Alma Mater Studiorum
Università di Bologna

School of Engineering
Forlì Campus

Master's Degree in Aerospace Engineering

Class LM-20

Graduation Thesis in:
Spacecraft Orbit Determination and Control

Improvement of Jupiter's satellites ephemerides using stellar occultation observations

Candidate:
Francesca Andreoli

Supervisor:
Prof. Marco Zannoni

III session

Academic Year 2017/2018

Abstract

The inner Galilean moons orbiting Jupiter are locked into the so-called "Laplace resonance", where the orbital periods of Ganymede, Europa and Io maintain a 4:2:1 ratio respectively. Resonant dynamics appear several times across the Solar System and determining whether the Jovian resonance is deepening or loosening would provide a direct insight into the origin and the evolution of the Solar System.

At the moment, our knowledge of the Galilean moons' dynamics is not accurate enough to establish in which direction the system is evolving. Including stellar occultation observations obtained from an orbiting spacecraft could result in a critical improvement of these estimations. Since the spacecraft's orbit and the stars' position are generally very well known, every time one of the moons crosses the line-of-sight from the spacecraft to the star its position can be constrained very accurately.

The results of this Thesis lay the foundations for the introduction of stellar occultation observations in the Orbit Determination process of planetary satellites and other celestial bodies. First, a geometrical/mathematical model of the distance between the moon's limbs and a given star detected in its surroundings was developed and tested. Afterwards, a code to detect and archive all the stellar occultation events in a given time span was implemented. Finally, a parametric Covariance Analysis was performed to obtain a preliminary assessment of the improvements provided by the introduction of stellar occultation observations and investigate the influence of each variable on the Orbit Determination problem.


RADIO SCIENCE AND
PLANETARY EXPLORATION
- LABORATORY -

Contents

List of Figures

List of Tables

Chapter 1

Acronyms

ESA	European Space Agency
EME2000	Earth Mean Equator and Equinox of Epoch J2000 coordinate system
EMO2000	Earth Mean Orbit and Equinox of Epoch J2000 coordinate system
ET	Ephemeris Time
FOV	Field of view
JPL	Jet Propulsion Laboratory
JUICE	JUpiter ICy moon Explorer
LOS	Line Of Sight
MONTE	Mission-analysis, Operations, and Navigation Toolkit Environment
NASA	National Aeronautics and Space Administration
OD	Orbit Determination
RMS	Root Mean Square
RTN	Radial Tangential Normale reference frame
UVS	Ultraviolet Spectrograph

Chapter 2

Introduction

2.1 The Jovian system

Jupiter is the fifth planet in line from the Sun and it is twice as massive as all the other planets combined. Earth would fit eleven times across Jupiter's equator, however the gas giant's atmosphere is predominantly made up of very light elements, such as helium and hydrogen, and if it has a solid core at all, it is probably about the size of the Earth. Jupiter's familiar stripes and swirls are actually cold, windy clouds of ammonia and water, while Jupiter's iconic Great Red Spot is a giant storm bigger than Earth that has raged for hundreds of years.

Figure 2.1: Color-enhanced picture of Jupiter's southern hemisphere taken by NASA's Juno spacecraft, credits: [8].

The gas giant does not offer an hospitable environment for the evolution of life as we know it, but the same is not true for some of its many moons. Jupiter has 79 satellites orbiting around it, but scientists are particularly interested in

the so-called "Galilean moons": Io, Europa, Ganymede and Callisto. These four satellites were discovered by Galileo Galilei in 1610 and they are some of the most interesting destinations in our Solar System still today. Io is the most volcanically active body in our planetary system. Europa has been under the spotlight since evidence of the existence of a liquid ocean under its icy crust was collected from the Galileo mission and the Hubble Space Telescope. This discovery makes Europa one of the the most promising places to look for present-day environments suitable for life. Ganymede is the biggest moon in our Solar System, even bigger than planet Mercury. On top of this, the three inner Galilean moons' motion follow a very interesting periodic pattern, called "Laplace resonance", for which in the same time Ganymede completes one orbit around Jupiter, Europa and Io complete two and four orbits respectively [9].

2.2 The Laplace resonance

The first investigation of the Jovian system's resonance dates back to 1798, when the French mathematician and astronomer Pierre-Simon marquis de Laplace showed in the *Traité de Mécanique Céleste* that Ganymede, Europa and Io are in mean motion resonance with ratio $4 : 2 : 1$. Since then, the resonant interaction between the inner Galilean moons has been a research topic for many scientists and to these days it is still quite a controversial matter.

One of the key factors affecting the Jovian system's dynamics is the tidal interaction between the giant gaseous planet and its closest moon Io: the orbital energy dissipation due to the tides that Jupiter raises on Io modifies the semi-major axis of the moon and this in turns affects the orbits of Europa and Ganymede, due to their resonant interaction. So far, the researches on the contribution of this dissipation mechanism to the Galilean moons dynamics have led to very different results, which disagree both in order of magnitude and sign. This means that scientists are still discussing not only the scope of the tidal interaction, but also whether the moons are accelerating (i.e. moving toward Jupiter) or decelerating (i.e. drifting away from it).

An additional challenge related to the estimation of the Jovian system ephemerides is given by the extent and the diversity of the available data. The measurements to be included in the estimation are spread across a very long time span, from the 19th century to the present days, and come from many different sources, from Earth observations to spacecraft trackings. This means that the data accuracy varies considerably from one sample to the other and that all the secular forces need to be modeled, as even the smaller effect becomes relevant on such a wide period of time.

The Jovian system can be considered as a downsized model of the Solar System, so understanding how the Laplace resonance is currently evolving could shed light not only on the Galilean moons' interaction, but also on the dynamics of the whole Solar System and on those of other resonant bodies. Up to now, two main theories have been formulated: the first one supports the idea that the Galilean moons evolved into the Laplace resonance after their creation, so that the actual configuration is the arrival point of the moons' orbital evolution; the second theory affirms that the Galilean moons were already locked into the Laplace resonance

the moment they were formed. Thus, if the resonance is currently deepening, that would support the first theory; conversely, if the resonance is loosening, that would favor the primordial theory (without excluding the other one).

In general, the concept of orbital resonance applies to any physical system in which the ratio of two orbital periods is a rational number. Assuming that T_1, the orbital period of the first body, is bigger or equal to T_2, the orbital period of the second body, the mathematical definition of resonance is given by:

$$\frac{T_1}{T_2} = \frac{a}{b} = q \geq 1$$

Where a and b are coprime integers. However, from a physical point of view, this definition is quite weak as it may be argued that one can always find a rational number that approximates the orbital periods ratio. In order to avoid this ambiguity, we add the conditions that a and b have to be small enough and that the resonance relation must be maintained at least for some multiple of $max\{T_1, T_2\}$.

This kind of resonance is not uncommon in the Solar System, but it usually involves two bodies only, as in the case of Tethis and Mimas (4:2), Dione and Enceladus (2:1) or Pluto and Neptune (2:3) [11].

The Laplace resonance which binds the first three Galilean moons can be considered as a double orbital resonance in which the ratio between the three orbital periods is not only rational, but also small and integer. The inner Galilean moons have been the only known case of three-body resonance until the recent discovery of a similar interaction between Pluto's small moons [10]. Small ratios are particularly useful as they allow each configuration to repeat in a relatively small time period. Defining Io as body number 1, Europa as body number 2 and Ganymede as body number 3, the expressions for the angular velocity in anomaly can be expressed as:

$$\mu_1 = 4\omega$$
$$\mu_2 = 2\omega$$
$$\mu_3 = \omega$$

Where $\omega = 51{,}0571\ deg/day$. In terms of mean motion one can write:

$$n_1 - 2n_2 = \eta_1$$
$$n_2 - 2n_3 = \eta_2$$

and researchers noticed that $\eta_1 = \eta_2 = \eta = 0.7395\ deg/day$. Combining the above relations the classical Laplace resonance equation is obtained:

$$n_1 - 3n_2 + 2n_3 = 0 \tag{2.1}$$

Assuming that the Laplace resonance is maintained in time, so that equation 2.1 holds true, the resonance itself is said to be in equilibrium if $\dot{\eta} = 0$, to be loosening if $\dot{\eta} > 0$ or deepening if $\dot{\eta} < 0$ [12].

2.3 The JUICE mission

So far, nine spacecrafts have visited Jupiter and NASA's mission Juno is currently orbiting it. The next expeditions to the Jovian system will be ESA's mission JUICE and NASA's Europa Clipper. The JUpiter ICy moon Explorer was selected by ESA in May 2012 to be the first large mission within the Cosmic Vision Program 2015–2025. In particular, this mission addresses two of the key science themes of the Program: *"What are the conditions for planet formation and the emergence of life?"* and *"How does the Solar System work?"*.

JUICE is expected to launch from Kourou, French Guiana, in June 2022 onboard the Ariane 5 ECA Rocket. The spacecraft will use an Earth-Venus-Earth-Earth gravity assist strategy and it is expected to reach Jupiter in July 2030. After the orbit insertion maneuver, JUICE will perform a 2.5 year tour in the Jovian system focusing on continuous observations of Jupiter's atmosphere and magnetosphere. This phase of the mission will also include frequent flybys of Callisto, which will enable unique remote observations of the moon and in situ measurements in its vicinity, and two flybys of Europa, focusing on the composition of the non water-ice material and the first subsurface sounding of an icy moon. The mission will culminate in a dedicated eight months orbital tour around Ganymede during which the spacecraft will perform detailed investigation of the moon and its environment. At the end of the mission the spacecraft will impact the moon, either following the free evolution of the orbit for several weeks or constraining the location of the crash with a modest fuel expenditure, if required.

Thanks to the JUICE mission scientists will be able to characterize the potential habitable environments of Ganymede, Europa, and Callisto. JUICE will also provide a thorough investigation of the Jovian system, which serves as a miniature Solar System in its own right, and will thus contribute to a better understan ding of the origins of our planetary system and other exoplanetary systems [13].

Figure 2.2: JUICE spacecraft, credits: [14]

The only Galilean moon that JUICE will not be able to reach directly is the inner one, Io. This little satellite is extremely interesting for the scientific community due to its tidal interaction with Jupiter which plays a key role in the resonant dynamics of the Jovian system and makes Io the most volcanically active body in the Solar System.

The main idea underlying this Thesis is that, despite the fact that JUICE will never come close to Io, ancillary observations of the little moon could bring precious information about its motion. In particular, a new application to stellar occultation observations will be investigated in the following chapters.

A strong campaign of stellar occultation observations has already been scheduled throughout the JUICE mission, in particular during the first year orbiting Jupiter, at the approach and departure of Europa flybys and in the Ganymede Elliptical Orbit (GEO) phase. At the moment, the main objective of the campaign is to spot and investigate Europa's plumes, proving the moon's geological activity. The stellar occultations observation will be performed by JUICE-UVS, an Ultraviolet Spectrograph which strongly relies on the heritage of Juno-UVS and whose design and expected performance can be found in [15]. When performing stellar occultation observations, the Spectrograph instrumentation first scans the FOV and selects a target stars, usually basing on its magnitude and location. Once the target has been acquired, the Ultraviolet Spectrograph locks on the star and measures its radiation emission. A stellar occultation spectra is usually characterized by a rapid and momentary drop in the radiation intensity, as shown in Figure 2.3.

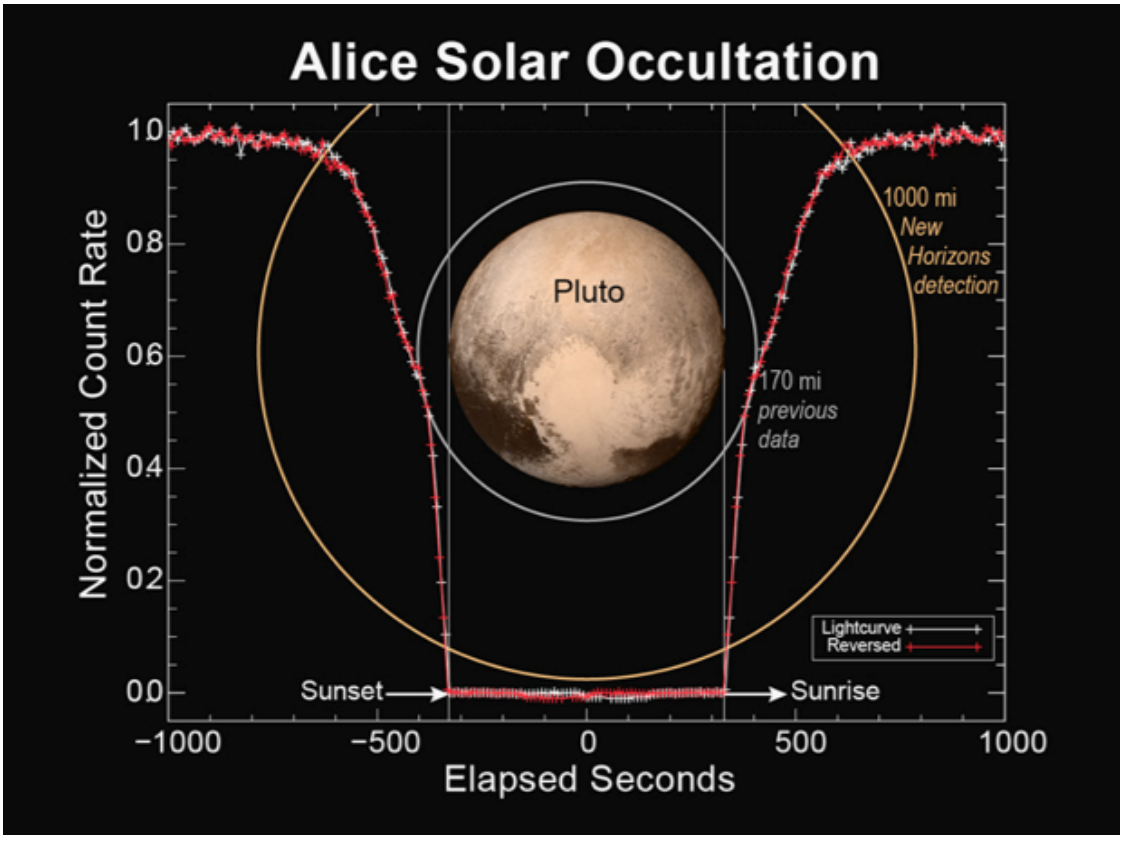

Figure 2.3: Solar occultation spectra acquired by the Alice instrument, credits: [26]

2.4 MONTE

MONTE (Mission Analysis, Operations, and Navigation Toolkit Environment) is the astrodynamic Python library developed by the Mission Design National Aeronautics and Space Administration and Navigation Section at the Jet Propulsion Laboratory, with sponsorship from NASA's Jet Propulsion Laboratory Multimission Ground System and Services (MGSS/AMMOS) program office. The library was built to support JPL's deep space exploration program and so far it has been used to fly different spacecrafts to Mars, Jupiter, Saturn, Ceres and many Solar System small bodies.

All the codes developed and implemented throughout this Thesis have been written using Python programming language. MONTE library was particularly useful as it provides all the basic astrodynamic infrastructure, such as trajectory models, coordinate frames, high precision time and astrodynamic event searches, and it can be used in conjunction with other Python scientific libraries to create customized applications [16].

Chapter 3

Stellar occultations model

3.1 Stellar occultations

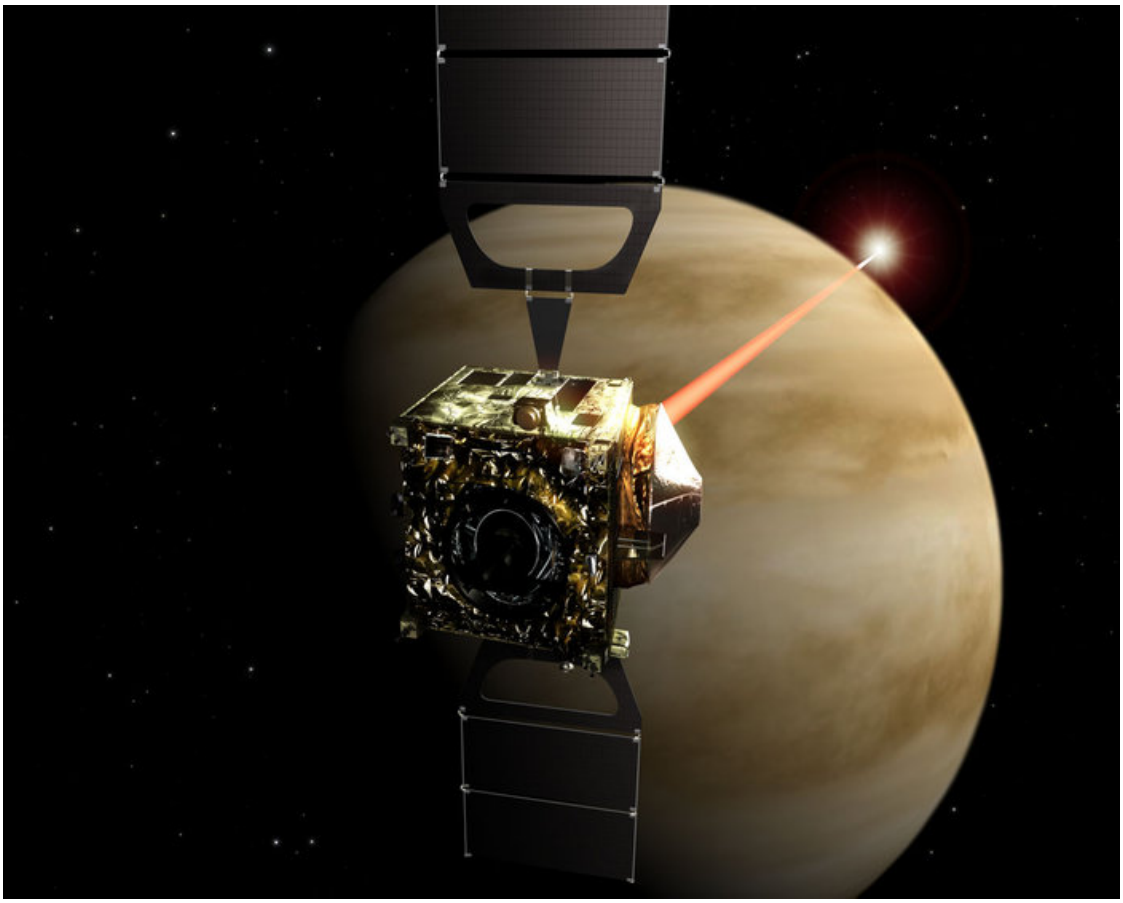

Figure 3.1: Venus Express performing a solar occultation at Venus, credits: [23]

In general, a stellar occultation can be defined as the event in which a third body occults a visible star as seen from the observer's point of view, as shown in Figure 3.1. In the past, this kind of observations have been exploited to discover and characterize exoplanets [6] [5], probe ring systems [2] [3] and investigate the atmosphere composition of distant celestial bodies [4]. The main advantage of

using stellar occultations in these studies is that the spatial resolution that can be reached with these observations is better than any other Earth-based method [24]. Regarding the orbit determination problem, stellar occultations have been used to develop autonomous navigation technologies which aim at increasing the spacecraft autonomy of operations in the proximity of a celestial body, such as in [7].

Throughout this work the term stellar occultation will refer to any configuration in which a star is occulted by the moon Io, as seen from the JUICE spacecraft. However, all the concepts and the models that will be presented hereafter can be easily applied to any observer-third body-star system.

For the purposes of this Thesis, a stellar occultation observation carries a critical information in order to determine the moon's position: since the inertial direction vector from the spacecraft to the star is generally very well known, every time that the moon's limbs cross it, the moon's position with respect to the spacecraft can be constrained very accurately. Of course, this kind of measurements is more effective in the transverse direction as one single occultation measurement brings no information as to the distance between the spacecraft and the moon. If one couples the beginning and the end of a stellar occultation, then a lower limit to the moon's projected dimensions, and thus an upper limit to the moon's distance from the spacecraft, can be set.

