(97)-2927

cc: PP.
EM.
IP.

on 28/01/97

*Put on Esrin Server for THS on 27/01/97 by PP

ENVISAT-1 MISSION CFI SOFTWARE

PPF_ORBIT SOFTWARE USER MANUAL

PO-IS-GMV-GS-0558

Prepared by: L.J. Alvarez (GMV)
E. Lopez (GMV)

Issue: 4.1
Date: January 15th, 1997

(97)-2927

cc: PP.
EM.
IP.

on 28/01/97

*Put on Esrin Server for THS on 27/01/97 by PP

ENVISAT-1 MISSION CFI SOFTWARE

PPF_ORBIT SOFTWARE USER MANUAL

PO-IS-GMV-GS-0558

Prepared by: L.J. Alvarez (GMV)
E. Lopez (GMV)

Issue: 4.1
Date: January 15th, 1997

Document Status Sheet

Issue	Changes	Date	Reason
4.1		15 January 1997	First issue

GMV, S.A.

P.T.M
c/Isaac Newton 11
TRES CANTOS
E-28760 Madrid SPAIN

Tel: +34-1-807 21 00
Fax: +34-1-807 21 99
Tlx: 48487 GMEV E

TABLE OF CONTENTS

1 SCOPE

The Software User Manual (SUM) of the Envisat-1 mission CFI software is composed of

- a general document describing the sections common to all the CFI software libraries
- a specific document for each of those libraries.

This document is the PPF_ORBIT Software User Manual. It provides a detailed description of the use of the CFI functions included within the PPF_ORBIT CFI software library.

2 ACRONYMS AND NOMENCLATURE

2.1 Acronyms

AOCS	Attitude and Orbit Control System
ANX	Ascending Node Crossing
CFI	Customer Furnished Item
CS	Coordinate System
DRS	Data Relay Satellite
ESA	European Space Agency
ESTEC	European Space Technology and Research Centre
FOS	Flight Operation Segment
GS	Ground Station
H/W	Hardware
I/F	Interface
LOS	Line Of Sight
PPF	Polar Platform
RAM	Random Access Memory
SBT	Satellite Binary Time
SUM	Software User Manual
S/W	Software
UTC	Universal Time Coordinated
UT1	Universal Time UT1
SSP	Sub Satellite Point

2.2 Nomenclature

CFI	A group of CFI functions, and related software and documentation, that will be distributed by ESA to the users as an independent unit
CFI function	A single function within a CFI that can be called by the user
Library	A software library containing all the CFI functions included within a CFI plus the supporting functions used by those CFI functions (transparently to the user)

3 APPLICABLE AND REFERENCE DOCUMENTS

3.1 Applicable documents

AD 1 Re-engineering of Mission Analysis Software for Envisat-1: Statement of Work.PO-SW-ESA-GS-0344. ESA/ESTEC/NW. Issue 3.0. 19/12/1995.

AD 2 ESA Software Engineering Standards. ESA PSS-05-0. ESA. Issue 2. February 1991

3.2 Reference documents

RD 1 Envisat-1 Mission CFI Software Description and Interface Definition Document. PO-ID-ESA-SY-00412

RD 2 Envisat-1 Mission CFI Software. Mission Conventions Document. PO-IS-GMV-GS-0561

RD 3 Envisat-1 Mission CFI Software General Software User Manual. PO-IS-GMV-GS-0556

RD 4 Envisat-1 Mission CFI Software PPF_LIB Software User Manual. PO-IS-GMV-GS-0557

