POLITECNICO DI TORINO

Master's Degree in AEROSPACE ENGINEERING

MASTER's Degree Thesis

Interplanetary Navigator: Development and Validation through Optimal Control Trajectory Optimisation

Advisor

Prof. Lorenzo Casalino

Candidate

Francesco Pio CRISTIANO

DECEMBER 2025

Abstract

This thesis follows my work at the French *Centre National d'Études Spatiales* for the development of a MATLAB Interplanetary Navigator for mission analysis and testing. I was hired as an intern to develop and validate the interplanetary orbital navigator. After the internship, I designed the guidance for an interplanetary trajectory towards an asteroid in order to further test the Navigator.

This work extends the existing CNES MATLAB Navigator from Earth-centric to interplanetary operations by enriching its dynamics and measurement toolkit within an Extended Kalman Filter (EKF) that runs over "orbital windows" and is driven by modular inputs, directives, and telecommands. The dynamics now include IAU-compliant inertial and body-fixed frames (ICRF/BCF), TDB time handling, Chebyshev-based ephemerides, third-body gravity (including solid tides where relevant), first-order relativistic terms, a plate-based solar-radiation-pressure model with eclipse logic, and thrust profiles. On the navigation side, the EKF processes optical line-of-sight (LOS) angles and a LiDAR range model for proximity operations. A mission-generation tool in Java produces scenarios, measurements, and reference ephemerides allowing for testing of the navigator in different missions and environments.

A campaign of validation tests demonstrates performance and limitations. Propagation accuracy is validated, and LOS+LiDAR measurements allow for sub-millimetre state accuracy in planetary orbits. Along non-orbital trajectories, such as the Juventas trajectory (HERA), simplified mean-thrust modelling can induce large transient errors, but the filter still re-converges once measurements become available. In deep-space cruise, LOS-only is insufficient, LiDAR is unavailable and long-distance range/Doppler would be required.

To have a case study in deep space, a guidance analysis allowed for the development of an interplanetary trajectory. The Optimal Control Theory has been studied and adapted on a Fortran trajectory optimizer to compute an optimal low-thrust transfer to the near-Earth asteroid 2000 SG_{344}. The terminal flyby was then evaluated with the navigator, which achieved high accuracy.

Overall, the navigator meets its goals for planetary orbits and interplanetary missions, with a few limitations. To achieve full interplanetary capability, future work should integrate long-range ranging (e.g., radio), refine thrust modelling, and implement attitude laws to fully couple SRP and sensing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Within CNES, the GNC–Trajectory department has begun developing an *Interplanetary Navigator* for satellites. A first prototype, developed by previous interns, was centred on navigation in Earth orbit and did not yet cover the specificities of interplanetary flight. This navigator had been developed in MATLAB® for study and prototyping purposes. Freed from flight constraints (real time navigation, memory limit, operational reliability), this environment serves as a test bed: it promotes readability, modularity and the rapid addition of new models, while offering extensive analysis and storage capabilities.

The objective of the internship is to extend this navigator to the interplanetary context. Concretely, this means integrating new dynamical models, measurement models adapted to interplanetary contexts, and tuning and validating navigation using simulated data and, where possible, real data. The targeted basic building blocks are:

- **Orbital Dynamics model**: gravity of the central body (two–body problem) supplemented by perturbations from third bodies of the Solar System and non–gravitational forces, such as solar radiation pressure and thrust;
- **Time scales and frames**: management of time scales (TT/TDB) and frame transformations (inertial/barycentric vs. planetocentric), necessary for the consistency of calculations, extrapolations, and measurements;
- **Ephemerides**: access to the states (position/velocity) of celestial bodies via a standard ephemeris source, in order to model perturbations or central body changes;
- **Measurement models**: support for directional *Line of Sight* (LOS) measurements from optical sensors, and preparation to integrate other types of measurements (range, Doppler, angles) if required;
- **Telecommands**: special commands to change simulation's parameters, such as the central body or the perturbing body;

- All of the above changes must finally be **validated** through precise testing in various scenarios and missions, both in orbit and in deep space trajectories.

The associated requirements are summarised below.

R-ID	Requirement
R-01	The navigator must be designed to be easily modified, tested and improved (modular architecture, clear interfaces, basic unit tests).
R-02	The navigator must read and process a configuration file, a measurement file and a telecommand file (robust I/O, logging, consistency checking).
R-03	The dynamics model must include at least the gravity of the central body around which the satellite orbits (two–body problem).
R-04	The navigator must be adapted to interplanetary navigation.
R-04.1	The navigator must operate with time scales suitable for the interplanetary context (management of TDB time scale and associated conversions).
R-04.2	The navigator must model the gravitational perturbations of other Solar System bodies.
R-04.3	The navigator must access the position and velocity of celestial bodies via a Chebyshev ephemeris source.
R-04.4	The solar radiation pressure (SRP) model must be improved to account for attitude laws.
R-04.5	The navigator must use interplanetary measurements for navigation (measurement model, uncertainty management, weighting).
R-04.6	The navigator must handle changes of central body and adaptation of dynamic models over the course of an interplanetary mission (trajectory segments, frames, events).
R-04.7	The navigator must model thrust acceleration (impulsive or continuous) and its injection into the dynamics.
R-04.8	All new developments must be validated through precise simulations and testing, in order to tune the Kalman Filter.

Priority is given to integrating the effects and measurements essential to robust interplanetary navigation, while keeping an open architecture for future extensions (new measurements, new physical models, new measures).

1.2 CNES and the GNC-Trajectory Service

The **Centre National d'Études Spatiales** (CNES) is the French space agency in charge of preparing, proposing and implementing France's space policy, in close coordination with European and international partners. It is involved in a wide range of missions, from Earth observation and telecommunications to navigation, science and exploration, as well as the associated ground segments and data systems [1].
Beyond the development of satellites and launchers, CNES provides technical expertise and programmatic support throughout the life cycle of space systems: early-phase studies, definition of mission requirements, support to development and qualification, and assistance during in-orbit operations. It also acts as a key interface between public authorities, industry and the research community, helping to structure and support the French and European space ecosystem.

1.2.1 GNC-Trajectory Department

The work presented in this thesis was carried out within the **GNC-Trajectory** service of CNES, which belongs to the Technical and Digital Directorate (DTN). The service is part of the teams responsible for *Guidance, Navigation and Control* (GNC) and flight dynamics. It brings together a small group of engineers specialising in orbital mechanics and space navigation.
The main activities of the service cover precise orbit determination (POD), orbit propagation and modelling, preparation and analysis of operational orbit determination chains, and the development of navigation algorithms for both Earth-orbiting and interplanetary missions. Depending on the project, the team may be involved from the feasibility-study stage up to in-flight validation and routine operations, working in close collaboration with other CNES departments and external partners. The internship on which this thesis is based took place in this service, in the context of its expertise in flight dynamics and operational orbitography.

1.3 Architecture of the MATLAB Navigator

Since the MATLAB navigator is not an embedded navigator, its basic architecture is built to be easier to analyse and modify. This MATLAB navigator plays both the role of a model library and a navigator: the dynamic models are introduced as functions or classes and they are used by the *main* to navigate with the Kalman Filter.

1.3.1 Global architecture

The architecture of the MATLAB Navigator is fairly linear:

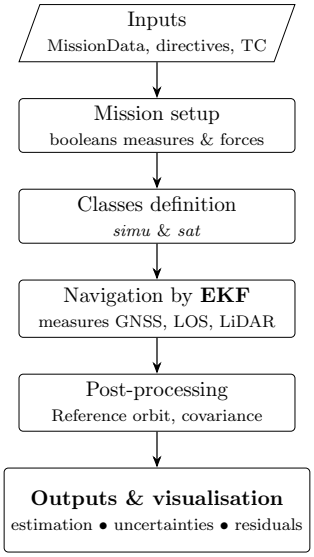


Figure 1.1: Architecture of the MATLAB Navigator.

At the centre of the Navigator is the Kalman Filter. The filter parameters are chosen from the input files.

1.3.2 Inputs

The initial values of the simulation are given by the input files:

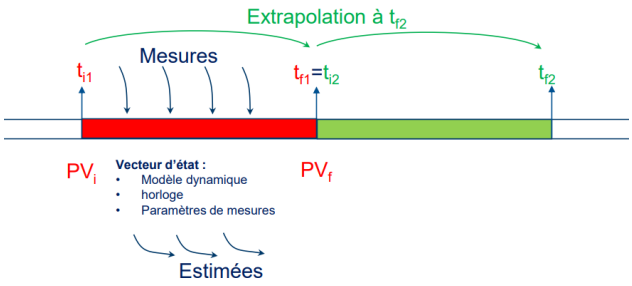
- **MissionData**: a .json file that defines all the aspects necessary to perform a simulation. Physical parameters, bodies present in the simulation, active forces, Kalman filter parameters, satellite parameters, position of the antenna, cameras, etc.
- **directives**: a text file that completely describes the mission: it provides the time, the initial orbit, the instants at which to extrapolate and all of the measures.
- **tc**: the Navigator is capable of modifying parameters via telecommands (tc). The file is read at the beginning of the simulation and the commands are stored in *simu*. Afterwards, they are executed at the indicated instant.

- Other data: the Navigator requires several other data sets to perform the simulation: planetary ephemerides (for interplanetary missions), harmonic coefficients of the potential, orientation of the body-fixed frame, etc.

These input files are read by the navigator, which uses them to build the classes and the base data of the mission. Most of the simulation data are stored in the variable *simu*: perturbing bodies, active forces, harmonic coefficients of the potential, etc.
By contrast, other data are stored in *sat*, the variable responsible for storing information on the satellites: clock time, state, measurements, etc.

1.3.3 Navigation via EKF

Navigation is carried out by the extended Kalman filter; the method consists of dividing the mission into orbital windows, indicated by the *EXTRAPOLATION* directive.



Each orbital window is processed individually by the navigator with the *Extended Kalman Filter*. During each orbital window, the navigator performs an extrapolation of the next state and uses the measurements of the window to estimate the next state. Finally, it uses the extrapolated values and the estimated values to correct the state.

Chapter 2

Physical Models of the Navigator

2.1 Space reference frames

Space reference frames form the basis of all navigation activity. Choosing the right frame is therefore essential: some operations can only be performed in an inertial frame, others only in a body-fixed frame. Between these two extremes there are also several intermediate frames that can be used depending on the type of application or the desired precision.
In the MATLAB Navigator, these two categories of frames are clearly distinguished: inertial frames for dynamics and state propagation versus body frames for local calculations.

2.1.1 Existing reference frames for navigation around the Earth

All of the reference frames already available in the MATLAB Navigator are suited to navigation applications around the Earth. The existing frames have been sufficiently tested on Earth orbits and are optimised for the measurements available in the Navigator.

- **Inertial frames**
  - **VEIS**: takes into account the motion of the Earth's pole but not the effects of precession/nutation. This frame is no longer used for extrapolation; however, the computation of the position of the Moon and Sun according to the Meeus model (sec. 2.3) provides positions in the VEIS frame.
  - **CIRF** (Celestial Intermediate Reference Frame): intermediate celestial frame used as a quasi-inertial frame in place of the VEIS. It takes into account the motion of the Earth's pole; the effects of precession and nutation are handled elsewhere.
  - **ICRF** (International Celestial Reference Frame): inertial frame defined by the IERS[1] standards. This frame is appropriate for applications outside Earth orbit.
- **Body-fixed frames**

[1] IERS: International Earth Rotation and Reference Systems Service

– **ITRF** (International Terrestrial Reference Frame): non-inertial frame fixed to the Earth, defined for applications around the Earth. It is notably used for the calculation of the terrestrial potential and the processing of GNSS and ground station measurements.

The algorithm of the MATLAB Navigator is based on that of SOFA[1][2] and complies with the IERS standards.
All of the frames available in the Navigator can be reached by vector rotations. In the MATLAB Navigator, transformations are performed by the function *CL_fr_convert* (from CelestLab, CNES), which automatically allows the user to pass from any frame to any other.
The expression of the transformation matrix from the ITRF to the GCRF is as follows:

$$M_{ITRF \to GCRF} = M_{CIRF \to GCRF} \cdot M_{TIRF \to CIRF} \cdot M_{ITRF \to TIRF} \tag{2.1}$$

With:

- $M_{ITRF \to TIRF}$: the pole motion matrix;
- $M_{TIRF \to CIRF}$: the diurnal rotation matrix;
- $M_{CIRF \to GCRF}$: the precession–nutation matrix.

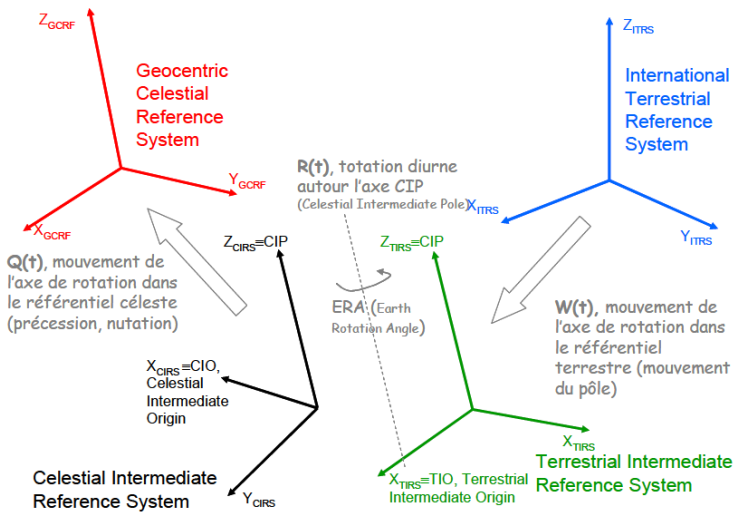


Figure 2.1: Reference systems of the IERS conventions: ITRS, TIRS, CIRS, GCRS

All the terrestrial reference frames used by the Navigator are explained in more detail in Appendix A

[2]SOFA: Standards of Fundamental Astronomy

2.1.2 Interplanetary evolution: BCF frame

The current reference frames are not sufficient for interplanetary applications, with the exception of the inertial frame ICRF. Although state extrapolation is performed with the existing frame ICRF[3], it is necessary to define new frames tied to the celestial bodies for the calculation of the planets' potentials.

The **BCF** (Body-Centered Fixed) frames are rotating frames fixed to specific celestial bodies that rotate with the celestial body itself. They are used for all bodies except the Earth (which uses the ITRF) for computing the potential of the central body.
Their orientation follows the IAU[4] conventions via the pole rotation (right ascension, declination) and a meridian angle that incorporates precession, nutation and even librations (for many moons), as well as variations in the length of the day. Concretely, the inertial → BCF transformation is expressed as a chain of time-dependent rotations (typically), which guarantee that the gravity field (spherical harmonics), the topography, and ground references remain fixed in this frame.

To define the orientation of the axes of these frames, one uses the IAU conventions defining the orientation parameters of the planet:

- α: right ascension of the planetary pole;
- δ: declination of the planetary pole;
- W: prime meridian.

Transformation matrix ICRF → BCF

The transformation matrix between the inertial frame (ICRF) and the body-fixed frame is computed from the parameters α, δ and W. These orientation parameters depend on coefficients specific to each body and on time, expressed on the TDB (Barycentric Dynamical Time) scale. They are computed from the following relations:

$$\alpha(T_{JD}) = a_\alpha + b_\alpha T_{cent} + \sum_{i=1}^{n} K_\alpha^i \, \sin\big(c_\alpha^i + d_\alpha^i T_{JD}\big) \tag{2.2}$$

$$\delta(T_{JD}) = a_\delta + b_\delta T_{cent} + \sum_{i=1}^{n} K_\delta^i \, \cos\big(c_\delta^i + d_\delta^i T_{JD}\big) \tag{2.3}$$

$$W(T_{JD}) = a_W + b_W T_{JD} + \sum_{i=1}^{n} K_W^i \, \sin\big(c_W^i + d_W^i T_{JD}\big) \tag{2.4}$$

The parameters depend on:

[3] In the code the ICRF frame is called "GCRF" because it was designed for terrestrial use; however, its use remains valid anywhere in the Solar System.
[4] IAU: International Astronomical Union

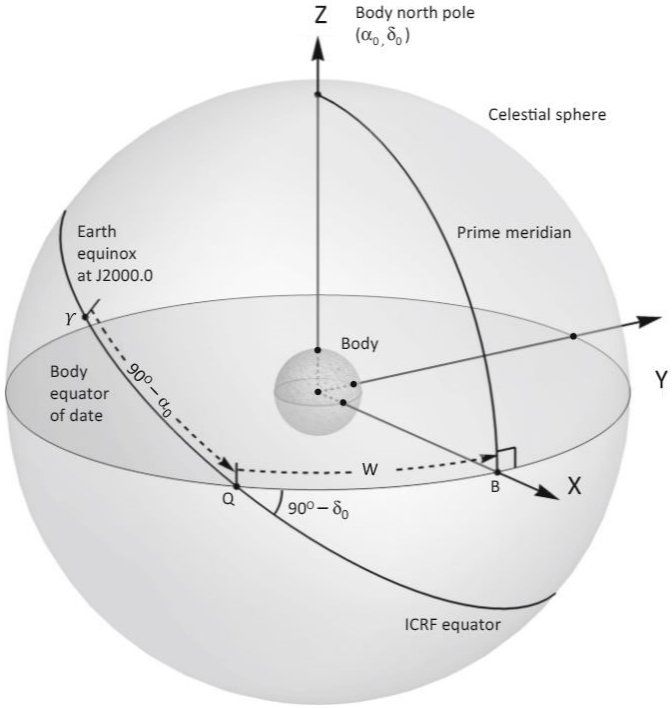


Figure 2.2: Parameters linked to the fixed frame of a body (α, δ, W)

- the time, expressed in Julian days (T_{JD}) and in centuries ($T_{cent} = T_{JD}/36525$)since J2000 ;
- coefficients specific to each body a, b, c, d, K.

All models are based on the IAU 2015, except for the Moon, which uses the IAU 2009 model.
The transformation between the ICRF and the frame tied to the body (BCF) is given by the following relation (three rotations):

$$M_{ICRF \to BCF} = R_3\Big(\tfrac{\pi}{2} + \alpha\Big)\, R_1\Big(\tfrac{\pi}{2} - \delta\Big)\, R_3(W) \tag{2.5}$$

The transformation matrix between the BCF frame and the ICRF is simply the transpose:

$$M_{BCF \to ICRF} = \big(M_{ICRF \to BCF}\big)^T \tag{2.6}$$

Rotation velocity ICRF → BCF

The rotation velocity between the two frames is calculated by taking the time derivative of the previous relations.

$$\dot{\alpha}(T_{JD}) = \frac{b_\alpha}{36525} + \sum_{i=1}^{n} K_\alpha^i d_\alpha^i \cos\big(c_\alpha^i + d_\alpha^i T_{JD}\big) \tag{2.7}$$

$$\dot{\delta}(T_{JD}) = \frac{b_\delta}{36525} - \sum_{i=1}^{n} K_\delta^i d_\delta^i \sin\left(c_\delta^i + d_\delta^i T_{JD}\right) \tag{2.8}$$

$$\dot{W}(T_{JD}) = b_W + \sum_{i=1}^{n} K_W^i d_W^i \cos\left(c_W^i + d_W^i T_{JD}\right) \tag{2.9}$$

After computing the derivatives, the function (inspired by CelestLab) *CL_rot_angVelocity* uses the angles and their derivatives to compute the rotation velocity between frames $\omega_{ICRF \to BCF}$.
The MATLAB function *fr_icrf2bcf* enables the coefficients of any celestial body (found in the data files or in the simulation variable *simu*) to be obtained and the parameters to be calculated in order to perform the transformation.
As for all frames, the transformation is entirely managed by *CL_fr_convert*, which requires the time in CJD, the initial frame and the final frame. The formula *"BCF_astre"* can be used to perform the transformation to the BCF frame of the body named *astre*.

2.1.3 Local frames

The local frames (tied to the satellite) defined in the MATLAB Navigator are: the inertial frame (IN), the platform frame (PF), the QSW frame and the TNW frame.
The transitions between these different frames are described in the following paragraphs.

QSW frame

The QSW frame is defined as follows:

- Q: radial axis
- W: normal to the orbit
- S: completes the triad

The components are computed from the position/velocity of the satellite in the inertial frame:

$$\vec{Q} = \frac{\vec{P}}{||\vec{P}||} \tag{2.10}$$

$$\vec{W} = \frac{\vec{Q} \wedge \vec{V}}{||\vec{Q} \wedge \vec{V}||} \tag{2.11}$$

$$\vec{S} = \vec{W} \wedge \vec{Q} \tag{2.12}$$

TNW frame

The TNW frame is defined as follows:

- T: tangential axis
- W: normal to the orbit
- N: completes the triad

The components are computed from the position/velocity of the satellite in the inertial frame:

$$\vec{T} = \frac{\vec{V}}{||\vec{V}||} \tag{2.13}$$

$$\vec{W} = \frac{\vec{P} \wedge \vec{V}}{||\vec{P} \wedge \vec{V}||} \tag{2.14}$$

$$\vec{N} = \vec{W} \wedge \vec{T} \tag{2.15}$$

2.2 Time reference frames

Date formats

The types of date already defined in the Navigator before the interplanetary evolution are:

- Date **JD**: The Julian Days (JD) are a continuous count of days used in astronomy. JD 0 is set at 1 January 4712 (4713 BC, Julian calendar) at noon.
- Date **CJD**: corresponds to the number of Julian days since 01/01/1950 at midnight (CJD = JD – 2433282.5).
- Date **counter**: value corresponding to an onboard clock. The correspondence between the counter date and a CJD date is defined in the navigator.

  *Format: number of seconds since the last update of the counter (**seconds**, integer) + number of fractions of a second within the second (**fractions of a second**, real)*
- Date **GPS**: date in GNSS[5] format. Specific functions allow conversion between a CJD date and a GPS date.

  *Format: **Days** (integer) **Seconds** (real) **Fractions of a second** (real)*

Terrestrial scales

The types of time scales already defined in the Navigator before the interplanetary evolution are:

- **TAI**: International Atomic Time. It is a time scale provided by the BIPM (Bureau International des poids et mesures) and based on nearly 150 atomic clocks worldwide. It makes it possible to define the time standard and the reference time scale used throughout the world.
- **TT**: Terrestrial Time. It is a time scale established from TAI and defined by the International Astronomical Union for making time measurements on astronomical observations performed from the Earth. This time scale was previously called TDT (Terrestrial Dynamic Time) or ET (Ephemeris Time). TT is defined by the following relation:

$$TT = TAI + 32.184s \tag{2.16}$$

- **UTC**: Coordinated Universal Time. It is an international standard on which civil time is based. UTC differs from TAI by an integer number of seconds. The number of seconds separating these two time scales is updated when UTC drifts more than 0.9s from UT1. As of today this number is 37s, so we have the following relation:

$$UTC = TAI - 37s \tag{2.17}$$

[5]GNSS: Global Navigation Satellite System

- **UT1**: UT1 drifts variably relative to TAI. When the desired precision is not greater than one second one may consider that UTC=UT1. In the MATLAB Navigator the offset TAI–UT1 is set to zero, but this choice is configurable via the simulation data (in *simu.Temps.UT1_UTC*).

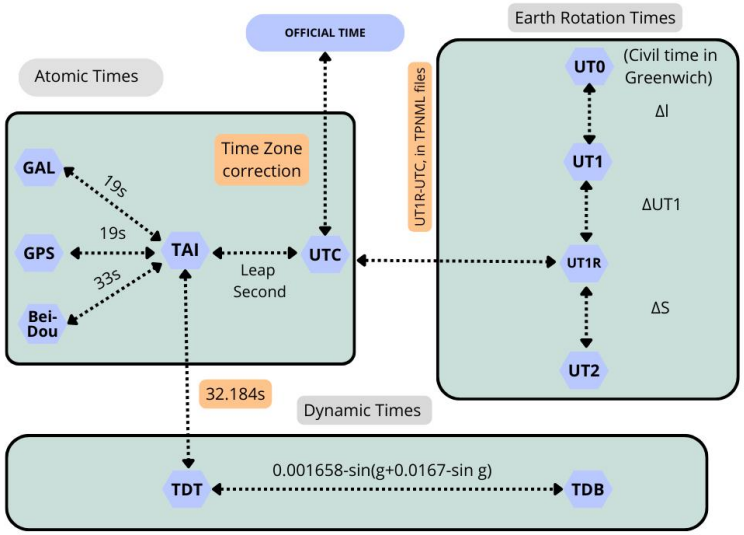


Figure 2.3: Conversions between the time scales

As with the spatial reference frames, the time scales present in the Navigator are sufficient for Earth orbits, but poorly suited for interplanetary missions: they do not take into account relativistic effects (related to the distances and gravitational potentials between bodies).

Barycentric Dynamical Time (TDB)

The *TDB* is a relativistic coordinated time scale, intended for astronomy to take into account time dilation in the calculation of orbits and ephemerides (planets, asteroids, comets, satellites). In the MATLAB Navigator, it is used for computing the positions of bodies and the rotation axes of bodies. To convert between scales, we use the difference $TDB{-}TT$ as implemented in SOFA:

$$TDB - TT = 0.001657 \sin g, \tag{2.18}$$

$$g = 6.24 + 0.017202 \cdot JD2000_{TT}, \tag{2.19}$$

where $JD2000_{TT} = JD_{TT} - 2451545$ is the Julian date in TT counted since 01/01/2000. This approximation typically provides a precision of the order of 50 μs on $TDB{-}TT$ (more complete models exist in SOFA).
In practice, the error induced by replacing the TDB by the TT remains negligible in many cases;

2.3 Calculation of the positions of celestial bodies

The MATLAB navigator was initially developed for Earth satellites, where the only celestial bodies involved in navigation are the Moon and the Sun. Their effects are calculated by mathematical models: those of Meeus. This choice simplifies calculations and storage, but it is not sufficient for interplanetary missions.
The Meeus models provide the position of the Moon and the Sun in an Earth-centred frame. They require a date in Terrestrial Time (TT) and provide the position of the Moon in the EoD frame and the position of the Sun in the VEIS frame.
Moreover, the analytical models do not provide velocity information. Consequently, certain operations that require these data cannot be carried out using only the analytical ephemerides, such as changes of central body. If necessary, it would nevertheless be possible to compute this velocity by finite difference.

In order to compute the effects of other bodies on the dynamics of the satellite in interplanetary missions, it is necessary to know their position.
For interplanetary missions, one must know uniformly and accurately the positions and velocities of many bodies (planets, moons, sometimes asteroids) in a barycentric inertial frame (ICRF) and a relativistic time scale (TDB). These data are necessary to model the gravitational perturbations of third bodies or to handle transitions of sphere of influence and changes of central body.
Hence the need to integrate the reading of numerical ephemerides that complement or replace the Meeus models as soon as the navigation leaves the terrestrial neighbourhood.

2.3.1 Meeus analytical model

There is one Meeus model for the Moon and one for the Sun. These models provide the position of the Moon and the Sun in an Earth-centred frame. The implemented models are taken from Patrius. The lunar model has the particularity that it allows the number of terms in the series to be selected when computing the Moon's position, which makes it possible to reduce computation time at the expense of accuracy.
The models require a date in Terrestrial Time (TT) and provide the position of the Moon in the EoD frame and the position of the Sun in the EME2000 frame. The functions for converting these frames to the GCRF frame have been implemented in BOLERO[6].
The analytical models do not provide velocity information. Consequently, certain operations requiring these data cannot be performed using the analytical ephemerides, such as changes of central body. If necessary, however, it would be possible to compute this velocity by finite difference.

[6]BOLERO is CNES's reference library for orbital navigation. It is originally developed in C and it contains all physical models and navigation methods for a Navigator.

2.3.2 Calculation by numerical ephemerides

In order to enable the navigator to simulate interplanetary missions, it is always necessary to know the position and velocity of any celestial body. The ephemerides used by the MATLAB navigator are provided in the form of Chebyshev coefficients, which then make it possible to calculate the position of a body via Chebyshev polynomials. The reading of the ephemerides is carried out by the function *LectureEph.m*, which automatically uses the text files found locally. The reading is usually performed only once at initialisation: thereafter, the resulting data are stored in the variable *simu.simuAstres*. This allows the position and velocity at any time to be computed very efficiently.

Position of a body

Most of the ephemerides used give the position of the barycentre of the body relative to the Solar System Barycentre (SSB), except for planet–moon systems (Earth–Moon, Mars–Phobos, ...) which are ephemerides of position/velocity relative to the barycentre of the two-body system or the more massive body.

The numerical ephemerides are provided as series of Chebyshev coefficients $a_0, \ldots, a_n$ (order n), in kilometres in the ICRF frame. For each coordinate x, y, z, a set of coefficients is valid over a time interval $[t_1, t_2]$. The position at time $t \in [t_1, t_2]$ is obtained by evaluating, for each coordinate, the corresponding Chebyshev series (after reducing the time to $\bar{t} \in [-1, 1]$). These ephemerides are described as *piecewise*: the timeline is divided into segments of N days, each equipped with its own coefficients. This scheme offers a good compromise between precision and storage: the desired accuracy is achieved with a moderate order n; if necessary, one adjusts n or the length of the segment to control the error.
Example for the coordinate x:

$$x = \sum_{i=0}^{n} a_i T_i(\bar{t}) \tag{2.20}$$

With the recurrence relation,

$$\begin{cases} T_0 = 1 \\ T_1 = \bar{t} \\ T_i(\bar{t}) = 2T_{i-1}(\bar{t}) - T_{i-2}(\bar{t}) \end{cases} \tag{2.21}$$

And,

$$\bar{t} = \frac{2(t - t_1)}{t_2 - t_1} - 1 \tag{2.22}$$

By performing this process on the three coordinates (x, y, z), one obtains the position of the body in the ICRF frame. If available, one uses the data stored in *simu*.

Velocity of a body

The velocity of a body is computed from the same input data as the position, using the derivative of the Chebyshev polynomial.
Example for the coordinate V_x:

$$V_x = \frac{dx}{dt} = \sum_{i=0}^{n} a_i \frac{dT_i(\bar{t})}{dt} \tag{2.23}$$

With the recurrence relation,

$$\begin{cases} \frac{dT_0}{dt} = 0 \\ \frac{dT_1}{dt} = \frac{d\bar{t}}{dt} \\ \frac{dT_i(\bar{t})}{dt} = 2\bar{t}\frac{dT_{i-1}(\bar{t})}{dt} + 2\frac{d\bar{t}}{dt}T_{i-1}(\bar{t}) - \frac{dT_{i-1}(\bar{t})}{dt} \end{cases} \tag{2.24}$$

Where,

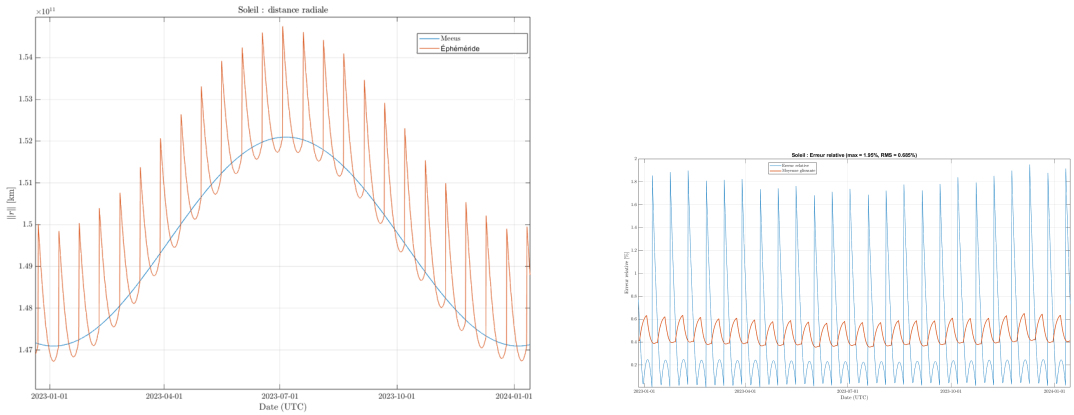
$$\frac{d\bar{t}}{dt} = \frac{2}{t_2 - t_1} \tag{2.25}$$

The ephemerides give the velocities in ICRF, centred on the SSB, in km/day. Afterwards, the calculation function automatically converts them into m/s.

The positions and velocities are computed by the function *CalculPositionVitesseEph.m*, which provides the ephemerides results in m and m/s.

Comparison between ephemerides and Meeus model

Computing the position of the Sun and Moon with the Meeus model is very advantageous in terms of computation cost; however, the results present errors. Computing the position with the ephemerides, even with simplifications, will always be more precise than the analytical models. Here we therefore compare positions computed by the Meeus model with those from the ephemerides.
With regard to the Sun, below we show the radial distance between 1 January 2023 and 1 January 2024.



(a) Sun: radial distance

(b) Sun: relative error

Figure 2.4: Comparison Meeus vs ephemerides for the Sun (2023–2024).

The results are clear: even if the mean error remains moderate ($\sim$2 %), the difference is very variable and can reach up to 2.5 billion metres.
It is always preferable to work with the position computation of bodies performed by the ephemerides. The MATLAB navigator has no constraints in computation time, so it is the preferred option.

2.3.3 Calculation with *CalcBodyPos.m*

Computing the position of one celestial body relative to another is a delicate process involving several steps that must be performed correctly:

- Conversion to TDB
- reading the ephemerides of both bodies
- computing the positions/velocities of the bodies relative to the SSB
- computing the relative positions/velocities between the two bodies
- conversion to the correct frame

In the MATLAB navigator this is handled entirely by the function *CalcBodyPos.m*, which performs all these steps automatically. This function uses the default ephemerides (with *CalculPositionVitesseEph.m*) but it can also use the mathematical models of the Moon and the Sun.
The method *simu.interpolationAstres* computes the positions and velocities of all bodies considered during the simulation using *CalcBodyPos.m*, interpolates them by *spline* and finally stores the results in *simu.simuAstres*. After this precomputation step, *CalcBodyPos.m* uses these interpolated positions and velocities for subsequent calculations, thereby optimising the simulation. Several tests were carried out to validate that using the spline instead of the direct position computation does not produce errors.

This function can also be used to compute the distance of the satellite with respect to any celestial body.

2.4 Accelerations

The navigation dynamics around the Earth were modelled by the following accelerations:

$$\vec{\gamma}_{total} = \vec{\gamma}_{Central\,Body} + \sum \vec{\gamma}_{Perturbing\,Bodies} + C_s\vec{\gamma}_{SRP} + C_f\vec{\gamma}_{Drag} \tag{2.26}$$

with:

- $\vec{\gamma}_{Central\,Body}$ the gravitational acceleration of the central body (More thoroughly explained in Appendix B.1).
- $\vec{\gamma}_{Perturbing\,Bodies}$ the gravitational acceleration due to the perturbing bodies: Moon and Sun.
- $\vec{\gamma}_{SRP}$ the acceleration due to solar radiation pressure, assuming a spherical satellite.
- $\vec{\gamma}_{Drag}$ the acceleration due to atmospheric drag (More thoroughly explained in Appendix B.2).

This dynamic model is optimal for terrestrial missions, but it must be modified and extended for interplanetary missions, taking into account the following effects:

- $\vec{\gamma}_{Central\,Body}$ must be computable for all celestial bodies in the solar system.
- $\vec{\gamma}_{Perturbing\,Bodies}$ must be adapted to take into account planets other than the Moon and the Sun (using ephemerides in place of mathematical models).
- More precise calculation of $\vec{\gamma}_{SRP}$: satellite modelled by plates and taking attitude laws into account.
- It is necessary to add the thrust acceleration model $\vec{\gamma}_{Thrust}$.