The stellar occultation observations can be performed by optical cameras, but more often photometric instruments are used. These measure the intensity of radiation of a given wavelength and in a specific direction. The same mathematical model applies to both kinds of instrumentation.

3.2 Geometrical model

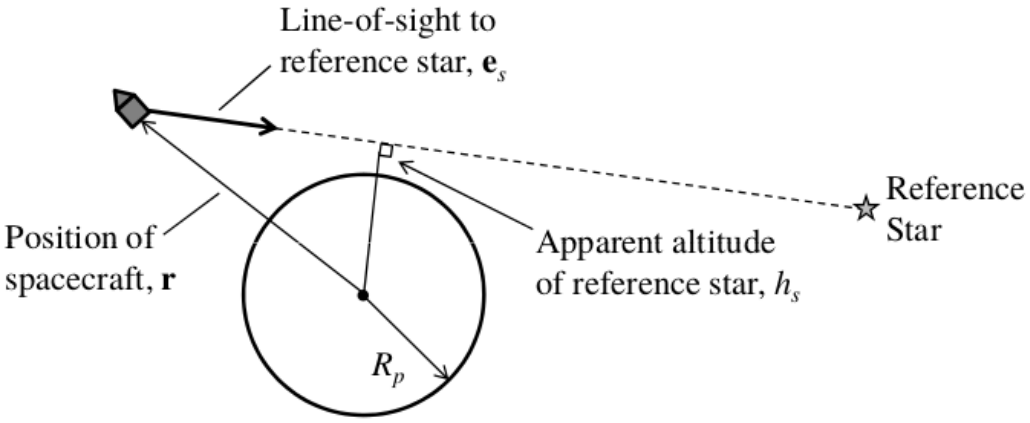

Figure 3.2: Stellar occultations geometrical model, credits: [25]

From a geometrical point of view, modeling a stellar occultation observation reduces to the computation of the 2D distance between an ellipse and a point. The ellipse is given by the projection of the moon's ellipsoid on the focal plane of the

spacecraft's camera and it depends on the satellite's shape, orientation and distance from the spacecraft. The point is given by the projection of the star direction on the focal plane and it only depends on its right ascension and declination since the stars are considered to be at infinite distance from the camera.

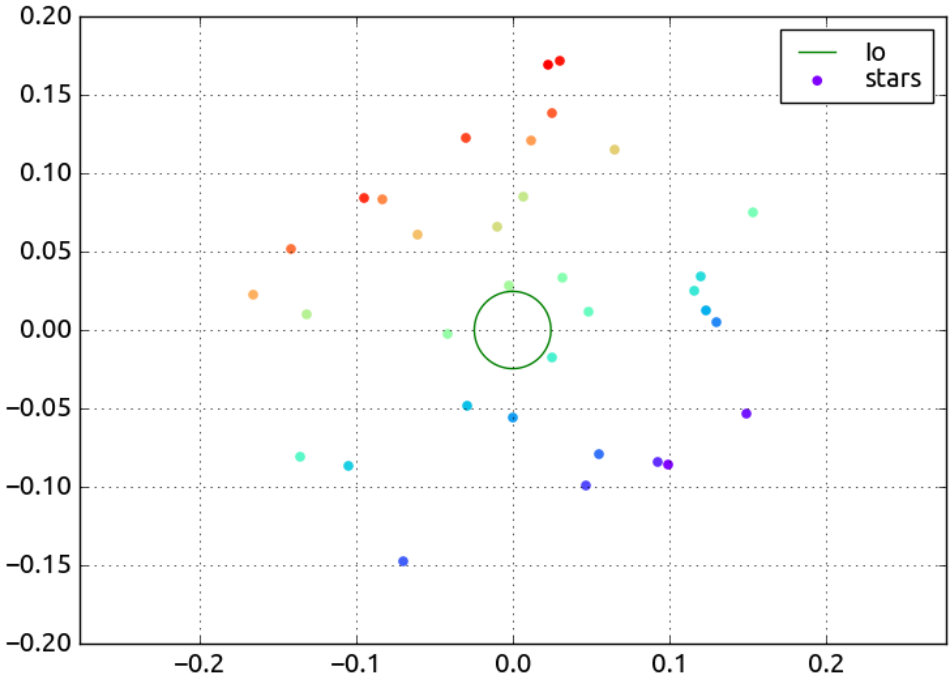

Figure 3.3: Io's ellipsoid and stars' direction projection on the focal plane

3.2.1 Ellipsoid projection

The algebraic process to obtain C, the projected ellipse matrix, was adapted from [22] and it starts from the ellipsoid matrix Q. This represents the equation of an ellipsoid in homogeneous coordinates, expressed in a reference frame which has the origin coincident with the center of the ellipsoid and the axes coincident with the principal axes of the ellipsoid (so that the frame is rotating with the celestial body):

$$Q = \begin{bmatrix} 1/a^2 & 0 & 0 & 0 \\ 0 & 1/b^2 & 0 & 0 \\ 0 & 0 & 1/c^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Where a, b and c are the principal semi-axes of the ellipsoid.

The first transformation applied to the ellipsoid matrix is a translation from the body-fixed frame to the camera frame. The translation vector $\vec{trl}$ is expressed in the body-fixed frame and it is directed from the camera to the observed body. This translation transformation can be expressed in homogeneous matrix form as:

$$T = \begin{bmatrix} 1 & 0 & 0 & trl[0] \\ 0 & 1 & 0 & trl[1] \\ 0 & 0 & 1 & trl[2] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $trl[0]$, $trl[1]$ and $trl[2]$ are the first, second and third component of the translation vector $\vec{trl}$ [1]. The translation is then applied as follows:

$$Q_t = T^{-T} Q T^{-1}$$

The rotation from the body-fixed frame to the camera frame was split into two rotations passing from EMO2000 so that the overall rotation transformation can be written as:

$$Q_c = R_{EC}^{-T} R_{BE}^{-T} Q_t R_{BE}^{-1} R_{EC}^{-1}$$

Where R_{EC} and R_{BE} are 4x4 matrices representing the rotation from EMO2000 to the camera frame and the rotation from the body-fixed frame to EMO2000 respectively, in homogeneous coordinates.

The matrix Q_c obtained after translation and rotation needs to be projected on the camera focal plane in order to obtain C. This is done multiplying for the intrinsic camera parameters matrix K:

$$K = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Where f is the focal length, which is defined as the distance from the lens to the principal foci of the lens. The projection transformation is performed as follows:

$$C^{-1} = K Q_c K^T$$

Here K is expressed as 3x4 matrix so that a 3x3 matrix (C^{-1}) is obtained from a 4x4 matrix (Q_c). Finally, the matrix has to be inverted to get the projected ellipse matrix C, in homogeneous coordinates.

3.2.2 Star direction projection

In order to obtain the star projection point on the camera focal plane, the star direction unit vector is normalized by its third component and projected using matrix K_s:

$$\vec{x} = K_s \vec{x_s} \frac{1}{z_s} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \frac{1}{z_s}$$

In this case, K_s is the intrinsic camera parameters matrix expressed as a 3x3 matrix, which is obtained omitting the fourth column of zeros in K.

[1] Python indexing system was adopted

3.2.3 2D distance

Initially, the geometric 2D distance between the moon's ellipse and the star's position point was considered. This approach allows the measurements to keep their physical meaning and maintain a direct connection to the real-life scenario. Although this was very useful to plot the stars' distribution around Io and have an intuitive validation of the model, the architecture of the problem resulted to be quite inconvenient in order to compute the partial derivatives at a later stage.

For this reason, the algebraic distance was adopted instead. In this way the direct relation to the physical distance is lost, but the expression to be implemented is much easier:

$$h = \vec{x}^T C \vec{x} \tag{3.1}$$

Where h is the ellipse-point distance, $\vec{x}$ is the star 2D position vector in homogeneous coordinates and C is the projected ellipse matrix. This approach relies on the simple fact that when a point lays on the ellipse limbs it has to satisfy equation $h = \vec{x}^T C \vec{x}$ and thus $h = 0$. Analogously, $h > 0$ when the star is not occulted (i.e. outside the ellipse area), and $h < 0$ when the star is occulted (i.e. inside the ellipse area). In this way, the beginning of a stellar occultation can be identified looking for the moment in which h turns from positive to negative, while the opposite is true for the end of the occultation. Figure 3.4 provides a visual confirmation of the distribution of h in the surroundings of the projected ellipse.

The integral code written to detect all the stellar occultations taking place in a given time span can be found in Appendix A.2.

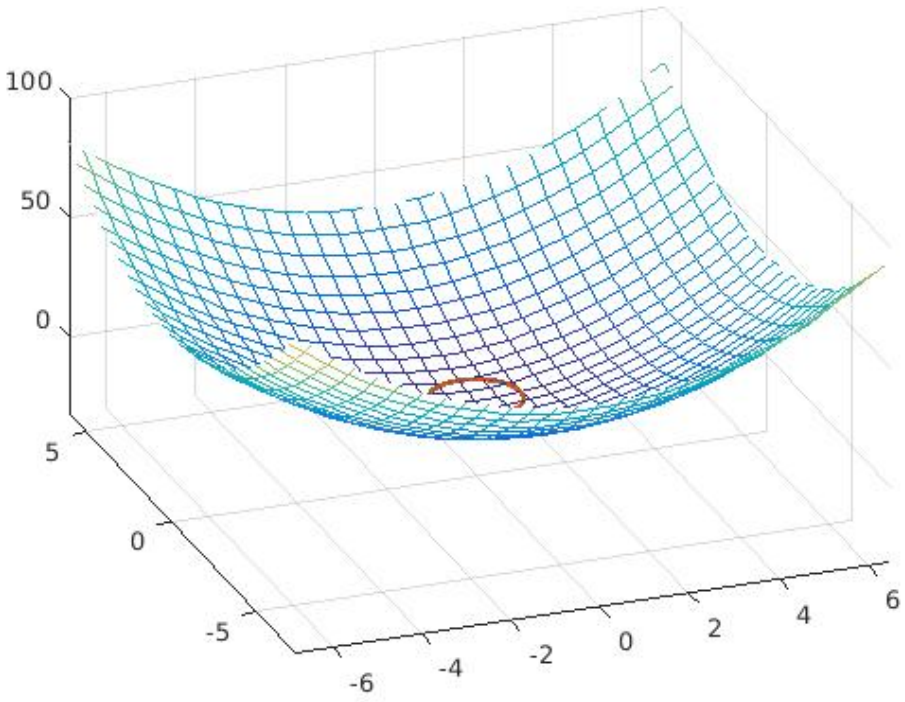

Figure 3.4: 3D behavior of h as a function of the distance from the projected ellipse (red line)

3.3 Model implementation

Hereafter the salient parts of the code developed to compute the 2D distance between Io and the surrounding stars will be briefly commented in order to show how the models and the techniques presented above have been applied in practice. All the referenced codes can be found in A.1.

One of the main simplifications adopted throughout this work is that the camera frame is always pointing towards Io. Indeed, the frame is defined by two direction vectors: one refers to the Z axis, which is always pointing from JUICE to Io, and the other defines the XY plane reference direction and points from JUICE to Jupiter. This simplification allows to detect all the potential occultations and implement the optical parameters at a later stage. In order to find a stellar occultation, the first thing to do is to investigate which stars can be found in the background around Io at the time of interest, as seen from JUICE. Thus a star catalog needs to be loaded and then used as a database containing all the useful information regarding a given set of stars, such as their position, magnitude, parallax and spectral type. For this work, the UCACT-PI star catalog was adopted. It results from the merge of UCAC2 and Tycho-2 catalogs, but with parallax information from Hipparcos and magnitudes corrections for Cassini's clear filter.

Using MONTE it is possible to obtain a list of all the stars which lay inside a given search circle centered around a given direction vector. In this case, the direction of interest was the one from JUICE to Io and the radius adopted for the search circle 10 *deg*. The minimum value for the stars' apparent magnitude was set to -2 since there are no brighter stars in the catalog anyway, while the maximum was set to 6 to ensure the stars were bright enough to be detected by UVS. In order to determine the distance between Io's limbs and each star around it, the moon's ellipsoid projection has to be computed following the procedure detailed in Section 3.2.1.

Finally, iterating over the number of stars detected in the surroundings of Io, the star direction projection is computed and the distance between the moon's ellipse and the star point is obtained.

Chapter 4

Partial derivatives

After having defined a suitable model for the stellar occultation events, one has to understand which are the main parameters affecting the chosen observable. In fact, this is the information conveyed by the partial derivatives: in which measure is h influenced by the variation of the each parameter? Some of these parameters are of interest and we want to estimate them, so the partials represent the sensitivity of the measurements to these parameters. Others are just an input, not to be estimated, and the partials represent the influence of an error in their knowledge on the measurements.

In this case, the first step consisted in identifying all the variables involved and then include the ones worth considering (i.e. which have a relevant influence on the observable). This was one of the most critical and complex parts of the work. Since the model for the observable h had been designed from scratch, there was no immediate reference as to the influence that other parameters have on it. After a preliminary selection, 11 parameters were included in the partial derivatives analysis:

- The star position, expressed as right ascension and declination;
- The shape of Io, expressed as the three semi-major axis defining the moon's ellipsoid;
- The 3-dimensional position of Io;
- The 3-dimensional position of JUICE;

Other parameters, such as the focal length f, the coordinate system rotations and the spacecraft's attitude, were first considered and then discarded due to their limited (or null) influence on the problem, as it will be explained in the following sections.

In order to validate the results of the calculations, both the numerical and the algebraic partial derivatives were computed and then compared. Numerical partials may be affected by numerical errors but they are more straightforward to obtain, so that they can be easily used as a first reference to double-check the algebraic ones.

4.1 Algebraic Partials

Table 4.1 contains the equations of the partial derivatives of h with respect to all the aforementioned parameters. To derive the expression of the algebraic partials, one has to go back and start from equation (3.1). Differentiating the equation of h with respect to a generic scalar parameter q leads to the following expression:

$$\frac{\partial h}{\partial q} = \frac{\partial \vec{x}^T}{\partial q} C\vec{x} + \vec{x}^T \frac{\partial C}{\partial q}\vec{x} + \vec{x}^T C \frac{\partial \vec{x}}{\partial q}$$

Since the first and the third term on the right hand side are scalars, they can be summed up to obtain:

$$\frac{\partial h}{\partial q} = 2\vec{x}^T C \frac{\partial \vec{x}}{\partial q} + \vec{x}^T \frac{\partial C}{\partial q}\vec{x} \tag{4.1}$$

As can be seen from equation (4.1), the right hand side is made up by two main components: the first one contains the partial derivative of the star 2D position on the focal plane x with respect to q, while the second contains the partial derivative of the projected ellipse matrix C with respect to q.

4.1.1 Ellipse matrix derivative

From section 3.2.1, one can retrieve the expression of matrix C as a function of Q_c, which becomes:

$$C = (C_{inv})^{-1} = (KQ_c^{-1}K^T)^{-1}$$

So that:

$$\frac{\partial C}{\partial q} = \frac{\partial (C_{inv})^{-1}}{\partial q} \tag{4.2}$$

Recalling the formulation for the matrix inverse derivative B.1, equation (4.2) can be expressed as:

$$\frac{\partial C}{\partial q} = \frac{\partial (C_{inv})^{-1}}{\partial q} = -(C_{inv})^{-1}\frac{\partial C_{inv}}{\partial q}(C_{inv})^{-1} =$$

$$= -C\frac{\partial (KQ_c^{-1}K^T)}{\partial q}C = -C(\frac{\partial K}{\partial q}Q_c^{-1}K^T + K\frac{\partial Q_c^{-1}}{\partial q}K^T + KQ_c^{-1}\frac{\partial K^T}{\partial q})C$$

And applying again (B.1) to Q_c in the middle term of the right hand side:

$$\frac{\partial C}{\partial q} = -C(\frac{\partial K}{\partial q}Q_c^{-1}K^T + KQ_c^{-1}\frac{\partial Q_c}{\partial q}Q_c^{-1}K^T + KQ_c^{-1}\frac{\partial K^T}{\partial q})C$$

So that the computation of the partial derivative of C reduces to the computation of the partial derivative of K and Q_c.

As shown in section 3.2.1, matrix K is a function of f only. It can be demonstrated that the focal length does not affect the observable h. In fact, both the projected ellipse and the star position point are scaled by f and its contribution cancels out when the terms are multiplied for each other in (4.1). This means that $\frac{\partial K}{\partial q} = 0$ and thus the first and the third term on the right hand side nullify.

The full expression for matrix Q_c can be retrieved from section 3.2.1 as follows:

$$Q_c = R_{EC}^{-T} R_{BE}^{-T} T^{-T} Q T^{-1} R_{BE}^{-1} R_{EC}^{-1} \tag{4.3}$$

Now, one has to evaluate the partial derivative of Q_c with respect to each one of the parameters. Looking at equation (4.3) and at the expression of the matrices R_{EC}, R_{BE} and T in section 3.2.1, it is clear that Q_c is not affected by the star's position coordinates. In fact, only a, b and c, the semi-major axis of Io's ellipsoid, and $trl[0]$, $trl[1]$ and $trl[2]$, the coordinates defining Io's position in the body-fixed frame with respect to the camera, appear in the equation.

For the first triplet of parameters we obtain:

$$\frac{\partial Q_c}{\partial q} = R_{EC}^{-T} R_{BE}^{-T} T^{-T} \frac{\partial Q}{\partial q} T^{-1} R_{BE}^{-1} R_{EC}^{-1}$$

Where $\frac{\partial Q}{\partial q}$ becomes for a, b and c respectively:

$$\frac{\partial Q}{\partial a} = \begin{bmatrix} -2/a^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial Q}{\partial b} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -2/b^3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial Q}{\partial c} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -2/c^3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The parameters defining Io's position appear in the translation matrix only, so that the partial derivative of Q_c becomes:

$$\frac{\partial Q_c}{\partial q} = R_{EC}^{-T} R_{BE}^{-T} \frac{\partial T^{-T}}{\partial q} Q T^{-1} R_{BE}^{-1} R_{EC}^{-1} + R_{EC}^{-T} R_{BE}^{-T} T^{-T} Q \frac{\partial T^{-1}}{\partial q} R_{BE}^{-1} R_{EC}^{-1}$$

And once again using equation (B.1) one can write:

$$\frac{\partial T^{-1}}{\partial q} = -T^{-1} \frac{\partial T}{\partial q} T^{-1}$$

Where $\frac{\partial T}{\partial q}$ for $x = trl[0]$, $y = trl[1]$ and $z = trl[2]$ can be written as:

$$\frac{\partial T}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial T}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial T}{\partial z} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The partials derivatives with respect to JUICE's position can be obtain simply changing the sign of the derivatives calculated for Io's translation. In fact, from a physical point of view, moving JUICE has the same and opposite effect on the translation vector as moving Io. Additionally, $\frac{\partial T}{\partial q}$ has to be rotated from Io body-fixed to EME2000 as JUICE's position is expressed in the inertial frame.

In the first place also the six angles defining the rotation from the body-fixed frame to EMO2000 and the rotation from EMO2000 to the camera frame were considered. However, their influence on the problem was quite limited and their observability was deemed small enough to neglect them without major consequences. Similarly, the attitude of the spacecraft was first considered and then discarded. In fact, the orientation of JUICE affects both the ellipsoid projection and the star direction projection in the same way, so that in the end its effect vanishes.