RD 5 Envisat-1 Products Specifications. PO-RS-MDA-GS-2009

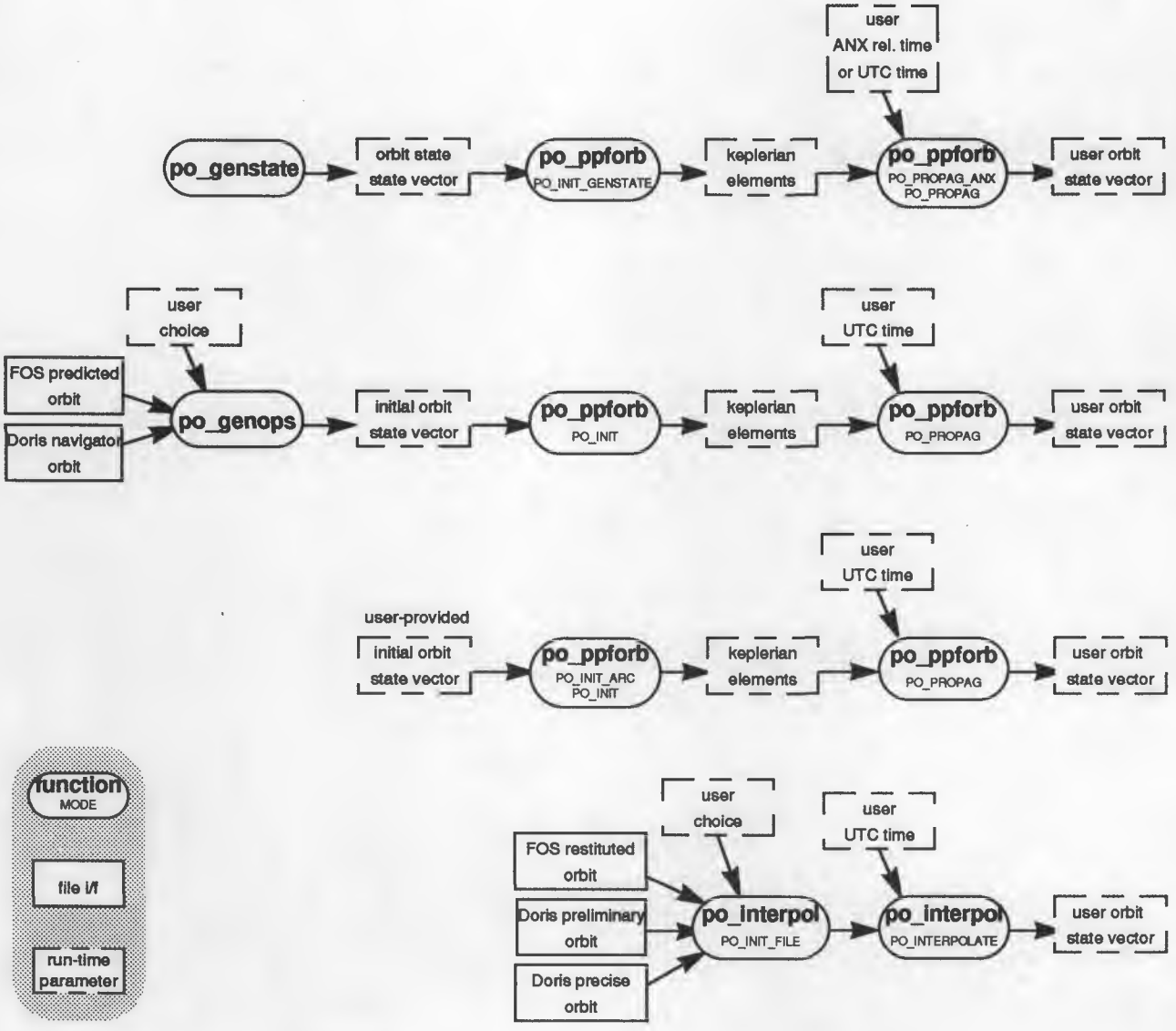
4 INTRODUCTION

This software library contains all the CFI functions allowing accurate computation of orbit state vectors, either at ascending node or (by propagation) at any point in the Envisat-1 orbit.

It includes the Envisat-1 orbit interpolator, orbit propagator and several routines used to feed the propagator with either simulated, predicted or restituted initial state vectors, i.e. the following CFI functions:

- **po_genstate**: This software is an orbit state vector generator which generates state vectors to perform orbit simulations. It generates a cartesian state vector around the true ascending node crossings in function of the date (processing time), the longitude of the ascending node, the satellite Repeat Cycle Length, and the mean local solar time. This state vector can then be fed to the propagator **po_ppforb** in order to initialize the propagation.
- **po_ppforb**: This software is a propagator which allows accurate prediction of osculating Cartesian and Kepler state vectors for user requested time segments, plus the calculation of orbit, and satellite ground track related physical parameters, as well as the direction towards the Sun and the Moon as seen from the satellite position. Input data are specified in an Earth fixed frame.
- **po_genops**: This software generates Envisat-1 Cartesian State Vectors for satellite operations using either Flight Dynamics predicted ascending node state vectors, or the DORIS Navigator Data (extracted in near real time at processing station PDHS). The software function processes and filters the DORIS navigator level 0 data to verify its usability to create a DORIS generated Cartesian State Vector. Otherwise, in case the level 0 DORIS data are too degraded by noise, the software selects the Flight Dynamics predicted Cartesian State vector. The selected state vector can then be fed to the propagator **po_ppforb** in order to initialize the propagation.
- **po_interpol**: This software generates off-line Extended Envisat-1 Cartesian State Vectors based on the interpolation of orbit restituted state vector files (DORIS Preliminary orbit, DORIS precise orbit, ESOC restituted orbit), selectable by the user. The user defines the UTC time for which an interpolated state vector has to be generated. There is no need to use the propagator in conjunction with **po_interpol**.

The possible sequences of calls allowing to produce an orbit state vector are shown in the following dataflow diagram:



A detailed description of each function is provided in section 7. Please refer also to:

- RD 2 for a detailed description of the time references and formats, coordinate systems, parameters and models used in this document
- RD 3 for a complete overview of the CFI, and in particular the detailed description of error handling functions

5 LIBRARY INSTALLATION

For a detailed description of the installation of any CFI library, please refer to RD 3.

Notes that example data files are provided with this CFI:

- orbit files to be used with **po_genops**
- orbit files to be used with **po_interpol**

These files are orbit file examples, all relevant for the PDS V1 test reference orbit.

6 LIBRARY USAGE

Note that to use the PPF_ORBIT software library, the following other CFI software libraries are required:

- `PPF_LIB` (see RD 4).

To use the PPF_ORBIT software library in a user application, that application must include in his source code either:

- `ppf_orbit.h` (for a C application)
- `ppf_orbit.inc` (for a Fortran application)

To link correctly his application, the user must include in his linking command flags like (assuming *cfi_libs_dir* and *cfi_include_dir* are the directories where respectively all CFI libraries and include files have been installed, see RD 3 for installation procedures):

`-I`*cfi_include_dir* `-L`*cfi_lib_dir* `-lppf_orbit -lppf_lib`

All functions described in this document have a name starting with the prefix `po_`

To avoid problems in linking a user application with the `PPF_ORBIT` software library due to the existence of names multiple defined, the user application should avoid naming any global software item beginning with either the prefix `PO_` or `po_`.

To preserve compatibility with the historical CFI function names, it is possible to call the CFI functions described in this document from a user application with or without the `po_` prefix. This does not apply to the error handling functions, which are described in the General SUM (see RD 3).

This is summarized in the table below.

Function Name	Enumeration value	int
Main CFI Functions		
po_genstate genstate	PO_GENSTATE_ID	0
po_ppforb ppforb	PO_PPFORB_ID	1
po_genops genops	PO_GENOPS_ID	2
po_interpol interpol	PO_INTERPOL_ID	3
Error Handling Functions		
po_verbose	not applicable	
po_silent		
po_vector_code		
po_vector_msg		
po_print_msg		

Notes about the table:

- to transform the status vector returned by a CFI function to either a list of error codes or list of error messages, the enumeration value (or the corresponding integer value) described in the table must be used
- the error handling functions have no enumerated values

7 CFI FUNCTIONS DESCRIPTION

The following sections describe each CFI function.

The calling interfaces are described both for C users and Fortran users.