2.4.1 Gravity of the Central Body

The acceleration due to the central body, already implemented in the first version of the MATLAB Navigator, followed the model below:
We denote x, y and z the coordinates of the satellite in an inertial frame centred on the central body. We denote r, φ, λ the spherical coordinates of the satellite, r is the distance, φ the latitude and λ the longitude:

$$\begin{aligned} x &= r\cos\lambda\cos\varphi \\ y &= r\sin\lambda\cos\varphi \\ z &= r\sin\varphi \end{aligned} \tag{2.27}$$

With these notations we have:

$$\vec{\gamma}_{Central\,Body} = \vec{\nabla} U \tag{2.28}$$

With,

$$U = \frac{GM}{r} \sum_{l=0}^{n} \sum_{m=0}^{n} \left(\frac{R_{eq}}{r} \right)^n \bar{P}_{lm}(\sin\,\varphi)(\bar{C}_{lm} \cos m\lambda + \bar{S}_{lm} \sin m\lambda) \tag{2.29}$$

where $\bar{C}_{lm}$ and $\bar{S}_{lm}$ are the normalised harmonic coefficients of the potential of degree l and order m, given in the fixed frame of the central body. A more detailed explanation of this dynamical model is available in Appendix B.1

Interplanetary Evolution

The calculation of the potential has been adapted to different choices of central body: the default inertial frame is ITRF if the central body is the Earth, or BCF if the central body is another; the conversion to BCF is managed by *CL_fr_convert*, as explained in section 2.1.

If the harmonic coefficients of the potential C and S are available in the file *astre_pot_dione*, the method *simu.LectureHarmPot* stores them and the potential calculation function *CalcAccPot.m* uses them up to the order of precision n. Otherwise, the code is also adapted to perform a simple potential calculation by

$$\vec{\gamma}_{Central\,Body} = \frac{GM}{r^2} \vec{r} \tag{2.30}$$

2.4.2 Gravity of Perturbing Bodies

A perturbing body is any star (other than the central body) having a gravitational influence on the satellite. The study of the satellite's motion is carried out in a frame tied to the central body; it is in fact the differential attraction exerted by the perturbing body on the satellite and on the central body that must be taken into account.
The calculation of the effect of the perturbing bodies was already implemented in the previous version of the Navigator; however, it has been adapted to other stars besides the Moon and the Sun.
The acceleration of the perturbing bodies is broken down into two accelerations, a direct acceleration and a tidal acceleration $\vec{\gamma}_{Pert\,Bodies} = \vec{\gamma}_{Direct} + \vec{\gamma}_{tide}$

Direct Acceleration

Let O be the position of the centre of gravity of the central body, A the position of the centre of gravity of the perturbing body of mass M_p and S the position of the centre of gravity of the satellite.
If we assume that the central body and the perturbing body are both point masses, then the direct acceleration is:

$$\vec{\gamma}_{direct} = GM_p \left(\frac{\vec{SA}}{\|\vec{SA}\|^3} - \frac{\vec{OA}}{\|\vec{OA}\|^3} \right) \tag{2.31}$$

The distances between the central body and the perturbing body are computed with the help of the function *CalcBodyPos.m*, described in paragraph 2.3.

Solid Tidal Acceleration

Celestial bodies are subject to a deformation caused by perturbing bodies. This deformation depends on several factors: the body itself, the characteristics of the perturbing bodies and their distances. These deformations can have significant effects on the satellite dynamics, called solid tide accelerations.
The solid tide acceleration on the Navigator is based on the Love model developed at order 2.

$$\vec{\gamma}_{tide} = \frac{3}{2} k_2 \frac{GM_p}{r_b^3} \frac{R^5}{r^4} \left[(1 - 5\cos^2(\Psi) \frac{\vec{r}}{r} + (2\cos(\Psi) \frac{\vec{r_b}}{r_b} \right] \tag{2.32}$$

with:
G : Gravitational constant.
M_p : Mass of the perturbing body.
k_2 : Love number of the central body.
$\vec{r_b}$: Position of the perturbing body with respect to the central body.
$\vec{r}$: Position of the satellite with respect to the central body.
R : Equatorial radius of the central body.
Ψ : Angle between the perturbing body and the satellite from the central body

where the most important factor is the Love number k_2 of the central body, which indicates its rate of deformation. The MATLAB Navigator can also use the order 3 development of the Love model.

$$\vec{\gamma}_{\text{tide},3} = \frac{1}{2} k_3 \mu_b \frac{R^7}{r^5 r_b^4} \left[\left(15 - 35\cos^2\Psi\right) \cos\Psi \frac{\vec{r}}{r} + \left(15\cos^2\Psi - 3\right) \frac{\vec{r_b}}{r_b} \right] \tag{2.33}$$

In the MATLAB Navigator, the Love number k_2 is considered non-zero only for the Earth and the Moon. For the time being, all other bodies are considered undeformable.

2.4.3 Solar Radiation Pressure

The model for calculating Solar Radiation Pressure (SRP) present in the MATLAB Navigator considered the satellite to be spherical; since solar radiation pressure is one of the main forces acting during interplanetary missions, the introduction of a more precise model was necessary.
When the satellite is in shadow, the solar radiation pressure is equal to 0.

Modelling the solar radiation pressure on a plate

The new satellite model is described by 3 plates indicated by their normal vector plus a plate that describes the solar panels.

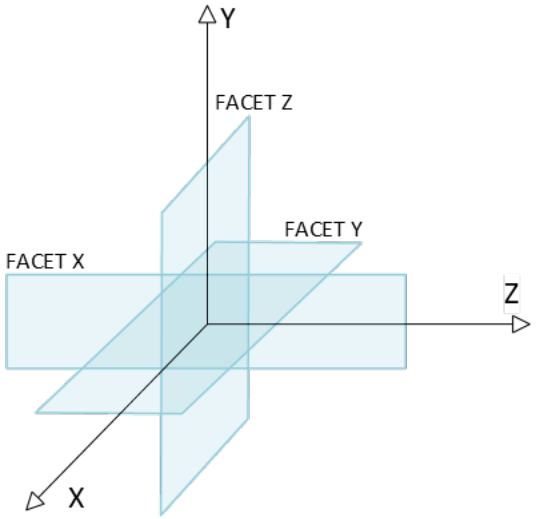


Figure 2.5: Satellite plate model

Let:

- k_s the coefficient of specular reflection of the plate.
- k_d the coefficient of diffuse reflection of the plate.
- k_a the absorption coefficient of the plate.

S_p the surface area of the plate and m_{sat} the mass of the satellite. If $\vec{N_p}$ is the normal vector to the plate then:

$$\vec{\gamma}_{SRP\,Plate} = P_s \frac{S_p}{m_{sat}} \cdot (\vec{u_s} \cdot \vec{N_p}) \cdot (\vec{F_s} + \vec{F_D} + \vec{F_A}) \tag{2.34}$$

$$\vec{\gamma}_{SRP} = \sum_{Plates} \vec{\gamma}_{SRPplate} \tag{2.35}$$

where:

$\vec{F_s} = \left(2k_s \cdot (\vec{u_s} \cdot \vec{N_p})\right) \cdot \vec{N_p}$
$\vec{F_D} = k_d \cdot \left(\vec{u_s} - \frac{2}{3}\vec{N_p}\right)$
$\vec{F_A} = k_A \cdot \vec{u_s}$
With P_s the constant of solar radiation pressure ($4.56 \times 10^{-6} N/m^2$).

The calculation of the Solar Radiation Pressure uses a more precise model of the satellite in order to take into account the orientation of the plates relative to the Sun: the orientation depends by the dynamic attitude law. This attitude law (better described in sec. 2.5) is not yet taken into account by the model. In fact, at present the satellite is considered "inertial" with all the plates oriented towards their normal axis and the solar panels oriented towards the Sun.
The normal vectors to the plates do not take attitude into account, because the effect of the satellite orientation is described by the normal vector, except for the solar panels where the orientation is described by the attitude. However, the current satellite model considers the panels always turned towards the Sun.
The current navigator manages the attitude only for Line of Sight measurements and it does not yet take into account the attitude law (sec. 2.5) commanded by telecommand. This attitude law must influence the orientation of the plates and produce a different solar radiation pressure. However, without an attitude law, the navigator considers the plates always turned towards their normal vector and the solar panels always turned towards the Sun.
The value P_s is modified to take into account the Satellite/Sun distance using the formula

$$P'_s = \frac{AU}{d^2} \tag{2.36}$$

with:
AU the astronomical unit (Distance Sun - Earth) and d the distance Sun/Satellite. This function allows the SRP acceleration to be described at any location in the Solar System.

Eclipse model
The algorithm presented here is taken from reference [2].
The hypotheses made are as follows:

- The Sun and the central body are spheres
- Only the occultation of the Sun by the central body is taken into account; occultations due to Moons are not taken into account.

Let $\vec{r}$ the vector linking the central body to the satellite, $\vec{s}$ the vector linking the satellite to the central body and $r_\odot$ the vector linking the central body to the Sun.
The radius of the central body is denoted R_B and the radius of the Sun is denoted $R_\odot$

$$a = \arcsin \frac{R_\odot}{\|\vec{r_\odot} - \vec{r}\|}; \qquad b = \arcsin \frac{R_B}{\|\vec{s}\|}; \qquad c = \arccos \frac{\vec{s}^T(\vec{r_0} - \vec{r})}{\|\vec{s}\|\|\vec{r_\odot} - \vec{r}\|} \tag{2.37}$$

with a the apparent angle of the Sun, b the apparent angle of the central body and c the separation angle of the 2 bodies.
If $a + b \leq c$ there is no occultation, the shadow factor $\nu = 1$.
If $c < b - a$ the occultation is complete, so the shadow factor is $\nu = 0$.

If $c < a - b$ the central body is in front of the Sun but the Sun still appears. The coverage factor is $\nu = 1 - \frac{S}{\pi R_0}^2$
This modelling is not considered for the heliocentric phases of the missions, where the Sun is the central body, however the MATLAB Navigator is capable of considering the eclipse of any celestial body.

2.4.4 Thrust

The introduction of a thrust acceleration is carried out by Telecommand (see section 3.3). The Navigator applies a special treatment to thrust accelerations. Whenever a thrust manoeuvre occurs, its effect is meaned over the entire Orbital Window, and the state is propagated considering the thrust constant in the window. This choice is justified by the behaviour of the Kalman Filter: modest simplifications in the thrust model do not cause trajectory divergence because the filter continuously corrects the satellite state; small acceleration errors are therefore acceptable.
In order for the correction to work, however, the thrust error has to be taken into account by the Kalman Filter through the State Vector, the Covariance, the Transition Matrix and the Covariance matrix (more thoroughly explained in 3.2).

Instantaneous thrust

The calculation of the instantaneous thrust acceleration $\vec{\gamma}_{Thrust}(t)$ is done using the following equation:

$$\vec{\gamma}_{Thrust}(t) = \frac{\vec{F}}{m_i - \frac{\|\vec{F}\|}{g_0 I_{sp}}(t - t_i)} \tag{2.38}$$

with:
g_0: Gravitational acceleration at the Earth's surface. I_{sp}: Specific impulse. m_i: Initial mass. t_i: Start date of the thrust $\vec{F}$: Thrust in Newtons.

Mean thrust calculation

$t_f = t_i + \Delta t$
$T_f = \min(t_f, t_2)$
$T_i = \max(t_i, t_1)$
With $t_i - t_f$ being the interval between the start and the end of the manoeuvre and $t_1 - t_2$ the interval between the start and the end of the Orbital Window.
The mean thrust, considered constant in each of the six steps of the Runge-Kutta Integrator, is calculated as:

$$\gamma_{Mean} = \begin{cases} \gamma_e + \frac{\gamma_{Thrust}(T_i) + \gamma_{Thrust}(T_f)}{2} \frac{T_f - Ti}{t_2 - t_1} & \text{if } T_f - T_i > 0 \\ 0 & \text{if } T_f - T_i <= 0 \end{cases} \tag{2.39}$$

2.4.5 Future perspectives for the orbital dynamics

The current acceleration model of the navigator is very accurate and adapted to orbital missions; however, improvements are always necessary to achieve even greater accuracy, especially in interplanetary flight and manoeuvre-intensive missions. These improvements mainly concern SRP, Thrust and the Hill and Relativistic forces.

- The **SRP** currently considers a more precise satellite model: the plate model. However, the calculation of the acceleration does not take into account the current attitude law. When the navigator will be able to manage this law (by telecommand, see section 3.3) this aspect of the dynamics will need to be improved.
- The current **Thrust** model is greatly simplified, and this works well with simple missions; however, in certain missions with few measurements available, the Kalman Filter might have problems correcting the error.
- The **Hill** acceleration is a periodic perturbation linked to orbital dynamics, depending on the satellite frequency and a modelled noise; the effect is small but useful to refine the long-term model.
- The **relativistic** acceleration is a correction stemming from general relativity, depending on the mass and rotation of the central body; the influence is very small but not negligible over long durations or for high precision orbits.
- One could consider modelling a **drag** acceleration adapted to other bodies in the solar system with an atmosphere (Mars, Venus, ...)

Even though there are always possible improvements, the current dynamic model has been tested and it is very accurate for navigation throughout the solar system.

2.5 Modelling the attitude

In the MATLAB Navigator, the attitude of the satellite is defined as the rotation between the inertial frame and the platform frame.
Originally, the satellite was represented by a sphere in order to simplify the model. To improve the reliability of interplanetary dynamics, it was later modelled by the *plate model* (see 2.4.3). This description makes it possible to link the attitude to the orientation of the plates in the inertial frame. *Note*: this does not affect the drag model, for which the description by faces does not intervene.
The attitude is involved in three functions:

- computing the acceleration due to solar radiation pressure;
- precise positioning of the GNSS / station / inter-satellite antennas for modelling theoretical measurements;
- performing optical *Line of Sight* (LOS) measurements on *landmarks*.

The Navigator is not designed to propagate the attitude dynamics itself. At this stage, the attitude used in the Navigator comes from LOS measurements. In other words, in the absence of an explicitly defined attitude law, the applied orientation corresponds to pointing toward the *landmark* of the measurement.

2.5.1 General principle

The frames used by the *MATLAB Navigator* for the attitude are:

- the platform frame (PF): axes of the satellite (positions of antennas or cameras are defined in this frame);
- the inertial frame (IN): integration frame (CIRF or ICRF), also used for integration during the simulation.

The passage from (PF) to (IN) is provided, in each LOS measurement, in the form of an attitude quaternion:

$$q_{\text{att}} = q_{(PF)\to(IN)}.$$

This attitude is therefore used by the LOS measurements.

2.5.2 Future prospects: development of attitude laws

For applications other than those directly related to LOS measurements (and, where applicable, for LOS when the attitude is not provided), one envisages defining generic attitude laws:

- geocentric law (GEOCENTRIC);

- yaw–steering law (YAW-STEERING);
- heliocentric law (HELIOCENTRIC).

The frames associated with these laws are:

- (QSW): Q radial (centre of the Earth $\rightarrow$ satellite), W normal to the orbit, S completes the triad (approximately oriented along the velocity);
- (TRL): pitch–roll–yaw (*Tangage–Roulis–Lacet*), defined by a constant rotation with respect to QSW;
- $(TRL)_0$: frame fixed with respect to the platform; it coincides with TRL if the roll, pitch and yaw angles are zero. The passage between TRL and $(TRL)_0$ is defined by the attitude law.

These laws must position the axes of the platform frame with respect to QSW. The following conventions are adopted:

- $(TRL) \rightarrow (QSW)$ is defined by the constant matrix

$$P_{(TRL)\rightarrow(QSW)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix};$$

- $(TRL)_0 \rightarrow (TRL)$ is defined by the attitude law, in the form of an attitude matrix:

$$M_{\text{att}} = P_{(TRL)_0\rightarrow(TRL)};$$

- $(PF) \rightarrow (TRL)_0$ is fixed once and for all in the mission parameters, in the form of the matrix

$$P_{(PF)\rightarrow(TRL)_0}.$$

The overall passage between (PF) and (QSW) is therefore written

$$P_{(PF)\rightarrow(QSW)} = P_{(PF)\rightarrow(TRL)_0}\, P_{(TRL)_0\rightarrow(TRL)}\, P_{(TRL)\rightarrow(QSW)}.$$

Implications for the models. The formulation by attitude laws—although it has not yet been integrated—mainly affects the solar radiation pressure model:

- **Geocentric**: fixed orientation in the local orbital frame; the attitude matrix is the identity, $M_{\text{att}} = P_{(TRL)_0\rightarrow(TRL)} = I$. The panel normal vector is zero: $\vec{N}_{PF} = \vec{0}$.
- **Heliocentric**: the panels are oriented toward the Sun.
- **Body-pointing**: law assumed during LOS *Landmark* measurements, pointing toward the barycentre of a celestial body.

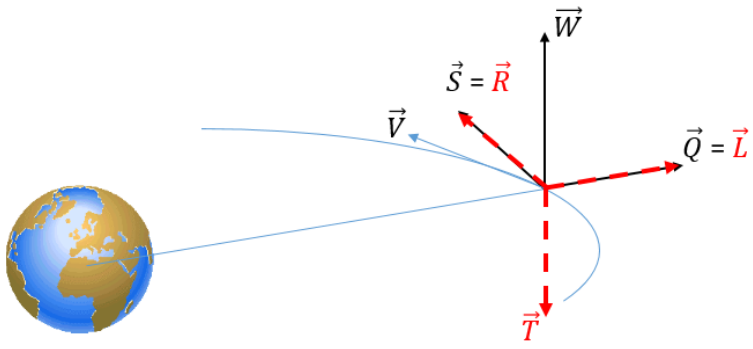


Figure 2.6: QSW and pitch–roll–yaw (TRL) frames

The change of attitude law is driven by telecommands.

Chapter 3

Interplanetary Navigation

Navigation is defined as the ability to know the satellite's state (position, speed, etc.) at any time. In the Interplanetary Navigator, navigation is performed by an Extended Kalman Filter, which uses the previously explained physical models to predict the next state and the measurements to correct said state. In the end, measurements are the heart of navigation.

3.1 Measurements

In the space environment the most reliable kind of measurement is the GNSS measurement; however, the satellite needs to be in Earth's proximity to work. Before my arrival, most available measurements were GNSS measurements. However, for interplanetary applications, new measurements have to be introduced and tested in order for the navigator to estimate the current state outside of Earth's influence. As interplanetary measurements, the most important is the Line of Sight measurement, as well as a LiDAR[1]-like range measurement. Before discussing interplanetary measurements, here is a general overview of all the measurements available before the interplanetary modifications.

3.1.1 GNSS measurements

The GNSS Measurements are the main kind of measurements used by the navigator in terrestrial orbits. All the details on these is shown in Appendix C

Code measurement

The receiver continuously receives the code transmitted by the GNSS satellite, with a delay corresponding to the travel time of the signal between the transmitting and receiving satellites. The navigator uses this delay as a measurement.

[1]LiDAR: Light Detection and Ranging, also called *two-way range measurement*

Phase measurement

Imperfect synchronisation between the receiver and satellite clocks introduces an error into the phase measurement that must be estimated. This error can be used to estimate the satellite's state.

Code difference measurement

The simple difference measurement is constructed from two measurements from the same transmitting GNSS satellite P_e to two different receivers P_{r1} and P_{r2}.
In the navigator, the PR measurements are expressed as emission dates in GNSS time t_e^{GNSS}.

Phase difference measurement

The simple difference phase measurement follows the same principle as the simple difference code measurement, being constructed from two measurements from the same transmitting GNSS satellite P_e to two different receivers P_{r1} and P_{r2}.

Inter-satellite measurement

The *ISL distance* measurement corresponds to the propagation time of the signal between a transmitting satellite and a receiving satellite. It is obtained by comparing the emission date and the reception date, assuming that the clocks are synchronised and the antennas placed at the centre of mass of the satellites. This measurement therefore makes it possible to access directly the instantaneous distance between the two satellites in an inertial frame, and constitutes an essential datum for intersatellite navigation.

3.1.2 Interplanetary measurements: Line of Sight

Consider a satellite equipped with a camera capable of carrying out optical measurements, which we call the "observer", and a target being observed, which may be of various types (centroid of a body, landmark on a body, another satellite, etc.). The measurement developed is the so-called "Line of Sight Landmark" measurement, which consists of the coordinates of the target in the image. Knowing the actual position of the landmark as well as the characteristics of the camera, it is then possible to compare the theoretical coordinates with the measured coordinates.

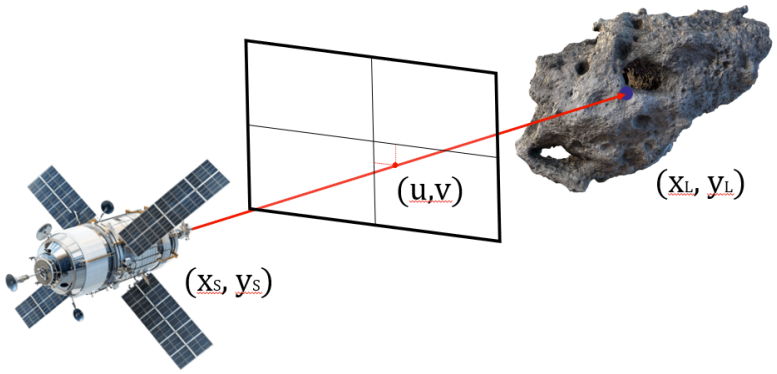


Figure 3.1: Line of Sight measurement

In the MATLAB navigator, the target is often the barycentre of a body, with the satellite pointing towards the central body (law *BodyCenterPointing*); however, the Landmark measurement is flexible enough to be used on other bodies or other targets. The position of the barycentre is known from the ephemerides and, knowing the position of the satellite and the pointing direction of the camera, it is possible to compute the measurements in an inertial frame. The attitude is necessary to compute the theoretical measurements in the camera frame.

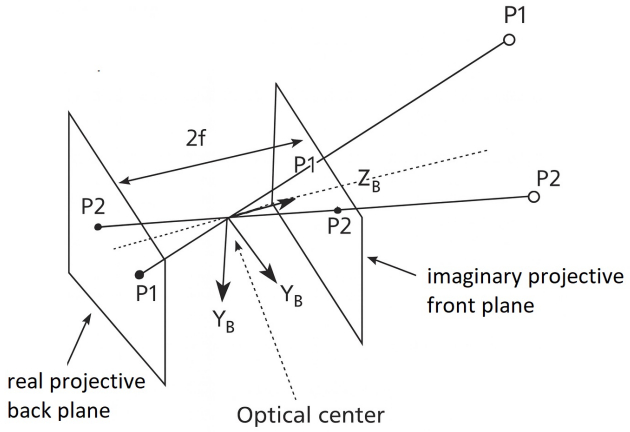


Figure 3.2: Line of Sight measurement

Calculation of the theoretical measurement

We measure u and v, which are the pixel coordinates of the target in the image plane. We do not take into account the light propagation time. Let c^I be the position of the target in the inertial frame and p^I_{sat} the position of the observer satellite in the inertial frame. The vector linking the camera to the point of interest (Line of Sight) is:

$$L^I = \begin{bmatrix} c_x^I - (p_x^I + \Delta_x) \\ c_y^I - (p_y^I + \Delta_y) \\ c_z^I - (p_z^I + \Delta_z) \end{bmatrix} \tag{3.1}$$

with Δ_i the vector from the centre of the satellite to the camera in the inertial frame. Expressing this vector in the camera frame and denoting M_{IC} the transformation matrix from the inertial frame to the camera frame, we have:

$$L^C = M_{IC} L^I \tag{3.2}$$

We then find:

$$u = f \frac{L_x^C}{L_z^C} \tag{3.3}$$

$$v = f \frac{L_y^C}{L_z^C} \tag{3.4}$$

with f the focal distance of the camera.
The transformation matrix from the inertial frame to the camera frame, M_{IC}, is a matrix described by two rotations:

$$M_{IC} = M_{PC} \cdot M_{IP} \tag{3.5}$$

where

- M_{PC}: the transformation matrix between the platform frame and the camera frame, described in the initial MissionData
- M_{IP}: the rotation matrix between the inertial frame and the platform frame, which effectively describes the attitude of the satellite. The attitude of the satellite is provided by the measurement directive.

All of the rotations are described by quaternions. The transformation between a quaternion and a rotation matrix is performed with the MATLAB function *quaternion.m* (and conversely with *rotmat.m*), with type equal to "point". The rotation of the line of sight vectors is performed by the function *rotatepoint(q, v)*.

Calculation of partial derivatives with respect to the observer's position

We seek the partial derivatives of the measurement with respect to the position of the satellite in the inertial frame.
We have:

$$\frac{\partial u}{\partial p_x^I} = f\left(\frac{\frac{\partial L_x^C}{\partial p_x^I} L_z^C - L_x^C \frac{\partial L_z^C}{\partial p_x^I}}{(L_z^C)^2}\right) \tag{3.6}$$

under the assumption that $\frac{\partial f}{\partial p_x^I} = 0$.
Furthermore, under the assumption that $\frac{M_{ICx}}{\partial p_x^I} = 0$ and that $\frac{\partial \Delta_x}{\partial p_x^I} = 0$.

$$\frac{\partial L^C}{\partial p_x} = M_{IC}\frac{\partial L^I}{\partial p_x} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial L_x^I}{\partial p_x} \\ \frac{\partial L_y^I}{\partial p_x} \\ \frac{\partial L_z^I}{\partial p_x} \end{bmatrix} = \begin{bmatrix} -M_{11} \\ -M_{21} \\ -M_{31} \end{bmatrix} \tag{3.7}$$

$$\frac{\partial L^C}{\partial p_y} = \begin{bmatrix} -M_{12} \\ -M_{22} \\ -M_{32} \end{bmatrix} \tag{3.8}$$

$$\frac{\partial L^C}{\partial p_z} = \begin{bmatrix} -M_{13} \\ -M_{23} \\ -M_{33} \end{bmatrix} \tag{3.9}$$

The final result is therefore:

$$\boxed{\frac{\partial u}{\partial p_x^I} = \frac{f(L_x^C M_{31} - L_z^C M_{11})}{(L_z^C)^2}} \tag{3.10}$$

$$\boxed{\frac{\partial u}{\partial p_y^I} = \frac{f(L_x^C M_{32} - L_z^C M_{12})}{(L_z^C)^2}} \tag{3.11}$$

$$\boxed{\frac{\partial u}{\partial p_z^I} = \frac{f(L_x^C M_{33} - L_z^C M_{13})}{(L_z^C)^2}} \tag{3.12}$$

For the component v of the measurement, by the same method and with the same assumptions we find:

$$\boxed{\frac{\partial v}{\partial p_x^I} = \frac{f(L_y^C M_{31} - L_z^C M_{21})}{(L_z^C)^2}} \tag{3.13}$$

$$\boxed{\frac{\partial v}{\partial p_y^I} = \frac{f(L_y^C M_{32} - L_z^C M_{22})}{(L_z^C)^2}} \tag{3.14}$$

$$\boxed{\frac{\partial v}{\partial p_z^I} = \frac{f(L_y^C M_{33} - L_z^C M_{23})}{(L_z^C)^2}} \tag{3.15}$$

3.1.3 Interplanetary measurement: LiDAR

As explained in Sec. 3.1.2, Line of Sight (LOS) measurements alone are insufficient for interplanetary navigation. The LOS vector conveys angular information about the satellite's position, but not its range to the target. Navigating without range causes radial errors to accumulate, leading to divergent estimates. For this reason, it is necessary to introduce an additional interplanetary measurement for distance estimation.
LiDAR is a two-way type of measurement: the instrument emits a signal toward a celestial object, the signal reflects off the surface, and the satellite detects the return. By recording the times of emission and reception (with the reflection occurring at the surface), the satellite infers the target's distance.

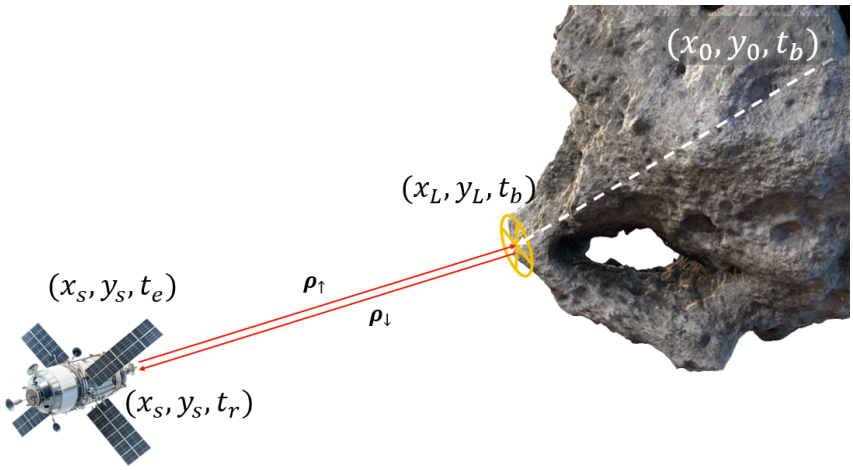


Figure 3.3: LiDAR measurement representation

In both LiDAR and LOS cases, the satellite points at its target, which may be the body's barycentre or a surface landmark of known position. The attitude follows a body-pointing law; however, for LiDAR the attitude is not critical, since the measurement time matters more than the exact pointing vector. The landmark position, or rather its position relative from a known ephemeris (body baricenter), is assumed to be well known from onboard data.
The validation phase of the navigator (sec. 4.1) will show the necessity of a range measurement such as LiDAR, which allows the navigator to correctly perform manoeuvres and be utilised in non-orbital trajectories.

Calculation of the theoretical measurement

The measurement dynamics are defined by three instants:

- the instant of emission of the signal, t_e
- the instant of reflection at the surface, t_b
- the instant of reception of the signal, t_r

The measurement is represented by three pieces of information: the reception time t_r, the emission time t_e, and the distance of the landmark. Also, the navigator must be able to compute the landmark's position.
The measurement describes two distances: the *backward distance* $\rho_\downarrow$, defined as the norm of the difference between the satellite's position at reception, $\vec{P}_r(t_r)$, and the landmark's position at reflection, $\vec{P}_L(t_b)$,

$$\rho_\downarrow \;=\; \left\|\vec{P}_r(t_r) - \vec{P}_L(t_b)\right\|. \tag{3.16}$$

and the *forward distance*, defined as the norm of the difference between the landmark's position at reflection, $\vec{P}_L(t_b)$, and the satellite's position at emission, $\vec{P}_r(t_e)$,

$$\rho_\uparrow \;=\; \left\|\vec{P}_L(t_b) - \vec{P}_r(t_e)\right\| \tag{3.17}$$

The reflection time is defined as

$$t_b \;=\; t_r \;-\; \frac{\rho_\downarrow}{c} = t_r - \frac{\left\|\vec{P}_L(t_b) - \vec{P}_r(t_e)\right\|}{c} \tag{3.18}$$

Since t_b appears on both sides of the computation, an iterative method is required to compute it. Starting from an initial guess $t_b < t_r$:

- Compute the landmark position at t_b: $\vec{P}_L(t_b)$.
- Compute the backward distance: $\rho_\downarrow = \|\vec{P}_r(t_r) - \vec{P}_L(t_b)\|$.
- Update t_b to t_b' using $\rho_\downarrow$:
$$t_b' \;=\; t_r \;-\; \frac{\rho_\downarrow}{c}$$
- Compute the error $\epsilon = |t_b - t_b'|$ and iterate.

When the error falls below a prescribed threshold (usually $10^{-6}m$ or $10^{-12}m$), t_b (and with it $\vec{P}_L(t_b)$) is determined.
Given the emission time t_e, the forward distance is

$$\rho_\uparrow = ||\vec{P}_L(t_b) - \vec{P}_r(t_e)|| \tag{3.19}$$

Finally, the modeled range measurement M is computed as

$$M = \rho_\uparrow + \rho_\downarrow + 2c\,\Delta t_{\text{instr}}, \tag{3.20}$$

which also accounts for the instrument delay Δt_{instr}.
Because this solution is obtained iteratively, convergence depends on the actual range. In practice, LiDAR is directed towards relatively nearby targets (typically up to a few thousand kilometers). For more distant objects, the algorithm may fail to converge and, even if it does, the range information is weak: very distant landmarks are nearly stationary

relative to the spacecraft, so $\vec{P}_L(t_b)$ changes little and the modeled measurement M becomes ill-conditioned and error-prone.

Calculation of partial derivatives with respect to the observer's state

Since all state updates are done at the end of an orbital window, all partial derivatives have to be transitioned to the end of the orbital window t_{final}.

$$\boxed{\left.\frac{\partial M(t_r)}{\partial x}\right|_{t_{final}} = \left.\frac{\partial M(t_r)}{\partial x}\right|_{t_r} \frac{\partial x(t_r)}{\partial x(t_{final})} = \left.\frac{\partial M(t_r)}{\partial x}\right|_{t_r} \mathbf{\Phi}_{t_r \to t_{final}}} \tag{3.21}$$

where $\mathbf{\Phi}_{t_r \to t_{final}}$ is the transition matrix between t_r and t_{final}. We linearize the spacecraft kinematics about a chosen reference epoch t_f (local constant–velocity model):

$$\vec{P}_r(t) \;\approx\; \vec{P}_r \;+\; \vec{V}_r\,(t - t_f), \tag{3.22}$$

The transition to t_f is not a problem for position related derivatives, however it is important to take into account for velocity derivatives.

$$\frac{\partial \vec{P}_r(t)}{\partial \vec{P}_r} = I, \tag{3.23}$$

$$\frac{\partial \vec{P}_r(t)}{\partial \vec{V}_r} = (t - t_f)\, I \;=\; -\,I\,(t_f - t). \tag{3.24}$$

The line-of-sight unitary vectors (as in the theoretical measurement):

$$\hat{\mathbf{u}}_{\uparrow} \;=\; \frac{\vec{P}_L(t_b) - \vec{P}_r(t_e)}{\rho_{\uparrow}}, \qquad \hat{\mathbf{u}}_{\downarrow} \;=\; \frac{\vec{P}_r(t_r) - \vec{P}_L(t_b)}{\rho_{\downarrow}}.$$

Using $\partial\|\mathbf{a}\|/\partial\mathbf{a} = \mathbf{a}^{\top}/\|\mathbf{a}\| = \mathbf{u}^{\top}$ and the chain rule, we obtain:

Derivative w.r.t. position at t_f. Both legs depend on $\vec{P}_r$ only through $\vec{P}_r(t_r)$ and $\vec{P}_r(t_e)$; therefore

$$\boxed{\left.\frac{\partial M}{\partial \vec{P}_r}\right|_{t_f} \;=\; \hat{\mathbf{u}}_{\downarrow} \;+\; \hat{\mathbf{u}}_{\uparrow}} \tag{3.25}$$

Derivative w.r.t. velocity at t_f. Using the chain rule under the constant–velocity linearization, we write

$$\left.\frac{\partial M}{\partial \boldsymbol{V}_r}\right|_{t_{final}} = \frac{\partial M}{\partial \boldsymbol{P}_r}\frac{\partial \boldsymbol{P}_r(t)}{\partial \boldsymbol{V}_r(t_f)} = \hat{\mathbf{u}}_{\downarrow}\frac{\partial \boldsymbol{P}_r(t_r)}{\partial \boldsymbol{V}_r(t_f)} + \hat{\mathbf{u}}_{\uparrow}\frac{\partial \boldsymbol{P}_r(t_e)}{\partial \boldsymbol{V}_r(t_f)}$$

From 3.24, the kinematic Jacobians are

$$\frac{\partial P_r(t_r)}{\partial V_r(t_f)} = I\left(t_r - t_f\right) \tag{3.26}$$

$$\frac{\partial P_r(t_e)}{\partial V_r(t_f)} = I\left(t_e - t_f\right) \tag{3.27}$$

Substituting these expressions yields

$$\left.\frac{\partial M}{\partial \vec{V}_r}\right|_{t_f} = \hat{\mathbf{u}}_{\downarrow}\left(t_r - t_f\right) \; + \; \hat{\mathbf{u}}_{\uparrow}\left(t_e - t_f\right) \tag{3.28}$$

Define the time gaps

$$\Delta t_{\uparrow} \; = \; t_r - t_e, \qquad \Delta t_r \; = \; t_r - t_f,$$

Therefore, the velocity derivative becomes

$$\boxed{\left.\frac{\partial M}{\partial \vec{V}_r}\right|_{t_f} \; = \; \left(\hat{\mathbf{u}}_{\downarrow} + \hat{\mathbf{u}}_{\uparrow}\right)\Delta t_r \; - \; \hat{\mathbf{u}}_{\uparrow}\,\Delta t_{\uparrow}} \tag{3.29}$$

The measurement matrix

More details on the processing of the measurements and their derivatives are provided in Section 3.2.3 for LOS and Lidar or in [3] and [4] for GNSS measurements.

