085
 aej@robotics.jpl.nasa.gov

Mr. A. Jonsson,
 NASA Ames Research Center
 Mail Stop 2692
 Moffett Field,
 Ca 94035
 USA
 Tel: +1 650 604 2799
 Fax: +1 650 604 3594
 jonsson@ptolemy.arc.nasa.gov

Mr. K. Kapellos,
 Trasy Space
 Leuvensesteenweg 510/43
 B1930 Zaventem
 Belgium
 Tel: +32 2 773.7417
 Fax: +32 2 773.7910
 konstantinos.kapellos@trasy.be

Mr. T. Kasai,
 NASDA Tsukuba Space Center
 211 Sengen TsukubaShi
 Ibaraki, 305
 Japan
 Tel: +81 298 52 2863
 Fax: +81 298 60 6517
 kasai.toru@nasda.go.jp

Mr. S. Kimura,
 Communications Research Lab.
 421 NukuiKitamachi Koganei
 Tokyo 1848795
 Japan
 Tel: +81 423 27 7514
 Fax: +81 423 27 6699
 shin@crl.go.jp

Mr. D. King,
 MacDonald Dettwiler
 Space and Advanced Robotics Ltd.
 9445 Airport Road
 Brampton, Ontario
 Canada
 Tel: +1 905 790 2800
 Fax: 1 905 790 4452
 dking@mdrobotics.ca

Mr. R. Knight,
 Jet Propulsion Laboratory
 4800 Oak Grove Drive
 Pasadena,
 CA 91109-8099
 USA
 Tel: +1 818 393 5374
 Fax: +1 818 393 5244
 knight@aig.jpl.nasa.gov

Mr. B. Koopen,
 Universiteit van Amsterdam
 Fideliolaan 352
 NL-1183 PX Amstelveen
 The Netherlands
 Tel: +31 20 645 1774
 bkoopen@wins.uva.nl

Mr. R. Krenn,
 DLR (FFDR)
 Postfach 1116
 D82230 Wessling-Oberpfaffenhofen
 Germany
 Tel: +49 8153 28 1463
 Fax: +49 8153 28 1450
 rainer.krenn@dlr.de

Mr. R. Krickl,
 Vienna University of Technology
 Inst. of Flexible Automation
 Getreidemarkt 14/17
 A-1010 Wien
 Austria
 Tel: +43 676 3284 185
 Fax: +43 159 53711
 rk@flexaut.tuwien.ac.at

Mr. T. Kubota,
 Inst. of Space and Astronautical Science
 311, Yoshinodai,
 Sagamihara
 Kanagawa 2298510
 Japan
 Tel: +81 42 759 8305
 Fax: +81 42 759 8304
 kubota@cnenet.isas.ac.jp

Mr. Y. Kunii,
 Chuo University
 11327 Kasuga,
 Bunkyo-ku
 Tokyo 112
 Japan
 Tel: +81 3 3817 1866
 Fax: +81 3 3817 1847
 kunii@elect.chuou.ac.jp

Mr. Y. Kuroda,
 Meiji University
 111, Higashimita
 Tamaku
 2148571 Kawasaki
 Japan
 Tel: +81 44 934 7183
 Fax: +81 44 934 7183
 ykuroda@isc.meiji.ac.jp

Mr. S. Lacroix,
 LAASCNRS
 7, Avenue du Colonel Roche
 F-31077 Toulouse Cedex
 France
 Tel: +33 5 6133 6266
 Fax: +33 5 6133 6455
 simon.lacroix@laas.fr

Mr. M. Lamboley,
 CNES
 18, Avenue Edouard Belin
 F-31401 Toulouse Cedex 4
 France
 Tel: +33 5 6127 3983
 Fax: +33 5 61 28 17 48
 mercel.lamboley@cnes.fr

Mr. K. Landzettel,
 DLR
 Postfach 1116
 D-82230 Wessling-Oberpfaffenhofen
 Germany
 Tel: +49 8153 28 2403
 Fax: +49 81 53 28 1134
 Klaus.Landzettel@dlr.de