Input and output parameters of each CFI function are described in tables, where C programming language syntax is used to specify:

- parameter types (e.g. long, double)
- array sizes of N elements (e.g. param[N])
- array element M (e.g. [M])

Fortran users should adapt the tables using Fortran syntax equivalent terms:

- parameter types (e.g. long <=> INTEGER*4, double <=>REAL*8)
- array sizes of N elements (e.g. param[N] <=> param (N))
- array element M (e.g. [M] <=> (M+1))

7.1 po_genstate

7.1.1 Overview

This software is an orbit state vector generator which is to be used in conjunction with the **po_ppforb** orbit propagator function, in order to generate state vectors to perform orbit simulations. It generates a cartesian state vector around the true ascending node crossings in function of:

- the date (processing time),
- the longitude of the ascending node,
- the satellite Repeat Cycle and Cycle Length, and
- the mean local solar time at ascending node

This state vector is fed to the propagator in order to initialize the propagation (i.e. use **po_ppforb** with mode PO_INIT_GENSTATE).

7.1.2 Calling interface

The calling interface of the **po_genstate** CFI function is the following (input parameters are underlined):

```
#include <ppf_orbit.h>
{
      long irep, icyc, iorb0, iorb, ierr[4], status;
      double mjdr[2], rlong, ascmlst, mjdp[2], pos[3], vel[3];

      status = po_genstate(mjdr, &irep, &icyc, &rlong, &ascmlst,
                           &iorb0,&iorb,
                           mjdp, pos, vel, ierr);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include <ppf_orbit.inc>

      INTEGER*4 IREP, ICYC, IORB0, IORB, IERR(4), STATUS
      REAL*8 MJDR(2), RLONG, ASCMLST, MJDP(2), POS(3), VEL(3)

      STATUS = PO_GENSTATE(MJDR, IREP, ICYC, RLONG, ASCMLST,
     &                     IORB0, IORB,
     &                     MJDP, POS, VEL, IERR)
```

7.1.3 Input parameters

The **po_genstate** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mjdr[2]	double	[0]	UTC of the reference orbit (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 of the reference orbit (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
irep	long *	-	Repeat cycle of the reference orbit	days	>= 3
icyc	long *	-	Cycle length of the reference orbit	orbits	>= 43
rlong	double *	-	Geocentric longitude of the [Earth fixed] ascending node (Earth fixed CS)	deg	>= 0 < 360
ascmlst	double *	-	Mean local solar time at ascending node	decimal hours	>= 0 < 24
iorb0	long *	-	Absolute orbit number of the reference orbit	orbits	-
iorb	long *	-	Absolute orbit number of the requested orbit	orbits	-

7.1.4 Output parameters

The output parameters of the **po_genstate** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_genstate	long	-	Main status flag	-	-1, 0, +1
mjdp[2]	double	[0]	UTC of the calculated osculating state vector (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 of the calculated osculating state vector (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
pos[3]	double	all	Calculated osculating position vector (X, Y, Z) (Earth fixed CS)	m	-
vel[3]	double	all	Calculated osculating velocity vector (X, Y, Z) (Earth fixed CS)	m/s	-
ierr[4]	long	all	Status vector	-	-

7.1.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **po_genstate** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_ORBIT software library **po_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **po_genstate** CFI function by calling the function of the PPF_ORBIT software library **po_vector_code** (see RD 3).

Error type	Error message	Cause and Impact	Error code	Error No
ERR	Computed state vector does not meet tolerance requirements.	Output Envisat-1 state vector does not satisfy the loose tolerance requirements.	PO_CFI_GENSTATE_FAILED_TESTSTATE_ERR	6
ERR	Wrong input parameters	Not allowed mjdr[0] value Calculation not performed	PO_CFI_INVALID_MDJR_0_ERR	32
ERR	Wrong input parameters	Not allowed mjdr[1] value Calculation not performed	PO_CFI_INVALID_MDJR_1_ERR	33
ERR	Wrong input parameters	Not allowed irep value Calculation not performed	PO_CFI_INVALID_IREP_ERR	34
ERR	Wrong input parameters	Not allowed irep value Calculation not performed	PO_CFI_INVALID_ICYC_ERR	35
ERR	Wrong input parameters	Not allowed rlong value Calculation not performed	PO_CFI_INVALID_LONGITUDE_ERR	36
ERR	Wrong input parameters	Not allowed ascmlst value Calculation not performed	PO_CFI_INVALID_MLST_ERR	37
ERR	Wrong input parameters	Not allowed iorb0 value Calculation not performed	PO_CFI_INVALID_IORB0_ERR	38
ERR	Wrong input parameters	Not allowed iorb value Calculation not performed	PO_CFI_INVALID_IORB_ERR	39

7.1.6 Runtime performances

The following runtime performances have been measured.

Sparcstation 20 [ms]	IBM RS6000 [ms]

7.2 po_ppforb

7.2.1 Overview

This software is a propagator which allows accurate prediction of osculating Cartesian and Kepler state vectors for user requested time segments, plus the calculation of orbit, and satellite ground track related physical parameters, as well as the direction towards the Sun and the Moon as seen from the satellite position. Input data are specified in an Earth fixed frame.

The detailed functions of the Orbit Propagator are:

- For propagation over complete orbits:
  - 1. Accurate prediction of osculating Orbit State Vectors in the Earth-fixed rotating coordinate system and osculating Kepler states in the Quasi-Inertial geocentric coordinate system of the Quasi-mean system of date, for user requested relative or absolute times which must fall within plus/minus two orbital periods of the reference node crossing epoch. This node crossing epoch is being obtained by the initialization using an osculating Orbit State Vector in the Earth-fixed rotating coordinate system at, or near (+/- 5 deg) true node, as provided by the FOCC.
  - 2. Calculation of orbit, altitude and ground track related quantities as well as the direction to Sun and Moon as seen from the S/C position, along the whole orbit.
- For propagation over short restituted orbit arcs:
  - 1. Accurate prediction of osculating Orbit State Vectors in the Earth-fixed rotating coordinate system and osculating Kepler states in the Quasi-Inertial geocentric coordinate system of the Quasi-mean system of date, for user requested relative or absolute times which must fall within plus/minus 3 minutes of a restituted Orbit State Vector epoch. Now the constraint for the initialization using an Orbit State Vector at or near true ascending node is not applicable. The restituted Orbit State Vector is an osculating Cartesian State Vector in the Earth-fixed rotating coordinate system, as provided by the FOCC.
  - 2. Calculation of orbit, altitude and ground track related quantities as well as the direction to Sun and Moon as seen from the S/C position, along this arc.