3.2 Kalman Filter

The Kalman filter (KF) is designed to optimally estimate the state of a linear dynamic system when the system model and the measurements are subject to Gaussian noise. The Kalman filter assumes that the system dynamics and the measurement equations are linear. The extended Kalman filter (EKF) operates by linearising the non-linear system dynamics and the measurement model around the current state estimate (and not around a reference trajectory). This linearisation is performed at each iteration of the filter.
In fact, the filter is modelled as a loop, where each increment corresponds to an *orbital window* (FO), i.e. a predefined time interval between two extrapolations. In a classical Kalman filter, the prediction (or extrapolation) step is performed up to the time of the next available measurement. However, in this particular context, the approach is different: the prediction is made at regular intervals defined by these orbital windows, as explained in 1.3.1.

Dynamic model

Suppose that the quantities of interest are realisations of a random process, $\boldsymbol{x}(t)$, whose distribution we know at an initial time t_0 and whose temporal evolution follows the following stochastic differential equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{B}(t)\boldsymbol{w}(t) \tag{3.30}$$

with $\boldsymbol{B}(t)$ the control matrix of the noise.
Assume that the initial distribution of $\boldsymbol{x}(t_0)$ and the distribution of $\boldsymbol{w}(t)$ (model noise) are Gaussian, with a mean and a covariance given by:

$$E[\boldsymbol{x}(t_0)] = \bar{\boldsymbol{x}}_0 \quad \text{and} \quad E[(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}_0)(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}_0)^T] = \boldsymbol{P}_0 \tag{3.31}$$

$$E[\boldsymbol{w}(t)] = 0 \quad \text{and} \quad E[\boldsymbol{w}(t)\boldsymbol{w}^T(\tau)] = \boldsymbol{Q}(t)\delta(t-\tau) \tag{3.32}$$

We can also write this as: $\boldsymbol{x}(t_0) \sim\ N(\bar{\boldsymbol{x}}(t_0), \boldsymbol{P}_0)$ and $\boldsymbol{w}(t) \sim\ N(0, \boldsymbol{Q})$.
With $\boldsymbol{Q}(t)$ the power spectral density of $\boldsymbol{w}(t)$ and $\delta(t-\tau)$ the Dirac delta.
We also assume that:

$$E[\boldsymbol{w}(t)(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}_0)^T] = 0, \quad \forall t \tag{3.33}$$

Note that even if $\boldsymbol{x}(t_0)$ and $\boldsymbol{w}(t)$ are assumed Gaussian, $\boldsymbol{x}(t)$ is not Gaussian for $t > t_0$, because $\boldsymbol{f}$ is a non-linear function.

Measurement model

The dynamic model is defined in the continuous domain. However, the measurements are noisy representations of the state elements obtained at discrete times t_k.

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.34}$$

We define similarly some relations:

$$E[\boldsymbol{v}_k] = 0 \quad \text{and} \quad E[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}_k \delta_{kj} \tag{3.35}$$

With $\boldsymbol{R}(t)$ the power spectral density of $\boldsymbol{v}(t)$ and δ_{kj} the Kronecker delta. We similarly assume that:

$$E[\boldsymbol{v}_k(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}_0)^T] = 0 \quad \text{and} \quad E[\boldsymbol{w}(t)\boldsymbol{v}_k^T] = 0, \quad \forall k, t \tag{3.36}$$

As previously, it should be noted that $\boldsymbol{y}_k$ does not remain Gaussian for all k, because $\boldsymbol{h}$ is a non-linear function.

Extended Kalman Filter (EKF)

We rewrite the continuous-discrete Extended Kalman Filter (CT-DT EKF) equations:

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{B}(t)\boldsymbol{w}(t) \\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{v}_k \end{cases} \tag{3.37}$$

The algorithm is described below:

Algorithm 1 Extended Kalman Filter (EKF)

Inputs: $\hat{\boldsymbol{x}}_{k-1}^{+}$ and $\boldsymbol{P}_{k-1}^{+}$
Outputs: $\hat{\boldsymbol{x}}_{k}^{+}$ and $\boldsymbol{P}_{k}^{+}$

(1) Extrapolate state by numerical integration (ode45)

$$\hat{\boldsymbol{x}}_{k}^{-} = \int_{t_{k-1}}^{t_k} \boldsymbol{f}(\boldsymbol{x}(\tau), \tau) d\tau = \boldsymbol{\phi}_{k-1,k}(\hat{\boldsymbol{x}}_{k-1}^{+}), \qquad \boldsymbol{x}(t_{k-1}) = \hat{\boldsymbol{x}}_{k-1}^{+}, \qquad \boldsymbol{x}_0^{+} = \bar{\boldsymbol{x}}_0$$

(2) Compute the transition matrix between $k-1$ and k : $\boldsymbol{\Phi}_{k-1,k}$ (see 3.2.2)

(3) Extrapolate the covariance

$$\boldsymbol{P}_{k}^{-} = \boldsymbol{\Phi}_{k-1,k} \boldsymbol{P}_{k-1}^{+} \boldsymbol{\Phi}_{k-1,k}^{T} + \boldsymbol{Q}_k, \quad \boldsymbol{P}_0^{+} = \boldsymbol{P}_0$$

(4) Compute the estimated theoretical measurement $\hat{\boldsymbol{y}}_k$ and the measurement matrix $\boldsymbol{H}_k$

$$\hat{\boldsymbol{y}}_k = \boldsymbol{h}(\hat{\boldsymbol{x}}_{k}^{-}) \qquad \text{and} \qquad \boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k}^{-}}$$

(5) Compute the Kalman gain

$$\boldsymbol{K}_k = \boldsymbol{P}_{k}^{-} \boldsymbol{H}_{k}^{T} (\boldsymbol{H}_k \boldsymbol{P}_{k}^{-} \boldsymbol{H}_{k}^{T} + \boldsymbol{R}_k)^{-1}$$

(6) Correct state and correct covariance

$$\hat{\boldsymbol{x}}_{k}^{+} = \hat{\boldsymbol{x}}_{k}^{-} + \boldsymbol{K}_k (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)$$

$$\boldsymbol{P}_{k}^{+} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k}^{-} (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^{T} + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_{k}^{T}$$

Filter tuning:

- Initial covariance on state vector: $\boldsymbol{P}_0$
- Model errors: $\boldsymbol{Q}_k$
- Measurement noise covariance: $\boldsymbol{R}_k$

The theoretical definition of $\boldsymbol{Q}_k$ is written below. However, in practice it corresponds to a filter tuning parameter.

$$\boldsymbol{Q}_k = \int_{t_{k-1}}^{t_k} \int_{t_{k-1}}^{t_k} \boldsymbol{\phi}(t_k, \tau) \boldsymbol{B}(\tau) E[\boldsymbol{w}(t) \boldsymbol{w}^{T}(\sigma)] \boldsymbol{B}^{T}(\sigma) \boldsymbol{\phi}^{T}(t_k, \sigma) d\tau d\sigma \tag{3.38}$$

The extra difficulty of the EKF compared to a KF is due to the fact that at each iteration,

the linearisation must be updated. The linearisation is computed based on the current estimate, because no predefined nominal trajectory is available. Thus, to calculate $\boldsymbol{\phi}_{k-1,k}$, we use the previous estimate, and to calculate $\boldsymbol{H}_k$, we use the state prediction at the next step.
An explanatory diagram of the algorithm is also shown below.

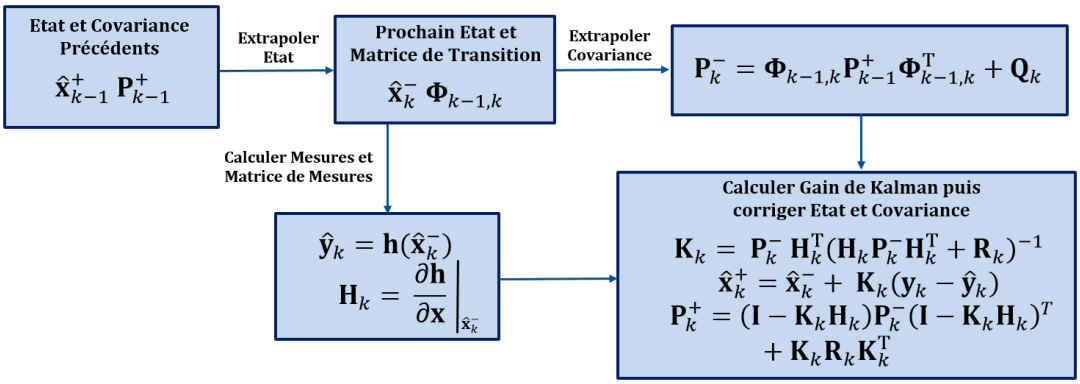


Figure 3.4: EKF algorithm

Remark : The covariance correction equation adopted is the *Joseph form*:

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{3.39}$$

This formula is totally equivalent to the equation previously used in the navigator:

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{3.40}$$

The first formulation is more complex, but it ensures that the covariance matrix remains strictly symmetrical, which was not always the case with the old formulation. Indeed, the old approach could lead to non-symmetric covariance matrices and negative diagonal values $\sigma_{x_i}^2$ during the first iterations of the filter. The *Joseph form*, on the other hand, is symmetric by construction and solves these problems linked to numerical approximations when subtracting very large and similar numbers.

The use of the *Joseph form* is common practice in the implementation of sequential filters. Its main advantage is that it guarantees that the estimation error covariance matrix $\boldsymbol{P}_k^+$ is positive definite, at the cost of a slightly greater computational effort.

3.2.1 State Vector

Here we describe the state vector that the Navigator uses for estimation via the Kalman Filter. The state consists of all the simulation variables that must be estimated and corrected during navigation. In particular, they are arranged in the following order for each receiver satellite:

State vector parameter	Notation	Size
Position	$\mathbf{r}$	(3,1)
Velocity	$\mathbf{v}$	(3,1)
Clock	$\mathbf{H}$	(3,1)
Solar Radiation Pressure coefficient	$\mathbf{C}_{SRP}$	(1,1)
Drag coefficient	$\mathbf{C}_f$	(1,1)
Phase bias	$\mathbf{B}$	(nb_{sat},1)
Thrust Error	$\gamma_{\mathbf{e}}$	(3,1)

Table 3.1: State vector

The state vectors of the nb_{rec} satellites are concatenated. Each state vector is of variable size depending on the forces or measurements considered in the *Mission Data* and *Directives* for each satellite.

The last term, the thrust-error, is itself a state estimated by the Kalman Filter. It is initialized to zero, and at the first thrust event the following settings are applied:

- The thrust-error state is set to the zero vector, $\mathbf{0}_{3\times1}$.
- The covariance matrix $\mathbf{P}$ initializes the thrust-error block as a diagonal matrix $\text{diag}(\text{Cov}_I, \text{Cov}_I, \text{Cov}_I)$, where Cov_I is specified in the thrust telecommand.
- The additive process-noise matrix (for the thrust-error state) is initialized as $\text{diag}(\text{Cov}_M, \text{Cov}_M, \text{Cov}_M)$, where Cov_M is specified in the thrust telecommand.
- The transition Matrix takes into account two new matrices for the thrust effects.

3.2.2 Transition matrix Φ

Let $i_{nb_{rec}}$ the number of receiver satellites, the state transition matrix from state t_k to state t_{k+1} is as follows:

$$\mathbf{\Phi}_{k-1,k} = \begin{pmatrix} \mathbf{\Phi}_{Sat_1} & 0 & 0 & 0 \\ 0 & \mathbf{\Phi}_{Sat_2} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & \mathbf{\Phi}_{Sat_{i_{nb_{rec}}}} \end{pmatrix} \tag{3.41}$$

In turn, the state transition matrix associated with a satellite is as follows:

$$\boldsymbol{\Phi}_{Sat_i} = \begin{pmatrix} \boldsymbol{\Phi}_{PV_i} & 0 & \boldsymbol{\Phi}_{C_{s_i}\to PV_i} & \boldsymbol{\Phi}_{C_{f_i}\to PV_i} & 0 & \boldsymbol{\Phi}_{Man\to PV_i} \\ 0 & \boldsymbol{\Phi}_{H_i} & 0 & 0 & 0 & 0 \\ 0 & 0 & \Phi_{C_{s_i}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_{C_{f_i}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{\Phi}_{B_{n_i}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{\Phi}_{Man} \end{pmatrix} \tag{3.42}$$

For applications of this type, it is desirable to have an easily computable form for the state transition matrix, which is accurate for short time intervals. For $\boldsymbol{\Phi}_{PV}$, only the central force is considered because it is well known that the transition matrix does not need to be calculated as accurately as the state itself.

The matrix $\boldsymbol{G}$ is the gradient matrix of the gravitational field: it describes how a small change in position influences the gravitational force experienced by a satellite. The calculation model for $\boldsymbol{G}$ only takes into account the potential of the central body and, if we are in an Earth orbit, the *Earth flattening* J_2, neglecting other terms.

Currently, the MATLAB Navigator uses the J_2 model around the Earth and the following model around another star.

$$\boldsymbol{G}_k = -GM\left(\frac{\boldsymbol{I}_3}{\|\boldsymbol{P}_{k^-}\|^3} - 3\frac{\boldsymbol{P}_{k^-}\boldsymbol{P}_{k^-}^T}{\|\boldsymbol{P}_{k^-}\|^5}\right), \quad \boldsymbol{G}_{k^+} = -GM\left(\frac{\boldsymbol{I}_3}{\|\boldsymbol{P}_{k^+}\|^3} - 3\frac{\boldsymbol{P}_{k^+}\boldsymbol{P}_{k^+}^T}{\|\boldsymbol{P}_{k^+}\|^5}\right) \tag{3.43}$$

$$\boldsymbol{H}_G = \boldsymbol{I}_3 + \frac{\Delta t^2}{6}\left(2\boldsymbol{G}_k + \boldsymbol{G}_{k^+}\right), \tag{3.44}$$

$$\boldsymbol{H}_D = \Delta t\left(\boldsymbol{I}_3 + \frac{\Delta t^2}{12}\left(\boldsymbol{G}_k + \boldsymbol{G}_{k^+}\right)\right), \tag{3.45}$$

$$\boldsymbol{B}_G = \frac{\Delta t}{2}\left(\boldsymbol{G}_k + \boldsymbol{G}_{k^+}\right), \tag{3.46}$$

$$\boldsymbol{B}_D = \boldsymbol{I}_3 + \frac{\Delta t^2}{6}\left(\boldsymbol{G}_k + 2\boldsymbol{G}_{k^+}\right), \tag{3.47}$$

$$\boldsymbol{\Phi}_{PV} = \begin{bmatrix} \boldsymbol{H}_G & \boldsymbol{H}_D \\ \boldsymbol{B}_G & \boldsymbol{B}_D \end{bmatrix} \tag{3.48}$$

Taking into account only the central term in the acceleration we have:

$$\vec{\gamma} = -\frac{\mu}{\|P\|^3}\vec{P} \tag{3.49}$$

And therefore the partial derivatives:

$$\frac{\partial \vec{\gamma}}{\partial \vec{P}} = -\mu(\frac{1}{\|P\|^3} I_{3\times 3} - 3\vec{P}\frac{\vec{P}^T}{\|P\|^5}) \tag{3.50}$$

The factor $\vec{P}\vec{P}^T$ is a dyadic product, which gives a 3×3 matrix and should not be confused with the dot product. We thus have the following expression for G:

$$G = \frac{\partial \vec{\gamma}}{\partial P} = \frac{\mu}{\|P\|^5} \begin{pmatrix} 3x^2 - r^2 & 3xy & 3xz \\ 3yx & 3y^2 - r^2 & 3yz \\ 3zx & 3zy & 3z^2 - r^2 \end{pmatrix} \tag{3.51}$$

This calculation of G is suitable for orbits where the central body's acceleration is predominant. For other applications where 3rd-body forces or SRP are not negligible it would be appropriate to include them in the calculation of G [5] [2].
For the other dynamic parameters:
The clock bias state: Φ_H

$$\mathbf{\Phi}_H = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.52}$$

The drag coefficient being modelled as constant: $\mathbf{\Phi}_{Cf_i} = 1$

$$\mathbf{\Phi}_{Cf_i->PV_i} = \begin{pmatrix} \frac{\Delta t^2}{2}\vec{\gamma}_{drag} \\ \Delta t\vec{\gamma}_{drag} \end{pmatrix} \tag{3.53}$$

Similarly, for the SRP : $\mathbf{\Phi}_{SRP_i} = 1$

$$\mathbf{\Phi}_{SRP_i->PV_i} = \begin{pmatrix} \frac{\Delta t^2}{2}\vec{\gamma}_{SRP} \\ \Delta t\vec{\gamma}_{SRP} \end{pmatrix} \tag{3.54}$$

For the phase bias: $\mathbf{\Phi}_{Bias_i} = 1$
For the thrust error: $\mathbf{\Phi}_{Man_i} = I_3$

$$\mathbf{\Phi}_{Man_i->PV_i} = \begin{pmatrix} \frac{\Delta t^2}{2} M_{X->inertial} \\ \Delta t M_{X->inertial} \end{pmatrix} \tag{3.55}$$

with $X = QSW$ or TNW or $Inertial$ depending on the thrust frame

3.2.3 Measurement matrix *H*

The measurement matrix H is the matrix which carries all the measurements' M partial derivatives with relation to the state vector x. It derives from the measurement function

$h(x)$ and the state x

$$\mathbf{h}(\mathbf{x}) = \Big[M\Big], \qquad \mathbf{x}^\top = \big[\, \mathbf{r}^\top \;\; \mathbf{v}^\top \;\; \mathbf{H}^\top \;\; C_{SRP} \;\; C_f \;\; \mathbf{B}^\top \;\; \boldsymbol{\gamma}_e^\top \big],$$

with $\mathbf{r}, \mathbf{v}, \mathbf{H}, \gamma_\mathbf{e} \in \mathbb{R}^3$, $C_{SRP}, C_f \in \mathbb{R}$ and $\mathbf{B} \in \mathbb{R}^{nb_{sat}}$. (Here $\mathbf{H}$ denotes the clock, not to be confused with the measurement matrix H.) The measurement matrix is the Jacobian

$$H \;=\; \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \;\in\; \mathbb{R}^{2\times(11+nb_{sat})}.$$

For each measurement, the measurement matrix carries different types of derivatives. In this navigator, the two interplanetary measurements available are the Line of Sight and the LiDAR, while there are still the GNSS measurements available.

The construction of $\mathbf{H}_{LOS}$

For the case of LOS measurements, the position of the target is expressed in an inertial frame whose origin coincides with the centre of the satellite's central body, with the same origin and date as for p^I_{sat}. The image coordinates u and v are treated as two distinct measurements.
The measurement function is

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} u \\ v \end{bmatrix}$$

In the current model (without propagation time, focal length and alignments known, without instantaneous coupling), the LOS depends at time t only on the observer's position $\mathbf{r}$. We obtain

$$H \;=\; \big[\; \underbrace{H_\mathbf{r}}_{2\times3} \;\big|\; \underbrace{\mathbf{0}}_{2\times3} \;\big|\; \underbrace{\mathbf{0}}_{2\times3} \;\big|\; \underbrace{\mathbf{0}}_{2\times1} \;\big|\; \underbrace{\mathbf{0}}_{2\times1} \;\big|\; \underbrace{\mathbf{0}}_{2\times nb_{sat}} \;\big|\; \underbrace{\mathbf{0}}_{2\times3} \;\big].$$

The non-zero block $H_\mathbf{r}$ groups the derivatives of (u, v) with respect to $\mathbf{r} = (x, y, z)$ in the inertial frame. With L^C the LOS vector in camera, M_{IC} the inertial→camera matrix and f the focal length, we have

$$H_\mathbf{r} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{bmatrix} = \tag{3.56}$$

$$= \begin{bmatrix} \dfrac{f\,(L^C_x M_{31} - L^C_z M_{11})}{(L^C_z)^2} & \dfrac{f\,(L^C_x M_{32} - L^C_z M_{12})}{(L^C_z)^2} & \dfrac{f\,(L^C_x M_{33} - L^C_z M_{13})}{(L^C_z)^2} \\ \dfrac{f\,(L^C_y M_{31} - L^C_z M_{21})}{(L^C_z)^2} & \dfrac{f\,(L^C_y M_{32} - L^C_z M_{22})}{(L^C_z)^2} & \dfrac{f\,(L^C_y M_{33} - L^C_z M_{23})}{(L^C_z)^2} \end{bmatrix}. \tag{3.57}$$

The other blocks are zero: the velocity $\mathbf{v}$, the clock $\mathbf{H}$, C_{SRP}, C_f and the GNSS biases $\mathbf{B}$ do not affect the instantaneous optical projection in this model.

The construction of $\mathbf{H}_{LiDAR}$

In the current model we simplify the spacecraft's dynamics and consider the clock to be correct: the LiDAR measurement only depends on the satellite's position and speed

$$\mathbf{h}(\mathbf{x}) = \Big[\rho\Big]$$

$$H \;=\; \big[\; \underbrace{H_{\mathbf{r}}}_{1\times3} \;\big|\; \underbrace{H_{\mathbf{v}}}_{1\times3} \;\big|\; \underbrace{\mathbf{0}}_{1\times3} \;\big|\; \underbrace{\mathbf{0}}_{1\times1} \;\big|\; \underbrace{\mathbf{0}}_{1\times1} \;\big|\; \underbrace{\mathbf{0}}_{1\times nb_{sat}} \;\big|\; \underbrace{\mathbf{0}}_{1\times3} \;\big].$$

The non-zero blocks $H_{\mathbf{r}}$ and $H_{\mathbf{r}}$ group the derivatives of the distance ρ with respect to $\mathbf{r} = (x, y, z)$ and v_x, v_y, v_z in the inertial frame.

$$H_{LiDAR} = \begin{bmatrix} \frac{\partial\rho}{\partial x} & \frac{\partial\rho}{\partial v_x} \\ \frac{\partial\rho}{\partial y} & \frac{\partial\rho}{\partial v_y} \\ \frac{\partial\rho}{\partial z} & \frac{\partial\rho}{\partial v_z} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \hat{u}_{x_\downarrow} + \hat{u}_{x_\uparrow} & (\hat{u}_{x_\downarrow} + \hat{u}_{x_\uparrow})\,\Delta t_r \;-\; \hat{u}_{x_\uparrow}\,\Delta t_\uparrow \\ \hat{u}_{y_\downarrow} + \hat{u}_{y_\uparrow} & (\hat{u}_{y_\downarrow} + \hat{u}_{y_\uparrow})\,\Delta t_r \;-\; \hat{u}_{y_\uparrow}\,\Delta t_\uparrow \\ \hat{u}_{z_\downarrow} + \hat{u}_{z_\uparrow} & (\hat{u}_{z_\downarrow} + \hat{u}_{z_\uparrow})\,\Delta t_r \;-\; \hat{u}_{z_\uparrow}\,\Delta t_\uparrow \end{bmatrix}^{\mathsf{T}} \tag{3.58}$$

Although higher precision would require to take into account the clock derivatives, the actual model is simplified and considers the time clock to be correct.
The partial derivatives calculated previously will each have its place in the matrix $\boldsymbol{H}$. Each row corresponds to a measurement; the navigator stores the available measurements in a specific order consistent with the measurement noise matrix $\boldsymbol{R}$.

Complete measurement matrix H

The matrix H used by the MATLAB Navigator takes into account all the available measurements, whether GNSS or LOS. The structure of the matrix uses a precise order. We consider a code measurement for satellite 1 c_1, a code measurement for satellite 2 c_2, a phase measurement for satellite 1 ph_1, a phase measurement for satellite 2 ph_2, a single difference code measurement $c_2 - c_1$, a single difference phase measurement $ph_2 - ph_1$, an ISL distance measurement d_1, the two Line of Sight measurements LOS_u and LOS_v, one for each image coordinate, and finally the LiDAR range measurement ρ, in this precise order. The GNSS measurements are all attached to the same transmitting satellite.

$$\boldsymbol{H} = \begin{pmatrix} H_{m_1} & 0 \\ 0 & H_{m_2} \\ H_{l_1} & 0 \\ 0 & H_{l_2} \\ -H_{m_1} & H_{m_2} \\ -H_{l_1} & H_{l_2} \\ H_{d_1} & H_{d_2} \\ H_{LOS_{u1}} & 0 \\ H_{LOS_{v1}} & 0 \\ 0 & H_{LOS_{u2}} \\ 0 & H_{LOS_{v2}} \\ H_{\rho_1} & 0 \\ 0 & H_{\rho_2} \end{pmatrix}$$

We build the measurement function matrix $\boldsymbol{H}_k$. The number of rows is equal to the number of measurements and the number of columns is equal to the sum of the state vector of the 2 receiver satellites. Each sub-matrix H_i is associated with measurement i and contains the partial derivatives.

3.2.4 Covariance Matrix $\boldsymbol{P}$

According to the theory of the Extended Kalman Filter, the propagation of the covariance matrix is carried out according to the following formula:

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1,k} \boldsymbol{P}_{k-1}^+ \boldsymbol{\Phi}_{k-1,k}^T + \boldsymbol{Q}_k \tag{3.59}$$

The first step consists in determining the model noise matrix $\boldsymbol{Q}_k$. The model error is related to the dynamics (velocity, acceleration), while the position is obtained by integration. Consequently, there is no model error necessary for the position. For example, it is useful to define $\boldsymbol{Q}_k$ as being zero and to observe the estimation error to determine at what point it begins to diverge.
Suppose a model error of 10^{-9} in acceleration after 30 seconds. The model noise on velocity should then be:

$$\boldsymbol{Q}_{V-dt} = 10^{-9} \times 30 = 3 \times 10^{-8} \text{ m}^2/\text{s}^2$$

Thus, for an orbital window containing n transmitting satellites, the model noise matrix is

defined as follows:

$$\boldsymbol{Q}_{kdt} = \begin{pmatrix} \boldsymbol{Q}_{P-dt} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{Q}_{V-dt} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{Q}_{H-dt} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{Q}_{C_f-dt} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{Q}_{C_{SRP}-dt} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{Q}_{Bias-dt} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{Q}_{Man-dt} \end{pmatrix}$$

where each sub-matrix $\boldsymbol{Q}_{i-dt}$ is square and symmetric. In particular:

$$\boldsymbol{Q}_{Bias-dt} = \begin{pmatrix} \boldsymbol{Q}_{Bias1-dt} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{Q}_{Bias2-dt} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{Q}_{Biasn-dt} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The matrix $\boldsymbol{Q}_{Bias-dt}$ is of size *NbChannels* $\times$ *NbChannels*, with *NbChannels* being the maximum number of channels associated with the GNSS transmitting satellites. This parameter is defined in the *Mission Data*.
To obtain $\boldsymbol{Q}_k$, we use the following relation which links the continuous model to the discrete model:

$$\boldsymbol{Q}_{kdt} = \boldsymbol{Q}_k \times \Delta t$$

Although this matrix is constant, it is necessary to take into account the fact that the number of phase biases changes with time. The model noises are initialised only if there is indeed a transmitting satellite; otherwise, the value is reset to zero. This distinction is made based on the list of PRN numbers of the transmitting satellites. The conditioning of the covariance matrix $\boldsymbol{P}_{k-1}^{+}$ is based on the PRN numbers of the satellites of each orbital window. Thus, from the information contained in the list of PRN numbers of the transmitting satellites, the covariance matrix is recreated.

3.3 Telecommands

Interplanetary missions are more complex in several respects. Many mission parameters are likely to change, and these changes must be carried out during the simulation. The current method implemented in the navigator consists of providing precise indications of the parameters to change at a specific instant: the telecommands. That is why the MATLAB navigator must be able to read and update the telecommands at input.

The telecommands are written in the file *tc*, and they can modify several aspects of navigation.

- DMANO. Allows the definition of a thrust acceleration. The start of the thrust will be the activation instant of the telecommand.
- DCORPSC. Allows the change of the central body of the mission. Fundamental for interplanetary journeys.
- DCORPSP. Allows the change of the perturbing bodies and their active forces.
- DYN_MODEL. Allows the definition of the accelerations to consider during navigation

All the telecommands are read at the same time as the directives and they are stored in *simu.TCs*. Activation of a telecommand is carried out at the beginning of the orbital window that contains the telecommand activation time (except for DMANO which actually begins to compute the thrust acceleration at the exact instant indicated by the telecommand).
At the beginning of each orbital window, before starting extrapolation, the MATLAB Navigator executes the method *simu.executerTelecommandes*, which makes it possible to check whether the telecommands *DCORPSC, DCORPSP* and *DYN_MODEL* are to be executed in this window.

TC DMANO

The telecommand format indicates

- start date of the thrust (cjd in days and seconds)
- F_x, F_y, F_z the forces in the three axes;
- m_i the mass at the start of the thrust;
- dt the total duration of the thrust;
- the frame in which the forces are given (usually *INERTIEL*);
- I_{sp} of the thruster.

The acceleration is calculated by the formula indicated in 2.4.4, and it is computed at each integration step.

TC DCORPSC

DCORPSC is the telecommand responsible for changing the central body. In the case of DCORPSC, the change of central body implies a change of state of the satellite: one computes the distance and the velocity between the first central body and the second

$$P_{CC_1 \to CC_2} = P_{CC_2 \to SSB} - P_{CC_1 \to SSB} \tag{3.60}$$

$$V_{CC_1 \to CC_2} = V_{CC_2 \to SSB} - V_{CC_1 \to SSB} \tag{3.61}$$

and afterwards the state of the satellite is modified

$$P_{sat \to CC_2} = P_{sat \to CC_1} + P_{CC_1 \to CC_2} \tag{3.62}$$

$$V_{sat \to CC_2} = V_{sat \to CC_1} + V_{CC_1 \to CC_2} \tag{3.63}$$

In addition, this telecommand also changes the current potential acceleration model: the harmonic coefficients of the potential, the radius and the gravitational constant of the central body are updated.

TC DCORPSP

Often used with the change of central body, the telecommand DCORPSP allows the modification of the active forces of the perturbing bodies. The forces are indicated by booleans: direct or tide. *Remark*: the Love coefficients for the tide acceleration (2.4.2) are modelled only for the Moon and the Earth; the other bodies always have a zero tide acceleration.

TC DYN_MODEL

By a list of booleans, it is possible to modify the active forces during the simulation. The order of the booleans is as follows:

- Acceleration of the central body
- Drag acceleration
- Solar radiation pressure
- Hill acceleration
- Relativistic acceleration

Future prospects

The four telecommands managed by the MATLAB Navigator are the most important for performing navigation during interplanetary missions. However, there are several other

telecommands that can be introduced: change of attitude law or rejection thresholds for measurements, etc.

Chapter 4

Validation Tests

4.1 Test generation

To validate the navigator, it is necessary to test it on different scenarios. However, generating suitable missions is not simple. Real data from interplanetary missions is rare and hard to find, and there is not yet a CNES mission providing *Line of Sight* nor *LiDAR* type measurements.
Creating such measurements is also complex because it requires propagating a complete mission and simulating the associated observations. To meet this need, a tool was developed in Java with the Patrius library. The tool I developed during the internship allows to simulate a mission, generate the corresponding measurements and produce the input files necessary to test the navigator. The development also depends on other CNES libraries such as Lotus, Orbito and Hera Project.

Landmark Measurement

The Line of Sight measurement generation was handled by the *LandmarkMeasurement* class from LOTUS, which makes it possible to make Landmark measurements towards a celestial body. In particular, the aim of the tests was to simulate a landmark located at the barycentre of the celestial body and to use the *generateSimulatedMeasurement* function to generate simulated measurements with Gaussian noise.
The *LandmarkMeasurement* class uses the same formulation as the Navigator for Line of Sight measurements (3.1.2), taking into account both the rotation between the camera and the satellite and the attitude of the satellite itself.

Relative Two Way Distance Measurement

The LiDAR-type measurement was also generated by a class from LOTUS, the *RelativeTwoWaysDistanceMeasurement*, which makes it possible to generate LiDAR measurements towards a landmark with a known ephemeris. In particular, the aim of the tests was to simulate a landmark located at the barycentre of the celestial body, while the distance

measurement is taken from the reflection on the celestial body's surface, and to use the *generateSimulatedMeasurement* method to generate simulated measurements with Gaussian noise.

Finally, the LOS and LiDAR measurements are written to the directives file in a format readable by the Navigator.

4.1.1 The mission generator

Interplanetary missions are often very complicated: for this reason, the tool has been designed to be simple to use, easily extensible and modifiable, both in the mission dynamics and in the measurement parameters.
The interplanetary mission is divided into phases, each defined by a set of specific parameters. The phase constructor, *PhaseConfig*, takes all this data as input

- **Name:** phase identifier.
- **Central body:** body around which the phase is propagated.
- **Integration frame:** frame used for propagation.
- **Start date:** initial instant of the phase.
- **End date:** final instant of the phase.
- **Initial orbit:** starting orbit (only for the first phase).
- **Perturbing bodies:** list of bodies whose direct acceleration should be taken into account.
- **Occulting bodies (SRP):** bodies capable of occulting the sun (influences the SRP).
- **Attitude:** applied attitude law (generally *BodyCenterPointing*).
- **Generate measurements (genMeasures):** boolean indicator to activate or not the generation of measurements.
- **useDrag:** boolean indicator to activate or not drag forces.
- **Interpolation step:** time step used for interpolation.
- **LOS Measurement step:** time step between each LOS measurement.
- **LiDAR Measurement step:** time step between each LiDAR measurement.
- **Stop event:** condition which interrupts propagation before the end date.
- **Data on manoeuvres:** parameters defining the manoeuvres carried out during the phase.

The MATLAB Navigator is designed to always consider a central body; during an interplanetary trajectory, the central body can change several times. This division into phases makes it possible to better manage the complete mission and to generate the telecommands necessary for changes during the phases.
It is also possible to modify the measurement conditions: type of camera, standard deviation, resolution, orientation on the satellite and, in addition, bias on the camera or on the position of the planet's barycentre.

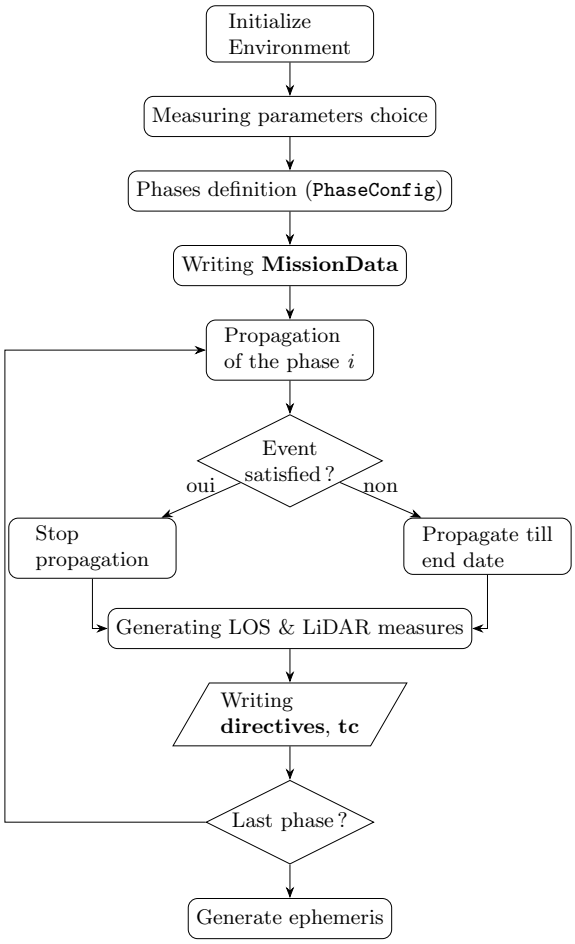


Figure 4.1: Structure of the test generation code

All the mission parameters are written into the MissionData and the progress of the mission is described by the directives and the telecommands.
The propagator simulates each phase taking into account the specified forces and thrusts, generating the measurements at each specified interval and producing the input files for the Navigator as well as a final reference ephemeris.