4.1.2 Star vector derivative

From section 3.2.1, one can retrieve the expression for $\vec{x}$ and differentiate it with respect to the generic scalar parameter q:

$$\frac{\partial \vec{x}}{\partial q} = \frac{\partial (K\vec{x_s}\frac{1}{z_s})}{\partial q} = \frac{\partial \frac{1}{z_s}}{\partial q} K\vec{x_s} + \frac{1}{z_s}\frac{\partial K}{\partial q}\vec{x_s} + \frac{1}{z_s}K\frac{\partial \vec{x_s}}{\partial q}$$

As explained in the previous paragraph, $\frac{\partial K}{\partial q} = 0$ since matrix K is a function of f only. If $\vec{x_s}$ is expressed as a function of the star's right ascension (Ra) and declination (Dec) as follows:

$$\vec{x_s} = \begin{bmatrix} cos(Ra)cos(Dec) \\ sin(Ra)cos(Dec) \\ sin(Dec) \end{bmatrix}$$

one can see that $\vec{x_s}$ is a function of Ra and Dec while $\frac{1}{z_s}$ is a function of Dec only. Thus, the derivative of $\vec{x}$ with respect to the right ascension becomes:

$$\frac{\partial \vec{x}}{\partial Ra} = \frac{1}{z_s} K \begin{bmatrix} -sin(Ra)cos(Dec) \\ cos(Ra)cos(Dec) \\ 0 \end{bmatrix}$$

While the derivative with respect to the star's declination can be written as:

$$\frac{\partial \vec{x}}{\partial Dec} = -\frac{cos(Dec)}{z_s^2} K\vec{x_s} + \frac{1}{z_s} K \begin{bmatrix} -cos(Ra)sin(Dec) \\ -sin(Ra)sin(Dec) \\ cos(Dec) \end{bmatrix}$$

Table 4.1: Algebraic partial derivatives recap

Parameter	Algebraic partial
a	$\frac{\partial h}{\partial a} = -\vec{x}^T C(KQ_c^{-1}R_{EC}^{-T}R_{BE}^{-T}T^{-T}\frac{\partial Q}{\partial a}T^{-1}R_{BE}^{-1}R_{EC}^{-1}Q_c^{-1}K^T)C\vec{x}$
b	$\frac{\partial h}{\partial b} = -\vec{x}^T C(KQ_c^{-1}R_{EC}^{-T}R_{BE}^{-T}T^{-T}\frac{\partial Q}{\partial b}T^{-1}R_{BE}^{-1}R_{EC}^{-1}Q_c^{-1}K^T)C\vec{x}$
c	$\frac{\partial h}{\partial c} = -\vec{x}^T C(KQ_c^{-1}R_{EC}^{-T}R_{BE}^{-T}T^{-T}\frac{\partial Q}{\partial c}T^{-1}R_{BE}^{-1}R_{EC}^{-1}Q_c^{-1}K^T)C\vec{x}$
x	$\frac{\partial h}{\partial x} = -\vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial x}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial x}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
y	$\frac{\partial h}{\partial y} = -\vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial y}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial y}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
z	$\frac{\partial h}{\partial z} = -\vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial z}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial z}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
x_{sc}	$\frac{\partial h}{\partial x} = \vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial xsc}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial xsc}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
y_{sc}	$\frac{\partial h}{\partial y} = \vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial ysc}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial ysc}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
z_{sc}	$\frac{\partial h}{\partial z} = \vec{x}^T C(R_{EC}^{-T}R_{BE}^{-T}(-T^{-1}\frac{\partial T}{\partial zsc}T^{-1})^T QT^{-1}R_{BE}^{-1}R_{EC}^{-1} + R_{EC}^{-T}R_{BE}^{-T}T^{-T}Q(-T^{-1}\frac{\partial T}{\partial zsc}T^{-1})R_{BE}^{-1}R_{EC}^{-1})C\vec{x}$
Ra	$\frac{\partial h}{\partial Ra} = 2\vec{x}^T C\frac{1}{z_s}K\frac{\partial \vec{x_c}}{\partial Ra}$
Dec	$\frac{\partial h}{\partial Dec} = -2\vec{x}^T C(\frac{cos(Dec)}{z_s^2}K\vec{x_s} + \frac{1}{z_s}K\frac{\partial \vec{x_c}}{\partial Dec})$

4.2 Numerical Partials

The numerical partials were obtained imposing a given variation of one parameter at the time and then applying the finite difference method to obtain the approximation of the partial derivative:

$$\frac{\partial h}{\partial q} \approx \frac{h(q + \Delta q) - h(q)}{\Delta q} \tag{4.4}$$

Appendix A.3 reports the code section which computes the partial derivative of h with respect to the first ellipsoid semi-major axis a as an example of the implementation scheme adopted. First, Δa has to be chosen and summed to the actual value of a, which is stored among the information regarding Io and its shape 'Io Ellipsoid'. Then C matrix is computed and given as an input to the function which returns the value of the perturbed observable h_{da} for all the stars in the surroundings of Io. Finally, the partials associated with each star are obtained using equation (4.4). The values delivered by this code where compared to the ones obtained using the algebraic equations detailed in section 4.1 at the same time and for the same set of stars. The same procedure was iterated for all the other parameters.

Figures 4.1, 4.2 and 4.3 show an example of visual validation of the partials derivatives for a single star on a 2-days time span. The relative error between algebraic and numerical partials was obtained as:

$$relative\ error = \frac{algebraic - numerical}{numerical} 100$$

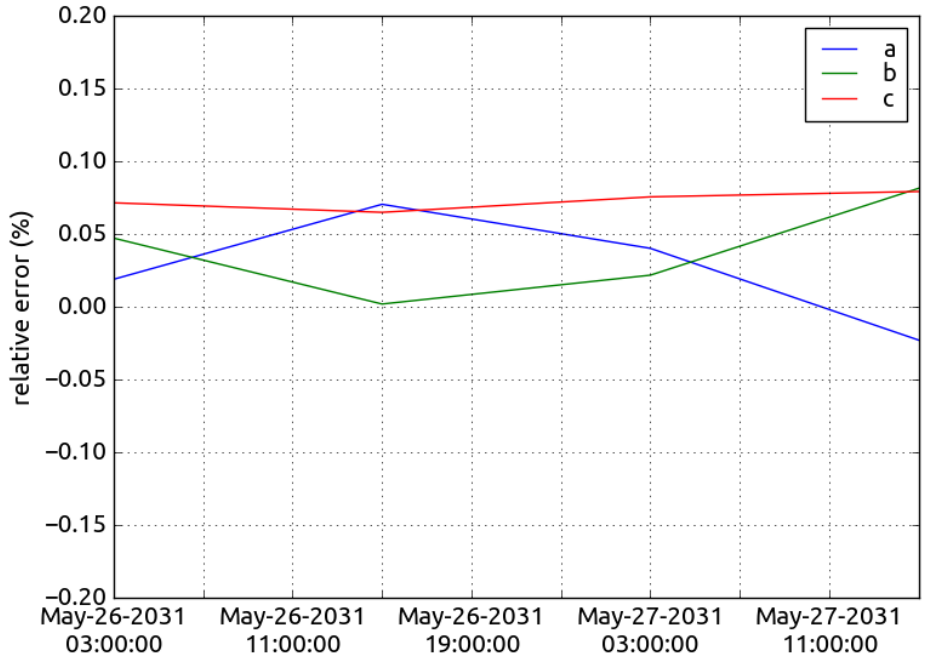

Figure 4.1: Time evolution of the relative error associated to Io's ellipsoid's semi-major axis a, b, c

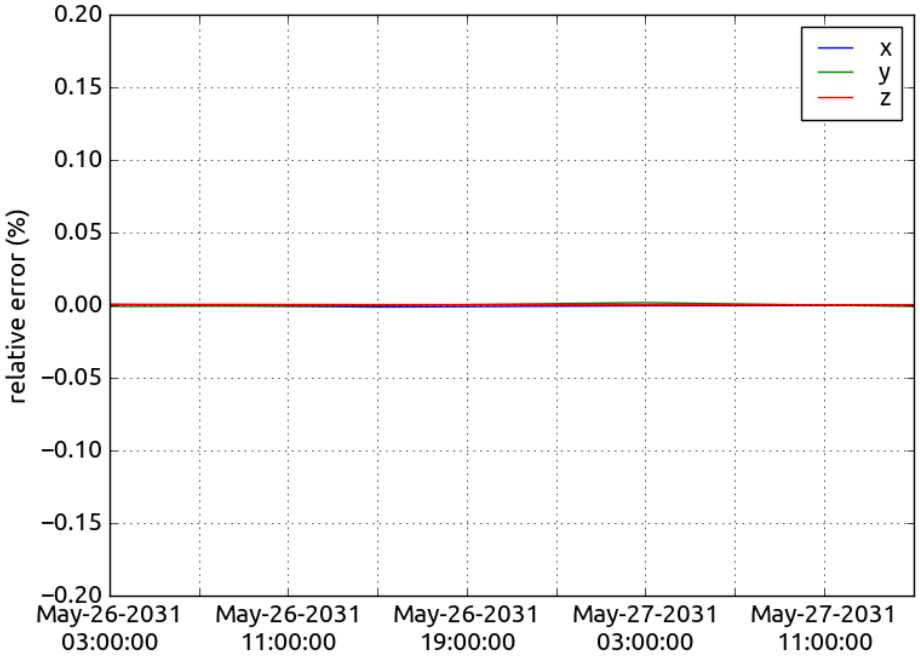

Figure 4.2: Time evolution of the relative error associated to the Io's position coordinates x, y, z

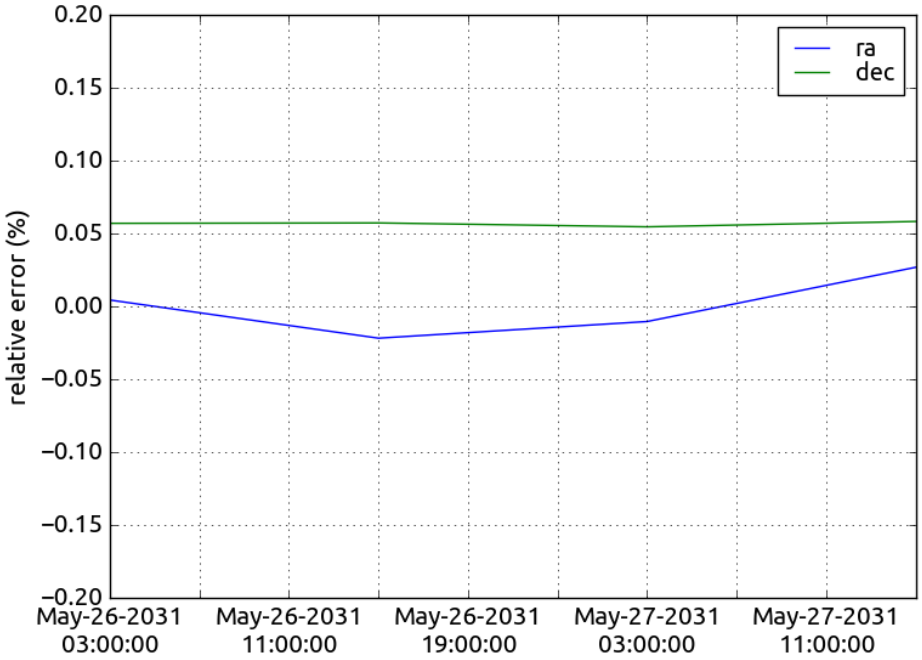

Figure 4.3: Time evolution of the relative error associated to the star's right ascension and declination

The validation was successfully completed as the magnitude of the relative error computed at different times and for different stars was found to be smaller than 0.2% for all the parameters of interest.

Chapter 5

Covariance Analysis

When confronting a new physical problem, one has to investigate which are the main parameters affecting the system and to what extent their misrepresentation can influence the outcome of the analysis. Generally, in a satellite orbit determination problem two different kinds of parameters can be identified: measurement parameters and dynamic parameters. The former define the relationship between the satellite at a given time and an observation of the satellite at that same time. The latter affect the time evolution of the state of the satellite [18]. For example, in the stellar occultation problem treated here the JUICE spacecraft's position is a measurement parameter, while the positions of the other Galilean moons are dynamic parameters. The standard deviation associated with one of the satellite's state components results from the uncertainties on both measurement and dynamic parameters. The critical point is to understand which are the fundamental sources of ambiguity so that one knows where to focus in order to improve the estimation of the unknown quantities.

When evaluating the relevance of a parameter's misrepresentation two factors have to be considered: what is the impact of the parameter's inaccuracy on the estimation of the state of the satellite and how likely the parameter is to be misrepresented. The usual approach in this case is to develop a model of the physical problem at hand and obtain an estimation of the satellite's state in nominal conditions. Subsequently, the same model is used to solve a new orbit determination problem in which the input value of the parameter being studied differs of what is believed to be the standard deviation of the parameter itself. The solutions delivered by the two settings are compared so that the time evolution of the effects of one sigma variation of the given parameter on the satellite's state is obtained. Of course, this process can be quite expensive from a computational point of view. If this is the case, the Covariance Analysis can be an effective alternative to obtain (almost) the same results.

The Covariance Analysis consists in computing the state covariance matrix only, without actually estimating the state of the satellite. Using this technique one can obtain the standard deviation of the components of the state, which is the square root of the diagonal terms of the covariance matrix, and investigate how the uncertainty on each parameter influences the estimated uncertainty on the satellite's state. However, the true error between the estimated state and the a priori state cannot be determined as the actual state estimation is not performed.

Usually, the true error is bigger than the standard deviation, so that the results of the Covariance Analysis tend to be optimistic and this is the main drawback related to this technique.

5.1 Variance, covariance and correlation

In general, the covariance matrix P associated to n estimated variables is a n-dimensional squared symmetric matrix whose structure can be schematized as follows:

$$P = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1i} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2i} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \sigma_{1i} & \sigma_{2i} & \cdots & \sigma_i^2 & \cdots & \sigma_{in} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{in} & \cdots & \sigma_n^2 \end{bmatrix}$$

The elements on the diagonal correspond to the variance of each variable, which is the square of the standard deviation and for the generic i-th random variable X it is defined as:

$$\sigma_i^2 = var(X) = E[(X - E[X])^2] = cov(X, X)$$

Where E is the expectation operator and $E[X]$ is the expected value of the variable X. The off-diagonal elements represent the covariance of the variables, which is a measure of their joint variability and for the generic i-th variable X and j-th variable Y it is defined as:

$$\sigma_{ij} = cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

So the covariance matrix includes information regarding both the uncertainty associated to each variable and how each one is correlated with the others. In fact, one can also compute the correlation coefficient, which is a normalized form of the covariance and it can be easily extracted from matrix P as follows:

$$\rho_{ij} = corr(X, Y) = \frac{cov(X, Y)}{\sigma_i \sigma_j} = \frac{E[(X - E[X])(Y - E[Y])]}{\sigma_i \sigma_j}$$

The correlation coefficient varies between 1 (i.e. perfect direct relationship between the variables, or correlation) and -1 (i.e. perfect inverse relationship between the variables, or anticorrelation). If the two variables considered are independent their correlation number is equal to zero and they are said to be "uncorrelated".

In order to have a visual and immediate indication of how the variables are influenced by each other, the correlation matrix can be computed and plotted. This appears as a large squared symmetric table where the color of the cells varies from white (no correlation, $\rho_{ij} = 0$) to black (full correlation, $|\rho_{ij}| = 1$) depending on the relationship between the corresponding variables. The correlation matrix for the stellar occultation problem treated here can be found in section **??**.

5.2 Covariance matrix derivation

The covariance matrix P is obtained using the following formulation:

$$P = (P_0^{-1} + A^T W A)^{-1} \tag{5.1}$$

Where P_0 is the *a priori* covariance matrix, which contains the information regarding the initial uncertainty of the parameters, A is the matrix containing the partial derivatives of the measured observable with respect to each one of the parameters and W is the weighting matrix. A qualitative analysis of this formula suggests that the covariance matrix relies both on the a priori knowledge of the system and on the contribution of the measurements. However, the measure in which these two kind of information are included in the calculation may vary. Indeed, if the initial uncertainty associated to the parameters is high, the contribution of P_0 will be lower since the a priori knowledge does not bear much information. Analogously, the function of the weighting matrix is to account for the accuracy and reliability of the measurements: if these are deemed to be a good source of information, their weights will be higher and their contribution more significant. On the other hand, if the uncertainty associated to measurements is too high, their weights will be lowered accordingly.

Hereafter the matrices appearing in (5.1) will be analyzed individually and their application to the stellar occultation problem will be illustrated.

5.2.1 Partial derivatives matrix

The function of matrix A is to bring into the estimation the new information associated to the measurements performed. In practice, if one has m measurements $\vec{z} = (z_1, z_2, ..., z_m)$ and n variables to be estimated $\vec{x} = (x_1, x_2, ..., x_n)$, A is a (m, n) matrix defined as:

$$A = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_n} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial x_n}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial x_1} & \frac{\partial z_m}{\partial x_2} & \cdots & \frac{\partial z_m}{\partial x_n} \end{bmatrix}$$

In the stellar occultation problem, the measurements correspond to the stellar occultations detected and selected throughout the JUICE mission. Originally, the distance h, modeled in section 3.2, was adopted as observable. However, this choice made the selection of the weighting coefficients of matrix W quite burdensome because of the non-direct physical meaning so in the end the observable was switched from the distance measurement h to the occultation time measurement t. The partial derivative associated to the new observable were easily obtained considering that when an occultation take place $h(t, \vec{x}) = 0$ and thus [1]:

$$\frac{dh}{d\vec{x}} = \frac{\partial h}{\partial \vec{x}} + \frac{\partial h}{\partial t}\frac{\partial t}{\partial \vec{x}} = 0$$

[1] here $\vec{x}$ is the vector containing all the variables involved in the estimation, not the 2D distance vector mentioned in Chapter 3 and Chapter 4

$$\frac{\partial t}{\partial \vec{x}} = -\left(\frac{\partial h}{\partial t}\right)^{-1} \frac{\partial h}{\partial \vec{x}} \tag{5.2}$$

Where the term on the left hand side corresponds to the time partials, the first derivative on the right hand side was computed numerically, as outlined in 4.2, and the second are the distance partials.

At this point, the variables to be included in the estimation had to be selected. After a few trials, the final choice included the parameters listed in Table 5.1. A part from the moons' state, all the other parameters have been estimated independently at each occultation event. This adjustment allows to take into account the uncertainty associated to each variable in a simpler but less conservative way with respect to the consider parameters technique.

Table 5.1: Parameters included in the covariance analysis [2]

Symbol	Description	Quantity
$\vec{x}_M$, $\vec{v}_M$	position and velocity of the four Galilean moons at the starting time t_0, expressed in the inertial frame EME2000	24
a, b, c	shape of Io, expressed as the three semi-major axis defining the moon's ellipsoid and estimated independently at each occultation	$3m$
Ra, Dec	star position, expressed as right ascension and declination in the inertial frame EME2000 and estimated independently at each occultation	$2m$
$\vec{x_{sc}}$	JUICE position, expressed as the x, y and z coordinates in the inertial frame EME2000 and estimated independently at each occultation	$3m$

5.2.1.1 Building the partial derivatives matrix

As mentioned above, each row of matrix A corresponds to one occultation, or more precisely to the time in which the occultation takes place, t_i. The partials derivatives of t_i with respect to the parameters $a(t_i)$, $b(t_i)$, $c(t_i)$, $Ra(t_i)$, $Dec(t_i)$, $x_{sc}(t_i)$, $y_{sc}(t_i)$ and $z_{sc}(t_i)$ had already been computed as shown in section 4.1. The partials with respect to the state of the moons at t_0 was obtained using the chain rule as follows:

$$\frac{\partial t_i}{\partial \vec{x}_M(t_0)} = \frac{\partial t_i}{\partial \vec{x}_{IO}(t_i)} \frac{\partial \vec{x}_{IO}(t_i)}{\partial \vec{x}_M(t_0)} \tag{5.3}$$

$$\frac{\partial t_i}{\partial \vec{v}_M(t_0)} = \frac{\partial t_i}{\partial \vec{x}_{IO}(t_i)} \frac{\partial \vec{x}_{IO}(t_i)}{\partial \vec{v}_M(t_0)} \tag{5.4}$$

Where the subscript M indicates any of the moons Io, Europa, Ganymede or Callisto. Note that the i-th occultation time measurement is influenced directly by

[2] m is the number of stellar occultations considered

the position of Io, and in turn by the position and velocity of the other moons due to their gravitational influence on Io's orbital evolution. From table 4.1 one can get:

$$\frac{\partial h}{\partial \vec{x}_{IO}(t_i)} = \left[\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}, \frac{\partial h}{\partial z}\right] \tag{5.5}$$

And then equation (5.2) can be used to switch from h to t_i and obtain $\frac{\partial t_i}{\partial \vec{x}_{IO}(t_i)}$.

