Università degli Studi di Padova

Dipartimento di Ingegneria Industriale DII
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Precise Orbit Determination of Low-Earth Satellites: a Comparison of Batch Estimation and Real-Time Filtering Techniques

Candidato:
Lorenzo Fiorese
Matricola 1237235

Relatore:
Prof. Stefano Casotto

Correlatore:
Dr. Massimo Bardella

Anno Accademico 2022–2023

To my family, for their unconditional support.
To my love, for her sincere affection.
To my friends, for the good times.

Contents

Abstract

Precise Orbit Determination (POD) has become more and more relevant over the past decades due to the ever increasing requirements artificial satellites must satisfy. This is true for both commercial and scientific missions that find applications in a wide variety of fields such as navigation, communication and Earth observation. In most cases, the correct acquisition and interpretation of data would not be possible without extremely accurate knowledge of the satellite's orbit. This is particularly true in the field of space geodesy, where centimeter-level accuracy is required to measure delicate geophysical parameters. In this regard, the objective of this thesis is to analyze and compare the main techniques used in statistical orbit estimation. Unlike Initial Orbit Determination (IOD), we add the realistic assumption that the model used for dynamic propagation is inevitably erroneous, as are the measurements used to track the satellite. These estimation techniques must therefore demonstrate their ability to filter these data in order to derive an optimal estimate of the satellite's state, along with other geometric and dynamic parameters. The comparison is made by distinguishing real-time filtering techniques (Kalman filters) from offline processes that are executed "a posteriori" after the measurements have been collected. In this second case, we are referring to the least squares batch estimators or differential corrections algorithms. Finally, the effects of smoothing techniques are studied as a post-processing solution that allows for the improvement of the estimate previously made with a real-time filter. Furthermore, for the good performance of an orbital estimator, a critical aspect is the correct and complete modeling of the accelerations acting on a satellite in orbit. This is particularly true for offline estimators like the batch, given that since they collect measurements over fairly long time arcs, they are sensitive even to the smallest perturbations. In this regard, within the context of the numerical modeling of forces, a significant part of the work involves analyzing the development and implementation of a routine that allows for the calculation of the perturbation resulting from Earth Radiation Pressure (ERP), considering the contribution of both visible (albedo) and Infrared (IR) components. The entire work is conducted within the context of the Navigation and Space Geodesy group at the University of Padova, led by Professor Stefano Casotto. Here, the use of numerous previously available Fortran libraries, along with independently developed routines and programs, yields results that clearly confirm the validity of the implemented estimation techniques. Generally, sub-decimeter level accuracies are achieved on synthetic data obtained from an estimated orbit of Sentinel-3A, provided by the European Space Agency (ESA). A careful analysis of the results shows that real-time filtering techniques tend to yield better results compared to batch estimation, for which it is necessary to deepen and expand the currently available force model. Furthermore, even though the data are satisfactory, the understanding is that only a test on real measurements can definitively determine the performance of the developed software. Nonetheless, this work has allowed for a deep dive into the issue of precise orbit determination, motivating further efforts aimed at improving the current project.

Chapter 1

Introduction

1.1 Context

The orbit determination problem finds its roots back to the XVII century, with the first who tackled the determination of celestial bodies trajectories being Kepler (c. 1610) and Legendre (c. 1750). However, a first formal, analytical and computational basis of the issue was given by Gauss (c. 1810) who, along with Legendre, is claimed to be the father of the *least squares method*. The rediscovery of the asteroid Ceres (1801) and the calculation of its orbit made by Gauss is considered to be the first time astronomers had constructed and predicted an orbit through observations.
Thanks to the technological advances made in the last century in observational instruments, and the more and more demanding precision requirements, needed especially for certain applications such as geodesy, Earth observation and Global Positioning System, orbit determination has seen an extensive improvement in determining the position and velocity of artificial satellites orbiting the Earth, with missions that require centimeter-level accuracy. For example, TOPEX/Poseidon launched in 1992 measured the ocean surface topography with an accuracy below 5 centimeters, a result that would not be attainable if its orbit was not known with extreme accuracy. Nevertheless, besides the technological improvements, one major leap in estimation theory occurred when Rudolf E. Kalman introduced the now called *Kalman Filter* in his work *A New Approach to Linear Filtering and Prediction Problems* [11]. The sequential version of the least squares algorithm was immediately recognised as a powerful tool suited for real time estimation and particularly appropriate for including process noise, laying the foundations for many modern applications in the most varied fields and disciplines from navigation, tracking, economics up to medical monitoring and many others. However, Orbit estimation remains one of the main fields of use.

When we talk about *precise orbit determination* (POD), we are distinguishing it from *preliminary orbit determination*, more recently known as *Initial Orbit Determination* (IOD), and even if in both cases we are dealing with determining a particular trajectory, these two proceedings exhibit significant differences. In preliminary orbit determination we are concerned with the determination of the six orbital elements with no a priori information about the dynamical state and the assumption that bodies move under the influence of a point mass force only, but with the sufficient number of observations that make the solution to the problem unique, which is clearly six, and these observations often being two sets of range and angles or three direction vectors. Even if this method is fairly accurate for large celestial bodies, it definitely isn't satisfactory for small artificial satellites that require high tracking accuracy and are more susceptible to non gravitational forces and perturbations due to their small mass or high velocity in the vicinity of the planet's atmosphere, giving rise to the need of a more sophisticated method that accurately models the largest number of perturbations, and includes the effects of random perturbations and measurements errors. It is in this context that the concept of precise orbit determination becomes necessary.

Predicting, Filtering and *Smoothing* are the three main topics involved in POD applications.

Predicting is the process by which future states are determined from current observations and a priori information through propagation techniques; in other words, given an initial state at epoch, the differential equations governing the motion are integrated to future times. However, the first complications encountered lie in the fact that both the initial state and the mathematical model, along with geometrical and dynamical parameters used to propagate the trajectory, are never known exactly, meaning that sooner or later the actual path will diverge from the predicted one. For this reason, periodic measurements are needed to update knowledge on the object state, but these are themselves subject to systematic and random errors which must be accounted for when performing orbit determination. This leads to the need of a *Filtering* procedure of these data, which is getting a *better estimate* of the current state from current and previous observations. Finally, *Smoothing* is the process of searching for a better estimate of the state at a time, including future observations, making it suitable for improving the estimate in offline applications rather than real-time tracking.

In summary, *precise orbit determination* is the process of determining a *best estimate* of the state of an orbiting satellite considering that:

- we are not able to directly measure the state (position and velocity) of a satellite;
- the mathematical model might be more or less approximate and incomplete;
- both the mathematical model used for propagation and the observations-state relations possess nonlinear characteristics;
- the observations available at any time epoch are often less than the state vector components to be determined;
- errors in numerical integration, computer truncation and roundoff errors are present;
- observations contain systematic and random errors.

In this thesis the orbit determination problem will be addressed through the application of two main types of filtering techniques, the *Batch* estimation and the *Kalman* filter, then, the effects of *smoothing* techniques are discussed.

1.2 Thesis outline

The aim of this thesis is to study, analyze, implement and compare the main techniques used for the Precise Orbit Determination (POD) of low-Earth satellites, namely the *batch estimation*, *Kalman filters* and *smoothing*. The work is carried out in the framework of the Navigation and Space Geodesy group led by Professor Stefano Casotto at the University of Padova, and the main focus is the development of a collection of routines and programs written in Fortran, whose purpose is to test and validate orbit determination procedures on satellite data. First, an in-depth discussion on the numerical modeling of the main accelerations acting on a satellite is carried out, given that an effective POD software must primarily be able to simulate many acting perturbations as precisely as possible, in order to increase the prediction accuracy during propagation. In doing so, the numerical implementation of these accelerations is thoroughly examined with a *special perturbations* approach, which is well-suited for numerical integration as every effect is modeled as a single-point acceleration. Here, particular attention is given to the algorithmic implementation of an Earth Radiation Pressure (ERP) model developed by Knocke (1988) [13] which, in the context of precision modeling, is expected to have noticeable effects on the batch estimation results. Then, the linearization of the orbit model is discussed as a preliminary requirement for the understanding and development of subsequent chapters, that dive deep into the three major estimators used for orbit determination. The first of these is the *batch* processor, which is immediately identified as the off-line estimator as opposed to real-time filters. Here, the derivation of the nonlinear least squares approach, also known as *differential correction*, is thoroughly exposed as well as the concepts of parameters segmentation and estimation of empirical accelerations, which are discussed as useful tools when dealing with this kind of estimation. Next, the dissertation proceeds by deriving the basic real-time filter from a sequential version of the batch processor, as in Tapley [24], and a particular case of this sequential estimation happens to coincide with the famous Kalman filter. This real-time filter is first presented in its conventional form, then, the Extended Kalman Filter (EKF) is derived and proposed as an improved version of the conventional one as its performances are better in the presence of high nonlinearities, which makes it suited for the orbit determination problem. Subsequently, the inclusion of process noise in the Kalman filter is addressed considering two main approaches: the State Noise Compensation algorithm (SNC) and the Dynamic Model Compensation (DMC) [22, 8], where their differences and peculiarities are highlighted in the context of avoiding filter divergence. Afterwards, the orbit determination through Optimal Smoothing (OS) is shown and derived following two different approaches: the *forward-backward* and the RTS smoothers. Here, it is shown how the forward-backward formulation consists of two independent filters, with the forward one being a conventional Kalman, while the second is a backward information filter. On the other hand, the RTS formulation is derived and presented as an alternative that is more compact and lighter from an algorithmic implementation standpoint. The last Chapter, contains the critic discussion and assessment of the results obtained from the implementation in Fortran of these three orbit determination approaches. The simulations are performed starting from available data of the Sentinel-3A satellite, provided by the European Space Agency (ESA) in the form of inertial position and velocity components over a period of roughly seven days. From these coordinates, a synthetic observations history is created with the Earth-fixed coordinates of fifty existing ground stations, that are available from the International GNSS Service (IGS). It's important to highlight here that given the synthetic nature of the measurement data, this is not a true precise orbit determination, which instead should use real measurements that involve numerous complications. Nonetheless, the purpose of the thesis is not so much to perform a true orbital determination, but to compare different methodologies, highlighting their peculiarities, so, under such circumstances, the best outcome that can be achieved is not an absolute orbital estimate, but the determination of the force and kinematic models that were used for the generation of the input orbit data. In this regard, the use of synthetic data rather than real data provides a controlled environment in which it is easier to draw detailed conclusions. First, the results associated to the batch estimation are presented. A box-wing satellite model with simulated attitude dynamics is firstly implemented during estimation, then, the results obtained from the use of a cannon-ball model are shown as well as the beneficial effects of parameters segmentation for this latter case. Next, the results obtained with the EKF algorithm are discussed showing its properties of stability and convergence when a reduced-dynamics approach is used together with the DMC algorithm, then, the adverse effects caused by the lack of process noise are displayed in terms of filter divergence. Finally, the smoother results are exposed, showing the improvements on the solution precision that are associated to a lower estimated covariance. This is done while distinguishing *dynamical* and *bias* states as those components that are, or are not, affected by smoothing procedures. Moreover, the effect of initial conditions on the smoother performance is further analyzed, while exposing the major drawback that this estimator inherits from the forward conventional filter: its sensitivity to initial position and velocity. The thesis is then concluded with a critical evaluation of the results obtained, together with a reflection on the main limitations of this work and possible further improvements.

Chapter 2

Numerical Orbit Modeling

In the context of precise orbit determination a critical aspect is orbit propagation. Prediction of future states is carried out by means of trajectory integration over time, based on the mathematical models used to describe the forces acting on the spacecraft. For this reason, the importance of a well defined dynamical model becomes clear, and the higher is the number of forces described, the higher will be the prediction accuracy, at the expense, however, of computational effort. Except for the gravity field of a central point mass, which alone leads to Keplerian orbits, other major forces called *perturbations* that should be considered for precise orbit determination can be distinguished into two main categories based on their nature: *gravitational perturbations* and *non-gravitational perturbations*. The former are related to every dynamical effect on the satellite trajectory that can be derived as the gradient of a gravitational field generated by different massive bodies or any mass distribution, that being constant or time-varying, while the latter are due to different physical phenomena whose effect does not depend on the satellite's mass, but strongly depend on its geometry. These latter perturbations often require the computation of integrals over the satellite's surfaces and the knowledge of their physical properties such as reflective coefficients and temperature, for this reason, they are frequently called *skin forces* and their non-conservative nature leads to variations in orbit energy. Hereafter, the main perturbations of interest for this analysis on POD applications are listed with a brief description, and later on, their mathematical modeling is described.

- *Aspherical gravitational potential*: the Earth, but also every celestial body in general, is not a perfect homogeneous sphere, meaning that the central point mass is actually an approximation of reality, and if one were to neglect this fact a good trajectory prediction would not be possible. In mathematical terms this effect can be considered using the expansion of the gravitational potential in spherical harmonics, and the degree of this expansion determines the accuracy of the representation. In addition to this, the aspherical and inhomogeneous characteristics of the Earth are not of static nature, but periodic deformations of the geoid occur due to the tidal forces exerted by other celestial bodies like the Sun and the Moon, causing *solid tides* and *ocean tides* which make the geopotential variable over time.

- *Atmospheric drag*: for near Earth satellites, the drag force due to relative velocity between the spacecraft and Earth's atmosphere cannot be neglected for POD applications, even if the atmospheric density at typical LEO altitudes can be fourteen orders of magnitude smaller compared to sea-level. The border line below which atmospheric effects should be considered is usually set at 1000 km, and here, if the right countermeasures are not taken, the satellite experiences first a circularization phase due to higher velocity at perigee that quickly decreases apogee altitude, and then a spiraling phase that eventually leads to disintegration in atmosphere or surface impact.

- *Third body effects*: considering that the Earth is the only boding attracting the spacecraft might be insufficient for many applications. For Earth orbiting satellites the other two main attractors are the sun and the moon, whose perturbing effect becomes greater as orbit altitude increases. For this reason, in LEO satellites applications that do not require high levels of precision these effects might be neglected, but for missions requiring POD they certainly become appreciable.

- *Solar radiation pressure*: when a satellite is exposed to solar radiation it experiences a perturbing force due to the transfer in momentum between impinging photons and satellite's surfaces. Here, the optical characteristic of the materials covering the spacecraft become important, as they influence the mechanism by which the photons are absorbed or reflected, thus, the net force acting on the satellite.
- *Earth radiation pressure*: For Earth orbiting satellites another source of radiation pressure is Earth itself. Even if for a wide variety of applications the previously mentioned perturbations are sufficient to accurately propagate and predict a satellite's trajectory, for other missions such as the US/French TOPEX/Poseidon satellite altimeter whose purpose was the mapping of oceans surface with an accuracy under 10 cm, neglecting the effect of earth radiation would lead to insufficient performance in position and velocity estimation. Moreover, since this effect becomes more tangible for LEO satellites over long periods of time, for orbit estimation performed with a batch processor, the presence of this force model in the propagation of the reference trajectory can surely benefit the estimation performance, for this reason, particular attention will be given in the discussion of its mathematical model and algorithm implementation. On the other hand, for real-time filters where the trajectory is continuously updated after short time spans, earth radiation pressure can be neglected in favor of a lower computational demand.

2.1 Equations of Motion

Considering a geocentric inertial reference frame the equations of motion of and Earth orbiting satellite can be expressed as

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_{kepl} + \ddot{\mathbf{r}}_{pert}, \tag{2.1.1}$$

where $\ddot{\mathbf{r}}_{kepl}$ is the acceleration due to a central point mass while $\ddot{\mathbf{r}}_{pert}$ encloses all remaining accelerations acting on the satellite, i.e. the aforementioned perturbations. This latter term can be further separated into *gravitational* and *non-gravitational* components:

$$\ddot{\mathbf{r}}_{pert} = \ddot{\mathbf{r}}_{g} + \ddot{\mathbf{r}}_{ng}, \tag{2.1.2}$$

which in turn comprise different sources of acceleration distinguished by their nature, namely:

$$\begin{aligned} \ddot{\mathbf{r}}_{g} &= \ddot{\mathbf{r}}_{geo} + \ddot{\mathbf{r}}_{nb} + \ddot{\mathbf{r}}_{st} + \ddot{\mathbf{r}}_{ot} \\ \ddot{\mathbf{r}}_{ng} &= \ddot{\mathbf{r}}_{drag} + \ddot{\mathbf{r}}_{SRP} + \ddot{\mathbf{r}}_{ERP} + \ddot{\mathbf{r}}_{emp} \end{aligned}$$

where

- $\ddot{\mathbf{r}}_{geo}$ is the acceleration due to the non-spherical Earth's gravitational potential,
- $\ddot{\mathbf{r}}_{nb}$ is the acceleration due to n-body effects,
- $\ddot{\mathbf{r}}_{st}$ is the acceleration associated to solid Earth tides,
- $\ddot{\mathbf{r}}_{ot}$ is related to ocean tides,
- $\ddot{\mathbf{r}}_{drag}$ is the acceleration caused by atmospheric drag,

- $\ddot{\mathbf{r}}_{SRP}$ is the acceleration given by the Solar Radiation Pressure,
- $\ddot{\mathbf{r}}_{ERP}$ is the acceleration contribution from the Earth Radiation Pressure, both in the visible (albedo) and InfraRed (IR) spectrum,
- $\ddot{\mathbf{r}}_{emp}$ represents the empirical accelerations which have no physical counterpart, but are a means of accounting for small unmodeled perturbative effects.

Then, the motion of a satellite around the Earth is completely modeled by the following equation

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_{kepl} + \ddot{\mathbf{r}}_{geo} + \ddot{\mathbf{r}}_{nb} + \ddot{\mathbf{r}}_{st} + \ddot{\mathbf{r}}_{st} + \ddot{\mathbf{r}}_{ot} + \ddot{\mathbf{r}}_{drag} + \ddot{\mathbf{r}}_{SRP} + \ddot{\mathbf{r}}_{ERP}. \tag{2.1.3}$$

The following Sections discuss the derivation of mathematical models for the perturbative accelerations, with considerations on their numerical and algorithmic implementation.

2.2 Geopotential

We know that, as a first approximation, Earth orbiting satellites move on Keplerian orbits that are mathematically described by conic sections like ellipses, parabolas and hyperbolas. These particular trajectories are obtained as solutions of the equations that govern the motion of a body subjected to a radially symmetric force field, described by Newton's law of universal gravitation, according to which the force experienced by two point masses attracting each other is given by

$$|\mathbf{F}| = G\frac{m_1 m_2}{r^2}, \tag{2.2.1}$$

where r is the distance between m_1 and m_2, and G is the gravitational constant ($G \approx 6.6743 \times 10^{-11}\,\mathrm{Nm}^2/\mathrm{kg}^2$). In terms of acceleration, the effect that m_2 has on m_1 cane be expressed as follows

$$\ddot{\mathbf{r}}_1 = \frac{\mathbf{F}}{m_1} = -G\frac{m_2}{r^2}\mathbf{e}_r, \tag{2.2.2}$$

with $\mathbf{e}_r$ being the unit vector pointing from m_2 to m_1, and for a more general treatment, we recall that the acceleration vector can be interpreted as the gradient of a scalar *gravitational potential* function U. So, if m_2 is replaced by the more general central mass M, the gravitational potential at a distance r from M is

$$U = G\frac{M}{r}, \qquad \nabla U = \ddot{\mathbf{r}}. \tag{2.2.3}$$

The potential described in (2.2.3) is clearly radially symmetric and often known as the Earth's *spherical* gravitational potential. However, as anticipated before, the Earth is not perfectly spherical nor homogeneous in density distribution; in fact, it is well known that due to its constant rotation it can be represented more like and oblate ellipsoid rather than a sphere, with the equatorial radius surpassing the polar radius by about 20 km. This oblateness has tangible effects on Earth orbiting satellites which manifest themselves as a secular regression of the Right Ascension of the Ascending Node (RAAN) for prograde orbits, that is actually a precession of the orbital plane, given that the perturbing effect can be seen as a torque that tends to align the orbital plane to the equator. The Earth's flattening at the poles is certainly the strongest gravitational perturbation acting on satellites, nevertheless, it is not the only one and for POD applications a more realistic representation of the geopotential is necessary.

A more general formulation for the geopotential, that allows to model non-spherical and inhomogeneous masses, can be expressed as an integral over the volume of the central body

$$U = G\int_{Vol} \frac{\rho(\mathbf{s})}{|\mathbf{r}-\mathbf{s}|}\,\mathrm{d}s^3, \tag{2.2.4}$$

here, the density is a function of the location inside the body $\rho(\mathbf{s})$, and $dm = \rho(\mathbf{s})\mathrm{d}s^3$ is the infinitesimal mass element at a distance $|\mathbf{r}-\mathbf{s}|$ from the satellite, see figure 2.2.1. The crucial aspect is that the term $|\mathbf{r}-\mathbf{s}|^{-1}$, often called *slant range*, can be expanded as a series of *Legendre polynomials.* Given the generating function

$$\frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty} P_n(x)t^n, \tag{2.2.5}$$

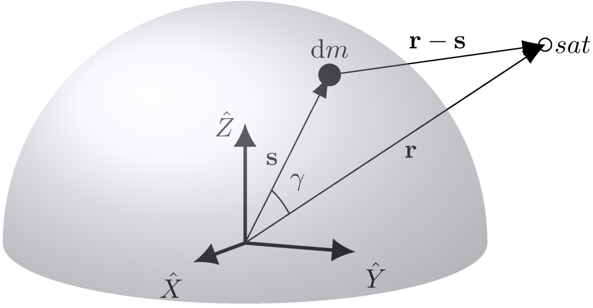

Figure 2.2.1: Earth element dm and its contribution to the Geopotential.

we can write

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \frac{1}{r\sqrt{1-2\frac{s}{r}\cos\gamma+\left(\frac{s}{r}\right)^2}} = \frac{1}{r}\sum_{n=0}^{\infty}\left(\frac{s}{r}\right)^n P_n\big(\cos\gamma\big), \quad \text{with} \quad \cos\gamma = \frac{\mathbf{r}\cdot\mathbf{s}}{r\,s}, \tag{2.2.6}$$

and (2.2.4) becomes

$$U = \frac{G}{r}\int_{Vol}\sum_{n=0}^{\infty}\left(\frac{s}{r}\right)^n P_n\big(\cos\gamma\big)\rho(\mathbf{s})\mathrm{d}s^3, \tag{2.2.7}$$

where $\mathrm{P}_n\big(\cos\gamma\big)$ is the Legendre polynomial of degree n with argument $\cos\gamma$, that can be computed following the compact Rodriguez's formula

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^2-1\right)^n. \tag{2.2.8}$$

It is interesting to notice how equation (2.2.7) gives the *Newtonian* potential if we consider the 0^{th} degree Legendre polynomial only, in fact, knowing that $\mathrm{P}_0(x) = 1$ we obtain

$$U = \frac{G}{r}\int_{Vol}\rho(\mathbf{s})\mathrm{d}s^3 = \frac{GM_\oplus}{r}, \tag{2.2.9}$$

where $M_\oplus$ is the Earth mass.

Now, indicating with (ϕ,λ) and (ϕ',λ') the geocentric latitude and longitude of the satellite and mass element respectively, a further expansion of (2.2.7) is obtained considering that from the addition theorem of spherical harmonics, we can write P_n as a series of *associated Legendre functions* of degree n and order m like

$$P_n\big(\cos\gamma\big) = \sum_{m=0}^{n}(2-\delta_{0m})\frac{(n-m)!}{(n+m)!}P_{nm}\big(\sin\phi\big)P_{nm}\big(\sin\phi'\big)\cos(m(\lambda-\lambda')), \tag{2.2.10}$$

where δ_{0m} is the Kronecker delta, and $\cos\gamma$ has been expressed using the spherical trigonometry law of cosines as

$$\cos\gamma = \sin\phi'\sin\phi + \cos\phi'\cos\phi\cos(\lambda - \lambda'), \tag{2.2.11}$$

moreover, in (2.2.10), a compact formula for the degree n and order m associated function is given similarly to (2.2.8) as

$$P_{nm}(x) = (1 - x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x). \tag{2.2.12}$$

From the computational point of view, the Legendre polynomials and associated Legendre functions are seldom computed using equations (2.2.8) and (2.2.12) since, although providing a direct expression for $P_n(x)$ and $P_{nm}(x)$, they require the computation of several derivatives. For this reason, a recursive algorithm is preferable and is given by the following relations [6]:

$$\bar{P}_{nn}\big(\sin\phi\big) = f_n \cos\phi \bar{P}_{n-1,n-1}\big(\sin\phi\big), \qquad n \geq 1, \tag{2.2.13}$$

$$\bar{P}_{nm}\big(\sin\phi\big) = g_{nm} \sin\phi \bar{P}_{n-1,m}\big(\sin\phi\big) - h_{nm}\bar{P}_{n-2,m}\big(\sin\phi\big), \qquad n \geq m+1, \tag{2.2.14}$$

where $\bar{P}$ indicates that the Legendre function has been fully normalized with the normalizing coefficients f_n, g_{nm} and h_{nm} defined as

$$f_n = \sqrt{\frac{(1+\delta_{1n})(2n+1)}{2n}} \tag{2.2.15}$$

$$g_{nm} = \sqrt{\frac{(2n+1)(2n-1)}{(n+m)(n-m)}} \tag{2.2.16}$$

$$h_{nm} = \frac{g_{nm}}{g_{n-1,m}} = \sqrt{\frac{(2n+1)(n-m-1)(n+m-1)}{(2n-3)(n+m)(n-m)}}. \tag{2.2.17}$$

Finally, by substitution of equation (2.2.10) into (2.2.7) and separating terms related to satellite's position to the ones associated to Earth mass distribution, we can arrive at the complete expansion of the geopotential

$$U = \frac{GM_{\oplus}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{R_{\oplus}}{r}\right)^n \bar{P}_{nm}\big(\sin\phi\big)\Big[\bar{C}_{nm}\cos(m\lambda) + \bar{S}_{nm}\sin(m\lambda)\Big], \tag{2.2.18}$$

and here,

- $M_{\oplus}$ is the Earth's mass;
- $R_{\oplus}$ is the mean Earth radius and r is the geocentric satellite distance;
- $\bar{P}_{nm}$ Are normalized associated Lagrange polynomials of degree n and order m;
- $\bar{C}_{nm}$ and $\bar{S}_{nm}$ are normalized harmonic *Stokes* coefficients;
- ϕ and λ are the satellite's geocentric longitude and latitude.

The coefficients $\bar{C}_{nm}$ and $\bar{S}_{nm}$ describe the internal mass distribution of the central body as well as its shape, and their expressions are given by Montenbruck (2000) [19]

$$\bar{C}_{nm} = N_{nm}\frac{(2-\delta_{0m})}{M_{\oplus}}\frac{(n-m)!}{(n-m)!}\int_{Vol}\left(\frac{s}{R_{\oplus}}\right)^n P_{nm}\big(\sin\phi'\big)\cos(m\lambda')\rho(\mathbf{s})\mathrm{d}s^3 \tag{2.2.19}$$

$$\bar{S}_{nm} = N_{nm}\frac{(2-\delta_{0m})}{M_{\oplus}}\frac{(n-m)!}{(n-m)!}\int_{Vol}\left(\frac{s}{R_{\oplus}}\right)^n P_{nm}\big(\sin\phi'\big)\sin(m\lambda')\rho(\mathbf{s})\mathrm{d}s^3, \tag{2.2.20}$$

with N_{nm} being the normalizing coefficient defined as

$$N_{nm} = \sqrt{\frac{(n+m)!}{(2-\delta_{0m})(2n+1)(n-m)!}}. \tag{2.2.21}$$

The purpose of the normalization procedure is to make the Lgenedre functions and Stokes coefficients more uniform by a numerical standpoint. Non-normalized vales would span over a range of more than ten orders of magnitude, even for low degree and low order models, potentially leading to a loss of precision in finite digit arithmetic.

Although the formulation for the geopotential in equation (2.2.18) completely describes the gravitational potential of a generic mass, it is common to find another expression in literature that highlights the structure and properties of the normalized Stokes coefficients in (2.2.19) and (2.2.20). As mentioned earlier, they only contain terms related to the Earth shape and density distribution, and interesting aspects arise when considering certain values for the degree n and order m. First of all, it can be shown that if the center of the coordinate system used to define all spatial quantities is located at the Earth's center of mass, the coefficients C_{10}, C_{11} and S_{11} all vanish; also, S_{10} is zero by definition. This means that the first summation with respect to the index n can start from $n = 2$ rather than from $n = 0$ and the 0-*th* degree term is factored out as the newtonian potential. Moreover, given that every S_{n0} vanish by definition, we can group the contributions from every C_{n0} in a summation of its own with the following definition

$$-C_{nm} = J_n, \tag{2.2.22}$$

and we finally arrive at

$$U = \frac{GM_\oplus}{r}\Bigg\{1 - \sum_{n=2}^{\infty} J_n \left(\frac{R_\oplus}{r}\right)^n \bar{P}_n(\sin\phi) + \\ + \sum_{n=2}^{\infty}\sum_{m=1}^{n} \left(\frac{R_\oplus}{r}\right)^n \bar{P}_{nm}(\sin\phi)\Big[\bar{C}_{nm}\cos(m\lambda) + \bar{S}_{nm}\sin(m\lambda)\Big]\Bigg\}. \tag{2.2.23}$$

The trigonometric arguments of the associated Legendre polynomials represent spherical harmonics in equation (2.2.23), where their degree and order actually represent lines over the Earth's surface where these polynomials are null. Based on the values of n and m, these spherical harmonics can be grouped into three different types. The first infinite sum, in the index n and coefficients J_n, gives the Geopotential part related to *zonal harmonics* that describe variations in potential dependent on latitude only. In other words, the field associated to this components is symmetric about the polar axis and can be represented as bands in latitude. The first of them, associated to J_2, separates the Earth's surface into three regions with a large equatorial band of positive mass concentration that describes the Earth's oblateness mentioned before, which is by far the strongest gravitational perturbation after the spherical term. The second sum in indexes n and m gives *sectoral harmonics* and *tesseral harmonics*. Sectoral harmonics are found when $n = m$ and unlike zonal harmonics, they represent bands of longitude delimited by n circumferences passing through the polar axis, dividing the Earth's surface into $2n$ slices of alternating positive and negative mass concentration sectors. The strongest perturbation associated to a sectoral harmonic is related to the Legendre function of degree and order 2, that is $C_{2,2}$. The physical explanation is the fact that the Earth is more of an ellipse rather than a circle even considering its equatorial cross section, with the major axis at 14.7° W to the Greenwich meridian. Geostationary satellites are the ones particularly sensitive to this perturbation for they are fixed relative to the Earth's surface and thus, they do not average out this effect along their orbit, drifting towards stable regions of least potential. Tesseral harmonics are obtained when $n \neq m \neq 0$, they divide the Earth's surface into tiles that extend both in latitude and longitude in a sort of chessboard-like fashion, and often, sectoral harmonics are considered a subset of them, as a special case in which $n = m$.