4.2 Orbital tests

An orbital trajectory is a simple trajectory carried around a central body, without manoeuvres, change of central bodies and close to the planet. It is the simplest type of environment for space navigation, and it is the environment for which navigator has been developed. Orbital tests are used to validate certain functionalities of the Navigator. All the tests carried out were generated with Patrius (sec. 4.1).
Among the functionalities of the Navigator, those that we wish to validate are:

- Extrapolation in ICRF (instead of CIRF);
- Calculation of the acceleration potential around another planet;
- Accuracy of the conversion to BCF;
- Calculation of the position of the perturbing bodies via the ephemerides;
- Modelling of the satellite plates for solar radiation pressure (SRP);
- Navigation by *Line of Sight* (LOS): stability of the measurements, reliability of the navigation and limits.
- Navigation by both *Line of Sight* and *LiDAR*: stability of the measurements, reliability of the navigation and limits.

We expect very precise results for state propagation alone; these performances depend on the dynamic model, which must be correctly modelled. On the other hand, LOS navigation should be less precise. As explained in sec. 3.1.2, LOS measurements are essentially angular (phase measurements) and poorly suited to estimating the radial distance: we therefore expect the radial error not to be corrected by the filter and the estimate to diverge in the long term. With LOS+LiDAR navigation, finally, we should obtain the best possible accuracy the navigator can reach.

4.2.1 Moon

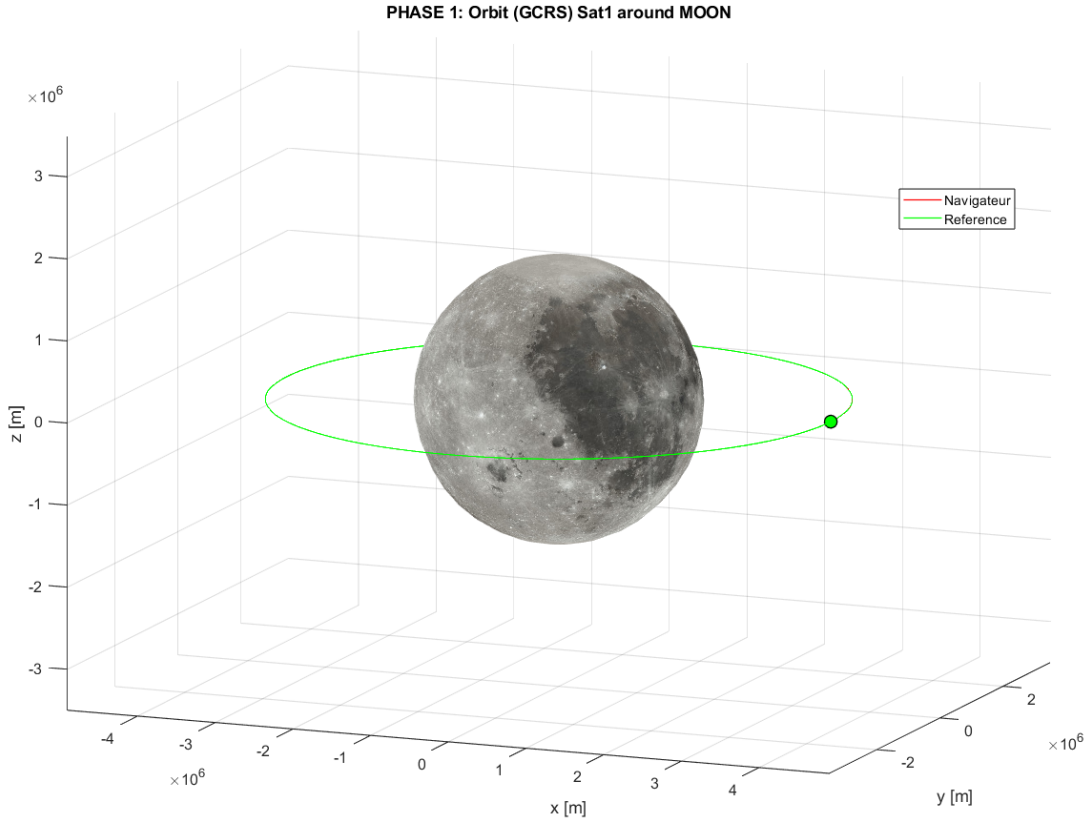


Figure 4.2: Orbit around the Moon

Test configuration

The dynamical model is configured to use the Moon as the central body, with its gravitational potential represented up to degree and order 20. Only the central lunar gravity field is active in this test. All additional forces (drag, solar radiation pressure, relativistic corrections and Hill approximation) are disabled in the dynamic settings.

Quantity	Value
Central body	Moon
Perturbing bodies	None
Reference frame	ICRF
Epoch (CJD)	26333.00043
Epoch (calendar)	05 February 2022 00:00:37.15
Time scale	TAI
Semi-major axis a	3.4985×10^6 m
Eccentricity e	4.22×10^{-4}
Inclination i	0°
RAAN Ω	0°
Argument of periapsis ω	0°
True anomaly	180°
Duration	24 h
Extrapolation step	10
LOS measurement step	10
LiDAR measurement step	1

Table 4.1: Initial orbital elements for Test #1 (single spacecraft around the Moon).

Initial satellite's state:

$$\mathbf{r}_0 = \begin{bmatrix} 3.5 \times 10^6 & 0 & 0 \end{bmatrix}^T \text{m}, \qquad \mathbf{v}_0 = \begin{bmatrix} 0 & 1183.553997 & 0 \end{bmatrix}^T \text{m/s}.$$

The process noise associated with the orbital state is specified as diagonal in the configuration file. Each Cartesian position component (X, Y, Z) is assigned a model noise covariance of 1×10^{-6} in the units of the state, while each velocity component (V_X, V_Y, V_Z) is assigned a model noise covariance of 1×10^{-12}. All other model noise entries are set to zero.
To quantify the error, the RMS will be computed. For position, the components of the error vector are denoted by $e_{p,\mathrm{R}}(t_i)$, $e_{p,\mathrm{AT}}(t_i)$ and $e_{p,\mathrm{CT}}(t_i)$, while for velocity, the corresponding components are $e_{v,\mathrm{R}}(t_i)$, $e_{v,\mathrm{AT}}(t_i)$ and $e_{v,\mathrm{CT}}(t_i)$.
For each component, the root-mean-square (RMS) error is computed over the N samples of the arc according to

$$\mathrm{RMS}_{r,k} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_{p,k}(t_i)^2}, \qquad \mathrm{RMS}_{v,k} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_{v,k}(t_i)^2},$$

with $k \in \{\mathrm{R}, \mathrm{AT}, \mathrm{CT}\}$.

Tests without measurements

In order to validate the accuracy of the dynamic model, we use a test around the Moon without any measurement, to compare the orbit propagated by the navigator with the real orbit.
The performance of the navigator is evaluated in the local orbital frame associated with the reference trajectory. At each epoch, the estimation error is defined as the difference between the estimated and reference state, projected onto the radial (R), along-track (AT) and cross-track (CT) directions.

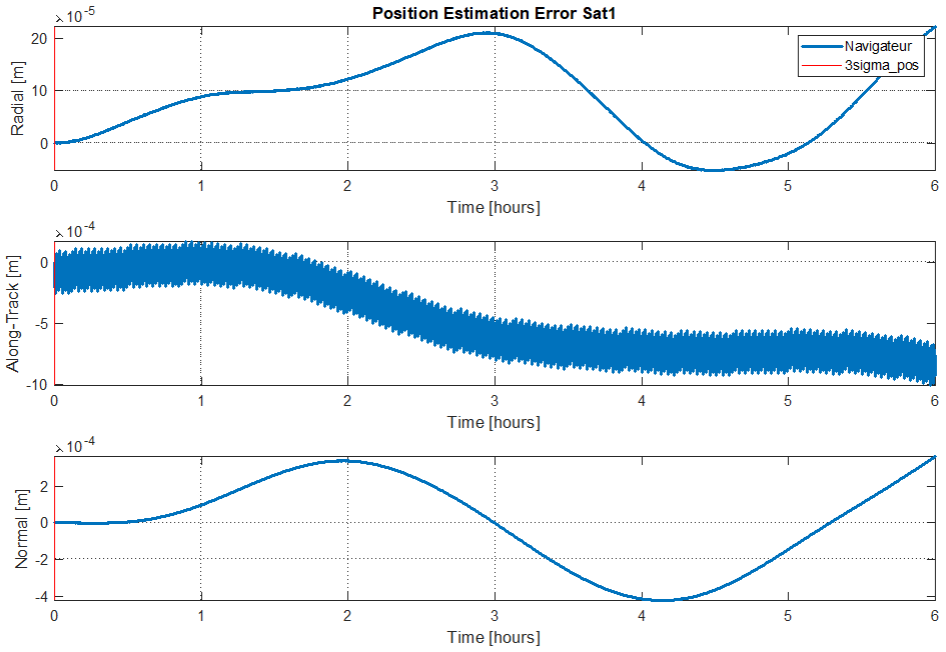


Figure 4.3: Results of the simulation without measurements around the Moon

The estimation errors are obtained by transforming the x,y,z ICRF coordinates into QSW coordinates (as explained in 2.1.3), in order to observe the radial, along-track or normal errors to the orbital plane. The errors show that the models used are extremely precise and follow the real orbit well. The Moon has a potential that is calculated to degree 20 and the acceleration models give an orbit almost perfectly equal to the reference one. This type of test is very useful for visualising the reliability of the dynamic model; however, without a Kalman filter, an error can easily propagate and cause the estimation to diverge while never being corrected. Therefore, the no-measurement test gives no information on the performance of the filter. The prediction of the state is very precise, but without error correction, an initial error would propagate throughout the whole mission.

Navigation by Line of Sight

A test with LOS measurements allows the navigation with only LOS measurements to be validated. The adopted scenario uses an entry orbit different from the real orbit; we expect the measurements to correct this error.
We introduce a simulated error of 100 m on the tangential component of the satellite's position. LOS navigation being angular only, we expect the radial component to be unstable and diverge, while the Kalman filter remains capable of correcting the error and correctly estimating the along-track and normal components.

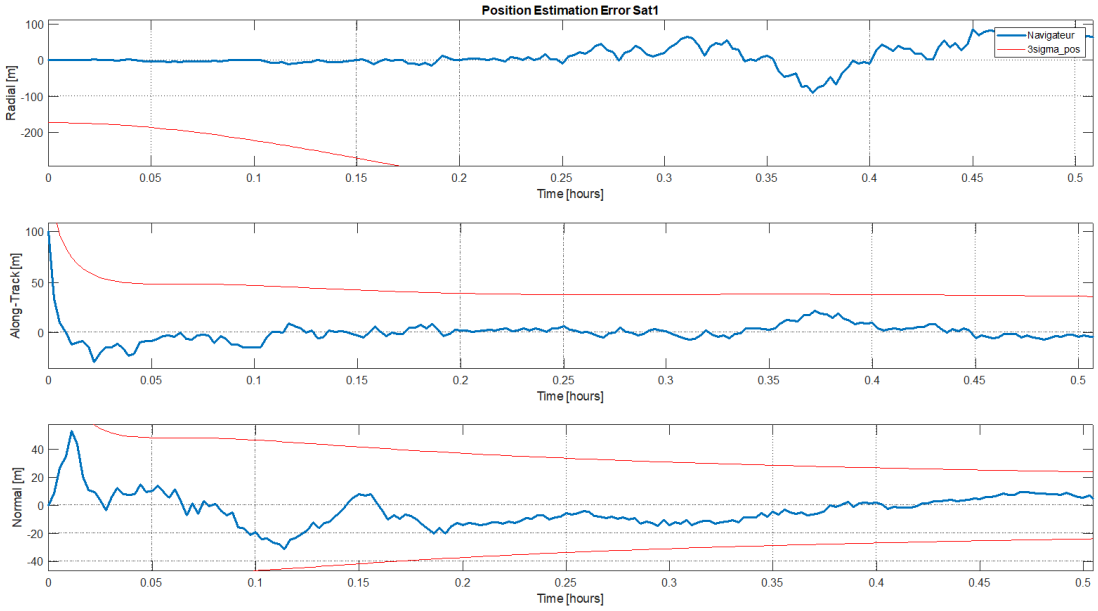


Figure 4.4: Correction of the orbit error: estimation of the first 30 minutes

We observe that the initial error is corrected in a few seconds. The filter correctly estimates the satellite's position, even if the input measurements are noisy. Oscillations persist, however, especially at the beginning of the test, because LOS measurements are rarely used without other sensors, such as a LIDAR. In addition, on board the satellite, LOS image processing can be computationally expensive; the number of measurements is therefore reduced. Here, one LOS measurement per orbital window (every 30 s) was simulated. Finally, in the long term, navigation remains precise.

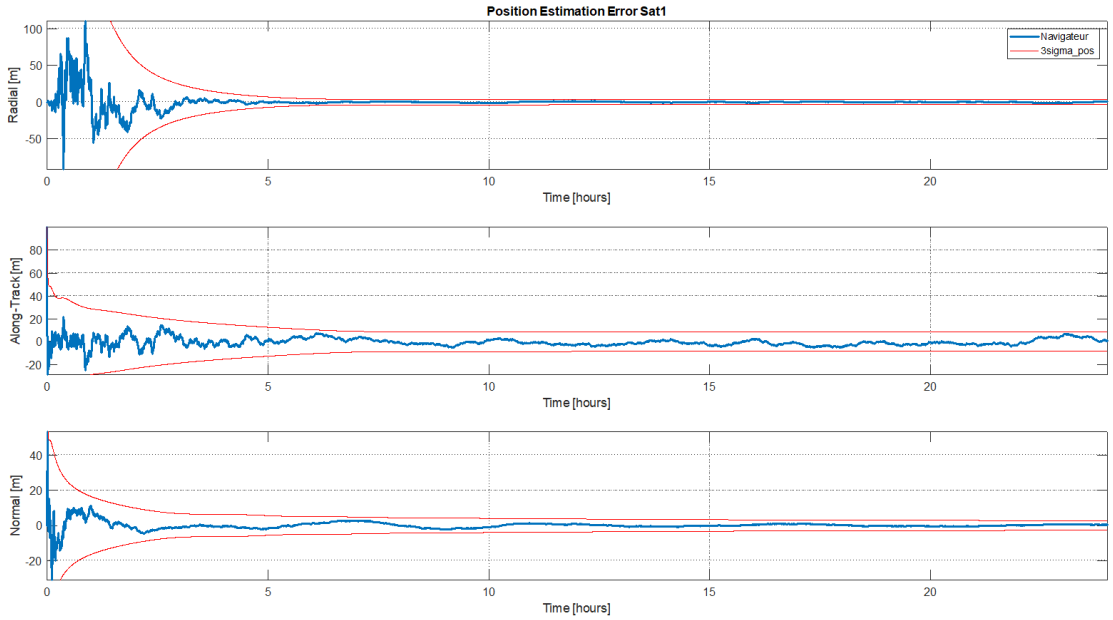


Figure 4.5: Position estimation over the entire test

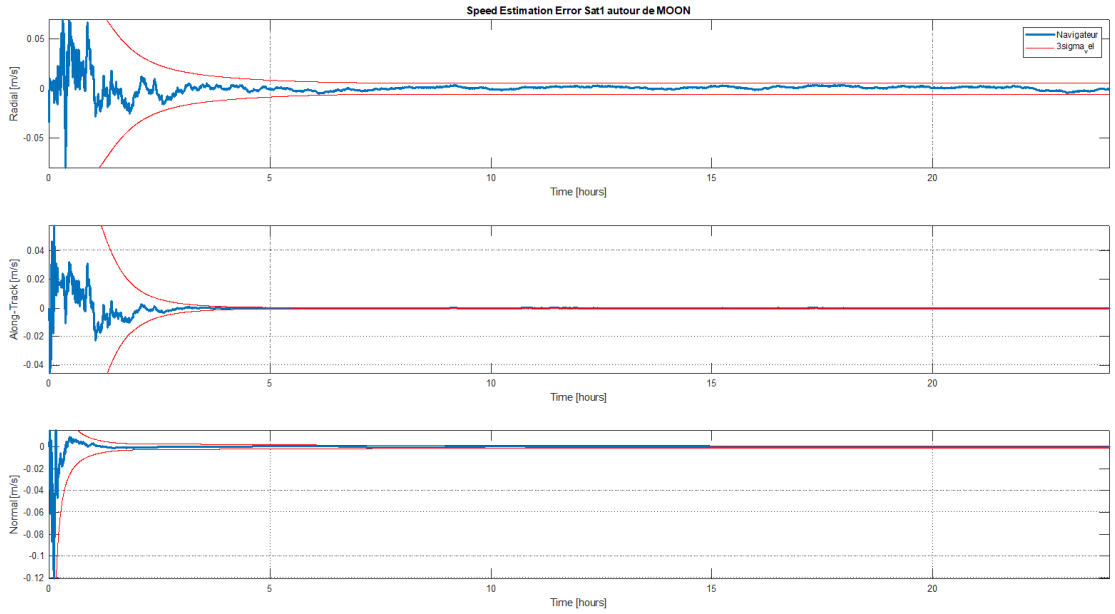


Figure 4.6: Velocity estimation over the entire test

The resulting RMS values for position and velocity in the three directions are

Direction	RMS Position	RMS Speed
Radial	0.516931	0.001809
Along-Track	2.759374	0.000050
Normal	0.469412	0.000175
3D	2.846351	0.001818

Table 4.2: Error in position and speed with LOS measurements

Navigation by Line of Sight & LiDAR

As shown before, navigation by Line of sight requires a few hours to correct the error; however when covariance reduces, the navigator provides optimal results over the long term. Now, the objective is to compare navigation performed with only LOS to navigation that combines LOS and LiDAR. The result will obviously be much more precise since the LiDAR measurements are also much more frequent.

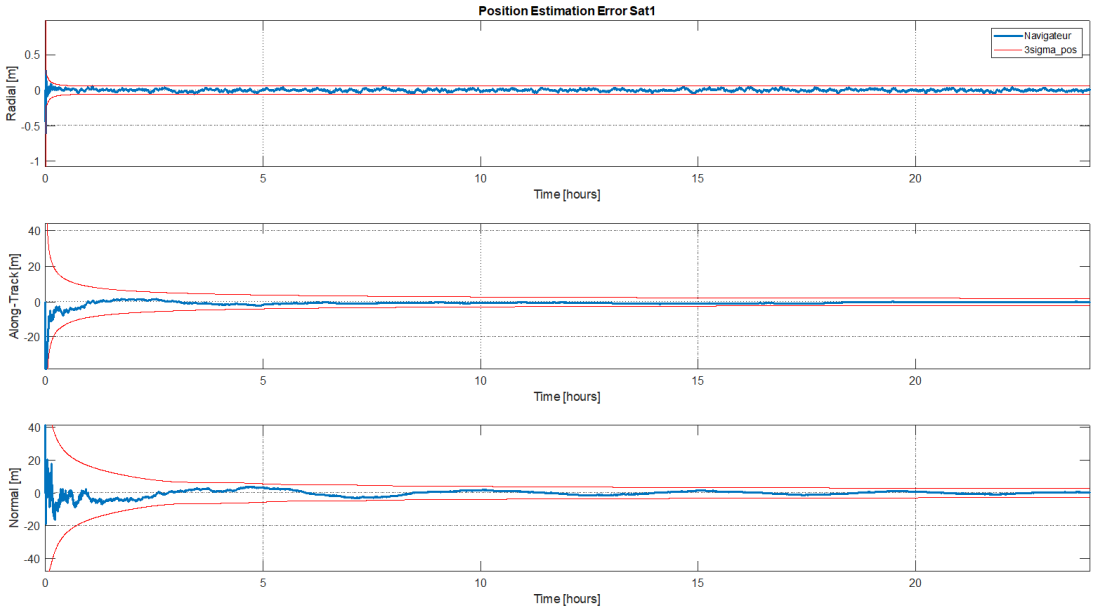


Figure 4.7: Position estimation over the whole mission, with measurements in LOS and LiDAR

As expected, the estimation is significantly more precise and stable, the navigator corrects the initial error in a matter of seconds rather than hours compared with LOS navigation. The overall precision reaches the tenth of millimetres although there still are very little oscillations in normal estimation.
The overall performance of the navigation can be understood analysing the RMS results.

Direction	RMS Position	RMS Speed
Radial	0.015784	0.000405
Along-Track	0.599101	0.000012
Normal	0.766491	0.000221
3D	0.972975	0.000461

Table 4.3: Error in position and speed with LOS+LiDAR measurements

This table clearly shows better results than the LOS-only navigation. This will be clear in the final comparison.

Final Comparison

Direction	RMS Position		RMS Speed	
	LOS only	LOS + LiDAR	LOS only	LOS + LiDAR
Radial	0.516931	0.015784	0.001809	0.000405
Along-Track	2.759374	0.599101	0.000050	0.000012
Normal	0.469412	0.766491	0.000175	0.000221
3D	2.846351	0.972975	0.001818	0.000461

Table 4.4: Comparison of RMS position and speed errors for LOS-only and LOS+LiDAR navigation.

As it is clearly shown, the LiDAR measurements give significantly more precise navigation on all directions, not only radial. Overall, a simple test like this shows an acceptable error also for LOS-only navigation, however orbits are rarely as simple as this one.

4.2.2 Didymos

A final orbital test is carried out around Didymos. Didymos is a binary asteroid (Didymos + Dimorphos) located at about 1.6 AU from the Sun. It is the target of the Hera mission (ESA), launched on 7 October 2024, which will reach the system at the end of 2026 after an interplanetary cruise, then will evolve in a quasi-orbit around the pair. It will deploy the Milani and Juventas CubeSats to map, probe the interior of Dimorphos and precisely measure the effects of the impact.

Here, we test the effectiveness of navigation around a small celestial body. For this case, the orbit is elliptical and retrograde, with a non-zero inclination and eccentricity. The satellite approaches up to 470 m above the surface.

Test configuration

The dynamical model is configured to use the Didymos system as the central body, with its gravitational potential represented up to degree and order 20. In addition to the central gravity field of Didymos, the perturbing bodies are the Moon, the Earth and the Sun. The only non-gravitational active force is the Solar Radiation Pressure.

Quantity	Value
Central body	Didymos
Perturbing bodies	Earth, Moon, Sun
Reference frame	ICRF
Epoch (CJD)	28139.56233
Epoch (calendar)	16 January 2027 13:29:45.657
Time scale	TAI
Semi-major axis a	1752.86
Eccentricity e	0.4866
Inclination i	140.00°
RAAN Ω	10.00°
Argument of periapsis ω	316.71°
True anomaly	93.29°
Duration	48 h
Extrapolation step	30 s
LOS measurement step	30
LiDAR measurement step	1

Table 4.5: Mission parameters for Test #2 (single spacecraft around Didymos).

Initial satellite's state:

$$\mathbf{r}_0 = \begin{bmatrix} 1011.366202 & -641.686084 & 677.622735 \end{bmatrix}^T \text{ m},$$

$$\mathbf{v}_0 = \begin{bmatrix} -0.047348 & -0.133598 & 0.103500 \end{bmatrix}^T \text{ m/s}.$$

The process noise associated with the orbital state is specified as diagonal in the configuration file, where the noise entries are the same as the previous mission.

Navigation by Line of Sight

The navigation results around a small body in a peculiar orbit highlight the limitations of relying solely on line-of-sight measurements. This test is performed with an initial error of $100, \mathrm{m}$ in both the along-track and normal position components, as well as a $0.5, \mathrm{m/s}$ error in the along-track velocity. In orbit around an asteroid, such errors are difficult to correct, making this scenario a stringent test of the navigator's ability to recover from significant errors in an unusual orbital configuration.

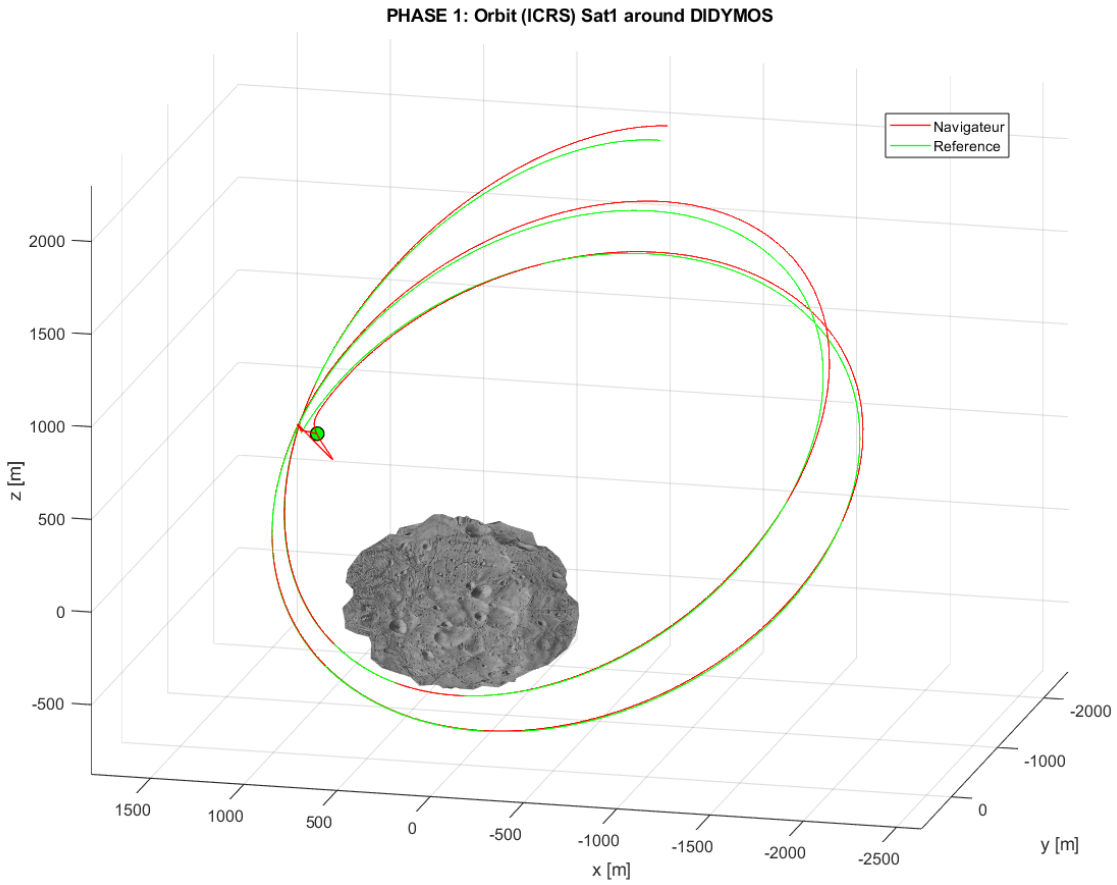


Figure 4.8: Orbit around Didymos with initial error. Note the correction of the error.

On figure 4.9, we observe the result of correcting the initial error.

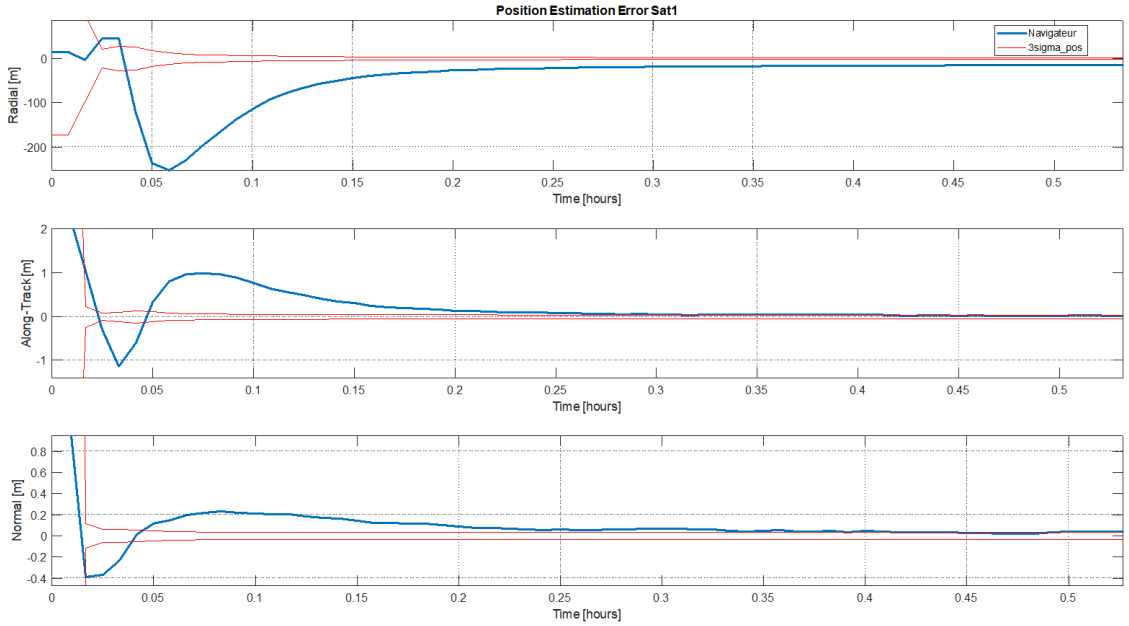


Figure 4.9: Correction of the initial error on the orbit around Didymos.

The initial error on the tangential and normal components is indeed corrected; however, the correction is slower on the radial and not definitive. In fact, around more massive bodies, gravitational dynamics exerts a more marked effect on the Kalman filter, which facilitates the reduction of errors; around Didymos, we therefore expect less precise performance, especially on the radial.

It can be seen that the residual error lies outside the 3σ bounds. This behaviour indicates an inconsistency in the physical modelling of the mission: a mismatch between the gravitational model used by the Navigator and the one implemented in Java used to generate the test. The discrepancy becomes larger as the spacecraft moves away from Didymos, but decreases as it approaches the central body. Unfortunately, Didymos' gravitational harmonics are CNES internal data, impossible to have access to. For this reason, all missions around Didymos show errors outside the expected covariance: the navigator believes it knows the true orbit but underestimates its uncertainties.

Although this behaviour is clearly undesired, Figure 4.10 shows that the navigator is able to correct the initial error; in this case it demonstrates the robustness of the navigator in correcting the error even when the available physical parameters are not precise. It also suggests that accurate navigation around unknown bodies is still achievable. However this only works in the directions the navigator has information on, which are Along Track and Normal. Without LiDAR measurements, the radial estimation diverges.

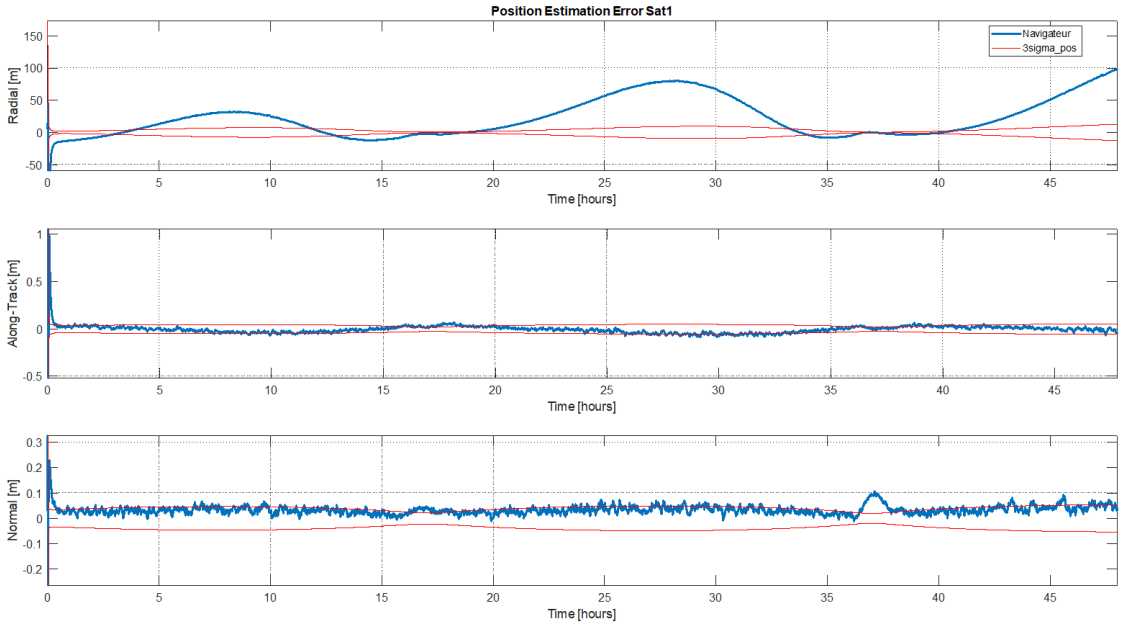


Figure 4.10: Complete estimation around Didymos.

The results obtained remain, nevertheless, sufficiently precise around non-radial directions, as shown in Table 4.6.

Direction	RMS Position	RMS Speed
Radial	48.216298	0.004028
Along-Track	0.034537	0.002389
Normal	0.040505	0.000310
3D	48.216327	0.004694

Table 4.6: Error in position and speed with LOS measurements

Navigation by Line of Sight & LiDAR

The navigator will now be tested in the same orbit and the same initial errors with the help of LiDAR measurements.
The results obtained show an increase in precision and especially in the reactivity of the navigator. As shown in Figure 4.11, the initial error is corrected much faster than in the LOS-only test.

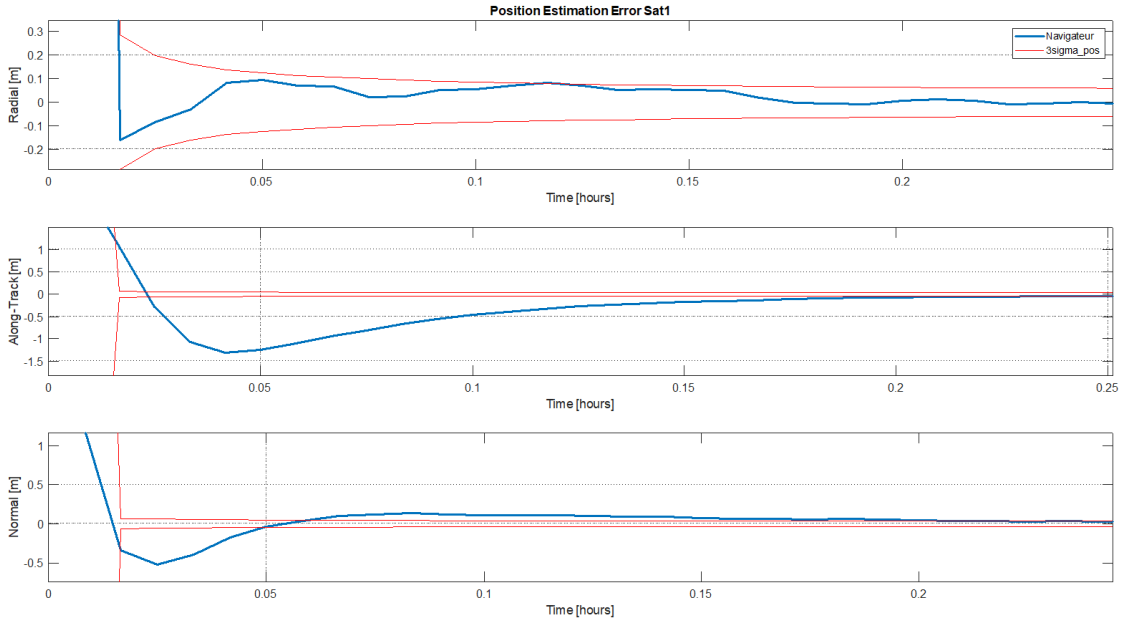


Figure 4.11: Correction of the initial error with LOS+LiDAR measurements

Results still show that the covariance is too narrow, and some estimates fall outside of the 3σ limit. The reason for this occurrence is an error in the modelling as explained previously. Nonetheless, the combined effect of LOS and LiDAR measurements defines a significantly more precise estimation around the asteroid.