A proper sequence of Orbit Propagator calls consist of:

- one initialization call
- a number of propagation calls

7.2.2 Calling interface

The calling interface of the **po_ppforb** CFI function is the following:

```
#include <ppf_orbit.h>
{
      long mode, ierr[4], status;
      double mjdr[2], xm[6], mjdp[2], x[6], pos[3], vel[3], acc[3];
      double res[54];

      status = po_ppforb(&mode, mjdr, xm, mjdp, x,
                         pos, vel, acc, res, ierr);
}
```

For Fortran programs the declaration and calling procedure is as follows (note that the C preprocessor must be used because of the presence of the #include statement):

```
#include <ppf_orbit.inc>

      INTEGER*4 MODE, IERR(4), STATUS
      REAL*8 MJDR(2), XM(6), MJDP(2), X(6), POS(3), VEL(3), ACC(3)
      REAL*8 RES(54)

      STATUS = PO_PPFORB(MODE, MJDR, XM, MJDP, X,
     &                   POS, VEL, ACC, RES, IERR)
```

CAUTION: Note that for historical reasons, some parameters are both input and output parameters. So please read very carefully the following sections.

7.2.3 Input parameters

The next table shows the allowed combinations of INITIALISATION and PROPAGATION modes.

Initialization (I) mode ⇒ Propagation (P) mode	Initial state vector (Reference)	Propagation mode (see RD 2)	Maximum propagation time
I = -2 / 98 ⇒ P = 2 / 102	Near centre of the local orbital arc (Earth fixed CS)	Longitude independent	3 min wt the initial orbit state vector
I = -1 / 99 ⇒ P = 1 / 2 / 101/ 102	Within 5 deg of [Eath fixed] ascending node (Earth fixed CS)	Longitude independent	two nodal periods wrt the true [Earth fixed] ascending node
I = 0 / 100 ⇒ P = 2 / 102	Within 5 deg of tue [Earth fixed] ascending node (Earth fixed CS)	Longitude dependent	two nodal periods wrt the true [Earth fixed] ascending node

The **po_ppforb** CFI function has the following input parameters in initialisation modes:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long *		Initialization mode		-2, -1, 0 98, 99, 100
mjdp[2]	double	[0]	UTC of the initial osculating state vector (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 of the initial osculating state vector (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
pos[3]	double	all	Initial osculating position vector (X, Y, Z) (Earth fixed CS)	m	-
vel[3]	double	all	Initial osculating velocity vector (X, Y, Z) (Earth fixed CS)	m/s	-

The **po_ppforb** CFI function has the following input parameters in propagation modes:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long *	-	Propagation mode	-	1, 2 101, 102
mjdr[2]	double	[0]	UTC of the true [Earth-fixed] ascending node (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 of the true [Earth-fixed] ascending node (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
xm[6]	double	all	Mean Kepler elements at the true [Earth-fixed] ascending node (Mean of Date CS)	-	-
		[0]	Semi-major axis	m	> 0
		[1]	Eccentricity		>= 0 < 1
		[2]	Inclination	deg	>= 0 <= 180
		[3]	Right ascension of the ascending node	deg	>= 0 < 360
		[4]	Argument of perigee	deg	>= 0 < 360
		[5]	Mean anomaly	deg	>= 0 < 360
mjdp[2]	double	[0]	In mode = 1 / 101 ΔT: prediction time relative to the true [Earth fixed] ascending node (UTC time) In mode = 2 / 102 ΔT: UTC prediction time (UTC time)	 s decimal days (Processing format)	-
		[1]	DUMMY	-	-

It is also possible to use enumeration values rather than integer values for some of the input arguments, as shown in the table below:

Input	Description	Enumeration value	Int
mode	Initialization mode	PO_INIT_ARC	-2
		PO_INIT_GENSTATE	-1
		PO_INIT	0
		PO_INIT_ARC + PO_NO_RESULTS	98
		PO_INIT_GENSTATE + PO_NO_RESULTS	99
		PO_INIT + PO_NO_RESULTS	100
	Propagation mode	PO_PROPAG_ANX	1
		PO_PROPAG	2
		PO_PROPAG_ANX + PO_NO_RESULTS	101
		PO_PROPAG + PO_NO_RESULTS	102
	Optional value to be added to the mode value, in order to avoid producing the optional results vector (and therefore improve performance of the function) See above mode values for usage.	PO_NO_RESULTS	100

7.2.4 Output parameters

The output parameters of the **po_ppforb** CFI function are in initialisation modes:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_ppforb	long	-	Main status flag	-	-1, 0, +1
mjdr[2]	double	[0]	UTC of the true [Earth fixed] ascending node (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1of the true [Earth fixed] ascending node (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
xm[6]	double	all	Mean Kepler elements at the true [Earth fixed] ascending node (Mean of Date CS)	-	-
		[0]	Semi-major axis	m	> 0
		[1]	Eccentricity	-	>= 0 < 1
		[2]	Inclination	deg	>= 0 <= 180
		[3]	Right ascension of the ascending node	deg	>= 0 < 360
		[4]	Argument of perigee	deg	>= 0 < 360
		[5]	Mean anomaly	deg	>= 0 < 360
x[6]	double	all	Osculating Kepler elements of the initial state vector (Mean of Date CS)	-	-
acc[3]	double	all	Osculating acceleration vector of the initial state vector (Earth fixed CS)	m/s^2	-
res[54]	double	all	Optional results vector corresponding to the initial state vector In mode = -2 / -1 / 0 the results vector is calculated. In mode = 98 / 99 / 100 the results vector is NOT calculated, i.e. all parameters are set to 0.	-	-
ierr[4]	long	all	Status vector	-	-