Mr. M. Lemaitre,
 ONERA CERT/DCSD
 2. ave E. Belin
 BP 4025
 F-31055 Toulouse Cedex
 France
 Tel: +33 5 6225 2660
 Fax: +33 5 6225 2564
 Michel.Lemaitre@cert.fr

Mr. G. Leonis,
 Spacebel Informatique SA
 I. Vandammestraat 7, bus 1
 B1560 Hoeilaart
 Belgium
 Tel: +32 2 658 2036
 Fax: +32 2 658 2090
 Guy.Leonis@spacebel.be

Mr. B. Liang,
 Beijing Institute of Control Engineering
 P.O. Box 2729
 Beijing City
 Peoples Republic of China
 Tel: +86 10 6837 8688
 Fax: +86 10 6254 3110

Mr. G. Limouzin,
 Matra Marconi Space
 31, Av. des Cosmonautes
 Z.I. du Palays
 F-31402 Toulouse CEDEX 4
 France
 Tel: +33 5 6219.6921
 guy.limouzin@tls.mms.fr

Mr. L. Liu,
 Beijing Institute of Control Engineering
 P.O. Box 2729
 Beijing City
 Peoples Republic of China
 Tel: +86 10 6837 8651

Mr. H. Lutz,
 European Space Agency (TOS-MM)
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 3631
 Fax: +31 71 565 5430
 hlutz@estec.esa.nl

Mr. P. Lux,
DaimlerChrysler Aerospace AG
RaumfahrtInfrastruktur
Postfach 286156
D-28361 Bremen
Germany
Tel: +43 421 539 4220
Fax: +49 421 539 4603
peter.lux@ri.dasa.de

Mr. O. Ma,
MacDonald Dettwiler
Space and Advanced Robotics Ltd.
9445 Airport Road
Brampton, Ontario
Canada
Tel: +1 905 790 2800
Fax: +1 905 790 4400
oma@spar.ca

Mr. G. Magnani,
Politecnico di Milano
Piazza Leonardo da Vinci 32
I-20133 Milano
Italy
Tel: +39 2 2399 3673
Fax: +39 2 2399 3412
magnani@elet.polimi.it

Mr. M. Maimone,
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 354 0592
Fax: +1 818 393 4085
mark.maimone@jpl.nasa.gov

Mr. N. Matsuhira,
Toshiba Corporation
Toshiba R&D Center
Komukai Works
1, KomukaiToshibacho
Saiwaiku, Kawasaki 2108582
Japan
Tel: +81 44 549 2193
Fax: +81 44 549 2382
nobuto.matsuhira@toshiba.co.jp

Mr. K. Matsumoto,
National Aerospace Laboratory
7441 Jindaijihigashimachi, Chofu
1828522 Tokyo
Japan
Tel: +81 422 47 5911
Fax: +81 422 40 3170
matumoto@nal.go.jp

Mr. A. Matthysen,
European Space Agency (TOS-WAS)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 5653258
arne@wgs.estec.esa.nl

Mr. S. Matunaga,
Tokyo Institute of Technology
2121 Ookayama, Meguroku
Tokyo 152
Japan
Tel: +81 3 5734 3176
Fax: +81 3 5734 3176
sab@mes.titech.ac.jp

Mr. H. Meiland,
Fokker Space B.V.
Newtonweg 1
Postbus 32070
NL-2303 DB Leiden
The Netherlands
Tel: +31 71 5245097
Fax: 31 71 524 5499
h.meiland@fokkerspace.nl

Mr. M. Miyata,
NASDA Tsukuba Space Center
211 Sengen TsukubaShi
Ibaraki, 305
Japan
Tel: +81 298 52 2369
Fax: +81 298 52 2415
Miyata.Makoto@nasda.go.jp

Mr. M. Montemerlo,
NASA Headquarters
300 E. Street, S.W.
Washington DC 20546
USA
Tel: +1 202 358 4664
Fax: +1 202 358 3096
mmontemerlo@hq.nasa.gov