As was shown earlier in equation (2.2.3), once a mathematical formulation for the complete scalar geopotential is available, the acceleration acting on an orbiting satellite can be computed by taking its gradient, that is, to obtain the acceleration components in the inertial geocentric J2000 reference frame we must compute

$$\begin{aligned}\mathbf{a}_{geo} &= \nabla U \\ &= \begin{bmatrix} \frac{\partial U}{\partial x} & \frac{\partial U}{\partial y} & \frac{\partial U}{\partial z} \end{bmatrix}.\end{aligned} \tag{2.2.24}$$

However, since the geopotential in (2.2.23) is expressed in spherical coordinates, we must use the chain rule of derivation to get

$$\ddot{x} = \frac{\partial U}{\partial r}\frac{\partial r}{\partial x} + \frac{\partial U}{\partial \phi}\frac{\partial \phi}{\partial x} + \frac{\partial U}{\partial \lambda}\frac{\partial \lambda}{\partial x}, \tag{2.2.25}$$

$$\ddot{y} = \frac{\partial U}{\partial r}\frac{\partial r}{\partial y} + \frac{\partial U}{\partial \phi}\frac{\partial \phi}{\partial y} + \frac{\partial U}{\partial \lambda}\frac{\partial \lambda}{\partial y}, \tag{2.2.26}$$

$$\ddot{z} = \frac{\partial U}{\partial r}\frac{\partial r}{\partial z} + \frac{\partial U}{\partial \phi}\frac{\partial \phi}{\partial z} + \frac{\partial U}{\partial \lambda}\frac{\partial \lambda}{\partial z}, \tag{2.2.27}$$

and the partial derivatives of the geopotential with respect to the spherical coordinates are

$$\frac{\partial U}{\partial r} = \frac{GM_\oplus}{r^2}\left\{1 + \sum_{n=2}^{\infty}\left(\frac{R_\oplus}{r}\right)^n (n+1)\sum_{m=0}^{n}\bar{P}_{nm}(\sin\phi)\left[\bar{C}_{nm}\cos(m\lambda) + \bar{S}_{nm}\sin(m\lambda)\right]\right\}, \tag{2.2.28}$$

$$\begin{aligned}\frac{\partial U}{\partial \phi} &= \frac{GM_\oplus}{r}\sum_{n=2}^{\infty}\left(\frac{R_\oplus}{r}\right)^n \left[\bar{C}_{nm}\cos(m\lambda) + \bar{S}_{nm}\sin(m\lambda)\right]\times \\ &\qquad \times\left[\bar{P}_{n,m+1}(\sin\phi) - m\tan\phi\bar{P}_{n,m+1}(\sin\phi)\right],\end{aligned} \tag{2.2.29}$$

$$\frac{\partial U}{\partial \lambda} = \frac{GM_\oplus}{r}\sum_{n=2}^{\infty}\left(\frac{R_\oplus}{r}\right)^n m\bar{P}_{nm}(\sin\phi)\left[\bar{S}_{nm}\cos(m\lambda) - \bar{C}_{nm}\sin(m\lambda)\right], \tag{2.2.30}$$

while partial derivatives $\frac{\partial r}{\partial \mathbf{r}}$, $\frac{\partial \phi}{\partial \mathbf{r}}$ and $\frac{\partial \lambda}{\partial \mathbf{r}}$ are computed through transformations between Cartesian inertial frame and spherical coordinates frame, that must account for temporal variations between the two caused by various effects like polar motion, nutation, precession and sidereal rotation.

The expansion of the geopotential in spherical harmonics requires the knowledge of the Stokes coefficients $\bar{C}_{nm}$ and $\bar{S}_{nm}$, and as was shown earlier, these coefficients describe the Earth's mass distribution through integrals over a volume of arbitrary shape and arbitrary density distribution $\rho(\mathbf{s})$. Clearly, it is difficult to imagine how these coefficients could be derived by actual integration of equations (2.2.19) and (2.2.20), for this reason, Stokes coefficient are actually obtained from satellite measurements, where missions like GRACE (NASA-DLR) helped in developing a high definition time-varying model for the Earth's gravitational field. Furthermore, summation in (2.2.23) cannot practically be computed since it is infinite, in reality, one considers terms up to a certain degree and order depending on the particular mission that is considered, knowing that the effects of higher order terms are progressively smaller and negligible. For POD applications in LEO orbits a 120×120 (degree and order) geopotential can be enough to adequately recover orbital data from measurements.

2.2.1 Solid Earth tides

The influence of third bodies on an Earth orbiting satellite's dynamics is not only direct, as a third body effect, but in a sense it can be indirect by means of Earth's solid and ocean tides. This is the case for the lunisolar tidal perturbations, where the gravitational pull of the sun and moon are strong enough to actually deform the Earth's shape, and therefore its mass distribution, leading to small periodical variations in the gravitational field of the Earth that may give insufficient accuracy for POD applications if they are not accounted for.

Considering the case of solid Earth tides, these deformations can be accurately described by the generating potential of the lunisolar tide, so that the tidal force acting on a point $\mathbf{r}$ located on the surface of the Earth, can be computed as the gradient of this potential:

$$\mathbf{F}_T = \nabla_r U(\mathbf{r}), \tag{2.2.31}$$

moreover, considering a non rotating reference frame K centered at the Earth's center of mass, the tidal force acting on the point $\mathbf{r}$ generated by a perturbing body (Moon or Sun), can also be defined as

$$\mathbf{F}_T(\mathbf{r}) = \mathbf{F}(\mathbf{r}) - \mathbf{F}(\mathbf{r}_E). \tag{2.2.32}$$

Here, the second term $\mathbf{F}(\mathbf{r}_E)$ appearing on the RHS of equation (2.2.32) arises because K is a non inertial frame. If we consider a co-rotating frame, the gravitational potential of the perturbing body of mass M_p, gives a potential U on a point on the Earth's surface $\mathbf{r}$ that can be expressed as [19],

$$U = \frac{GM_p}{|\mathbf{s}-\mathbf{r}|} + \frac{1}{2}n^2 d^2, \tag{2.2.33}$$

where M_p is the mass of the perturbing body (Moon or Sun), $\mathbf{s}$ is its geocentric position, n is its mean motion about an axis passing through the system's center of mass and d is its distance from this axis. A brief analysis shows that the perturbing potential U_t has the following form [2]:

$$U_t = \frac{GM_p}{R_\oplus} \sum_{n=2}^{\infty} \left(\frac{R_\oplus}{s}\right)^{n+1} P_n\big(\cos\psi\big), \tag{2.2.34}$$

and here d is the distance of the tide generating body from the Earth's center of mass and ψ is its zenith angle, that is the angle between $\mathbf{s}$ and $\mathbf{r}$. It is interesting to note how, considering the first term in (2.2.34) for $n = 2$, the perturbing potential is proportional to GM_p/s^3, meaning that the solid Earth tide caused by the Moon is almost twice as strong as the one generated by the Sun alone. Moreover, since this potential depends on ψ we can expect it to contain many different periods due to the Earth's rotation, also, depending on s both for the Sun and Moon, periodical variations arise because of eccentric orbits. Going back to (2.2.34), it would be convenient to express it in spherical coordinates similarly to what was done for the geopotential, and by means of the spherical harmonics rotation theorem we get

$$U_t = \frac{GM_p}{R_\oplus} \sum_{n=2}^{\infty}\sum_{m=0}^{n} \left(\frac{R_\oplus}{s}\right)^{n+1} (2-\delta_{0m})\frac{4\pi}{2n+1} Y_n^m(\phi,\lambda) Y_n^{*m}(\delta,\alpha)\mathrm{e}^{im\theta_g}, \tag{2.2.35}$$

where (ϕ, λ) are the geocentric coordinates of the point of interest in the gravitational field, (δ, α) are the equatorial coordinates of the perturbing body, $Y_n^m(\phi, \lambda)$ is a surface spherical harmonic with Y^* being its complex conjugate. Note that we are only interested on the real part of (2.2.35). Similarly to what was done for the geopotential, where we separated terms related to the Earth from terms related to the satellite position, here we need to separate geographic terms containing (ϕ, λ) from the astronomical ones containing (δ, α), to do so, we introduce the following time dependent function:

$$B_{nm}(t) = (2-\delta_{0m})\frac{4\pi}{2n+1}\frac{M_p R_\oplus}{GM_\oplus}\left(\frac{R_\oplus}{s}\right)^{n+1} Y_n^{*m}(\delta,\alpha)\mathrm{e}^{im\theta_g}, \tag{2.2.36}$$

which is useful to express the perturbing potential as:

$$U_t(\phi,\lambda,t) = g\sum_{n=2}^{\infty}\sum_{m=0}^{n} B^*_{nm}(t)Y_n^m(\phi,\lambda), \tag{2.2.37}$$

with g being the gravitational acceleration on the Earth's surface.

Even though the introduction of the time varying function $B_{nm}(t)$ has drastically simplified the expression for the perturbing tidal potential, that is expressed as a function of spherical coordinates and time, the main difficulty lies precisely in determining $B_{nm}(t)$. This function requires to accurately model the perturbing bodies motion in order to know the evolution of $\delta(t)$, $\alpha(t)$ and $s(t)$, however, this is simplified if the motions are described in the ecliptic coordinate system, whereas in first approximation and except for few cases, every body of the solar system lies on the ecliptic plane, with the mean longitudes becoming linear functions of time allowing for an harmonic representation of $B_{nm}(t)$ and the tidal potential.

Besides these deeper considerations, the gist of this discussion on solid tides is that due to the presence of perturbing bodies, the Earth's mass distribution in not invariant in time, thus, the geopotential alone cannot fully describe the effects of the Earth's gravitational field on an orbiting satellite. It has been demonstrated that, through spectral decomposition, the modeling of the tidal perturbations really comes down to geopotential variations expressed through time-varying Stokes coefficients, which are represented as a constant term plus a perturbation term:

$$\bar{C}'_{nm}(t) = \bar{C}^{static}_{nm} + \Delta\bar{C}_{nm}, \qquad \bar{S}'_{nm}(t) = \bar{S}^{static}_{nm} + \Delta\bar{S}_{nm} \tag{2.2.38}$$

where

$$\Delta\bar{C}_{nm} = \frac{k_n R_\oplus^{n+1}}{GM_\oplus}\sqrt{\frac{4(n+2)(n-m)\,!}{(n+m)\,!}}q_{nm} \tag{2.2.39}$$

$$\Delta\bar{S}_{nm} = \frac{k_n R_\oplus^{n+1}}{GM_\oplus}\sqrt{\frac{4(n+2)(n-m)\,!}{(n+m)\,!}}u_{nm} \tag{2.2.40}$$

with the q and u functions being respectively

$$q_{nm} = \sum_{j=2}^{3}\frac{GM_j}{r_{1j}}\frac{2(n-m)\,!}{(n+m)\,!}P_{nm}\big(\sin\phi'_j\big)\cos(m\lambda'_j) \tag{2.2.41}$$

$$u_{nm} = \sum_{j=2}^{3}\frac{GM_j}{r_{1j}}\frac{2(n-m)\,!}{(n+m)\,!}P_{nm}\big(\sin\phi'_j\big)\sin(m\lambda'_j), \tag{2.2.42}$$

where the subscript $j = 1,\ 2,\ 3$ refers to Earth, Moon and Sun respectively, while coordinates ϕ' and λ' are the geocentric earth-fixed latitude and longitude of the *j-th* perturbing body. Considering that the Earth is elastic only to the first order, whereas rate-dependent constituents like fluids dissipate energy, a more realistic model should also include a parameter that accounts for the lag with which the solid tide arises due to perturbing forces. Finally, note that since the first degree term ($n = 2$) of the tidal potential is proportional to $1/r^3$, the acceleration acting on a satellite, computed as the gradient of this potential, falls down with at least $1/r^4$, meaning that this perturbation is particularly important for low-orbiting satellites that require high accuracy in POD applications, while it can be consciously neglected for higher satellites.

2.2.2 Ocean Tides

When considering the effects of tidal forces in the Earth's mass distribution and the consequent effects on an Earth orbiting satellite, one must consider also the displacement of liquid bodies over the surface of the Earth due to Sun and Moon perturbations. This phenomena is known as ocean tides, and even if its treatment is similar to the case of solid tides it deserve a discussion on its own given the different behaviour of fluids compared to solid constituents. In this case, the perturbing potential is expressed as

$$\begin{aligned} U_{ot} = 4\pi G\rho_w R_\oplus \sum_{\mu}\sum_{n=0}^{\infty}\sum_{m=0}^{n}\sum_{+}^{-} \frac{1+k_n}{2n+1}\left(\frac{R_\oplus}{r}\right)^{n+1} \times \\ \times P_{nm}(\sin\Phi)\bar{C}^{\pm}_{\mu nm}\sin(\bar{\eta}_\mu\bar{\beta}(t) \pm m\lambda \pm \bar{\epsilon}^{\pm}_{\mu nm}), \end{aligned} \tag{2.2.43}$$

where ρ_w is the seawater density, k_n are the load deformation coefficients of degree n, μ is the harmonic constituent coefficient of the ocean tide, $\bar{\beta}(t) = [\tau,\, s,\, h,\, p,\, N',\, p_1]$ are Doodson arguments defining lunar and solar ephemeris, $\bar{\eta} = [\eta_1,\, \eta_2,\, \dots,\, \eta_6]$ are Doodson coefficients, $\bar{C}^{\pm}_{\mu nm}$ are the tidal constituents amplitudes and $\bar{\epsilon}^{\pm}_{\mu nm}$ are the phase angles. Similarly to what was done for solid Earth tides, we can conveniently represent the effects of ocean tides as variations over time of the Stokes coefficients in the geopotential, where this formulation has the additional benefit of being linear in the parameters. If we define

$$\begin{bmatrix} C^{\pm} \\ S^{\pm} \end{bmatrix}_{\mu nm} = \bar{C}^{\pm}_{\mu nm}\begin{bmatrix} \sin \\ \cos \end{bmatrix}(\bar{\eta}_\mu\beta(t) \pm m\lambda)$$

$$\begin{bmatrix} A \\ B \end{bmatrix}_{\mu nm} = \begin{bmatrix} C^{+} + C^{-} \\ S^{+} + S^{-} \end{bmatrix}\cos(\bar{\eta}_\mu\beta(t)) + \begin{bmatrix} S^{+} + S^{-} \\ C^{+} + C^{-} \end{bmatrix}\sin(\bar{\eta}_\mu\beta(t)),$$

and

$$F_{nm} = \frac{4\pi R_\oplus^2\rho_w}{M_\oplus}\sqrt{\frac{(n+m)\,!}{(n-m)\,!(2n+1)(2-\delta_{0m})}}\left(\frac{n+k'_n}{2n+1}\right),$$

the total potential that includes gravitational and tidal (ocean) contributes can be written as

$$\begin{aligned} U = \frac{GM_\oplus}{r}\sum_{n=2}^{\infty}\sum_{m=1}^{n}\left(\frac{R_\oplus}{r}\right)^n \bar{P}_{nm}(\sin\phi)\times \\ \times\left[\left(\bar{C}'_{nm} + F_{nm}\sum_{\mu} A_{\mu nm}\right)\cos(m\lambda) + \left(\bar{S}'_{nm} + F_{nm}\sum_{\mu} B_{\mu nm}\right)\sin(m\lambda)\right]. \end{aligned} \tag{2.2.44}$$

Ocean tides have smaller effects on a satellite dynamics if compared to solid tides up to about one order of magnitude, mainly because of the difference in the average density of saltwater that is around five times smaller than the average density of solid Earth.

2.3 Third Body

Third body perturbations refer to the effects that massive celestial bodies, other than the Earth, impose on geocentric satellites. This effect, in contrast to other perturbations, gets more intense at higher altitudes where disturbances distinctive of the LEO environment like atmospheric drag get weaker, however, for high precision applications like satellite geodesy it must be modeled even for very-low orbiting satellites. A simple straightforward formulation for the third body effects can be derived from Newton's law of gravitation applied to a system of three bodies that we will identify with their masses $M_\oplus$, m_s and M_p, respectively the Earth's mass, the satellite's mass and

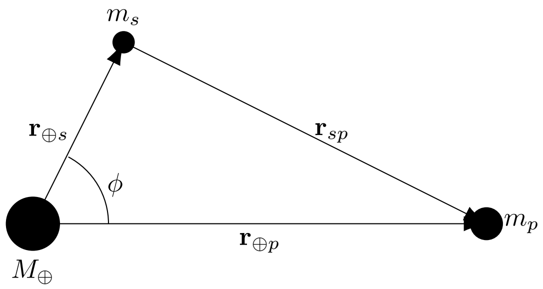

Figure 2.3.1: Schematic representation for the third body perturbation.

the perturbing body's mass as in figure 2.3.1. We can now refer to an external inertial reference frame I_{xyz} and consider the equations of motion for the Earth and the Satellite after identifying their respective positions in this system with the vectors $\mathbf{r}_E$ and $\mathbf{r}_s$, we have:

$$\ddot{\mathbf{r}}_\oplus = Gm_s \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{\oplus s}|^3} + GM_p \frac{\mathbf{r}_{\oplus p}}{|\mathrm{r}_{\oplus p}|^3} \tag{2.3.1}$$

$$\ddot{\mathbf{r}}_s = -GM_\oplus \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{\oplus s}|^3} - GM_p \frac{\mathbf{r}_{ps}}{|\mathrm{r}_{ps}|^3}, \tag{2.3.2}$$

but since we are interested in the satellite's motion relative to the Earth we subtract (2.3.1) from (2.3.2) and arrive at

$$\begin{aligned} \ddot{\mathbf{r}}_s - \ddot{\mathbf{r}}_\oplus &= -G(M_\oplus + m_s) \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{\oplus s}|^3} - GM_p \left[\frac{\mathbf{r}_{ps}}{|\mathrm{r}_{ps}|^3} + \frac{\mathbf{r}_{\oplus p}}{|\mathrm{r}_{\oplus p}|^3} \right] \\ &= -G(M_\oplus + m_s) \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{\oplus s}|^3} + GM_p \left[\frac{\mathbf{r}_{sp}}{|\mathrm{r}_{sp}|^3} - \frac{\mathbf{r}_{\oplus p}}{|\mathrm{r}_{\oplus p}|^3} \right], \end{aligned} \tag{2.3.3}$$

where we considered that $\mathbf{r}_{ps} = -\mathbf{r}_{sp}$. We can extend equation (2.3.3) to the more general case of N perturbing bodies noting that the first term on the RHS is always related to the satellite-Earth interaction, and if we define $\ddot{\mathbf{r}}_{s\oplus} = \ddot{\mathbf{r}}_s - \ddot{\mathbf{r}}_\oplus$ we can write

$$\ddot{\mathbf{r}}_{s\oplus} = -G(M_\oplus + m_s) \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{Es}|^3} + G \sum_{i=1}^{N} M_{p_i} \left[\frac{\mathbf{r}_{sp_i}}{|\mathrm{r}_{sp_i}|^3} - \frac{\mathbf{r}_{\oplus p_i}}{|\mathrm{r}_{\oplus p_i}|^3} \right]. \tag{2.3.4}$$

For what concerns LEO satellites the summation in (2.3.4) can be safely extended to the Sun and Moon only, knowing that the perturbative effect of other planets like Venus and Jupiter is around five orders of magnitude weaker in terms of acceleration [19, p. 55], thus

$$\ddot{\mathbf{r}}_{s\oplus} = -G(M_\oplus + m_s) \frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{Es}|^3} + G \sum_{i=1}^{2} M_{p_i} \left[\frac{\mathbf{r}_{sp_i}}{|\mathrm{r}_{sp_i}|^3} - \frac{\mathbf{r}_{\oplus p_i}}{|\mathrm{r}_{\oplus p_i}|^3} \right], \tag{2.3.5}$$

with $i = 1$ referring to the Moon and $i = 2$ for the Sun.

The derivation of equation (2.3.5) is suitable for numerical integration where the perturbative acceleration is computed at every step once all variables involved are known, nevertheless, some care is required when the perturbing body is significantly further away from the Earth and satellite, which is especially the case for the Sun. In this situation $\mathbf{r}_{sp}^3$ and $\mathbf{r}_{\oplus p}^3$ are very high and similar to each other, leading to a considerably small value for the difference inside the brackets in (2.3.4) that may cause numerical instability and errors. Considering this, an interesting insight into the dynamics of an Earth orbiting satellite perturbed by a third massive body, that can also help in dealing with the just mentioned problem, can be obtained from a *general perturbation* approach in which the aim is to develop analytical solutions for the changes in orbital element caused by a perturbative function $\mathcal{R}$. For the particular case of the two body dynamics perturbed by a third body we can write

$$\ddot{\mathbf{r}}_{s\oplus} = -G(M_\oplus + m_s)\frac{\mathbf{r}_{\oplus s}}{|\mathrm{r}_{Es}|^3} + \nabla\mathcal{R}_p, \tag{2.3.6}$$

where the first term on the RHS of the equation is still the two-body component, while the acceleration due to the third body is written as the gradient of a perturbative potential $\mathcal{R}_p$ whose expression is [15]

$$\mathcal{R}_p = GM_p\left[\frac{1}{|\mathbf{r}_{sp}|} - \frac{\mathbf{r}_{\oplus s}\cdot\mathbf{r}_{\oplus p}}{|\mathbf{r}_{\oplus p}|^3}\right]. \tag{2.3.7}$$

Now, let us redefine some quantities in order to simplify notation, referring to figure 2.3.1, as

$$\begin{aligned}\mathbf{r}_{sp} &= \boldsymbol{\rho}\\ \mathbf{r}_{\oplus s} &= \mathbf{r}\\ \mathbf{r}_{\oplus p} &= \mathbf{r}_p,\end{aligned}$$

then, the expression for the perturbation potential, knowing that $\mathbf{r}\cdot\mathbf{r}_p = rr_p\cos(\phi)$, can be rewritten as

$$\begin{aligned}\mathcal{R} &= GM_p\left[\frac{1}{\rho} - \frac{rr_p}{r_p^3}\cos(\phi)\right]\\ &= GM_p\left[\left(r^2+r_p^2-2rr_p\cos(\phi)\right)^{-\frac{1}{2}} - \frac{rr_p}{r_p^3}\cos(\phi)\right]\\ &= \frac{GM_p}{r_p}\left\{\left[1+\left(\frac{r}{r_p}\right)^2 - 2\left(\frac{r}{r_p}\right)\cos(\phi)\right]^{-\frac{1}{2}} - \left(\frac{r}{r_p}\right)\cos(\phi)\right\},\end{aligned} \tag{2.3.8}$$

where $\cos(\phi) = (\mathbf{r}\cdot\mathbf{r_p})/(rr_p)$ and we used the Law of Cosines for $\rho = \sqrt{r^2+r_p^2-2rr_p\cos(\phi)}$. The square brackets in the final step of equation (2.3.8) can be expanded using Legendre Polynomials by recognising the generating function (see equation (2.2.5)) in the following manner

$$\left[1+\left(\frac{r}{r_p}\right)^2 - 2\left(\frac{r}{r_p}\right)\cos(\phi)\right]^{-\frac{1}{2}} = \sum_{n=0}^{\infty}\left(\frac{r}{r_p}\right)^n \mathrm{P}_n\big(\cos(\phi)\big), \tag{2.3.9}$$

recalling that $\mathrm{P}_n\big(\cos(\phi)\big)$ is the Legendre Polynomial of degree n in the argument $\cos(\phi)$. Then, knowing that $\mathrm{P}_0\big(\cos(\phi)\big) = 1$ and $\mathrm{P}_1\big(\cos(\phi)\big) = \cos(\phi)$ we arrive at

$$\begin{aligned}\mathcal{R} &= \frac{GM_p}{r_p}\left\{1+\left(\frac{r}{r_p}\right)\cos(\phi) + \left[\sum_{n=2}^{\infty}\left(\frac{r}{r_p}\right)^n \mathrm{P}_n\big(\cos(\phi)\big)\right] - \left(\frac{r}{r_p}\right)\cos(\phi)\right\}\\ &= \frac{GM_p}{r_p}\left[1+\sum_{n=2}^{\infty}\left(\frac{r}{r_p}\right)^n \mathrm{P}_n\big(\cos(\phi)\big)\right].\end{aligned} \tag{2.3.10}$$

The formulation we have arrived at circumvents the obstacle of possible numerical instability due to the differencing of small and similar values, but at the expense of introducing an infinite sum. Nonetheless, the series rapidly converges for $r < r_p$, which is the case, allowing to adequately approximate the solution with few terms. Interestingly, neglecting terms associated to degrees greater than 2 we obtain the following approximated expression for the perturbing potential

$$\begin{aligned}\mathcal{R} \approx \frac{GM_p}{r_p}\left[1+\left(\frac{r}{r_p}\right)^2 \mathrm{P}_2\big(\cos(\phi)\big)\right] &= \frac{GM_p}{r_p}\left[1+\left(\frac{r}{r_p}\right)^2 \frac{1}{2}\big(3\cos^2(\phi)-1\big)\right] \\ &= \frac{GM_p}{r_p}\left[1+\frac{3}{2}\left(\frac{r}{r_p}\right)^2 \cos^2(\phi) - \frac{1}{2}\left(\frac{r}{r_p}\right)^2\right] \\ &= \frac{GM_p}{r_p} + \frac{GM_p}{r_p^3}\left[\frac{3}{2}r^2\cos^2(\phi) - \frac{1}{2}r^2\right].\end{aligned} \tag{2.3.11}$$

Then, by taking the gradient of (2.3.11) with respect to $\mathbf{r}$ we get an approximated expression for the perturbing acceleration caused by the third body, that is

$$\nabla_{\mathbf{r}}\mathcal{R} \approx \nabla_{\mathbf{r}}\left(\frac{GM_p}{r_p}\right) + \nabla_{\mathbf{r}}\left(\frac{3}{2}\frac{GM_p}{r_p^3}r^2\cos^2(\phi)\right) - \nabla_{\mathbf{r}}\left(\frac{1}{2}r^2\right), \tag{2.3.12}$$

where for the first and third terms on the RHS of (2.3.12) we simply have

$$\nabla_{\mathbf{r}}\left(\frac{GM_p}{r_p}\right) = \mathbf{0} \tag{2.3.13}$$

$$\nabla_{\mathbf{r}}\left(\frac{1}{2}r^2\right) = \frac{1}{2}\nabla_{\mathbf{r}}r^2 = \frac{1}{2}2\mathbf{r} = \mathbf{r} = r\hat{\mathbf{e}}_r, \tag{2.3.14}$$

while, regarding the second term, let us rewrite it considering the following relation

$$\begin{aligned} r^2\cos^2(\phi) &= r^2\left(\frac{\mathbf{r}\cdot\mathbf{r}_p}{r\,r_p}\right)^2 \\ &= \frac{(\mathbf{r}\cdot\mathbf{r_p})^2}{r_p^2}\end{aligned} \tag{2.3.15}$$

so

$$\begin{aligned}\nabla_{\mathbf{r}}\left(\frac{3}{2}\frac{GM_p}{r_p^3}r^2\cos^2(\phi)\right) &= \frac{3}{2}\frac{GM_p}{r_p^3}\nabla_{\mathbf{r}}\left[\frac{(\mathbf{r}\cdot\mathbf{r}_p)^2}{r_p^2}\right] \\ &= \frac{3}{2}\frac{GM_p}{r_p^3}2\frac{(\mathbf{r}\cdot\mathbf{r}_p)\mathbf{r}_p}{r_p^2} \\ &= 3\frac{GM_p}{r_p^3}\frac{rr_p^2(\hat{\mathbf{e}}_r\cdot\hat{\mathbf{e}}_{r'})\hat{\mathbf{e}}_{r'}}{r_p^2} \\ &= 3\frac{GM_p}{r_p^3}r(\hat{\mathbf{e}}_r\cdot\hat{\mathbf{e}}_{r'})\hat{\mathbf{e}}_{r'},\end{aligned} \tag{2.3.16}$$

and finally, using the results from (2.3.13), (2.3.14) and (2.3.16) into equation (2.3.12) we arrive at

$$\nabla_{\mathbf{r}}\mathcal{R} = \ddot{\mathbf{r}} \approx \frac{GM_p}{r_p^3}r\big[3(\hat{\mathbf{e}}_r\cdot\hat{\mathbf{e}}_{r'})\hat{\mathbf{e}}_{r'} - \hat{\mathbf{e}}_r\big]. \tag{2.3.17}$$

The result we have just arrived at, gives an approximate expression for the acceleration due to the influence of a third body M_p on the satellite's motion and although being approximate, it gives a more comprehensive insight if compared to equations (2.3.5) or (2.3.3). From equation (2.3.17) we can see how the perturbing acceleration increases linearly with increasing distance from the orbited body r, while it decreases with the cube of the satellite-perturbing body distance r_p. Moreover, we can further see how this equation tells us that the satellite is accelerated towards the Earth when $\hat{\mathbf{e}}_r$ and $\hat{\mathbf{e}}_{r'}$ are perpendicular, i.e. $\cos(\phi) = 0$, where (2.3.17) reduces to

$$\ddot{\mathbf{r}} \approx -\frac{GM_p}{r_p^3}\mathbf{r}, \tag{2.3.18}$$

on the other hand, the satellite is accelerated away from the Earth for both $\hat{\mathbf{e}}_r = \hat{\mathbf{e}}_{r'}$ and $\hat{\mathbf{e}}_r = -\hat{\mathbf{e}}_{r'}$, that is $\cos(\phi) = \pm 1$, and here (2.3.17) becomes

$$\ddot{\mathbf{r}} \approx 2\frac{GM_p}{r_p^3}\mathbf{r}, \tag{2.3.19}$$

and note how the perturbing acceleration directed away from the Earth is double in magnitude if compared to the acceleration towards the Earth.

The determination of third body perturbations on Earth orbiting satellites caused by the Sun and Moon requires the knowledge of their position at every time of interest. Considering that the distance of perturbing bodies r_p from the Earth is far greater than the geocentric satellite's distance r and it is also elevated to the third power at the denominator of (2.3.17), for many applications it is sufficient to consider low-precision solar and lunar coordinates that come from analytical theories of the motion of the Sun and the Moon, thus, the perturbing body coordinates are computed from appropriate mean orbital elements that are considered to be valid around a reference date. As an example, the following orbital elements approximate the Sun's orbit relative to the Earth with a validity of several decades around the year 2000 [19],

$$\begin{aligned}
a &= 149\,600\,000\,\mathrm{km}\\
e &= 0.016709\\
i &= 0.000^\circ\\
\Omega + \omega &= 282.94^\circ\\
M &= 357.5256^\circ + 35\,999.049^\circ \cdot T\\
T &= (\mathrm{JD} - 2\,451\,545.0)/36\,525.0
\end{aligned}$$

then, the position coordinates needed are computed using equations for Keplerian orbits. A similar approach can be used for the Moon, whereas a larger number of terms is required since its orbit is highly perturbed by the Earth and the Sun. Although this approximate method is sufficient in many cases and it is suited for the derivation of analytical models for the satellite's perturbed motion, for high precision applications Sun and Moon coordinates might be required very frequently and to a higher accuracy, in this case, the use of Development Ephemerides (DE) that are publicly available from the Jet Propulsion Laboratory (JPL) is advised. These ephemerides are considered a standard for high precision planetary coordinates and are available in the form of Chebyshev approximations derived from detailed numerical integration of the equations of motion, considering relativistic corrections, the influence of selected asteroids and other perturbations like the lunisolar torque.

2.4 Atmospheric Drag

Atmospheric drag can be the strongest non-gravitational perturbation for low orbiting satellites. Even though at typical LEO altitudes (700-800 km) the atmospheric density can be as low as

$2 \times 10^{-14}\,\mathrm{kg/m^3}$, the high relative velocity of the satellite with respect to the atmosphere gives rise to noticeable accelerations that can not be ignored. In fact, this perturbation may become the major factor in determining the End Of Life (EOL) of a particular mission, whereas GOCE (ESA 2009), that somehow represents a pathological case with an altitude of 255 km, had to compensate for the atmospheric drag with a continuously active electric ion engine and its fixed solar panels were shaped like aerodynamic fins to stabilize the spacecraft. Eventually, when it ran out of Xenon propellant, it only took less than three weeks to re-enter and disintegrate near the Falkland Islands. The force arising from satellite-atmosphere interaction can be separated into three orthogonal components: drag, lift and binormal. For an orbiting satellite the major component will always be the drag which is directed opposite to its relative velocity with the atmosphere as a braking force, whereas lift and binormal forces are usually several orders of magnitude weaker and can be safely neglected in most cases. The detailed modelling of this perturbation is a difficult task and often requires many preliminary assumptions due to three main reasons. First, the complete knowledge of the atmosphere physical properties is needed, and even if we know that in first approximation its density decreases exponentially with altitude, it also depends on latitude, the period of the year and geomagnetic and solar activity, leading to perceivable variations on its value that can change from hour to hour. Secondly, computing the acceleration exerted by the atmospheric drag requires the knowledge of the interaction between gas molecules, that can be neutral or charged, and the satellite's surfaces, where the two main scattering mechanisms involved are specular elastic reflection and diffuse reflection of gas particles. Thirdly, as can be expected, the satellite's attitude law becomes relevant for non spherical satellites as their cross-section facing the impinging atmosphere may not be constant over time, causing the resulting force to vary both in magnitude and direction.