The output parameters of the **po_ppforb** CFI function in propagation modes are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_ppforb	long	-	Main status flag	-	-1, 0, +1
mjdp[2]	double	[0]	UTC at predicted time (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1 at predicted time (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
x[6]	double	all	Osculating Kepler elements at predicted time (Mean of Date CS)	-	-
pos[3]	double	all	Osculating position vector at predicted time (Earth fixed CS)	m	-
vel[3]	double	all	Osculating velocity vector at predicted time (Earth fixed CS)	m/s	-
acc[3]	double	all	Osculating acceleration vector at predicted time (Earth fixed CS)	m/s^2	-
res[54]	double	all	Optional results vector corresponding to the predicted state vector In mode = 1 / 2 the results vector is calculated. In mode = 101 / 102 the results vector is NOT calculated, i.e. all its parameters are set to zero.	-	-
ierr[4]	long	all	Status vector	-	-

7.2.5 Optional results vector

The optional results vector contains many additional parameters which can be optionally calculated by the **po_ppforb** function (see input mode parameter). It contains:

Result parameter (res element)	Description (Reference)	Unit (Format)	Allowed Range
[0]	Orbit radius (Mean of Date CS)	m	>= 0
[1]	Radial orbit velocity component (Mean of Date CS)	m/s	-
[2]	Transversal orbit velocity component (Mean of Date CS)	m/s	-
[3]	Orbit velocity magnitude (Mean of Date CS)	m/s	>= 0
[4]	Right ascension of the satellite (Mean of Date CS)	deg	>= 0 < 360
[5]	Declination of the satellite (Mean of Date CS)	deg	>= -90 <= +90
[6]	Earth rotation angle [H]	deg	>= 0 < 360
[7]	Geocentric longitude of satellite and SSP (Earth fixed CS)	deg	>= 0 < 360
[8]	Geodetic latitude of satellite and SSP (Earth fixed CS)	deg	>= -90 <= +90
[9]	Geocentric latitude of the SSP (Earth fixed CS)	deg	>= -90 <= +90
[10]	Satellite centred aspect angle (Earth fixed CS)	deg	>= 0 <= 180
[11]	Geocentric aspect angle (Earth fixed CS)	deg	>= 0 <= 180
[12]	SSP centred aspect angle (Earth fixed CS)	deg	>= 0 <= 180
[13]	Geocentric distance of the SSP (Earth fixed CS)	m	>= 0
[14]	Radius of curvature parallel to meridian at the SSP (Earth fixed CS)	m	>= 0
[15]	Radius of curvature orthogonal to meridian at the SSP (Earth fixed CS)	m	>= 0
[16]	Radius of curvature along groundtrack at the SSP (Earth fixed CS)	m	>= 0

Result parameter (res element)	Description (Reference)	Unit (Format)	Allowed Range
[17]	Mean local solar time at the SSP	decimal hour	>= 0 < 24
[18]	True local solar time at the SSP	decimal hour	>= 0 < 24
[19]	True sun's (centre) right ascension (Mean of Date CS)	deg	>= 0 < 360
[20]	True Sun's (centre) declination (Mean of Date CS)	deg	>= -90 <= +90
[21]	True Sun's semi-diameter	deg	>= 0
[22]	Moon's (centre) right ascension (Mean of Date CS)	deg	>= 0 < 360
[23]	Moon's (centre) declination (Mean of Date CS)	deg	>= -90 <= +90
[24]	Moon's semi-diameter	deg	>= 0
[25]	Area of Moon lit by Sun		>= 0
[26]	Satellite eclipse flag 0 = No 1 = Yes		0, 1
[27]	Geodetic altitude of the satellite (Earth fixed CS)	m	-
[28]	Geodetic latitude rate of satellite and SSP (Earth fixed CS)	deg/s	-
[29]	Geocentric latitude rate of the SSP (Earth fixed CS)	deg/s	-
[30]	Geocentric longitude rate of satellite and SSP (Earth fixed CS)	deg/s	-
[31]	Right ascension rate of the satellite (Mean of Date CS)	deg/s	-
[32]	Geocentric distance rate of the SSP (Earth fixed CS)	m/s	-
[33]	Radius of curvature parallel to meridian rate at the SSP (Earth fixed CS)	m/s	-
[34]	Radius of curvature orthogonal to meridian rate at the SSP (Earth fixed CS)	m/s	-
[35]	Geodetic altitude rate of the satellite (Earth fixed CS)	m/s	-

Result parameter (res element)	Description (Reference)	Unit (Format)	Allowed Range
[36]	Northward component of the velocity relative to the Earth of the SSP (Topocentric CS)	m/s	-
[37]	Eastward component of the velocity relative to the Earth of the SSP (Topocentric CS)	m/s	-
[38]	Magnitude of the velocity relative to the Earth of the SSP (Topocentric CS)	m/s	>= 0
[39]	Azimuth of the velocity relative to the Earth of the SSP (Topocentric CS)	deg	>= 0 < 360
[40]	Geodetic latitude rate-rate of satellite and SSP (Earth fixed CS)	deg/s^2	-
[41]	Geocentric latitude rate-rate of the SSP (Earth fixed CS)	deg/s^2	-
[42]	Geocentric longitude rate-rate of satellite and SSP (Earth fixed CS)	deg/s^2	-
[43]	Right ascension rate-rate of the satellite (Mean of Date CS)	deg/s^2	-
[44]	Geocentric distance rate-rate of the SSP (Earth fixed CS)	m/s^2	-
[45]	Radius of curvature parallel to meridian rate-rate at the SSP (Earth fixed CS)	m/s^2	-
[46]	Radius of curvature orthogonal to meridian rate-rate at the SSP (Earth fixed CS)	m/s^2	-
[47]	Geodetic altitude rate-rate of the satellite (Earth fixed CS)	m/s^2	-
[48]	Northward component of the acceleration relative to the Earth of the SSP (Topocentric CS)	m/s^2	-
[49]	Eastward component of the acceleration relative to the Earth of the SSP (Topocentric CS)	m/s^2	-
[50]	Groundtrack tangential component of the acceleration relative to the Earth of the SSP (Topocentric CS)	m/s^2	-
[51]	Azimuth of the acceleration relative to the Earth of the SSP (Topocentric CS)	deg	>= 0 < 360
[52]	Nodal period	s	>= 0

Result parameter (res element)	Description (Reference)	Unit (Format)	Allowed Range
[53]	UTC of next ascending node (UTC time)	decimal days (Processing format)	-

7.2.6 Warnings and errors

Next table lists the possible error messages that can be returned by the **po_ppforb** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_ORBIT software library **po_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **po_ppforb** CFI function by calling the function of the PPF_ORBIT software library **po_vector_code** (see RD 3).