Mr. S.J. Moorehead,
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh,
PA 15213
USA
Tel: +1 412 268 7086
Fax: +1 412 268 5895
sjm@ri.cmu.edu

Mr. S. Moser,
European Space Agency (TOS-MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3960
smoser@estec.esa.nl

Mr. L. Motter,
Societe Europeenne des Satellites
Chateau de Betzdorf
L6815 Betzdorf
Luxemburg
Tel: +352 710.725 221
Fax: +352 710.725 390
lorenzomotter@sesastra.com

Mr. N. Muscettola,
NASA Ames Research Center
MS 2692
Moffett Field,
CA 94035
USA
Tel: +1 650 604 4744
Fax: +1 650 604 3594
mus@ptolemy.arc.nasa.gov

Mr. K. Nakamura,
Meiji University
111, Higashimita
Tamaku
2148571 Kawasaki
Japan
Tel: +81 449.347.183
Fax: +81 44 9.347.183
nakamura@capecod.mind.meiji.ac.jp

Mr. S. Nakasuka,
University of Tokyo
731 Hongo Bunkyo
Tokyo 1138656
Japan
Tel: +81 3 3812 2111
Fax: +81 3 5841 8560
nakasuka@space.t.utokyo.ac.jp

Mr. I. Nakatani,
Inst. of Space&Astronautical Science
311, Yoshinodai, Sagami-hara
Kanagawa 2298510
Japan
Tel: +81 427 59 8303
Fax: +81 427 59 8303
Nakatani@nsl.isas.ac.jp

Mr. P. Nayak,
 RIACS
 NASA Ames Research Center
 Moffett Field,
 CA 94035
 USA
 Tel: +1 650 604 4756
 Fax: +1 650 604 3594
 nayak@ptolemy.arc.nasa.gov

Mr. C. Neveu,
 Caelum Res. Corp.
 NASA Ames Research Center
 Bldg 2694
 Moffett Field,
 CA 94035
 USA
 Tel: +1 650 604 2525
 Fax: +1 650 604 4036
 neveu@artemis.arc.nasa.gov

Mr. D. Nicolini,
 European Space Agency
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 4373
 dnicolin@estec.esa.nl

Mr. J. Nijskens,
 OSTC
 Rue de la Science 8
 B-1000 Bruxelles
 Belgium
 Tel: +32 2 238.0588
 Fax: +32 2 305.912
 nijs@belspo.be

Ms. N. Nishigori,
 Fujitsu Limited
 93, Nakase 1chome, Mihamaku
 Chibashi, Chiba
 2610023
 Japan
 Tel: +81 43 299 3248
 Fax: +81 43 299 3012
 RHE00337@nifty.ne.jp

Mr. M. Novara,
 European Space Agency (IMT-TS)
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 4003
 Fax: +31 71 565 5184
 mnovara@estec.esa.nl

Mr. W.J. Ockels,
 European Space Agency (TOS-E)
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 3505
 Fax: +31 71 565 5590
 wockels@estec.esa.nl

Mr. K. Ohno,
 Fujitsu Labs Ltd
 11 Kamikodanaka 4chome
 Nakaharaku Kawasaki 211
 Japan
 Tel: +81 44 754 2089
 Fax: +81 44 754 2582
 ohno@stars.flab.fujitsu.co.jp

Mr. Y. Okhami,
 NASDA
 World Trade Center Building
 241 Hamamatsucho, Minatoku
 Tokyo, 105 8060
 Japan
 Tel: +81 298 52 2196
 Fax: +81 298 50 1915
 ohkami.yoshiaki@nasda.go.jp

Mr. M.J.A. Oort,
 Fokker Space B.V.
 Newtonweg 1
 Postbus 32070
 NL-2303 DB Leiden
 The Netherlands
 Tel: +31 71 5245448
 Fax: +31 71 5245499
 m.oort@fokkerspace.nl