The terms $\frac{\partial \vec{x}_{IO}(t_i)}{\partial \vec{x}_M(t_0)}$ and $\frac{\partial \vec{x}_{IO}(t_i)}{\partial \vec{v}_M(t_0)}$ are usually referred to as "state transition matrix" and they describes how the position of Io at time t_i is influenced by the state of the Galilean moons (including Io itself) at the starting time t_0.

5.2.1.2 Implementing the partial derivatives matrix

The state transition matrix was computed as a part of the orbit determination problem by integration of the variational equations. In this work the partials were obtained from the satellites' partials ephemerides released by the JPL, thus assuming to use the same dynamical model. In particular, the state transition matrix was derived using MONTE's method `M.ParamList.transitionMatrix( Epoch t, ParamList q )`. To do so, Io's parameters of interest, $\vec{x}_{IO} = (x_{IO}, y_{IO}, z_{IO})$, were declared as a ParamList, `q_Io`, and the state of all the moons was defined as a separate ParamList, `q_moons`. All the elements in `q_Io` have to be represented in Io body-fixed frame, to be consistent with the reference frame of the partials in (5.5), while the parameters in `q_moons` must be defined in EME2000, so that the resulting partial derivatives to be included in A are expressed in the inertial frame. Asking for `q_Io.transitionMatrix( t_plot, q_moons )` returns the 3x24 transition matrix to be used in (5.3) and (5.4), selecting the terms relative to the moons' position or velocity respectively.

5.2.2 Weighting matrix

Matrix W was introduced in formulation 5.1 to account for the reliability of the measurements, weighting each of them accordingly. In general, if m measurements are available, W is a m-dimensional squared diagonal matrix. It is common practice to assume the measurements noise to be un-correlated (white) and use the reciprocals of the variance associated to each measurement as diagonal entries, such that W results to be:

$$W = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_n^2} \end{bmatrix}$$

In this way, the higher the variance (i.e. the uncertainty) associated to a measurement, the lower to corresponding weight and the smaller its contribution to the estimation process.

Having switched from the observable h to the time of the occultation t, the choice of the weights for the stellar occultation problem was quite straightforward.

In fact, the measurements timing accuracy is given by the characteristics of the UVS instrument and the on-board clock, so that W becomes:

$$W = \frac{1}{\sigma_{time}^2} I$$

Where I is the identity matrix.

The time accuracy for JUICE is $1ms$ at best so that, in order to be conservative, $\sigma_{time} = 5ms$ was chosen as nominal value.

5.2.3 A priori covariance matrix

The purpose of the a priori covariance matrix P_0 is to include the contribution of the current knowledge of the problem in the estimation. From a mathematical point of view, this is a way to obtain a reliable solution when the problem is not fully observable, such as when the measurements do not carry much information about some variables. This is the case, for example, the outer moons' state.

As the name suggests, P_0 has the same structure and characteristics as matrix P, so section 5.1 can be taken as a reference. In this case, P_0 was chosen to be a diagonal matrix, thus excluding any a priori correlation between different variables. If n is the number of parameters included in the estimation, P_0 is a n-dimensional square diagonal matrix whose structure simplifies to:

$$P_0 = \begin{bmatrix} \sigma_{01}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{02}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{0n}^2 \end{bmatrix}$$

As shown by equation 5.1, the a priori covariance matrix has to inverted in order to obtain P. In this way, the higher the initial uncertainty of one parameter, the lower the diagonal element and the smaller the contribution of the a priori knowledge associated to that parameter.

5.2.3.1 Choice of the a priori standard deviation

The choice of the a priori σ_0 values for the estimated parameters is a quite delicate one. In principle, one can retrieve this information from the literature. However, the material is not always easily accessible. Additionally, the numerical effect of the a priori knowledge must be taken into account as well as the choice of P_0 has a relevant influence on the numerical stability of the problem.

So after many researches and a few tuning investigations, the final a priori values were selected as reported in table 5.2.

Table 5.2: A priori standard deviations

Parameter	σ_0	Reference
Io initial position	$50km$	[1]
Europa initial position	$0.5km$	[1]
Ganymede initial position	$0.05km$	[1]
Callisto initial position	$0.5km$	[1]
Moons initial velocity	$0.0001km/s$	N/A
Io dimensions	$6.3km$	[20]
stars position	$10\mu as$	[19]
JUICE position	$0.1km$	[21]

According to [1], the current uncertainty on the Galilean moon's position is $5km$ in all the three radial, downtrack and out-of-plane directions. Yet, the a priori σ_0 on Io's position was set to $50km$ not to constrain the estimation.

On the other hand, the accuracy on the other three Galilean moons' position is likely to improve considerably. In fact, JUICE will orbit Ganymede for 8 months, gaining crucial information on the moons' dynamics, so that it can be safely assumed that the satellite's position uncertainty will decrease to $0.05km$ approximately. Analogously, since both JUICE and Europa Clipper will perform various flybys of Europa and Callisto, the uncertainty on the two moons' position was reduced to $0.5km$. Also, the values in [1] have been calculated propagating the moons' trajectory from 1990 to 2020, which is a considerably long period of time, so the uncertainty is likely to be a bit overestimated if applied to a mission which lasts 2.5 years.

Unfortunately no reliable reference was found for the a priori moons' velocity uncertainty so the value of 0.0001 km/s was chosen believing it to be a reasonable one.

The uncertainty on Io's dimensions is related to the unevenness of its surface rather than to the precision of the ellipsoid semi-axes definition. In fact, the time of the occultation is affected by the actual morphology of Io's surface, i.e. by the irregularity of its limbs as seen from the camera. Additionally, the uncertainty associated to the semi-axes of the ellipsoid shape is considerably lower than the average dimension of Io's geographical features, so this approach is conservative anyway. Since the average elevation of the moon's mountains is $6.3km$, this value was taken as a priori σ_0.

The stars' position accuracy adopted is the one recently obtained thank to the GAIA mission. As stated in [19], the uncertainty on the location of the brightest

stars (i.e. whose apparent magnitude is lower than 12) varies between 5 and 16 μas. Since in this work only the stellar occultations of stars whose apparent magnitude is lower than 6 mag were considered, the a priori σ_0 was taken to be equal to 10 μas, being conservative.

According to [21], the uncertainty associated to the JUICE spacecraft's position will be 10 m in terms of formal error. So to adopt a conservative value for the a priori formal error, 0.1 km was chosen.

5.3 Covariance matrix propagation

Once the covariance matrix P has been computed, one knows what is the estimated uncertainty associated to each variable at the reference time. However, hardly ever the investigation is restricted to a singular time instant and so a tool to propagate the estimation in time is needed. Once again, the state transition matrix is used, just as in section 5.2.1.1, but this time it is applied through matrix multiplication as follows:

$$P(t) = \Phi(t, t_0) P(t_0) \Phi(t, t_0)^T$$

Where $P(t_0)$ is the covariance matrix calculated at the reference time t_0, $\Phi(t, t_0)$ is the state transition matrix from t_0 to the current time t, and $P(t)$ is the covariance matrix associated to time t.

The state transition matrix was provided by MONTE using the `M.ParamList.transitionMatrix( Epoch t, ParamList q )` method, as explained in section 5.2.1.1. The only difference is that in this case the ParamList `q_Io` is defined in Io's RTN reference frame, so that the propagated covariance matrix is expressed in this frame and one does not need to manually rotate it. In fact, from a physical point of view, it is more meaningful to analize the position uncertainty in the radial, tangential and normal direction of the orbital reference frame.

Chapter 6

Results

Hereafter the results obtained from the Covariance Analysis will be discussed. The main aim of this investigation is to understand how the a priori knowledge of the estimated parameters and the selection of the stellar occultations affect the estimation of the position of the target moon, Io.

The analysis will start presenting the results of the Covariance Analysis for the nominal case, then a parametric analysis which highlights the influence of each parameter on the estimation will follow and in the end the effects of the number and distribution of the stellar occultation observations will be examined.

6.1 Nominal case analysis

For the nominal case 239 occultation events were considered, imposing a maximum frequency of one observation every three days and a star apparent magnitude between -2 and 6 *mag*. The occultations available on ESA's Cosmos website [23] were used as a loose reference for the scheduled number of stellar occultation events. As expected, the quantity of occultations detected by the code developed for this Thesis is considerably higher that the one foreseen by ESA, even excluding the observations disturbed by other celestial bodies. This is due to the fact that the simplified camera frame used here assumes that UVS is always pointing towards Io, thus detecting more occultations than the actual instrument would.

The time interval considered throughout the analysis coincides with JUICE's 2.5 years tour of the Jovian system (January 2030 - June 2033). The only difference is that instead of including also the very first orbital phase after the Jupiter Orbit Insertion (JOI), it was chosen to set the beginning of the investigation in April 2030 so to limit the maximum distance between Juice and Io and deal with more regular orbits, as can be seen in Figure 6.1. Looking at the graph, it is clear that, even if the occultations have been selected to be no closer that three days from each other, the distribution of the measurements is not uniform. In fact, in the last phase of the mission, when JUICE will be orbiting Ganymede, the occultations are denser with respect to the first period. A clear indication of this unevenness is that the time interval considered here goes from 20^{th} April 2030 to 26^{th} June 2033, so that the middle value falls on 22^{th} November 2031, while the $(239/2)^{\text{th}}$ occultation takes place on 27^{th} March 2032. This means that roughly the 62% of the occultations is

detected in the second half of the mission.

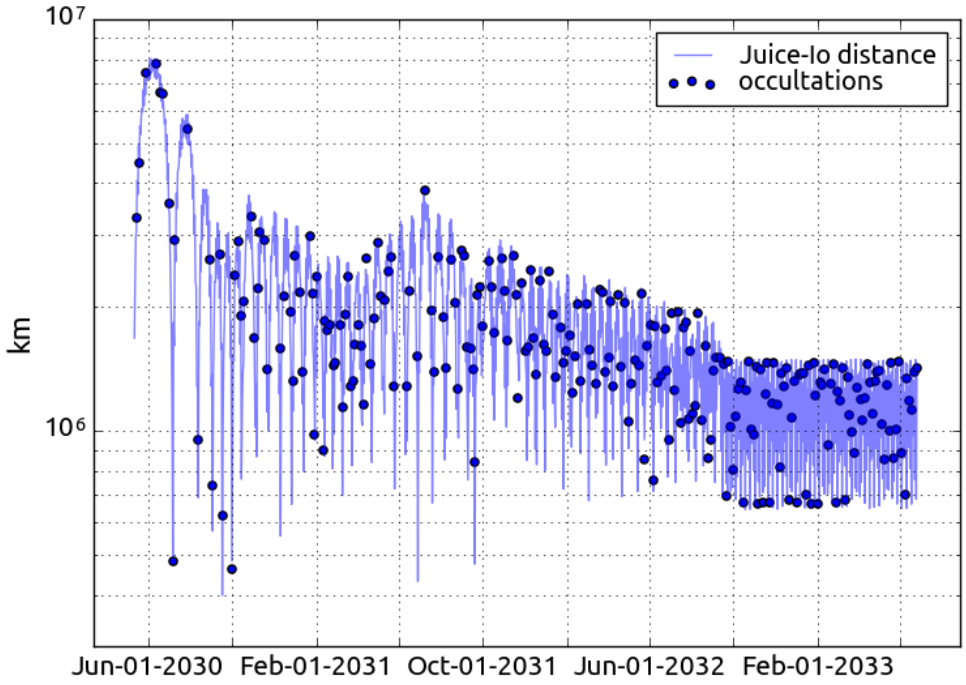

Figure 6.1: Occultations time distribution

6.1.1 Time evolution

To gain a first insight into the effectiveness of the estimation, one can analyze the time evolution of the standard deviation of the variable of interest, which in this case is the position of Io. In order to distinguish between the uncertainty associated to each position component, the standard deviation in the radial, normal and transverse direction were plotted separately.

To better appreciate the estimation improvement, the value of the ratio σ/σ_0 is plotted against time in Figure 6.4. This time a semi-logarithmic scale was chosen to better display the differences between the three components.

The previous plots are affected by visible periodic oscillations. To highlight the long-period variations, the moving average technique was applied. This basically consists in averaging a certain set of values and each time a new entry is added to the set, the oldest one is discarded. This allows to absorb many of the short-period oscillations while keeping the general trend of the variable evolution. In this case, the value of σ in each direction was computed 10 times each orbital period of Io and then averaged on the same time interval. As shown by Figures 6.3 and 6.5, this procedure allows to obtain a much "clearer" picture of the estimated standard deviation.

As can be seen from Figure 6.2 and 6.3, the time evolution of the uncertainty in the R, T and N directions differs from one component to the other. As predicted by the Euler-Hill equations, the oscillations in the R direction are bounded, while the T component shows an increasing trend moving away from the central region. This is due to the fact that the R and T components are coupled, so that the uncertainties in the R direction determine a variation in the transverse position that accumulates in time, leading to an increasingly higher error in this direction.

Looking at Figure 6.5, one can see that the uncertainty ratio in the R and T directions remains well below 10^{-2}, meaning that the estimated uncertainty can be considered to be independent from the initial condition. This means that choosing a different value for σ_{0R} or σ_{0T} would not affect the estimated standard deviation. On the other hand, the N component ratio oscillates just above 10^{-2}, so that the predicted value cannot be considered to be completely independent from the a priori uncertainty. This is probably due to the fact that a variation of Io's position in the N direction is not completely observable with the stellar occultation measurements.

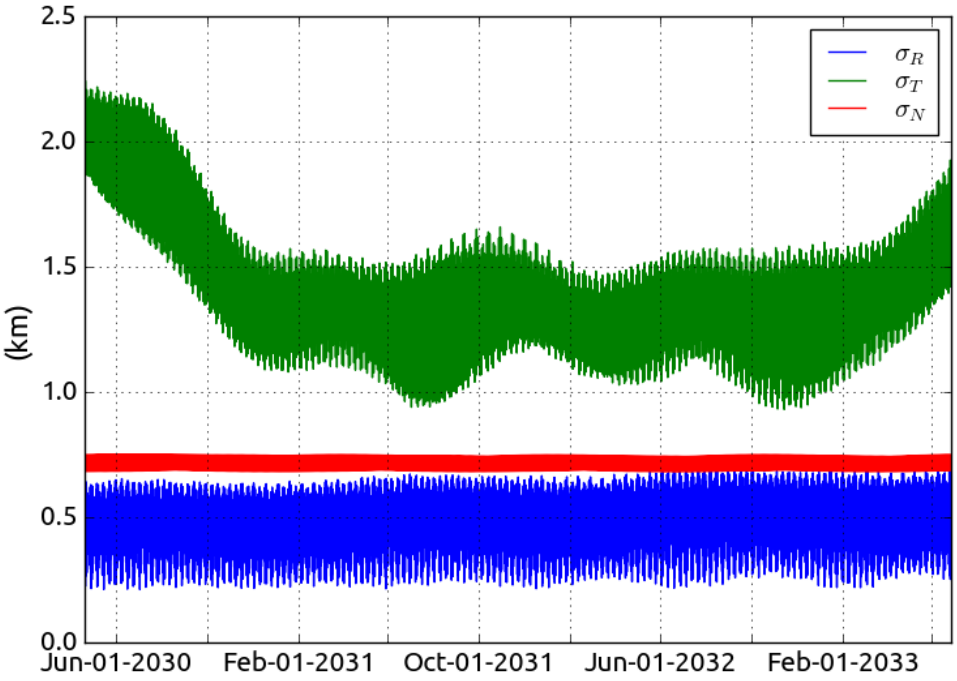

Figure 6.2: Time evolution of the formal uncertainty in the position of Io, in the radial, transverse and normal direction

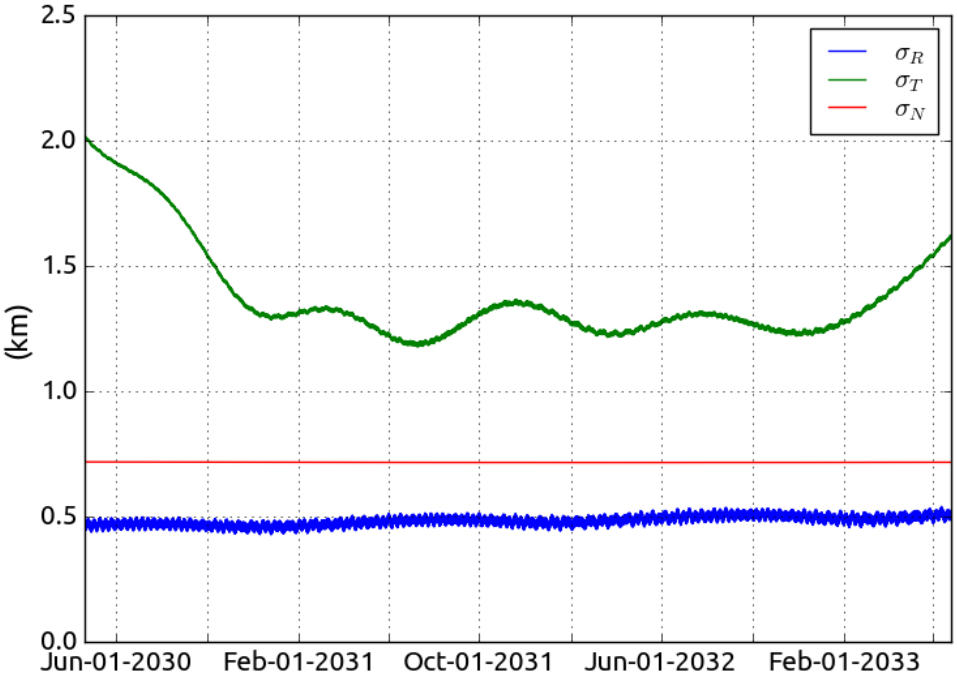

Figure 6.3: Long-period time evolution of the formal uncertainty in the position of Io, in the radial, transverse and normal direction

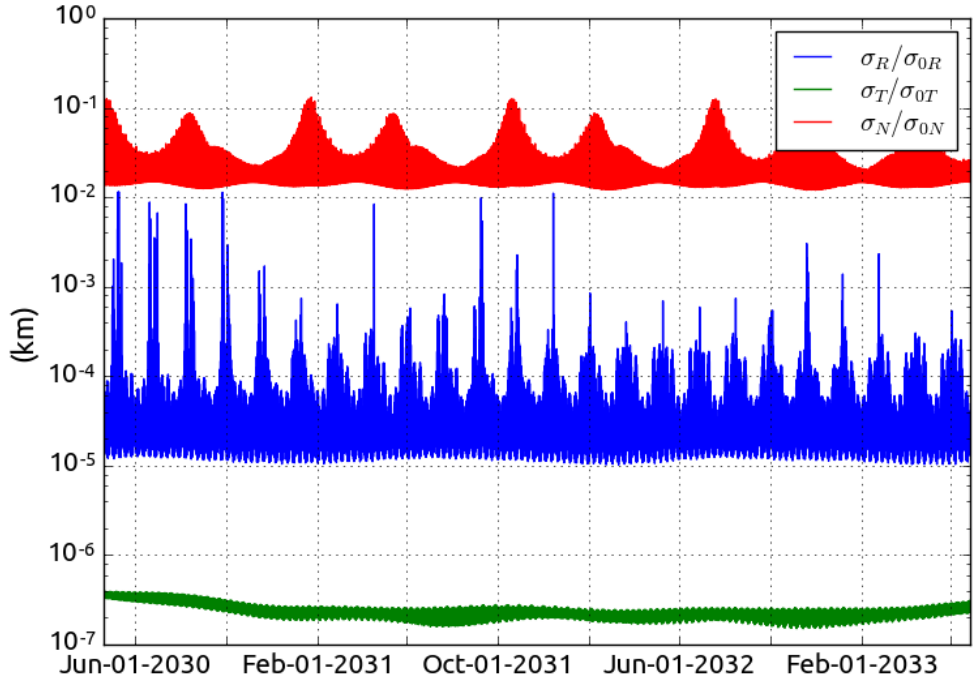

Figure 6.4: Time evolution of σ/σ_0 in the position of Io, in the radial, transverse and normal direction

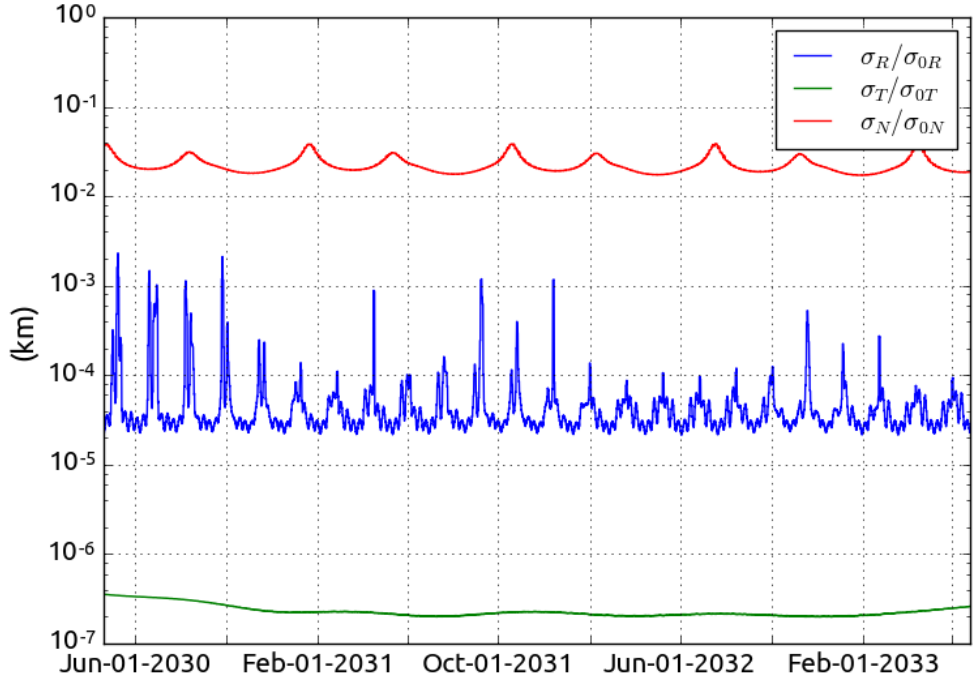

Figure 6.5: Long-period time evolution of σ/σ_0 in the position of Io, in the radial, transverse and normal direction

6.1.2 Correlation matrix

Figure 6.6 reports the correlation matrix associated to the four Galilean moons' state in correspondence of the 30$^{\text{th}}$ detected occultation, on 27$^{\text{th}}$ June 2031.