Error type	Error message	Cause and impact	Error code	Error No
ERR	Invalid initial state vector	Initial Envisat-1 state vector does not satisfy the loose tolerance requirements. Calculation not performed	PO_CFI_PPFORB_INIT_INVALID_PPF_SV_ERR	0
ERR	Invalid propagated state vector	Propagated Envisat-1 state vector does not satisfy the loose tolerance requirements.	PO_CFI_PPFORB_PRED_INVALID_PROP_SV_ERR	33
ERR	Wrong input parameters	Not allowed mode value Calculation not performed	PO_CFI_PPFORB_NOT_VALID_MODE_ERR	96
ERR	Wrong input parameters	po_ppforb needs to be initialized before propagated Calculation not performed	PO_CFI_PPFORB_NOT_INITIALIZED_ERR	97

7.2.7 Runtime performances

The following runtime performances have been measured (TBC).

mode	Sparcstation 20 [ms]	IBM RS6000 [ms]
-2		
-1		
0		
1		
2		
98		
99		
100		
101		
102		

7.3 po_interpol

7.3.1 Overview

The **po_interpol** function is used by the Envisat-1 ground segment off-line processing chains to compute an Envisat-1 Cartesian state vector and its associated parameters (e.g. geolocation parameters) at a requested time, using either (selectable by the user):

- Doris Precise Orbit product file(s)
- Doris Preliminary Orbit product file(s)
- ESOC Restituted Orbit file(s)

For each of the 3 types of Orbit file above, one or more files can be used by **po_interpol** to cope with processing of products which span a time interval covered by more than a single Orbit file.

The three sources of data above are listed by decreasing order of orbital precision achieved. In normal processing, the **po_interpol** software will use the most precise source available at the time of processing. Optionally, usage of a specific type of file can be enforced.

In order to avoid reading the source file(s) at each call, **po_interpol** has 2 modes of operation:

- INITIALISATION, which reads the appropriate file(s) and sets up internal data; it is performed only once.
- INTERPOLATION, which ignores the source file(s) and uses the internal data to produce the requested Cartesian state vector; it´s to be performed as often as needed for a given processing session.

It is the responsibility of the software calling **po_interpol** to prepare the input files, i.e. to ensure that files relevant for the processing being performed are available. See RD 5 for a description of the files.

CAUTION: Note that to guarantee correct interpolation, the file(s) made available must cover the time span of the product being processed, PLUS 4 minutes before and 4 minutes after. This will avoid degradation of the interpolation at the edges of the product time interval.

In order to guarantee continuous and correct interpolation over the whole processing session, the initialisation call uses as inputs the intended start UTC time and end UTC time of the Envisat-1 product to be processed off-line. This allows **po_interpol** to select not only the most precise source available, but also the source file(s) which spans the needed time interval.

The process calling **po_interpol** is responsible for identifying the number and names of input file(s) required. This must be deducted from the time interval of the product to be processed off-line and from the available Orbit files.

The output format of the data generated by **po_interpol** at the required time is identical to the output format of the **po_ppforb** software.

A proper sequence of **po_interpol** calls consist of:

- initialization call (mode PO_INIT_FILE)
- subsequent interpolation calls (mode PO_INTERPOLATE)

7.3.2 Calling interface

The calling interface of the **po_interpol** CFI function is the following (input parameters are underlined[1]):

```
#include <ppf_orbit.h>
{
      long mode, choice, ndc, ndp, ner, selected, ierr[10], status;
      double mjdr0, mjdr1, mjdp[2], x[6], pos[3], vel[3],acc[3],
             res[54];
      char **doris_precise_file, **doris_prelim_file;
      char **esoc_rest_file;

      status = po_interpol(&mode,&choice,
                           &ndc, doris_precise_file,
                           &ndp, doris_prelim_file,
                           &ner, esoc_rest,
                           &mjdr0, &mjdr1,
                           mjdp, x, pos, vel, acc,
                           &selected, res, ierr);
}
```

1. Number of files, filenames and choice are not used in Interpolation mode (mode 2 or 102)

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include <ppf_orbit.inc>

      INTEGER*4 MODE, CHOICE, NDC, NDP, NER, SELECTED,
     &          IERR(10), STATUS
      REAL*8 MJDR0, MJDR1, MJDP(2), X(6), POS(3), VEL(3), ACC(3),
     &RES(54)
      CHARACTER*LENGTH_NAME DORIS_PRECISE_FILE(NUM_FILES)
      CHARACTER*LENGTH_NAME DORIS_PRELIM_FILE(NUM_FILES)
      CHARACTER*LENGTH_NAME ESOC_REST_FILE(NUM_FILES)