Mr. G. Ortega,
 ESA
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 3668
 gortega@wk.estec.esa.nl

Mr. W. Paetsch,
 DaimlerBenz Aerospace Infrastructure
 Space Infrastructure
 Postfach 105909
 D-28059 Bremen
 Germany
 Tel: +49 421 539 4356
 Fax: +49 421 539 4832
 wolfgang.paetsch@ri.dasa.de

Mr. J.C. Parrish,
 NASA Headquarters
 Mail Code SM
 300 E. Street, S.W.
 Washington DC 20546
 USA
 Tel: +1 301 405 0291
 Fax: +1 209 391 9785
 jparrish@hq.nasa.gov

Mr. L.F. Penin,
 National Aerospace Laboratory
 7441 Jindaijihigashimachi, Chofu
 182 8522 Tokyo
 Japan
 Tel: +81 422 40 3172
 Fax: +81 422 49 8813
 penin@nal.go.jp

Mr. J. Pereira Do Carmo,
 European Space Agency
 Postbus 299
 NL-2200 AG Noordwijk
 The Netherlands
 Tel: +31 71 565 3962
 scarmo@estec.esa.nl

Mr. J.C. Piedboeuf,
 Canadian Space Agency
 6767, route de l'Aéroport
 SaintHubert
 Quebec J3Y 8Y9
 Canada
 Tel: +1 450 926 4688
 Fax: +1 450 926 4695
 jeanclaude.piedboeuf@space.gc.ca

Mr. M.E. Price,
 Defence Evaluation and Research Agency
 Room 1004
 Farnborough
 Hampshire GU14 0LX
 United Kingdom
 Tel: +44 1252 392531
 Fax: +44 1252 396331
 meprice@scs.dera.gov.uk

Mr. Z. Pronk,
 NLR
 Voorsterweg 31
 NL-8316 PR Marknesse
 The Netherlands
 Tel: +31 527 248223
 Fax: +31 527 248210
 pronk@nlr.nl

Mr. P. Putz,
European Space Agency (TOS MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5814
Fax: +31 71 565 5419
pputz@estec.esa.nl

Mr. K. Rajan,
NASA Ames Research Center
Mail Stop 2692
Moffett Field,
Ca 94035
USA
Tel: +1 650 604 0573
Fax: +1 650 604 3594
kanna@ptolemy.arc.nasa.gov

Mr. B. Real Planells,
European Space Agency
Robert Bosch Strasse 5
D-64293 Darmstadt
Germany
Tel: +49 6151 902 123
Fax: +49 6151 903.010
bplanell@esoc.esa.de

Mr. D.A. Rey,
Canadian Space Agency
6767, route de l'Aéroport
SaintHubert
Quebec J3Y 8Y9
Canada
Tel: +1 450 926 4690
Fax: +1 450 926 4695
Daniel.Rey@space.gc.ca

Mr. F. Richard,
Alcatel Space Industries
26, Av. J.F. Champollion
BP 1187
F-31037 Toulouse Cedex
France
Tel: +33 5 3435 5213
Fax: +33 5 3435 6169
francois.richard@space.alcatel.fr

Mr. M. Rieschel,
European Space Agency (TOS-MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5623
Fax: +31 71 565 5545
mriesche@estec.esa.nl

Mr A.C. Rolfe,
JET Joint Undertaking
J20.01
Abingdon,
Oxfordshire OX14 3EA
United Kingdom
Tel: +44 1235 464.502
Fax: +44 1235 465.373
alan.rolfe@jet.uk

Mr. E. Romani,
DATASPAZIO
Via Laurentina 756
I-00143 Roma
Italy
Tel: +39 6 50.03.7214
Fax: +39 6 50.22.331
enrico.romani@dataspazio.it

Mr. J. Rossmann,
Institute of Robotics Research (IRF)
OttoHahnStr. 8
D-44227 Dortmund
Germany
Tel: +49 231 755.4656
Fax: +49 231 755.4653
rossmann@irf.de