Looking at the section corresponding to Io's state, a strong correlation can be identified between the radial position of the moon and the tangential velocity. This should be expected as broadening or shrinking Io's orbits determines a change in the orbital speed. Also, the tangential position is correlated to the radial velocity since the orbital speed changes along the orbital trajectory.

The chart also shows a strong correlation between the in-plane components of the state of Io, Europa and Ganymede, due to the orbital resonance which binds them. Conversely, the interaction between Io and Callisto is weaker since the latter is not involved in the resonance. Also, the outer moons are very correlated with each other since they are not observed independently.

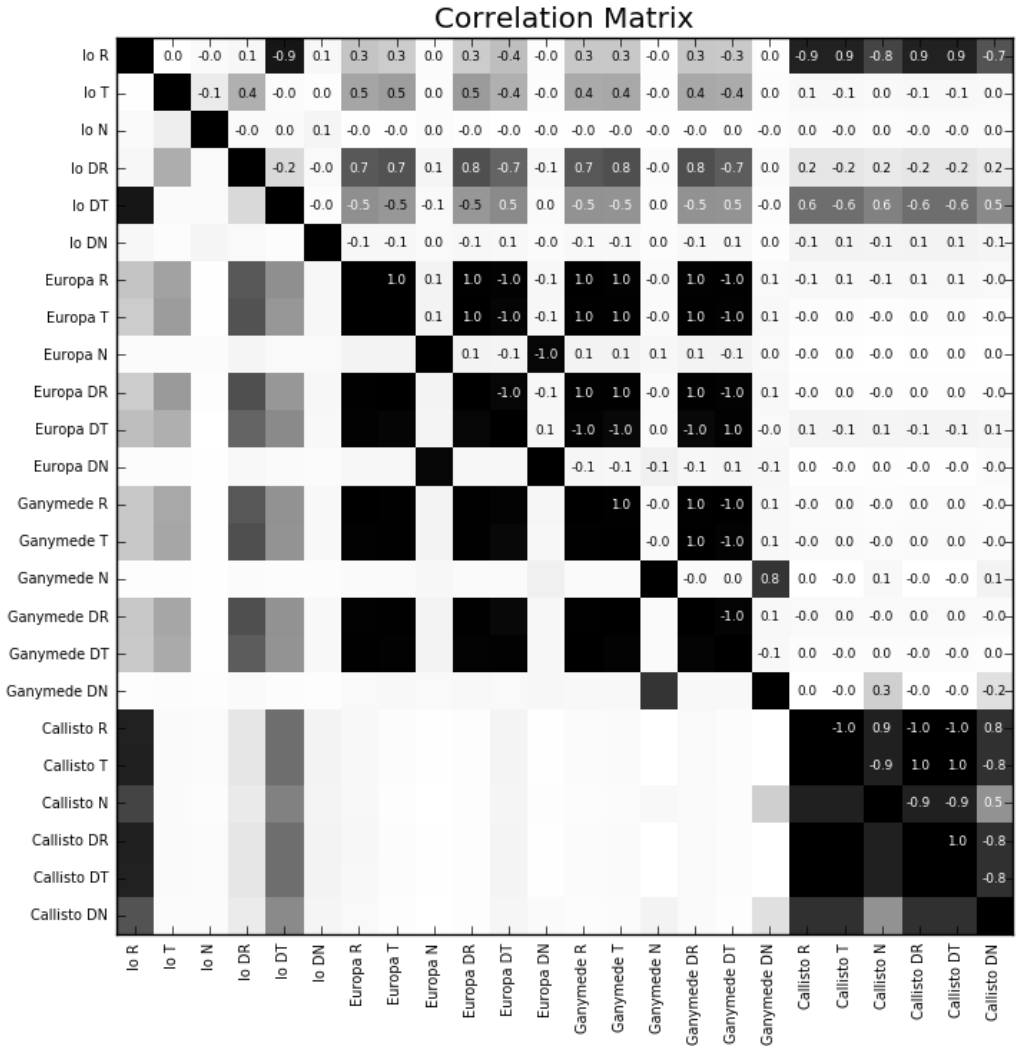

Figure 6.6: Correlation matrix relative to the moons' state at time 27-JUN-2031 20:20:48.3803 ET

6.2 Estimated uncertainty parametric analysis

A key point in the Covariance Analysis is to understand how and in which measure the different parameters affect the outcome of the estimation. In order to do so, the a priori uncertainty σ_0 associated to each parameter was iterated over a set of values distributed around the nominal one and the corresponding σ on Io's position was computed. Using this technique on each and one variable at the time allows to "isolate" the effect of that variable on the given problem.

To summarize the results of this analysis in one plot for each parameter, the time-averaged standard deviation on Io's position was considered. In practice, for each value of the parameter being studied, the covariance matrix at the reference time was computed and then propagated in time. The values of σ_R, σ_T and σ_N at each time instant were computed and then averaged on the whole time span.

6.2.1 Timing accuracy

The timing accuracy of JUICE-UVS enters in the covariance matrix formula 5.1 as weighting coefficient through matrix W, and thus from an intuitive point of view it represents the degree of reliability of the measurements. As can be seen from Figure 6.7, as long as the timing accuracy is kept below 10^{-2} its influence on the estimation is rather modest. However, as the time precision deteriorates, so does the effectiveness of our estimate. Still, one may argue that 5 s is a rather loose accuracy for a space-certified instrumentation and so overall the influence of this parameter can be considered quite limited. This plot also shows that the nominal value adopted in this work is a reasonable one and that a further refinement of the time accuracy would not result in a significant improvement of the estimation.

Since this parameter determines what is the influence of the measurements information in the estimation process, it also plays a role in ensuring the convergence of the problem. In fact, some numerical instabilities might arise when the measurements weights become too small. A first indication of this inconsistency can be spotted in the trend of σ_N on the far left end of the plot.

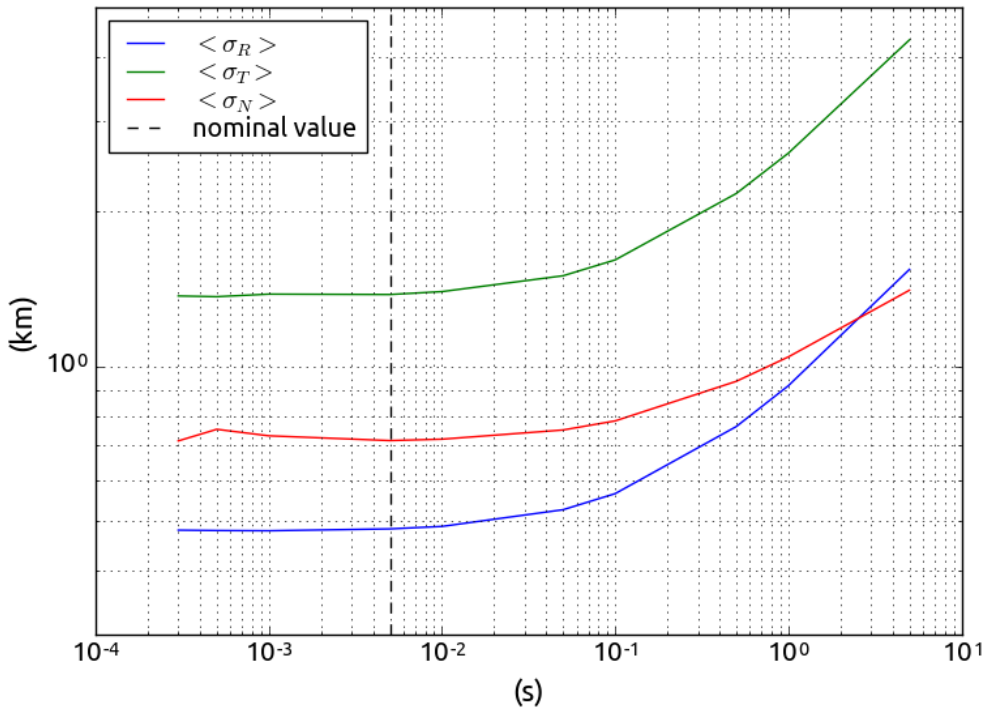

Figure 6.7: Standard deviation on Io's position as a function of the timing accuracy

6.2.2 Io position accuracy

Figure 6.8 shows how an error on the position of Io at the initial time t_0 can affect the prediction of its position at a later time.

As expected, the uncertainty associated to the R and T components remains almost unchanged despite of the variations in the a priori error. This is because, as explained in section 6.1.1, the radial and transverse components can be considered to be independent from the initial conditions. Conversely, the estimation of Io's position in the N direction deteriorates as the a priori uncertainty increases, confirming the dependence of the normal position component on the initial conditions. Still, the estimation error stabilizes below and above a certain value. In the first case, this is due to the fact that even if the position of the satellite is extremely well known, other uncertainty sources come into play and prevail on the beneficial effect of a tighter initial constraint. On the other hand, if the knowledge of the starting position is already very poor, a further degradation of the a priori error does not affect the estimated uncertainty.

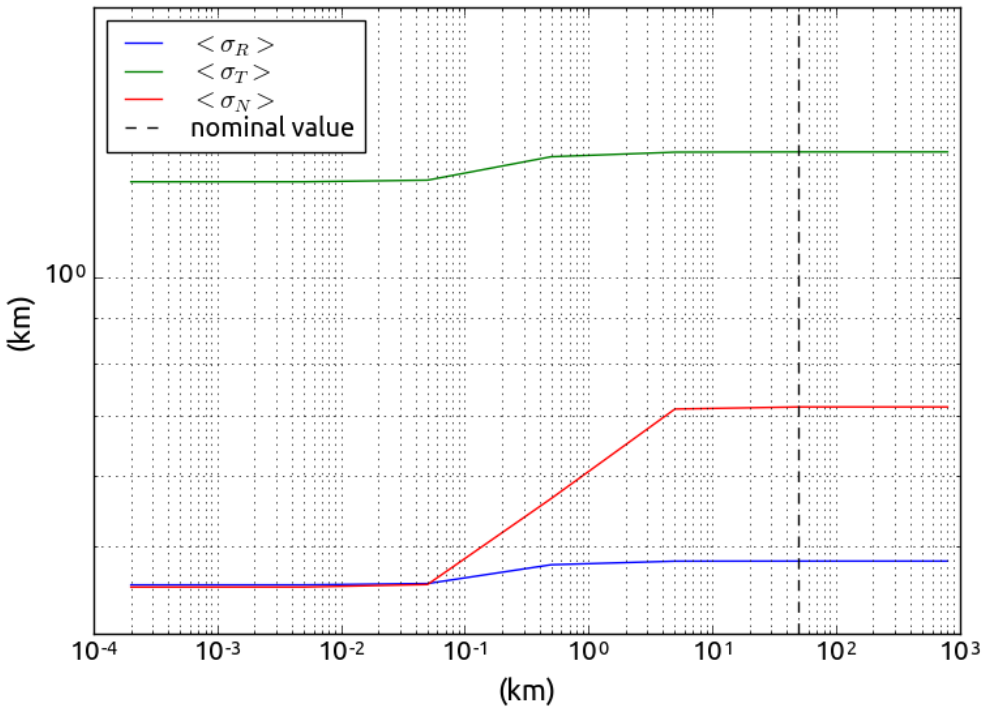

Figure 6.8: Standard deviation on Io's position as a function of the a priori uncertainty on Io's position at t_0

6.2.3 Other moons' position accuracy

Figure 6.9, 6.10 and 6.11 show how the initial uncertainty on the position of Europa, Ganymede and Callisto respectively can affect the quality of the prediction of Io's position in time. These three plots show a common trend which results in an increase of σ on Io as σ_0 on the moons' position deteriorates. Predictably, this behavior is more marked when an error on the position of Europa or Ganymede is considered. This is due to fact that the orbital resonance which binds the three inner Galilean moons amplifies the effect of an initial uncertainty on their position. Additionally, the closer the moon is to Io, the more significant its influence. In fact, it is reasonable that a variation on the position of Europa would affect Io more that a variation on the position of Callisto, which is considerably further away [1].

Once again the plots confirm that the values adopted for the a priori uncertainties on the moons' position are satisfactory, but while a variation of σ_0 on Ganymede's or Callisto's position has no effect for a wide interval of values, a degradation of Europa's position uncertainty would result in a slightly less accurate prediction of Io's position. However, overall these parameters can be considered to have a modest influence on the problem.

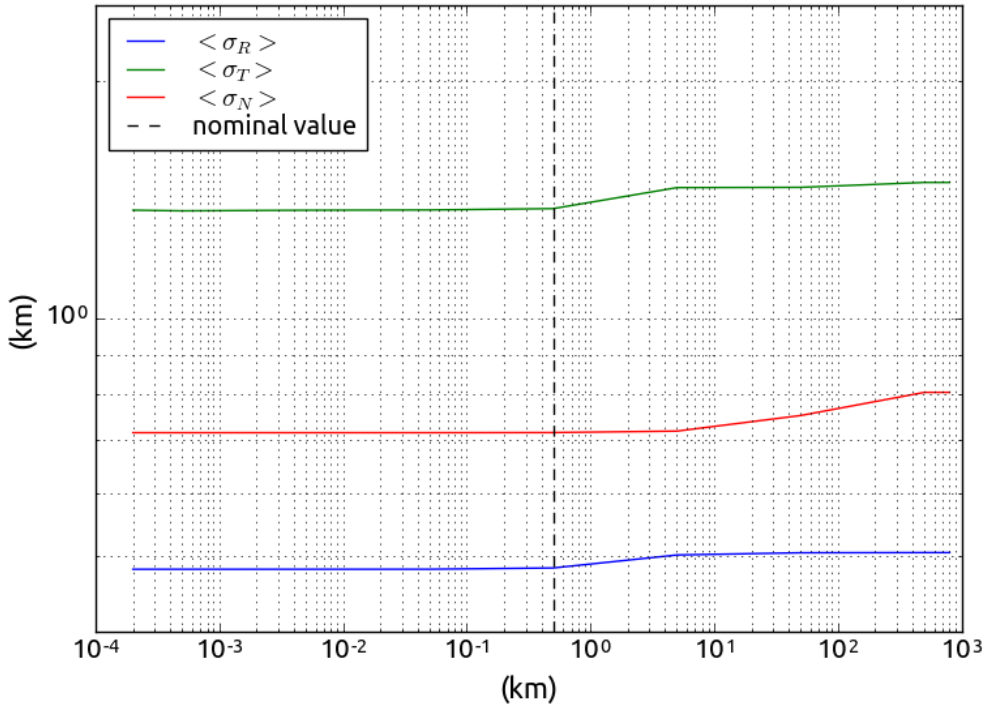

Figure 6.9: Standard deviation on Io's position as a function of the a priori uncertainty on Europa's position at t_0

[1] the distance between Io and Callisto is almost 6 times that between Io and Europa

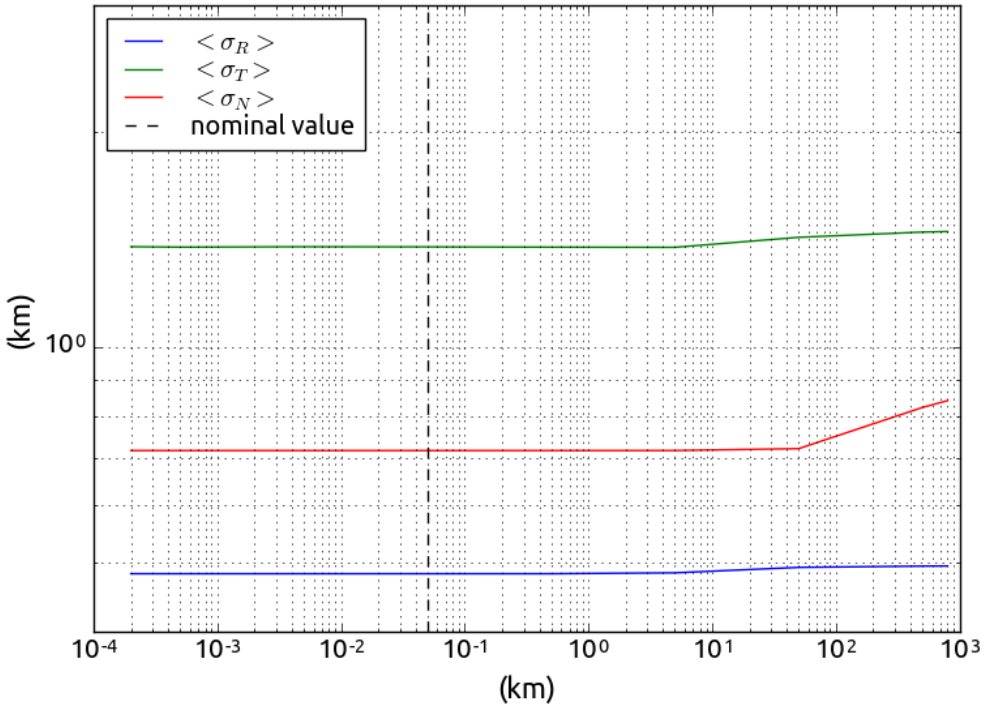

Figure 6.10: Standard deviation on Io's position as a function of the a priori uncertainty on Ganymede's position at t_0

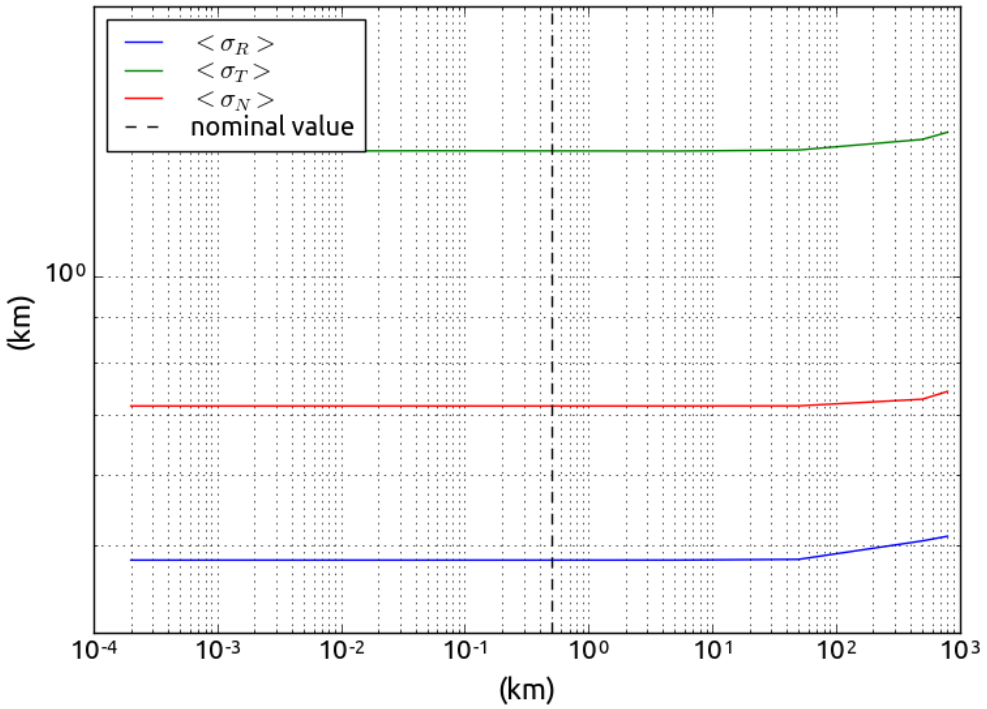

Figure 6.11: Standard deviation on Io's position as a function of the a priori uncertainty on Callisto's position at t_0

6.2.4 Moons' velocity accuracy

Considering the Galilean system as a whole it is clear that the position of Io is affected not only by the moons' location, but also by their velocity. Since the a priori uncertainty adopted for the velocity is the same for all the four moons, this was treated as a single variable. Thus, Figure 6.12 shows how the estimation of Io's position is influenced by the initial uncertainty on the velocity of the four Galilean moons. Once again, as expected, the higher the a priori error, the wider the resulting standard deviation. However, in this case the nominal value adopted is not the best possible so that the estimation could benefit from a further refinement of the initial constrain on the moons' velocity. Conversely, increasing the a priori uncertainty would result in a wider error on the predicted position of Io. Anyhow, the values of the standard deviation remain well below the a priori ones and considering that the graph represents the combined effect of the velocity uncertainty on all the four moons, its effect on the estimation is not dramatic.