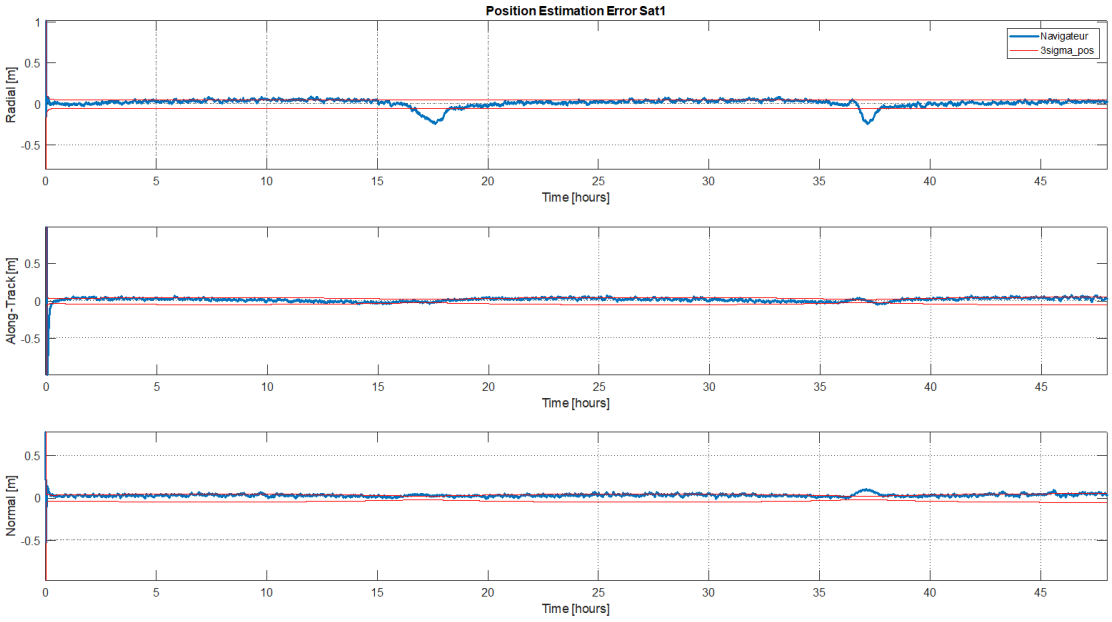


Figure 4.12: Estimation around Didymos with LiDAR measurements

The overall estimation mission shows extremely precise navigation, with sub-centimetre errors, in Table 4.7.

Direction	RMS Position	RMS Speed
Radial	0.046601	0.000216
Along-Track	0.031666	0.000171
Normal	0.040078	0.000310
3D	0.069143	0.000415

Table 4.7: Error in position and speed with LOS+LiDAR measurements

Final Comparison

In Table 4.8 the comparison shows the difference between the LOS-only and the LOS+LiDAR navigation.

Direction	RMS Position		RMS Speed	
	LOS	LOS + LiDAR	LOS	LOS + LiDAR
Radial	48.216298	0.046601	0.004028	0.000216
Along-Track	0.034537	0.031666	0.002389	0.000171
Normal	0.040505	0.040078	0.000310	0.000310
3D	48.216327	0.069143	0.004694	0.000415

Table 4.8: Comparison of RMS position and speed errors for LOS and LOS+LiDAR measurements.

4.3 Interplanetary tests

The MATLAB navigator has been developed to be adaptable to all kinds of interplanetary navigation, providing an estimate of the satellite's state even when its trajectory is not a simple orbit. Propagation in deep space is programmed to be just as precise as in orbits close to planets; however, the state correction via measurements requires measurements adapted to deep-space conditions.
The measurements developed in this internship, Line of Sight and LiDAR, can provide information on the satellite's state relative to other landmarks, but they strongly depend on the distance to those landmarks. In deep space, distant bodies tend to appear almost immobile relative to the spacecraft, so the measurements carry very little information and the estimation diverges.
In this section, the navigator will be tested on non-orbital trajectories, in order to analyse its response to a simplified manoeuvre model and to navigation around more distant bodies. In particular, two cases are considered: the HERA mission (the trajectory of the Juventas CubeSat) and an interplanetary mission towards another asteroid. The latter is designed from scratch.

4.3.1 Juventas mission: first week

The capabilities of the navigator are particularly well illustrated during a navigation test for the HERA mission, with the CubeSat Juventas. The trajectory of Juventas is atypical and manoeuvres play a crucial role. This test consists in evaluating LOS navigation during the first week of the mission, taking into account the effects of manoeuvres.

As mentioned before, the measurements by Line of Sight do not take into account the radial distance and are not capable of correcting range errors. Navigation by only LOS prevents the correct handling of manoeuvres during the simulation since the model is simplified. Distance measurements can correct those errors; without them, the estimation is expected to diverge. Moreover, the current model of acceleration by thrust is very simplified, which is expected to generate significant errors in the estimation

Test configuration

In this scenario the spacecraft moves in the vicinity of the Didymos system, which is taken as the central gravitating body. The dynamics are propagated in the inertial `ICRF` frame, using a spherical-harmonic representation of the gravity field of Didymos up to degree and order 20. The gravitational attraction of the Moon, the Earth and the Sun is included as third–body perturbations. Also Solar Radiation Pressure is taken into account.

Table 4.9: Initial orbital elements for the Didymos test case (single spacecraft).

Quantity	Value
Central body	Didymos
Perturbing bodies	Earth, Moon, Sun
Reference frame	ICRF
Epoch (CJD)	28139.56233
Epoch (calendar)	16 January 2027 13:29:45.657 (TAI)
Semi-major axis a	-2.263×10^3 m
Eccentricity e	12.320
Inclination i	93.49°
RAAN Ω	282.33°
Argument of periapsis ω	326.11°
True anomaly	341.16°
Duration	$7 days$
Integration step	30 s

The osculating Cartesian state at the reference epoch is

$$\mathbf{r}_0 = \begin{bmatrix} 4759.868642 & -15665.117650 & -21410.562487 \end{bmatrix}^T \text{m},$$

$$\mathbf{v}_0 = \begin{bmatrix} 0.010040 & -0.077149 & 0.109334 \end{bmatrix}^T \text{m/s}.$$

Stochastic modelling of the orbital state follows the same philosophy adopted in the other test cases.

Navigation by Line of Sight

As explained before, the Juventas mission heavily relies on manoeuvres in order to follow the correct trajectory. These errors are expected to be corrected by the Kalman Filter. However, in a LOS-only navigation the radial error will not be corrected.

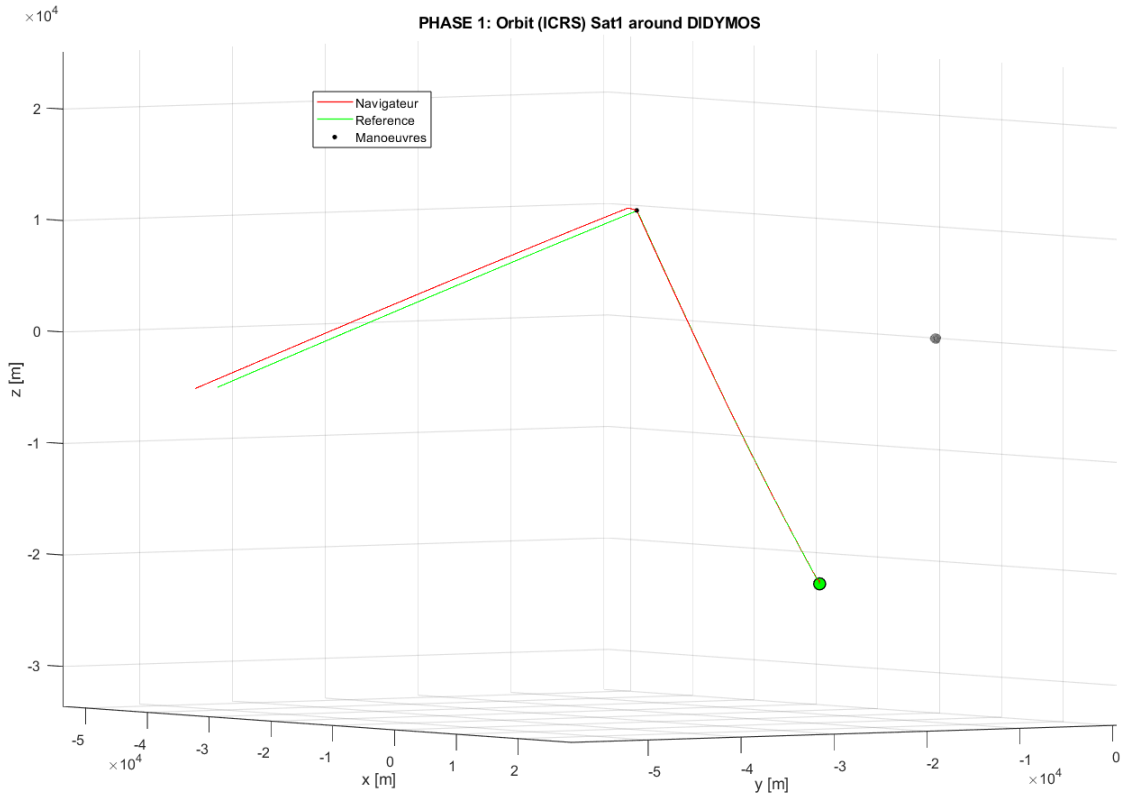


Figure 4.13: Trajectory during the first week of the HERA mission with Juventas.

Navigation is precise at the beginning of the trajectory; however, an error appears during the manoeuvre.

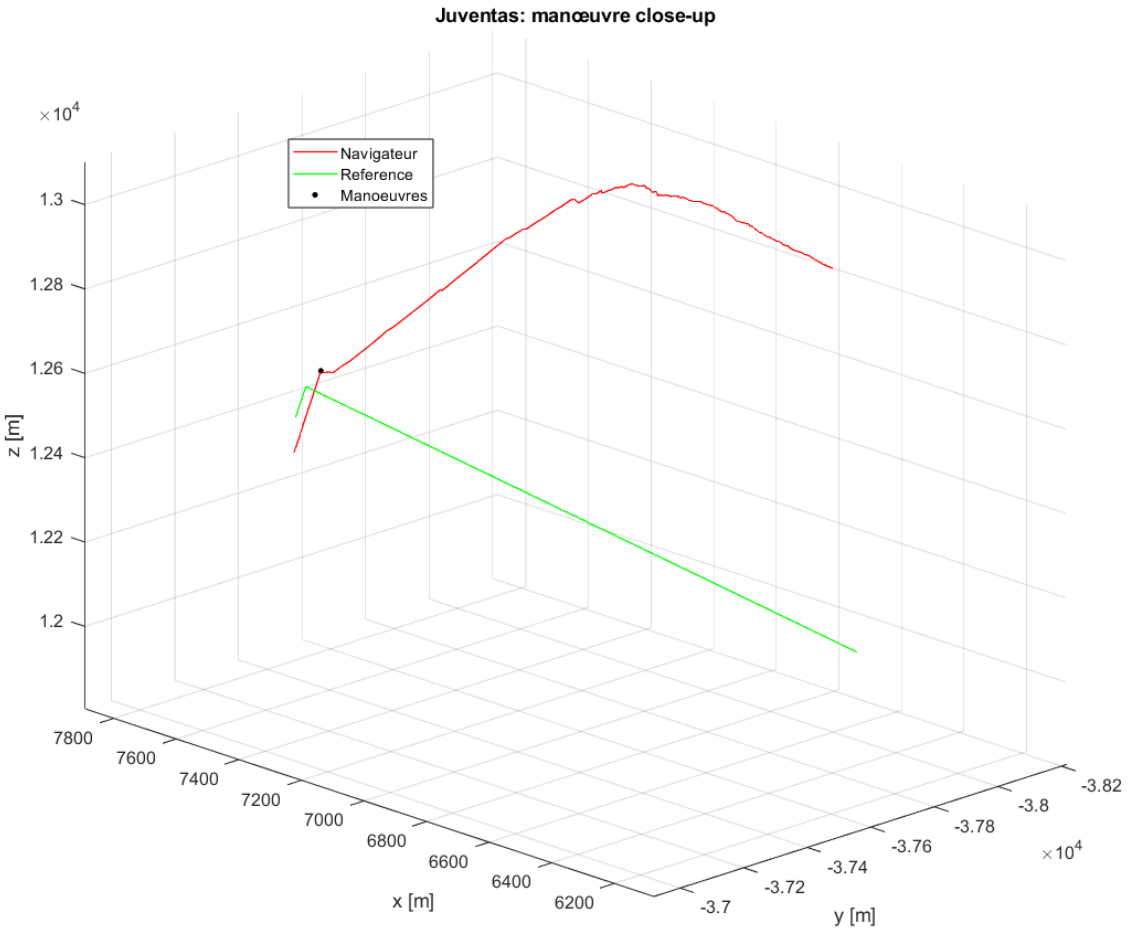


Figure 4.14: Zoom on the first manoeuvre of Juventas.

Although the manoeuvre is correctly applied, the navigator's manoeuvre model (which only uses mean acceleration in orbital window) leads to a divergence in the estimation. On the other hand, the LOS measurements allow for the effective correction of the tangential and normal components, while the radial error persists. This tendency is clearly visible in the estimation of the trajectory, presented in figure 4.15.

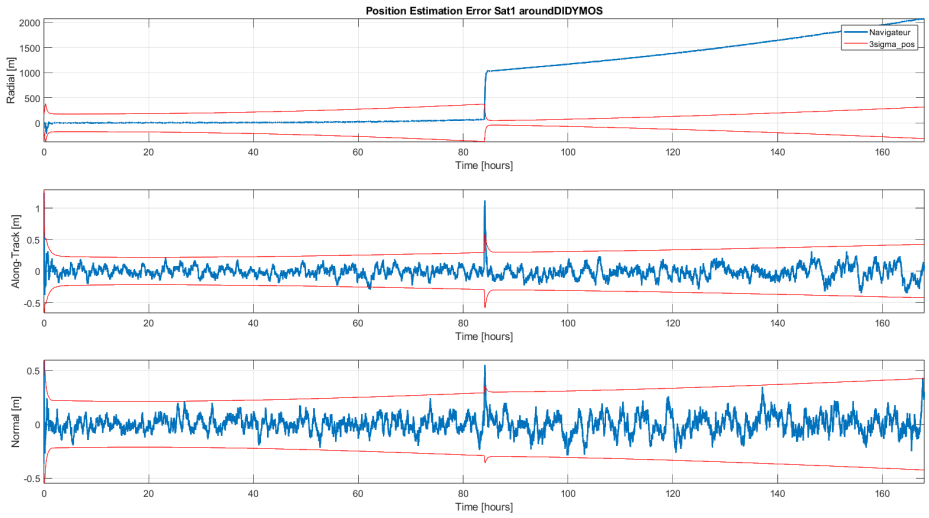


Figure 4.15: Estimation error of the Juventas trajectory.

The performances are listed in Table 4.10

Direction	RMS Position	RMS Speed
Radial	1515.966104	0.003899
Along-Track	0.116177	0.007326
Normal	0.097455	0.000074
3D	1515.966111	0.008299

Table 4.10: Error in position and speed with LOS measurements

A comparison with the navigation with both LOS and LiDAR will show that this problem is easily avoidable with a distance measurement.

Navigation by Line of Sight & LiDAR

A new test integrating LiDAR measurements will show the performance of the estimation during the first week of the HERA mission for Juventas.

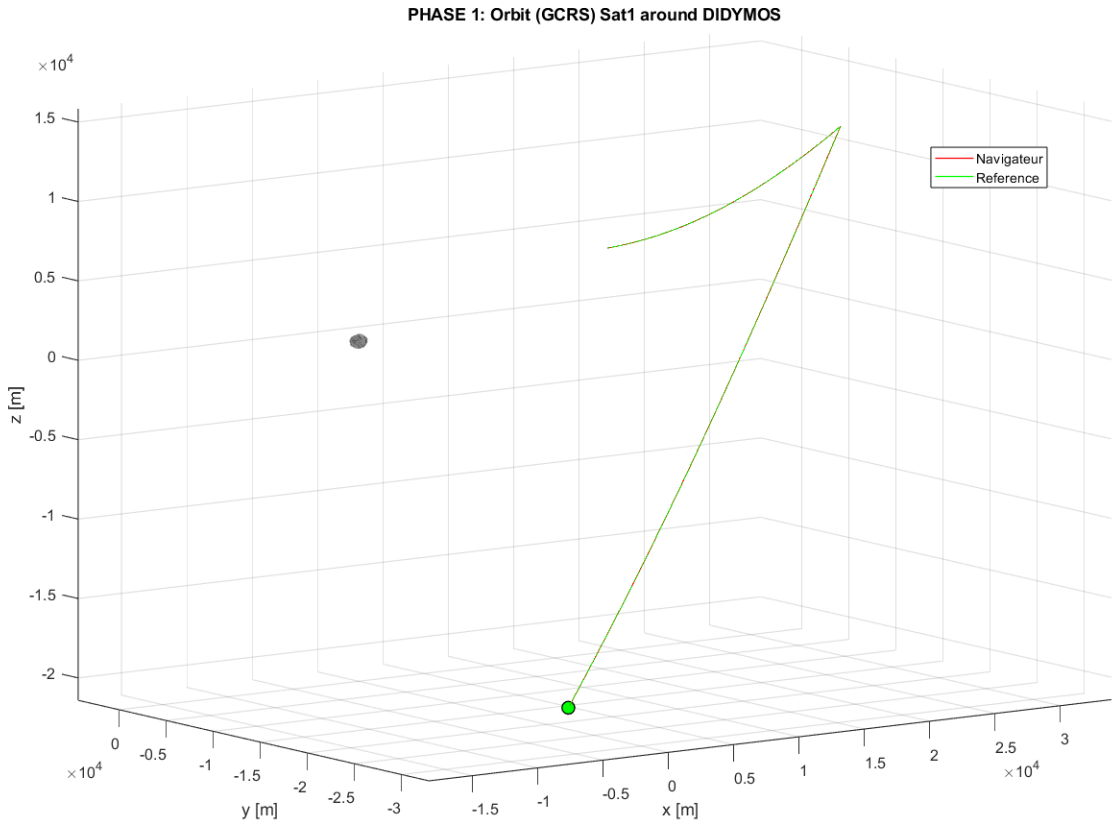


Figure 4.16: Juventas Orbit of the first 7 days of the mission, with LiDAR measurements

As shown in Figure 4.16, the navigator now follows the reference precisely, executing the manoeuvre with perfect precision. The simplified model for thrust acceleration indeed causes errors in the state, but the Kalman Filter is capable of using the information in the range measurement and the angular measurements to correct the state. This is clearly visible in the estimation plot, Figure 4.17

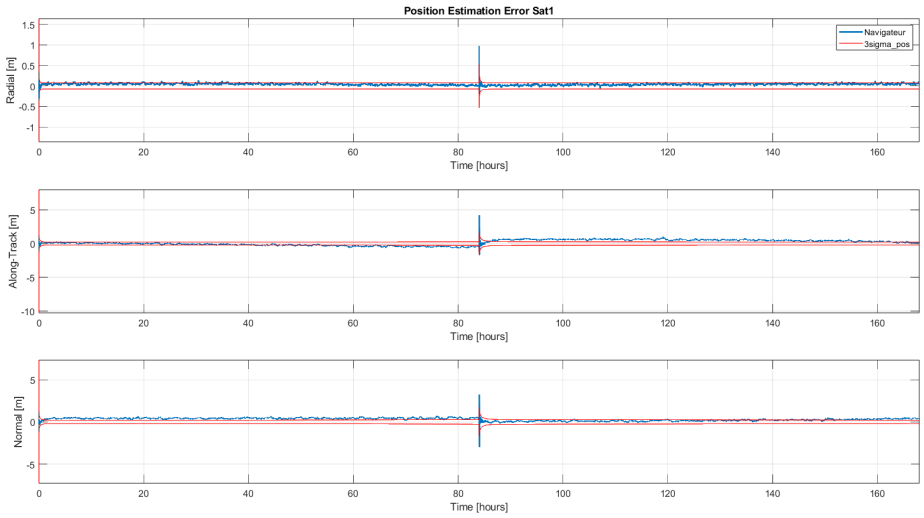


Figure 4.17: Position estimation during the first week of the Juventas trajectory, with LiDAR measurements

The estimation plot shows that measurement updates promptly correct the thrust error, keeping the overall state error at the centimetre level. However, the estimate occasionally falls just outside the predicted covariance bounds, suggesting residual dynamical mismodelling. Because the same pattern was observed in the Didymos orbital case (Sec. 4.2), the most plausible cause is a mismatch between the asteroid gravity model used by the navigator and the one used by the mission generator in Patrius (Sec. 4.1). Since both models rely on CNES internal data, harmonising them should be feasible and would likely remove this discrepancy.

Direction	**RMS Position**	**RMS Speed**
Radial	0.046096	0.000216
Along-Track	0.032475	0.000171
Normal	0.039401	0.000310
3D	0.068788	0.000415

Table 4.11: Error in position and speed with LOS+LiDAR measurements

Final Comparison

Direction	RMS Position		RMS Speed	
	LOS	**LOS + LiDAR**	**LOS**	**LOS + LiDAR**
Radial	1515.966104	0.046096	0.003899	0.000216
Along-Track	0.116177	0.032475	0.007326	0.000171
Normal	0.097455	0.039401	0.000074	0.000310
3D	1515.966111	0.068788	0.008299	0.000415

Table 4.12: Comparison of RMS position and speed errors for LOS and LOS+LiDAR measurements.

The final comparison of the two different navigation methods once again shows that LiDAR measurements are necessary for space navigation, especially with manoeuvres.

4.3.2 Interplanetary trajectory

For the HERA mission, internal CNES data were available for analysis; by contrast, real interplanetary trajectories are not available for testing. To validate the navigator in a realistic deep-space context and to study its behaviour over a complete interplanetary transfer, a full mission from Earth to an asteroid will be designed in Chapter 5 using an optimal-control-based trajectory optimisation.

Chapter 5

Guidance and Interplanetary Trajectory Optimisation

5.1 Introduction to Interplanetary Trajectory Optimisation

The work presented in the first part of this thesis concentrated on the development of an interplanetary navigator. The navigator, implemented in a high-level environment for ease of analysis and modularity, brings together detailed dynamical models and measurement models into an Extended Kalman Filter architecture. Despite significant progress, only a limited amount of real mission data were available for validation. In particular, the absence of long-range interplanetary trajectories in which line-of-sight (LOS) and LiDAR measurements are combined meant that the navigation performance could not be assessed over the entire cruise. To overcome this limitation and to test the navigator in a realistic setting, a synthetic mission was created. Rather than waiting for flight data to become available, the thesis also focused on the development of a representative interplanetary transfer: a low-thrust spiral from Earth's vicinity to a small asteroid.
This second part of the work thus serves a dual purpose. Firstly, it generates a physically consistent reference trajectory, which will later be used to simulate measurements and evaluate the navigation filter. Secondly, it sheds light on the theory and practice of low-thrust trajectory optimisation, to provide an overview of both Guidance and Navigation aspects of an interplanetary space mission. The optimisation problem is framed as the minimisation of propellant consumption subject to the dynamics of a spacecraft equipped with a continuous low-thrust propulsion system.

5.2 Optimisation of a Space Trajectory

Trajectory optimisation can be performed using either direct methods, in which the control history is discretised and a large nonlinear programming problem is solved, or indirect methods that derive necessary conditions for optimality by means of the calculus of variations. Indirect methods are characterised by high numerical precision and compact parameterisation but often suffer from convergence difficulties. In this work, we adopt an indirect approach because it provides a rigorous framework for deriving the control law and reduces the number of unknown parameters to a manageable set.
In general, an indirect method is defined by:

- A **state vector** x, which collects the mission quantities to be tracked
- A **control vector** u, used to explore candidate solutions and find the optimal one
- An **independent variable**, usually the time t
- **Differential equations** $\dot{x} = f(x, u, t)$ describing the evolution of the state as a function of time and control
- **Boundary conditions** $\chi(x(t_0), x(t_f), t_0, t_f) = 0$, usually set by mission requirements
- **Functional** $J = \varphi(x(t_0), x(t_f), t_0, t_f) + \int_{t_o}^{t_f} \phi(x(t), u(t), t)dt$ (we will use the Mayer form, where $\phi = 0$)

After defining all parameters of the optimal control problem, the necessary conditions for the optimal solution are two: the functional J must be stationary and the boundary conditions must be satisfied.

5.2.1 Optimal Control Theory

The optimisation of an interplanetary trajectory typically begins by partitioning the trajectory into multiple arcs over which the variables are continuous and the control law is the same. Continuity across arcs is enforced by boundary conditions at the arc endpoints, where each variable assumes the value $x_{(j-1)_+}$ and x_{j_-}, where '+' and '-' symbolize the values to the left or the right of point j. This helps take into account discontinuities, such as flybys or impulsive manoeuvres. The functional J is then augmented to take into account the costate variables λ and the adjoint constants μ.

$$J^* = \varphi + \mu^{\mathrm{T}}\chi + \sum_{j=1}^{n} \int_{t_{j-1}^+}^{t_j^-} \lambda^{\mathrm{T}} \left(f - \dot{x}\right) \mathrm{d}t$$

The Hamiltonian of the system is defined as $H = \lambda^T f$ (in Mayer form we consider $\phi = 0$).

The values for J and J^* depend on the values at the boundaries for time t, state x, time derivatives of the state $\dot{x}$ and control u.
Each arbitrary variation $\delta x, \delta u, \delta x_{(j-1)_+}, \delta x_{j_-}, \delta t_{(j-1)_+}, \delta t_{j_-}$ causes

$$\begin{aligned}\delta J^* = &\left(- H_{(j-1)_+} + \frac{\partial \varphi}{\partial t_{(j-1)_+}} + \mu^{\mathrm{T}} \frac{\partial \chi}{\partial t_{(j-1)_+}} \right) \delta t_{(j-1)_+} + \left(H_{j_-} + \frac{\partial \varphi}{\partial t_{j_-}} + \mu^{\mathrm{T}} \frac{\partial \chi}{\partial t_{j_-}} \right) \delta t_{j_-} \\ &+ \left(\lambda^{\mathrm{T}}_{(j-1)_+} + \frac{\partial \varphi}{\partial x_{(j-1)_+}} + \mu^{\mathrm{T}} \left[\frac{\partial \chi}{\partial x_{(j-1)_+}} \right] \right) \delta x_{(j-1)_+} + \left(- \lambda^{\mathrm{T}}_{j} + \frac{\partial \varphi}{\partial x_{j_-}} + \mu^{\mathrm{T}} \left[\frac{\partial \chi}{\partial x_{j_-}} \right] \right) \delta x^-_j \\ &+ \sum_j \int_{t_{(j-1)_+}}^{t_{j-}} \left[\left(\frac{\partial H}{\partial x} + \dot{\lambda}^{\mathrm{T}} \right) \delta x + \frac{\partial H}{\partial u} \, \delta u \right] dt, \qquad j = 1, \ldots, n\end{aligned}$$

In order to find a maximum or minimum, the functional J has to be stationary, which means the first variation δJ must be equal to zero for any admissible $\delta x, \delta u, \delta x_{(j-1)_+}, \delta x_{j_-}, \delta t_{(j-1)_+}, \delta t_{j_+}$. Adjoint variables and multipliers (adjoint constants) are introduced to nullify the coefficients of each variation.

Adjoint (costate) dynamics — Euler–Lagrange form:

$$\frac{d\lambda}{dt} = - \left(\frac{\partial H}{\partial x} \right)^{\mathrm{T}} \tag{5.1}$$

Control optimality condition

$$\left(\frac{\partial H}{\partial u} \right)^{\mathrm{T}} = 0 \tag{5.2}$$

With this formulation, the optimal solution also depends on bounds on the control, which is independent of the functional. When control is bounded, such as when the thrust magnitude is bounded as $0 \leq T \leq T_{\max}$, the optimal control lies on the boundary of the admissible set.

Optimal boundary conditions

$$-\lambda^{-\mathrm{T}}_j + \frac{\partial \varphi}{\partial x^-_j} + \mu^{\mathrm{T}} \left[\frac{\partial \chi}{\partial x^-_j} \right] = 0, \qquad j = 1, \ldots, n \tag{5.3}$$

$$\lambda^{+\mathrm{T}}_j + \frac{\partial \varphi}{\partial x^+_j} + \mu^{\mathrm{T}} \left[\frac{\partial \chi}{\partial x^+_j} \right] = 0, \qquad j = 0, \ldots, n-1 \tag{5.4}$$

$$H^-_j + \frac{\partial \varphi}{\partial t^-_j} + \mu^{\mathrm{T}} \frac{\partial \chi}{\partial t^-_j} = 0, \qquad j = 1, \ldots, n \tag{5.5}$$

$$-H^+_j + \frac{\partial \varphi}{\partial t^+_j} + \mu^{\mathrm{T}} \frac{\partial \chi}{\partial t^+_j} = 0, \qquad j = 0, \ldots, n-1 \tag{5.6}$$

where j_- and j_+ denote values immediately before and after point j, respectively. Without μ^T:, the optimal boundary conditions are defined as

$$\sigma(x_{(j-1)_+}, x_{j_-}, \lambda_{(j-1)_+}, \lambda_{j_-}, t_{(j-1)_+}, t_{j_-},) \tag{5.7}$$

which are different from the boundary conditions described in 5.2

5.2.2 Boundary Value Problem

Optimal control theory often leads to a system of boundary differential equations in which the initial values of the variables are not known a priori. The solution is obtained by integrating the variational system so as to satisfy all the boundary conditions χ and the optimal boundary conditions σ.
The main complication of indirect methods is often the solution of the resulting boundary-value problem (BVP), which, in general,

- divides the integration domain into subintervals where the differential equations may change,
- has subinterval durations that are unknown,
- involves boundary conditions that can be nonlinear,
- may include variables that are discontinuous across internal boundaries.

The uncertainty on the duration is handled through the stretched variable ϵ

$$\epsilon = j - 1 + \frac{t - t_{j-1}}{\tau_j} \tag{5.8}$$

where τ_j is the duration of the j-th interval.
The problem is then reformulated without explicitly distinguishing between state variables x and adjoint variables λ; it becomes a differential problem in $y = (x, \lambda)$:

$$\frac{dy}{dt} = f^*(z, \epsilon) \tag{5.9}$$

Since the problem now depends on boundary parameters, which may be variables or constants, it is finally written in terms of the vector $z = (y, c)$, where c is a vector of constant parameters. Thus:

$$\frac{dz}{d\epsilon} = f(z, \epsilon) \tag{5.10}$$

Inside z we have

$$\frac{dy}{d\epsilon} = \tau_j \frac{dy}{dt}, \qquad \frac{dc}{d\epsilon} = 0 \tag{5.11}$$

Boundary conditions are defined as

$$\Psi(s) = 0 \tag{5.12}$$

where s is the vector of boundary values $s = (y_0, y_1, \ldots, y_n, c)$. None of the boundary values is assumed to be known, and the solution is obtained through an iterative procedure in which each iteration starts from the values computed in the previous one. This means that the BVP requires an initial guess p^1, and the r^{th} iteration starts from $z(0) = p^r$. A

variation on the initial guess Δp causes a variation on the boundary conditions $\Delta\Psi$ defined as

$$\Delta\Psi = \left[\frac{\partial\Psi}{\partial p}\right]\Delta p \tag{5.13}$$

The objective is to have $\Psi = 0$, so each iteration the variation has to be $\Delta\Psi = -\Psi^{r-1}$. This means that at each iteration, the initial values have to be corrected with

$$\Delta p = p^{r+1} - p^r = -\left[\frac{\partial\Psi}{\partial p}\right]^{-1}\Psi^r \tag{5.14}$$

where

$$\left[\frac{\partial\Psi}{\partial p}\right] = \left[\frac{\partial\Psi}{\partial s}\right]\left[\frac{\partial s}{\partial p}\right] \tag{5.15}$$

The first one can be immediately acquired by deriving the boundary conditions with respect to their parameters; the second one depends on the boundary values assumed by $\left[\frac{\partial z}{\partial p}\right]$ in all the intervals $\epsilon = 0, 1, ..., n$. It can be calculated by integrating the homogeneous system

$$\left[\dot{\frac{\partial z}{\partial p}}\right] = \left[\frac{\partial g}{\partial z}\right]\left[\frac{\partial z}{\partial p}\right] \tag{5.16}$$

where

$$\left[\frac{\partial z}{\partial p}\right] = [g(\epsilon)] \tag{5.17}$$

and

$$[\dot{g}] = \frac{d}{d\varepsilon}\left[\frac{\partial z}{\partial p}\right] = \left[\frac{\partial}{\partial p}\left(\frac{dz}{d\varepsilon}\right)\right] = \left[\frac{\partial f}{\partial p}\right] = \left[\frac{\partial f}{\partial z}\right][g] \tag{5.18}$$

This formulation also helps finding initial values of the system:

$$[g(0)] = \left[\frac{\partial z(0)}{\partial p}\right] = [I] \tag{5.19}$$

Finally, the Boundary Value Problem only depends on two elements, $\left[\frac{\partial\Psi}{\partial s}\right]$ and $\left[\frac{\partial s}{\partial p}\right]$. Also, the $\left[\frac{\partial\Psi}{\partial s}\right]$ matrix can be derived numerically by analysing the response $\Delta\Psi(\Delta p_i)$ of the system to small variation Δp_i of parameters p_i.

5.3 Optimization of a space trajectory: flyby near asteroid 2000 SG_{344}

5.3.1 The mission

The asteroid 2000 SG_{344} is a small near-Earth asteroid first observed in 2000. It is assumed to have a mass of about 7.1×10^7 kg and a diameter of about 37 m. Its importance stems from the fact that it has been cited as one of the near-Earth objects with a relatively high listed probability of impacting Earth within the next century, with estimates around 1 in 360 between 2069 and 2121 [6]. The Italian Space Agency has expressed strong interest in this asteroid, and for this reason, Professor Lorenzo Casalino developed a trajectory optimizer in Fortran, which I will use to find the optimal flyby trajectory and to test the Interplanetary Navigator.

5.3.2 Setup of the variational problem

The trajectory is heliocentric and subject to a low-thrust acceleration vector. As a low-thrust mission, the expected duration is on the order of one to two years, modelled here with a single arc (although the method can operate with multi-arc trajectories). We adopt an inertial reference frame aligned with the ecliptic and describe the spacecraft state in spherical-polar coordinates (r, ϑ, φ) with associated velocities (u, v, w) measured in a local frame.
The objective function to be minimised is the propellant mass consumed during the manoeuvres, or equivalently to maximise the final mass m_f.

$$\min\ J_{\text{Mayer}} \ = \ -m(t_f) \quad \text{subject to} \quad \dot{m} = -\frac{T}{I_{sp} g_0}.$$

The state vector

The state vector comprises 14 variables:

$$y = [r, \vartheta, \varphi, u, v, w, m, \lambda_r, \lambda_\vartheta, \lambda_\varphi, \lambda_u, \lambda_v, \lambda_w, \lambda_m]^T \tag{5.21}$$

As explained in Sec. 5.2.1, the λ variables are the adjoint (costate) variables of the optimal control problem.