Mr. S.L. Roumeliotis,
University of Southern California
Department of Electrical Eng.
Los Angeles,
CA 90089
USA
Tel: +1 213 740 2224
Fax: +1 213 740 7512
stergios@robotics.usc.edu

Mr. N. Rouquette,
Jet Propulsion Laboratory
Maik Stop 301-270
4800 Oak Grove Drive
Pasadena, CA 91109-8099
USA
Tel: +1 818 354 9600
Fax: +1 818 393 6004
Nicolas.Rouquette@jpl.nasa.gov

Mr. H. Schindler,
Matra Marconi Space
31, Av. des Cosmonautes
Z.I. du Palays
F-31402 Toulouse CEDEX 4
France
Tel: +33 5 62.19.78.07
Fax: +33 5 62 19 77 14
herve.schindler@hs.mms.fr

Mr. H. Schoonmade,
National Aerospace Laboratory (NLR)
P.O. Box 153
NL-8300 AD Emmeloord
The Netherlands
Tel: +31 527 248.219
Fax: +31 527 248.210
mschoonm@nlr.nl

Mr. R. Seiler,
European Space Agency (TOS-YMM)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5551
Fax: +31 71 565 5637
rseiler@estec.esa.nl

Mr. L. Serafini,
European Space Agency (TOS-MPE)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 5653014
lserafin@estec.esa.nl

Mr. J.B. Serrano Martinez,
GMV S.A.
c/ Isaac Newton, 11
P.T.M. Tres Cantos
28760 Madrid
Spain
Tel: +34 91 807 2100
Fax: +34 91 807 2199
jbserrano@gmv.es

Ms. K. Shillcutt,
Carnegie Mellon University
Field Robotics Center
5000 Forbes Avenue
Pittsburgh,
PA 15213
USA
Tel: +1 412 268 7086
Fax: +1 412 268 5895
kimberly@ri.cmu.edu

Mr. R. Siegart,
Swiss Federal Institute of Technology
Autonomous Systems Lab
EPFL-DMTISR
CH 1015 Lausanne
Switzerland
Tel: +41 21 693 3825
Fax: +41 21 693 3866
roland.siegwart@epfl.ch

Mr. S. Siewert,
University of Colorado
Space Grant College
Campus Box 520
803090520
Boulder CO
USA
Tel: +1 303 939 4676
Fax: +1 303 939 6446
ssiewert@ball.com

Mr. C. Simon,
Matra Marconi Space
31, Ave. des Cosmonautes
Z.I. du Palays
F-31402 Toulouse CEDEX 4
France
Tel: +33 6219 7097
christian.simon@mms.tls.fr

Mr. M.T.J. Spaan,
Universiteit van Amsterdam
H. Cleijndertweg 25 k2
NL-1025 DE Amsterdam
The Netherlands
Tel: +31 20 637 2302
mtjspaans@wins.uva.nl

Mr. C. Stavrinidis,
European Space Agency (TOS-M)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3993
cstavrinides@estec.esa.nl

Mr. L. Steinicke,
Space Applications Services
Leuvensesteenweg 325
B-1932 Zaventem
Belgium
Tel: +32 2 721 5484
Fax: +32 2 721 5444
ls@sas.be

Mr. Z. Sun,
Tsinghua University
Dept. of Computer Science & Tech.
100084 Beijing
Peoples Republic of China
Tel: +86 10 6278 8939
Fax: +86 10 6277 1138
szqdc@mail.tsinghua.edu.cn

Mr. A. Suskind,
Verhaert Design and Development N.V.
Hogenakkerhoekstraat 9
B-9150 Kruikebeke
Belgium
Tel: +32 3 250 1414
Fax: +32 3 253 1464
alain.suskind@verhaert.com

Mr. K. Takadama,
ATR Human Information Processing
Research Labs.
4105, 12 Kabutodai, Kizucho, Sourakugun
Kyoto 6190224
Japan
Tel: +81 774 75 2665
Fax: +81 774 75 2665
keiki@hip.atr.co.jp