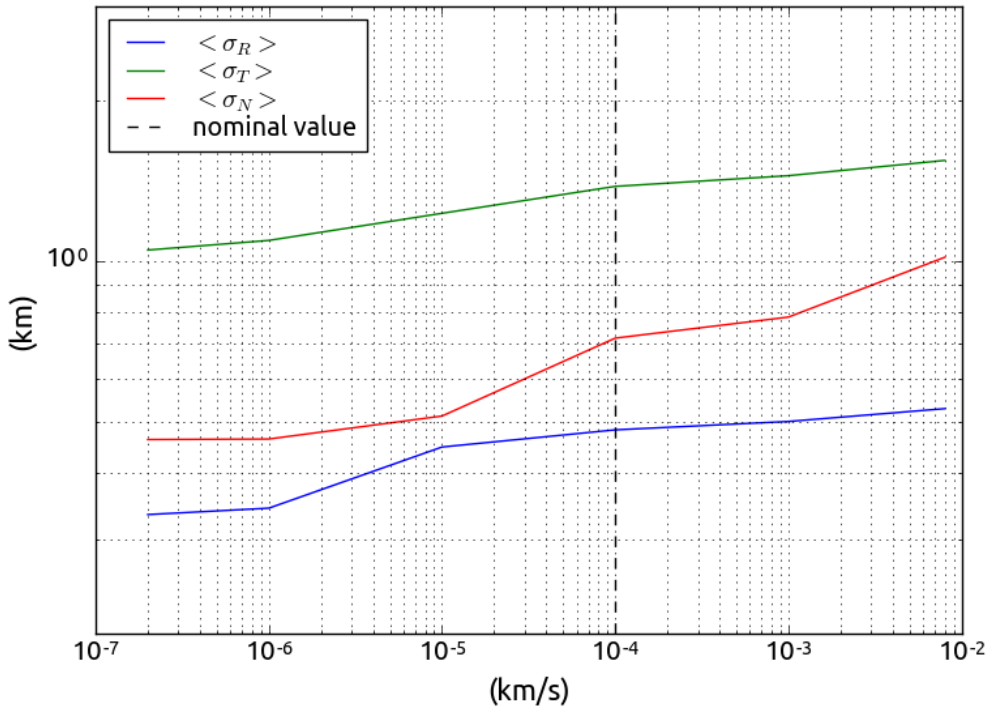

Figure 6.12: Standard deviation on Io's position as a function of the a priori uncertainty on the moons' velocity at t_0

6.2.5 Io's shape accuracy

Figure 6.13 shows how the estimation is affected by the uncertainties in the modeling of Io's shape. As explained in section 5.2.3.1, the unevenness of Io's surface affects the time of the stellar occultations and, lacking a detailed model of Io's elevations and depressions, this translates in an uncertainty on the measurements and thus on the position of Io. For the sake of simplicity, in this work the error associated to the moon's geographical characteristics was translated into an uncertainty in the value of Io's ellipsoid's semi-major axis. Although this is not an extremely rigorous approach, it is much simpler and conservative.

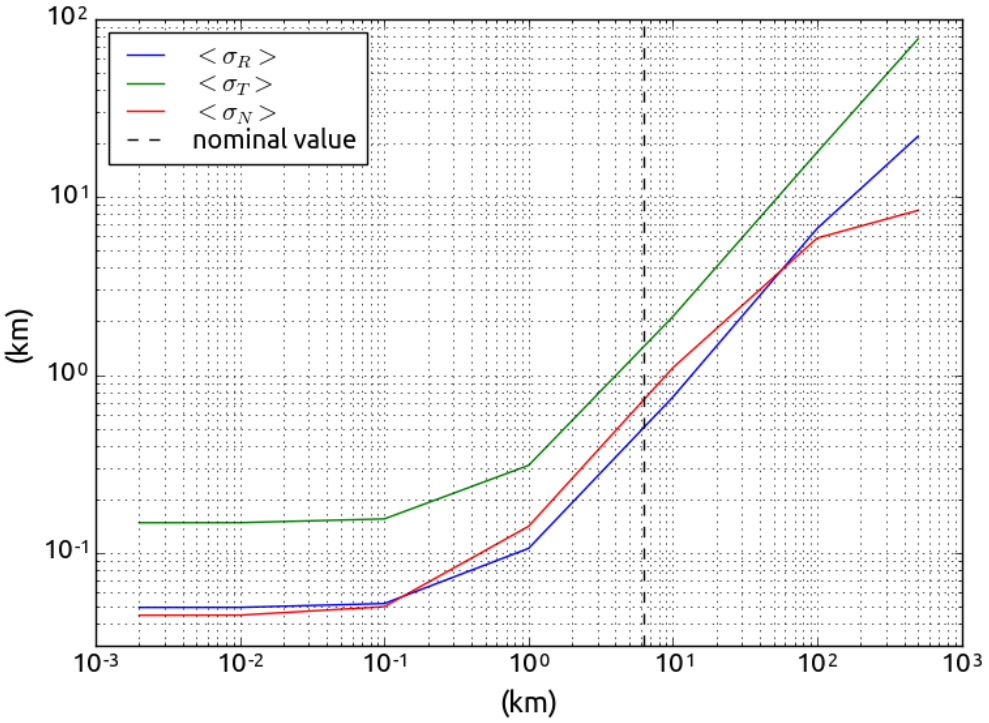

Figure 6.13: Standard deviation on Io's position as a function of the a priori uncertainty on the shape of Io

From the graph it is clear that the the knowledge of Io's morphology considerably influences the estimation. Indeed, one order of magnitude variation of the a priori uncertainty associated to Io's shape parameters results in a comparable variation in the predicted position of the moon. This is actually something quite foreseeable since the principle underlying the stellar occultation observations is to detect the exact time instant in which Io's limbs cross the LOS to the star. Thus if the limbs are highly irregular and the knowledge of the geographical characteristics is too poor, the occultation measurements lose their meaning and deliver very little information. On the other hand, the plot shows that the estimation still has considerable room for improvement. In fact, the nominal value adopted in this analysis is right in the middle of the steepest curve region, suggesting that even a modest improvement in the a priori knowledge of Io's shape could bring substantial

benefits to the estimation. Of course, in this case the a priori uncertainty cannot be reduced directly, as this would mean that Io's mountains have become smaller, but a detailed model of the moon's characteristics could be implemented so that the estimator would know where and when to expect an irregularity, adapting its prediction accordingly.

6.2.6 Spacecraft's position accuracy

An error on the position of the spacecraft affects the estimation of Io's location in a similar way as the moon's shape uncertainties do. In fact, JUICE's position modifies the LOS direction and if this is not constrained accurately enough the stellar occultation measurement is not able to provide sufficient information regarding the position of the moon. However, the current uncertainty on the spacecraft's location is better than the one associated to the moon's shape and indeed looking at Figure 6.14 one can see that the nominal value in this case lays in the asymptotic region. This means that the current knowledge of the spacecraft position is already good enough to obtain the best possible estimation.

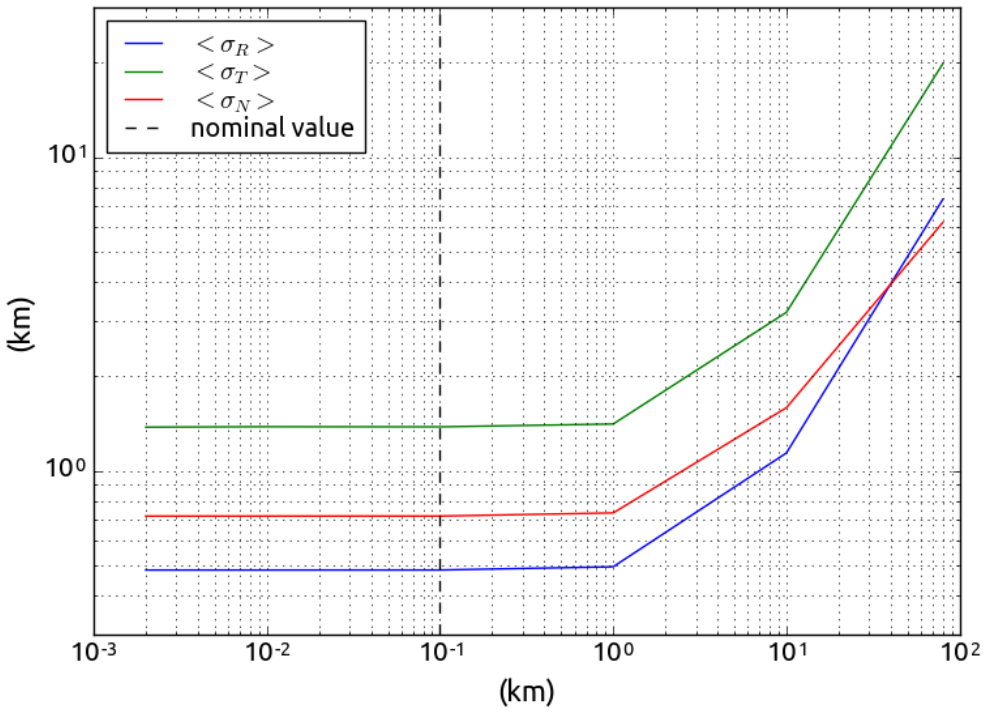

Figure 6.14: Standard deviation on Io's position as a function of the a priori uncertainty on the position of JUICE

6.2.7 Stars' position accuracy

The last variable which affects the direction of the LOS from JUICE to the stars is of course the stars' angular position. Luckily enough, the data stored in the star catalogs are usually very accurate from this point of view. As can be seen from Figure 6.15, the position accuracy reached by the GAIA mission's star catalog is definitely good enough with respect to the uncertainties level of this estimation. In fact, even deteriorating the a priori knowledge of a few orders of magnitude, does not affect the estimation considerably. The prediction worsens appreciably only when σ_0 becomes bigger than 10^{-6}. Considering that for the current star catalogs the stars' position precision hardly ever goes below 100 mas, which correspond to $4 * 10^{-6}$ rad, the accuracy of this variable can be considered satisfactory whatever the source of information.

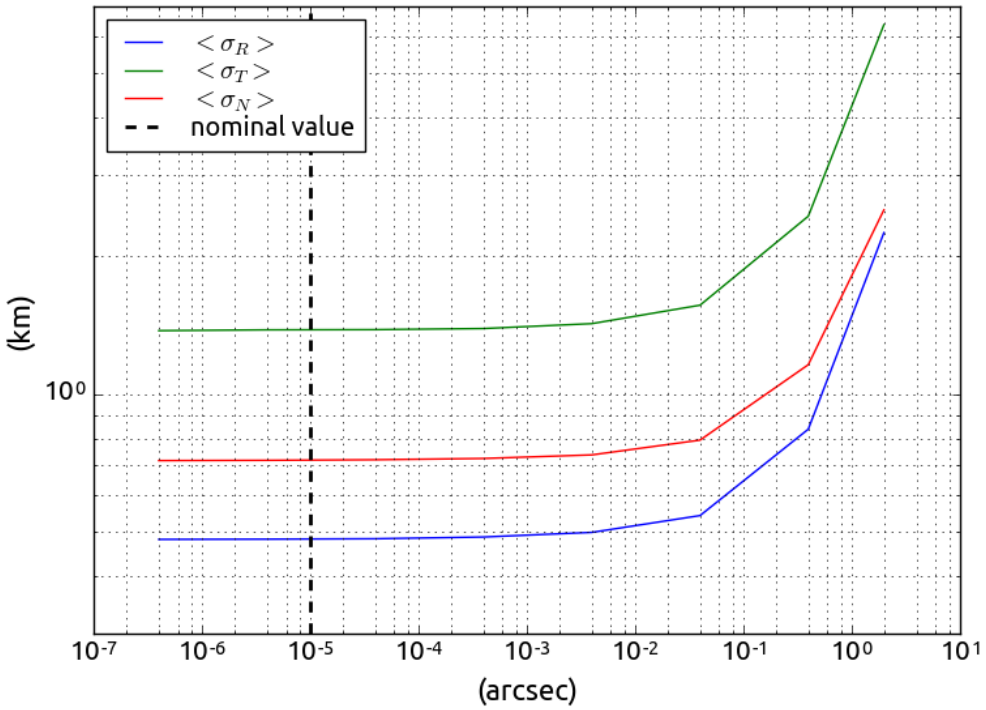

Figure 6.15: Standard deviation on Io's position as a function of the a priori uncertainty on the stars' position

6.3 Influence of stellar occultations selection

Since the estimation model presented in this Thesis solely relies on stellar occultation measurements, it is quite obvious that the number of occultation events considered will influence the accuracy of the estimation. Simply, the more the stellar occultations detected, the more the information supplied to the estimator and the more accurate the prediction. However, it is interesting to investigate how the choice of the selection criteria used to determine which observations to take and which to discard can affect the outcome of the estimation problem.

6.3.1 Number of occultations

As mentioned above, the error on Io's estimated position is expected to decrease the more occultation measurements are considered. This is exactly what happens in Figure 6.16, where the standard deviation on Io's position was plotted against the number of occultations included in the Covariance Analysis. The number of measurements considered N was increased requiring the occultations to be separated by and increasingly smaller time span. The trend of the σ curves displayed here is proportional to $\frac{1}{\sqrt{N}}$, as expected when adding new independent measurements to the estimation. Indeed, the estimation error is considerably higher when just a few occultations are considered and it diminishes with a progressively shallower inclination moving to the right, as more and more occultations are included.

This is a predictable behavior if one considers that if just a small number of measurements are supplied to the estimator, the information available is rather scattered, so that adding a few other observations brings in a lot of new data if compared to the ones already accessible, with considerable benefits on the estimation. Conversely, when a few *hundreds* measurements have already been included, a substantial volume of information has already been processed by the estimator so that adding a small number of occultations does not make much difference.

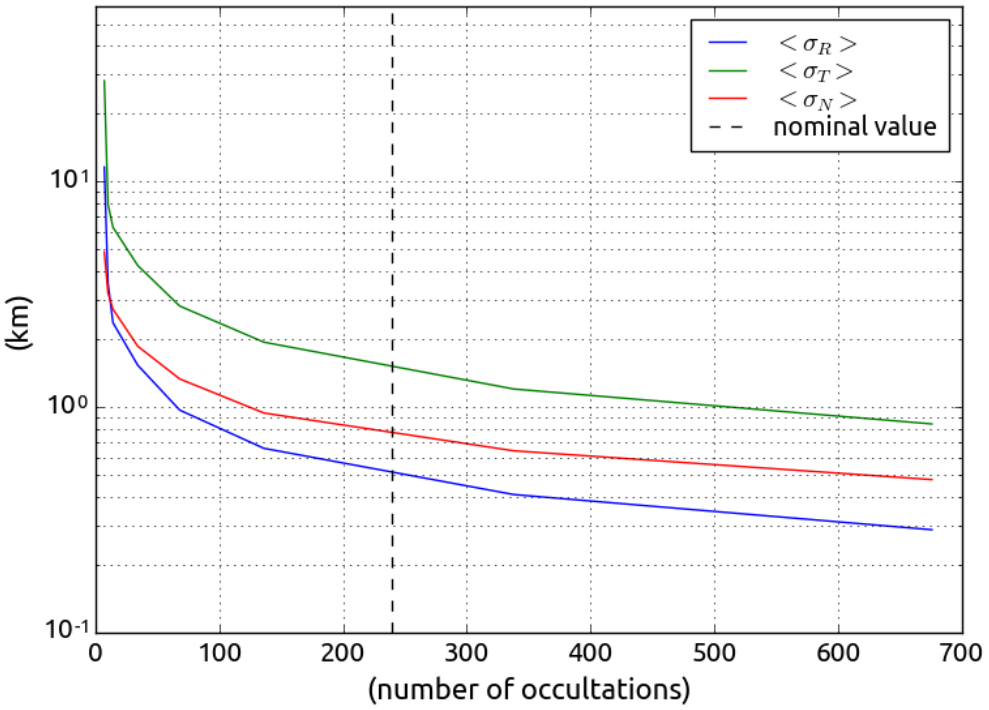

Figure 6.16: Standard deviation on Io's position as a function of the number of occultations

6.3.2 Occultations distribution

A part from the total number of occultations included in the estimation, the second most influential specification is how sparse the measurements are.

Knowing that the occultations are scattered along the time interval of interest as outlined in section 6.1, one can investigate the influence of the distribution of the observations considering the first and the second half of the occultations separately. To do so, the total number of occultations detected was increased to 676, imposing a frequency of 3 occultations per day. This is a necessary step since otherwise each set would contain only 119 occultations and, as can be seen in Figure 6.16, this is not a sufficient number of measurements to perform a sound analysis.

First, the occultations were halved so that each set contained the same number of occultations (i.e 676/2=338). In this way the number of measurements considered is the same, but the first half results to be sparser than the second. Figure 6.17 shows how, compared to the reference solution which spans on the whole time interval, both the first and the second set deliver a poorer estimation. However, it is also clear that the first half outperforms the second, thus confirming that, in terms of occultations distribution, the sparser the better.