The differential equations that describe the evolution of these **state variables** are:

$$\dot{r} = u, \tag{5.22}$$

$$\dot{\vartheta} = \frac{v}{r\cos\varphi}, \tag{5.23}$$

$$\dot{\varphi} = \frac{w}{r}, \tag{5.24}$$

$$\dot{u} = -\frac{1}{r^2} + \frac{v^2}{r} + \frac{w^2}{r} + \frac{T}{m}\sin\gamma_T + a_r^{(Earth)}, \tag{5.25}$$

$$\dot{v} = -\frac{uv}{r} + \frac{vw}{r} + \frac{T}{m}\cos\gamma_T\cos\psi_T + a_\vartheta^{(Earth)}, \tag{5.26}$$

$$\dot{w} = -\frac{uw}{r} - \frac{v^2\tan\varphi}{r} + \frac{T}{m}\cos\gamma_T\sin\psi_T + a_\varphi^{(Earth)}, \tag{5.27}$$

$$\dot{m} = -\frac{T}{I_{sp}g_0} \tag{5.28}$$

where γ_T and ψ_T are the corresponding angles for the thrust T and the terms $a_i^{(Earth)}(t, r, \vartheta, \varphi)$ are the perturbations caused by Earth's gravity.
As for the evolution of the **co-state variables**, their differential equations follow from the Euler–Lagrange equations:

$$\dot{\lambda}_r = \frac{1}{r^2}\left[\lambda_\vartheta\frac{v}{\cos\phi} + \lambda_\phi w + \lambda_u\left(-\frac{2}{r} + v^2 + w^2\right) + \right. \tag{5.29}$$

$$\left. + \lambda_v\left(-uv + vw\tan\phi\right) + \lambda_w\left(-uw - v^2\tan\phi\right)\right] + \mu_r^{Earth} \tag{5.30}$$

$$\dot{\lambda}_\vartheta = 0 + \mu_\vartheta^{Earth} \tag{5.31}$$

$$\dot{\lambda}_\phi = \frac{1}{r\cos^2\phi}\left(-\lambda_\vartheta v\sin\phi - \lambda_v vw + \lambda_w v^2\right) + \mu_\varphi^{Earth} \tag{5.32}$$

$$\dot{\lambda}_u = \frac{1}{r}\left(-\lambda_r r + \lambda_v v + \lambda_w w\right) \tag{5.33}$$

$$\dot{\lambda}_v = \frac{1}{r}\left[-\lambda_\vartheta\frac{1}{\cos\phi} - 2\lambda_u v + \lambda_v\left(u - w\tan\phi\right) + 2\lambda_w v\tan\phi\right] \tag{5.34}$$

$$\dot{\lambda}_w = \frac{1}{r}\left(-\lambda_\phi - 2\lambda_u w - \lambda_v v\tan\phi + \lambda_w u\right) \tag{5.35}$$

$$\dot{\lambda}_m = \frac{T}{m^2}\lambda_V \tag{5.36}$$

where the terms $\mu_i^{Earth}(t, r, \vartheta, \varphi, u, v, w, \lambda)$ are the effect of Earth's gravitational perturbations on the co-state variables.
The adjoint variables quantify the sensitivity with respect to each state component; they will be used in Sec. 5.3.2.

The control vector

For a space trajectory, the control is defined by the spacecraft's thrust magnitude T and its direction, given by the angles γ_T and ψ_T.

$$u = (T, \gamma_T, \psi_T)^T \tag{5.37}$$

To compute the optimal thrust direction, optimal control theory uses the adjoint variables described in Sec. 5.3.2. These variables allow the optimal thrust direction to be computed at each step of the optimization and to depend on the state variables as displayed here.

$$\sin\gamma_T = \frac{\lambda_u}{\lambda_V} \tag{5.38}$$

$$\cos\gamma_T \cos\psi_T = \frac{\lambda_v}{\lambda_V} \tag{5.39}$$

$$\cos\gamma_T \sin\psi_T = \frac{\lambda_w}{\lambda_V} \tag{5.40}$$

where

$$\lambda_V = \sqrt{\lambda_u^2 + \lambda_v^2 + \lambda_w^2} \tag{5.41}$$

is the magnitude of the primer vector, which is parallel to the optimal thrust direction.
To compute the optimal thrust magnitude, the switching function is used:

$$S_F = \frac{\lambda_V}{m} - \frac{\lambda_m}{c} \tag{5.42}$$

At each step, if $S_F > 0$ the thrust is set to its maximum value $T_{\max}$; if $S_F < 0$ the thrust is set to (near) zero. This is consistent with a *bang-bang* control, where thrust is either on or off. The effective exhaust speed c is considered constant during this trajectory.

The optimal solution

The optimal solution is subject to prescribed boundary conditions on the initial state $(r_0, \vartheta_0, \varphi_0, u_0, v_0, w_0)$ and on the final state at arrival $(r_f, \vartheta_f, \varphi_f, u_f, v_f, w_f)$. In the Mayer formulation, the performance index is $J = -m_f$.
Following the calculus of costate variables (Sec. 5.3.2) $(\lambda_r, \lambda_\vartheta, \lambda_\varphi, \lambda_u, \lambda_v, \lambda_w, \lambda_m)$, the Hamiltonian is

$$\begin{aligned} H = {} & \lambda_r u + \lambda_\vartheta \frac{v}{r\cos\varphi} + \lambda_\varphi \frac{w}{r} + \lambda_u\Big(-\frac{1}{r^2} + \frac{v^2+w^2}{r}\Big) + \\ & + \lambda_v\Big(-\frac{uv}{r} + \frac{vw}{r}\Big) + \lambda_w\Big(-\frac{uw}{r} - \frac{v^2\tan\varphi}{r}\Big) + \lambda_m\Big(-\frac{T}{I_{sp}g_0}\Big) + \frac{T}{m}\,\boldsymbol{p}\cdot\boldsymbol{u}_T + \\ & + \lambda_u a_r^{(Earth)} + \lambda_v a_\vartheta^{(Earth)} + \lambda_w a_\varphi^{(Earth)} \end{aligned} \tag{5.43}$$

where $\boldsymbol{p} = (\lambda_u, \lambda_v, \lambda_w)$ is the primer vector and $\boldsymbol{u}_T$ is the unit vector in the thrust direction.

The optimality conditions require that the control maximise H at each time. When the thrust level is fixed, this reduces to steering the thrust along the primer direction; the switching function determines whether to thrust or coast based on the sign of λ_m and the magnitude of $\boldsymbol{p}$.
Since this trajectory is modelled with a single arc, the only boundary conditions are the initial and final states.

5.4 Solution Method and Algorithmic Structure

The algorithm is implemented in a *Fortran* software, which uses the BVNGL [7] method to solve the variational problem. The structure of the code can be summarised as follows:

1. **Initial guess:** Choose an initial guess for the unknown parameters, including the initial values of the costate variables. Suitable guesses can be obtained from simplified analytic solutions or by scaling shorter-duration transfers.

2. **Integration:** Propagate the state and costate differential equations forward in time along each arc, applying the optimality condition to compute the thrust direction. During coast arcs the thrust is minimal; during thrust arcs it is aligned with the primer vector λ_V.

3. **Matching conditions:** At the end of each arc, enforce continuity of the state variables and apply the matching conditions for the costates. If intermediate events such as flybys or imposed constraints occur, additional algebraic conditions must be satisfied.

4. **Residual evaluation:** Assemble a residual vector containing the differences between the actual terminal values and the desired boundary conditions. These residuals quantify how far the current guess is from satisfying the two-point boundary-value problem.

5. **Newton iteration:** Solve a linear system for the corrections to the unknown parameters using the Jacobian of the residual vector. Update the parameters and repeat the integration until the residuals fall below prescribed tolerances.

6. **Convergence check:** Monitor convergence by tracking the norm of the residuals and the variation of the parameters. If convergence stalls, adjust the initial guess or refine the segmentation of the trajectory (e.g., by adding or removing thrust arcs) and repeat the procedure.

5.5 Results of the Optimised Trajectory

The optimisation carried out for the mission departing from Earth and performing a flyby of asteroid 2000 SG_{344} used an initial spacecraft mass of approximately 21 kg, a specific impulse of 2100 s, and a maximum thrust of a few tenths of a newton. The optimiser allowed the departure date and flight duration to vary freely so as to minimise propellant consumption, subject to a minimum-distance constraint at the asteroid.

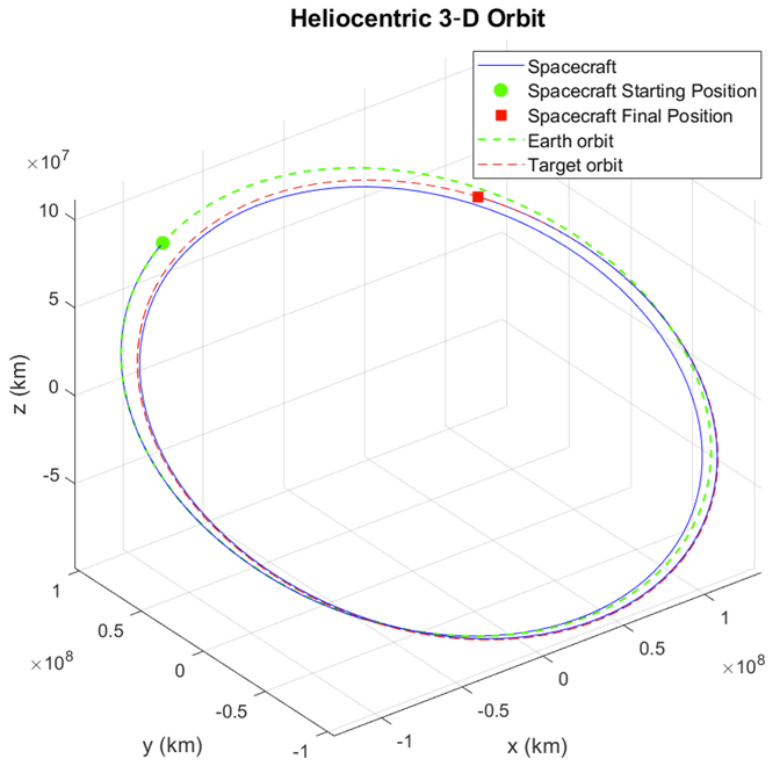


Figure 5.1: Heliocentric view of the optimised trajectory from Earth to the asteroid. The low-thrust spiral gradually intercepts the asteroid at the target epoch. The final portion corresponds to a close flyby used for navigation testing.

Figure 5.1 illustrates the three-dimensional trajectory in the heliocentric ICRF. The spacecraft slowly spirals away from Earth's orbit, gradually reducing its radial velocity relative to the Sun. The low thrust level implies that the spacecraft performs multiple revolutions around the Sun before reaching the asteroid. The optimised mission duration is about 640 days (approximately 1.75 years). Over this time the spacecraft mass decreases by only about 0.8 kg, highlighting the efficiency of electric propulsion (5.3), continuously approaching the asteroid.

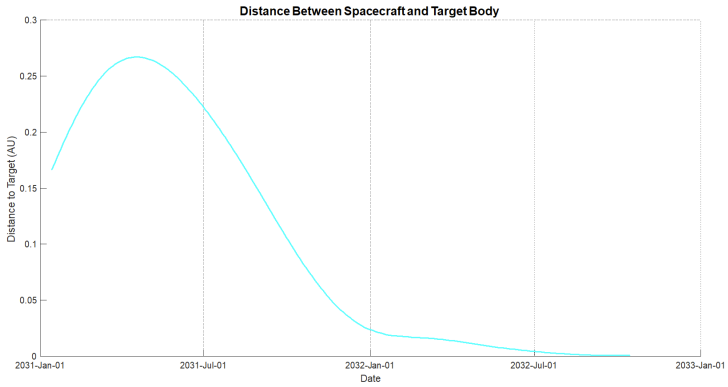


Figure 5.2: Distance from the asteroid during the mission

Figure 5.3: Mass consumption during the mission and its effect on the spacecraft's heliocentric speed

The thrust vector exhibits big and instantaneous variations; its magnitude remains below 1 mN and the peaks of thrust coincide precisely with the asteroid approach (see Fig. 5.2). The thrust history shows that the optimiser applies the bang-bang model of thrust throughout most of the cruise, with thrust components that jump from 0 to around 1 mN.

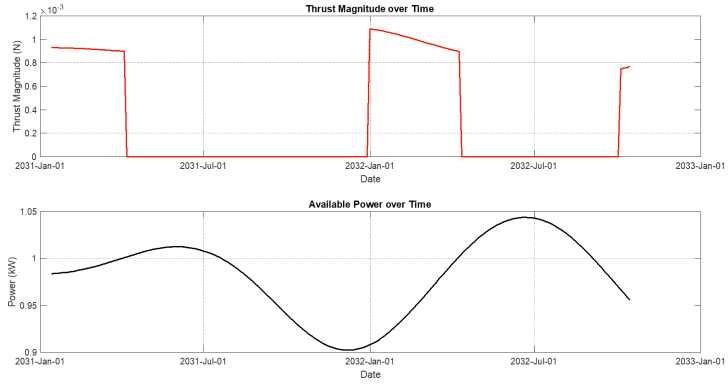


Figure 5.4: Thrust history and corresponding power

		Departure	Arrival
Epoch	[TDB]	13 Jan 2031, 19:49	14 Oct 2032, 06:46
Position	(x, y, z) [10^6 km]	(−60.2, 134.3, −0.0087)	(107.2, 94.7, −0.172)
Velocity	(v_x, v_y, v_z) [km s^{-1}]	(−27.7, −12.3, 0.0004)	(−21.8, 21.9, −0.004)
Mass	[kg]	21	20.22

Table 5.1: Departure and arrival conditions in the heliocentric ecliptic frame.

5.6 Navigation Test on the Optimised Trajectory

Tackling the guidance design of an interplanetary mission posed new challenges in the astrodynamic aspects of the mission and in the optimisation techniques. The newly optimised trajectory will be used to validate the MATLAB Interplanetary Navigator, allowing control and insight into both the Guidance and the Navigation components of GNC. The trajectory is used to generate measurement data (as explained in Sec. 4.1), and the navigator is tested in this environment.
As indicated in the first part of the thesis, the current navigator uses Line of Sight (LOS) and range (LiDAR) measurements. LOS measurements provide accurate angular information but limited range accuracy, while LiDAR measurements carry range information and are intended to compensate LOS limitations. However, these two measurement types are typically most precise only in the vicinity of space objects and targets: LOS requires identifiable targets to image, and LiDAR requires a surface for the signal to reflect from. Such targets exist in deep space, but they provide little information about the spacecraft's absolute position when they are very far away: the bearing to a distant body varies slowly relative to the spacecraft, and the range to a far celestial body changes only weakly along interplanetary trajectories. Interplanetary navigation is usually supported by ground-antenna radiometric measurements, which the current navigator does not implement.
What do we expect, then? The whole trajectory will be tested; results will likely show increasing uncertainty as the spacecraft recedes from Earth, and improved accuracy during the final approach to asteroid 2000 SG_{344}.

5.6.1 Departure from Earth

The first-phase navigation test, at departure from Earth, illustrates the limitations of the MATLAB navigator.
The spacecraft begins its journey far from suitable nearby optical targets; the best choice is the Moon. We verified that along the trajectory the Moon is visible and well illuminated by the Sun, making it a good LOS target. As for range measurements, their precision is jeopardised by the large distance to the Moon; therefore, range measurements will target the Moon but the results will show the combination of Line of Sight and LiDAR is not sufficient for navigation when far away from targets.

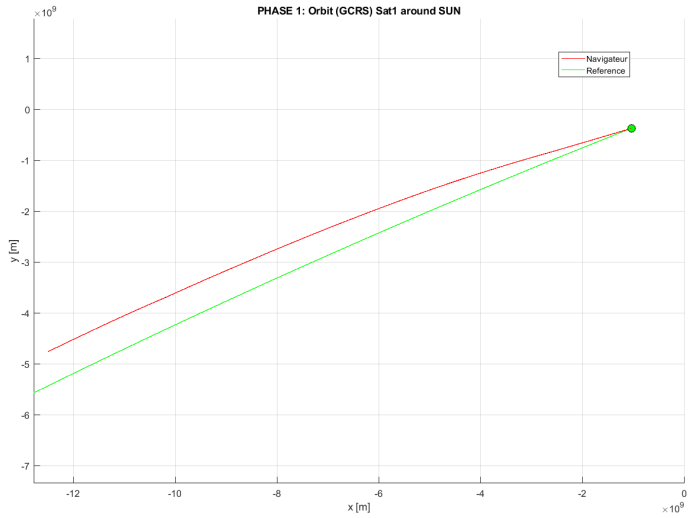


Figure 5.5: Trajectory of the first five days of the mission towards 2000 SG_{344}

As the trajectory shows, the navigator has not sufficient information for navigation. The plot shows a slow and steady divergence of the navigator's estimation, not any errors, which indicates the absence of information regarding the position and the subsequent slow divergence of the estimate. This is clear also looking at the estimation plot.

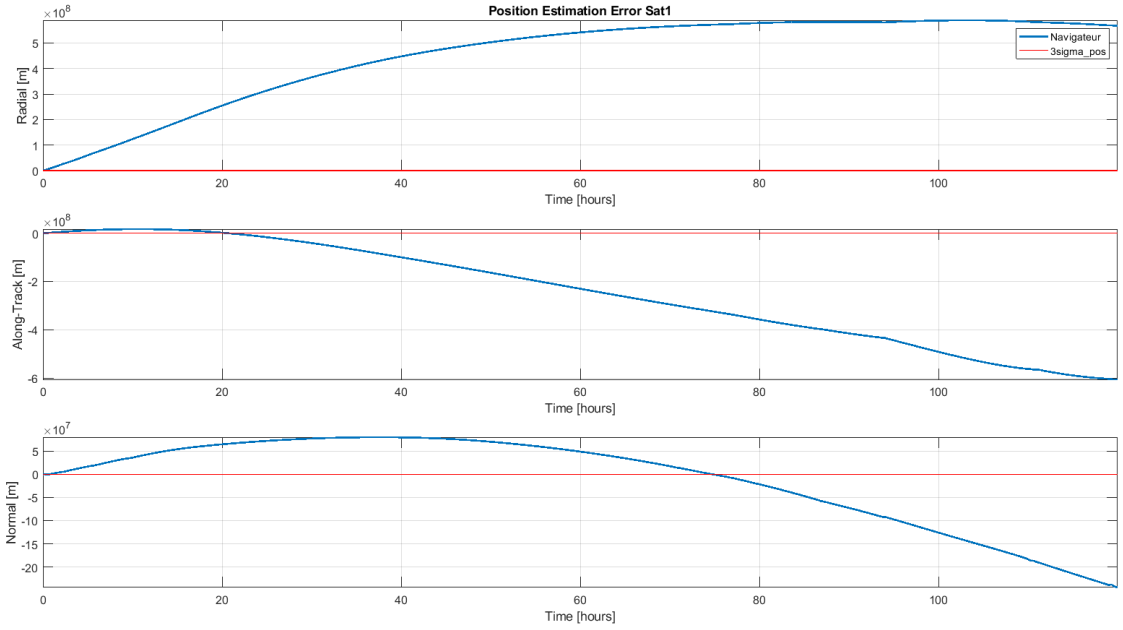


Figure 5.6: Estimation during the first 100 hours of the mission towards 2000 SG_{344}

The estimation diverges quickly: even though the measurements are correctly taken and correspond to the true trajectory, the resulting estimate carries no reliable information about the spacecraft state. With no information, the estimation slowly diverges from the real trajectory. The navigator is not capable of deep-space navigation using only these

measurements, and the result is far from the real one.

Direction	RMS Position	RMS Speed
Radial	575719701.91	1000.97
Along-Track	438704180.42	2642.56
Normal	118230985.41	1463.05
3D	733411957.22	3182.07

Table 5.2: Error in position and speed during the departure phase of the mission towards 2000 SG_{344}

Testing the navigator on other parts of this trajectory would add little value. The MATLAB navigator has been designed for rendezvous, flybys, and close approaches to celestial bodies. To achieve better navigation in deep space it would be necessary to implement antenna-based measurements capable of estimating the spacecraft state far from other bodies; however, CNES is not prioritising this aspect at the moment.

5.6.2 Final approach to 2000 SG_{344}

In this section we test the navigator's performance in its best operational scenario: in the vicinity of a small body. Here, LOS and range measurements exploit most of the available information. The spacecraft performs a close flyby of the asteroid (without significant dynamical perturbation), and the navigator points to the asteroid to take Line of Sight and LiDAR measurements.

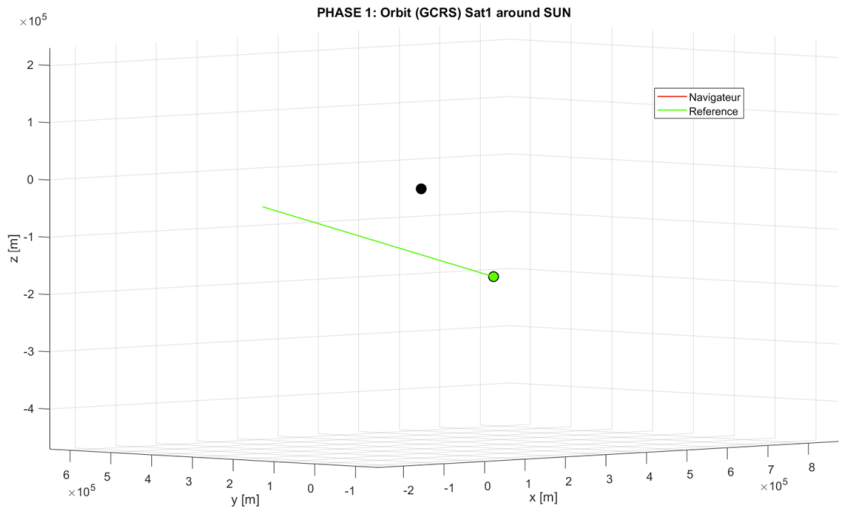


Figure 5.7: Flyby of 2000 SG_{344}

In this test, the last two days of the approach are simulated, with an LOS measurement

every 30 s and a LiDAR measurement every 1 s. The spacecraft approaches the asteroid to a minimum distance of a few km.

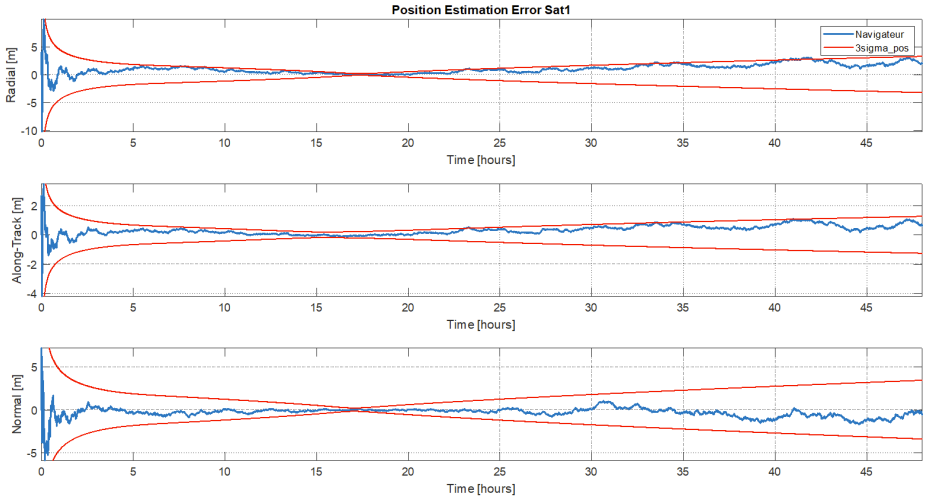


Figure 5.8: Position estimation during the flyby. The closest approach is around 11 hours from the beginning.

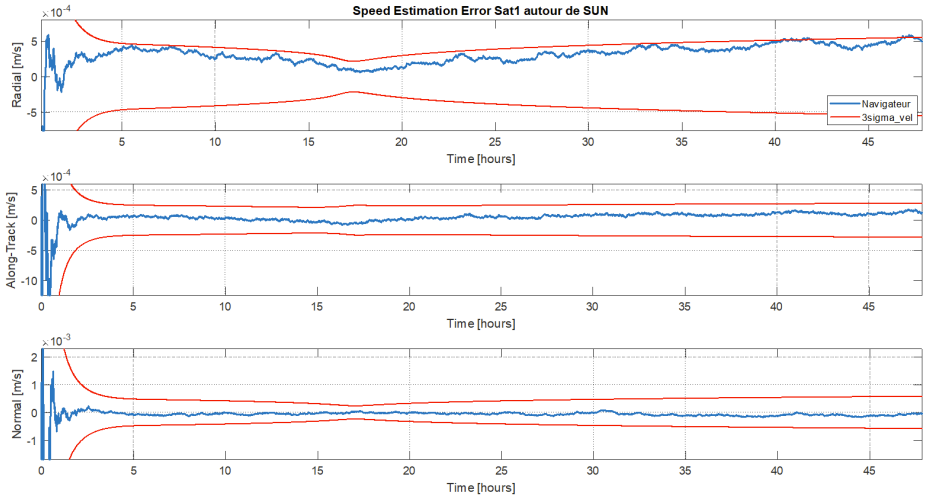


Figure 5.9: Speed estimation during the flyby. The closest approach is around 11 hours from the beginning.

The estimation plots show that the solution accuracy improves as the spacecraft approaches the asteroid and then slowly degrades as it recedes. The 3σ bounds shrink during the approach because the geometry and measurement quality improve; during the outbound leg, the covariance grows again and the estimate tends to lie near the upper bound. This behaviour is consistent with the incomplete model for the SRP, which doesn't take into account the attitude. Although the thrust model is imprecise, the thrust error state is capable of absorbing this error and keep the results within the expected covariance; however,

the constant bias and derive of the estimation surely is a consequence of the inaccurate model for the SRP.
Overall, the results demonstrate very high accuracy in both phases of the flyby, but they also indicate that, in its current configuration, the navigator is not suitable for long-duration low-thrust trajectories.

	RMS Position		**RMS Speed**	
Direction	**At flyby**	**After flyby**	**At flyby**	**After flyby**
Radial	0.43773	1.71768	0.00023	0.00041
Along-Track	0.09463	0.60072	0.00003	0.00010
Normal	0.15662	0.68433	0.00004	0.00009
3D	0.47444	1.94412	0.00023	0.00043

Table 5.3: Error in position and speed with LOS measurements

Overall, the navigator performs quite accurately; during the flyby, at maximum proximity to the asteroid, measurements are more accurate and the estimation reaches its peak precision; after the flyby, as the spacecraft gets further away, the estimation becomes less accurate. Although the navigator does not perform with its maximum achievable precision, this behaviour is expected: the MATLAB Navigator has been conceived as an orbital navigator, with no real interplanetary measurements. These results show that the navigator is however capable of working outside of its intended environment with fairly optimal results and achieve great results. With small refinements on the SRP model and new interplanetary measurements the navigator should be perfectly ready for real interplanetary cruises.

Chapter 6

Conclusion

6.1 Conclusion

To conclude, at the end of this internship, the initial objectives were achieved, notably for the required functionalities. However, during validation tests over the last few months, certain limitations have shown that the initial requirements were sufficient for the initial objective of an orbital navigator, but they were not sufficient to simulate interplanetary navigation at 100 %.
As shown in section 4.1, the current state of the navigator does indeed allow good estimation around a planet: the navigator is capable of correcting errors within the predicted covariance and maintaining an extreme level of accuracy. However, interplanetary navigation based solely on LOS measurements is not reliable; range measurements are also necessary
To obtain a complete estimate, it is necessary to measure the distance to the celestial objects. This can be achieved thanks to LiDAR measurements: LiDAR uses two-way signals to estimate the distance to the celestial bodies. It is indeed the LOS + LiDAR combination that allows navigation without GNSS.
However, both Line of Sight and LiDAR measurements require a nearby landmark, which means that high-precision estimation can only be achieved in close proximity to a celestial body. This behaviour is evident in all the test cases: as the spacecraft moves away, the estimation gradually diverges. This is perfectly coherent with the initial design of the MATLAB Navigator, which is intended to be used in orbits or nearby celestial bodies. To further challenge the navigator, an interplanetary trajectory was designed using optimal control theory to test its performance in deep space, which is not its intended environment. During the guidance development, the optimisation of an interplanetary trajectory posed new challenges in understanding the deep space environment, as well as the Optimal Control Theory; the guidance analysis found the optimal trajectory for a close flyby of an asteroid named 2000 SG_{344}.
The guidance phase of the thesis allowed for the complete analysis of an interplanetary mission, from the guidance phase to the navigation phase. The whole trajectory has been tested with the navigator; however, the results in the interplanetary cruise, as expected,

weren't of high precision. The final approach, closest to the asteroid, proved the high reliability of the navigator in the proximity of space objects.
As for next developments, the addition of attitude laws is necessary for a correct estimation of solar radiation pressure and required to complete the dynamic model. Also, a long-range distance measurement is necessary for interplanetary cruises.
This thesis has allowed me to address all phases of a project: from design and development to final validation through rigorous testing in various scenarios. It has also brought me to both the Guidance and Navigation analysis of an interplanetary mission, giving me a well-rounded perspective on interplanetary mission design and implementation.

Appendices

Appendix A

Appendix A: Terrestrial Reference Frames

A.1 Terrestrial frames

A.1.1 Transition VEIS/ITRF

The transition between (ITRF) and Veis is performed:

- by passing from ($ITRF$) to the true terrestrial frame, taking into account the pole model (u, v)

$$R_2(u).R_1(v) = \begin{pmatrix} \cos(u) & 0 & -\sin(u) \\ 0 & 1 & 0 \\ \sin(u) & 0 & \cos(u) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(v) & \sin(v) \\ 0 & -\sin(v) & \cos(v) \end{pmatrix} \tag{A.1}$$

  Considering $\cos(x) \approx 1$ and $\sin(x) \approx x$ for u and v:

$$R_2(u).R_1(v) \approx \begin{pmatrix} 1 & uv \approx 0 & -u \\ 0 & 1 & v \\ u & -v & 1 \end{pmatrix} \tag{A.2}$$

- by passing from the true terrestrial frame to Veis: rotation by the sidereal time θ_S, in which the UT1–UTC difference is neglected.

$$R_3(\theta_S) = \begin{pmatrix} \cos(\theta_S) & -\sin(\theta_S) & 0 \\ \sin(\theta_S)n & \cos(\theta_S) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.3}$$

$$\theta_S = a_1 + \omega_T(t_{TUC} - t_0) \tag{A.4}$$

with $t_0 = 01/01/1990$

$a_1 = 1.743078993205$ rad

Finally:

$$M_{ITRF->Veis} = \begin{pmatrix} \cos(\theta_S) & -\sin(\theta_S) & -u\cos(\theta_S) - v\sin(\theta_S) \\ \sin(\theta_S) & \cos(\theta_S) & -u\sin(\theta_S) + v\cos(\theta_S) \\ u & -v & 1 \end{pmatrix} \tag{A.5}$$

And correspondingly:

$$M_{Veis->ITRF} = \begin{pmatrix} \cos(\theta_S) & \sin(\theta_S) & u \\ -\sin(\theta_S) & \cos(\theta_S) & -v \\ -u\cos(\theta_S) - v\sin(\theta_S) & -u\sin(\theta_S) + v\cos(\theta_S) & 1 \end{pmatrix} \tag{A.6}$$

In order to transform the velocity of the satellite from one frame to the other, it is necessary to know the time derivative of the transition matrix VEIS/ITRF.
We therefore have the following matrices:

$$\omega_{Veis->ITRF} = \begin{pmatrix} -\sin(\theta_S) & \cos(\theta_S) & 0 \\ -\cos(\theta_S) & -\sin(\theta_S) & 0 \\ u\sin(\theta_S) - v\cos(\theta_S) & -u\cos(\theta_S) - v\sin(\theta_S) & 0 \end{pmatrix} \tag{A.7}$$

$$\omega_{ITRF->Veis} = \begin{pmatrix} -\sin(\theta_S) & -\cos(\theta_S) & u\sin(\theta_S) - v\cos(\theta_S) \\ \cos(\theta_S) & -\sin(\theta_S) & -u\cos(\theta_S) - v\sin(\theta_S) \\ 0 & 0 & 0 \end{pmatrix} \tag{A.8}$$

For the transformation of a PV record, we therefore have the following relation:

$$\begin{pmatrix} P \\ V \end{pmatrix}_{Veis} = \begin{pmatrix} M_{ITRF->Veis} & 0 \\ \omega_{ITRF->Veis} & M_{ITRF->Veis} \end{pmatrix} \begin{pmatrix} P \\ V \end{pmatrix}_{ITRF} \tag{A.9}$$

A.1.2 Pole motion matrix TIRF/ITRF

The terrestrial pole motion is the motion of the CIP relative to the terrestrial frame. This motion is described by two angular coordinates (u, v) expressed in arcseconds. These coordinates are made available on a daily basis by the IERS bulletins.

The pole motion matrix has the expression:

$$M_{TIRF \to ITRF} = R_3(s')R_2(-u)R_1(-v) \tag{A.10}$$

With s' the non-rotating origin of the TIO considered constant in BOLERO and whose value is: $s' = -47\mu asT$ with T the number of centuries since J2000 (TT).
The function *IERS__Polar_Motion_Matrix* makes it possible to compute this matrix.

A.1.3 Diurnal rotation matrix $M_{CIRF \to TIRF}$

This matrix is related to the rotation of the Earth about its axis. It is a rotation about the CIP axis (z-axis of the TIRF frame) by an angle called ERA (Earth Rotation Angle). The rotation of the Earth is not a regular motion with a constant rotation rate. As with the terrestrial pole motion, this angle is not directly predictable by a model. Its value is obtained through observations (mainly VLBI) and disseminated by the IERS bulletins. The bulletins do not supply the ERA directly, but UT1, which is related to the ERA.
The angle is calculated by the following relation [8]:

$$ERA = 2\pi(0.7790570.7790572732640 + 1.00273781191135448 \times t_{UT1}) \tag{A.11}$$

With t_{UT1} the time expressed in UT1 Julian days since J2000.
We then have:

$$M_{CIRF \to TIRF} = R_3(ERA) \tag{A.12}$$

A.1.4 Precession and nutation matrix $M_{GCRF \to CIRF}$

The orientation of the Earth's rotation axis (CIP) evolves under the action of the Sun, the Moon and the planets with a long-period motion called precession and a shorter-period motion called nutation. Both are included in the motion of the CIP relative to the celestial frame GCRF.
The matrix represents $M_{GCRF \to CIRF}$, i.e. the combined effects of precession, nutation and bias, and defines the orientation of the CIP and of the celestial longitude origin (the CIO), via the following relation:

$$M_{GCRF \to CIRF} = R_3(e)R_2(d)R_3(-(e+s)) \tag{A.13}$$

With,

- $e = atan^2\left(\frac{X}{Y}\right)$ if $X^2 + Y^2 > 0$ otherwise $e = 0$;
- $d = atan\left(\sqrt{\frac{X^2+Y^2}{1-(X^2+Y^2)}}\right)$;

- s the non-rotating origin of the CIO in the GCRF. This term is computed from the expansion of a polynomial series based on the SOFA algorithm and the IERS standards (function *IERS__Cio_Locator_S*).

The coordinates X, Y are the coordinates of the celestial pole CIP. They are computed by the function *IERS__XY* using the Fukushima–Williams precession angles and the IAU 2000A nutation model (series expansion). The model takes into account luni-solar and planetary nutation. There is a series decomposition of 678 terms for the luni-solar effects and 687 terms for the planetary effects.
A performance study showed that the computation time of this nutation model was particularly expensive in CPU time. A simpler model was therefore implemented consisting of 80 terms for the luni-solar effects and addition of a bias for the planetary terms (IAU 2000B model).
These models are based on the SOFA algorithms *nut00a* and *nut00b* and are described in the IERS 2010 convention. The IAU 2000B model provides the motion of the celestial pole with an accuracy that does not result in a deviation greater than 1 mas compared with that of the IAU 2000A model over the period 1995–2050.

Appendix B

Appendix B: Accelerations

B.1 Central Body Acceleration

$$\vec{\gamma}_{Central\,Body} = \vec{\nabla} U \tag{B.1}$$

With,

$$U = \frac{GM}{r} \sum_{l=0}^{n} \sum_{m=0}^{n} \left(\frac{R_{eq}}{r} \right)^n \bar{P}_{lm}(\sin\,\varphi)(\bar{C}_{lm} \cos m\lambda + \bar{S}_{lm} \sin m\lambda) \tag{B.2}$$

where $\bar{C}_{lm}$ and $\bar{S}_{lm}$ are the normalised harmonic coefficients of the potential of degree l and order m, given in the fixed frame of the central body.