A fundamental expression for the drag force and the subsequent acceleration imparted on the satellite can be simply obtained by considering the momentum exchange between a surface and a mass-portion of the atmosphere. If we call this small mass of gas particles Δm we can write

$$\Delta m = \rho A v_r \Delta t, \tag{2.4.1}$$

where v_r is the magnitude of the already mentioned relative velocity, ρ is the atmospheric density, A is the satellite's cross section and Δt is the small time interval during which Δm collides with A. Then, the impulse Δp experienced by the satellite can be expressed as

$$\Delta p = \Delta m v_r = \rho A v_r^2 \Delta t, \tag{2.4.2}$$

and we know that the force is related to the impulse through $F = \frac{\Delta p}{\Delta t}$, thus

$$F = \rho A v_r^2, \tag{2.4.3}$$

which gives the acceleration as

$$a_D = \rho \frac{A}{m} v_r^2, \tag{2.4.4}$$

with m being the satellite's mass. In order to be consistent with common notation in aerodynamic applications we introduce the factor $\frac{1}{2}$ and we explicit the drag acceleration direction with the relative velocity unit vector $\mathbf{e}_v$, but most importantly, we introduce the drag coefficient C_D into (2.4.4) that now becomes

$$\mathbf{a}_D = -\frac{1}{2} C_D \frac{A}{m} \rho v_r^2 \mathbf{e}_v. \tag{2.4.5}$$

Here, the minus sign underlines how this acceleration is always anti-parallel to the relative velocity, and the C_D coefficient is a dimensionless quantity that incorporates the effects related to the satellite's surface material and its interaction with the atmosphere molecules. Typical values of this coefficient can range between 1 and 3, depending on the characteristic shape of the satellite and the behaviour of the gas molecules flow. Regarding this last aspect, as altitude increases, so does the mean free path λ of the molecules, and as λ gets greater than the satellite's reference dimension l, particles interactions with the body become independent from one another as collision between them becomes more and more unlikely. This flow regime, characterised by a Knudsen number $K = \lambda/l > 10$, is known as *free molecular flow* and here the typical C_D value for a spherical body is around 2.2. As altitude decreases and the Knudsen number gets lower than 0.1 the flow regime behaves more like a *continuum flow*, where the gas molecules form a continuous stream of particles that envelop and "shield" the body. In this case, the flow-satellite interactions change so much that the drag coefficient for a sphere drops down to around 1.

2.4.1 Absorption and Diffuse Reflection of Molecules

A more detailed description of the mechanisms involved in modeling the atmospheric drag effects can be obtained if we analyze in more depth the interaction of gas molecules with the satellite, that is separated in a first *absorption* phase where the molecules collide and are trapped by the surfaces and a subsequent *diffuse reflection.* The main hypothesis is that atmospheric molecules are in *free molecular flow* so they interact with the satellite independently form each other, while the second hypothesis is that their velocity follows a Maxwell distribution, that is:

$$f(u, v, w) = \frac{\rho}{m}(2\pi r T)^{-\frac{3}{2}} e^{-\frac{u^2+v^2+w^2}{2rT}}, \tag{2.4.6}$$

where ρ is the atmospheric density, m is the mean molecular mass, $r = R/m$ is the specific gas constant and T is the absolute temperature.

Regarding the absorption process we start by considering an oriented surface element $d\mathbf{S}$ with velocity $\mathbf{v}$ relative to the atmosphere and a system of axis where $\mathbf{x}_1$ is normal to the surface, $\mathbf{x}_2$ is the projection of $\mathbf{v}$ on $d\mathbf{S}$ and $\mathbf{x}_3 = \mathbf{x}_1 \times \mathbf{x}_2$, then, we can write the relative velocity $\mathbf{v}_r = \begin{Bmatrix} u_r & v_r & w_r \end{Bmatrix}^T$ of molecules with respect to the surface as

$$\begin{aligned} u_r &= u - \mathbf{v}\sin(\theta) \\ v_r &= v - \mathbf{v}\cos(\theta) \\ w_r &= w. \end{aligned}$$

Note that the knowledge of the relative velocity $\mathbf{v}_r$ depends on the complex dynamics of the atmosphere, however, given the high velocities LEO satellites, a reasonable approximation is obtained assuming that the atmosphere co-rotates jointly with the Earth's surface, thus

$$\mathbf{v}_r = \mathbf{v} - \boldsymbol{\omega}_\oplus \times \mathbf{r},$$

where $\mathbf{r}$ is the geocentric satellite position and $\boldsymbol{\omega}_\oplus$ is the Earth's angular velocity. This assumption has been shown to be in good agreement with reality, leading to uncertainties in the drag perturbation of less than 5% [12]. Using (2.4.6) we can define a distribution of relative velocities:

$$f_r(u_r, v_r, w_r) = \frac{\rho}{m}(2\pi r T)^{-\frac{3}{2}} e^{-\frac{u_r^2+v_r^2+w_r^2}{2rT}}, \tag{2.4.7}$$

then, the momentum imparted on the surface by a single gas particle of mass m with relative velocity $\mathbf{v}_r$ will be

$$d\mathbf{p} = m\mathbf{v}_r$$

and we are able to calculate the number of molecules with velocity $\mathbf{v}_r + \Delta\mathbf{v}_r$ that collide with the surface over a time dt as

$$f_r(u_r, v_r, w_r) du_r dv_r dw_r V, \tag{2.4.8}$$

with V being the volume containing those molecules, that is

$$V = -\mathbf{v}_r \cdot d\mathbf{S}dt = -u_r dtdS.$$

Consequentially, the resulting force acting on $d\mathbf{S}$ can be computed as an integral over the distribution of relative velocity components

$$d\mathbf{F} = \int\int\int_{vel} -mu_r dS f_r(u_r, v_r, w_r) du_r dv_r dw_r \mathbf{v}_r, \tag{2.4.9}$$

and considering that only the frontal part of the satellite gets hit by the gas molecules, which is especially true the case for free molecular flow regime, we can integrate (2.4.9) only for the relative velocities with $u_r < 0$, obtaining

$$\begin{aligned}
\frac{dF_1}{dS} &= -\frac{\rho \mathrm{v}^2}{2\sqrt{\pi}s^2}\left[s\sin(\theta)e^{-s^2\sin^2(\theta)} + \sqrt{\pi}\left(\frac{1}{2} + s^2\sin^2(\theta)\right)\big(1 + \operatorname{erf}(s\sin(\theta))\big)\right] \\
\frac{dF_2}{dS} &= -\frac{\rho \mathrm{v}^2\cos(\theta)}{2\sqrt{\pi}s}\left[e^{-s^2\sin^2(\theta)} + \sqrt{\pi}s\sin(\theta)\big(1 + \operatorname{erf}(s\sin(\theta))\big)\right] \\
\frac{dF_3}{dS} &= 0,
\end{aligned} \tag{2.4.10}$$

where the *velocity factor* s and the function $\operatorname{erf}(x)$ are defined as

$$s = \frac{\mathrm{v}}{\sqrt{2rT}} \qquad \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{x^2}\, dx.$$

As we can see, equations (2.4.10) express the effects of atmosphere-satellite interactions in terms of pressures acting normal and parallel to the surface (directions 1 and 2 of the aforementioned axes), we can then define the forces acting on the satellite due to absorption and separate them into *drag* and *lift* components as

$$\begin{aligned}
d\mathbf{D} &= -\frac{1}{2}\rho \mathrm{v}^2 dS\left[\frac{e^{-s^2\sin^2(\theta)}}{s\sqrt{\pi}} + \sin(\theta)\left(1 + \frac{1}{2s^2}\right)\big(1 + \operatorname{erf}(s\sin(\theta))\big)\right]\hat{\mathbf{e}}_{\mathrm{v}} \\
d\mathbf{L} &= -\frac{1}{2}\rho \mathrm{v}^2 dS\left[\frac{\cos(\theta)}{2s^2}\big(1 + \operatorname{erf}(s\sin(\theta))\big)\right]\hat{\mathbf{e}}_{\mathrm{n}},
\end{aligned} \tag{2.4.11}$$

where the two unit vectors $\mathbf{e}_{\mathrm{v}} = \mathbf{v}/\mathrm{v}$ and $\mathbf{e}_{\mathrm{n}} = \hat{\mathbf{e}}_{\mathrm{v}} \times \mathbf{x}_3$ represent the directions parallel and normal to the relative velocity.

Let us now proceed with the process of diffuse re-emission. Here, the main hypothesis is that absorbed molecules are trapped on the satellite's surfaces for a period that is long enough to surpass the time associated to the mean free path at that particular altitude, in other words, we can treat the molecules on the satellite's surface as an ideal gas in equilibrium. Then, if we also assume that the emission is diffuse over an hemisphere, thus equal in all directions, the only aerodynamic force arising will be given by the re-emission pressure p_r, which will be half of the gas' partial pressure p_p in contact with the surface. We start by computing the number of absorbed molecules per unit of time and unit of area as

$$\begin{aligned}
N_a &= \int_{-\infty}^{\infty} dw_r \int_{-\infty}^{\infty} dv_r \int_{-\infty}^{0} -u_r f_r(u_r, v_r, w_r) dU_r \\
&= \frac{\rho}{m}\sqrt{\frac{rT}{2\pi}}\left[e^{-s^2\sin^2(\theta)} + \sqrt{\pi}s\sin(\theta)\big(1 + \operatorname{erf}(s\sin(\theta))\big)\right],
\end{aligned} \tag{2.4.12}$$

then, if we assume that there are no chemical reactions and that mass is conserved the number of absorbed molecules must be equal to the number of reflected molecules, which can be computed from (2.4.12) with $s = 0$:

$$N_r = \frac{\rho_p}{m}\sqrt{\frac{rT_p}{2\pi}}. \tag{2.4.13}$$

Now, using the identity equation $N_a = N_r$ we can find the density ρ_p to be

$$\rho_p = \rho\sqrt{\frac{T}{T_p}}\left[e^{-s^2\sin^2(\theta)} + \sqrt{\pi}s\sin(\theta)\big(1+\operatorname{erf}(s\sin(\theta))\big)\right], \tag{2.4.14}$$

and we can obtain the re-emission diffuse pressure recalling that it is half of the partial pressure p_p, which in turn is computed with the ideal gas law $p_p = \rho_p r T_p$, thus

$$p_r = \frac{1}{2}\rho r\sqrt{TT_p}\left[e^{-s^2\sin^2(\theta)} + \sqrt{\pi}s\sin(\theta)\big(1+\operatorname{erf}(s\sin(\theta))\big)\right]. \tag{2.4.15}$$

Finally, we can now combine equations (2.4.11) and (2.4.15) to get the total aerodynamic force acting on the satellite due to absorption and diffuse reflection, which is compactly written as

$$\mathbf{F}_{aero} = -\frac{1}{2}\rho \mathrm{v}^2 S\left(C_D\hat{\mathbf{e}}_{\mathrm{v}} + C_L\hat{\mathbf{e}}_{\mathrm{n}}\right), \tag{2.4.16}$$

where C_D is the aforementioned drag coefficient while C_L is the *lift coefficient.* Equation (2.4.16) highlights how the drag and lift coefficients are dimensionless parameters that enclose every process that is related to the atmosphere-surface interaction and for simple geometrical shapes, like planes, spheres and cylinders, they can be calculated analytically. As a first approximation for preliminary studies, artificial satellites subjected to atmospheric drag are often modelled as spheres, referred to as the *cannonball* model, while for demanding application a more suitable *box-wing* model should be implemented in which the satellite complex geometry is approximated as a collection of several simpler shapes (planes, spheres and cylinder precisely). Here, the total aerodynamic force is computed as the sum of every single element contribution and considering that for most applications the lift component can be neglected, we arrive at the following expression for the drag acceleration

$$\begin{aligned}\ddot{\mathbf{r}}_D &= -\frac{1}{2}\rho \mathrm{v}_r^2\left(\frac{C_D A}{m}\right)\hat{\mathbf{e}}_{\mathrm{v}}\\ &= \frac{1}{2}\rho \mathrm{v}_r^2 B\hat{\mathbf{e}}_{\mathrm{v}},\end{aligned} \tag{2.4.17}$$

and note how the *ballistic coefficient* has been introduced as

$$B = \frac{C_D A}{m}.$$

This coefficient is a typical parameter used to give and idea of the satellite's susceptibility to the drag perturbation and is often estimated in POD applications. If a satellite has a high ballistic coefficient it means that its surface area or drag coefficient is high compared to his mass, meaning that aerodynamic forces can have a strong effect over its inertia and can greatly affect its trajectory. Moreover, if the satellite is not spherical and follows a particular attitude law, the ballistic coefficient becomes a function of time $B(t)$ which leads to significant complications in predicting and estimating its orbit.

Lastly, as was mentioned at the beginning, other than the atmosphere-surface interaction and the variability of $B(t)$, another major challenge in precise modelling of the drag perturbation is the knowledge of the atmosphere's physical properties, and for what concerns the computation of drag acceleration using equation (2.4.17), it really comes down to the determination of the atmosphere density ρ, which in itself is a considerably onerous task. Changes in atmospheric density values are mainly due to the combined effect of three different aspects [18]. The first is related to the nature of the atmosphere's molecular composition and properties arising from the ideal gas law and the hydrostatic equation, from which a simple constant exponential model for the atmosphere can be derived where the density decreases exponentially with altitude:

$$\rho = \rho_0 e^{-\frac{h_{ellp}-h_0}{H}}, \tag{2.4.18}$$

where ρ_0 is a reference density at a reference altitude h_0, h_{ellp} is the altitude above the ellipsoid and H is a scale factor that changes along different altitude bands. Equation (2.4.18) describes the *exponential model* and although it might be a good approximation for design studies it is far from being sufficient for POD applications. Besides the fact that even a constant time invariant atmospheric model should account for latitudinal and longitudinal variations, mainly because of the equatorial bulge and massive mountain groups, a more sophisticated model must also consider temporal variations of the atmospheric properties, that can be summarized in the following categories:

- **Diurnal variations**: This variation occurs daily and it's caused by the heating of atmosphere portion under direct sunlight, which causes an atmospheric bulge that elevates high density layers to higher altitudes. Note that due to Earth rotation, this bulge is not directly under the sun but is found at longitudes where the local time is around 2 p.m.

- **Incident solar flux**: The incident radiation emitted by the Sun directly affects the upper atmosphere that is nearly instantaneously heated by the Extreme Ultra-Violet radiation (EUV) causing a variation in its density. Consequentially, it becomes essential to determine the intensity of the solar activity, but since the Earth's atmosphere blocks these EUV radiation, it is not possible to directly measure it from the surface. Nonetheless, the atmosphere is transparent to incident solar radiation with a wavelength of 10.7 cm which is actually correlated to EUV, for this reason, the solar flux is quantified in terms of the $F_{10.7}$ index which is measured in *Solar Flux Units*, where

  $$1\ \mathrm{SFU} = 1 \times 10^{-22} \frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{Hz}}.$$

  Typical values for $F_{10.7}$ can range from 70 to more than 300 SFU.

- **Geomagnetic activity**: The Earth's magnetic field acts as a barrier that shields the surface from high energy ionizing particles emitted by the Sun. However, this charged particles still collide with the upper layers of the atmosphere, which again, translates into heating that increases the atmospheric density at higher altitudes. These density variations can then be related to temporal variations in the Earth and Sun magnetic field, that must be measured to determine the heat generated, for this purpose, the *planetary index* K_p is used as an indicator of the geomagnetic activity especially below the auroral zones where ionization is the strongest. K_p is quasi-logarithmic, and its linear counterpart, the *planetary amplitude* a_p, is measured every three hours, then, eight a_p values are averaged to compute the *daily planetary amplitude* A_p. This effect is particularly noticeable for satellites at altitudes between 300 km and 1000 km.

- **Sun spots cycle**: Often known as the *Solar Cycle*, this is the most notable periodicity in the solar activity with solar maxima and minima repeating roughly every 11 years.

- **Solar rotation cycle**: Within the 11 years cycle, considerable variations in the $F_{10.7}$ value occur at higher frequencies due to the 27 day synodic rotation of the Sun, that drags different active solar regions with itself, increasing the variability of the observed incident flux.

In addition to these fluctuations, other minor and difficult to predict changes in the atmospheric density are caused by winds, random fluctuations like solar flares or magnetic storms, ocean tides and even atmospheric tides.

This brief discussion already suggests the difficulty in determining an atmospheric density model that is suitable for POD applications. From the beginning of the space age, the atmosphere density has been inferred from observations in the evolution of satellites motion, but rapid technological development in satellite and ground system instrumentation allowed direct measurements of the upper atmosphere composition and temperature, as well as ionic and electron content. This collection of data allowed the development of several high spatial and temporal resolution models for the atmosphere such as the Jacchia-Roberts (1971) or the more recent NRLMSISE-00, which gives several atmospheric properties like temperature and composition depending on the given temporal and geographical information as well as daily $F_{10.7}$ and A_p indices.

2.5 Solar Radiation Pressure Model

Solar Radiation Pressure (SRP) is another non-conservative perturbation acting on Earth orbiting satellites whose detailed modeling is decisive for POD applications. Unlike other perturbations this effect is essentially constant for all altitudes given that the Earth-Sun distance is by far greater than typical semi-major axis values for Earth satellites. Furthermore, its value is highly influenced by probable periods of eclipse in which the satellite is shadowed by the Earth, passing through regions of *penumbra* where the radiation pressure is significantly fainter, to regions of *umbra* where it is completely absent. Considering typical altitudes for GNSS satellites and a representative magnitude of $10^{-8}\,\mathrm{m/s^2}$, SRP can lead to perturbations in the range of 100 m over one orbital period [20], and for LEO satellites it becomes comparable to the drag perturbation around altitudes of 600 km (with $F_{10.7} = 150\,\mathrm{SFU}$ and $\rho = 1.2 \times 10^{-13}\,\mathrm{kg/m^3}$).

The mechanism by which this radiation pressure is able to produce such perturbations is that of momentum exchange between the impinging photons and the satellite's surfaces, and this interaction can be classified into three different processes that together give the total perturbation: *absorption*, *specular reflection* and *diffuse reflection*. We start by considering that from Einstein's theory of special relativity the energy of a particle can be expressed as

$$E = \sqrt{(cp)^2 + (m_0c^2)^2}, \tag{2.5.1}$$

where:

- c is the speed of light in vacuum;
- p is the particle's momentum;
- m_0 is its rest mass;

and since for a photon $m_0 = 0$:

$$E = cp. \tag{2.5.2}$$

Moreover, from the Planck-Einstein relation we know that a photon's energy is proportional to its frequency f:

$$E = hf, \tag{2.5.3}$$

thus, comparing (2.5.2) and (2.5.3) we get an expression for its momentum p

$$p = \frac{hf}{c}, \tag{2.5.4}$$

which also appears to be proportional to the photon's frequency and h is the Planck's constant. This result tells us that the photon carries a certain amount of momentum with itself that is completely ceded to the hit surface in the case of total absorption. Let us consider now the case of multiple photons with different energies hitting a surface of unit area over a period of unit time Δt at 1 Astronomical Unit (AU) distance from the Sun. If we define the average number of photons with frequency f as $n(f)$ we can write

$$p_n = n(f)\frac{hf}{c}\Delta t, \tag{2.5.5}$$

then, derivation with respect to time leads to the momentum exchange per unit area per unit time

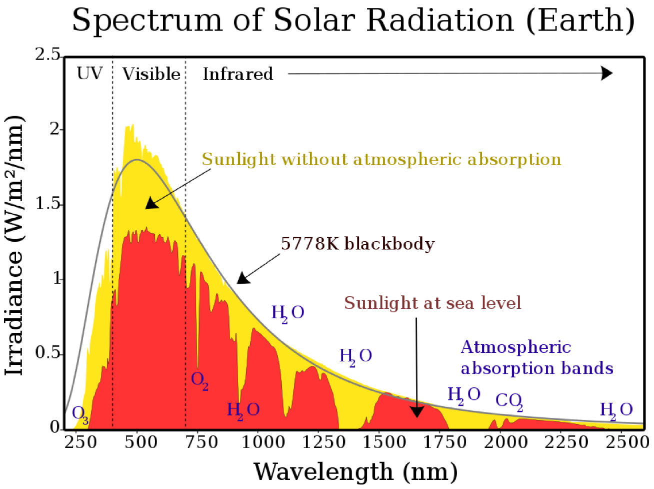

Figure 2.5.1: Solar Irradiance Spectrum

$$\frac{dp_n}{dt} = n(f)\frac{hf}{c}, \tag{2.5.6}$$

which is a force per unit area, i.e., a pressure acting on the impinged surface. We can now integrate (2.5.6) over all characteristic frequencies of the solar electromagnetic spectrum, figure 2.5.1, obtaining the total pressure exerted on a surface at 1 AU from the Sun

$$P_\odot = \int_{f_{\mathrm{SUN}}} \frac{dp_n}{dt}\,df = \int_{f_{\mathrm{SUN}}} \frac{h}{c}n(f)f\,df = \frac{E_\odot}{c}, \tag{2.5.7}$$

and we have shown how

$$E_\odot = \int_{f_{\mathrm{SUN}}} hn(f)f\,df \tag{2.5.8}$$

gives the *solar irradiance* $E_\odot$, often called *solar constant*, that at 1 AU has a value of 1361 W/m², however, noticeable variations in this value may arise from solar activity fluctuations mentioned in Section 2.4. Interestingly, we can compute the value of solar radiation pressure at 1 AU considering that in vacuum $c = 299792458\,\mathrm{m/s}$:

$$P_\odot = \frac{E_\odot}{c} = 4.54 \times 10^{-6}\,\frac{\mathrm{N}}{\mathrm{m}^2}.$$

There are many factors involved in determining the overall effect of SRP in the satellite's trajectory other than the variability of $E_\odot$, the major ones being the surfaces optical properties, that dictate the nature of the photon-satellite interaction, and the spacecraft's orientation that exposes variable cross sections to the striking radiation. In any case we begin with a few important assumptions that will guide the development of the SRP model:

1. The optical surface properties are completely defined by the tree dimensionless coefficients α, δ and ρ which are respectively the coefficient of *absorption*, *diffuse reflectivity* and *specular reflectivity*, whereas no transmissivity coefficient is present since we assume that no component of the incident radiation gets transmitted through the surface;

2. Regarding the diffusive reflection the surface behaves like a perfect Lambertian body, meaning that the reflected radiation scatters equally in all directions reduced by a factor of $\cos(\theta)$, with θ being the angle between the incident direction and the normal to the surface, hence obeying the Lambert's cosine law.

Note that the coefficients α, δ and ρ represent respectively the fraction of incident radiation that is absorbed, diffusively reflected and specularly reflected, and since these are assumed to be the only three types of interaction involved it must be true that

$$\alpha + \delta + \rho = 1. \tag{2.5.9}$$

Now, if we consider a surface element dA whose normal unit vector is identified by $\hat{\mathbf{n}}$ and forms an angle θ with the surface-Sun direction identified by $\hat{\mathbf{e}}_\odot$, we are able to write the SRP contributions from absorption and reflection components separately. For the absorbed fraction the entire momentum is transferred to the surface, hence the resulting force is directed anti-parallel to $\hat{\mathbf{e}}_\odot$, that is (figure 2.5.2):

$$d\mathbf{F}_\alpha = -\frac{E_\odot}{c} dA \cos(\theta) \alpha\, \hat{\mathbf{e}}_\odot, \tag{2.5.10}$$

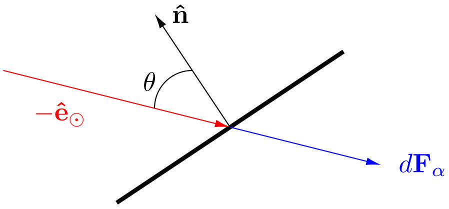

Figure 2.5.2: Absorbed incident radiation.

while for the case of perfectly specular reflection no momentum exchange occurs in the direction parallel to the surface, so the resulting force is directed anti-parallel to the surface's normal and from simple considerations in conservation of momentum we get (figure 2.5.3)

$$d\mathbf{F}_\rho = -\frac{E_\odot}{c} dA \cos(\theta) 2 \cos(\theta) \rho\, \hat{\mathbf{n}}. \tag{2.5.11}$$

The diffuse reflection modeling is a little more complicated since the force that stems from it is directed along both $\hat{\mathbf{n}}$ and $\hat{\mathbf{e}}_\odot$ directions, where this happens because the incoming radiation is first absorbed by the surface and later re-emitted equally in all directions. In order to evaluate the effect of the diffuse reflection alone we start by assessing the total diffuse radiation intensity over an hemisphere centered at the surface element. Keeping in mind that this intensity will always be reduced by the factor $\cos(\theta)$ we can expect it to be proportional to the following integral:

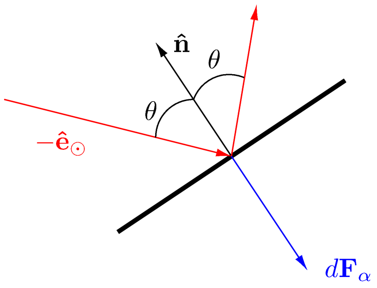

Figure 2.5.3: Reflected incident radiation.

$$\begin{aligned}\int_0^{2\pi}\int_0^{\pi/2}\sin(\theta)\cos(\theta)\,d\theta\,d\lambda &= 2\pi\int_0^{\pi/2}\frac{\sin(2\theta)}{2}\,d\theta\\ &= \pi\left[\frac{\cos(2\theta)}{2}\right]\Bigg|_0^{\pi/2}\\ &= \pi\end{aligned} \tag{2.5.12}$$

and to obtain the component of this radiation that is perpendicular to the surface we can compute the integral as [16]

$$\begin{aligned}\int_0^{2\pi}\int_0^{\pi/2}\sin(\theta)\cos^2(\theta)\,d\theta\,d\lambda &= 2\pi\int_0^{\pi/2}\sin(\theta)\cos^2(\theta)\,d\theta\\ &= 2\pi\left[-\frac{\cos^3(\theta)}{3}\right]\Bigg|_0^{\pi/2}\\ &= \frac{2}{3}\pi,\end{aligned} \tag{2.5.13}$$

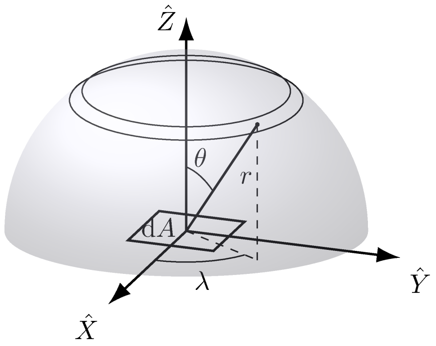

Figure 2.5.4: Spherical coordinates for the description of hemispherical radial flux interception from a Lambertian surface.

then, we divide (2.5.13) by (2.5.12) to conclude that the diffused radiation along the normal to the surface is equal to 2/3 of the total diffused radiation, and we can write the SRP force generated by diffuse reflection as (figure 2.5.5)

$$d\mathbf{F}_\delta = -\frac{E_\odot}{c} dA \cos(\theta)\delta \left(\hat{\mathbf{e}}_\odot + \frac{2}{3}\hat{\mathbf{n}}\right), \tag{2.5.14}$$

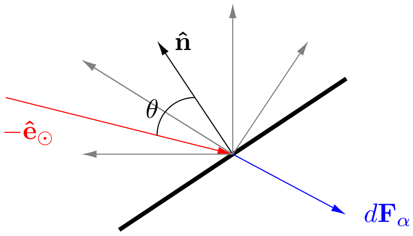

Figure 2.5.5: Diffusely reflected incident radiation.

and finally, equations (2.5.10), (2.5.11) and (2.5.14) can now be grouped into a single equation for the force acting on a surface element dA due to solar radiation pressure accounting for all the three interactions involved:

$$d\mathbf{F}_{SRP} = -\frac{E_\odot}{c} dA \cos(\theta) \left[(\alpha + \delta)\,\hat{\mathbf{e}}_\odot + \left(2\rho\cos(\theta) + \frac{2}{3}\delta\right)\hat{\mathbf{n}}\right]. \tag{2.5.15}$$

The previous equation is a good representation of the force acting on an infinitesimal planar surface dA due to the momentum exchange with the impinging photons, and we saw how different surface properties in terms of the coefficients α, ρ and δ lead to different results. The remaining step would be to compute the finite total force $\mathbf{F}_{SRP}$ acting on the satellite and conceivably this can be done by integrating (2.5.15) over the entire satellite surface, which may be trivial for simple shapes like spheres and cylinders, but becomes unimaginable for satellites of complex geometries composed of several different materials all with their own optical properties. In practice, the best workaround to this complex task is the development of a suitable box-wing model (already mentioned in Section 2.4.1), also known as *macro-model*, where the intricate satellite aspect is approximated by a six-faced box in place of the main body and two rectangles representative of the solar panels. Even more detailed representations, known as *micro-model*, may discretize complex features as an ensemble of several simple shapes like small planar surfaces A_i of arbitrary dimension each identified by its normal unit vector $\hat{\mathbf{n}}_i$ and its optical properties α_i, ρ_i and δ_i, so the finite force becomes

$$\mathbf{F}_{SRP} = -\frac{E_\odot}{c}\sum_{i=1}^{N} A_i \cos(\theta_i)\left[(\alpha_i + \delta_i)\hat{\mathbf{e}}_\odot + \left(2\rho_i\cos(\theta_i) + \frac{2}{3}\delta_i\right)\hat{\mathbf{n}}_i\right]. \tag{2.5.16}$$

Interestingly, we may perform integration of (2.5.15) for a spherical satellite, whereas only an hemispherical portion is directly illuminated by the Sun. If we denote the satellite's radius with R and assume that optical properties are uniform everywhere, using spherical coordinates θ and λ as shown in figure 2.5.4 we can write

$$\mathbf{F}^{sph}_{SRP} = -\frac{E_\odot}{c}\int_0^{\pi/2}\int_0^{2\pi}\left[(\alpha + \delta)\,\hat{\mathbf{e}}_\odot + \left(2\rho\cos(\theta) + \frac{2}{3}\delta\right)\hat{\mathbf{n}}\right] R^2\cos(\theta)\sin(\theta)\,d\theta\,d\lambda, \tag{2.5.17}$$

however, due to the symmetry of the spherical model we expect every component of the reflected radiation (both specular and diffuse) that is perpendicular to the satellite-Sun direction to cancel out, thus, the resulting force will be directed along the $\hat{\mathbf{e}}_\odot$ direction and we multiply the reflective terms by $\cos(\theta)$ before performing integration

$$\mathbf{F}_{SRP}^{sph} = -\frac{E_\odot}{c}\int_0^{\pi/2}\int_0^{2\pi}\left[(\alpha+\delta)+\left(2\rho\cos^2(\theta)+\frac{2}{3}\delta\cos(\theta)\right)\right]\hat{\mathbf{e}}_\odot R^2\cos(\theta)\sin(\theta)\,d\theta\,d\lambda, \quad (2.5.18)$$

note that the term related to absorption has remained unchanged since it was already directed along $\hat{\mathbf{e}}_\odot$. We now proceed to perform integration considering the terms in brackets one at a time. Regarding the absorption component we have

$$\begin{aligned}(\alpha+\delta)\int_0^{\pi/2}\int_0^{2\pi} R^2\cos(\theta)\sin(\theta)\,d\theta\,d\lambda &= (\alpha+\delta)2\pi R^2\int_0^{\pi/2}\cos(\theta)\sin(\theta)\,d\theta\\ &= (\alpha+\delta)2\pi R^2\left[-\frac{\cos(2\theta)}{2}\right]\Bigg|_0^{\pi/2}\\ &= (\alpha+\delta)\pi R^2,\end{aligned} \quad (2.5.19)$$

the specular reflection part leads to

$$\begin{aligned}2\rho\int_0^{\pi/2}\int_0^{2\pi} R^2\cos^3(\theta)\sin(\theta)\,d\theta\,d\lambda &= 4\rho\pi R^2\int_0^{\pi/2}\cos^3(\theta)\sin(\theta)\,d\theta\\ &= 4\rho\pi R^2\left[-\frac{\cos^4(\theta)}{4}\right]\Bigg|_0^{\pi/2}\\ &= \rho\pi R^2,\end{aligned} \quad (2.5.20)$$

and the diffuse reflection gives

$$\begin{aligned}\frac{2}{3}\delta\int_0^{\pi/2}\int_0^{2\pi} R^2\cos^2(\theta)\sin(\theta)\,d\theta\,d\lambda &= \frac{4}{3}\delta\pi R^2\int_0^{\pi/2}\cos^2(\theta)\sin(\theta)\,d\theta\\ &= \frac{4}{3}\delta\pi R^2\left[-\frac{\cos^3(\theta)}{3}\right]\Bigg|_0^{\pi/2}\\ &= \frac{4}{9}\delta\pi R^2,\end{aligned} \quad (2.5.21)$$

we then combine the results from (2.5.19), (2.5.20) and (2.5.21) into equation (2.5.18) obtaining

$$\mathbf{F}_{SRP}^{sph} = -\frac{E_\odot}{c}\pi R^2\left[\alpha+\delta+\rho+\frac{4}{9}\delta\right]\hat{\mathbf{e}}_\odot, \quad (2.5.22)$$

which can be further simplified considering that from equation (2.5.9) we have $\alpha+\delta = 1-\rho$, so

$$\begin{aligned}\mathbf{F}_{SRP}^{sph} &= -\frac{E_\odot}{c}\pi R^2\left[1+\frac{4}{9}\delta\right]\hat{\mathbf{e}}_\odot\\ &= -\frac{E_\odot}{c}A_c C_R\hat{\mathbf{e}}_\odot.\end{aligned} \quad (2.5.23)$$

This equation we just derived is a very common form for the modeling of solar radiation pressure effects on satellites and we introduced the *radiation pressure coefficient* C_R, also know as the *satellite reflectivity*. Even though equation (2.5.23) has been derived for the particular case of a spherical satellite, in which we found that $C_R = 1+(4/9)\delta$, this result is still useful for satellites with arbitrary constant or time-varying cross-section A_c. In fact, for a more general case, the C_R coefficient becomes a scaling factor for the SRP acceleration that encapsulates all the effects related to surface properties, shapes and orientation, with the benefit of having the resulting force acting only along $\hat{\mathbf{e}}_\odot$, and for this reason, this parameter is often estimated as a result of the orbit determination process. Given these considerations, we can write the acceleration due to SRP on a satellite of arbitrary cross-section A_c once its mass is known as

$$\ddot{\mathbf{r}}_{SRP} = -\frac{E_\odot}{c}\frac{1AU^2}{r_\odot^2}\frac{A_c}{m}C_R\hat{\mathbf{e}}_\odot, \tag{2.5.24}$$

where we also introduced the ratio between 1 Astronomical Unit and the actual Sun-satellite distance $r_\odot$ since we assume that $E_\odot$ is the fixed value for the solar constant at 1 AU, where variations in the distance due to the Earth's eccentric orbit, can lead to variations in solar radiation pressure of about ±3%.