Still, one may argue that the first half spans a longer time interval compared to the second, since less occultations take place in the first period but the first and second set were imposed to contain the same number of measurements. To solve this discrepancy, a second case was considered in which the occultations were halved splitting the time interval rather then the number of measurements. The result is that the first half contains 212 occultations and spans from 20^{th} April 2030 to 22^{th} November 2031, while the second half contains 464 occultations and it covers from 22^{th} November 2031 to 26^{th} June 2033. The results of this analysis are shown in Figure 6.18 and it is clear that, at least for a few hundred measurements, the first set outperforms the second again. Of course, since the first set of measurements contains 212 occultations only, the plot cannot go any further. However, comparing the trend with the other graphs as well, it is reasonable to assume that the estimation delivered by the first set will be more accurate even when a larger number of occultations was considered.

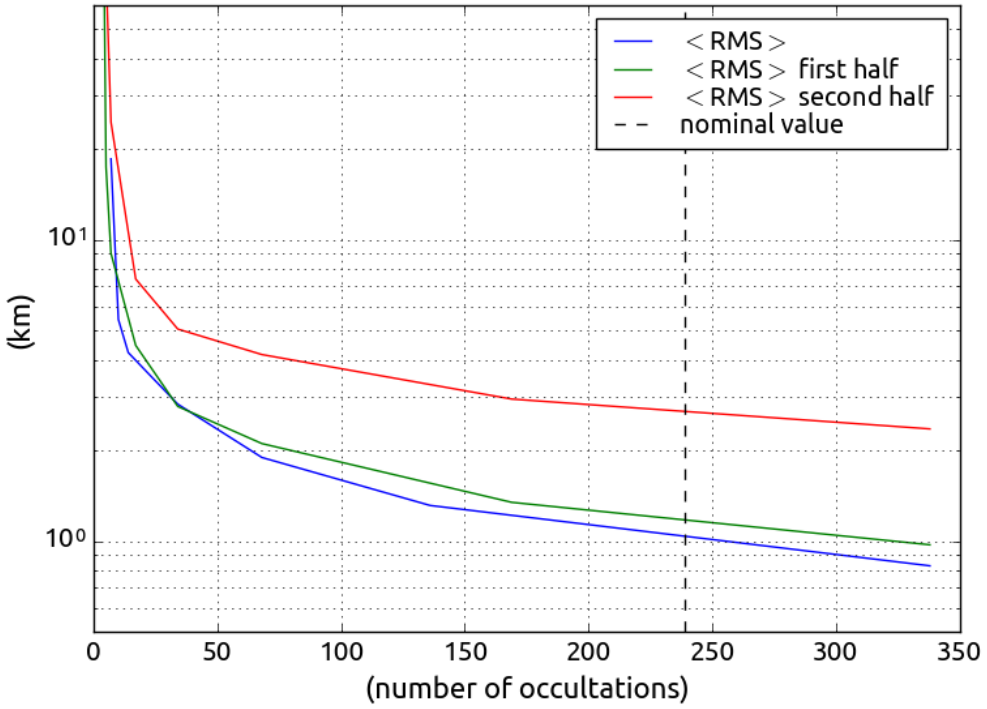

Figure 6.17: Standard deviation on Io's position as a function of the occultations distribution - the first and the second half contain the same number of measurements

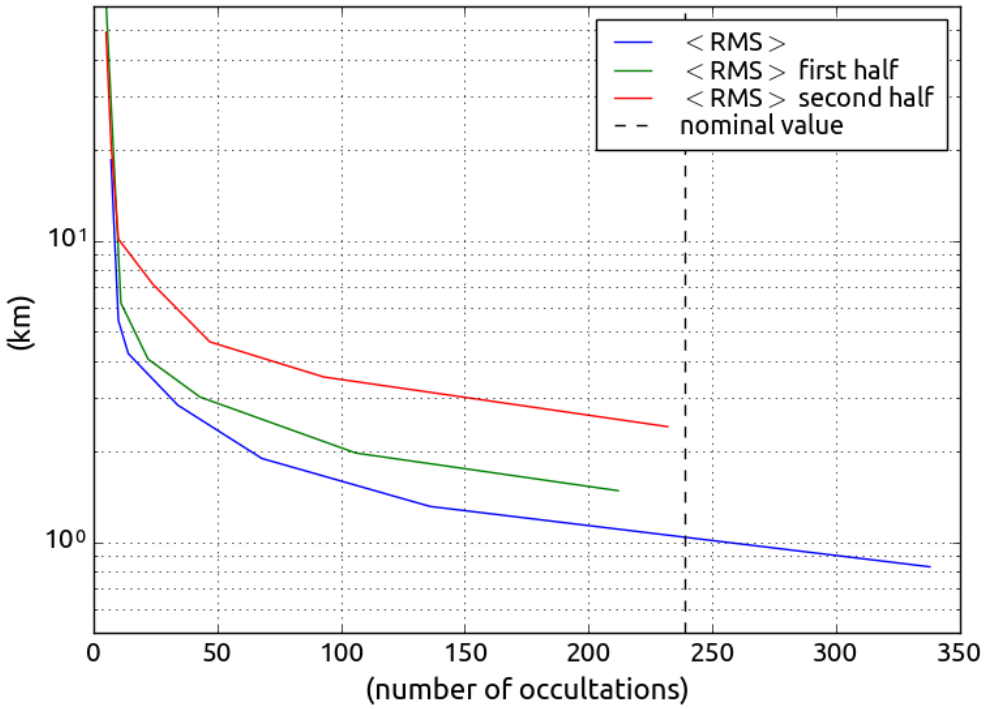

Figure 6.18: Standard deviation on Io's position as a function of the occultations distribution - the first and the second half cover the same time interval

Chapter 7

Conclusions

Overall, the results of the Covariance Analysis can be considered satisfactory and the stellar occultations proved to be a promising way to improve the satellites' ephemerides and constrain the Laplace resonance evolution. Through the estimation process it was possible to tighten the constraint on Io's position up to two orders of magnitude, which is remarkable taking into account the fact that only stellar occultation observations were included. Additionally, these measurements can be performed even if the spacecraft is relatively far away from the moon and using an instrument and a technology that would be present onboard anyway. So a part from the objective potential of stellar occultations, the benefits they could bring is even higher considering that virtually no expense would be required to include these measurements in the mission plan.

Adopting reasonable values for the measurements accuracy and for the a priori uncertainties, the standard deviation on Io's position in the radial, transverse and normal direction reached a minimum of 0.5 km, 1.2 km and 0.75 km respectively. While the R and T components can be considered completely independent from the initial conditions, the N component is not fully observable through stellar occultations measurements and is thus effected by the choice of the a priori uncertainty.

Subsequently, a parametric analysis was performed to investigate the influence of the different parameters on the estimation. Above all, a strong dependence on the accuracy of Io's shape model was observed. In fact, this can be considered the key feature to further improve the estimation. The position of JUICE affects the estimated uncertainty as well, however the current knowledge of the spacecraft's position is sufficiently accurate to avoid the error on Io's position to diverge. The influence of the other parameters considered varies slightly from one to the other, but overall their effect on the estimation is modest.

Finally, the influence of the number of stellar occultations included in the Covariance Analysis was investigated. Predictably, the accuracy on the position of Io improves as more and more occultations are added to the estimation, showing a trend proportional to $\frac{1}{\sqrt{N}}$. As a thumb rule, no sound investigation can be performed with less than 100 measurements approximately, at least in this case. A brief analysis of the effects of the stellar occultations distribution in time was performed and it looks like the sparser the measurements, the better the estimation. However, further investigations on the optimal selection of the stellar occultations

are needed to draw more decisive conclusions.

7.1 Further developments

The Covariance Analysis performed and discussed in this Thesis is just a first step in the investigation of the potential benefits that the stellar occultation observations can bring to the Orbit Determination process. Hereafter the most immediate developments and improvements that can be implemented in the work done so far will be listed and briefly discussed.

First of all, a full Orbit Determination analysis could be performed. The Covariance Analysis presented in this work relies exclusively on stellar occultations measurements. In order to make the analysis more realistic one can introduce different kinds of observations, such as Doppler and range measurements, and estimate also the orbit of JUICE and the other satellites, which are now constraint by the a priori values.

One of the most relevant simplification applied to this analysis is that UVS is always pointing toward Io but, of course, this is quite an unrealistic scenario. So one straightforward improvement would be to implement the actual pointing direction of the spectrograph. This would allow to reduce the total number of occultations detected and select them more accurately and more realistically. However, one should not expect this upgrade to improve the quality of the estimation. In fact, as mentioned in Chapter 6, the number of occultations considered in the above analysis was compared to the one published by ESA to be sure that the investigation was as authentic as possible.

As already pointed out, the most effective way to improve the current estimation would be to include a detailed mapping of Io's geographical characteristics. This would not probably be a straightforward implementation since the geometrical projection techniques used in this work assume that Io can be modeled as a three-dimensional ellipsoid. Thus one may think about modifying the current mathematical formulation of the problem to allow for the introduction of an irregular model of Io's shape.

Regarding the influence of the stellar occultations choice, in this analysis the effects that the number of occultations considered and their time distribution have on the estimation have been investigated. However, a deeper examination is needed to assess which are the optimal selection criteria. This is a crucial point when it comes to actually plan the occultations campaign: since the number of measurements that can be performed throughout a mission is usually limited, it is essential to identify which occultation events are more significant and bring the most information.

Finally, moving to a more mathematical aspect, one could include the angular error on the frame rotations in the Covariance Analysis. As explained in section 4.1, this source of uncertainty was neglected in the current investigation as its influence was found to be sufficiently small to discard it without consequential discrepancies. However, including it would benefit the completeness and robustness of the analysis.

Having implemented all these improvements, it would certainly be interesting to

apply the models and techniques developed for JUICE and Io to other spacecraft-satellite couples. The adjustment to a different system should be straightforward, but the results and insights it may bring could be less predictable. Also, the model could be adapted to detect and investigate various kinds of occultations, such as between two different moons or between stars and ring systems.

Since the employment of stellar occultations to determine the position of celestial bodies is quite an unexplored field, a whole range of additional applications could be discussed, but this goes beyond the scope of this Thesis.

Appendix A

A.1 Code - geometrical model

```
#define camera frame
camera_frame = M.DirectionFrame(boa, 'camera frame',
'EMO2000', M.TimeInterval(),
M.PositionDir( M.TrajQuery(boa, 'Io',
'Juice', 'EMO2000')),
M.PositionDir( M.TrajQuery(boa,
'Jupiter Barycenter',
'Juice', 'EMO2000')) , False )
```

```
#star search
st=M.StarCatalogBoa.read(boa)
directionij=io.state(t_plot).pos()-juice.state(t_plot).pos()
stars_aroundg=st.search(t_plot, 'EMO2000', M.Dbl3Vec(directionij
                                    ),
M.UnitDbl(angle, 'deg'),
minMagnitude, maxMagnitude)
```

```
#multiplication
Itw4x4=np.matrix([[1, 0, 0, trl[0]],
[0, 1, 0, trl[1]],
[0, 0, 1, trl[2]],
[0, 0, 0, 1.0]])
Qt=np.transpose(np.linalg.inv(Itw4x4))*Q*np.linalg.inv(Itw4x4)
Qi=np.transpose(np.linalg.inv(R_BE4x4))*Qt*np.linalg.inv(R_BE4x4
                                    )
Qc=np.transpose(np.linalg.inv(R_EC4x4))*Qi*np.linalg.inv(R_EC4x4
                                    )
Cinv=K3x4*Qc.I*K3x4.T
C=Cinv.I
```

```
#compute h
h=[]
for n in range(len(stars_aroundg)):
starnumber=n
starvector=(stars_aroundg[starnumber].dir(t_plot,'camera frame')
                                    )
starv=np.matrix([[starvector[0]],[starvector[1]],[starvector[2]]
                                    ])
starvproj=np.dot(K,starv)
```

```
  star2D=np.matrix([[starvproj.item(0)/starvproj.item(2)],
  [starvproj.item(1)/starvproj.item(2)],
  [1.0]])
  h.append(np.transpose(star2D)*C*star2D)
```