B.1.1 Legendre Functions

The Legendre functions can be defined by relation to the Legendre polynomials:

$$P_{n,m}(x) = \left(1 - x^2\right)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x)$$

where $P_n(x)$ is the Legendre polynomial of degree n.
We have the following recurrence relations:

$$P_{n,n}(x) = (2n-1)\sqrt{(1-x^2)} P_{n-1,n-1}(x)$$
$$P_{n,n-1}(x) = (2n-1) x P_{n-1,n-1}(x)$$

and,

$$(n-m)P_{n,m}(x) = (2n-1)xP_{n-1,m}(x) - (n+m-1)P_{n-2,m}(x)$$

B.1.2 Working Variables

The following variables are introduced:

$$\alpha_m = \cos^m(\varphi)\cos(m\lambda)$$
$$\beta_m = \cos^m(\varphi)\sin(m\lambda)$$

and

$$W_{n,m} = \left(\frac{R_{eq}}{r}\right)^{n+1} \frac{1}{\cos^m \varphi} P_{n,m}(\sin\varphi)$$

with the composite terms:

$$A_{n,m} = \alpha_m W_{n,m}$$
$$B_{n,m} = \beta_m W_{n,m}$$

With these variables, one can write:

$$U(r,\varphi,\lambda) = \frac{GM}{r} \sum_{t}^{\infty} \sum_{m=0}^{n} \left(C_{n,m} A_{n,m} + S_{n,m} B_{n,m}\right)$$

The variables α_m and β_m are constructed as follows:

$$\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} \alpha_m \\ \beta_m \end{pmatrix} = \begin{pmatrix} \cos\lambda\cos\varphi & -\sin\lambda\cos\varphi \\ \sin\lambda\cos\varphi & \cos\lambda\cos\varphi \end{pmatrix} \begin{pmatrix} \alpha_{m-1} \\ \beta_{m-1} \end{pmatrix} = \begin{pmatrix} \hat{x} & -\hat{y} \\ \hat{y} & \hat{x} \end{pmatrix} \begin{pmatrix} \alpha_{m-1} \\ \beta_{m-1} \end{pmatrix}$$

and by using the recurrence relations on the Legendre functions:

$$W_{0,0} = \frac{R_{eq}}{r}$$
$$W_{n,n} = (2n-1)\frac{R_{eq}}{r} W_{n-1,n-1}$$
$$W_{n,n-1} = (2n-1)\frac{R_{eq}}{r} \hat{z} W_{n-1,n-1}$$
$$W_{n,m} = \frac{(2n-1)}{(n-m)} \frac{R_{eq}}{r} \hat{z} W_{n-1,m} - \frac{(n+m-1)}{(n-m)} \left(\frac{R_{eq}}{r}\right)^2 W_{n-2,m} \quad m = 0..n-2$$

B.1.3 Calculation of Derivatives

To calculate the partial derivatives with respect to the position, we use the relations on the elementary complex potentials defined by Cunningham:

$$V_{n,m} = \frac{1}{r^{n+1}} P_{n,m}(\sin\varphi) e^{im\lambda} = \frac{1}{R_{eq}^{n+1}} \left(A_{n,n} + iB_{n,m}\right)$$

We find the following derivatives on the coefficients A and B:

$$\begin{aligned}
\frac{\partial A_{n,m}}{\partial x} &= -\frac{1}{2R_1}\left[A_{n+1,m+1} - (n-m+2)(n-m+1)A_{n+1,m-1}\right] \text{ for } m = 1..n \\
&= -\frac{1}{R_1}A_{n+1,1} \text{ for } m = 0 \\
\frac{\partial A_{n,m}}{\partial y} &= -\frac{1}{2R_1}\left[B_{n+1,m+1} + (n-m+2)(n-m+1)B_{n+1,m-1}\right] \text{ for } m = 1..n \\
&= -\frac{1}{R_1}B_{n+1,1} \text{ for } m = 0
\end{aligned}$$

and:

$$\begin{aligned}
\frac{\partial B_{n,m}}{\partial x} &= -\frac{1}{2R_1}\left[B_{n+1,m+1} - (n-m+2)(n-m+1)B_{n+1,m-1}\right] \text{ for } m = 1..n \\
&= 0 \text{ for } m = 0 \\
\frac{\partial B_{n,m}}{\partial y} &= -\frac{1}{2R_1}\left[-A_{n+1,m+1} - (n-m+2)(n-m+1)A_{n+1,m-1}\right] \text{ for } m = 1..n \\
&= 0 \text{ for } m = 0 \\
\frac{\partial B_{n,m}}{\partial z} &= -\frac{1}{R_1}(n-m+1)B_{n+1,m} \text{ for } m = 0..n
\end{aligned}$$

B.1.4 Calculation of the Acceleration

The acceleration due to the potential is written:

$$\vec{\gamma}_{Central\,Body} = \vec{\nabla}U \tag{B.3}$$

and in rectangular coordinates we have:

$$\gamma_\xi = \frac{\partial U}{\partial \xi} = \frac{GM}{R_{eq}}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\left(C_{n,n}\frac{\partial A_{n,m}}{\partial \xi} + S_{n,m}\frac{\partial B_{n,m}}{\partial \xi}\right) \tag{B.4}$$

with $\xi = x, y$ or z, which can also be written in the form:

$$\begin{aligned}
\gamma_\xi &= -\frac{\mathrm{GM}}{\mathrm{R}_t^2}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\gamma_{n,m} \\
&\text{with} \\
\gamma_{n,m} &= -\mathrm{R}\left(\mathrm{C}_{n,m}\frac{\partial A_{n,m}}{\partial \xi} + \mathrm{S}_{n,m}\frac{\partial B_{n,m}}{\partial \xi}\right)
\end{aligned}$$

The previous expressions allow the calculation of γ :

$$m = 1, \ldots, n \quad \begin{cases} \gamma_{\bar{x}_{n,m}} = \dfrac{C_{n,m}}{2}\Big[A_{n+1,m+1} - (\chi_{n,m}+1)\chi_{n,m}A_{n+1,m-1}\Big] \\ \qquad\qquad + \dfrac{S_{n,m}}{2}\Big[B_{n+1,m+1} - (\chi_{n,m}+1)\chi_{n,m}B_{n+1,m-1}\Big], \\ \gamma_{\gamma_{n,m}} = \dfrac{C_{n,m}}{2}\Big[B_{n+1,m+1} + (\chi_{n,m}+1)\chi_{n,m}B_{n+1,m-1}\Big] \\ \qquad\qquad + \dfrac{S_{n,m}}{2}\Big[-A_{n+1,m+1} - (\chi_{n,m}+1)\chi_{n,m}A_{n+1,m-1}\Big], \\ \gamma_{z_{n,m}} = 2\chi_{n,m}\Big(\dfrac{C_{n,m}}{2}A_{n+1,m} + \dfrac{S_{n,m}}{2}B_{n+1,m}\Big) \end{cases} \tag{B.5}$$

$$\chi_{n,m} = n - m + 1.$$

And,

$$m = 0 \quad \begin{cases} \gamma_{n,0} = C_{n,0}A_{n+1,1} + S_{n,0}B_{n+1,1}, \\ \psi_{n,0} = C_{n,0}B_{n+1,1} - S_{n,0}A_{n+1,1}, \\ \gamma_{z_{n,0}} = (n+1)\Big(C_{n,0}A_{n+1,0} + S_{n,0}B_{n+1,0}\Big) \end{cases} \tag{B.6}$$

B.1.5 Normalisation

Instead of the classical normalisation of the harmonic coefficients of the Earth potential, we use the following normalisation, simpler to implement:

$$m \neq 0 \begin{cases} \overline{C_{n,m}} = \frac{\eta^m C_{n,m}}{2} \\ \overline{S_{n,m}} = \frac{\eta^m S_{n,m}}{2} \end{cases}$$

(where η is an arbitrary constant),
and:

$$m = 0 \begin{cases} \overline{C_{n,0}} = C_{n,0} \\ \overline{S_{n,0}} = S_{n,0} \end{cases}$$

The factor 2 is used to simplify the calculation of γ when $m \neq 0$.
The elementary potentials become:

$$\overline{W_{n,m}} = \frac{W_{n,n}}{\eta^m}$$

And the recurrence relations:

$$\bar{W}_{0,0} = \frac{R_{eq}}{r}$$
$$\bar{W}_{n,n} = (2n-1)\frac{R_e}{r}\frac{W_{n-1,n-1}}{\eta}$$
$$\bar{W}_{n,n-1} = (2n-1)\frac{R_{eq}}{r}\hat{z}W_{n-1,n-1}$$
$$\bar{W}_{n,m} = \frac{(2n-1)}{(n-m)}\frac{R_{eq}}{r}\hat{z}W_{n-1,m} - \frac{(n+m-1)}{(n-m)}\left(\frac{R_{eq}}{r}\right)^2 W_{n-2,m} \quad m = 0..n-2$$

In the Navigator, η was chosen equal to 20.

B.2 Atmospheric drag force

The drag acceleration due to atmospheric braking is written:
$\gamma_{Drag} = k\vec{S}$ with $\vec{S}$ the tangential unit vector (of the local orbital frame) and k the amplitude of the drag force. k is a parameter estimated by the filter.
The drag acceleration is considered constant during the sequence.
The calculation of the atmospheric drag acceleration is based on two models, the model chosen depending on the altitude of the satellite. For an altitude between 150 and 1000 km, the US76 model is used. For an altitude between 1000 km and 3000 km the DTM model is used.

$$\vec{\gamma}_{drag} = -\rho_0 e^{-\beta(h)(h-h_0)}\|\vec{V}\|\vec{V} \tag{B.7}$$

with:

$$\vec{V} = \vec{V}_{IN} - \begin{pmatrix} 0 \\ 0 \\ \omega_t \end{pmatrix} \wedge \vec{P_{IN}}$$

$\vec{V}_{IN}$: Satellite velocity in the inertial frame.
ω_t: Rotation rate of the Earth. ($\omega_t = 7.2921151467 \times 10^{-5}$ rad/s)
$\vec{P}_{IN}$: Satellite position in the inertial frame.

$$\beta(h) = \begin{cases} 0.072224 - 0.00030374h + 6.8107 \times 10^{-7}h^2 - \\ \quad 6.8654 \times 10^{-10}h^3 + 2.5151 \times 10^{-13}h^4 & \text{if } h < h_r \\ \frac{1}{19.187+0051572h} & \text{if } h >= h_r \end{cases} \tag{B.8}$$

$$h = \|\vec{P_{IN}}\| - R_t\sqrt{1 - \left(\frac{e\vec{P}_{IN_z}}{\|\vec{P}_{IN}\|}\right)^2} \tag{B.9}$$

with $\rho_0 = 0.254068762 \times 10^{-9}$ kg/m3
$h_r = 965.0746538482$ km

$h_0 = 200$ km
$e = 0.08181922146$ km

Appendix C

GNSS measurements

C.1 GNSS measurements techniques

C.1.1 Code measurement

The receiver continuously receives the code from the GNSS satellite, with a delay corresponding to the time taken by the wave to cover the distance between the emitting satellite and the receiving satellite. This delay is measured by the receiver as the difference between the moment of reception (receiver clock) and the moment of emission (emitter clock). However, the clocks of the receiver and of the satellite are not synchronised, which introduces an error in the measurement of the time that must be estimated. The accuracy of the code measurement depends on the receivers and is of metric order of magnitude [9].

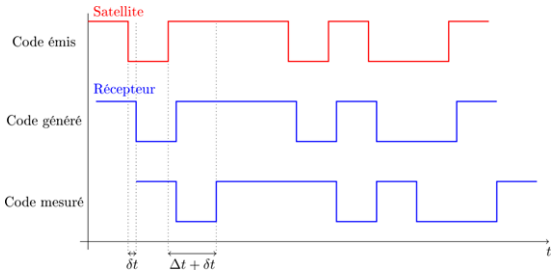


Figure C.1: GNSS code measurement

C.1.2 Phase measurement

Another positioning technique by GNSS relies on the measurement of the phase shift between the signals received and those generated by the receiver.

Another positioning technique by GNSS relies on the measurement of the phase shift between the signals received and those generated by the receiver. This measurement can be performed on the different carriers used by the GNSS. As in the code measurement, the absence of perfect synchronisation between the clocks of the receiver and of the satellite introduces a synchronisation error in the phase shift measurement, which must be estimated. The signal emitted by the satellite is received by the receiver, comprising an integer number

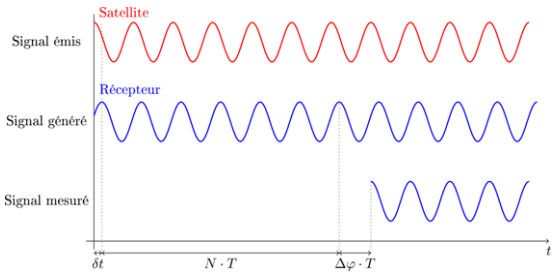


Figure C.2: GNSS phase measurement

of cycles and a fractional part of a cycle. However, only the fractional part of the propagation time of the signal between the satellite and the receiver, counted in number of cycles, can be measured by the receivers. The integer number of cycles that have elapsed since the beginning of the measurement is unknown, creating what is called an ambiguity (or a bias) in the phase measurement. The integer ambiguity corresponds to the integer number of cycles elapsed at the beginning of the measurement.
During an observation session, it is crucial not to interrupt the signal observed towards a satellite. In case of interruption, a cycle slip occurs, modifying the value of the integer ambiguity. This will have to be estimated again by the navigator during the next observation session.
For the GPS system, the wavelengths associated with the L1 and L2 carriers are respectively $\lambda_1 = 19.0$ cm and $\lambda_2 = 24.4$ cm. The accuracy of the phase measurement is millimetric [9].

C.1.3 Navigation message

In addition to being modulated by the phase using the code, GNSS signals are also modulated by the navigation message. This message contains information usable by the receiver, such as:

- The position of the satellites (ephemerides) and information concerning them (state of the satellite).
- Data for determining the emission date of the signal, according to the satellite's time scale.
- The satellite clock correction to be applied to compensate for its drift relative to GNSS time.
- More general information, such as a global parametric model of the ionosphere (to correct its effect) and an almanac of all satellites (health, approximate position).

C.1.4 Broadcast data and SP3

Broadcast (BRDC) data and SP3 are two types of data used in satellite navigation, but they differ in their origin, accuracy and use.
Broadcast data are provided directly by the GNSS satellites via the navigation message that each satellite regularly sends to the receivers. The Broadcast message contains in particular the satellites' ephemerides, the satellite time, the coefficients for the ionospheric correction. They are mainly used for real-time navigation applications.
SP3 files (Standard Product #3) are produced by the IGS (International GNSS Service). The raw data coming from numerous ground GNSS receivers are processed to compute the precise orbits and clocks of the satellites. An SP3 file contains very precise information on the positions and clocks of GNSS satellites. SP3 data are much more precise than the Broadcast ephemerides, with a typical accuracy of a few centimetres for the satellite orbits.

These data are obtained after complex processing of the GNSS observations, often several hours or days after the initial observations. SP3 files are used for applications requiring high precision, such as precise point positioning (PPP) or geodesy.
The analysis products, such as SP3 files, can be classified according to their latency, i.e. the time elapsed between the time of the last observations and the moment when the file is produced. The table below presents the versions of the orbit and clock product produced by the IGS for the GPS constellation as well as the Broadcast solution.

Solution	**1D RMS accuracy**	**Clock** 1σ $[ns]$	**Latency**
Broadcast	1.0 m	2.5 ns (= 75 cm)	Real time
Ultra-fast (predicted)	5 cm	1.5 ns (= 45 cm)	Real time
Ultra-fast (observed)	3 cm	0.05 ns (= 15 cm)	3-9 hours
Rapid	2.5 cm	0.025 ns (= 7.5 cm)	17-41 hours
Final	2.5 cm	0.02 ns (= 6 cm)	12-19 days

Table C.1: Performance of GPS Broadcast and IGS solutions [10]

C.2 Modelling of navigator measurements

In this navigator, as in **BOLERO**, the measurement is expressed in the form of the emission date in GNSS time: t_e^{GPS} associated with the reception date in receiver time t_r^r, expressed in counter time T_c. The receiver uses the navigation message to determine the transmitter clock and its relativistic correction: $h_{e_{app}} = h_e + \Delta t_{rel_e}$ and the position of the transmitter at the emission date $\vec{P}_{e/ITRF}(t_e^{GPS})$. The emitter part is therefore resolved upstream by the receiver.

Starting from the measurements, we seek to estimate: the trajectory of the emitter $\vec{\boldsymbol{P}}_r(t)$ and $\vec{\boldsymbol{V}}_r(t)$ and the remaining components of the state vector. The adopted reference frame is the CIRF; however, the partial derivatives remain the same for the ICRF.

Concerning the times, the lower index indicates the type of event: "e" for emission date, "r" for reception date. The upper index is the time scale used (receiver time, transmitter TGPS,...).

C.2.1 Pseudo-range satellite GNSS – satellite

The pseudo-range measurement, called *Pseudo-Range* (PR in metres) is modelled as follows:

$$PR_{code} = c\,(t_r^r - t_e^e) \tag{C.1}$$

These times are related to the reference time, here t^{GPS}, by a clock bias.

$$\begin{array}{rclcl} t_r^r & = & t_r^{GPS} + h_{r_{app}} & = & t_r^{GPS} + (h_r + \Delta t_{rel_r}) \\ t_e^e & = & t_e^{GPS} + h_{e_{app}} & = & t_e^{GPS} + (h_e + \Delta t_{rel_e}) \end{array} \tag{C.2}$$

with $h_{r/e_{app}}$ the apparent clock bias which includes the clock bias $h_{r/e}$ as well as the relativistic effect $\Delta t_{rel_{r/e}}$. Considering errors related to the speed and path of the signal, we obtain the following pseudo-range:

$$PR_{code} = \rho + c\,(h_r - h_e) + d_{iono} + d_{tropo} + d_{MT} + c\Delta t_{rel_r} - c\Delta t_{rel_e} \tag{C.3}$$

with the following considerations:

- $\rho = c\,(t_r^{GPS} - t_e^{GPS}) = ||\vec{P}_r(t_r^{GPS}) - \vec{P}_e(t_e^{GPS})||$ the geometric distance.
- $d_{iono} = 0$ by measurement combination (bi-frequency receiver or phase-code combination).
- $\Delta t_{rel_r} = -2\dfrac{\vec{P}_{r/Iner}\vec{V}_{r/Iner}}{c^2}$ the relativistic correction on the receiver clock.
- The tropospheric effect is null: $d_{tropo} = 0$.
- The multipath effect is null: $d_{MT} = 0$.

With all these simplifications, the measurement becomes:

$$PR_{code} = \rho + c\,(h_r - h_e) + c\Delta t_{rel_r} - c\Delta t_{rel_e} \tag{C.4}$$

As stated previously, the measurement is represented in the navigator by the emission date in system time: t_e^{GPS}. Rearranging equation C.1,

$$t_e^e = t_r^r - \frac{PR_{code}}{c}$$

By replacing PR_{code} by its expression, we obtain the theoretical code measurement:

$$\boxed{M_{code} = t_e^{GPS} = t_r^r - h_r - \frac{\rho}{c} + 2\frac{\vec{P}_{r/Iner}\vec{V}_{r/Iner}}{c^2}} \tag{C.5}$$

The navigation message provides the apparent clock bias $h_{e_{app}}$, which helps to obtain the previous relation by writing:

$$t_e^{GPS} = t_e^e - h_{e_{app}}$$

This is why the relativistic term of the emitter does not appear explicitly in the equation whereas that of the receiver is separated from the clock bias.

- Partial derivative of ρ with respect to the receiver position (vector of dimension 3)

$$\boxed{\frac{\partial \vec{\rho}}{\partial \boldsymbol{P}_r} \approx \frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}} \tag{C.6}$$

- Partial derivative of ρ with respect to time

$$\boxed{\frac{\partial \rho}{\partial t_r} \approx \frac{\partial \vec{\rho}}{\partial \boldsymbol{P}_r}\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big) \approx \big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)} \tag{C.7}$$

We now compute the partial derivatives of the **theoretical measurement** M with respect to the parameters of the state vector.

- Partial derivative with respect to the **position**

$$\boxed{\left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_{final}} \approx -\frac{1}{c}\frac{\partial \vec{\rho}}{\partial \boldsymbol{P}_r} \approx -\frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)} \tag{C.8}$$

- Partial derivative with respect to the **velocity**

$$\boxed{\left.\frac{\partial \vec{M}}{\partial \boldsymbol{V}_r}\right|_{t_{final}} \approx \left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_r}(t_r - t_{final}) \approx -\frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)(t_r - t_{final})} \tag{C.9}$$

- Partial derivative with respect to the **clock parameters** (T_c, $d = \Delta f/f$ and $dd = \dot{\Delta} f/f$)
  We have already shown the clock relations in Equation 3.52 which will be useful here.

  - Partial derivative with respect to h_r

$$\boxed{\left.\frac{\partial M}{\partial h_r}\right|_{t_{final}} \approx -\big(1 - \frac{1}{c}\frac{\partial \rho}{\partial t_r^{GPS}}\big) \approx -\left(1 - \frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)\right)} \tag{C.10}$$

  - Partial derivative with respect to d

$$\boxed{\left.\frac{\partial M}{\partial d}\right|_{t_{final}} \approx -\left(1 - \frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)\right)(t_r^{GPS} - t_{final})} \tag{C.11}$$

  - Partial derivative with respect to dd

$$\boxed{\left.\frac{\partial M}{\partial dd}\right|_{t_{final}} \approx \frac{1}{2}\left(1 - \frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)\right)(t_r^{GPS} - t_{final})^2} \tag{C.12}$$

- Partial derivative with respect to the **phase bias**

$$\boxed{\left.\frac{\partial M}{\partial biais}\right|_{t_{final}} = 0} \tag{C.13}$$

C.2.2 Phase satellite GNSS – satellite

The phase measurement is calculated from the PR measurement and with the satellite bias recovered in the state vector:

$$PR_{phase} = \rho + c\,(h_r - h_e) - d_{iono} + d_{tropo} + d_{MT} + c\Delta t_{rel_r} - c\Delta t_{rel_e} - \lambda N + \lambda\Phi_{W_{up}} \tag{C.14}$$

With the same simplifications as before, we add that the Wind-Up is zero: $\lambda\Phi_{W_{up}} = 0$. The latter is an effect due to the electromagnetic nature of the circularly polarised waves, intrinsic to GNSS signals. It depends on the relative orientation of the satellite and receiver antennas and the direction of the line of sight.

Moreover, λN is the bias or ambiguity of phase estimated in the state vector.
The measurement becomes:

$$\begin{aligned} PR_{phase} &= \rho + c\,(h_r - h_e) + c\Delta t_{rel_r} - c\Delta t_{rel_e} - \lambda N \\ PR_{phase} &= PR_{code} - \lambda N \end{aligned} \tag{C.15}$$

The measurement is also expressed by the emission date in GNSS time and is modelled as follows:

$$t_e^e = t_r^r - \frac{PR_{phase}}{c} \tag{C.16}$$

By replacing PR_{code} by its expression, we obtain the theoretical phase measurement:

$$\boxed{M_{phase} = t_e^{GPS} = t_r^r - h_r - \frac{\rho}{c} + 2\frac{\vec{P}_{r/Iner}\vec{V}_{r/Iner}}{c^2} + \frac{biais}{c}} \tag{C.17}$$

Where $biais = \lambda N$.
All the partial derivative results obtained for the GNSS Distance Measurement are identical here except for:

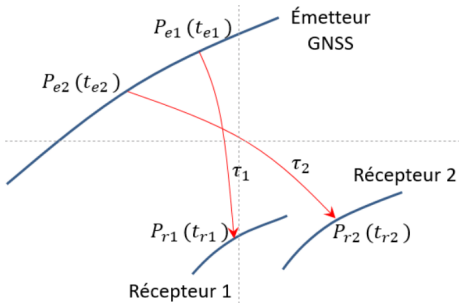
- Partial derivative of the measurement with respect to the **phase bias**

$$\boxed{\left.\frac{\partial M}{\partial biais}\right|_{t_{final}} = \frac{1}{c}} \tag{C.18}$$

C.2.3 Single difference of code satellite 1 – GNSS satellite – satellite 2

The single difference measurement is built from two measurements of the same emitting GNSS satellite P_e towards two different receivers P_{r1} and P_{r2}.

In the figure: t_{e1} and t_{e2} are the emission dates of the signals towards receivers 1 and 2; t_{r1} t_{r2} are the reception dates at receivers 1 and 2.
The single difference code measurement is therefore equal to:



$$SDPR_{12} = PR_2 - PR_1 \qquad \text{(C.19)}$$

In the navigator, PR measurements are expressed as emission dates in GNSS time t_e^{GNSS}.
Thus:

$$\begin{aligned} t_{e1}^{GNSS} &= t_{r1}^{r1} - \tfrac{PR_1}{c} - h_{e1} \\ t_{e2}^{GNSS} &= t_{r2}^{r2} - \tfrac{PR_2}{c} - h_{e2} \end{aligned} \qquad \text{(C.20)}$$

Thus:

$$t_{e2}^{GNSS} - t_{e1}^{GNSS} = t_{r2}^{r2} - t_{r1}^{r1} - \frac{PR_2 - PR_1}{c} - (h_{e2} - h_{e1}) \qquad \text{(C.21)}$$

The single difference measurement is expressed as the difference of the emission dates in GNSS time:

$$t_{e2}^{GNSS} - t_{e1}^{GNSS} = t_{r2}^{r2} - t_{r1}^{r1} - \frac{SDPR_{12}}{c} - (h_{e2} - h_{e1}) \qquad \text{(C.22)}$$

This measurement is modelled as:

$$\boxed{M_{SDCode} = t_{e2}^{GNSS} - t_{e1}^{GNSS} = t_{r2}^{GNSS} - t_{r1}^{GNSS} - \frac{\rho_2 - \rho_1}{c} + 2\frac{\vec{P}_{r2/Iner}\vec{V}_{r2/Iner} - \vec{P}_{r1/Iner}\vec{V}_{r1/Iner}}{c^2}} \qquad \text{(C.23)}$$

where ρ_1 and ρ_2 are the geometric distances between the emitter at the emission date and receivers 1 and 2 at the reception date. Thus, by making single difference measurements, we get rid of the error on the clock of the emitting satellite and therefore only the errors on the clocks of the receiving satellites remain. The ionospheric effect has also been removed by measurement combination. The last term of the equation is the relativistic correction (part due to the eccentricity of the orbit) on the clocks of receivers 1 and 2.

- Partial derivative with respect to the **clock parameters** (T_c, $d = \Delta f/f$ and $dd = \dot{\Delta f}/f$)

If we have chosen to estimate the clock for satellite 1:

$$\begin{aligned}
\frac{\partial M}{\partial T_{c1}} &= +(1 - \frac{1}{c}\frac{\partial \rho_1}{\partial t_{r1}}) \\
\frac{\partial M}{\partial d} &= \frac{\partial M}{\partial T_{c1}}.(t_{r1}^{GNSS} - t_{PVfinal}) \\
\frac{\partial M}{\partial dd} &= \frac{1}{2}\frac{\partial M}{\partial T_{c1}}.(t_{r1}^{GNSS} - t_{PVfinal})^2 = \frac{1}{2}\frac{\partial M}{\partial d}.(t_{r1}^{GNSS} - t_{PVfinal})
\end{aligned} \tag{C.24}$$

where $t_{PVfinal}$ is the end date of the current orbital window.

Otherwise these partial derivatives are zero: $\frac{\partial M}{\partial T_{c1}} = 0$, $\frac{\partial M}{\partial d_1} = 0$ and $\frac{\partial M}{\partial dd_1} = 0$
If we have chosen to estimate the clock for satellite 2:

$$\begin{aligned}
\frac{\partial M}{\partial T_{c2}} &= -(1 - \frac{1}{c}\frac{\partial \rho_2}{\partial t_{r2}}) \\
\frac{\partial M}{\partial d} &= \frac{\partial M}{\partial T_{c2}}.(t_{r2}^{GNSS} - t_{PVfinal}) \\
\frac{\partial M}{\partial dd} &= \frac{1}{2}\frac{\partial M}{\partial T_{c2}}.(t_{r2}^{GNSS} - t_{PVfinal})^2 = \frac{1}{2}\frac{\partial M}{\partial d}.(t_{r2}^{GNSS} - t_{PVfinal})
\end{aligned} \tag{C.25}$$

Otherwise these partial derivatives are zero: $\frac{\partial M}{\partial T_{c2}} = 0$, $\frac{\partial M}{\partial d_2} = 0$ and $\frac{\partial M}{\partial dd_2} = 0$

- Partial derivative with respect to the **positions** and **velocities** of receivers 1 and 2

If we do not free the position of a receiver (see $i_{P_{r1}}$, $i_{P_{r2}}$:)
$\Longrightarrow$ the partial derivatives with respect to that one are zero, otherwise:

$$\begin{aligned}
\overrightarrow{\frac{\partial M}{\partial P_{r2}}} &= \frac{-1}{c}\overrightarrow{\frac{\partial \rho_2}{\partial P_{r2}}} \\
\overrightarrow{\frac{\partial M}{\partial V_{r2}}} &= \overrightarrow{\frac{\partial M}{\partial P_{r2}}}.(t_{r2}^{GNSS} - t_{PVfinal}) \\
\overrightarrow{\frac{\partial M}{\partial P_{r1}}} &= \frac{1}{c}\overrightarrow{\frac{\partial \rho_1}{\partial P_{r1}}} \\
\overrightarrow{\frac{\partial M}{\partial V_{r1}}} &= \overrightarrow{\frac{\partial M}{\partial P_{r1}}}.(t_{r1}^{GNSS} - t_{PVfinal})
\end{aligned} \tag{C.26}$$

- Partial derivative with respect to the **phase bias** :

$$\begin{aligned}
\frac{\partial M}{\partial biais_1} &= 0 \\
\frac{\partial M}{\partial biais_2} &= 0
\end{aligned} \tag{C.27}$$

C.2.4 Single difference of phase satellite 1 – GNSS satellite – satellite 2

The single difference phase measurement follows the same principle as the single difference code measurement, being built from two measurements of the same emitting GNSS satellite P_e towards two different receivers P_{r1} and P_{r2}.

The single difference phase measurement is equal to:

$$SDPhase_{12} = M_{Phase_2} - M_{Phase_1} \tag{C.28}$$

Where $Phase_2$ and $Phase_1$ are the phase measurements defined by equation C.17 corresponding respectively to receiving satellites 2 and 1.
Phase measurements are expressed as emission dates in GNSS time t_e^{GNSS}.
Thus:

$$\begin{aligned} t_{e1}^{GNSS} &= t_{r1}^{r1} - \tfrac{PR_1}{c} - h_{e1} + \tfrac{biais_1}{c} \\ t_{e2}^{GNSS} &= t_{r2}^{r2} - \tfrac{PR_2}{c} - h_{e2} + \tfrac{biais_2}{c} \end{aligned} \tag{C.29}$$

Thus:

$$t_{e2}^{GNSS} - t_{e1}^{GNSS} = t_{r2}^{r2} - t_{r1}^{r1} - \frac{PR_2 - PR_1}{c} - (h_{e2} - h_{e1}) + \frac{biais_2 - biais_1}{c} \tag{C.30}$$

The SD measurement is therefore expressed as the difference of the emission dates in GNSS time:

$$t_{e2}^{GNSS} - t_{e1}^{GNSS} = t_{r2}^{r2} - t_{r1}^{r1} - \frac{SDR_{12}}{c} - (h_{e2} - h_{e1}) + \frac{biais_2 - biais_1}{c} \tag{C.31}$$

This measurement is modelled as:

$$M_{SDPhase} = t_{e2}^{GNSS} - t_{e1}^{GNSS}$$

$$\boxed{M_{SDPhase} = t_{r2}^{GNSS} - t_{r1}^{GNSS} - \frac{\rho_2 - \rho_1}{c} + \frac{biais_2 - biais_1}{c} + 2\frac{\vec{P}_{r2/Iner}\vec{V}_{r2/Iner} - \vec{P}_{r1/Iner}\vec{V}_{r1/Iner}}{c^2}} \tag{C.32}$$

where ρ_1 and ρ_2 are the geometric distances between the emitter at the emission date and receivers 1 and 2 at the reception date. The ionospheric effect has been removed by measurement combination. The last term of the equation is the relativistic correction (part due to the eccentricity of the orbit) on the clocks of receivers 1 and 2.

It should be noted that this measurement alone only allows the estimation of a difference of biases: $\Delta biais = biais_2 - biais_1$. In the navigator, this relative state $\Delta biais$ is not present in the state vector; the difference of biases must therefore be performed directly after the estimation.

The partial derivatives obtained for the GNSS Single Difference Code Measurement are reused with the exception of:

- Partial derivative with respect to the **phase bias** :
  If we have chosen to estimate the phase bias for satellite 1 :

$$\frac{\partial M}{\partial biais_1} = -\frac{1}{c} \tag{C.33}$$

If we have chosen to estimate the phase bias for satellite 2 :

$$\frac{\partial M}{\partial biais_2} = +\frac{1}{c} \tag{C.34}$$

C.2.5 ISL distance

Let S_1 be the position and velocity of the emitting satellite at the emission date in the inertial frame and S_2 the position and velocity of the receiving satellite at the reception date in the inertial frame.

$$\begin{aligned} S_1 &= \begin{bmatrix} x_1 & y_1 & z_1 & v_{x_1} & v_{y_1} & v_{z_1} \end{bmatrix}^T \\ S_2 &= \begin{bmatrix} x_2 & y_2 & z_2 & v_{x_2} & v_{y_2} & v_{z_2} \end{bmatrix}^T \end{aligned}$$

The distance ρ between satellites S_1 and S_2 is written:

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{C.35}$$

We suppose for the moment that the clocks of satellites S_1 and S_2 are synchronised. We also suppose that the measurement is made between antennas located at the centre of mass of each respective satellite. Let t_r be the reception date of the measurement. The emission date t_e is the measurement:

$$\boxed{M_{DistISL} = t_r - \frac{\rho}{c}} \tag{C.36}$$

To find the theoretical measurement, it is necessary to:

- Know the position of the emitter at the emission date $\vec{P}_e(t_e)$
- Know the position of the receiver at the reception date $\vec{P}_r(t_r)$

One then obtains:

$$\rho = ||\vec{P}_r(t_r) - \vec{P}_e(t_e)|| \tag{C.37}$$

We propose to interpolate the orbital windows of the satellites at the correct dates.