Mr. J-L. Terrailon,
European Space Agency (TOS-WSD)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3765
Fax: +31 71 565 4295
Jeanloup@ws.estec.esa.nl

Mr. P. Torasso,
Universita' di Torino
C.so Svizzera 185
10149 Torino
Italy
Tel: +39 011.670.6709
Fax: +39 011 751.603
torasso@di.unito.it

Mr. J.R. Torley,
University of Colorado
1169 Berglund Road
Colorado Springs,
CO 80920 USA
Tel: +1 319 598 3599
jtorley@mail.uccs.edu

Mr. K. Tsuchiya,
Kyoto University
SakyoKu
Yoshida HonMachi,
6068501 Kyoto
Japan
Tel: +81 774 38 3960
Fax: +81 774 38 3960
tsuchiya@space.kuaero.kyotou.ac.jp

Mr. S-I. Tsuda,
Tokai University
1117 Kiatakaname, Hiratsuka
Kanagawa 25912
Japan
Tel: +81 463 58 1211
Fax: +81 463 50 2060
stsuda@keyaki.cc.utokai.ac.jp

Mr. K. Tsujita,
Kyoto University
Dept. of Aeronautics & Astron.
SakyoKu
Yoshida HonMachi,
6068501 Kyoto
Japan
Tel: +81 774 38 3961
Fax: +81 775 38 3962
tsujita@space.kuaero.kyotou.ac.jp

Mr. E. Tunstel,
Jet Propulsion Laboratory
Mail Stop 107102
4800 Oak Grove Drive
Pasadena, CA 91109-8099
USA
Tel: +1 818 393 2666
Fax: +1 818 354 8172
tunstel@robotics.jpl.nasa.gov

Mr. G. Ulbrich,
European Space Agency (TOS-MMO)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 5515
Fax: +31 71 565 5430
gulbrich@estec.esa.nl

Mr. Y. Umetani,
Toyota Technical Institute
2121, Hisakata
Tempakuku
4688511 Nagoya
Japan
Tel: +81 52 809 1790
Fax: +81 52 809 1794
umetani@toyotati.ac.jp

Mr. M.H.I. van den Briel,
University of Maastricht
Lammergierstraat 48
NL-6215 AJ Maastricht
The Netherlands
Tel: +31 43 347 2159
mhl.vandenbriel@student.unimaas.nl

Mr. R.J.N. van der Holst,
European Space Agency (TOS-MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3276
rvdholst@estec.esa.nl

Mr. E. van Meyeren,
Fokker Space B.V.
Newtonweg 1
Postbus 32070
NL-2303 DB Leiden
The Netherlands
Tel: +31 71 524 5097
e.van.meijeren@fokkerspace.nl

Mr. M. Van Winnendael,
European Space Agency (TOS-MMA)
Keplerlaan 1
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3510
Fax: +31 71 565 5419
mvwinnen@estec.esa.nl

Mr. G. Vassura,
University of Bologna
viale Risorgimento 2
I-40136 Bologna
Italy
Tel: +39 051 644.3458
Fax: +39 051 644.3412
gabriele.vassura@mail.ing.unibo.it

Mr. T. Vaz Maia,
European Space Agency (TOS-MMA)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 3961
tvazmaia@estec.esa.nl

Mr. M. Vergauwen,
Catholic University of Leuven
ESATPSI
Kardinaal Mercierlaan 94
B-3001 Heverlee
Belgium
Tel: +32 16 321064
Fax: +32 16 321723
maarten.vergauwen@esat.kuleuven.ac.be

Mr. G. Visentin,
European Space Agency (TOS-WKR)
Postbus 299
NL-2200 AG Noordwijk
The Netherlands
Tel: +31 71 565 4835
gvisentin@estec.esa.nl

Ms. S. Wakabayashi,
National Aerospace Laboratory
7441 Jindaijihigashimachi, Chofu
1828522 Tokyo
Japan
Tel: +81 422 40 3171
Fax: +81 422 40 3148
waka@nal.go.jp