A.2 Code - occultations detection

```
import os
import Monte as M
import numpy as np
import mpylab as plt
import time

#load boa stuff
gin='/home/monte/lockfile/juice/Juice_monte_lockfile_latest.boa'
sat='/home/monte/ephem/se_jup310_de430_17991218_22000114_r131115.
                                    boa'
pln='/home/monte/ephem/pe_de430_18491226_21500122_r130321.boa'
juice='/home/naif/kernels/juice/spk/juice_latest.boa'
boa=M.BoaLoad([pln,sat,juice,gin])
trajs=M.TrajSetBoa.read(boa)

#useful constants
j2000=M.Epoch('01-JAN-2000 12:00 ET')
minMagnitude=-2 #leave it like this, there are no brighter stars
                                    in the catalog
maxMagnitude=6 #tuned with the GIF
angle=10 #The radius of the search circle, in the unit of angle.
distanceJuiceIo_max=10**10 #max distance between Juno and G
hyperbolic_const=0.0018 #in minutes
mean_radiusG=1821.6 #km
t_tol=10**(-3) #occultation time tolerance (while cycle to refine
                                    occ time)   #02:01:23.6173, 0.
                                    278664112091, 16 iter
h_tol=10**(-16)
numiter_max=1000 #max number of iterations (while cycle to refine
                                    occ time)

#EMO2000
juice=M.TrajQuery(boa, 'Juice', 'Jupiter Barycenter', 'EMO2000')
io=M.TrajQuery(boa, 'Io', 'Jupiter Barycenter', 'EMO2000')
europa=M.TrajQuery(boa, 'Europa', 'Jupiter Barycenter', 'EMO2000')
ganymede=M.TrajQuery(boa, 'Ganymede', 'Jupiter Barycenter', '
                                    EMO2000')

#define camera frame
camera_frame = M.DirectionFrame(boa, 'camera frame', 'EMO2000',
M.TimeInterval(), M.PositionDir( M.TrajQuery(boa, 'Io', 'Juice', '
                                    EMO2000')),
M.PositionDir( M.TrajQuery(boa, 'Jupiter Barycenter', 'Juice', '
                                    EMO2000')) , False )
InertialFrame(boa, 'camera frame inertial', 'camera frame', '
                                    EMO2000', M.TimeInterval() )
velg=M.TrajQuery(boa, 'Io', 'Juice', 'camera frame inertial')
```

```
#get ellipsoid parameters
ellipse=M.ShapeBoa.read(boa,'Io Ellipsoid')
ell=ellipse.ellipsoid()
a_ellipsoid=ell.a()
a_ellipsoid=a_ellipsoid.value()
b_ellipsoid=ell.b()
b_ellipsoid=b_ellipsoid.value()
c_ellipsoid=ell.c()
c_ellipsoid=c_ellipsoid.value() #all in km
Q=np.matrix([[1/(a_ellipsoid**2), 0, 0, 0],[0, 1/(b_ellipsoid**2),
                                    0, 0],[0, 0, 1/(c_ellipsoid**2),
                                    0],[0, 0, 0, -1]])

#define intrinsic camera parameters matrix
f=1
K=np.matrix([[f, 0, 0],[0, f, 0],[0, 0, 1]])
K3x4=np.matrix([[f, 0, 0, 0],[0, f, 0, 0],[0, 0, 1, 0]])

#TIME CYCLE
start = time.time()
st=M.StarCatalogBoa.read(boa)
t_start=M.Epoch('01-JAN-2030 00:00:00.0000 ET')
t_end=M.Epoch('26-JUN-2033 00:00:00.0000 ET')

#define small delta_t --> proportional to delta_alfa
delta=[]
delta_t_time=M.Epoch.range(t_start,t_end,M.Duration('60:00'))
for k in range(len(delta_t_time)):
R=(juice.state(delta_t_time[k]).pos()-io.state(delta_t_time[k]).
                                    pos()).mag() #km
angular_aperture=mean_radiusG/R #rad
delta_alfa=0.05*angular_aperture #rad
vel=sqrt(velg.state(delta_t_time[k]).vel()[0]**2+velg.state(
                                    delta_t_time[k]).vel()[1]**2) #km
                                    /s
angular_velg=vel/R #rad/s
delta.append(delta_alfa/angular_velg) #s
delta_t=M.UnitDbl(min(delta),'sec') #s
if delta_t<M.UnitDbl(10.0,'sec'):
delta_t=M.UnitDbl(10.0,'sec')

occultation_time_vector1=[]
occultation_star_vector1=[]
occultation_distance_vector1=[]
occultation_time_vector1_exit=[]
occultation_star_vector1_exit=[]
occultation_distance_vector1_exit=[]
T=t_start
while T<t_end:
R=(juice.state(T).pos()-io.state(T).pos()).mag()
vel=sqrt(velg.state(T).vel()[0]**2+velg.state(T).vel()[1]**2)
DELTA_T=M.UnitDbl(hyperbolic_const/(vel/R),'min')
if (vel/R)>0.000002:
DELTA_T=M.UnitDbl(hyperbolic_const/(2.5*vel/R),'min')
if DELTA_T>M.UnitDbl(6.0,'hour'):
```

```
DELTA_T=M.UnitDbl(6.0,'hour')


#Get stars
directionij=io.state(T).pos()-juice.state(T).pos()
print("Refreshing stars_aroundg (T=\%s)..." \% (T))
stars_aroundg=st.search(T, 'EMO2000', M.Dbl3Vec(directionij) , M.
                                    UnitDbl(angle, 'deg'),
                                    minMagnitude, maxMagnitude)

#t cycle
t=M.Epoch.range(T,T+DELTA_T,delta_t) #G position refresh time
t.append(T+DELTA_T)
print("Looking for occultations from \%s to \%s..." \% (t[0],t[-1]
                                    ))
h_old=np.matrix(np.ones((600,1)))

for j in range(len(t)):
#Translation vector
rig=M.TrajQuery(boa, 'Io', 'Jupiter Barycenter', 'Io Mean Equator
                                    Prime_Meridian of Date Non-
                                    Inertial IAU_2000')
ric=M.TrajQuery(boa, 'Juice', 'Jupiter Barycenter', 'Io Mean
                                    Equator Prime_Meridian of Date
                                    Non-Inertial IAU_2000')
rig=np.matrix([[rig.state(t[j]).pos()[0]],[rig.state(t[j]).pos()[1
                                    ]],[rig.state(t[j]).pos()[2]]])
ric=np.matrix([[ric.state(t[j]).pos()[0]],[ric.state(t[j]).pos()[1
                                    ]],[ric.state(t[j]).pos()[2]]])
trl=rig-ric #in EMO!!!

#Rotation matrices
coord=M.CoordSetBoa.read(boa)
rotation_fromGtoEMO=np.matrix(coord.rotation(t[j],'EMO2000','Io
                                    Mean Equator Prime_Meridian of
                                    Date Non-Inertial IAU_2000').m().
                                    toArray())
rotation_fromEMOtoC=np.matrix(coord.rotation(t[j],'camera frame','
                                    EMO2000').m().toArray())
rotation_fromGtoC=np.matrix(coord.rotation(t[j],'camera frame','Io
                                     Mean Equator Prime_Meridian of
                                    Date Non-Inertial IAU_2000').m().
                                    toArray())

rotation_fromEMOtoC4x4=np.matrix(np.zeros((4,4)))
for i in range(3):
rotation_fromEMOtoC4x4[i,0:3]=rotation_fromEMOtoC[i,0:3]
rotation_fromEMOtoC4x4[3,3]=1

rotation_fromGtoEMO4x4=np.matrix(np.zeros((4,4)))
for m in range(3):
rotation_fromGtoEMO4x4[m,0:3]=rotation_fromGtoEMO[m,0:3]
rotation_fromGtoEMO4x4[3,3]=1

#Multiplication
```

```
Itw4x4=np.matrix([[1, 0, 0, trl[0]],[0, 1, 0, trl[1]],[0, 0, 1,
                                    trl[2]],[0, 0, 0, 1.0]])  #DO NOT
                                     ROTATE THE TRASL VECTOR
Qt=np.transpose(np.linalg.inv(Itw4x4))*Q*np.linalg.inv(Itw4x4)
Qi=np.transpose(np.linalg.inv(rotation_fromGtoEMO4x4))*Qt*np.
                                    linalg.inv(rotation_fromGtoEMO4x4
                                    ) #same with np.dot
Qc=np.transpose(np.linalg.inv(rotation_fromEMOtoC4x4))*Qi*np.
                                    linalg.inv(rotation_fromEMOtoC4x4
                                    ) #same with np.dot
Cinv=K3x4*Qc.I*K3x4.T
C=Cinv.I

#Compute h
if (juice.state(t[j]).pos()-io.state(t[j]).pos()).mag()<
                                    distanceJuiceIo_max:
h=[]
for n in range(len(stars_aroundg)):
starnumber=n
starvector=stars_aroundg[starnumber].dir(t[j],'camera frame')#T[k]
                                     t[j]
starv=np.matrix([[starvector[0]],[starvector[1]],[starvector[2]]])
starvproj=np.dot(K,starv)
star2D=np.matrix([[starvproj.item(0)/starvproj.item(2)],[starvproj
                                    .item(1)/starvproj.item(2)],[1.0]
                                    ])
h.append(np.transpose(star2D)*C*star2D)
if h_old[n]>0.0 and h[n]<0.0:
print("There is an occultation starting at time \%s with star
                                    number \%s" \% (t[j],n))
#Store occultation info
if occultation_time_vector1==[]:
occultation_time_vector1.append(t[j])
occultation_distance_vector1.append(h[n])
occultation_star_vector1.append(stars_aroundg[n])
elif occultation_time_vector1!=[]:
if stars_aroundg[n].name()!=occultation_star_vector1[-1].name():
occultation_time_vector1.append(t[j])
occultation_distance_vector1.append(h[n])
occultation_star_vector1.append(stars_aroundg[n])
elif h_old[n]<0.0 and h[n]>0.0:
print("There is an occultation finishing at time \%s with star
                                    number \%s" \% (t[j],n))
if occultation_time_vector1_exit==[]:
occultation_time_vector1_exit.append(t[j])
occultation_distance_vector1_exit.append(h[n])
occultation_star_vector1_exit.append(stars_aroundg[n])
elif occultation_time_vector1_exit!=[]:
if stars_aroundg[n].name()!=occultation_star_vector1_exit[-1].name
                                    ():
occultation_time_vector1_exit.append(t[j])
occultation_distance_vector1_exit.append(h[n])
occultation_star_vector1_exit.append(stars_aroundg[n])

h_old=h
```

```
T=T+DELTA_T


print time.time() - start

with open('/home/fandreoli/cases/stellar_occultation/
                                    occultations_LONG.txt', 'w') as f
                                    :
f.write('#\%s \%s, \%s\r\ n' \% ('Time interval considered:',
                                    t_start,t_end))
f.write('#\%s\%s \%s\%s \%s\%s \%s\%s\%s \%s\%s \%s\%s \%s\%s\r\ n
                                    ' \% ('minMagn=',minMagnitude,'
                                    maxMagn=',maxMagnitude,'search
                                    angle=',angle,'max distance Juice
                                    -Io=',distanceJuiceIo_max,'km','
                                    hyperbolic const=',
                                    hyperbolic_const,'h tolerance=',
                                    h_tol,'max number iter=',
                                    numiter_max))
f.write('#\%15s\r\ n' \% (' Occultations starting time '))
f.write('#\%15s  \%26s\r\n' \% ('  TIME  ','  STAR NUMBER  '))

start = time.time()
print('Refining the starting occultation time...')
#Refine occultation time
occultation_time_vector1A=[]
occultation_distance_vector1A=[]
occultation_star_vector1A=[]
for k in  range(len(occultation_time_vector1)):
low=occultation_time_vector1[k]-delta_t #since t=delta_t is the
                                    time between h + and -, it is an
                                    appropriate delta time
high=occultation_time_vector1[k]
numiter=0
h_acc=1.0
while h_acc>h_tol or h_acc<-h_tol and numiter<numiter_max:#(high-
                                    low).seconds()>M.UnitDbl(t_tol,'
                                    sec') and numiter<numiter_max:
numiter=numiter+1
add=(high-low)/2.0
new=low+add

#translation vector
rig=M.TrajQuery(boa, 'Io', 'Jupiter Barycenter', 'Io Mean Equator
                                    Prime_Meridian of Date Non-
                                    Inertial IAU_2000')
ric=M.TrajQuery(boa, 'Juice', 'Jupiter Barycenter', 'Io Mean
                                    Equator Prime_Meridian of Date
                                    Non-Inertial IAU_2000')
rig=np.matrix([[rig.state(new).pos()[0]],[rig.state(new).pos()[1]]
                                    ,[rig.state(new).pos()[2]]])
ric=np.matrix([[ric.state(new).pos()[0]],[ric.state(new).pos()[1]]
                                    ,[ric.state(new).pos()[2]]])
trl=rig-ric
```

```
#rotation matrices
rotation_fromGtoEMO=np.matrix(coord.rotation(new,'EMO2000','Io
                                    Mean Equator Prime_Meridian of
                                    Date Non-Inertial IAU_2000').m().
                                    toArray())
rotation_fromEMOtoC=np.matrix(coord.rotation(new,'camera frame','
                                    EMO2000').m().toArray())
rotation_fromGtoC=np.matrix(coord.rotation(new,'camera frame','Io
                                    Mean Equator Prime_Meridian of
                                    Date Non-Inertial IAU_2000').m().
                                    toArray())

rotation_fromEMOtoC4x4=np.matrix(np.zeros((4,4)))
for i in range(3):
rotation_fromEMOtoC4x4[i,0:3]=rotation_fromEMOtoC[i,0:3]
rotation_fromEMOtoC4x4[3,3]=1

rotation_fromGtoEMO4x4=np.matrix(np.zeros((4,4)))
for m in range(3):
rotation_fromGtoEMO4x4[m,0:3]=rotation_fromGtoEMO[m,0:3]
rotation_fromGtoEMO4x4[3,3]=1

#multiplication
Itw4x4=np.matrix([[1, 0, 0, trl[0]],[0, 1, 0, trl[1]],[0, 0, 1,
                                    trl[2]],[0, 0, 0, 1.0]])  #DO NOT
                                     ROTATE THE TRASL VECTOR
Qt=np.transpose(np.linalg.inv(Itw4x4))*Q*np.linalg.inv(Itw4x4)
Qi=np.transpose(np.linalg.inv(rotation_fromGtoEMO4x4))*Qt*np.
                                    linalg.inv(rotation_fromGtoEMO4x4
                                    ) #same with np.dot
Qc=np.transpose(np.linalg.inv(rotation_fromEMOtoC4x4))*Qi*np.
                                    linalg.inv(rotation_fromEMOtoC4x4
                                    ) #same with np.dot
Cinv=K3x4*Qc.I*K3x4.T
C=Cinv.I

#compute h
starvector=occultation_star_vector1[k].dir(new,'camera frame')
starv=np.matrix([[starvector[0]],[starvector[1]],[starvector[2]]])
starvproj=np.dot(K,starv)
star2D=np.matrix([[starvproj.item(0)/starvproj.item(2)],[starvproj
                                    .item(1)/starvproj.item(2)],[1.0]
                                    ])
h_acc=np.transpose(star2D)*C*star2D
if h_acc>0.0:
low=new
else:
high=new
print("There is an occultation starting at time \%s with star
                                    number \%s (\%s) (\%s)" \% (new,
                                    occultation_star_vector1[k].name
                                    (),k,numiter))

with open('/home/fandreoli/cases/stellar_occultation/
                                    occultations_LONG.txt', 'a') as f
                                    :
```

```
f.write('\%9s  \%9s\r\ n' \% (new,occultation_star_vector1[k].name
                                ()))

#Store occultation info
occultation_time_vector1A.append(new)
occultation_distance_vector1A.append(h_acc)
occultation_star_vector1A.append(occultation_star_vector1[k])

with open('/home/fandreoli/cases/stellar_occultation/
                                occultations_LONG.txt', 'a') as f
                                :
f.write('#\%15s\r\ n' \% (' Occultations finishing time '))
f.write('#\%15s  \%26s\r\ n' \% ('  TIME  ','  STAR NUMBER  '))

print('Refining the finishing occultation time...')
#Refine occultation time
occultation_timeA_vector1_exit=[]
occultation_distanceA_vector1_exit=[]
occultation_starA_vector1_exit=[]
for k in  range(len(occultation_time_vector1_exit)):
low=occultation_time_vector1_exit[k]-delta_t
high=occultation_time_vector1_exit[k]
numiter=0
h_acc=-1.0
while h_acc>h_tol or h_acc<-h_tol and numiter<numiter_max:
numiter=numiter+1
add=(high-low)/2.0
new=low+add

#translation vector
rig=M.TrajQuery(boa, 'Io', 'Jupiter Barycenter', 'Io Mean Equator
                                Prime_Meridian of Date Non-
                                Inertial IAU_2000')
ric=M.TrajQuery(boa, 'Juice', 'Jupiter Barycenter', 'Io Mean
                                Equator Prime_Meridian of Date
                                Non-Inertial IAU_2000')
rig=np.matrix([[rig.state(new).pos()[0]],[rig.state(new).pos()[1]]
                                ,[rig.state(new).pos()[2]]])
ric=np.matrix([[ric.state(new).pos()[0]],[ric.state(new).pos()[1]]
                                ,[ric.state(new).pos()[2]]])
trl=rig-ric

#rotation matrices
rotation_fromGtoEMO=np.matrix(coord.rotation(new,'EMO2000','Io
                                Mean Equator Prime_Meridian of
                                Date Non-Inertial IAU_2000').m().
                                toArray())
rotation_fromEMOtoC=np.matrix(coord.rotation(new,'camera frame','
                                EMO2000').m().toArray())
rotation_fromGtoC=np.matrix(coord.rotation(new,'camera frame','Io
                                Mean Equator Prime_Meridian of
                                Date Non-Inertial IAU_2000').m().
                                toArray())

rotation_fromEMOtoC4x4=np.matrix(np.zeros((4,4)))
for i in range(3):
```

```
rotation_fromEMOtoC4x4[i,0:3]=rotation_fromEMOtoC[i,0:3]
rotation_fromEMOtoC4x4[3,3]=1

rotation_fromGtoEMO4x4=np.matrix(np.zeros((4,4)))
for m in range(3):
rotation_fromGtoEMO4x4[m,0:3]=rotation_fromGtoEMO[m,0:3]
rotation_fromGtoEMO4x4[3,3]=1

#multiplication
Itw4x4=np.matrix([[1, 0, 0, trl[0]],[0, 1, 0, trl[1]],[0, 0, 1,
                                    trl[2]],[0, 0, 0, 1.0]])  #DO NOT
                                     ROTATE THE TRASL VECTOR
Qt=np.transpose(np.linalg.inv(Itw4x4))*Q*np.linalg.inv(Itw4x4)
Qi=np.transpose(np.linalg.inv(rotation_fromGtoEMO4x4))*Qt*np.
                                    linalg.inv(rotation_fromGtoEMO4x4
                                    ) #same with np.dot
Qc=np.transpose(np.linalg.inv(rotation_fromEMOtoC4x4))*Qi*np.
                                    linalg.inv(rotation_fromEMOtoC4x4
                                    ) #same with np.dot
Cinv=K3x4*Qc.I*K3x4.T
C=Cinv.I

#compute h
starvector=occultation_star_vector1_exit[k].dir(new,'camera frame'
                                    )
starv=np.matrix([[starvector[0]],[starvector[1]],[starvector[2]]])
starvproj=np.dot(K,starv)
star2D=np.matrix([[starvproj.item(0)/starvproj.item(2)],[starvproj
                                    .item(1)/starvproj.item(2)],[1.0]
                                    ])
h_acc=np.transpose(star2D)*C*star2D
if h_acc<0.0:
low=new
else:
high=new
print("There is an occultation finishing at time \%s with star
                                    number \%s (\%s) (\%s)" \% (new,
                                    occultation_star_vector1_exit[k].
                                    name(),k,numiter))

with open('/home/fandreoli/cases/stellar_occultation/
                                    occultations_LONG.txt', 'a') as f
                                    :
f.write('\%9s  \%9s\r\ n' \% (new,occultation_star_vector1_exit[k]
                                    .name()))

#Store occultation info
occultation_timeA_vector1_exit.append(new)
occultation_distanceA_vector1_exit.append(h_acc)
occultation_starA_vector1_exit.append(
                                    occultation_star_vector1_exit[k])

print time.time() - start
```

A.3 Code - numerical partials

```
#partial derivative with respect to a
delta_a=1.0

ui.setup.EditEllipsoidShape(boa,
Name = "Io Ellipsoid",
Body = "Io",
AxisA = (1.829400000000000e+03 + delta_a) *km)

#Get C
C_da = project_ellipsoid(boa, t_plot, 'Io', f)

#Get h
h_da = get_h(stars_aroundi,f,t_plot,C_da)

#Partials computation
partialsa=[]
for k in range(len(h_da)):
partialsa.append((float(h_da[k])-float(h[k]))/delta_a)
```

Appendix B

B.1 Matrix inverse derivative

Noticing that for a generic matrix A and a generic scalar parameter x:

$$0 = \frac{\partial I}{\partial x} = \frac{\partial A A^{-1}}{\partial x} = \frac{\partial A}{\partial x} A^{-1} + A \frac{\partial A^{-1}}{\partial x}$$

$$A \frac{\partial A^{-1}}{\partial x} = -\frac{\partial A}{\partial x} A^{-1}$$

The partial derivative of the inverse of matrix A can be written as:

$$\frac{\partial A^{-1}}{\partial x} = -A^{-1} \frac{\partial A}{\partial x} A^{-1} \tag{B.1}$$

Bibliography

[1] Jacobson, R.A. (2013) JUP310 - JPL satellite ephemeris.

[2] J. L. Ortiz, P. Santos-Sanz, [...], R. Iglesias-Marzoa, "The size, shape, density and ring of the dwarf planet Haumea from a stellar occultation," Nature volume 550, pages 219–223 (12 October 2017).

[3] D. Bérard, B. Sicardy, [...], D. Dunham, "The Structure of Chariklo's Rings from Stellar Occultations," The Astronomical Journal, 154:144 (21pp), 2017 October.

[4] Daniel Angerhausen, Klaus F. Huber, Avi M. Mandell, Michael W. McElwain, Stefan Czesla, Nikku Madhusudhan and Jon A. Morse, "Occultation Spectrophotometry of Extrasolar Planets with SOFIA," Formation, Detection, and Characterization of Extrasolar Habitable Planets Proceedings IAU Symposium No. 293, 2012 N. Haghighipour, ed.

[5] N. Winn, J, "Exoplanet Transits and Occultations," Exoplanets, edited by S. Seager. Tucson, AZ: University of Arizona Press, 2011, 526 pp. ISBN 978-0-8165-2945-2., p.55-77.

[6] Suzanne Aigrain, R. Angus, J. Barstow, V. Rajpaul, E. Gillen, H. Parviainen, B. Pope, S. Roberts, A. McQuillan, N. Gibson, T. Mazeh, F. Pont, S. Zucker, "The effects of stellar activity on detecting and characterising exoplanets," The 19th Cambridge Workshop on Cool Stars, Stellar Systems, and the Sun (CS19), Uppsala, Sweden, 06-10 June 2016, id.12.

[7] Mark Psiaki and Joanna Hinks, "Autonomous Lunar Orbit Determination Using Star Occultation Measurements," AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina, 20 - 23 August 2007.

[8] Photojournal: PIA21970 Source: NASA/JPL-Caltech/SwRI/MSSS/Gerald Eichstädt and Seán Doran Published: November 9, 2017

[9] *https* : *//solarsystem.nasa.gov/planets/jupiter/overview/*, last retrieved: 24 February 2019.

[10] M. R. Showalter and D. P. Hamilton, "Resonant interactions and chaotic rotation of Pluto's small moons," Nature volume 522, pages 45–49 (04 June 2015).

[11] Malhotra R., "Orbital resonances and chaos in the Solar System," D. Lazzaro et al., eds., Solar system formation and evolution, ASP Conference Series, Vol. 149, 1998.

[12] de Sitter, W., "On the periodic solutions of a special case of the problem of four bodies.", Proceedings of Royal Netherlands Academy of Arts and Science 11, 682-698 (1909).

[13] The JUICE Science Working Team, "JUICE Red Book," ESA/SRE(2014)1, September 2014.

[14] $https://www.cosmos.esa.int/web/juice/images-videos$, last retrieved: 24 February 2019.

[15] R. Gladstone, K. Retherford, J. Eterno, S. Persyn, M. Davis, M. Versteeg, T. Greathouse, K. Persson, G. Dirks, B. Walther, M. Araujo, A. Steffl, E. Schindhelm, J. Spencer, M. McGrath, F. Bagenal, P. Feldman, and L. Fletcher, "The Ultraviolet Spectrograph on the JUICE Mission (JUICE-UVS)", Vol. 8, EPSC2013-394-1, 2013

[16] Jonathon Smith, William Taber, Theodore Drain, Scott Evans, James Evans, Michelle Guevara, William Schulze, Richard Sunseri, Hsi-Cheng Wu, "MONTE Python for Deep Space Navigation," PROC. OF THE 15th PYTHON IN SCIENCE CONF. (SCIPY 2016).

[17] $https://juicesoc.esac.esa.int/occ\_analysis?model\_id = StellarOccsIo30$, last retrieved: 26 February 2019

[18] P. ARGENTIERO, J. J. LYNN, "ESTIMATION STRATEGIES FOR ORBIT DETERMINATION OF APPLICATIONS SATELLITES," GODDARD SPACE FLIGHT CENTER, Greenbelt, Maryland, February 1974.

[19] $https://www.cosmos.esa.int/web/gaia/science-performance$, last retrieved: 26 February 2019

[20] Paul Schenk, Henrik Hargitai, Ronda Wilson, Alfred McEwen, and Peter Thom, "The mountains of Io: Global and geological perspectives from Voyager and Galileo," JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 106, NO. El2, PAGES 33,201-33,222, DECEMBER 25, 2001

[21] Marzia Parisi, Stefano Finocchiaro and Luciano Iess, "MULTI-ARC AND BATCH-SEQUENTIAL FILTERS FOR THE ORBIT DETERMINATION OF ESA'S JUICE MISSION," EGU General Assembly Conference Abstracts. Vol. 14. 2012.

[22] Dario Modenini, Marco Zannoni, Riccardo L. Manghi and Paolo Tortora, "An analytical approach to autonomous optical navigation for a CubeSat mission to a binary asteroid system," Advances in the Astronautical Sciences, Univelt Inc., 2018, 163, pp. 139 - 149

[23] *https* : *//www.esa.int/spaceinimages/Images/*2006/04/*Solar_occultation_at_Venus*2, last retrieved: 26 February 2019

[24] *http* : *//occult.mit.edu/research/stellarOccultations.php*, last retrieved: 26 February 2019

[25] John Allen Christian, "Optical Navigation for a Spacecraft in a Planetary System," PhD Thesis, May 2010

[26] *https* : *//blogs.nasa.gov/pluto/*2015/10/30/*in* − *the* − *shadows* − *of* − *pluto* − *and* − *charon/*, last retrieved: 26 February 2019

Acknowledgments

Having completed the Master's Degree in Aerospace Engineering is one of the accomplishments I am most proud of. Above all, I am proud of myself. However, many people have contributed to this achievement and I would like to spend a few words to thank the ones who helped me the most along the way.

First of all, I want to thank my parents as none of this would have been possible without them. Thank you for having granted me my independence when I asked for it and having given me support and advice when I needed it.

I also owe a huge thank you to my supervisor, Marco Zannoni. This Thesis is also the result of many hours spent discussing in his office, infinite e-mail threads and sprints along the halls to catch him before he leaves for the train station.

Thank you to all the *Tecnopolli* for the memorable amount of food and *disagio* that we shared, making the months spent at the Lab hard to forget.

Thank you to my classmates for having shared the good and bad times and having made the long hours of lecture and the even longer exams sessions a little more bearable.

Thank you to my friends at Megiro Gym for having offered me the chance to sweat off (or better, kick off) stress and worries when I needed to and having spared me when I was too tired to even hit back.

Finally, thank you to my family and friends from Modena for making me feel like home every time I come back, even if I am often away.