- Partial derivative with respect to the **position** and **velocity**

$$\boxed{\frac{\partial M}{\partial P_r(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_r(t_f)}} \tag{C.38}$$

$$\boxed{\frac{\partial M}{\partial V_r(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_r}(t_m - t_f)} \tag{C.40}$$

$$\boxed{\frac{\partial M}{\partial P_e(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_e(t_f)}} \tag{C.39}$$

$$\boxed{\frac{\partial M}{\partial V_e(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_e}(t_m - t_f)} \tag{C.41}$$

C.3 Measurement and Partial derivatives: detailed computation

C.3.1 Pseudo-range satellite GNSS – satellite

Measurement format

The measurement contains the following elements:

- MESURE DISTANCE_SATELLITE
- GNSS constellation number
- Antenna number
- Identifier of the emitting GNSS satellite
- Reception date of the measurement t_r^r (Seconds, Decimals)
- Emission date of the measurement t_e^{GPS} (Day, Seconds)
- Measurement covariance $cov(t_e)$ $[m^2]$
- Position of the emitter $P_e(t_e^{GPS})$ in ITRF in $[m]$
- Covariance of the position of the emitter $cov(P_e)$ in ITRF in $[m^2]$

Partial derivatives

These calculations of partial derivatives are necessary in estimation algorithms: we seek to estimate a certain number of parameters that influence the position of the emitter and/or the receiver. These parameters influence the propagation time and therefore the measurements made. The positions of the emitter and the receiver depend on:

- time t
- dynamic parameters p_{dyn_e} and p_{dyn_r} parametrising the movement of the emitter and the receiver

If one wishes to derive ρ with respect to time, this would give:

$$\frac{\partial \rho}{\partial t_r} = \frac{\partial \rho}{\partial \vec{\boldsymbol{P}}_r} \frac{\partial \vec{\boldsymbol{P}}_r(t_r)}{\partial t_r} + \frac{\partial \rho}{\partial \vec{\boldsymbol{P}}_e} \frac{\partial \vec{\boldsymbol{P}}_e(t_e)}{\partial t_r}$$

$$\frac{\partial \rho}{\partial t_r} = \frac{\vec{\partial} \rho}{\partial \boldsymbol{P}_r} \frac{\partial \vec{\boldsymbol{P}}_r(t_r)}{t_r} + \frac{\vec{\partial} \rho}{\partial \boldsymbol{P}_e} \frac{\partial \vec{\boldsymbol{P}}_e(t_r - \rho/c)}{\partial t_r}$$

We end up with these two terms of partial derivatives. We rather write the following equation:

$$\rho^2 = c^2\tau^2(t, p_{dyn_e}, p_{dyn_r}) = ||\vec{P}_r(t_r, p_{dyn_r}) - \vec{P}_e(t_r - \tau(t, p_{dyn_e}, p_{dyn_r}), p_{dyn_e})||^2$$

Which we write in the form of an implicit equation in τ.

$$\begin{aligned} F(t, p_{dyn_e}, p_{dyn_r}, \tau(t, p_{dyn_e}, p_{dyn_r})) =& c^2\tau^2(t, p_{dyn_e}, p_{dyn_r}) - ||\vec{P}_r(t_r, p_{dyn_r}) + \\ & - \vec{P}_e(t_r - \tau(t, p_{dyn_e}, p_{dyn_r}), p_{dyn_e})||^2 \end{aligned}$$

$$F(t, p_{dyn_e}, p_{dyn_r}, \tau(t, p_{dyn_e}, p_{dyn_r})) = 0 \tag{C.42}$$

According to the *Implicit Function Theorem*:

$$\begin{aligned} \frac{\partial \tau}{\partial t} &= -\frac{\frac{\partial F}{\partial t}}{\frac{\partial F}{\partial \tau}} \\ \frac{\partial \tau}{\partial p_{dyn_r}} &= -\frac{\frac{\partial F}{\partial p_{dyn_r}}}{\frac{\partial F}{\partial \tau}} \\ \frac{\partial \tau}{\partial p_{dyn_e}} &= -\frac{\frac{\partial F}{\partial p_{dyn_e}}}{\frac{\partial F}{\partial \tau}} \end{aligned}$$

In this case, we compute:

$$\begin{aligned} \frac{\partial F}{\partial \tau} &= 2c^2\tau - 2\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r - \tau)\big) \cdot \left.\frac{\partial \vec{\boldsymbol{P}}_e(t)}{\partial t}\right|_{t=t_r-\tau} \\ &= 2c^2\tau - 2\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r - \tau)\big) \cdot \vec{\boldsymbol{V}}_e(t_r - \tau) \\ \frac{\partial F}{\partial t_r} &= -2\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r - \tau)\big) \cdot (\vec{\boldsymbol{V}}_r(t_r) - \vec{\boldsymbol{V}}_e(t_r - \tau)) \\ \frac{\partial F}{\partial p_{dyn_r}} &= -2\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r - \tau)\big) \cdot \left.\frac{\partial \vec{\boldsymbol{P}}_r(t)}{\partial p_{dyn_r}}\right|_{t=t_r} \\ \frac{\partial F}{\partial p_{dyn_e}} &= -2\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r - \tau)\big) \cdot \left.\frac{\partial \vec{\boldsymbol{P}}_e(t)}{\partial p_{dyn_e}}\right|_{t=t_r-\tau} \end{aligned}$$

And we apply the theorem:

$$\frac{\partial \tau}{\partial t} = \frac{\big((\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)(\vec{\boldsymbol{V}}_r(t_r) - \vec{\boldsymbol{V}}_e(t_r-\tau))}{c^2\tau - \big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)\cdot \vec{\boldsymbol{V}}_e(t_r-\tau)}$$

$$\frac{\partial \tau}{\partial p_{dyn_r}} = \frac{\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)\cdot \left.\dfrac{\partial \vec{\boldsymbol{P}}_r(t)}{\partial p_{dyn_r}}\right|_{t=t_r}}{c^2\tau - \big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)\cdot \vec{\boldsymbol{V}}_e(t_r-\tau)}$$

$$\frac{\partial \tau}{\partial p_{dyn_e}} = \frac{\big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)\cdot \left.\dfrac{\partial \vec{\boldsymbol{P}}_e(t)}{\partial p_{dyn_e}}\right|_{t=t_r-\tau}}{c^2\tau - \big(\vec{\boldsymbol{P}}_r(t_r) - \vec{\boldsymbol{P}}_e(t_r-\tau)\big)\cdot \vec{\boldsymbol{V}}_e(t_r-\tau)}$$

From these equations, the partial derivatives of $\rho = c\tau$ can be calculated.

- Partial derivative of ρ with respect to the receiver position (vector of dimension 3)

$$\frac{\vec{\partial}\rho}{\partial \boldsymbol{P}_r} = \frac{c\big(\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}\big)}{c\rho - \big(\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}\big)\cdot \vec{\boldsymbol{V}}_{e/Iner}}$$

  which is approximated by:

$$\boxed{\frac{\vec{\partial}\rho}{\partial \boldsymbol{P}_r} \approx \frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}} \tag{C.46}$$

- Partial derivative of ρ with respect to time

$$\frac{\partial \rho}{\partial t_r} = \frac{c\big(\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}\big)\cdot\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)}{c\rho - \big(\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}\big)\cdot \vec{\boldsymbol{V}}_{e/Iner}}$$

  which is approximated by:

$$\boxed{\frac{\partial \rho}{\partial t_r} \approx \frac{\vec{\partial}\rho}{\partial \boldsymbol{P}_r}\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big) \approx \big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)} \tag{C.47}$$

  Remark : *This is the formula that is implemented in BOLERO and in the navigator, but in practice, $\vec{\boldsymbol{V}}_{e/Iner}$ is taken to be zero, because we do not have the data. The calculation is therefore wrong, however without any consequence, because this partial derivative is then used in a formula where it is completely negligible (see further the calculation of the derivative of the measurement with respect to the on-board clock).*

It is important to note that the state vector parameters estimated are those at the end date of the orbital window t_{final}, and not at the date of the measurement t_r^{GPS}.
Now,

$$\boxed{\left.\frac{\partial M(t_r)}{\partial x}\right|_{t_{final}} = \left.\frac{\partial M(t_r)}{\partial x}\right|_{t_r} \frac{\partial x(t_r)}{\partial x(t_{final})} = \left.\frac{\partial M(t_r)}{\partial x}\right|_{t_r} \boldsymbol{\Phi}_{t_r \to t_{final}}} \tag{C.48}$$

with $\mathbf{\Phi}_{t_r \to t_{final}}$ the transition matrix between t_r^{GPS} and t_{final}.
To do the exact calculation, one would have to consider that the propagation in the orbital window is done by 5th order Hermite interpolation. However, we simplify the transition matrix by considering:

$$\begin{cases} \boldsymbol{P}_{t_{final}} &= \boldsymbol{P}_r + \boldsymbol{V}_r(t_{final} - t_r) \\ \boldsymbol{V}_{t_{final}} &= \boldsymbol{V}_r \end{cases}$$

and therefore:

$$\mathbf{\Phi}_{t_r \to t_{final}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{I}(t_{final} - t_r) \\ 0 & \boldsymbol{I} \end{pmatrix}$$

We now compute the partial derivatives of the **theoretical measurement** M with respect to the parameters of the state vector.

- Partial derivative with respect to the **position**

  $$\left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_r} = -\frac{1}{c}\left.\frac{\partial \rho}{\partial \boldsymbol{P}_r}\right|_{t_r} + 2\frac{\vec{\boldsymbol{V}}_r}{c^2}$$

  We use Equation C.46 and take the date back from t_r to t_{final} by $\mathbf{\Phi}_{t_r \to t_{final}}$. Also, the relativistic term is considered negligible.

  $$\boxed{\left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_{final}} \approx -\frac{1}{c}\frac{\vec{\partial \rho}}{\partial \boldsymbol{P}_r} \approx -\frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)} \tag{C.49}$$

- Partial derivative with respect to the **velocity**

  $$\left.\frac{\partial \vec{M}}{\partial \boldsymbol{V}_r}\right|_{t_{final}} = \left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_r} \frac{\partial \boldsymbol{P}_r(t_r)}{\partial \boldsymbol{V}_r(t_{final})}$$

  The partial derivative $\dfrac{\partial \vec{M}}{\partial \boldsymbol{P}_r}$ appears because the measurement M depends only on position and not on velocity. Now,

  $$\frac{\partial \boldsymbol{P}_r(t_r)}{\partial \boldsymbol{V}_r(t_{final})} = -\boldsymbol{I}(t_{final} - t_r)$$

  We therefore obtain:

  $$\boxed{\left.\frac{\partial \vec{M}}{\partial \boldsymbol{V}_r}\right|_{t_{final}} \approx \left.\frac{\partial \vec{M}}{\partial \boldsymbol{P}_r}\right|_{t_r} (t_r - t_{final}) \approx -\frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)(t_r - t_{final})} \tag{C.50}$$

- Partial derivative with respect to the **clock parameters** (T_c, $d = \Delta f/f$ and $dd = \dot{\Delta} f/f$)

  We have already demonstrated the clock relations in Equation 3.52 which will be useful here.

  - Partial derivative with respect to T_c

$$\left.\frac{\partial M}{\partial T_c}\right|_{t_r} = \frac{\partial t_r^{GPS}}{\partial T_c} - \frac{1}{c}\frac{\partial \rho}{\partial T_c} = \frac{\partial t_r^{GPS}}{\partial T_c} - \frac{1}{c}\frac{\partial \rho}{\partial t_r^{GPS}}\frac{\partial t_r^{GPS}}{\partial T_c}$$

$$\left.\frac{\partial M}{\partial T_c}\right|_{t_r} = \frac{\partial t_r^{GPS}}{\partial T_c}\big(1 - \frac{1}{c}\frac{\partial \rho}{\partial t_r^{GPS}}\big)$$

    Using the equation developed in **??** and denoting T_{cr} the reception date in counter time, we have:

$$t_r^{GPS} = t^{GPS} + \frac{(T_{cr} - T_c)}{1+d}\big[1 - \frac{1}{2}\frac{dd}{(1+d)^2}(T_{cr} - T_c)\big]$$

    And we calculate $\dfrac{\partial t_r^{GPS}}{\partial T_c}$ by differentiating with respect to T_c:

$$\left.\frac{\partial t_r^{GPS}}{\partial T_c}\right|_{T_{cr}} = \frac{-1}{1+d} + \frac{dd}{(1+d)^3}(T_{cr} - T_c)$$

    We can neglect d relative to 1 and likewise $dd(T_{cr} - T_c)$ is also negligible relative to 1.

$$\left.\frac{\partial t_r^{GPS}}{\partial T_c}\right|_{T_{cr}} \approx -1$$

    We end up with:

$$\left.\frac{\partial M}{\partial T_c}\right|_{t_r} \approx -\big(1 - \frac{1}{c}\frac{\partial \rho}{\partial t_r^{GPS}}\big)$$

    Which we calculate at the end date of the orbital window t_{final} using the transition matrix already calculated in Equation 3.52 (here = 1):

$$\boxed{\left.\frac{\partial M}{\partial T_c}\right|_{t_{final}} \approx -\big(1 - \frac{1}{c}\frac{\partial \rho}{\partial t_r^{GPS}}\big) \approx -\left(1 - \frac{1}{c}\Big(\frac{\vec{\boldsymbol{P}}_{r/Iner} - \vec{\boldsymbol{P}}_{e/Iner}}{\rho}\Big)\big(\vec{\boldsymbol{V}}_{r/Iner} - \vec{\boldsymbol{V}}_{e/Iner}\big)\right)} \tag{C.51}$$

    The second term is indeed negligible due to its factor in $1/c$. This justifies the approximation stated earlier.

  - Partial derivative with respect to d

$$\left.\frac{\partial M}{\partial d}\right|_{t_{final}} = \left.\frac{\partial M}{\partial T_c}\right|_{t_r} \frac{\partial T_c(t_r^{GPS})}{\partial d(t_{final})}$$

    We use Equation 3.52 to obtain the second term (here $= -\Delta t$).

$$\boxed{\left.\frac{\partial M}{\partial d}\right|_{t_{final}} \approx -\left(1-\frac{1}{c}\left(\frac{\vec{\boldsymbol{P}}_{r/Iner}-\vec{\boldsymbol{P}}_{e/Iner}}{\rho}\right)\left(\vec{\boldsymbol{V}}_{r/Iner}-\vec{\boldsymbol{V}}_{e/Iner}\right)\right)(t_r^{GPS}-t_{final})} \tag{C.52}$$

– Partial derivative with respect to dd

$$\left.\frac{\partial M}{\partial dd}\right|_{t_{final}} = \left.\frac{\partial M}{\partial T_c}\right|_{t_r} \frac{\partial T_c(t_r^{GPS})}{\partial dd(t_{final})}$$

We use Equation 3.52 to obtain the second term (here $= -\frac{1}{2}\Delta t^2$).

$$\left.\frac{\partial M}{\partial dd}\right|_{t_{final}} = \left.\frac{\partial M}{\partial T_c}\right|_{t_r} (-\frac{1}{2}\Delta t^2) = \frac{1}{2}\Delta t \left.\frac{\partial M}{\partial d}\right|_{t_{final}}$$

Which gives:

$$\left.\frac{\partial M}{\partial dd}\right|_{t_{final}} \approx -\frac{1}{2}\left(1-\frac{1}{c}\left(\frac{\vec{\boldsymbol{P}}_{r/Iner}-\vec{\boldsymbol{P}}_{e/Iner}}{\rho}\right)\left(\vec{\boldsymbol{V}}_{r/Iner}-\vec{\boldsymbol{V}}_{e/Iner}\right)\right)(t_r^{GPS}-t_{final})(t_{final}-t_r^{GPS})$$

$$\boxed{\left.\frac{\partial M}{\partial dd}\right|_{t_{final}} \approx \frac{1}{2}\left(1-\frac{1}{c}\left(\frac{\vec{\boldsymbol{P}}_{r/Iner}-\vec{\boldsymbol{P}}_{e/Iner}}{\rho}\right)\left(\vec{\boldsymbol{V}}_{r/Iner}-\vec{\boldsymbol{V}}_{e/Iner}\right)\right)(t_r^{GPS}-t_{final})^2} \tag{C.53}$$

- Partial derivative with respect to the **phase bias**

$$\boxed{\left.\frac{\partial M}{\partial biais}\right|_{t_{final}} = 0} \tag{C.54}$$

Note : In **BOLERO**, the counter time T_c is defined in the state whereas in this navigator it is directly the clock bias h_r. It is important to understand the relationship between T_c and the clock bias h_r:

$$T_c = TGPS + h_r$$

We must analyse the partial derivatives of the distance measurements with respect to T_c and h_r.

Partial derivative with respect to T_c :
The measurement M depends on T_c. Thus, we have $M = f(T_c)$. The partial derivative of M with respect to T_c is simply:

$$\frac{\partial M}{\partial T_c} = f'(T_c)$$

Partial derivative with respect to h_r :
Using the relation $T_c = TGPS + h_r$, we can write M as a function of h_r: $M = f(TGPS + h_r)$. The partial derivative of M with respect to h_r is:

$$\frac{\partial M}{\partial h_r} = \frac{\partial f(TGPS + h_r)}{\partial h_r}$$

Applying the chain rule, we obtain:

$$\frac{\partial M}{\partial h_r} = \frac{\partial f(TGPS + h_r)}{\partial (TGPS + h_r)} \cdot \frac{\partial (TGPS + h_r)}{\partial h_r}$$

Since $\frac{\partial (TGPS + h_r)}{\partial h_r} = 1$, this simplifies the equation to:

$$\frac{\partial M}{\partial h_r} = \frac{\partial f(TGPS + h_r)}{\partial (TGPS + h_r)}$$

Recognising that $TGPS + h_r$ is just T_c, we obtain:

$$\frac{\partial M}{\partial h_r} = \frac{\partial f(T_c)}{\partial T_c}$$

The partial derivatives of distance with respect to T_c and h_r are therefore identical:

$$\boxed{\frac{\partial M}{\partial T_c} = \frac{\partial M}{\partial h_r}}$$

This result is based on the fact that the variation of the measurement with respect to the counter date T_c is directly equivalent to its variation with respect to the clock bias h_r, because T_c is linearly related to h_r with a coefficient of 1. This applies to all measurements.

Phase satellite GNSS – satellite

Measurement format

The measurement contains the following elements:

- MESURE PHASE_SATELLITE
- Number of the satellite concerned
- GNSS constellation number
- Antenna number
- Identifier of the emitting GNSS satellite
- Reception date of the measurement t_r^r (Seconds, Decimals)
- Emission date of the measurement t_e^{GPS} (Day, Seconds)
- Measurement covariance $cov(t_e)$ $[m^2]$

- Position of the emitter $P_e(t_e^{GPS})$ in ITRF in $[m]$
- Covariance of the position of the emitter $cov(P_e)$ in ITRF in $[m^2]$
- CS/NCS: New GNSS satellite (Phase jump / No phase jump)

Partial derivatives

All the partial derivative results obtained for the GNSS Distance Measurement are identical here except for:

- Partial derivative of the measurement with respect to the **phase bias**

$$\boxed{\left.\frac{\partial M}{\partial biais}\right|_{t_{final}} = \frac{1}{c}} \tag{C.55}$$

Single difference of code satellite 1 – GNSS satellite – satellite 2

Measurement format

The measurement contains the following elements:

- MESURE SD_CODE
- Number indicating receiver satellite 1
- Number indicating receiver satellite 2
- GNSS constellation number
- Number of the antenna of receiver satellite 1 that captured the measurement
- Number of the antenna of receiver satellite 2 that captured the measurement
- Identifier of the emitting GNSS satellite
- Reception date of the measurement t_{r1}^{r1} (Seconds, Decimals)
- Reception date of the measurement t_{r2}^{r2} (Seconds, Decimals)
- Emission date of the measurement t_{e1}^{GNSS} (Day, Seconds)
- Emission date of the measurement t_{e2}^{GNSS} (Day, Seconds)
- Measurement covariance $cov(t_{e1})$ $[m^2]$
- Measurement covariance $cov(t_{e2})$ $[m^2]$
- Position of the emitter $P_e(t_{e1}^{GNSS})$ in ITRF in $[m]$

- Position of the emitter $P_e(t_{e2}^{GNSS})$ in ITRF in $[m]$
- Covariance of the position of the emitter $cov(P_e)$ in ITRF in $[m^2]$

Partial derivatives

We now compute the partial derivatives, at the end date of the orbital window, of the theoretical measurement M with respect to the parameters of the state vector.

- Partial derivative with respect to the **clock parameters** (T_c, $d = \Delta f/f$ and $dd = \dot{\Delta} f/f$)
  If we have chosen to estimate the clock for satellite 1:

$$\begin{aligned} \frac{\partial M}{\partial T_{c1}} &= +(1 - \frac{1}{c}\frac{\partial \rho_1}{\partial t_{r1}}) \\ \frac{\partial M}{\partial d} &= \frac{\partial M}{\partial T_{c1}}.(t_{r1}^{GNSS} - t_{PVfinal}) \\ \frac{\partial M}{\partial dd} &= \frac{1}{2}\frac{\partial M}{\partial T_{c1}}.(t_{r1}^{GNSS} - t_{PVfinal})^2 = \frac{1}{2}\frac{\partial M}{\partial d}.(t_{r1}^{GNSS} - t_{PVfinal}) \end{aligned} \tag{C.56}$$

  where $t_{PVfinal}$ is the end date of the current orbital window.

  Otherwise these partial derivatives are zero: $\frac{\partial M}{\partial T_{c1}} = 0$, $\frac{\partial M}{\partial d_1} = 0$ and $\frac{\partial M}{\partial dd_1} = 0$
  If we have chosen to estimate the clock for satellite 2:

$$\begin{aligned} \frac{\partial M}{\partial T_{c2}} &= -(1 - \frac{1}{c}\frac{\partial \rho_2}{\partial t_{r2}}) \\ \frac{\partial M}{\partial d} &= \frac{\partial M}{\partial T_{c2}}.(t_{r2}^{GNSS} - t_{PVfinal}) \\ \frac{\partial M}{\partial dd} &= \frac{1}{2}\frac{\partial M}{\partial T_{c2}}.(t_{r2}^{GNSS} - t_{PVfinal})^2 = \frac{1}{2}\frac{\partial M}{\partial d}.(t_{r2}^{GNSS} - t_{PVfinal}) \end{aligned} \tag{C.57}$$

  Otherwise these partial derivatives are zero: $\frac{\partial M}{\partial T_{c2}} = 0$, $\frac{\partial M}{\partial d_2} = 0$ and $\frac{\partial M}{\partial dd_2} = 0$

- Partial derivative with respect to the **positions** and **velocities** of receivers 1 and 2

  If we do not free the position of a receiver (see $i_{P_{r1}}$, $i_{P_{r2}}$:)
  $\Longrightarrow$ the partial derivatives with respect to that one are zero, otherwise:

$$\begin{aligned} \overrightarrow{\frac{\partial M}{\partial P_{r2}}} &= \frac{-1}{c}\overrightarrow{\frac{\partial \rho_2}{\partial P_{r2}}} \\ \overrightarrow{\frac{\partial M}{\partial V_{r2}}} &= \overrightarrow{\frac{\partial M}{\partial P_{r2}}}.(t_{r2}^{GNSS} - t_{PVfinal}) \\ \overrightarrow{\frac{\partial M}{\partial P_{r1}}} &= \frac{1}{c}\overrightarrow{\frac{\partial \rho_1}{\partial P_{r1}}} \\ \overrightarrow{\frac{\partial M}{\partial V_{r1}}} &= \overrightarrow{\frac{\partial M}{\partial P_{r1}}}.(t_{r1}^{GNSS} - t_{PVfinal}) \end{aligned} \tag{C.58}$$

- Partial derivative with respect to the **phase bias** :

$$\begin{aligned} \frac{\partial M}{\partial biais_1} &= 0 \\ \frac{\partial M}{\partial biais_2} &= 0 \end{aligned} \tag{C.59}$$

Single difference of phase satellite 1 – GNSS satellite – satellite 2

Measurement format

The measurement contains the following elements:

- MESURE SD_PHASE
- Number indicating receiver satellite 1
- Number indicating receiver satellite 2
- GNSS constellation number
- Number of the antenna of receiver satellite 1 that captured the measurement
- Number of the antenna of receiver satellite 2 that captured the measurement
- Number N_e of the emitting GNSS satellite (PRN)
- Reception date of the measurement t_{r1}^{r1} (Seconds, Decimals)
- Reception date of the measurement t_{r2}^{r2} (Seconds, Decimals)
- Emission date of the measurement t_{e1}^{GNSS} (Day, Seconds)
- Emission date of the measurement t_{e2}^{GNSS} (Day, Seconds)
- Measurement covariance $cov(t_{e1})$ $[m^2]$
- Measurement covariance $cov(t_{e2})$ $[m^2]$
- Position of the emitter $P_e(t_{e1}^{GNSS})$ in ITRF in $[m]$
- Position of the emitter $P_e(t_{e2}^{GNSS})$ in ITRF in $[m]$
- Covariance of the position of the emitter $cov(P_e)$ in ITRF in $[m^2]$
- CS/NCS: New GNSS satellite for receiver 1 (Phase jump / No phase jump)
- CS/NCS: New GNSS satellite for receiver 2 (Phase jump / No phase jump)

Partial derivatives

The partial derivatives obtained for the GNSS Single Difference Code Measurement are reused with the exception of:

- Partial derivative with respect to the **phase bias** :
  If we have chosen to estimate the phase bias for satellite 1 :

$$\frac{\partial M}{\partial biais_1} = -\frac{1}{c} \tag{C.60}$$

  If we have chosen to estimate the phase bias for satellite 2 :

$$\frac{\partial M}{\partial biais_2} = +\frac{1}{c} \tag{C.61}$$

ISL distance measurements

Measurement format

The measurement contains the following elements:

- MESURE DISTANCE_INTER_SATELLITE
- Number of the emitting satellite S_1
- Number of the receiving satellite S_2
- Number of the emitting satellite antenna
- Number of the receiving satellite antenna
- Reception date of the measurement t_r (Days / Seconds)
- Emission date of the measurement t_e (Days / Seconds)
- Measurement covariance $cov(t_e)$ (m^2)

Partial derivatives

The distance ρ between satellites S_1 and S_2 is written:

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{C.62}$$

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{C.63}$$

The partial derivatives with respect to position are:

$$\frac{\partial \rho}{\partial x_1} = \frac{x_1 - x_2}{\rho} \tag{C.64}$$

and:

$$\frac{\partial \rho}{\partial x_2} = \frac{x_2 - x_1}{\rho} \tag{C.65}$$

The partial derivatives of the other components are calculated in the same way.
The partial derivatives with respect to velocity are:

$$\frac{\partial \rho}{\partial v_{x_1}} = 0 \tag{C.66}$$

$$\frac{\partial \rho}{\partial v_{x_2}} = 0 \tag{C.67}$$

The partial derivatives of the measurement at the reception date t_m are:

$$\frac{\partial M}{\partial P_r} = \frac{\partial M}{\partial \rho}\frac{\partial \rho}{\partial P_r}$$

$$\frac{\partial M}{\partial P_r} = -\frac{1}{c}\frac{\partial \rho}{\partial P_r}$$

$$\frac{\partial M}{\partial V_r} = \frac{\partial M}{\partial P_r}\frac{\partial P_r}{\partial V_r}$$

Similarly for the emitter satellite:

$$\frac{\partial M}{\partial P_e} = -\frac{1}{c}\frac{\partial \rho}{\partial P_e}$$

The correction is not made at the measurement date t_m but at the end date of the orbital window t_f.
We use the transition matrix already demonstrated in subsection 3.2.1 :

$$\Phi_{PV_{t_i \to t_f}} = \begin{pmatrix} \frac{\partial P_2}{\partial P_1} & \frac{\partial P_2}{\partial V_1} \\ \\ \frac{\partial V_2}{\partial P_1} & \frac{\partial V_2}{\partial V_1} \end{pmatrix}$$

$$\Phi_{PV} = \begin{pmatrix} I_3 + \frac{\Delta t^2}{6}(2G_k + G_{k+1}) & \Delta t(I_3 + \frac{\Delta t^2}{12}(G_k + G_{k+1}) \\ \\ \frac{\Delta t}{2}(G_k + G_{k+1}) & I_3 + \frac{\Delta t^2}{6}(G_k + 2G_{k+1}) \end{pmatrix}$$

$$\Phi_{PV} \approx \begin{pmatrix} I_3 & \Delta t(I_3) \\ \\ \frac{\Delta t}{2}(G_k + G_{k+1}) & I_3 \end{pmatrix}$$

We use this matrix to know the partial derivative at the end of the orbital window.

- Partial derivative with respect to the **position**

$$\boxed{\frac{\partial M}{\partial P_r(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_r(t_f)}} \tag{C.68}$$

Similarly, for the emitter satellite:

$$\boxed{\frac{\partial M}{\partial P_e(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_e(t_f)}} \tag{C.69}$$

- Partial derivative with respect to the **velocity**

$$\frac{\partial M}{\partial V_r(t_f)} = \frac{\partial P_r(t_m)}{\partial V_r(t_f)}\frac{\partial M}{\partial P_r(t_m)}$$

$$\frac{\partial M}{\partial V_r(t_f)} = I_3(t_m - t_f)\frac{\partial M}{\partial P_r(t_m)}$$

$$\boxed{\frac{\partial M}{\partial V_r(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_r}(t_m - t_f)} \tag{C.70}$$

Similarly, for the emitter satellite:

$$\boxed{\frac{\partial M}{\partial V_e(t_f)} = -\frac{1}{c}\frac{\partial \rho}{\partial P_e}(t_m - t_f)} \tag{C.71}$$

Bibliography

[1] CNES. *CNES en bref*. 2024. URL: https://cnes.fr (cit. on p. 3).

[2] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods, and Applications*. Physics and astronomy online library. Springer Berlin Heidelberg, 2000. ISBN: 9783540672807. URL: https://books.google.fr/books?id=hABRnDlDkyQC (cit. on pp. 23, 44).

[3] Álvaro Flórez González. *Développement d'un navigateur multi-satellite pour des applications de vols en formation*. Tech. rep. XXX-NT-xxx-00xx-CNES. Document interne, non publié. CNES, 2024 (cit. on p. 37).

[4] service DTN/DV/OR. *BOLERO Manuel Algorithmique*. Tech. rep. XXX-NT-xxx-00xx-CNES. Document interne, non publié. CNES, 2020 (cit. on p. 37).

[5] F.L. Markley. "Approximate cartesian state transition matrix". In: *Journal of the Astronautical Sciences* 34.2 (1986), pp. 161–169 (cit. on p. 44).

[6] JPL. "Sentry: Earth Impact Monitoring". In: (2004). URL: https://cneos.jpl.nasa.gov/sentry/details.html#?des=2000%20SG344 (cit. on p. 82).

[7] Marilyn K. Gordon Lawrence F. Shampine. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. 1975 (cit. on p. 85).

[8] N Capitaine, B. Guinot, and D. McCarthy. "Definition of the Celestial Ephemeris origin and of UT1 in the International Celestial Reference Frame". In: *Astron. Astrophys.* 355.1 (2000), pp. 398–405 (cit. on p. 99).

[9] Orpheon. *Le positionnement GPS-GNSS*. 2024. URL: https://reseau-orpheon.fr/le-reseau-orpheon/le-positionnement-gps-gnss/ (cit. on pp. 107, 108).

[10] IGS. *Performances des broadcast et solutions IGS*. URL: https://igs.org/products/ (cit. on p. 109).

Glossary

Acronyms

CNES Centre National d'Études Spatiales, the French space agency.

DTN Direction Technique et Numérique, the technical and digital directorate of CNES.

GNC Guidance, Navigation and Control.

POD Precise Orbit Determination.

IAU International Astronomical Union.

BIPM Bureau International des Poids et Mesures.

SSB Solar System Barycentre (or Barycenter).

BOLERO CNES reference library for orbital navigation, providing physical models and navigation methods.

Patrius CNES Java library for orbit propagation, dynamics and mission analysis.

LOTUS CNES Java library including measurement modelling classes (e.g. landmark and LiDAR measurements).

Orbito CNES library used for orbit-related computations in the mission generator.

MATLAB Navigator's specifics

MissionData JSON input file defining the global configuration of a simulation: physical parameters, bodies, active forces, Kalman filter parameters, satellite properties, sensor locations and other mission settings.

Directives file Text input file describing the timeline of the mission: start and end times, initial orbit, extrapolation instants and list of measurements to be used during the simulation.

Telecommand file (tc) Input file used to modify navigator parameters during the simulation (e.g. forces, attitudes, thresholds) at specified times.

`simu` Main internal data structure of the Navigator storing simulation-wide information: physical constants, perturbing bodies, active forces, time scales and configuration parameters.

`sat` Internal data structure representing the spacecraft (or spacecrafts), storing clock time, state vectors, measurements and navigation results.

Orbital window Fixed-duration time interval between two extrapolation instants. The EKF runs over successive orbital windows, propagating and updating the state within each window.

Reference frames and time scales

Inertial frame Reference frame in which Newton's laws hold without fictitious forces; used for dynamics and state propagation (e.g. ICRF).

Body-fixed frame Frame rigidly attached to a celestial body, rotating with it; used for local computations on or near the body.

VEIS Vrai Équinoxe Initial du Système, historical celestial reference frame used in the Navigator.

CIRF Celestial Intermediate Reference Frame.

ICRF International Celestial Reference Frame.

GCRF Geocentric Celestial Reference Frame.

EME2000 Earth Mean Equator and Equinox of J2000 reference frame.

EoD Earth Orientation at Date frame.

ITRF International Terrestrial Reference Frame.

TIRF Terrestrial Intermediate Reference Frame.

BCF frame Body-centred, body-fixed frame of the central body, defined by its rotation axis and orientation model.

QSW frame Local orbital frame with axes along radial (R), along-track (S) and cross-track (W) directions, used to analyse position and velocity errors.

TNW frame Local orbital frame defined by tangential (T), normal (N) and W axes, conceptually similar to QSW.

Julian Date (JD) Continuous count of days used in astronomy and ephemerides.

CJD Modified Julian date used in the Navigator: number of days since 01/01/1950 at midnight (JD −2433282.5).

TAI International Atomic Time, time scale based on atomic clocks and used as fundamental reference.

TT Terrestrial Time, time scale derived from TAI and used for astronomical observations from Earth.

UTC Coordinated Universal Time, civil time scale differing from TAI by an integer number of leap seconds.

UT1 Earth-rotation-based time scale; for low precision applications it is often approximated as equal to UTC.

TDB Barycentric Dynamical Time, relativistic time scale used in barycentric ephemerides and interplanetary dynamics.

GPS time Time scale associated with GNSS systems, used for GNSS measurements and conversions.

Ephemerides and celestial mechanics

Ephemerides Tabulated or modelled positions (and often velocities) of celestial bodies as functions of time, used to compute perturbations and observation geometries.

Meeus analytical model Analytical model providing approximate positions of the Sun and Moon in an Earth-centred frame, with reduced computational cost and limited accuracy.

Chebyshev coefficients Coefficients of Chebyshev polynomial expansions used to represent positions and velocities of celestial bodies over time intervals, enabling efficient interpolation of numerical ephemerides.

Forces and dynamics

Spherical harmonics Expansion of the gravitational potential into degree and order terms, used to model non-spherical gravity fields of planets and small bodies.

Solid tides Deformation of a celestial body due to external gravity (e.g. Sun, Moon) causing additional perturbations in the gravity field.

Solar Radiation Pressure (SRP) Non-gravitational force generated by solar photons impacting spacecraft surfaces, modelled using plate areas, optical properties and Sun–spacecraft geometry.

Mean thrust model Simplified representation of thrust as an average acceleration over an orbital window, instead of resolving the detailed thrust profile.

Navigation and estimation concepts

State vector Vector containing the dynamical state of the spacecraft and possibly additional parameters (e.g. biases, clock states) to be estimated. In the MATLAB Navigator it is composed by position, velocity, clock bias, drag coefficient, SRP coefficient, thrust error.

Covariance matrix Symmetric positive-definite matrix representing the uncertainty of the state vector; propagated and updated in the EKF.

Process noise Random model error in the state dynamics, described by a process noise covariance matrix, accounting for unmodelled accelerations and uncertainties.

Measurement noise Random error affecting measurements, represented by a covariance matrix in the EKF measurement model.

Kalman filter Recursive estimation algorithm combining a dynamic model with noisy measurements to estimate the state and its uncertainty.

Extended Kalman Filter Nonlinear extension of the Kalman filter that linearises the dynamics and measurement models around the current estimate.

Transition matrix State transition matrix mapping state perturbations and covariance from one orbital window to the next.

Line of Sight (LOS) measurement Measurement of the direction from the spacecraft to a target (e.g. celestial body) expressed as a unit vector or angular quantities; used mainly to constrain tangential and normal components of the state.

LiDAR range measurement Distance measurement obtained from a LiDAR sensor by emitting laser pulses and measuring their two-way time of flight.

RMS error Root-mean-square error of position or velocity, computed over a set of samples to quantify navigation performance.

3σ **bounds** Confidence interval corresponding to three standard deviations of a Gaussian error, used to assess consistency between errors and covariance.

Guidance, optimal control and mission design

Guidance Problem of determining the desired trajectory or control law for the spacecraft to reach mission objectives.

Optimal Control Theory Theory that determines control laws for dynamical systems in order to optimise a performance index under constraints.

Optimal Control Problem Mathematical problem defined by dynamics, boundary conditions, constraints and an objective functional to be minimised or maximised.

State and costate Pair of vectors in optimal control: the state describes the physical system; costate (adjoint) variables describe sensitivities of the objective and satisfy adjoint equations.

Mayer-type cost functional Objective function depending only on the final state or final time (e.g. maximisation of final mass).

Mission generator Java-based tool (using Patrius and other CNES libraries) that propagates interplanetary missions, generates LOS and LiDAR measurements and produces MissionData, directives and telecommand files for the MATLAB Navigator.

Missions and celestial bodies

Hera mission ESA mission to the binary asteroid system Didymos–Dimorphos, designed to study the effects of a kinetic impact and demonstrate planetary defence techniques.

Didymos Near-Earth binary asteroid system and central body in several test cases of the thesis.

Dimorphos Small satellite of Didymos, target of the kinetic impact experiment.

Juventas CubeSat of the Hera mission with an atypical, manoeuvre-intensive trajectory, used in the thesis as a test case for navigation around a small body.

Milani CubeSat of the Hera mission dedicated to mapping and probing Dimorphos, an example of a mission combining camera and LiDAR for navigation.

2000 SG344 Small near-Earth asteroid with non-negligible impact probability on Earth in the next century; used as target of an interplanetary low-thrust trajectory optimisation in the thesis.