2.5.1 Eclipse Condition

As mentioned at the beginning of this Section, during the evolution of its orbit, a satellite may pass trough periods of partially or totally absent solar illumination as it gets obscured by the Earth or, less frequently, by the Moon, and this is true for most LEO satellites with the evident exception represented by constantly illuminated sun-synchronous ones. There exist two main approaches to the modeling of eclipses, the *cylindrical model* and the *conical model*. In the first case, the sun is assumed to be infinitely distant, so that all incoming rays are parallel and the Earth forms a cylindrical shadow, while for the latter case the finite Sun' dimension and distance are taken onto account, and a conical shadow arises. A simple scheme depicting the situation is given in figure 2.5.6, where inside the cone generated by the Earth's Shadow we can identify regions of completely absent illumination called *umbra* and regions of partial illumination called *penumbra*. Here, we focus mainly on the conical model since it is the one implemented when modeling the SRP with the Fortran libraries.

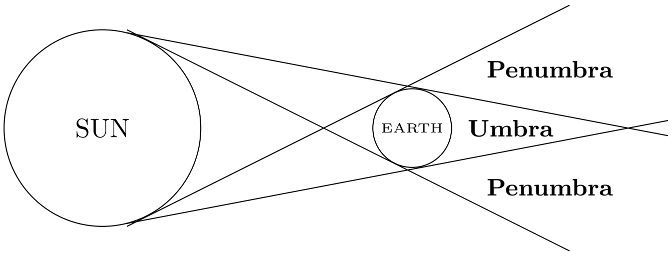

Figure 2.5.6: Schematic representation of the conical eclipse model. Note how there are both umbra and penumbra regions.

If a satellite encounters frequent and relatively long periods of eclipse, equation (2.5.15) becomes insufficient to accurately model the effects of SRP since it assumes that the spacecraft is constantly hit by rays of light, to fix this, a simple solution is to add a shadow function τ to the equation

$$\ddot{\mathbf{r}}_{SRP} = -\tau\frac{E_\odot}{c}\frac{1\,\mathrm{AU}}{r\odot^2}\frac{A_c}{m}C_R\hat{\mathbf{e}}_\odot, \tag{2.5.25}$$

where

- $\tau = 0$ if the satellite is in umbra regions,
- $0 < \tau < 1$ for regions of penumbra,
- $\tau = 1$ if the satellite is completely illuminated.

2.6 Earth Radiation Pressure Model

The Earth Radiation Pressure (ERP) model described here, is based on the work carried out by Knocke, Ries and Tapley [13] for the University of Texas Orbit determination program (UTOPIA), where precision modeling was aimed at explaining the anomalous mean secular decrease of 1.1 mm/day in semi-major axis of the LAGEOS satellite. Unexplained mean secular motions in the determination of a satellite's orbit can lead to undesirable inaccuracies in parameters estimation, since these unmodeled effects are inevitably absorbed by parameters that are estimated through filtering techniques. For this reason, a force model as detailed as possible is certainly preferable. It is interesting to note how the model proposed by Knocke, and here described in detail, dates back to 1988, but it is still widely used today to assess the ERP impact on the dynamics of an earth orbiting satellite. Rodriguez-Solano et al. [23], deeply investigated the non-negligible effects that earth radiation pressure has on the orbit determination of GPS satellites, where the comparison between precise Satellite Laser Range (SLR) measurements and the orbit predicted without ERP, showed a consistent radial bias of several centimeters as well as biases in the estimated position of ground stations, whereas the inclusion of ERP recovered this radial bias up to $1-2$ cm. Considering that GPS orbits are way over the LEO upper boundary (at more than 20000 km in altitude), for LEO satellites the effects of ERP can be expected to be much higher. Wyatt, Lochry and others predicted in 1963 that the magnitude of earth radiation pressure could be as high as 35% of the solar radiation pressure (SRP) for very low satellites ($200-300$ km), while it can be expected to reach values around 10% to 25% of SRP for the majority of LEO satellites. Another aspect that can affect the sensitivity to this perturbation is orbit eccentricity. For orbits with low eccentricity the effects of ERP acceleration can average out along the orbit and can be largely absorbed by the estimation of other model parameters, while for highly eccentric orbits the effects can give rise to along-track accelerations that do not average out during one orbit, thus compromising the estimation and recovery of certain parameters. These values suggest that accelerations due to ERP can be higher than expected, and along with empirical evidences of its relevance thanks to onboard accelerometers, motivate the efforts in developing an effective mathematical model to account for this dim force in the overall propagation of the satellite's trajectory.

For LEO satellites the Earth itself represents the major source of radiant flux after the sun, but while the latter can be considered as an infinitely distant and almost uniform source of energy, the former must be treated as an extended radiating body at a finite distance, which radiant flux may be significantly variable. This variability is based on the angle of incidence of the Sun relative to the earth surface, but also on the reflective and emissive properties of the Earth-atmosphere system that depends on season, geography, atmospheric composition, cloud coverage and surface type that is being considered as the source of radiant energy. From this it follows that while the mechanism behind the origin of the accelerations imparted to the satellite still consists in impinging photons and their momentum exchange, as for Solar radiation pressure, the main aspects peculiar to Earth radiation pressure arise from the difference in the source of this radiant energy. ERP involves two main fluxes related to two different wavebands as a consequence of the mechanism that regulates the balance of the energy incident on the Earth. The first waveband is related to the percentage of radiation from the sun that is immediately reflected by the Earth surface, consequently, this radiant flux displays a spectrum that is similar to the Sun spectral irradiance, figure 2.5.1, meaning that this source of radiant energy comprises wavelengths from 0.2 up to 4 μm which is referred to as the *broadband shortwave* or just *shortwave* radiation. It is important to underline how this waveband not only comprises the entire visible band ($0.38-0.75$ μm), but also a small portion of the UV spectrum ($0.2-0.38$ μm) and a consistent portion of the infrared radiation called *near infrared* ($0.75-4$ μm), which distinguishes it from the classical definition of *albedo* that considers the visible radiation only. The second waveband involved in this mathematical model is related to the portion of the incident energy from the sun that is absorbed by the Earth and later re-emitted in the infrared spectrum, which comprises wavelengths from 4 μm up to 50 μm and is therefore known as the *longwave infrared radiation* or *heat radiation* that is considered separately from the infrared portion contained in the aforementioned shortwave band. The differences between these two waveband not only rise from the different physical phenomenon from which they are generated, but also from the fact that while the shortwave radiation is clearly present only at the Earth surface under direct sunlight and exhibit variations due to surface-atmosphere characteristics and anisotropic effects, the longwave radiation is always present even when the satellite experiences periods of eclipse, and is almost constant due to the high thermal inertia of the Earth. The mathematical model based on the distinction between *shortwave* and *longwave* wavebands that will be described in detail below, has the advantage to exhibit a particular adaptability to a wide variety of Earth orbits, thanks to its generality and flexibility if compared to many analytical solutions that were developed even decades before the work of Knocke. These analytical solutions were based on simplifying assumptions, like constant albedo, radial acceleration only (Wyatt) or independence from solar position (Levin), whose purpose is to simplify the surface integrals that one encounters in determining the radiative exchange from a three dimensional extended body to the satellite, but drastically reduce their field of applicability.

2.6.1 Diffuse Earth Radiation Pressure Model

The main feature of the presented method, that also contributes to its generality and flexibility, is the fact that the total acceleration on the satellite due to Earth radiation pressure is computed as the vector sum of all elements in which the instantaneously visible Earth surface is subdivided into. These elements are approximated as planar surfaces orthogonal to the Earth mean-radius passing through their center, and here, the satellite is approximated as a constant cross-section sphere, often referred to as the *cannonball* model. It is readily highlighted how a more detailed satellite model such as a *box-wing* becomes crucial for more detailed analyses, where studies like Rodriguez-Solano et al. already show how such model can give deep insight in the effects of ERP that could not be perceived with a simpler cannonball model, where the presence of solar panels with a particular attitude profile leads to consequent dynamical effects. The other significant feature of the model is that it assumes that both shortwave and longwave components obey Lambert's law of diffuse radiation, which may sound quite restrictive at first, but it is actually a rather justified assumption. As a matter of fact the only portions of earth surface that really act in an almost purely specular reflective manner, as opposed to pure diffusion, are bodies of calm water that certainly are a negligible part of the total earth surface, in addition, satellite radiometric measurements confirm that most surfaces including snow, oceans, clouds and ice, behave mostly in a diffusive fashion. Only at high solar zenith angles some particular surfaces tend to behave more specularly than diffusively, but high solar zenith angles also correspond to very low magnitude of Earth radiation pressure, all of which support the assumption of a purely diffusive radiation model.

First, let's begin with the assumption that every surface element in which the total visible surface is divided, has the same albedo and emissivity properties of the corresponding true Earth portion of the surface-atmosphere system with the same area. Let us call this planar element as $d\mathrm{A}$, and A_i the surface that intercepts the flux radiated by $d\mathrm{A}$. Consider its radiance L, associated with the shortwave or longwave component, that can be written as

$$\mathrm{L} = \frac{d\Phi}{d\Omega d\mathrm{A}\cos\alpha}, \tag{2.6.1}$$

where

- $d\Phi$ is the portion of the total power [Watts] radiated by the surface element $d\mathrm{A}$, that is intercepted by A_i,
- $d\Omega$ is the solid angle defined as the ration between the intercepting surface A_i and the square of its distance r^2 from $d\mathrm{A}$: $d\Omega = \mathrm{A}_i/r^2$ measures in *steradians* [sr],
- α is the angle between the normal to the element $d\mathrm{A}$ and the line connecting $d\mathrm{A}$ to A_i also called the *viewing angle*,

thus, the radiated flux (power) intercepted by the area A_i is

$$\begin{aligned} d\Phi &= \mathrm{L}d\Omega d\mathrm{A}\cos\alpha \\ &= \mathrm{L}\frac{\mathrm{A}_i}{r^2}d\mathrm{A}\cos\alpha. \end{aligned} \tag{2.6.2}$$

As mentioned before, we wish to consider the effects of ERP on a satellite which is assumed to behave like a constant cross-section sphere. In this case, let the radius of this sphere be R_s and A_s its total area that is subdivided into smaller elements called $\Delta\mathrm{A}_s$. If the normal to this satellite's element makes an angle α_s with the line from $d\mathrm{A}$ to $\Delta\mathrm{A}_s$, following equation (2.6.2), the portion of the flux intercepted by $\Delta\mathrm{A}_s$ is

$$\Delta(d\Phi) = \mathrm{L}\frac{\Delta\mathrm{A}_s}{r^2}\cos\alpha_s d\mathrm{A}\cos\alpha, \tag{2.6.3}$$

and the total portion of the flux intercepted by the satellite can be determined by integrating (2.6.3) over the hemisphere visible to $d\mathrm{A}$ under the hypothesis that $r \gg \mathrm{R}_s$. Using spherical coordinates with constant radius R_s, polar angle α_s and azimuth angle θ, the infinitesimal area element $\Delta\mathrm{A}_s$ can be expressed as

$$\Delta\mathrm{A}_s = \mathrm{R}_s^2\sin\alpha_s d\alpha_s d\theta, \tag{2.6.4}$$

thus, the total flux emitted by the Earth surface element $d\mathrm{A}$ intercepted by the satellite is found by performing the following integration:

$$d\Phi = \int_{\theta=0}^{2\pi}\int_{\alpha_s=0}^{\pi/2}\frac{\mathrm{L}}{r^2}\mathrm{R}_s^2\sin\alpha_s\cos\alpha_s d\mathrm{A}\cos\alpha\, d\alpha_s\, d\theta, \tag{2.6.5}$$

that factoring out constants leads to

$$\begin{aligned} d\Phi &= \mathrm{L}\frac{\mathrm{R}_s^2}{r^2}d\mathrm{A}\cos\alpha\int_{\theta=0}^{2\pi}d\theta\int_{\alpha_s=0}^{\pi/2}\sin\alpha_s\cos\alpha_s\, d\alpha_s \\ &= \mathrm{L}\frac{\mathrm{R}_s^2}{r^2}d\mathrm{A}\cos\alpha\, 2\pi\int_{\alpha_s=0}^{\pi/2}\sin\alpha_s\cos\alpha_s\, d\alpha_s \\ &= \mathrm{L}\frac{\mathrm{R}_s^2}{r^2}d\mathrm{A}\cos\alpha\, 2\pi\int_{\alpha_s=0}^{\pi/2}\frac{\sin 2\alpha_s}{2}\, d\alpha_s \\ &= \mathrm{L}\frac{\mathrm{R}_s^2}{r^2}d\mathrm{A}\cos\alpha\,\pi\left[-\frac{\cos 2\alpha_s}{2}\right]_0^{\pi/2} \\ &= \mathrm{L}\frac{\pi\mathrm{R}_s^2}{r^2}d\mathrm{A}\cos\alpha, \end{aligned} \tag{2.6.6}$$

but $\pi\mathrm{R}_s^2$ is the satellite cross sectional area A_c, so

$$d\Phi = \mathrm{L}\frac{\mathrm{A}_c}{r^2}d\mathrm{A}\cos\alpha. \tag{2.6.7}$$

and here, even if the satellite is modeled as a sphere of constant radius, the result is more general and can be applied to any satellite of cross sectional area A_c.

Now, considering that radiance is a quantity dependent on wavelength, and recalling that in this model the Earth radiates in *shortwave* and *longwave* components, the total flux intercepted by the satellite's visible hemisphere can be expressed as

$$d\Phi = \left(\mathrm{L}_{SW} + \mathrm{L}_{LW}\right)\frac{\mathrm{A}_c}{r^2}\, dA\cos\alpha, \tag{2.6.8}$$

where

- L_{SW} represents *shortwave* radiance of the Earth's surface element dA,
- L_{LW} is the *longwave* radiance of the Earth's surface element dA,

and the following steps will be focused on finding proper expressions for these two quantities in terms of *albedo* and *emissivity.*

2.6.2 Shortwave Radiance L_{SW}

In this Section, the component of the total radiance given by the Earth surface element dA associated to the shortwave band $0.2 - 4$ μm, is written in terms of *albedo.* The classical definition for albedo is

$$\mathrm{a} = \frac{\Phi_{hem}}{\Phi_{in}} \tag{2.6.9}$$

where

- Φ_{hem} is the total flux out of dA in the shortwave band,
- Φ_{in} is the solar flux in the shortwave band incident on dA.

Now, Φ_{hem} can be thought as the integral of the infinitesimal portion of flux $d\Phi$ over an hemisphere centered at dA with radius r equal to the element-satellite distance, and since we are only interested in the shortwave component, the integral becomes

$$\Phi_{hem} = \int d\Phi_{SW}, \tag{2.6.10}$$

where, following equation (2.6.8)

$$d\Phi_{SW} = \mathrm{L}_{SW}\frac{d\mathrm{A}_{hem}}{r^2}d\mathrm{A}\cos\alpha, \tag{2.6.11}$$

in which the satellite's cross sectional area A_c has been replaced by the infinitesimal element of hemisphere surface $d\mathrm{A}_{hem}$, since we are interested in computing the total flux intercepted by the whole hemisphere centered at dA. Considering spherical coordinates α and θ similarly to (2.6.6), the infinitesimal hemisphere area becomes

$$d\mathrm{A}_{hem} = r^2 \sin\alpha d\theta d\alpha \tag{2.6.12}$$

and the integral in (2.6.10) leads to

$$\begin{aligned}
\Phi_{hem} &= \int_{\theta=0}^{2\pi}\int_{\alpha=0}^{\pi/2} \mathrm{L}_{SW} d\mathrm{A}\frac{r^2}{r^2}\sin\alpha\cos\alpha\, d\alpha\, d\theta \\
&= \mathrm{L}_{SW} d\mathrm{A}\int_{\theta=0}^{2\pi}\int_{\alpha=0}^{\pi/2} \sin\alpha\cos\alpha\, d\alpha\, d\theta \\
&= \mathrm{L}_{SW} d\mathrm{A}\, 2\pi \int_{0}^{\pi/2} \sin\alpha\cos\alpha\, d\alpha \\
&= \mathrm{L}_{SW} d\mathrm{A}\pi \left[\frac{-\cos 2\alpha}{2}\right]_0^{\pi/2}
\end{aligned} \tag{2.6.13}$$

that finally gives

$$\Phi_{hem} = \pi \mathrm{L}_{SW} d\mathrm{A}. \tag{2.6.14}$$

It is interesting to highlight how the integral calculation was performed assuming that L_{SW} is constant over α and θ, meaning that it does not depend on the particular direction considered, in accordance with what has been said above regarding the pure diffusive nature of the radiation encountering the satellite's surface. If one wish to improve the model admitting the effects of specular reflection, a particular analytical dependence of the radiance L_{SW} on the directions α and θ must be defined.

Now, going back to equation (2.6.9), an expression for the total shortwave solar flux Φ_{in} incident on dA can be simply determined by means of the *local solar irradiance* E_s

$$\mathrm{E}_s = \frac{\Phi_{sun}}{4\pi r_{ES}^2}, \tag{2.6.15}$$

where

- Φ_{sun} is the total solar flux,
- r_{ES} is the Earth-Sun distance,

and the Sun is considered as a source of diffuse radiation. Notice that r_{ES} is intended as the local distance between the Earth and the Sun, thus the *local solar irradiance* is different from the *solar constant* since the latter considers a sphere of fixed radius equal to 1 A.U., and if the normal to the Earth's surface element dA makes an angle θ_s with the line from dA to the sun, then the total incident shortwave solar flux can be expressed as

$$\Phi_{in} = \mathrm{E}_s \cos\theta_s d\mathrm{A}. \tag{2.6.16}$$

At this point, equations (2.6.14) and (2.6.16) can be used in (2.6.9) to obtain

$$\begin{aligned} \mathrm{a} &= \frac{\pi \mathrm{L}_{SW} d\mathrm{A}}{\mathrm{E}_s \cos\theta_s d\mathrm{A}} \\ &= \frac{\pi \mathrm{L}_{SW}}{\mathrm{E}_s \cos\theta_s}, \end{aligned} \tag{2.6.17}$$

that finally gives an expression for the *shortwave radiance* in terms of albedo as

$$\mathrm{L}_{SW} = \frac{\mathrm{a}\mathrm{E}_s \cos\theta_s}{\pi}. \tag{2.6.18}$$

2.6.3 Longwave Radiance L_{LW}

Now that the shortwave radiance has been defined in terms of albedo, in the following Section, the objective is to define the longwave radiance L_{LW} in terms of *emissivity* of the Earth's surface element dA. In general, the emissivity e indicates how a certain body is effective in emitting energy in the form of thermal radiation, and it is expressed as the dimensionless ration between the radiant exitance of the particular surface and the radiant exitance of a black body at the same temperature, which represents the perfect thermal radiation emitter. By this definition

$$\mathrm{e} = \frac{\mathrm{M}_{LW}}{\mathrm{M}_{BB}}, \tag{2.6.19}$$

where in this case

- M_{LW} is the longwave exitance of the surface element dA,
- M_{BB} represents the exitance of dA if Earth was a black body, thus a perfect absorber that re-emits all incident solar radiation equally in all directions.

Similarly to the proceedings of the previous Section, the two exitance appearing in equation (2.6.19) will be expressed in relation to various assumptions in order to obtain an expression for L_{LW}. Knowing that the definition of exitance M for a surface A is the ratio between the total flux exiting the surface and the surface itself

$$\mathrm{M} = \frac{\Phi}{\mathrm{A}}, \tag{2.6.20}$$

the black body exitance M_{BB} can be defined as the solar flux intercepted by Earth divided by the total surface of the Earth

$$\begin{aligned} \mathrm{M}_{BB} &= \frac{\mathrm{E}_s \pi \mathrm{R}_E^2}{4\pi \mathrm{R}_E^2} \\ &= \frac{\mathrm{E}_s}{4}, \end{aligned} \tag{2.6.21}$$

where at the numerator the local solar irradiance E_s is multiplied by the Earth's cross sectional area that intercepts the solar flux, thus R_E is the Earth radius, while at the denominator we find the total Earth's spherical surface reflecting the fact that all this incident energy is entirely re-emitted as thermal radiation in all directions.

Now, the longwave exitance of $d\mathrm{A}$ must be rewritten in terms of the longwave radiance, which is the quantity we are interested in, and again, following the definition of exitance we can write

$$\mathrm{M}_{LW} = \frac{\Phi_{LW}}{d\mathrm{A}}. \tag{2.6.22}$$

Considering that from equation (2.6.8) we have

$$d\Phi_{LW} = \mathrm{L}_{LW} \frac{\mathrm{A}_c}{r^2} d\mathrm{A} \cos\alpha, \tag{2.6.23}$$

the term Φ_{LW} can be computed similarly to (2.6.13) by integrating (2.6.23) over and hemisphere centered at $d\mathrm{A}$ with radius r equal to the surface element-satellite distance, thus, also in this case A_c is replaced by the infinitesimal surface element $d\mathrm{A}_{hem} = r^2 \sin\alpha \, d\theta \, d\alpha$, leading to

$$\begin{aligned} \Phi_{LW} &= \int_{\theta=0}^{2\pi} \int_{\alpha=0}^{\pi/2} \mathrm{L}_{LW} \, d\mathrm{A} \cos\alpha \sin\alpha \, d\alpha \, d\theta \\ &= \mathrm{L}_{LW} \, d\mathrm{A} \int_{\theta=0}^{2\pi} \int_{\alpha=0}^{\pi/2} \cos\alpha \sin\alpha \, d\alpha \, d\theta \\ &= \mathrm{L}_{LW} \, d\mathrm{A} \pi \left[\frac{\cos 2\alpha}{2} \right]_0^{\pi/2} \\ &= \pi \mathrm{L}_{LW} \, d\mathrm{A}, \end{aligned} \tag{2.6.24}$$

and substituting this result into (2.6.22) gives

$$\mathrm{M}_{LW} = \pi \mathrm{L}_{LW}, \tag{2.6.25}$$

that allows to write the longwave radiance in terms of longwave exitance

$$\mathrm{L}_{LW} = \frac{\mathrm{M}_{LW}}{\pi}. \tag{2.6.26}$$

Finally, using (2.6.19) and (2.6.21) into (2.6.26), gives

$$\mathrm{L}_{LW} = \frac{\mathrm{eM}_{BB}}{\pi}, \tag{2.6.27}$$

or, considering that $\mathrm{M}_{BB} = \mathrm{E}_s/4$,

$$\mathrm{L}_{LW} = \frac{\mathrm{eE}_s}{4\pi}. \tag{2.6.28}$$

As with shortwave radiance, the nature of the radiated energy is assumed to be diffuse, allowing to bring L_{LW} out of the integral in (2.6.24). Here, the reason in favour of this hypothesis is the fact that when Earth intercepts the solar flux it does not re-emits this energy immediately in a particular direction, but energy is absorbed and re-emitted in a delayed fashion in all directions, making it nearly isotropic.

Now that both longwave and shortwave radiances have been defined in terms of albedo and emissivity, it is possible to write the total flux exiting the Earth surface element dA intercepted by the satellite's visible hemisphere as

$$d\Phi = \left(\frac{\mathrm{aE}_s \cos\theta_s}{\pi} + \frac{\mathrm{eM}_{BB}}{\pi}\right) \frac{\mathrm{A}_c}{r^2} d\mathrm{A} \cos\alpha, \tag{2.6.29}$$

where, we recall that

- dA is the planar surface element that approximates a small portion of Earth surface with same area and same properties,
- a and e are the *albedo* and *emissivity* of the element of Earth surface dA,
- E_s and $\mathrm{M}_{BB} = \mathrm{E}_s/4$ are the *local solar irradiance* and the Earth *black body exitance* respectively,
- A_c is the satellite's cross sectional area,
- r is the area element-satellite distance,
- θ_s is the Sun azimuth angle, that is the angle between the normal to dA and the line connecting dA to the Sun,
- α is the *viewing angle*, that is the angle between the normal to dA and the line connecting dA to the satellite.

2.6.4 Satellite Model and Total Acceleration due to Earth Radiation

Now that the radiant properties of a single Earth surface element have been conveniently defined in the shortwave and longwave bands, the following steps aim at quantify the dynamical effects that this radiation has on the satellite in terms of total imparted acceleration from all the visible surface elements, but also depending on the assumed optical and emissive properties of the satellite's surfaces. Regarding the satellite model, based on the properties of the surfaces that get radiated, three main different mechanism of surface-radiation interaction can be identified, and the total effect can be modeled as a combination of these three components, similarly to what happens for solar radiation pressure. The first interaction mechanism is associated to the *black body* component, which represents the fraction of incident radiation that is absorbed as if the satellite was an ideal perfect absorber. The related force, always follows the direction of the incident flux, regardless of the particular shape of the impinged surface or its optical properties. The second component is associated to *reflection*, and the interesting aspect is that the relater force does not generally follow the direction of incidence. The two extremes are represented by perfectly specular reflection, in which the arising force is along the direction of the specularly reflected ray, and diffuse reflection that gives a force acting along the normal to the surface. In this case, unlike for absorption, the assumed satellite geometry and surface properties become relevant in determining the dynamical effects on the trajectory. The last component is related to the portion of incoming energy that is absorbed and later re-emitted as thermal radiation, and even in this case, the force generating from this mechanism is dependent on satellite geometry and surfaces properties.

In describing the development of this Earth Radiation Pressure model (ERP), Knocke introduces the satellite model proposed by Lochry (1966), in which the mentioned mechanisms of surface-radiation interaction are elegantly collected in a single parameter called the *augmentation factor* defined as follows

$$\mathrm{K} = 1 + a_s\mathrm{R} + b_s\xi, \tag{2.6.30}$$

where

- the unity term refers to the force associated to perfect absorption, i.e. the *black body* component,
- a_s and b_s are the fractions of total incident flux reflected and re-emitted respectively, where $a_s + b_s = 1$,
- R and ξ are called *reflection law effectiveness coefficient* and *emission law effectiveness coefficient* respectively,

also, in many cases the portions of K related to reflection and emission are gathered in a single parameter η_s improperly called *satellite reflectivity*, so that

$$\mathrm{K} = 1 + \eta_s, \tag{2.6.31}$$

with

$$\eta_s = a_s\mathrm{R} + b_s\xi. \tag{2.6.32}$$

The *effectiveness coefficients* R and ξ, are so called because they represent the efficiency to which the combined effects of surface properties, satellite geometry and orientation, produce a net force that is oriented along the line of incident radiation, that is, along the direction of r. Under these circumstances, it is useful to identify some extreme cases that help in understanding the range of values that K can assume. Consider for example a perfect specularly reflective planar surface normal to the incident flux, in this case, it is clear that all the incident flux is reflected so $a_s = 1$ and $b_s = 0$. Also, thanks to the orientation and surface properties, this configuration has maximum efficacy in exerting a force that is directed along the line of incidence, meaning that R = 1. Regarding the *emission law effectiveness coefficient*, the two extreme situations are related to two different re-emission processes. First, considering the case is which the satellite is a poor thermal conductor, the incoming energy will be re-emitted almost immediately before any heat transfer can occur, leading to a diffuse thermal emission that is similar in nature to a diffuse reflection. On the other hand, if the satellite can be represented as a perfect conductor, the intercepted radiation will be evenly spread across the body as thermal energy that will be later re-emitted evenly in all directions, and the total dynamical effect is null.

On the basis of these considerations it becomes clear that the ranges of values covered by the effectiveness coefficients are given by

$$0 \leq \mathrm{R} \leq 1 \tag{2.6.33}$$

$$0 \leq \xi \leq \mathrm{R}_d, \tag{2.6.34}$$

where R_d represents the reflection law effectiveness coefficient of a diffusely reflective body whose geometry, orientation and surface properties ensure that the resulting force will act along the line of incident radiation, and finally, the range for the *augmentation factor* is given by

$$1 \leq \mathrm{K} \leq 2, \tag{2.6.35}$$

with K = 1 corresponding to total absorption, and K = 2 represents perfect specular reflection along the line of incidence.