      STATUS = PO_INTERPOL(MODE, CHOICE,
     &                     NDC, DORIS_PRECISE_FILE,
     &                     NDP, DORIS_PRELIM_FILE,
     &                     NER, ESOC_REST,
     &                     MJDR0, MJDR1,
     &                     MJDP, X, POS, VEL, ACC,
     &                     SELECTED, RES, IERR)
```

Note that NUM_FILES must be set to the number of input files of that type, i.e. NDC, NDP or NER, whereas LENGTH_NAME must be set to the maximum string length of the filenames of that type.

7.3.3 Input parameters

The **po_interpol** CFI function has the following input parameters in initialisation mode:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long *		Initialization mode		0
choice	long *		Switch to select the file 0 = automatic selection 1 = Doris Precise Orbit 2 = Doris Preliminary Orbit 3 = ESOC Restituted Orbit		0, 1, 2, 3
ndc	long *		Number of Doris Precise Orbit files on input		>= 0 <= 16
doris_precise_file	char **	all	Filenames of the Doris Precise Orbit files	-	-
ndp	long *		Number of Doris Preliminary Orbit files on input		>= 0 <= 16
doris_prelim_file	char **	all	Filenames of the Doris Preliminary Orbit files	-	-
ner	long *		Number of ESOC Restituted Orbit files on input	-	>= 0 <= 16
esoc_rest_file	char **	all	Filenames of the ESOC Restituted Orbit files	-	-
mjdr0	double *		UTC associated with the start time of the Envisat-1 product to be processed	decimal days (MJD2000)	>= -18262 < +36525
mjdr1	double *		UTC associated with the end time of the Envisat-1 product to be processed	decimal days (MJD2000)	>= -18262 < +36525

The **po_interpol** CFI function has the following input parameters in interpolation mode:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long *	-	Interpolation mode: 2 = the results vector is calculated 102 = the results vector is NOT calculated	-	2, 102
mjdr0	double *	-	Requested UTC time	decimal days (MJD2000)	>= -18262 < +36525
mjdr1	double *	-	DUMMY	-	-

It is also possible to use enumeration values rather than integer values for all input flags:

Input	Description	Enumeration value	Int
mode	Execution mode	PO_INIT_FILE	0
		PO_INTERPOLATE	2
		PO_INTERPOLATE + PO_NO_RESULTS	102
	Optional value to be added to the mode value, in order to avoid producing the optional results vector (and therefore improve performance of the function) See above mode values for usage.	PO_NO_RESULTS	100
choice	Switch to select the source file (only used in initialisation)	PO_AUTO_SELECT	0
		PO_ONLY_DORIS_PRECISE	1
		PO_ONLY_DORIS_PRELIMINARY	2
		PO_ONLY_ESOC_RESTITUTED	3

7.3.4 Output parameters

The output parameters of the **po_interpol** CFI function in initialisation mode are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_interpol	long	-	Main status flag	-	-1, 0, +1
selected	long *	-	Selected file: 1 = Doris Precise Orbit 2 = Doris Preliminary Orbit 3 = ESOC Restituted Orbit	-	1, 2, 3
ierr[10]	long	all	Status vector	-	-

The output parameters of the **po_interpol** CFI function in interpolation mode are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_interpol	long	-	Main status flag	-	-1, 0, +1
selected	long *	-	Selected file: 1 = Doris Precise Orbit 2 = Doris Preliminary Orbit 3 = ESOC Restituted Orbit	-	1, 2, 3
mjdp[2]	double	[0]	UTC of the state vector (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1of the state vector (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
x[6]	double	all	Osculating Kepler elements of the state vector (Mean of Date CS)	-	-
		[0]	Semi-major axis	m	> 0.0
		[1]	Eccentricity	-	>= 0 < 1
		[2]	Inclination	deg	>= 0 <= 180
		[3]	Right ascension of the ascend-ing node	deg	>= 0 < 360
		[4]	Argument of perigee	deg	>= 0 < 360
		[5]	Mean anomaly	deg	>= 0 < 360

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
pos[3]	double	all	Osculating position vector of the state vector (Earth fixed CS)	m	-
vel[3]	double	all	Osculating velocity vector of the state vector (Earth fixed CS)	m/s	-
acc[3]	double	all	Osculating acceleration vector of the state vector (Earth fixed CS)	m/s^2	-
res[54]	double	all	Optional results vector associated with the state vector	-	-
ierr[10]	long	all	Status vector	-	-

It is also possible to use enumeration values rather than integer values for all output flags:

Input	Description	Enumeration value	Int
selected	Selected file	PO_NONE	-1
		PO_DORIS_PRECISE	1
		PO_DORIS_PRELIMINARY	2
		PO_ESOC_RESTITUTED	3

7.3.5 Optional results vector

The results vector is exactly the same as the **po_ppforb** optional results vector (see section 7.2.5), with the exception of:

- element [52]: reserved
- element [53]: reserved

7.3.6 Warnings and errors

Next table lists the possible error messages that can be returned by the **po_interpol** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_ORBIT software library **po_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **po_interpol** CFI function by calling the function of the PPF_ORBIT software library **po_vector_code** (see RD 3).

Error type	Error message	Cause and Impact	Error Code	Error No

7.3.7 Runtime performances

The following runtime performances have been measured (TBC).

mode	Sparcstation 20 [ms]	IBM RS6000 [ms]
0		
2		
102		

7.4 po_genops

7.4.1 Overview

The **po_genops** function is used by the Envisat-1 ground segment near real time processing chains to compute a Cartesian state vector at (or near) true ascending node, using either:

- the DORIS Navigator unconsolidated level-0 products (i.e. produced in near-real-time in the PDHS, according to the acquired downlink instrument data segment)
- the ESOC Predicted ascending node cartesian state vectors.

It is the responsibility of the software calling **po_genops** to prepare the input files, i.e. to ensure that files relevant for the processing being performed are available. See RD 5 for a description of the files.

The output format of the data generated by **po_genops** is compatible with the input of **po_ppforb**. It is either:

- a single Cartesian State Vector at ascending node (if valid DORIS Navigator data on input)
- a single Cartesian State Vector near (plus or minus 5 degrees in latitude) the ascending node, copied from the ESOC Predicted Orbit File (if no valid DORIS Navigator data on input)

po_genops allows to specify a subset of the DORIS file (e.g. in order to avoid a satellite manoeuvre).

The **po_genops** function processes and filters the DORIS navigator level 0 data to verify its usability to create a DORIS generated Cartesian State Vector at true ascending node. Otherwise, in case the unconsolidated level-0 DORIS data are too degraded by noise or in case the difference between the ESOC predicted Cartesian State Vector and the DORIS generated Cartesian State Vector is too large, the **po_genops** function selects the ESOC predicted Cartesian State Vector which is near true ascending node.

The internal noise is estimated by calculating the root-mean-square (RMS) of the difference between the DORIS generated cartesian state vector with the original DORIS Navigator data.

The cartesian state vector output by **po_genops** must be fed to the orbit propagator **po_ppforb** in order to initialize the propagation (using longitude-dependent mode of **po_ppforb**). So, a proper sequence of **po_genops** in conjunction with **po_ppforb** call consists of:

- call to **po_genops**
- call to **po_ppforb** initialization, longitude dependent mode (**po_ppforb** mode PO_INIT)
- successive calls to **po_ppforb** for state vector propagation (**po_ppforb** mode PO_PROPAG)

7.4.2 Calling interface

The calling interface of the **po_genops** CFI function is the following (input parameters are underlined):