Mr. Y. Wakabayashi,
NASDA Tsukuba Space Center
211 Sengen TsukubaShi
Ibaraki, 305
Japan
Tel: +81 298 52 2369
Fax: +81 298 52 2415
Wakabayashi.Yasufumi@nasda.go.jp

Mr. S. Walther,
DaimlerChrysler Aerospace AG
Dept. RI012
RaumfahrtInfrastruktur
Postfach 286156
D-28361 Bremen
Germany
Tel: +49 421 539 4138
Fax: +49 421 539 4155
stephan.Walther@ri.dasa.de

Mr. R. Ward,
Science Systems (Space) Ltd
23 Clothier Road,
Brislington
Bristol BS4 5SS
United Kingdom
Tel: +44 117 971.7251
Fax: +44 117 971.1125
roger.ward@scisys.com

Mr. Y. Watanabe,
NASDA Tsukuba Space Center
211 Sengen TsukubaShi
Ibaraki, 305
Japan
Tel: +81 298 592.967
Fax: +81 298.522.410
watanabe.yasuyuki@nasda.go.jp

Mr. C.R. Weisbin,
Jet Propulsion Laboratory
Mail Stop 179-224
4800 Oak Grove Drive
Pasadena, CA 91109-8099
USA
Tel: +1 818 354 2013
Fax: +1 818 393 3602
charles.r.weisbin@jpl.nasa.gov

Mr. D. Whitton,
MacDonald Dettwiler
84 Tremblay
Ste. Anne de Bellevue
Que 49x375
Canada
Tel: +1 514 457 0003
Fax: +1 514 457 0010
dwhitton@mdrobotics.ca

Mr. C.M. Wilklow,
University of Colorado
Colorado Space Grant Consort.
Campus Box 520
Boulder
CO 80309-0520
USA
Tel: +1 303 492 4383
Fax: +1 303 492 5456
wilklow@colorado.edu

Mr. R.K. Wilson,
Jet Propulsion Laboratory
Mail Stop 301-250D
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 354 1128
Fax: +1 818 393 5074
robert.k.wilson@jpl.nasa.gov

Mr. E.J. Wyatt,
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 354 1414
Fax: +1 818 393 6004
e.j.wyatt@jpl.nasa.gov

Mr. J. Yen,

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena,
CA 91109-8099
USA
Tel: +1 818 354 4744
Fax: +1 818 393 5007
Jeng.Yen@jpl.nasa.gov

Mr. K. Yoshida,

Tohoku University
Dept. Aeronautics and Space Eng.
Aoba 01, Sendai 9808579
Japan
Tel: +81 22 217 6992
Fax: +81 22 217 6992
yoshida@astro.mech.tohoku.ac.jp

Mr. T. Yoshimitsu,

Inst. of Space & Astronautical Science
311, Yoshinodai, Sagamihara
Kanagawa 2298510
Japan
Tel: +81 42 759 8311
Fax: +81 42 759 8304
kikko@nsl.isas.ac.jp

Mr. M. Zelikman,

Space Systems Finland
Kappelitie 6
SF-02200 Espoo
Finland
Tel: +358 400 704.839
Fax: +358 961.328.699
mika.zelikman@ssf.fi

Mr. P. Zetocho,

Air Force Research Laboratory/VSSS
3550 Aberdeen Ave SE
Kirtland AFB,
NM 87117-5776
USA
Tel: +1 505 853 4114
Fax: +1 505 846 6053
zetocho@plk.af.mil

Mr E. Zhang,

National University of Defence
Technology
4th Department
Changsha Hunan China 410073
Peoples Republic of China
Tel: +86 731 4506 400
Fax: +86 731 455 3549

European Space Agency
Agence spatiale européenne

Contact: ESA Publications Division
c/o ESTEC, PO Box 299, 2200 AG Noordwijk, The Netherlands
Tel (31) 71 565 3400 - Fax (31) 71 565 5433