Having defined the *augmentation factor* as a parameter that allows to describe the satellite-radiation interaction accounting for surface properties, satellite geometry and orientation, the acceleration imparted by the flux exiting an Earth surface element $d\mathrm{A}$ with albedo a and emissivity e, can now be computed. First, let us recall that if radiation impinges a surface A with a flux Φ, here intended as power and measured in Watts, the surface experiences a *radiation pressure* that for electromagnetic waves is generally given by

$$\mathrm{P} = \frac{\Phi}{c\mathrm{A}}, \tag{2.6.36}$$

where P is the *radiation pressure* $\left[\mathrm{N/m^2}\right]$ and c is the speed of light in vacuum $\approx 299\,792\,458$ m/s. Now we can exploit (2.6.36) to evaluate the Earth radiation pressure generated by the surface element $d\mathrm{A}$ simply by replacing the flux Φ with $d\Phi$ from equation (2.6.29), and the impinged area becomes the satellite's cross section A_c, thus

$$d\mathrm{P} = \frac{d\Phi}{c\mathrm{A}_c}, \tag{2.6.37}$$

or

$$d\mathrm{P} = \frac{1}{c\,r^2}\left(\frac{\mathrm{aE}_s\cos\theta_s}{\pi} + \frac{\mathrm{eM}_{BB}}{\pi}\right) d\mathrm{A}\cos\alpha, \tag{2.6.38}$$

and note that P becomes dP, representing a small portion of the total radiation pressure since it is associated to a single visible surface element dA.

Once the radiation pressure is available, the acceleration is promptly computed as

$$d\mathbf{a} = \mathrm{K}d\mathrm{P}\frac{\mathrm{A}_c}{m}\hat{\mathbf{r}}, \tag{2.6.39}$$

where m is the satellite mass and $\hat{\mathbf{r}}$ is the unit vector of the Earth element-satellite distance vector $\mathbf{r}$, that follows the direction of impinging radiation. Equation (2.6.39) can be explicitly written substituting dP with its expression from (2.6.38)

$$d\mathbf{a} = \mathrm{K}\left(\mathrm{aE}_s\cos\theta_s + \mathrm{eM}_{BB}\right)\frac{\mathrm{A}_c\cos\alpha}{\pi\,m\,r^2\,c}d\mathrm{A}\,\hat{\mathbf{r}}, \tag{2.6.40}$$

and in this expression the role and convenience of the *augmentation factor* K becomes clear, as it allows to write the acceleration as a single component that always acts along $\hat{\mathbf{r}}$. The use of this parameter, and the consequence it has on the direction of acceleration, might sound like a particularly restrictive modelling technique, however, as Knocke (1988) highlights in his work, most satellites are likely to behave much more as absorbers rather than reflectors, with typical values of a_s being around $0.1 - 0.13$, meaning that only a small fraction of the incident radiation gets reflected. Therefore, given that most of the acceleration components acting out of the direction of $\hat{\mathbf{r}}$ are due to reflection, and that the force associated to the black body term is always along $\hat{\mathbf{r}}$ regardless of satellite geometry or orientation, the Lochry's satellite model is actually well suited for most cases, but must be used wisely if the satellite displays a highly reflective behaviour.

The final step in evaluating the effect of Earth radiation pressure on the satellite dynamics, is the extension of equation (2.6.40) from the effect of a single Earth element dA, to the effect of the whole portion of Earth surface that is instantaneously visible from the satellite, referred to as the *effective cap*. A rigorous approach should define a particular albedo and emissivity model for the entire Earth, then, equation (2.6.40) could be integrated in the variable dA over the whole effective cap, however, fixing a particular model could lead to a loss of generality since any needed variation in the aforementioned model leads to a substantially different result. The complications associated with the computation of the aforementioned integral, can be circumvented in the implementation of the model as a numerical algorithm, by replacing integration with simple summation of the contribution from every single discrete Earth element. Thus, the total acceleration on the satellite due to Earth radiation pressure from the current effective cap is

$$\begin{aligned}\mathbf{a}_{cap} &= \sum_{j=1}^{N} d\mathbf{a}_j \\ &= \frac{\mathrm{K}}{\pi\, m\, c} \sum_{j=1}^{N} \left(\tau \mathrm{a}_j \mathrm{E}_s \cos\theta_{s_j} + \mathrm{e}_j \mathrm{M}_{BB}\right) \frac{\mathrm{A}_{c_j} \cos\alpha_j}{r_j^2} d\mathrm{A}_j\, \hat{\mathbf{r}}_j,\end{aligned} \tag{2.6.41}$$

where every quantity with the subscript j is associated to a particular Earth element $d\mathrm{A}_j$, for a total of N elements, and the parameter τ is equal to zero if the j^{th} element is not lit, and it is unitary otherwise. Notice that the satellite cross section A_{c_j} is also associated to a particular j^{th} element, to account for the general case of a non-spherical satellite. If the satellite is instead modelled as a cannonball, A_c is constant and equal for all elements.

2.6.5 Earth Radiation Model

In the Earth radiation pressure model proposed by Knocke, particular attention is given to the development of a model for Earth radiation that is capable of considering geometrical and temporal variations of albedo and emissivity values without severe discontinuities. The proposed solution, is the use of purely zonal spherical harmonics with time dependent coefficients, so that albedo and emissivity values are computed as

$$\mathrm{a} = \mathrm{a}_0 + \mathrm{a}_1 \mathrm{P}_1(\sin\phi) + \mathrm{a}_2 \mathrm{P}_2(\sin\phi) \tag{2.6.42}$$

$$\mathrm{e} = \mathrm{e}_0 + \mathrm{e}_1 \mathrm{P}_1(\sin\phi) + \mathrm{e}_2 \mathrm{P}_2(\sin\phi), \tag{2.6.43}$$

with

$$\mathrm{a}_1 = \mathrm{c}_0 + \mathrm{c}_1 \cos\left[\omega(JD - t_0)\right] + \mathrm{c}_2 \sin\left[\omega(JD - t_0)\right] \tag{2.6.44}$$

$$\mathrm{e}_1 = \mathrm{k}_0 + \mathrm{k}_1 \cos\left[\omega(JD - t_0)\right] + \mathrm{k}_2 \sin\left[\omega(JD - t_0)\right], \tag{2.6.45}$$

and

- JD is the Julian Date,
- t_0 is a reference epoch,
- ϕ is the latitude of the Earth surface element,
- ω is the frequency associated to temporal variations,
- $\mathrm{P}_n(\sin\phi)$ is the Legendre polynomial of degree n with argument $\sin\phi$.

Here, it is important to underline how ω is a frequency [rad/s] associated to a period of 365.25 days, thus representing seasonal variations in the values of albedo and emissivity that repeats every solar year. Moreover, the other significant aspect is the fact that the harmonics describing the geometrical variations are somehow similar to what happens for the geopotential, but here, only zonal harmonics are considered, meaning that albedo and emissivity values depend only on latitude ϕ. This simplification that might sound restrictive at first, is nevertheless confirmed by satellite radiometric measurements, also, effects due to any longitudinal variation are likely to average out thanks to earth rotation, especially in the long term.

2.6.6 Earth Surface Discretization

In Section 2.6.4 it has been shown how one can replace integration over the entire effective cap with simple summation, in order to compute the total acceleration due to ERP. Nevertheless, it is advisable to do it consciously, in the sense that the performed summation should approximate integration as closely as possible. A first obvious solution is to divide the effective cap in very small planar surface elements $d\mathrm{A}$, where it is clear that for the limit of $d\mathrm{A} \to 0$ and $N \to \infty$ summation becomes integration, but having a very high number of elements can become impractical and cumbersome by the computational standpoint. Knocke, proposes a method for surface discretization that allows to approximate the integral without the need for a large number of elements, in which the entire effective cap is divided into segments of Equal *Projected Attenuated Areas* (EPA) and equation (2.6.41) can be further simplified.

Consider a strip of infinitesimal area $d\mathrm{S}$ around the Earth surface that is defined by its distance $r = \|\mathbf{r}\|$ from the satellite and the angle ζ that $\mathbf{r}$ makes with $\mathbf{r}_s$, the geocentric satellite position, as shown in figure 2.6.1. The infinitesimal area *projected* along the strip-satellite direction is given by

$$\begin{aligned} d\mathrm{S}\cos\alpha &= 2\pi r \sin\zeta \, r \, d\zeta \\ &= 2\pi r^2 \sin\zeta \, d\zeta, \end{aligned} \tag{2.6.46}$$

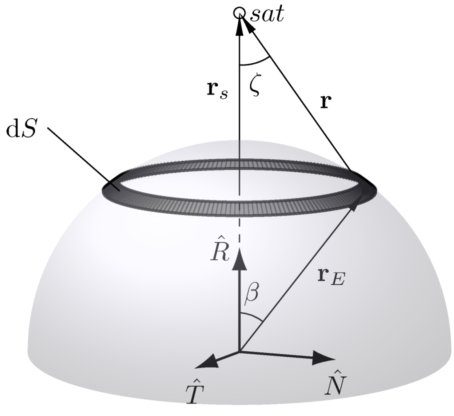

Figure 2.6.1: Representation of the Earth's surface infinitesimal strip.

and now consider the above quantity divided by πr^2, that represents the projected area *attenuated* by the aforementioned value

$$\frac{d\mathrm{S}\cos\alpha}{\pi r^2} = 2\sin\zeta \, d\zeta. \tag{2.6.47}$$

Integrating (2.6.47) from ζ_{i-1} to ζ_i, gives the area of a finite *projected* and *attenuated* ring as

$$\begin{aligned} \int_{\zeta_{i-1}}^{\zeta_i} 2\sin\zeta \, d\zeta &= 2\left[-\cos\zeta\right]\Big|_{\zeta_{i-1}}^{\zeta_i} \\ &= 2\left(\cos\zeta_{i-1} - \cos\zeta_i\right), \end{aligned} \tag{2.6.48}$$

and if this finite ring is divided into N_{r_i} elements, the *projected attenuated area* A^* that will be equal for every surface element in which the entire visible cap is divided into becomes

$$\mathrm{A}^* = \frac{2}{N_{r_i}}\left(\cos\zeta_{i-1} - \cos\zeta_i\right), \tag{2.6.49}$$

moreover, following this line of reasoning, it is clear that the EPA associated to the surface element directly under the satellite, called *sub-satellite cap*, is computed using equation (2.6.49) with $\zeta_{i-1} = \zeta_0 = 0$, $\zeta_i = \zeta_1$ and $N_{r_i} = 1$ since it is a single element, giving the following alternate expression for A^*

$$\mathrm{A}^* = 2\left(1 - \cos\zeta_1\right). \tag{2.6.50}$$

Interestingly, the total EPA of the effective cap is always given by equation (2.6.49), but can be computed following two different approaches. The effective cap can be seen as a large sub-satellite cap that spans from $\zeta_0 = 0$ to ζ_E, where the latter is the angle ζ associated to the upper limit of the visible Earth surface, and is given by

$$\zeta_E = \sin^{-1}\frac{\mathrm{R}_E}{r_s}, \tag{2.6.51}$$

so, calling A^*_{TOT} the EPA of the effective cap we have

$$\mathrm{A}^*_{TOT} = 2\left(1 - \cos\zeta_E\right), \tag{2.6.52}$$

but this projected attenuated area can also be computed as the sum of the total number N of EPA elements, thus, if equation (2.6.50) represents the EPA of a single element:

$$\mathrm{A}^*_{TOT} = 2N\left(1 - \cos\zeta_1\right), \tag{2.6.53}$$

then, equating (2.6.52) and (2.6.53) leads to

$$\cos\zeta_1 = \frac{N - 1 + \cos\zeta_E}{N}. \tag{2.6.54}$$

The above expression for $\cos\zeta_1$ represents a fundamental aspect for the Earth radiation pressure algorithm presented here, based on the equal projected attenuated area method. This equation shows that for a fixed value of ζ_E, that is a function of the satellite's position, and a given number of total surface elements N, which is arbitrary, the sub-satellite cap dimension will follow directly in terms of its viewing angle ζ_1. Then, once ζ_1 is known, the value of EPA is obtained with equation (2.6.50), and the interesting aspect is that this value will be the same for all Earth surface elements, allowing to drastically simplify the calculations that would be involved in integrating equation (2.6.41). As a matter of fact, the acceleration caused by the radiation coming from the entire effective cap can be rewritten as follows

$$\mathbf{a}_{cap} = \mathrm{A}^*\frac{K}{m\,c}\sum_{j=1}^{N}\left(\mathrm{a}_j\mathrm{E}_s\cos\theta_{s_j} + \mathrm{e}_j\mathrm{M}_{BB}\right)\mathrm{A}_{c_j}\hat{\mathbf{r}}_j, \tag{2.6.55}$$

where it can be noticed how the term in (2.6.41)

$$\frac{d\mathrm{A}\cos\alpha}{\pi r^2},$$

that would require integration over the effective cap, is now collected in the equal projected attenuated area A^* that is factored out of the summation.

The remaining step needed to completely define the the effective cap discretization, consists in determining the distribution of EPA elements in the rings between the sub-satellite cap, identified by ζ_1, and the effective cap, delimited by ζ_E. In doing so, the guiding principle is that every element must have the same value of A^*, meaning that there must be some sort of relationship between the number of elements in a ring and its width $\Delta\zeta$, therefore, the two quantities cannot be chosen independently. Clearly, it is advisable to fix the number of elements per ring N_{r_i} and get a distribution of angles ζ_i rather than the opposite, since we want N_{r_i} to be an integer while no particular restriction is present on the values for the rings width. A practice that is suitable for numerical implementation is the definition of a recursion that assigns the number of elements of a certain ring as a fixed increment from the previous ring, that is

$$N_{r_i} = N_{r_{i-1}} + n, \tag{2.6.56}$$

and proceeds from the sub-satellite cap with $N_{r_1} = 1$, up to the final ring delimited by ζ_E. Here, n will be referred to as the *incremental parameter*. This recursion allows to determine every ζ_i using equation (2.6.54) in which $\cos\zeta_E$ is replaced by $\cos\zeta_i$, and the total number of elements N is replaced with the cumulative number of elements K up to ζ_i. So, if the total number of rings, excluding the sub-satellite cap, is n_{ring} the recursion goes as follows

```
for  $i = 2, 3, \ldots, n_{ring}$
    $K = K + n(i-1)$
    $\cos\zeta_i = K\cos\zeta_1 - K + 1$
end,
```

(2.6.57)

and the relation between the total number of elements N, the number of rings n_{rings} and the incremental parameter n is

$$N = 1 + n\frac{n_{ring}}{2}\left(n_{ring} + 1\right), \tag{2.6.58}$$

so, for example, a discretization with 2 rings and an incremental parameter of 6 would give a total of 19 surface elements.

Now that the distribution of rings is defined as well as the number of elements in each of them, in order to compute the ERP acceleration, the knowledge of every satellite-to-element direction is needed and requires the computation of every element's center coordinates. These are described by vectors whose origin is in the Earth's center and have magnitude equal to the Earth's radius R_E, expressed in the local *radial-transverse-normal* (RTN) reference frame through the angles β and λ as follows (see figure 2.6.2)

$$\mathbf{R}_{E_i}^{RTN} = \mathrm{R}_E\cos\beta_i^*\hat{\mathbf{R}} + \mathrm{R}_E\sin\beta_i^*\left(\cos\lambda_i\hat{\mathbf{T}} + \sin\lambda_i\hat{\mathbf{N}}\right), \tag{2.6.59}$$

where

- β_i^* is the RTN zenith angle of the i^{th} element's center given by $\frac{\beta_{i-1}+\beta_i}{2}$,
- λ_i is the RTM azimuth angle of the i^{th} element's center,

then, (2.6.59) is transformed into inertial coordinates to obtain $\mathbf{R}_{E_i}^I$, and the element-to-satellite vector is simply

$$\mathbf{r}_i = \mathbf{r}_s - \mathbf{R}_{E_i}^I \tag{2.6.60}$$

recalling that $\mathbf{r}_s$ is the satellite inertial position. It is appropriate to say a few words about the computation of the local zenith and azimuth angles β and λ. The zenith angle is obtained through the angle γ, that relates ζ and β as follows, see figure 2.6.3,

$$\begin{aligned} \mathrm{R}_E\sin\gamma &= r_s\sin\zeta \\ \gamma &= \sin^{-1}\left(\frac{r_s\sin\zeta}{\mathrm{R}_E}\right), \end{aligned} \tag{2.6.61}$$

then

$$\beta = \gamma - \zeta, \tag{2.6.62}$$

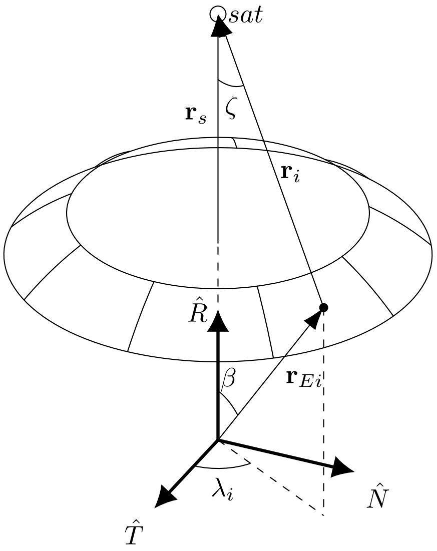

Figure 2.6.2: Angles and vectors arisign from the Earth's surface discretization for the ERP algorithm.

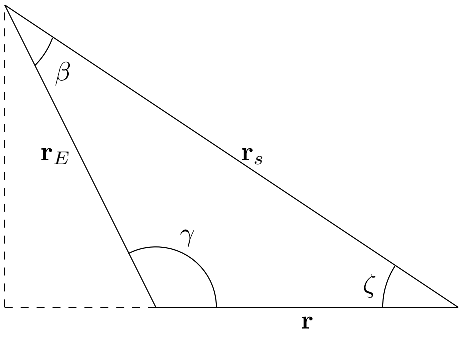

Figure 2.6.3: Schematic representation of the angles γ, β and ζ.

while the azimuth angles along the same ring λ_{r_i} are somehow arbitrary to the extent in which the first λ_{r_1} can either start from 0 or π/N_{r_i}, in other words, the ring can be rotated but must always contain N_{r_i} elements. Here, the distribution of element centers along the same ring in terms of local azimuth in the RTN frame will be defined by

$$\lambda_{r_i} = k\frac{2\pi}{N_{r_i}} - \frac{\pi}{N_{r_i}}, \qquad k = 1, 2, 3, \ldots, N_{r_i}. \tag{2.6.63}$$

2.6.7 The Earth Radiation Pressure Algorithm

In this Section, the main steps involved in the algorithm structure of the Earth radiation pressure model are outlined, then a flowchart is provided at the end.

The required information at the beginning is satellite position and velocity as well as its parameters like mass, cross section and reflectivity, the solar distance, the number of Earth surface rings and incremental parameter, respectively:

$$\mathbf{r}_s,\ \mathbf{v}_s,\ m,\ \mathrm{A}_c,\ \eta_s,\ r_{sun},\ n_{ring},\ n,$$

then, the total number of elements is obtained

$$N = 1 + n\frac{n_{ring}}{2}\left(n_{ring}+1\right).$$

From now on, the process repeats the following steps:

1. *Compute the dimension of the effective cap visible from the satellite:*

$$\zeta_E = \sin^{-1}\left(\frac{\mathrm{R}_E}{r_s}\right)$$
$$\beta_E = \frac{\pi}{2} - \zeta_E$$

2. *Calculate limits of sub-satellite cap and β angle:*

$$\zeta_1 = \cos^{-1}\left(N - 1 + \cos\zeta_E\right)$$
$$\gamma_1 = \sin^{-1}\left(\frac{r_s\sin\zeta_1}{\mathrm{R}_E}\right)$$
$$\beta_1 = \gamma_1 - \zeta_1$$

3. *Calculate remaining ring limits:*

```
for $i$ = $2, 3, \ldots, n_{ring}$
    $K = K + n(i-1)$
    $\zeta_i = \cos^{-1}\left(K\cos\zeta_1 - K + 1\right)$
    $\gamma_i = \sin^{-1}\left(\frac{r_s\sin\zeta_i}{\mathrm{R}_E}\right)$
    $\beta_i = \gamma_i - \zeta_i$
```

4. *Calculate ring centers:*

```
for $i$ = $2, 3, \ldots, n_{ring}+1$
    $\beta_i^* = \frac{\beta_{i-1} + \beta i}{2}$
```

5. *Radiation pressure due to sub-satellite cap:*

$$
\begin{aligned}
&\mathrm{A}^* = 2\,(1-\cos\zeta_1)\\
&\cos\theta_s = \hat{\mathbf{r}}_s \cdot \hat{\mathbf{r}}_{sun} \quad (\text{if } \cos\theta_s < 0 \text{ then } \cos\theta_s = 0)\\
&\phi = \sin^{-1}\left(\hat{\mathbf{Z}} \cdot \hat{\mathbf{r}}_s\right), \quad \hat{\mathbf{Z}} = \text{normal to Earth equator}\\
&\mathrm{a} = \mathrm{a}_0 + \mathrm{a}_1\mathrm{P}_1(\sin\phi) + \mathrm{a}_2\mathrm{P}_2(\sin\phi)\\
&\mathrm{e} = \mathrm{e}_0 + \mathrm{e}_1\mathrm{P}_1(\sin\phi) + \mathrm{e}_2\mathrm{P}_2(\sin\phi)\\
&\mathrm{Flux}_{\mathrm{SW}} = \mathrm{A}^*\mathrm{K}\frac{\mathrm{A}_c\mathrm{E}_s}{m\,c}\\
&\mathrm{Flux}_{\mathrm{LW}} = \mathrm{A}^*\mathrm{K}\frac{\mathrm{A}_c\mathrm{M}_{BB}}{m\,c}\\
&d\mathbf{a}_{ssc} = (\mathrm{a}\,\mathrm{Flux}_{\mathrm{SW}}\cos\theta_s + \mathrm{e}\,\mathrm{Flux}_{\mathrm{LW}})\,\hat{\mathbf{r}}_s
\end{aligned}
$$

6. *Compute and accumulate contribution from each element:*

$$
\begin{aligned}
\text{for } i \;&= 2, 3, \ldots, n_{ring} + 1\\
N_{r_i} &= n(i-1)\\
r &= \sqrt{\mathrm{R}_E^2 + r_s^2 - 2r_s\mathrm{R}_E\cos\beta_i^*}
\end{aligned}
$$

(a) *Compute element-satellite vector:*

$$
\begin{aligned}
&\text{for } j \;=\; 1, 2, \ldots, N_{r_i}\\
&\qquad \mathbf{R}_{E_j}^{RTN} = \mathrm{R}_E\cos\beta_i^*\hat{\mathbf{R}} + \mathrm{R}_E\sin\beta_i^*\left(\cos\lambda_j\hat{\mathbf{T}} + \sin\lambda_j\hat{\mathbf{N}}\right)\\
&\quad \text{Transform } \mathbf{R}_{E_j}^{RTN} \text{ into inertial coordinates obtaining } \mathbf{R}_{E_j}^{I}\\
&\qquad \hat{\mathbf{R}}_{E_j}^{I} = \mathbf{R}_{E_j}^{I}/\mathrm{R}_E\\
&\qquad \mathbf{r}_j = \mathbf{r}_s - \mathbf{R}_{E_j}^{I}\\
&\qquad \hat{\mathbf{r}}_j = \mathbf{r}_j/r
\end{aligned}
$$

(b) *Accumulate accelerations:*

$$
\begin{aligned}
&\phi = \sin^{-1}\left(\hat{\mathbf{Z}} \cdot \hat{\mathbf{R}}_E\right)\\
&\mathrm{a} = \mathrm{a}_0 + \mathrm{a}_1\mathrm{P}_1(\sin\phi) + \mathrm{a}_2\mathrm{P}_2(\sin\phi)\\
&\mathrm{e} = \mathrm{e}_0 + \mathrm{e}_1\mathrm{P}_1(\sin\phi) + \mathrm{e}_2\mathrm{P}_2(\sin\phi)\\
&\cos\theta_s = \hat{\mathbf{R}}_E \cdot \hat{\mathbf{r}}_{sun} \quad (\text{if } \cos\theta_s < 0 \text{ then } \cos\theta_s = 0)\\
&d\mathbf{a} = d\mathbf{a} + (\mathrm{a}\,\mathrm{Flux}_{\mathrm{SW}}\cos\theta_s + \mathrm{e}\,\mathrm{Flux}_{\mathrm{LW}})\,\hat{\mathbf{r}}_j
\end{aligned}
$$

and the process continues until the contribution from every element is calculated.

2.7 Empirical Accelerations

In the context of orbit determination and parameters estimation the objective is to obtain the best accordance between observed and calculated measurements. In an ideal case, these quantities become indistinguishable once the mathematical model used to propagate the satellite's reference orbit and to compute measurements completely and perfectly describes reality, but this is never the case. Even when the major perturbations are accurately described and the estimation process has reached its best solution, residuals in the computed values may still be present if compared to the actual measurements, and these are often caused by unmodeled forces as well as inevitable uncertainties. For this reason a common practice is to introduce a fictitious acceleration model which has no counterpart in the physical world, but can rather be thought as a mathematical tool that *absorbs* these unknown effects and its parameters are estimated during the orbit determination process.

The most common model for empirical accelerations is based on a Fourier expansion with constant and cyclical terms expressed in the Radial-Transverse-Normal (RTN) frame, in which the fundamental frequency is related to the orbital period $\omega_o = 2\pi/T$ and the summation is often carried up to the second term, namely

$$\mathbf{a}_{emp}^{RTN} = \mathbf{A} + \sum_{i=1}^{2} \mathbf{D}_i \cos(i\omega_o(t-t_0)) + \mathbf{S}_i \sin(i\omega_o(t-t_0)), \tag{2.7.1}$$

that in components is

$$\begin{aligned}
a_R &= A_R + \sum_{i=1}^{2} D_{Ri} \cos(i\omega_o(t-t_0)) + S_{Ri} \sin(i\omega_o(t-t_0)) \\
a_T &= A_T + \sum_{i=1}^{2} D_{Ti} \cos(i\omega_o(t-t_0)) + S_{Ti} \sin(i\omega_o(t-t_0)) \\
a_N &= A_N + \sum_{i=1}^{2} D_{Ni} \cos(i\omega_o(t-t_0)) + S_{Ni} \sin(i\omega_o(t-t_0)).
\end{aligned} \tag{2.7.2}$$

Here, the idea of using the orbital period as the fundamental frequency reflects the fact that perturbing accelerations are always “slow” and often cyclical with a frequency content that is explainable first of all with the satellite motion around the Earth as *once-per-rev* components, and even though the summation may be arbitrarily extended to higher frequencies, it is seldom necessary to go above the *twice-per-rev* terms. The constant terms A_R, A_T and A_N are instead associated to acceleration *biases* that often result in secular deviations from a Keplerian orbit, whereas unpredictable changes in the atmospheric drag force, which is clearly a secular term, may be absorbed by the A_T component, or, for the case of Earth Radiation Pressure, by the A_R term. From this consideration, the advantage of expressing this acceleration in the RTN frame is clear, giving a more comprehensible insight into the situation and an easier interpretation of the results since this reference system is intrinsically fixed to the satellite trajectory, whereas expressing it in other coordinate systems would make it harder to draw conclusions due to the relative motion of the satellite.

An alternative expression for the empirical accelerations makes use of the *argument of latitude* rather than the orbital angular velocity ω_o so that no prior knowledge of the orbital period T is necessary, which in any case may vary due to perturbations. Recall that the argument of latitude u is give by the sum of the argument of perigee ω and the true anomaly f, $u = \omega + f$, thus

$$\mathbf{a}_{emp}^{RTN} = \mathbf{A} + \sum_{i=1}^{2} \mathbf{D}_i \cos(i\,u) + \mathbf{S}_i \sin(i\,u), \tag{2.7.3}$$

and note that although being more consistent, equation (2.7.3) requires the calculation of osculating orbital elements at every time of interest.

2.8 Numerical Integration of the Equations of Motion

Once the values of all the accelerations acting on the satellite are available, depending on all the necessary parameters, it is necessary to derive the orbit in terms of position and velocity. To do this, with the special perturbation approach described here, it is necessary to adopt numerical integration, and a multitude of methods have been developed over the decades. The three main categories of numerical integrators can be identified as follows [19]:

- *Runge-Kutta Methods*: These are a family of implicit and explicit iterative methods, which include the Euler method, used in temporal discretization for the approximate solutions of simultaneous nonlinear equations. The most widely known member of the Runge-Kutta family is generally referred to as "RK4", the "classic Runge-Kutta method" or simply as "the Runge-Kutta method".
- *Multistep Methods*: These methods are used for the numerical solution of ordinary differential equations. They use information from previous steps to calculate the next value, attempting to gain efficiency by keeping and using the information from previous steps rather than discarding it. Three families of linear multistep methods are commonly used: Adams–Bashforth methods, Adams–Moulton methods, and the backward differentiation formulas (BDFs).
- *Extrapolation Methods*: Extrapolation is a statistical technique used in data science to estimate values of data points beyond the range of the known values in the data set. There are different types of extrapolation for predicting and evaluating trends in data. The two most widely used extrapolation methods are Linear Extrapolation and Polynomial Extrapolation.

Each of these methods has its own strengths and weaknesses, and the choice of which to use depends on the specific problem at hand. It's important to understand the underlying assumptions and limitations of each method to ensure accurate and reliable results.

Chapter 3

The Linearized Orbit Model

As mentioned previously, the orbit determination problem involves nonlinearities both in the mathematical model, described by the differential equations governing the spacecraft motion, and the observation equations that relate the state to the measurements. Given the circumstances, the linearization procedure becomes necessary for the application of the various estimation techniques discussed in this thesis. With this in mind, this chapter aims to briefly introduce the basic concepts and main terminologies that will be common to the subsequent discussions.

3.1 Linearization Procedure

The general relations describing the orbital dynamics can be written in the following compact form [3]:

$$\dot{\mathbf{X}}(t) = \mathbf{F}\left(\mathbf{X}, \mathbf{P}, t\right), \quad \mathbf{X}(t_0) \equiv \mathbf{X}_0, \tag{3.1.1}$$

with $\mathbf{X}$ being the n-dimensional state vector and $\mathbf{P}$ is a vector of constant dynamical parameters involved in the force model like gravitational parameters, drag and solar radiation coefficients and many others.

The observation equation relating the state to measurements at discrete times is:

$$\mathbf{Y}_i = \mathbf{G}\left(\mathbf{X}, \mathbf{Q}, t_i\right) + \boldsymbol{\varepsilon}_i, \quad E\left[\boldsymbol{\varepsilon}_i \boldsymbol{\varepsilon}_i^T\right] = \mathbf{R}_i, \tag{3.1.2}$$

where $\mathbf{Y}_i$ is the p-dimensional vector of measurements at time t_i, $\boldsymbol{\varepsilon}_i$ is the vector containing measurements errors, which is assumed to have zero mean and covariance matrix $\mathbf{R}_i$, and $\mathbf{Q}$ is a vector of constant geometrical parameters. Notice that having p measurements for l discrete observation times leads to a total of $m = p \times l \gg n$ measurements, making the problem overdetermined and allowing the least squares method to average out the effects of errors.

The linearization of equations (3.1.1) and (3.1.2) requires the knowledge of a reference trajectory $\mathbf{X}^*(\mathbf{X}_0^*, \mathbf{P}^*, t)$ that reasonably resembles the true trajectory $\mathbf{X}$, meaning that deviations

$$\delta\mathbf{X} = \mathbf{X} - \mathbf{X}^* \tag{3.1.3}$$

should be fairly small. Having also reference values for the parameters $\mathbf{P}^*, \mathbf{Q}^*$ and introducing the deviations $\delta\mathbf{Y}, \delta\mathbf{P}, \delta\mathbf{Q}$ similarly to equation (3.1.3), we can expand the dynamical and observation equations to the first order as:

$$\mathbf{F}\left(\mathbf{X}, \mathbf{P}, t\right) = \mathbf{F}\left(\mathbf{X}^*, \mathbf{P}^*, t\right) + \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^* \delta\mathbf{X}(t) + \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^* \delta\mathbf{P}, \tag{3.1.4}$$

$$\mathbf{G}\left(\mathbf{X}_i, \mathbf{Q}, t_i\right) = \mathbf{G}\left(\mathbf{X}_i^*, \mathbf{Q}^*, t_i\right) + \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]_i^* \delta\mathbf{X}(t_i) + \left[\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]_i^* \delta\mathbf{Q}, \tag{3.1.5}$$

and substituting into equations (3.1.1) and (3.1.2) leads to:

$$\delta\dot{\mathbf{X}}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^{*} \delta\mathbf{X}(t) + \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^{*} \delta\mathbf{P}, \tag{3.1.6}$$

$$\delta\mathbf{Y}_i = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]^{*}_{i} \delta\mathbf{X}(t_i) + \left[\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]^{*}_{i} \delta\mathbf{Q}, \tag{3.1.7}$$

where the partial derivatives in square brackets are the Jacobian matrices

$$\mathbf{A}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^{*}, \quad \mathbf{B}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^{*}, \tag{3.1.8}$$

$$\tilde{\mathbf{H}}_i = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]^{*}_{i}, \quad \tilde{\mathbf{K}}_i = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]^{*}_{i}, \tag{3.1.9}$$

and the superscript * indicates that the derivatives are evaluated along the reference trajectory.