```
#include <ppf_orbit.h>
{
      long mode, choice, selected, ierr[10], status;
      double mjd0, mjd1, mjdp[2], pos[3], vel[3], res[32];
      char *doris_nav_file, *esoc_pred_file;

      status = po_genops(&mode, &choice,
                         esoc_pred_file, doris_nav_file,
                         &mjd0, &mjd1,
                         mjdp, pos, vel,
                         &selected, res, ierr);
}
```

For Fortran programs the declaration and calling procedure is as follows (input parameters are underlined, note that the C preprocessor must be used because of the presence of the #include statement):

```
#include <ppf_orbit.inc>

      INTEGER*4 MODE, CHOICE, SELECTED, IERR(10), STATUS
      REAL*8 MJD0, MJD1, MJDP(2), POS(3), VEL(3), RES(32)
      CHARACTER*LENGTH_NAME DORIS_NAV_FILE, ESOC_PRED_FILE

      STATUS = PO_GENOPS(MODE, CHOICE,
     &                   ESOC_PRED_FILE, DORIS_NAV_FILE,
     &                   MJD0, MJD1,
     &                   MJDP, POS, VEL,
     &                   SELECTED, RES, IERR)
```

Note that LENGTH_NAME must be set to the string length of the filename of each type

7.4.3 Input parameters

The **po_genops** CFI function has the following input parameters:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
mode	long *	-	Operation mode 0 = use all records in the Doris Navigator unconsolidated level-0 product 1 = use records in the Doris Navigator unconsolidated level-0 product between mjd0 and mjd1	-	0, 1
choice	long *	-	Switch to select the file 0 = automatic selection 1 = Doris Navigator 2 = ESOC Predicted Orbit	-	0, 1, 2
esoc_pred_file	char *	all	Filename of the ESOC Predicted Orbit file	-	-
doris_nav_file	char *	all	Filename of the Doris Navigator unconsolidated level-0 product file	-	-
mjdr0	double *	-	In mode = 0 UTC associated with the start time of the Envisat-1 product to be processed In mode = 1 UTC of the start time of the sub set of records of the Doris file to be used	decimal days (MJD2000)	>= -18262.0 < +36525.0
mjdr1	double *		In mode = 0 UTC associated with the end time of the Envisat-1 product to be processed In mode = 1 UTC of the end time of the sub set of records of the Doris file to be used	decimal days (MJD2000)	>= -18262.0 < +36525.0

It is also possible to use enumeration values rather than integer values for all input flags:

Input	Description	Enumeration value	Int
mode	Operation mode	PO_ALL_REC	0
		PO_USER_REC	1
choice	Switch to select the source file (only used in initialisation)	PO_AUTO_FILE	0
		PO_ONLY_DORIS_FILE	1
		PO_ONLY_ESOC_FILE	2

7.4.4 Output parameters

The output parameters of the **po_genops** CFI function are:

C name	C type	Array Element	Description (Reference)	Unit (Format)	Allowed Range
po_genops	long	-	Main status flag	-	-1, 0, +1
selected	long *	-	Selected file: 1 = Doris Navigator 2 = ESOC Predicted Orbit	-	1, 2
mjdp[2]	double	[0]	UTC of the state vector (UT1 time)	decimal days (Processing format)	>= -18262 < +36525
		[1]	ΔUT1of the state vector (UT1 time)	s (Processing format)	>= -0.9 <= +0.9
pos[3]	double	all	Osculating position vector of the state vector (Earth fixed CS)	m	-
vel[3]	double	all	Osculating velocity vector of the state vector (Earth fixed CS)	m/s	-
res[32]	double	all	RESERVED	-	-
ierr[10]	long	all	Status vector	-	-

7.4.5 Warnings and errors

Next table lists the possible error messages that can be returned by the **po_genops** CFI function after translating the returned status vector into the equivalent list of error messages by calling the function of the PPF_ORBIT software library **po_vector_msg** (see RD 3).

This table also indicates the type of message returned, i.e. either a warning (WARN) or an error (ERR), the cause of such a message and the impact on the performed calculation, mainly on the results vector.

The table is completed by the error code and value. These error codes can be obtained translating the status vector returned by the **po_genops** CFI function by calling the function of the PPF_ORBIT software library **po_vector_code** (see RD 3).

Error type	Error message	Cause and Impact	Error Code	Error No

7.4.6 Runtime performances

The following runtime performances have been measured:

mode	Sparcstation 20 [ms]	IBM RS6000 [ms]
0		
1		

8 KNOWN PROBLEMS

The following precautions shall be taken into account when using PPF_ORBIT software library:

CFI function	Problem	Work around solution
po_interpol	Error handling not supported yet	Do not call error handling routines
po_genops	Error handling not supported yet	Do not call error handling routines
po_ppforb	po_ppforb init mode is affected by subsequent calls to po_interpol or po_genops	Do not break a sequence of po_ppforb calls with a call to po_genops ot po_interpol
po_genops	early version with some limitations: - do not read Doris Navigator file - do not handle leap second - not fully validated - not supported by error handling	
po_interpol	early version with some limitations: - no output state vector in init mode - do not handle leap second - not fully validated - not supported by error handling	