Finally, taking advantage of the above definitions the linearized form of equations (3.1.1) and (3.1.2) can be written in the following compact manner:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{p}, \tag{3.1.10}$$

$$\mathbf{y}_i = \tilde{\mathbf{H}}_i\mathbf{x}_i + \tilde{\mathbf{K}}_i\mathbf{q} + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, l, \tag{3.1.11}$$

in which the definitions $\mathbf{x} = \delta\mathbf{X}$, $\mathbf{y} = \delta\mathbf{Y}$, $\mathbf{p} = \delta\mathbf{P}$, $\mathbf{q} = \delta\mathbf{Q}$ are used.

3.2 The State Transition Matrix

In many filtering applications, such as the batch filter that will be discussed later, the estimation of the state occurs at particular instants of time or often at a single epoch, thus we need a mean to relate all measurements to that particular epoch. A linearized homogeneous system is described by the following system of differential equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t), \tag{3.2.1}$$

and the solution can be expressed as

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}_0, \tag{3.2.2}$$

where $\mathbf{\Phi}(t, t_0)$ is the *state transition matrix*, that for linearized systems can be defined as

$$\mathbf{\Phi}(t, t_0) = \frac{\partial \mathbf{X}}{\partial \mathbf{X}_0}. \tag{3.2.3}$$

The state transition matrix, as we can clearly see from equation (3.2.2), allows us to relate states at generic times t, to the state at epoch t_0, or in the more general case at any time t_k, and it satisfies the following relevant properties:

$$\mathbf{\Phi}(t_k, t_k) = I, \tag{3.2.4}$$

$$\mathbf{\Phi}(t_k, t_j) = \mathbf{\Phi}(t_j, t_k)^{-1}, \tag{3.2.5}$$

$$\mathbf{\Phi}(t_k, t_j) = \mathbf{\Phi}(t_k, t_i)\mathbf{\Phi}(t_i, t_j). \tag{3.2.6}$$

There are several ways to compute the state transition matrix for a linearized systems. In the case of orbit determination equation (3.2.1) represents a system of linear differential equations with time-dependent coefficients, and the most practical way to compute $\mathbf{\Phi}$ is to differentiate equation (3.2.1) noting that $\mathbf{x}_0$ is a constant. This yields

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{\Phi}}(t, t_0)\mathbf{x}_0. \tag{3.2.7}$$

We now substitute (3.2.7) and (3.2.2) into (3.2.1) to obtain

$$\dot{\mathbf{\Phi}}(t, t_0)\mathbf{x}_0 = \mathbf{A}(t)\mathbf{\Phi}(t, t_0)\mathbf{x}_0, \tag{3.2.8}$$

and since it must be true for all $\mathbf{x}_k$, it follows that

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}(t, t_0). \tag{3.2.9}$$

Equation (3.2.9) can then be integrated with initial condition $\mathbf{\Phi}(t_0, t_0) = I$, with I being the identity matrix.

3.3 Observations and Measurement Deviations

In orbit determination the role of observations is crucial, as a means by which the state of an orbiting object is periodically monitored and updated. In the linearization of the orbit model, the concept of measurement deviations was introduced as the difference between the actual measurements and computed measurements, i.e. measurements calculated from the mathematical model describing the reference trajectory, thus a brief insight of these concepts is needed.

There exist plenty of observation techniques, from radar ranging, radar doppler and altimeter data to satellite-to-satellite tracking with GPS and many others, but here we will discuss the two most common and useful by an intuitive standpoint: the *instantaneous range* and *instantaneous range rate*. First of all, the term *instantaneous* underlines the fact that the measurement is purely geometrical and does not contain any temporal bias, as would be the case for real observations, meaning that physical effects such as the speed of light, atmospheric influence but also the specific position of the receiving or emitting point are ignored. Instead, the observer and the observed objects are considered as geometrical points that instantaneously communicate. A more realistic model of measurement should at least account for the so-called 'time-of-light' principle [24] and the fact that the synchronization between transmitter and receiver clocks is never perfect. However, these aspects go beyond the scope of this thesis and a more simplified model is favored for its ease of implementation and understanding.

3.3.1 Instantaneous Range

Under these hypothesis the ideal range is simply the magnitude of the vector $\boldsymbol{\rho}$ that connects the ground station to the satellite. Considering an inertial reference frame centered in the planet, if the cartesian coordinates of the ground station are represented by the vector $\boldsymbol{\rho}_{gs}$ and the coordinates of the spacecraft are contained in the vector $\boldsymbol{\rho}_{sc}$, the instantaneous ideal range at time t_i can be expressed as

$$\begin{aligned} \rho_{c_i} &= \left|\boldsymbol{\rho}_{sc} - \boldsymbol{\rho}_{gs}\right| \\ &= \sqrt{\left(X_{sc} - X_{gs}\right)^2 + \left(Y_{sc} - Y_{gs}\right)^2 + \left(Z_{sc} - Z_{gs}\right)^2}, \end{aligned} \tag{3.3.1}$$

which is clearly a scalar quantity, and the subscript c indicates that this ideal range is the measurement *computed* from the spacecraft coordinates obtained with the reference trajectory. Now, if we call ρ_{o_i} the actual range measurement given by the ground station, the corresponding measurement deviation y_i at time t_i can be computed as

$$\mathrm{y}_i = \rho_{o_i} - \rho_{c_i}. \tag{3.3.2}$$

3.3.2 Instantaneous Range Rate

In some cases a measure of the time rate of change of the ideal range can be captured giving the *range rate*. The mathematical expression can be derived by differentiation with respect to time of the ideal range

$$\dot{\rho} = \frac{(X_{sc} - X_{gs})\left(\dot{X}_{sc} - \dot{X}_{gs}\right) + (Y_{sc} - Y_{gs})\left(\dot{Y}_{sc} - \dot{Y}_{gs}\right) + (Z_{sc} - Z_{gs})\left(\dot{Z}_{sc} - \dot{Z}_{gs}\right)}{\rho}, \tag{3.3.3}$$

that is also

$$\dot{\rho} = \frac{\boldsymbol{\rho} \cdot \dot{\boldsymbol{\rho}}}{\rho}, \tag{3.3.4}$$

and the associated measurement deviation is defined similarly to equation (3.3.2)

$$\mathrm{y}_i = \dot{\rho}_{o_i} - \dot{\rho}_{c_i}. \tag{3.3.5}$$

Notice that from equation (3.3.4) the range rate can be interpreted as the component of the relative velocity vector between the spacecraft and the ground station, along the line-of-sight direction $\boldsymbol{\rho}$.

Finally, from a practical point of view, it must be considered that in both equations (3.3.1) and (3.3.3), the ground station coordinates must account for the planet's rotation, that is

$$\begin{Bmatrix} X_{gs} \\ Y_{gs} \\ Z_{gs} \end{Bmatrix} = \begin{Bmatrix} R_p \cos\delta \cos(\lambda + \omega_p t) \\ R_p \cos\delta \sin(\lambda + \omega_p t) \\ R_p \sin\delta \end{Bmatrix}, \tag{3.3.6}$$

where R_p is the planet's mean radius, ω_p is the planet's rotation angular velocity and δ, λ are the ground station's latitude and longitude respectively.

Chapter 4

Batch Estimation

The Batch Filter or Batch Processor in orbit determination is a direct application of the least squares method, in particular, since the orbit determination problem involves nonlinear differential equations, nonlinear least squares must be used.

The name *Batch* underlines how the available measurements are all processed at once, precisely as a single batch of data to obtain the best estimate at a single reference time called *epoch*, making it an offline estimator as opposed to a real-time estimator, since the estimate is obtained only after a certain amount of observations are collected.

4.1 Relating Observations to an Epoch State

As mentioned before, the batch filter requires all measurements to be related to the particular epoch in which the state estimation occurs, in other words, we need to relate the measurements deviations $\mathbf{y}_i$ (3.1.11) at time t_i, to the state deviation $\mathbf{x}_0$ at epoch t_0, but also to the parameter deviations $\mathbf{p}$. To do so, taking a cue from Section 3.2, we can map the linear state deviations through the following relationship

$$\delta\mathbf{X} = \frac{\partial \mathbf{X}}{\partial \mathbf{X}_0}\delta\mathbf{X}_0 + \frac{\partial \mathbf{X}}{\partial \mathbf{P}}\delta\mathbf{P}. \tag{4.1.1}$$

Here, the aforementioned state transition matrix $\partial\mathbf{X}/\partial\mathbf{X}_0$ is recognized, and we will call it *dynamical state transition matrix* to distinguish it from the *parameter state transition matrix* defined by $\partial\mathbf{X}/\partial\mathbf{P}$, and they both can be combined in the *system state transition matrix* $\mathbf{\Phi}(t, t_0)$ through Kroneker sum:

$$\mathbf{\Phi}(t, t_0) = \mathbf{\Phi}_{xx}(t, t_0) \oplus \mathbf{\Phi}_{xp}(t, t_0). \tag{4.1.2}$$

This result allows to write the solution to equation (3.1.10) as

$$\mathbf{x}(t) = \mathbf{\Phi}_{xx}(t, t_0)\mathbf{x}_0 + \mathbf{\Phi}_{xp}(t, t_0)\mathbf{p}, \tag{4.1.3}$$

that can be inserted into the observation equation (3.1.11) to obtain

$$\mathbf{y}_i = \tilde{\mathbf{H}}_i\mathbf{\Phi}_{xx}(t.t_0)\mathbf{x}_0 + \tilde{\mathbf{H}}_i\mathbf{\Phi}_{xp}(t, t_0)\mathbf{p} + \tilde{\mathbf{K}}_i\mathbf{q} + \varepsilon_i. \tag{4.1.4}$$

Equation (4.1.4) maps state deviation $\mathbf{x}_0$ at epoch, along with parameters deviation $\mathbf{p}$ and $\mathbf{q}$, to every measurement deviation at any time t_i, and it can be written in a more compact form by using the following definitions:

$$\mathbf{H}_i^x = \tilde{\mathbf{H}}_i \mathbf{\Phi}_{xx}(t, t_0), \tag{4.1.5}$$

$$\mathbf{H}_i^p = \tilde{\mathbf{H}}_i \mathbf{\Phi}_{xp}(t, t_0), \tag{4.1.6}$$

and

$$\tilde{\mathbf{x}}_0 = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{p} \\ \mathbf{q} \end{pmatrix}, \tag{4.1.7}$$

then

$$\mathbf{y}_i = \begin{pmatrix} \mathbf{H}_i^x & \mathbf{H}_i^p & \tilde{\mathbf{K}}_i \end{pmatrix} \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{p} \\ \mathbf{q} \end{pmatrix} + \boldsymbol{\varepsilon}_i, \tag{4.1.8}$$

that finally leads to

$$\mathbf{y}_i = \mathbf{H}_i \tilde{\mathbf{x}}_0 + \boldsymbol{\varepsilon}_i, \tag{4.1.9}$$

where we have defined

$$\mathbf{H}_i = \begin{pmatrix} \mathbf{H}_i^x & \mathbf{H}_i^p & \tilde{\mathbf{K}}_i \end{pmatrix}. \tag{4.1.10}$$

4.1.1 Computing the State Transition Matrix for the Batch Filter

The computation of the state transition matrix has already been discussed previously in a general sense in Section 3.2, where equation (3.2.9) was derived considering that the state transition matrix is a mean of relating the dynamical state (position and velocity) at epoch to the state at different times. However, in Chapter 3, we introduced the fact that the state vector may also contain dynamical and geometrical parameters other than position and velocity only, and Section 4.1 showed the distinction between the *dynamical state transition matrix* and the *parameter state transition matrix*, and this Section describes how these two are computed.

For what concerns the dynamical state transition matrix, related to position and velocity only, the differential equations can be shown to be the same as in equation (3.2.9), that is

$$\dot{\mathbf{\Phi}}_{xx}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xx}(t, t_0), \quad \mathbf{\Phi}_{xx}(t_0, t_0) = I, \tag{4.1.11}$$

where $\mathbf{\Phi}_{xx}(t, t_0) = \partial \mathbf{X}(t)/\partial \mathbf{X}_0$.

For the parameter state transition matrix we take the gradient of equation (3.1.1) with respect to the dynamical parameters vector $\mathbf{P}$:

$$\frac{\partial \dot{\mathbf{X}}(t)}{\partial \mathbf{P}} = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^* \frac{\partial \mathbf{X}(t)}{\partial \mathbf{P}} + \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^*, \tag{4.1.12}$$

changing the order of operations and recalling that $\mathbf{A}(t) = [\partial \mathbf{F}/\partial \mathbf{X}]^*$

$$\frac{d}{dt}\left(\frac{\partial \mathbf{X}}{\partial \mathbf{P}}\right) = \mathbf{A}(t)\frac{\partial \mathbf{X}}{\partial \mathbf{P}} + \mathbf{B}(t), \tag{4.1.13}$$

where we have defined

$$\mathbf{B}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^*, \tag{4.1.14}$$

finally, the differential equations for the parameter state transition matrix are

$$\dot{\mathbf{\Phi}}_{xp}(t,t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t,t_0) + \mathbf{B}(t), \quad \mathbf{\Phi}_{xp}(t_0,t_0) = \mathbf{0}. \tag{4.1.15}$$

Equations (4.1.11) and (4.1.15) could be integrated considering the fact that they exhibit the property of being column-decoupled. However, since they must be integrated together with the dynamical equations the decoupled integration may turn into a tiresome effort.

4.2 The Weighted Least Squares Solution - Differential Correction

Trough linearization and the state transition matrix we have arrived at equation (4.1.9), that lays the foundations for the *Batch* processor by mapping state vector deviations ad epoch into computed measurement deviations at time t_i.

At this point we must consider the fact that with the batch algorithm a total number of $m = p \times l$ measurements is processed at once, where p is the number of observations at a single time, that is the dimension of the measurement deviations vector $\mathbf{y}_i$, and l is the total number of observation instants, meaning that we can formulate the problem as follows:

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_l \end{Bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_l \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_l \end{Bmatrix}, \tag{4.2.1}$$

meaning that $\mathbf{y}$ and $\boldsymbol{\varepsilon}$ are $m \times 1$ vectors of measurement deviations and measurement errors respectively, and $\mathbf{H}$ is a $m \times n$ matrix. If we use $\mathbf{x}$ instead of $\tilde{\mathbf{x}}_0$ for shortness we obtain

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon}. \tag{4.2.2}$$

Moreover, as the name suggests, the observations are not treated equally but a certain weighting factor is considered, because $\mathbf{y}$ is generally composed of different measurement types from different sources, and we may wish to give more relevance to the more accurate measurements over the ones subjected to higher errors.

Under the hypothesis that measurements are unrelated in time, which may sound restrictive but is often the case, and that measurement errors obey a *multivariate normal distribution* with zero mean $E[\varepsilon_i] = 0$, a suitable choice for the weighting matrix $\mathbf{W}_i$ at t_i is the inverse of the measurement error covariance matrix $\mathbf{R}_i = E\left[\varepsilon_i \varepsilon_i^T\right]$, so that every measurement is pondered with the inverse of its respective variance,

$$\mathbf{W}_i = \mathbf{R}_i^{-1}, \tag{4.2.3}$$

then, by combining every $\mathbf{W}_i$ matrix into a single block diagonal matrix, we get the complete weighting matrix for the batch processor:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & 0 & \dots & 0 \\ 0 & \mathbf{W}_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{W}_l \end{bmatrix}. \tag{4.2.4}$$

The fact that the weighting matrix is presented in a block diagonal shape is the consequence of the assumption that measurements are unrelated in time; however, correlation might still be present between observation occurring at the same time, meaning that submatrices $\mathbf{W}_i$ may not be diagonal.

Going back to equation (4.2.2), the state vector $\mathbf{x}$ of dimension n containing position, velocity and parameters deviations is the unknown of this matrix equation. In practice the total number m of measurements is way larger than the dimension of the state vector since, in many applications, orbiting objects are periodically tracked by multiple measurement sources over long periods of time. This ensures that the system is overdetermined, meaning that there are more scalar equations than there are unknowns, and the weighted least squares solution gives the value $\hat{\mathbf{x}}$ of $\mathbf{x}$ as the one that minimizes the sum of the squares of the calculated observation residuals $\mathbf{y}$ weighted by the matrix $\mathbf{W}$. Considering that the observation residuals are the difference between *observed* measurement deviations and *calculated* measurement deviations, often referred to as O - C ("o minus c"), the weighted least squares solution is the value of $\mathbf{x}$ that minimizes the following scalar performance index:

$$J(\mathbf{x}) = \frac{1}{2}\boldsymbol{\varepsilon}^T\mathbf{W}\boldsymbol{\varepsilon} = \sum_{i=1}^{l}\frac{1}{2}\boldsymbol{\varepsilon}_i^T\mathbf{W}_i\boldsymbol{\varepsilon}_i. \tag{4.2.5}$$

Here ε represents observation residuals, and from equation (4.2.2) $J(\mathbf{x})$ can also be written as

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{y} - \mathbf{H}\mathbf{x}\right)^T\mathbf{W}\left(\mathbf{y} - \mathbf{H}\mathbf{x}\right), \tag{4.2.6}$$

and by performin calculations we can expand it into

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{y}^T\mathbf{W}\mathbf{y} - \mathbf{y}^T\mathbf{W}\mathbf{H}\mathbf{x} - \mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{y} + \mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{x}\right), \tag{4.2.7}$$

moreover, considering that all terms in the RHS of equation (4.2.7) must be scalar, we can perform a transposition of $\mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{y}$ knowing that the resulting equation will be equivalent. This results in

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{y}^T\mathbf{W}\mathbf{y} - 2\mathbf{y}^T\mathbf{W}\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{x}\right). \tag{4.2.8}$$

Now, a first necessary condition for the solution $\hat{\mathbf{x}}$ to minimize $J(\mathbf{x})$, is that the derivative of the performance index with respect to $\mathbf{x}$ must be zero,

$$\frac{\partial J}{\partial \mathbf{x}} = \frac{1}{2}\frac{\partial}{\partial \mathbf{x}}\left(\mathbf{y}^T\mathbf{W}\mathbf{y} - 2\mathbf{y}^T\mathbf{W}\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{x}\right) = 0. \tag{4.2.9}$$

In order to perform the derivation we recall the following useful matrix properties. Being $\mathbf{z}$ a column vector of dimension n, $\mathbf{A}$ a matrix with n rows and arbitrary number of columns, and $\mathbf{B}$ a square $n \times n$ matrix:

$$\frac{\partial}{\partial \mathbf{z}}\left(\mathbf{A}^T\mathbf{z}\right) = \mathbf{A}^T, \quad \frac{\partial}{\partial \mathbf{z}}\left(\mathbf{z}^T\mathbf{B}\mathbf{z}\right) = 2\mathbf{z}^T\mathbf{B}, \tag{4.2.10}$$

and applying these properties in the computation of the derivative in equation (4.2.9) we get

$$\frac{\partial J}{\partial \mathbf{x}} = -\mathbf{y}^T\mathbf{W}\mathbf{H} + \mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{H} = 0, \tag{4.2.11}$$

or

$$\mathbf{x}^T\mathbf{H}^T\mathbf{W}\mathbf{H} = \mathbf{y}^T\mathbf{W}\mathbf{H}, \tag{4.2.12}$$

finally, by transposing both sides of equation (4.2.12) considering that $\mathbf{W}^T = \mathbf{W}$, we arrive at the so called *normal equations* in the least squares formulation:

$$\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)\mathbf{x} = \mathbf{H}^T\mathbf{W}\mathbf{y}. \tag{4.2.13}$$

If the *normal matrix* is positive definite it is also invertible and the weighted leas squares solution is

$$\hat{\mathbf{x}} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}, \tag{4.2.14}$$

representing the best estimate of the state vector deviations as the one that minimizes the sum of the squared measurement residuals (O - C).

Remembering that the measurement equation, eq. (4.2.2) for the batch processor, is the result of a stochastic process since it contains measurement errors $\boldsymbol{\varepsilon}$, and remembering that we assumed $E\left[\boldsymbol{\varepsilon}\right] = 0$ and $E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right] = \mathbf{W}^{-1}$, it follows that the weighted least squares solution $\hat{\mathbf{x}}$ is also a random variable with:

$$E\left[\hat{\mathbf{x}}\right] = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}E\left[\boldsymbol{\varepsilon}\right] = 0, \tag{4.2.15}$$

and

$$\begin{aligned} E\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^T\right] &= E\left[\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{y}\mathbf{y}^T\mathbf{W}^T\mathbf{H}\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\right] \\ &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right]\mathbf{W}\mathbf{H}\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1} \\ &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{W}^{-1}\mathbf{W}\mathbf{H}\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1} \\ &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}. \end{aligned} \tag{4.2.16}$$

Equation (4.2.15) shows that in the application of a batch filter the expected value of the least squares solution is zero, which is reasonable since $\hat{\mathbf{x}}$ represents state vector *deviations* from a reference solution, that clearly should be as small as possible to indicate a good estimation. Equation (4.2.16) instead shows that the matrix

$$\mathbf{P} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}, \tag{4.2.17}$$

corresponds to the variance-covariance matrix of the least squares solution, meaning that it is related to the accuracy of the estimate $\hat{\mathbf{x}}$ and is called *solution covariance matrix.* In general, high magnitude for the elements of the matrix $\mathbf{P}$ indicates a poorly accurate estimate, however some care must be taken in using these information as a valuation of the orbit determination accuracy. The expected value and the covariance from equations (4.2.15)-(4.2.16), actually represent the distribution of $\hat{\mathbf{x}}$ values that would be obtained by a repeated orbit determination experiment for the same orbit but with randomly generated measurements, however it's still true that the interval described by the covariance of $\hat{\mathbf{x}}$ most likely contains the real solution.

Another important aspect emerging from equations (4.2.13)-(4.2.14), is that in order for a solution to exist, the matrix

$$\boldsymbol{\Lambda} = \mathbf{H}^T\mathbf{W}\mathbf{H}, \tag{4.2.18}$$

also called *information matrix*, must be invertible, and if this is the case, it will also be positive definite. Moreover, its rank is related to *parameters observability.* If the *information matrix* is not full rank, it will not be invertible, meaning that the matrix $\boldsymbol{\Lambda}$ contains linearly dependent rows or columns, which translates into the impossibility to obtain a unique parameters estimation from the observation set $\mathbf{y}$, no matter how large it is or how accurate the measurements are. This kind of problem may arise from an inappropriately configured tracking systems or from an inconvenient formulation of the state vector; the number of parameters to be estimated should always be as small as possible, and parameters that do not appear in separate force functions should not be estimated separately, meaning that they should not appear separately in the state vector, rather their product or quotient may be estimated as a single quantity.

Going back to the least squares solution in equation (4.2.14), it must be remembered that $\hat{\mathbf{x}}$ is not a final estimate of the parameters contained in the state vector, but it is an estimate of the state vector *deviations* from the reference a priori values, that is a direct consequence of the linearization procedure. These state *deviations* can then be seen as *corrections* that should be applied to the a priori state vector

$$\mathbf{X}^*_{0new} = \mathbf{X}^*_{0old} + \hat{\mathbf{x}}, \tag{4.2.19}$$

so that a new a priori value is obtained, and a new reference trajectory can be propagated. Clearly, the approximation occurred in the linearization procedure has originated the need for this iterative process, where the initial a priori values are corrected untill convergence, and that's the reason why the *batch filter* is also referred to as the *differential correction method.*

4.2.1 Estimation with a Priori Information

In many cases a priori values for the state vector deviation $\bar{\mathbf{x}}$ and the associated a priori covariance matrix $\bar{\mathbf{P}}$ are available, and they may come from a priori initial conditions or the processing of a previous batch of data. In practice they can be treated as additional information in the guise of additional observations, so that $\bar{\mathbf{x}}$ can be interpreted as an observation of $\mathbf{x}$ at epoch,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon} \tag{4.2.20}$$

$$\bar{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta}. \tag{4.2.21}$$

Here, the a priori value for the state deviation at epoch contains an associated random error $\boldsymbol{\eta}$ with the following characteristics

$$E\left[\boldsymbol{\eta}\right] = 0, \quad E\left[\boldsymbol{\eta}\boldsymbol{\eta}^T\right] = \bar{\mathbf{P}}, \quad E\left[\boldsymbol{\eta}\boldsymbol{\varepsilon}^T\right] = 0, \tag{4.2.22}$$

meaning that it has zero mean, covariance matrix corresponding to $\bar{\mathbf{P}}$ and it is not correlated to measurement errors $\boldsymbol{\varepsilon}$.

Consequently, equations (4.2.20) and (4.2.21) can be combined in a single observation equation using the following definitions:

$$\mathbf{y} = \begin{Bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_l \\ \bar{\mathbf{x}} \end{Bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_l \\ \boldsymbol{\eta} \end{Bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_l \\ \mathbf{I} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \mathbf{R}_l & \vdots \\ 0 & \dots & \dots & \bar{\mathbf{P}} \end{bmatrix}, \tag{4.2.23}$$

then, using these in equation (4.2.14) and recalling that $\mathbf{W} = \mathbf{R}^{-1}$ we get

$$\hat{\mathbf{x}} = \left\{ [\mathbf{H}_i^T \vdots \mathbf{I}] \begin{bmatrix} \mathbf{W}_i & 0 \\ \dots & \dots \\ 0 & \bar{\mathbf{P}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_i^T \\ \dots \\ \mathbf{I} \end{bmatrix} \right\}^{-1} [\mathbf{H}_i^T \vdots \mathbf{I}] \begin{bmatrix} \mathbf{W}_i & 0 \\ \dots & \dots \\ 0 & \bar{\mathbf{P}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ \dots \\ \bar{\mathbf{x}} \end{bmatrix}, \tag{4.2.24}$$

that leads to

$$\hat{\mathbf{x}} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}^{-1}\right)^{-1}\left(\mathbf{H}^T\mathbf{W}\mathbf{y} + \bar{\mathbf{P}}^{-1}\bar{\mathbf{x}}\right), \tag{4.2.25}$$

where the *solution covariance* and *information* matrices are

$$\mathbf{P} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}^{-1}\right)^{-1}, \tag{4.2.26}$$

$$\boldsymbol{\Lambda} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}^{-1}\right). \tag{4.2.27}$$

Unlike the unweighted least squares case, here the solution form equation (4.2.25) actually minimizes the following performance index, that other than the measurements residuals, contains also the residuals related to the a priori estimate $\bar{\mathbf{x}}$,

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{y} - \mathbf{H}\mathbf{x}\right)^T \mathbf{W}\left(\mathbf{y} - \mathbf{H}\mathbf{x}\right) + \frac{1}{2}\left(\bar{\mathbf{x}} - \mathbf{x}\right)\bar{\mathbf{P}}^{-1}\left(\bar{\mathbf{x}} - \mathbf{x}\right). \tag{4.2.28}$$

Equation (4.2.25) is the least squares estimate with a priori information for the batch filter, where the measurement vector $\mathbf{y}$ contains the entire batch of data and is merged with the a priori estimate of the state deviations $\bar{\mathbf{x}}$. In this sense it represents a more general case compared to the least squares solution in equation (4.2.14), however, in both situations, the $n \times n$ normal matrix must be inverted and if n is large the inversion can lead to computational problems. This is why the use of more accurate algorithms is often preferred, such as Cholesky decomposition, rather than direct matrix inversion.

4.3 The Batch Processor Algorithm

This Section summarizes the algorithm developed to implement the batch filter for orbit determination. Let's start by recalling the problem statement: suppose that we wish to estimate the position and velocity of an orbiting spacecraft at a certain reference time t_0, called epoch, along with other dynamical and non-dynamical parameters, given a reference initial condition $\mathbf{X}^*(t_0) = \mathbf{X}_0^*$, an a priori estimate of the state deviations $\bar{\mathbf{x}}_0$, an associated a priori covariance matrix $\bar{\mathbf{P}}_0$, and a set of observations $\mathbf{Y}$ such as range, range-rate or others. The solution to this problem is found by solving the normal equations

$$\left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}_0^{-1}\right)\hat{\mathbf{x}}_0 = \mathbf{H}^T\mathbf{W}\mathbf{y} + \bar{\mathbf{P}}_0^{-1}\bar{\mathbf{x}}_0, \tag{4.3.1}$$

where t_0 is an arbitrary epoch and $\mathbf{y}$ is a vector of measurement deviations, that is the difference between observation data contained in $\mathbf{Y}$ and computed observations, previously called (O - C).

First, the reference initial condition $\mathbf{X}_0^*$ allows the propagation of the reference trajectory $\mathbf{X}^*$, that is integrated up to every observation time t_i along with the differential equations for the *system state transition matrix*

$$\dot{\mathbf{\Phi}}_{xx}(t_i, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xx}(t_i, t_0), \qquad \mathbf{\Phi}_{xx}(t_0, t_0) = I, \tag{4.3.2}$$

$$\dot{\mathbf{\Phi}}_{xp}(t_i, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t_i, t_0) + \mathbf{B}(t), \qquad \mathbf{\Phi}_{xp}(t_0, t_0) = 0, \tag{4.3.3}$$

which gives a total of $n_d + n_d \times n_d + n_d \times p$ differential equations, where n_d is the dimension of the dynamical system (the subscript d is used to distinguish from n which is often used for the dimension of the state vector) and p is the dimension of the dynamical parameters vector. For example, if we wish to estimate a satellite position and velocity along with the gravitational constant G, a total of 48 differential equations must be integrated simultaneously.

Having the reference trajectory enables us to evaluate the partial derivatives contained in the matrices

$$\mathbf{A}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^*, \qquad \mathbf{B}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^*, \tag{4.3.4}$$

$$\tilde{\mathbf{H}}_i = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]^*, \qquad \tilde{\mathbf{K}}_i = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]^*, \tag{4.3.5}$$

that together with the state transition matrix in equation (4.1.4), relate measurement deviations at any time t_i to state vector deviations at epoch. Moreover, as the reference trajectory is propagated to subsequent observation times t_i, the measurement deviations vector $\mathbf{y}$ can be computed as described in Section 3.3, as the difference between observed and computed measurements

$$\mathbf{y}_i = \mathbf{Y}_{oi} - \mathbf{Y}_{ci}, \tag{4.3.6}$$

where Y_{oi} is the available measurement and Y_{ci} is the computed measurement, both at time t_i.

Now, going back to equation (4.3.1), it can be noticed that the information matrix

$$\mathbf{\Lambda} = \mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}_0^{-1}, \tag{4.3.7}$$

is of $n \times n$ size, where n is the size of the state vector, while the RHS

$$\mathbf{H}^T\mathbf{W}\mathbf{y} + \bar{\mathbf{P}}_0^{-1}\bar{\mathbf{x}}_0, \tag{4.3.8}$$

is $n \times 1$. If the weighting matrix $\mathbf{W}$ is a block diagonal matrix, the most convenient way to build up these equations is to simply accumulate the matrices at every observation time as:

$$\mathbf{H}^T\mathbf{W}\mathbf{H} = \sum_{i=1}^{l} \mathbf{H}_i^T\mathbf{W}_i\mathbf{H}_i \tag{4.3.9}$$

$$\mathbf{H}^T\mathbf{W}\mathbf{y} = \sum_{i=1}^{l} \mathbf{H}_i^T\mathbf{W}_i\mathbf{y}_i, \tag{4.3.10}$$

where the definitions for $\mathbf{H}_i$, $\mathbf{W}_i$ and $\mathbf{y}_i$ are coherent with the ones described in Section 4.1 and 4.2. Notice that each time of observation can have a different number of concurrent measurements, which translates into the dimension p, of the measurement vector $\mathbf{y}_i$, being variable. However, equations (4.3.9) and (4.3.10) can also be used ignoring the fact that some measurements may happen at the same time, and if this is the case, the matrix $\mathbf{H}_i$ will always be $1 \times n$, the weighting matrix $\mathbf{W}_i$ will actually be a 1×1 scalar quantity containing the variance of the corresponding observation, and $\mathbf{y_i}$ will also be a scalar measurement deviation.

At this point the least squares solution $\hat{\mathbf{x}}_0$ can be computed

$$\hat{\mathbf{x}}_0 = \left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}_0^{-1}\right)^{-1}\left(\mathbf{H}^T\mathbf{W}\mathbf{y} + \bar{\mathbf{P}}_0^{-1}\bar{\mathbf{x}}_0\right), \tag{4.3.11}$$

and the reference initial conditions are *corrected*

$$\mathbf{X}^*_{0new} = \mathbf{X}^*_{0old} + \hat{\mathbf{x}}_0, \tag{4.3.12}$$

yet, before iterating, some considerations on the new value for the a priori estimate $\bar{\mathbf{x}}_{new}$ must be discussed. The role of $\bar{\mathbf{x}}_0$ and $\bar{\mathbf{P}}_0$, as discussed in Section 4.2.1, is that of additional information that is merged with observation data, consequently the value of $\mathbf{X}_0^* + \bar{\mathbf{x}}_0$ should be held constant at the beginning of each iteration. Considering that the reference initial conditions $\mathbf{X}_0^*$ are corrected by the quantity $\hat{\mathbf{x}}_0$ as in equation (4.3.12), the need to hold the quantity $\mathbf{X}_0^* + \bar{\mathbf{x}}_0$ constant, translates into the following condition for the value of $\bar{\mathbf{x}}_0^{new}$:

$$\begin{aligned} \mathbf{X}_0^{*new} + \bar{\mathbf{x}}_0^{new} &= \mathbf{X}_0^{*old} + \bar{\mathbf{x}}_0^{old} \\ &= \mathbf{X}_0^{*old} + \hat{\mathbf{x}}_0^{old} + \bar{\mathbf{x}}_0^{new}, \end{aligned} \tag{4.3.13}$$

that gives

$$\bar{\mathbf{x}}_0^{new} = \bar{\mathbf{x}}_0^{old} - \hat{\mathbf{x}}_0^{old}. \tag{4.3.14}$$

Nonetheless, in practice, $\bar{\mathbf{P}}_0$ is generally not a realistic representation oh the accuracy of $\bar{\mathbf{x}}_0$, but it is rather used to better condition the matrix $\mathbf{P}$. Consequently, the value of $\bar{\mathbf{x}}_0$ is set to zero for each iteration and $\bar{\mathbf{P}}_0$ is constructed as a diagonal matrix with large diagonal values, so that the second term in equation (4.2.28), related to a priori estimate residuals is small, and more importance is given to the observations residuals.

The entire process is then iterated with the new values for $\mathbf{X}_0^*$, $\bar{\mathbf{x}}_0$ and $\bar{\mathbf{P}}_0$ until convergence, and a common stopping criteria is based on the RMS value of the observation residuals:

$$\mathrm{RMS} = \sqrt{\frac{\sum\limits_{i=1}^{l} \hat{\boldsymbol{\varepsilon}}_i^T \mathbf{W}_i \hat{\boldsymbol{\varepsilon}}_i}{m}}, \tag{4.3.15}$$

where $m = p \times l$ is the total number of observations and $\hat{\boldsymbol{\varepsilon}}_i = \mathbf{y}_i - \mathbf{H}_i \hat{\mathbf{x}}_0$. When the RMS no longer changes the batch filter has converged, i.e. chosen an arbitrarily small value for ϵ, iterations are stopped when:

$$\frac{\left| \mathrm{RMS}^{new} - \mathrm{RMS}^{old} \right|}{\mathrm{RMS}^{old}} \leq \epsilon. \tag{4.3.16}$$

A flow chart summarizing the batch processor algorithm is shown in figure 4.3.1.

4.4 Parameters Segmentation

When a batch filter is used to fit measurements over significantly long periods of time (on the order of several days), the incompleteness of the force model can easily lead to relatively large observation residuals, even if the major perturbations such as geopotential, third body and solar radiation pressure are adequately modeled. This happens because unlike real-time estimators, that *update* their estimate as new information is available, the batch processor as described in the previous Section, fits the entire batch of data with a single set of constant parameters at a certain epoch, which are then considered valid for the entire estimation arc. Nonetheless, in the presence of unpredictable disturbances like spacecraft venting or the complex atmospheric dynamics, it would be desirable for some parameters to exhibit a sort of variability over time in order to reflect and capture these variations, something that a fixed-parameter model is unable to do. A first straightforward solution to this problem is to subdivide certain constant dynamical parameters into a collection of independent sub-parameters, which are constant over shorter spans inside the whole estimation arc, a technique that goes under the name of *parameters segmentation*. For example, to capture the unpredictable atmospheric dynamics whose information is contained in measurement data, even for highly accurate mathematical model for the drag acceleration it might be insufficient to estimate a single C_D value over long arcs, thus, multiple C_{Di} are estimated, each valid for a shorter period of time, usually on the order of few hours. Catania et al. [4] showed how having a time history for the drag coefficient rather than a single fixed value, allowed to detect and absorb unpredictable phenomena during the Launch and Early Operations Phase (LEOP) of Sentinel-3A, like exceptionally high solar activity, that manifested as a peak in C_D values, or even out-gassing during instruments heating that led to negative C_D values, suggesting that is acted as a positive acceleration. As a matter of fact, the process of parameters segmentation augments the estimation state vector dimension, giving a sort of increase in the degrees of freedom with which the filter is able to fit measurement data. Moreover, the time profile of these segmented parameters and their variations from a priori values, provides a powerful tool for understanding the force model accuracy and interpreting anomalous results. This Section provides a detailed insight into the mechanization of parameter segmentation for batch estimation. We focus on the implications this has on the structures and handling of matrices during the propagation of the state transition matrix. Additionally, we discuss the accumulation of information prior to the computation of the least squares solution.

4.4.1 Propagating the State Transition Matrix

The batch filter gives the best estimate of the state through nonlinear least squares. The method is also known as *differential correction* since the nonlinear model is approximated by its linearized counterpart, where the estimated values are *deviations* from a reference solution rather than the

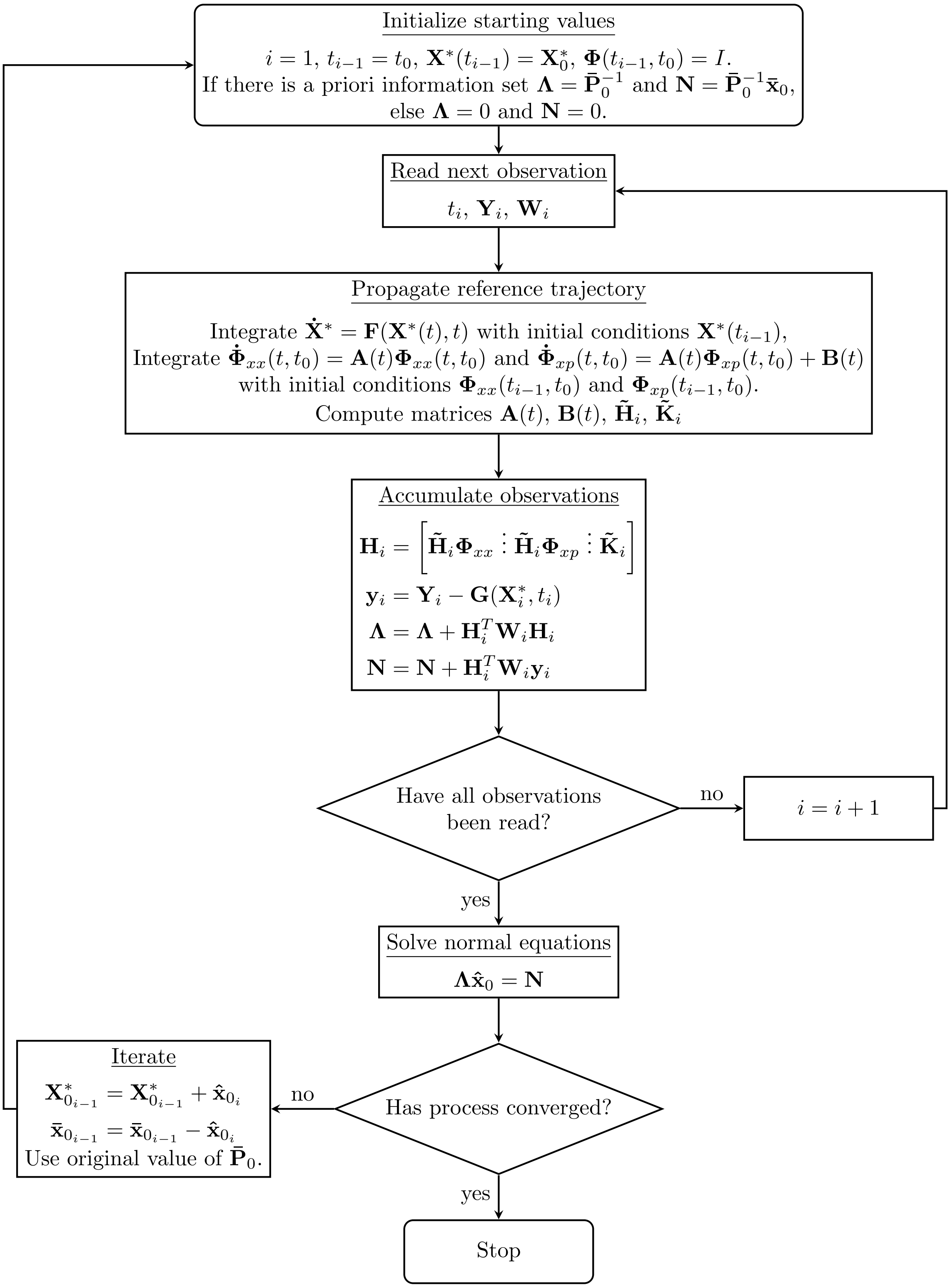
Initialize starting values
$i = 1$, $t_{i-1} = t_0$, $\mathbf{X}^*(t_{i-1}) = \mathbf{X}_0^*$, $\boldsymbol{\Phi}(t_{i-1}, t_0) = I$.
If there is a priori information set $\boldsymbol{\Lambda} = \bar{\mathbf{P}}_0^{-1}$ and $\mathbf{N} = \bar{\mathbf{P}}_0^{-1}\bar{\mathbf{x}}_0$,
else $\boldsymbol{\Lambda} = 0$ and $\mathbf{N} = 0$.
Read next observation
t_i, $\mathbf{Y}_i$, $\mathbf{W}_i$
Propagate reference trajectory
Integrate $\dot{\mathbf{X}}^* = \mathbf{F}(\mathbf{X}^*(t), t)$ with initial conditions $\mathbf{X}^*(t_{i-1})$,
Integrate $\dot{\boldsymbol{\Phi}}_{xx}(t, t_0) = \mathbf{A}(t)\boldsymbol{\Phi}_{xx}(t, t_0)$ and $\dot{\boldsymbol{\Phi}}_{xp}(t, t_0) = \mathbf{A}(t)\boldsymbol{\Phi}_{xp}(t, t_0) + \mathbf{B}(t)$
with initial conditions $\boldsymbol{\Phi}_{xx}(t_{i-1}, t_0)$ and $\boldsymbol{\Phi}_{xp}(t_{i-1}, t_0)$.
Compute matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\tilde{\mathbf{H}}_i$, $\tilde{\mathbf{K}}_i$
Accumulate observations
$\mathbf{H}_i = \left[\tilde{\mathbf{H}}_i\boldsymbol{\Phi}_{xx} \vdots \tilde{\mathbf{H}}_i\boldsymbol{\Phi}_{xp} \vdots \tilde{\mathbf{K}}_i\right]$
$\mathbf{y}_i = \mathbf{Y}_i - \mathbf{G}(\mathbf{X}_i^*, t_i)$
$\boldsymbol{\Lambda} = \boldsymbol{\Lambda} + \mathbf{H}_i^T\mathbf{W}_i\mathbf{H}_i$
$\mathbf{N} = \mathbf{N} + \mathbf{H}_i^T\mathbf{W}_i\mathbf{y}_i$
Have all observations been read?
no
$i = i + 1$
yes
Solve normal equations
$\boldsymbol{\Lambda}\hat{\mathbf{x}}_0 = \mathbf{N}$
Has process converged?
no
Iterate
$\mathbf{X}_{0_{i-1}}^* = \mathbf{X}_{0_{i-1}}^* + \hat{\mathbf{x}}_{0_i}$
$\bar{\mathbf{x}}_{0_{i-1}} = \bar{\mathbf{x}}_{0_{i-1}} - \hat{\mathbf{x}}_{0_i}$
Use original value of $\bar{\mathbf{P}}_0$.
yes
Stop

Figure 4.3.1: Flow chart for the batch processor algorithm.

solution itself and the process is then iterated to convergence. For a conventional orbit determination problem solved with a batch processor, we saw how the state vector usually comprises the satellite's position $\mathbf{r}$, velocity $\mathbf{v}$ and a set of constant geometrical and dynamical parameters $\mathbf{P}$, i.e, $\mathbf{X} = \begin{bmatrix}\mathbf{r} & \mathbf{v} & \mathbf{P}\end{bmatrix}^T$, whose estimated values refer to a reference epoch t_0. For a general treatment of the parameters segmentation approach, we first need to address the new structure of the state vector $\mathbf{X}$. Now, we must assume that certain parameters contained in $\mathbf{P}$ are segmented, meaning that they occupy more than one "slot" in the state vector, precisely one for each of their sub-parameters. In other words, the parameter vector $\mathbf{P}$ is further subdivided into a portion of constant parameters $\mathbf{P}_c$ and a portion of parameters that are segmented, that we will identify as $\mathbf{P}_{sk}$, where the index $s = 1, \ldots, S$ indicates the particular parameter that is considered (C_D, C_R, ...), while $k = 1, \ldots, K$ refers to the single sub-parameter that derives from segmentation. Consequentially, the state vector $\mathbf{X}$ becomes:

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{P}_c \\ \mathbf{P}_{sk} \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{P}_c \\ P_{11} \\ P_{12} \\ \vdots \\ P_{SK} \end{bmatrix} \tag{4.4.1}$$

and we recall that if $\mathbf{X}^*$ is a reference solution, the deviations state vector $\mathbf{x}$ will be defined as

$$\mathbf{x} = \mathbf{X} - \mathbf{X}^* = \begin{bmatrix} \mathbf{r} - \mathbf{r}^* \\ \mathbf{v} - \mathbf{v}^* \\ \mathbf{P}_c - \mathbf{P}_c^* \\ \mathbf{P}_{sk} - \mathbf{P}_{sk}^* \end{bmatrix}. \tag{4.4.2}$$

As an example, consider the case in which over an arc of 24 hours, a single C_R parameter is estimated, meaning that its validity corresponds to the whole estimation arc, while the C_D parameter is segmented into twelve 2-hours validity C_D's. Here, the state vector is structured as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ C_R \\ C_{D1} \\ C_{D2} \\ \vdots \\ C_{D12} \end{bmatrix}.$$

Next, we recall that the state transition matrix $\mathbf{\Phi}(t_i, t_0)$ in the batch filter, plays the important role of mapping state deviations $\mathbf{x}_0$ at epoch t_0 to deviations $\mathbf{x}_i$ of the state at each observation time t_i, which are then mapped into measurement deviations $\mathbf{y}_i$ with the observation-state mapping matrix $\mathbf{H}_i$ as described in Chapter 3. If $\mathbf{X}$ is the state vector, the state transition matrix can be defined as

$$\mathbf{\Phi}(t, t_0) = \left[\frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)}\right] = \left[\frac{\partial \mathbf{X}}{\partial \mathbf{X}_0}\right], \tag{4.4.3}$$

where $\mathbf{X}_0$ is the state vector at epoch t_0 and the equation above describes the *system state transition matrix.* For the orbit determination problem, if we partition the state vector into dynamical and parameter components as

$$\mathbf{X}_d = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix}, \qquad \mathbf{X}_p = \mathbf{P} = \begin{bmatrix} \mathbf{P}_c \\ \mathbf{P}_{sk} \end{bmatrix}, \tag{4.4.4}$$

this matrix can be partitioned into *dynamical state transition matrix* $\mathbf{\Phi}_{xx}(t_i, t_0)$ and *parameter state transition matrix* $\mathbf{\Phi}_{xp}(t_i, t_0)$, where

$$\mathbf{\Phi}_{xx}(t_i, t_0) = \left[\frac{\partial \mathbf{X}_d}{\partial \mathbf{X}_{d0}}\right], \qquad \mathbf{\Phi}_{xp}(t_i, t_0) = \left[\frac{\partial \mathbf{X}_d}{\partial \mathbf{P}_0}\right], \tag{4.4.5}$$

meaning that the *system state transition matrix* can be written as

$$\mathbf{\Phi}(t_i, t_0) = \begin{bmatrix} \mathbf{\Phi}_{xx} & \mathbf{\Phi}_{xp} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{4.4.6}$$

where, if $\mathbf{P}$ is of dimension p, $\mathbf{\Phi}_{xx}$ is 6×6, $\mathbf{\Phi}_{xp}$ is $6 \times p$, the lower left part is a $p \times 6$ null matrix and the lower right part is an $p \times p$ identity matrix. The structure highlighted in (4.4.6) motivates the partition into dynamical and parameter components and the differential equations for propagating the state transition matrices in (4.4.5) are the following (see Chapter 3):

$$\dot{\mathbf{\Phi}}_{xx}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xx}(t, t_0), \qquad \mathbf{\Phi}_{xx}(t_0, t_0) = \mathbf{I}, \tag{4.4.7}$$

$$\dot{\mathbf{\Phi}}_{xp}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t, t_0) + \mathbf{B}(t), \qquad \mathbf{\Phi}_{xp}(t_0, t_0) = \mathbf{0}. \tag{4.4.8}$$

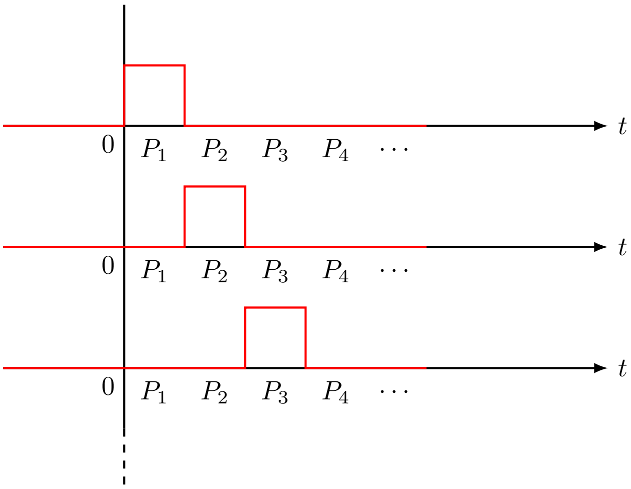

Figure 4.4.1: Schematic representation of the parameters segmentation procedure. Note how the parameter P is segmented into N P_i piecewise constant sub-parameters, each active during subsequent periods of validity.

For what concerns the parameter segmentation procedure, no particular considerations are needed for the dynamical state transition matrix $\mathbf{\Phi}_{xx}$, as its algorithmic implementation remains unchanged from the conventional least squares estimation. However, if some parameters are segmented, the propagation of the parameter state transition matrix $\mathbf{\Phi}_{xp}$ requires careful considerations on its computational implementation. As mentioned earlier, in the linearized orbit model the state transition matrix allows to map deviations of the state at epoch t_0 to deviations of the state at epoch t_i. For the dynamical part (position and velocity) this means that along the entire estimation arc, information contained in every measurement up to the last observation, will contribute at improving the estimate of the satellite's initial position and velocity at epoch t_0. If parameters were not segmented, this would also be true for them and the state transition matrix propagation would proceed unhindered without any particular regard. However, when segmentation is performed for some of the parameters contained in $\mathbf{P}$, the resulting sub-parameters have a certain period of validity as schematically shown in figure 4.4.1, thus, they are active only during that period. This means that a particular sub-parameter P_{sk}, valid throughout the interval Δt_{sk}, should not be influenced by the information content of measurements outside its span of validity, that is to say, the forcing term $\mathbf{B}(t)$ of equation (4.4.8) must be structured so that it properly forces the currently active parameters only. If we follow the definition of $\mathbf{P}$ in (4.4.4) and assume that $\mathbf{P}_c$ has dimensions $c \times 1$, while $\mathbf{P}_{sk}$ is composed of s parameters each with its own k sub-parameters and $k_1 + k_2 + \cdots + k_s = p$, then, the parameter state transition matrix $\boldsymbol{\Phi}_{xp}$ has the following structure:

$$\boldsymbol{\Phi}_{xp} = \begin{array}{c} \begin{array}{ccccc} \overbrace{\qquad}^{6\times c} & \overbrace{\qquad}^{6\times k_1} & \overbrace{\qquad}^{6\times k_2} & & \overbrace{\qquad}^{6\times k_s} \end{array} \\ \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{r}}{\partial \mathbf{P}_{1k_1}} & \frac{\partial \mathbf{r}}{\partial \mathbf{P}_{2k_2}} & \cdots & \frac{\partial \mathbf{r}}{\partial \mathbf{P}_{Sk_s}} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{v}}{\partial \mathbf{P}_{1k_1}} & \frac{\partial \mathbf{v}}{\partial \mathbf{P}_{2k_2}} & \cdots & \frac{\partial \mathbf{v}}{\partial \mathbf{P}_{Sk_s}} \end{bmatrix}_{6\times p} \end{array}, \tag{4.4.9}$$

where the first c columns are related to those parameters that are not segmented, i.e., their period of validity corresponds to the whole estimation arc, while all other groups of k_i columns, $i = 1, 2, \ldots, S$, are occupied by the segmented parameters. Note how every parameter P_{sk_s} has its own number of sub-parameters k_s, depending on the length of its validity period. We now proceed to examine the form of equation (4.4.8) during its integration over time, given all the considerations above and the structure highlighted in (4.4.9). We recall that the definitions for matrices $\mathbf{A}(t)$ and $\mathbf{B}(t)$ are:

$$\mathbf{A}(t) = \left[\frac{\partial \mathbf{F}(t)}{\partial \mathbf{X}_d}\right]^*_{6\times 6}, \qquad \mathbf{B}(t) = \left[\frac{\partial \mathbf{F}(t)}{\partial \mathbf{P}}\right]^* = \begin{bmatrix} \mathbf{0} \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}} \end{bmatrix}_{6\times p}, \tag{4.4.10}$$

in particular, note how the forcing term $\mathbf{B}(t)$ contains the partial derivatives of acceleration $\mathbf{a}$ with respect to the parameters $\mathbf{P}$. Then, given the initial conditions, as integration begins we have that $\mathbf{Phi}_{xp}(t_0, t_0)$ is the null matrix, while $\mathbf{B}(t_0)$ has non-null columns only in correspondence of the non-segmented parameters $\mathbf{P}_c$ and all the first sub-parameters that are active at t_0, namely

$$\boldsymbol{\Phi}_{xp}(t_0, t_0) = \mathbf{0}, \tag{4.4.11}$$

and

$$\mathbf{B}(t_0) = \begin{array}{c} \begin{array}{cccccccccccccc} \overbrace{\quad}^{6\times c} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad}^{6\times k_1}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad}^{6\times k_2}} & & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad}^{6\times k_s}} \end{array} \\ \begin{bmatrix} \mathbf{0} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{a}}{\partial P_{11}} & 0 & \cdots & 0 & \frac{\partial \mathbf{a}}{\partial P_{21}} & 0 & \cdots & 0 & \cdots & \frac{\partial \mathbf{a}}{\partial P_{S1}} & 0 & \cdots & 0 \end{bmatrix}_{6\times p} \end{array}, \tag{4.4.12}$$

For the sake of clarity we will assume that every segmented parameter has been divided into the same number of sub-parameters, that is to say, they all have equal periods of validity and $k_1 = k_2 = \cdots = k_s$, furthermore, all these periods will be indicated by Δt_1, Δt_2,…, Δt_s. As propagation continues, for $t > t_0$ we will eventually reach the end of the first validity period Δt_1 and fall into Δt_2. Here, we need to *activate* propagation for all the second sub-parameters by properly placing the forcing terms and by zeroing out the previously populated columns of $\mathbf{B}(t)$. So, for $t_k \in \Delta t_2$ the parameter state transition matrix and e forcing matrix are:

$$\boldsymbol{\Phi}_{xp}(t_k, t_0) = \begin{array}{c} \begin{array}{ccccccccccccc} \overbrace{\quad}^{6\times c} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_1}} & & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_s}} \end{array} \\ \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{r}}{\partial P_{11}} & \frac{\partial \mathbf{r}}{\partial P_{12}} & 0 & \cdots & 0 & \cdots & \frac{\partial \mathbf{r}}{\partial P_{S1}} & \frac{\partial \mathbf{r}}{\partial P_{S2}} & 0 & \cdots & 0 \\ \frac{\partial \mathbf{v}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{v}}{\partial P_{11}} & \frac{\partial \mathbf{v}}{\partial P_{12}} & 0 & \cdots & 0 & \cdots & \frac{\partial \mathbf{v}}{\partial P_{S1}} & \frac{\partial \mathbf{v}}{\partial P_{S2}} & 0 & \cdots & 0 \end{bmatrix}_{6\times p} \end{array}, \tag{4.4.13}$$

and

$$\mathbf{B}(t_k) = \left[\begin{array}{cccccccccccc} \overbrace{\quad}^{6\times c} & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_1}} & & \multicolumn{5}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_s}} \\ \mathbf{0} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}_c} & 0 & \frac{\partial \mathbf{a}}{\partial P_{11}} & 0 & \cdots & 0 & \cdots & 0 & \frac{\partial \mathbf{a}}{\partial P_{S1}} & 0 & \cdots & 0 \end{array}\right]_{6\times p}, \qquad (4.4.14)$$

note how in $\mathbf{B}(t)$, only the second column of every $6 \times k_i$ sub-matrix contains the forcing term, underlining that propagation has arrived to the Δt_2 interval. The transition matrix $\mathbf{\Phi}_{xp}$ on the other hand, still preserves columns associated to previously active parameters, the ones valid throughout Δt_1 in the current case. This latter aspect illustrates the fact that the evolution of $\mathbf{\Phi}_{xp}$ columns associated to now inactive sub-parameters, does not terminate with the end of the corresponding period of validity, it simply continues freely without any forcing term. As a matter of fact, keeping the free evolution of these columns during integration is necessary to keep "memory" of all the past sub-parameters history. All these past values will inevitably influence future states as integration is never reinitialized in the batch processor and every information is mapped to the same reference epoch t_0, thus, every sub-parameter estimation is the result of a proper combination of information from measurements contained in its validity period and the past history of all previous ones. Referring to equations (4.4.11), (4.4.12), (4.4.13) and (4.4.14), the integration goes on likewise for every subsequent column along the whole estimation arc, until every Δt_k has been covered and the parameter state transition matrix is entirely populated. Accordingly, when integration falls into the last validity period, i.e., $t_k \in \Delta t_s$ the structures are

$$\mathbf{\Phi}_{xp}(t_k, t_0) = \left[\begin{array}{cccccccccc} \overbrace{\quad}^{6\times c} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_1}} & & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_s}} \\ \frac{\partial \mathbf{r}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{r}}{\partial P_{11}} & \frac{\partial \mathbf{r}}{\partial P_{12}} & \cdots & \frac{\partial \mathbf{r}}{\partial P_{1k_1}} & \cdots & \frac{\partial \mathbf{r}}{\partial P_{S1}} & \frac{\partial \mathbf{r}}{\partial P_{S2}} & \cdots & \frac{\partial \mathbf{r}}{\partial P_{Sk_s}} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{P}_c} & \frac{\partial \mathbf{v}}{\partial P_{11}} & \frac{\partial \mathbf{v}}{\partial P_{12}} & \cdots & \frac{\partial \mathbf{v}}{\partial P_{1k_1}} & \cdots & \frac{\partial \mathbf{v}}{\partial P_{S1}} & \frac{\partial \mathbf{v}}{\partial P_{S2}} & \cdots & \frac{\partial \mathbf{v}}{\partial P_{Sk_s}} \end{array}\right]_{6\times p}, \qquad (4.4.15)$$

and

$$\mathbf{B}(t_k) = \left[\begin{array}{cccccccccc} \overbrace{\quad}^{6\times c} & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_1}} & & \multicolumn{4}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{6\times k_s}} \\ \mathbf{0} & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}_c} & 0 & \cdots & 0 & \frac{\partial \mathbf{a}}{\partial P_{Sk_1}} & \cdots & 0 & \cdots & 0 & \frac{\partial \mathbf{a}}{\partial P_{Sk_s}} \end{array}\right]_{6\times p}. \qquad (4.4.16)$$

4.4.2 Accumulating Matrices

Section 4.2 showed how we can arrive at a compact form for the observation equation (4.2.2). The total number l of elements in the measurement deviations vector $\mathbf{y}$ can be quite high for long estimation arcs and if one were to build the normal equations (4.2.13) by matrix multiplications, which involve the $l \times n$ observation-state map $\mathbf{H}$, the process could become highly demanding by a computational standpoint. For this reason, Section 4.2 also shown how it is surely more convenient to accumulate both sides of the normal equations as propagation and measurement deviations are processed in sequence (we assume that $\mathbf{W}$ is the identity matrix for simplicity):

$$\begin{aligned}\mathbf{H}^T\mathbf{H} &= \sum_{i=1}^{m}\mathbf{H}_i^T\mathbf{H}_i \\ &= \sum_{i=1}^{m}\left[\tilde{\mathbf{H}}_i\mathbf{\Phi}_{xx}(t_i,t_0) \quad \tilde{\mathbf{H}}_i\mathbf{\Phi}_{xp}(t_i,t_0)\right]^T\left[\tilde{\mathbf{H}}_i\mathbf{\Phi}_{xx}(t_i,t_0) \quad \tilde{\mathbf{H}}_i\mathbf{\Phi}_{xp}(t_i,t_0)\right]\end{aligned} \tag{4.4.17}$$

$$\begin{aligned}\mathbf{H}^T\mathbf{y} &= \sum_{i=1}^{m}\mathbf{H}_i^T\mathbf{y}_i \\ &= \sum_{i=1}^{m}\left[\tilde{\mathbf{H}}_i\mathbf{\Phi}_{xx}(t_i,t_0) \quad \tilde{\mathbf{H}}_i\mathbf{\Phi}_{xp}(t_i,t_0)\right]^T\mathbf{y}_i,\end{aligned} \tag{4.4.18}$$

where, if every measurement is processed individually, the size of $\tilde{\mathbf{H}}_i$ is 1×6, $\mathbf{\Phi}_{xx}(t_i,t_0)$ is 6×6, $\Phi_{xp}(t_i,t_0)$ is $6 \times p$ and the measurement deviation vector is actually a scalar, but for clarity, it will still be treated as 1×1 vector $\mathbf{y}_i$. Another important assumption, that simplifies the following treatment, is that only dynamical parameters are considered, which signifies that there is no need for a $\mathbf{K}$ matrix, see equation (4.1.4). Note that this simplification is backed up by the fact that segmented parameters are always dynamical (C_D, C_R, ...) rather than geometrical (station coordinates, phase center offset, ...). Accumulating matrices in this manner allows to handle multiplications in a more effective way. While the dynamical part of the state vector is always composed of six elements like position and velocity, in batch filter applications it is not uncommon for the parameter vector $\mathbf{P}$ to be very large and sparsely populated, especially when segmentation is performed and what was once a single parameter has now transformed into multiple sub-parameters. For example, we might need to estimate a satellite position and velocity over a week, along with the coordinates of tens of ground stations and several dynamical parameters with a relatively short validity period. In such situations the state vector dimension n quickly becomes very large, and the $\mathbf{H}^T\mathbf{H}$ matrix will most likely be sparse. Under these circumstances, it becomes clear how not only it is convenient to accumulate matrices as in (4.4.18) and (4.4.17), but every multiplication involved for each i-th measurement should be performed to favor computational efficiency, while also ensuring that information related to segmented parameters is accumulated, thereby preserving the history of all previous values.

To do so, using the terminology developed for the parameter segmentation process, let us start by examining the matrix structures involved when the propagation has led us to time $t_k \in \Delta t_k$, and to lighten the treatise, we assume that all dynamical parameters are segmented, so there is no $\mathbf{P}_c$. Here, the state transition matrices are available as

$$\mathbf{\Phi}_{xx}(t_k,t_0) = \begin{bmatrix}\Phi_{rr}(t_k,t_0) & \Phi_{rv}(t_i,t_0) \\ \Phi_{vr}(t_k,t_0) & \Phi_{vv}(t_i,t_0)\end{bmatrix}_{6\times 6} \tag{4.4.19}$$

and

$$\mathbf{\Phi}_{xp}(t_k,t_0) = \begin{bmatrix}\overbrace{\frac{\partial\mathbf{r}}{\partial P_{11}} \cdots \frac{\partial\mathbf{r}}{\partial P_{1k}} \; 0 \cdots 0}^{6\times k_1} & \cdots & \overbrace{\frac{\partial\mathbf{r}}{\partial P_{S1}} \cdots \frac{\partial\mathbf{r}}{\partial P_{Sk}} \; 0 \cdots 0}^{6\times k_s} \\ \frac{\partial\mathbf{v}}{\partial P_{11}} \cdots \frac{\partial\mathbf{v}}{\partial P_{1k}} \; 0 \cdots 0 & \cdots & \frac{\partial\mathbf{v}}{\partial P_{S1}} \cdots \frac{\partial\mathbf{v}}{\partial P_{Sk}} \; 0 \cdots 0\end{bmatrix}_{6\times p}, \tag{4.4.20}$$

Note that for the parameter state transition matrix $\mathbf{\Phi}_{xp}$ the time indices are (t_k,t_0) indicating that this matrix is used to map from the current time of observation t_k to the beginning of the whole estimation arc t_0. As we can see, $\mathbf{\Phi}_{xp}$ is composed of S submatrices, one for every $\mathbf{P}_{sk}$, and each is populated up to the $k-$th column, since all previous ones, up to the $(k-1)$-th, have been evolving unforcedly, see (4.4.13) and (4.4.14). Now, every $\mathbf{H}_i$ matrix is formally computed as

$$\mathbf{H}_i = \left[\tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx}(t_i, t_0) \quad \tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xp}(t_i, t_0)\right]_{1\times n}, \tag{4.4.21}$$

however, considering that $\boldsymbol{\Phi}_{xp}$ has many null columns due to segmentation, it in not convenient to perform the product $\tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xp}(t_i, t_0)$ directly, rather it is more advisable to multiply $\tilde{\mathbf{H}}_i$ with only the non-null columns of $\boldsymbol{\Phi}_{xp}$, which is made possible the row-by-column product. If we then identify every non-null portion of $\boldsymbol{\Phi}_{xp}$ associated to each set of parameters $\mathbf{P}_{sk}$ with

$$\boldsymbol{\Phi}^{sk}_{xp} = \begin{bmatrix} \dfrac{\partial \mathbf{r}}{\partial P_{s1}} & \dfrac{\partial \mathbf{r}}{\partial P_{s2}} & \cdots & \dfrac{\partial \mathbf{r}}{\partial P_{sk_s}} \\ \dfrac{\partial \mathbf{v}}{\partial P_{s1}} & \dfrac{\partial \mathbf{v}}{\partial P_{s2}} & \cdots & \dfrac{\partial \mathbf{v}}{\partial P_{sk_s}} \end{bmatrix}_{6\times k}, \tag{4.4.22}$$

and omitting time indices (t_i, t_0) for compactness, equation (4.4.21) can be rewritten as

$$\mathbf{H}_i = \Big[\overbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx}}^{1\times 6} \quad \overbrace{\underbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}^{1k}_{xp}}_{1\times k} \quad \mathbf{0}}^{1\times k_1} \quad \overbrace{\underbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}^{2k}_{xp}}_{1\times k} \quad \mathbf{0}}^{1\times k_2} \quad \cdots \quad \overbrace{\underbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}^{Sk}_{xp}}_{1\times k} \quad \mathbf{0}}^{1\times k_s}\Big]. \tag{4.4.23}$$

where this matrix has dimensions $1 \times n$, with $\boldsymbol{\Phi}_{xx}$ being related to the position-velocity portion of the state vector, while every $\tilde{\mathbf{H}}_i \boldsymbol{\Phi}^{sk}_{xp}$ is associated to every segmented parameter that was or is currently active at time $t_k \in \Delta t_k$.

Following from the definition of $\mathbf{H}_i$ given above, the next step focuses on the implications it has on the structure of the resulting product $\mathbf{H}_i^T \mathbf{H}_i$ that is computed in equation (4.4.18) for every observation. This product gives the normal matrix, which is often referred to as **HTH**, pronounced "h-t-h". If, for compactness, we use the definitions:

$$\begin{aligned} \tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx} &= \mathbf{H}_{xx} \\ \tilde{\mathbf{H}}_i \boldsymbol{\Phi}^{sk}_{xp} &= \mathbf{H}_{sk}, \end{aligned}$$

then, through simple row-by-columns product rules applied to the structure highlighted in (4.4.23), we arrive at

$$\mathbf{HTH} = \mathbf{H}_i^T \mathbf{H}_i = \begin{bmatrix} \mathbf{H}_{xx}^T \mathbf{H}_{xx} & \mathbf{H}_{xx}^T \mathbf{H}_{1k} & \mathbf{0} & \mathbf{H}_{xx}^T \mathbf{H}_{2k} & \mathbf{0} & \cdots & \mathbf{H}_{xx}^T \mathbf{H}_{sk} & \mathbf{0} \\ \mathbf{H}_{1k}^T \mathbf{H}_{xx} & \mathbf{H}_{1k}^T \mathbf{H}_{1k} & \mathbf{0} & \mathbf{H}_{1k}^T \mathbf{H}_{2k} & \mathbf{0} & \cdots & \mathbf{H}_{1k}^T \mathbf{H}_{sk} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_{2k}^T \mathbf{H}_{xx} & \mathbf{H}_{2k}^T \mathbf{H}_{1k} & \mathbf{0} & \mathbf{H}_{2k}^T \mathbf{H}_{2k} & \mathbf{0} & \cdots & \mathbf{H}_{2k}^T \mathbf{H}_{sk} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{H}_{sk}^T \mathbf{H}_{xx} & \mathbf{H}_{sk}^T \mathbf{H}_{1k} & \mathbf{0} & \mathbf{H}_{sk}^T \mathbf{H}_{2k} & \mathbf{0} & \cdots & \mathbf{H}_{sk}^T \mathbf{H}_{sk} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}_{n\times n}. \tag{4.4.24}$$

As can be seen, due to symmetry of the product and sparseness that arises from segmentation, only the highlighted portion must be computed and then accurately positioned inside the **HTH** matrix, thereby avoiding multiplication by zero and increasing computational efficiency. Hence, the number of operations to be carried out in equation (4.4.24) is not $n \times n$, as would be the case for direct execution of $\mathbf{H}_i^T \mathbf{H}_i$, but it is drastically reduced considering that

$$\begin{aligned}\mathbf{H}_{xx}^T\mathbf{H}_{xx} &= 6\times 6\\ \mathbf{H}_{xx}^T\mathbf{H}_{sk} &= 6\times k\\ \mathbf{H}_{sk_i}^T\mathbf{H}_{sk_j} &= k_i\times k_j,\end{aligned}$$

which is true especially at the beginning, when only few parameters are active and the matrix is highly sparse. Examples of the **HTH** structure deriving from parameter segmentation can be found in R. Kroes (2006) [14]. Only when enough time has passed so that all sub-parameters have been activated, the matrix becomes entirely populated and the only computational advantage we can take derives from symmetry.

When dealing with segmentation of parameters, particular attention should be posed to their observability. Clearly, the idea is that the more a parameter gets segmented, the more it will be able to absorb time-varying unmodeled or unpredictable effects, nevertheless, as its validity period gets shrunk, it becomes less observable as fewer observation may be available for every single sub-parameter estimation. Additionally, considerable attention is required when the validity period is not and integer sub-multiple of the whole estimation arc. In this situation, the “trailing” periods close to the arc’s end might be so truncated as to become too small to provide a sufficient number of observation that ensures parameter’s observability, thus, detailed check on the consistency between validity duration and observation density is advised.

4.5 Empirical Accelerations in Batch Estimation

In the context of precise orbit determination using nonlinear least squares estimators such as the Batch filter, a crucial aspect is the completeness and precision of the force model that is used to propagate the dynamical and variational equations. Unlike real-time filters, where new information is processed as soon as it is available over short periods of time, the batch filter collects observations over a long arc and subsequently processes them to obtain an estimate of the state at a single reference epoch t_0. This characteristic makes it more sensitive to the lack of precision modeling or approximations in certain forces, especially for perturbations that exhibit tangible effects in the long term, whereas a real-time filter would be rather insensible to them, since it is periodically updating the state estimate along the trajectory. For this reason, batch filters usually exploit very accurate models for the accelerations acting on a satellite, where accounting for weaker perturbations like Earth radiation pressure and thermal re-radiation, can lead to better fits of the observed data and lower observation residuals. However, when dealing with real data, even when every possible effort has been made to accurately represent the forces acting on a satellite, the observation residuals may still not be satisfactory enough, or may still exhibit some information content, suggesting that there are perturbative effects that the model is not able to capture. In these circumstances, a powerful tool that can help in absorbing these unexplained behaviours is the addition of an *empirical acceleration*. The general model has been introduced in Section 2.7 and as the name suggests, these accelerations has no physical counterpart, meaning that it is not used to model the effects of a particular natural phenomena. Instead, its purpose is to take the place of every unknown acceleration that comes from unmodeled forces or approximations, leading to a better fitting of data through estimation of its parameters. For a satellite orbiting near Earth, it is reasonable to expect that the perturbations acting on it, will show a somewhat periodic behavior, with frequencies mostly proportional to the orbital period, if not constant. For this reason, empirical accelerations are often modeled as a Fourier series as follows

$$\mathbf{a}_{emp}^{RTN} = \mathbf{A} + \sum_{i=1}^{2}\mathbf{D}_i\cos(iu) + \mathbf{S}_i\sin(iu). \tag{4.5.1}$$

Here, the summation is truncated at the second term as it is common practice to model the empirical acceleration with frequencies up to half the orbital period, moreover, the harmonic terms contain the *argument of latitude* u, as described in equation (2.7.3). As indicated by the superscript RTN, these accelerations are modeled in Radial, Transverse and Normal components. This orthogonal reference system follows the satellite motion and gives a more intuitive insight on the nature of the acting perturbations, making it easier to interpret results. Consequently, equation (4.5.1) can be expanded in its components as follows

$$\mathrm{a}^{R}_{emp} = \mathrm{A}_R + \sum_{i=1}^{2} \mathrm{D}_{R_i}\cos(iu) + \mathrm{S}_{R_i}\sin(iu) \tag{4.5.2}$$

$$\mathrm{a}^{T}_{emp} = \mathrm{A}_T + \sum_{i=1}^{2} \mathrm{D}_{T_i}\cos(iu) + \mathrm{S}_{T_i}\sin(iu) \tag{4.5.3}$$

$$\mathrm{a}^{N}_{emp} = \mathrm{A}_N + \sum_{i=1}^{2} \mathrm{D}_{N_i}\cos(iu) + \mathrm{S}_{N_i}\sin(iu), \tag{4.5.4}$$

and we define the following vectors of parameters

$$\mathbf{A} = \begin{Bmatrix} \mathrm{A}_R \\ \mathrm{A}_T \\ \mathrm{A}_N \end{Bmatrix}, \qquad \mathbf{C}_i = \begin{Bmatrix} \mathrm{D}_R \\ \mathrm{D}_T \\ \mathrm{D}_N \end{Bmatrix}_i, \qquad \mathbf{S}_i = \begin{Bmatrix} \mathrm{S}_R \\ \mathrm{S}_T \\ \mathrm{S}_N \end{Bmatrix}_i, \tag{4.5.5}$$

that are estimated in the batch processor to completely define the empirical acceleration $\mathbf{a}_{emp}$. Note that if $i = 1, 2$ and the bias $\mathbf{A}$ is present, a total of 15 parameters must be estimated.

4.5.1 Empirical Accelerations - Parameters Estimation

Concerning the estimation of parameters contained in the empirical acceleration model through batch processing, we must focus on the structure of the *Parameter State Transition Matrix* $\mathbf{\Phi}_{xp}$ (PSTM) and the consequences it has on assembling the matrices for the normal equations in the least squares formulation. In addition, leveraging the results obtained in the previous Section, the estimation of such parameters is also discussed in the perspective of parameter segmentation. This is because, in addition to implementing a model for empirical accelerations, it can be beneficial to segment the same parameters that compose it, attributing to each a certain validity period. First, we recall that the variational equations for the PSTM are given by

$$\dot{\mathbf{\Phi}}_{xp}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t, t_0) + \mathbf{B}(t), \qquad \mathbf{\Phi}_{xp}(t, t_0) = \mathbf{0}, \tag{4.5.6}$$

where

$$\mathbf{A}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right]^*, \qquad \mathbf{B}(t) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{P}}\right]^*. \tag{4.5.7}$$

Here, $\mathbf{F}(\mathbf{X}, \mathbf{P}, t)$ represents the functional of the nonlinear dynamical state $\mathbf{X}$ and $\mathbf{P}$ is the vector of dynamical parameters to be estimated. Note that, as usual, both $\mathbf{A}(t)$ and $\mathbf{B}(t)$ are evaluated along the reference trajectory $\mathbf{X}^*$. At first, for simplicity, we will assume that the parameters related to the empirical acceleration model are the only dynamical parameters to be estimated in the state vector, and that their validity is equal to the entire estimation arc. Equation (4.5.6) shows the importance of the term $\mathbf{B}(t)$ as a forcing function that represents sensitivity of the dynamical state to variations in the parameters values. The structure of this matrix, in the case of empirical accelerations, depends on the order of the parameters within the vector $\mathbf{P}$. Therefore, if we choose to gather them in (4.5.5) such that the constant terms are at the beginning and the cyclical terms are ordered according to the harmonic order i, we obtain the following:

$$\mathbf{P} = \begin{bmatrix} \mathrm{A}_R & \mathrm{A}_T & \mathrm{A}_N & \{\mathrm{D}_R & \mathrm{S}_R & \mathrm{D}_T & \mathrm{S}_T & \mathrm{D}_N & \mathrm{S}_N\}_i \end{bmatrix}^T_{1\times 15}, \tag{4.5.8}$$

where the curly braces highlight how parameters D and S for radial, transverse and normal components, are gathered within their harmonic order i. Recall that usually $i = 1, 2$, but it can be arbitrarily extended to any order. Then, the forcing matrix $\mathbf{B}(t)$ can be interpreted as

$$\mathbf{B}(t) = \begin{bmatrix} \frac{\partial \mathbf{v}}{\partial \mathbf{P}} \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial \mathbf{a}}{\partial \mathbf{P}} \end{bmatrix}, \tag{4.5.9}$$

with $\mathbf{v}$ and $\mathbf{a}$ being the velocity and acceleration vectors. Accordingly, we proceed by examining the structure of the lower 3×15 portion, where $\mathbf{a}$ will be the empirical acceleration alone since it is the only one depending on the estimated parameters. Furthermore, since the acceleration components are orthogonal and each dependent on different sets of parameters, we note that partial derivatives for each of them will be non-zero only when taken with respect to corresponding parameters in the RTN system. Thus, if we partition the parameter vector $\mathbf{P}$ as

$$\mathbf{P}_A = \begin{Bmatrix} \mathrm{A}_R \\ \mathrm{A}_T \\ \mathrm{A}_N \end{Bmatrix}, \qquad \mathbf{P}_{CDS_i} = \begin{Bmatrix} \mathrm{D}_R \\ \mathrm{S}_R \\ \mathrm{D}_T \\ \mathrm{S}_T \\ \mathrm{D}_N \\ \mathrm{S}_N \end{Bmatrix}_i, \tag{4.5.10}$$

we can write

$$\begin{aligned} \frac{\partial \mathbf{a}}{\partial \mathbf{P}_A} &= \begin{bmatrix} \frac{\partial \mathrm{a}_R}{\partial \mathrm{A}_R} & 0 & 0 \\ 0 & \frac{\partial \mathrm{a}_T}{\partial \mathrm{A}_T} & 0 \\ 0 & 0 & \frac{\partial \mathrm{a}_N}{\partial \mathrm{A}_N} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{4.5.11}$$

$$\begin{aligned} \frac{\partial \mathbf{a}}{\partial \mathbf{P}_{DS_i}} &= \begin{bmatrix} \frac{\partial \mathrm{a}_R}{\partial \mathrm{D}_{R_i}} & \frac{\partial \mathrm{a}_R}{\partial \mathrm{S}_{R_i}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial \mathrm{a}_T}{\partial \mathrm{D}_{T_i}} & \frac{\partial \mathrm{a}_T}{\partial \mathrm{S}_{T_i}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial \mathrm{a}_N}{\partial \mathrm{D}_{N_i}} & \frac{\partial \mathrm{a}_N}{\partial \mathrm{S}_{N_i}} \end{bmatrix} \\ &= \begin{bmatrix} \cos(iu) & \sin(iu) & 0 & 0 & 0 & 0 \\ 0 & 0 & \cos(iu) & \sin(iu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos(iu) & \sin(iu) \end{bmatrix}, \end{aligned} \tag{4.5.12}$$

and the structure of the complete matrix $\mathbf{B}(t)$, considering cyclical terms up to twice-per-rev, becomes

$$\mathbf{B}(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times6} \\ \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_A}\right)_{3\times3} & \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_{DS_1}}\right)_{3\times6} & \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_{DS_2}}\right)_{3\times6} \end{bmatrix}_{6\times15}. \tag{4.5.13}$$

The above expression for $\mathbf{B}(t)$, when used in equation (4.5.6), enables the propagation of the parameter state transition matrix when the empirical acceleration model is implemented, as well as the estimation of its parameters when their validity period coincides with the entire estimation arc.

4.5.2 Empirical Accelerations - Parameter Segmentation

We will now proceed to extend the results to the case in which every parameter involved in the empirical acceleration model has a period of validity that is shorter than the whole estimation arc, thus, they are segmented into a certain number of sub-parameters. For clarity, we assume that every parameter undergoes the same segmentation process, i.e., for every RTN component, $\mathbf{A}$, $\mathbf{D}$ and $\mathbf{S}$ are divided into the same number N of sub-parameters. In this instance, using a terminology similar to the one of Section 4.4, we rewrite equation (4.5.1) as

$$\mathbf{a}_{emp}^{RTN}(t_k) = \mathbf{A}_k + \sum_{i=1}^{2} \mathbf{D}_{ik}\cos(iu) + \mathbf{S}_{ik}\sin(iu), \qquad k = 1,\, 2,\, 3, \ldots,\, N, \tag{4.5.14}$$

where t_k indicates that the expression for $\mathbf{a}_{emp}^{RTN}$ is valid for every $t \in \Delta t_k$, which is the k-th validity period and the parameter vector $\mathbf{P}$ of equation (4.5.8) becomes

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_1 & \mathbf{P}_2 & \mathbf{P}_3 & \cdots & \mathbf{P}_N\end{bmatrix}, \tag{4.5.15}$$

whose dimension is $1 \times (15 \times N)$, and each $\mathbf{P}_k$ of dimension 1×15 is given by

$$\mathbf{P}_k = \begin{bmatrix}\mathrm{A}_R & \mathrm{A}_T & \mathrm{A}_N & \left\{\mathrm{D}_R \quad \mathrm{S}_R \quad \mathrm{D}_T \quad \mathrm{S}_T \quad \mathrm{D}_N \quad \mathrm{S}_N\right\}_i\end{bmatrix}_k^T . \tag{4.5.16}$$

Note how the segmentation of empirical acceleration parameters quickly increases the state vector dimension, as they are present in groups of 15 for each sub-interval of validity. As a result, if the estimation state vector $\mathbf{X}$ contains position, velocity and empirical acceleration parameters only, its dimension goes from $n = 6+15$ for the case of non-segmentation, to $n = 6+(15\times N)$ when those parameters are segmented into N sub-parameters. Consequently, the integration of (4.5.6) must follow the considerations derived in Section 4.4, where we shown how the PSTM is gradually populated as consecutive sub-intervals are covered during propagation, and the forcing matrix $\mathbf{B}(t)$, properly "activates" parameters as soon as they become valid whilst preserving memory of the past. For this particular case this means that, from equation (4.5.13), the matrix $\mathbf{B}(t)$ of dimension $6 \times p$, with $p = 15 \times N$, can be generally represented as

$$\mathbf{B}(t_k) = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_k & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix} \tag{4.5.17}$$

underlining the already established fact that at time $t_k \in \Delta t_k$ only the set of parameters $\mathbf{P}_k$ are forced by their corresponding partial derivatives contained in $\mathbf{B}_k$, that is

$$\mathbf{B}_k = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} & \mathbf{0}_{3\times6} \\ \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_{Ak}}\right)_{3\times3} & \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_{DS_{1k}}}\right)_{3\times6} & \left(\frac{\partial \mathbf{a}}{\partial \mathbf{P}_{DS_{2k}}}\right)_{3\times6} \end{bmatrix}_{6\times15}, \tag{4.5.18}$$

where

$$\mathbf{P}_{Ak} = \begin{Bmatrix}\mathrm{A}_R \\ \mathrm{A}_T \\ \mathrm{A}_N\end{Bmatrix}_k, \qquad \mathbf{P}_{CS_{ik}} = \begin{Bmatrix}\mathrm{D}_R \\ \mathrm{S}_R \\ \mathrm{D}_T \\ \mathrm{S}_T \\ \mathrm{D}_N \\ \mathrm{S}_N\end{Bmatrix}_{ik}, \qquad i = 1,\, 2 \quad , \quad k = 1,\, 2,\, 3,\, \ldots,\, N. \tag{4.5.19}$$

At the same time, the parameter state transition matrix $\mathbf{\Phi}_{xp}$ has also dimension $6 \times p$, with $p = 15 \times N$, and it can be represented as the collection of N sub-matrices each of dimension 6×15. Moreover, it is populated up to the k-th 6×15 sub-matrix, for the same reasons explained in the previous Section, i.e.,

$$\boldsymbol{\Phi}_{xp}(t_k, t_0) = \begin{bmatrix} \overbrace{\begin{matrix} \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_1} & \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_2} & \cdots & \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_k} \end{matrix}}^{6\times(15\times k)} & \mathbf{0} & \cdots & \mathbf{0} \\ \begin{matrix} \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_1} & \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_2} & \cdots & \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_k} \end{matrix} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}_{6\times(15\times N)} . \tag{4.5.20}$$

In this case, similarly to what happened in Section 4.4.2, the observation-state mapping matrix $\mathbf{H}_i$ referring to the i-th observation $\mathbf{y}_i$ of the m total, is again given by

$$\mathbf{H}_i = \begin{bmatrix} \tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx}(t_i, t_0) & \tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xp}(t_i, t_0) \end{bmatrix}_{1\times n} , \tag{4.5.21}$$

and if we identify the non-null portion of $\boldsymbol{\Phi}_{xp}(t_k, t_0)$ in equation (4.5.20) as

$$\boldsymbol{\Phi}^k_{xp} = \begin{bmatrix} \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_1} & \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_2} & \cdots & \dfrac{\partial \mathbf{r}}{\partial \mathbf{P}_k} \\ \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_1} & \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_2} & \cdots & \dfrac{\partial \mathbf{v}}{\partial \mathbf{P}_k} \end{bmatrix}_{6\times(15\times k)} , \tag{4.5.22}$$

equation (4.5.21) can be rearranged to give

$$\mathbf{H}_i = \big[\overbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx}}^{1\times 6} \quad \overbrace{\underbrace{\tilde{\mathbf{H}}_i \boldsymbol{\Phi}^k_{xp}}_{1\times(15\times k)} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}}^{1\times p}\big]. \tag{4.5.23}$$

Then, the structure of $\mathbf{H}_i^T \mathbf{H}_i$ for the segmentation of empirical accelerations is

$$\mathbf{H}_i^T \mathbf{H}_i = \begin{bmatrix} \mathbf{H}_{xx}^T \mathbf{H}_{xx} & \mathbf{H}_{xx}^T \mathbf{H}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{H}_k^T \mathbf{H}_{xx} & \mathbf{H}_k^T \mathbf{H}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}_{n\times n} , \tag{4.5.24}$$

where the following definitions have been used

$$\begin{aligned} \mathbf{H}_{xx} &= \tilde{\mathbf{H}}_i \boldsymbol{\Phi}_{xx} \\ \mathbf{H}_k &= \tilde{\mathbf{H}}_i \boldsymbol{\Phi}^k_{xp}, \end{aligned}$$

and again, we note how the segmentation of parameters leads to matrices that are sparse, particularly at the beginning of propagation where only few sub-parameters have been activated. As a matter of fact, $\mathbf{H}_i^T \mathbf{H}_i$ is gradually populated and only when the last parameters are activated it becomes full-rank and ready for least squares estimation, since we recall that $\mathbf{H}^T \mathbf{H}$ must be eventually inverted. Finally, accumulation for the RHS of normal equations, can also benefit from considerations on the sparseness of $\mathbf{H}_i$ in terms of computational efficiency. In fact, when performing the product $\mathbf{H}_i^T \mathbf{y}_i$ only the non-zero components of $\mathbf{H}_i$ should be involved, thus, at time $t_k \in \Delta t_k$ we have

$$\begin{aligned} \mathbf{H}_i^T \mathbf{y}_i &= \begin{bmatrix} \mathbf{H}_{xx} & \mathbf{H}_k & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T \mathbf{y}_i, \\ &= \begin{bmatrix} \mathbf{H}_{xx} \mathbf{y}_i & \mathbf{H}_k \mathbf{y}_i & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T, \end{aligned} \tag{4.5.25}$$

with $\mathbf{y}_i$ being the i-th scalar measurement deviation from the total of l simultaneous observations at t_k.

Chapter 5

The Kalman Filter

The batch filter, or least squares method, gives an estimate of the state at epoch by processing the entire batch of measurements over the considered observation span all at once, and then iterates to convergence, making it unsuitable for real-time applications. Moreover, the fact that the single estimate at epoch must adequately fit the entire observation data, makes it sensitive to errors or approximations in the dynamical model, as well as the presence of measurement biases. An alternative approach that handles these problems can be derived by a *sequential* formulation of the batch filter, which in turns leads to the same structure of the *Kalman filter*, referring to the work of *Rudolf E. Kalman*, whose treatment of the problem received more popular acclaim. This alternative formulation processes observations as soon as they are received, making it suitable for real-time tracking and less susceptible to model errors.

Here, two main formulations of this filter will be discussed: the *Kalman filter* and the *extended Kalman filter*, where the two differ in the usage of the reference trajectory. The former keeps the same reference over the entire estimation, while the latter updates it after every observation is processed.

5.1 Derivation of Sequential Estimation

Consider two different times of observation, the first being t_j and the second t_k. At time t_j an estimate $\hat{\mathbf{x}}_j$ of the state vector deviations and its associated covariance matrix $\mathbf{P}_j$ are available, whether it is from a previous batch of data or from a priori information. For the linearized orbit model discussed in Chapter 3, these two quantities can be propagated to time t_k as:

$$\bar{\mathbf{x}}_k = \mathbf{\Phi}(t_k, t_j)\hat{\mathbf{x}}_j \tag{5.1.1}$$

$$\bar{\mathbf{P}}_k = \mathbf{\Phi}(t_k, t_j)\mathbf{P}_j\mathbf{\Phi}^T(t_k, t_j), \tag{5.1.2}$$

where $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{P}}_k$ become *a priori information* for the time of observation t_k. Now, if an additional observation is performed at t_k we have

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \boldsymbol{\varepsilon}_k, \tag{5.1.3}$$

and before going on, some clarifications about the observation-state mapping matrix $\mathbf{H}_k$ are necessary. Differently from the batch processor, where we needed to map measurement deviations at a generic time t_k to a specific epoch t_0, for the Kalman filter, measurement deviations at time t_k are usually mapped to the state deviations at the same time t_k, meaning that the multiplication of $\tilde{\mathbf{H}}_k$ with the state transition matrices $\mathbf{\Phi}_{xx}(t, t_k)$ and $\mathbf{\Phi}_{xp}(t, t_k)$ in equations (4.1.5) and (4.1.6), is not needed. Consequently, when referring to the Kalman filter, the matrix $\mathbf{H}_k$ is intended to be:

$$\mathbf{H}_k = \begin{bmatrix}\tilde{\mathbf{H}}_k & \mathbf{0} & \tilde{\mathbf{K}}_k\end{bmatrix} \tag{5.1.4}$$

where the definitions from equation (3.1.9) are used. Note the presence of $\mathbf{0}$ in the position corresponding to the parameter state transition matrix. This happens because now we are no more mapping observation deviations at a generic time t to state deviations at the reference epoch t_0, now the state transition matrix is reinitialized at every observation instant, and the initial condition for the PSTM is the null matrix.

From the results obtained in Chapter 4, the best estimate of the state deviations at time t_k is given by

$$\hat{\mathbf{x}}_k = \left(\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k + \bar{\mathbf{P}}_k^{-1}\right)^{-1}\left(\mathbf{H}_k^T\mathbf{W}_k\mathbf{y}_k + \bar{\mathbf{P}}_k^{-1}\bar{\mathbf{x}}_k\right), \tag{5.1.5}$$

where again, the measurement error vector $\boldsymbol{\varepsilon}_k$ is assumed to satisfy $E\left[\boldsymbol{\varepsilon}_k\right] = 0$, and $E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right] = \mathbf{R}_k\delta_{kj}$ with $\mathbf{W}_k = \mathbf{R}_k^{-1}$. However, equation (5.1.5) directly comes from the least squares method for the batch processor, and the major computational problem is related to the inversion of the $n \times n$ information matrix

$$\boldsymbol{\Lambda}_k = \mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k + \bar{\mathbf{P}}^{-1}, \tag{5.1.6}$$

that we recall being related to the *solution covariance matrix* $\mathbf{P}_k$ at time t_k as

$$\begin{aligned}\mathbf{P}_k &= \boldsymbol{\Lambda}_k^{-1}\\ &= \left(\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k + \bar{\mathbf{P}}_k^{-1}\right)^{-1}.\end{aligned} \tag{5.1.7}$$

and from equation (5.1.7) it follows that

$$\mathbf{P}_k^{-1} = \mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k + \bar{\mathbf{P}}_k^{-1}. \tag{5.1.8}$$

Now, premultiplying equation (5.1.8) by $\mathbf{P}_k$ and then postmultiplying it by $\bar{\mathbf{P}}_k$ gives:

$$\bar{\mathbf{P}}_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k\bar{\mathbf{P}}_k + \mathbf{P}_k, \tag{5.1.9}$$

but also

$$\mathbf{P}_k = \bar{\mathbf{P}}_k - \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k\bar{\mathbf{P}}_k. \tag{5.1.10}$$

Now, postmultiplying equation (5.1.9) by the quantity $\mathbf{H}_k^T\mathbf{W}_k$:

$$\begin{aligned}\bar{\mathbf{P}}_k\mathbf{H}_k^T\mathbf{W}_k &= \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T\mathbf{W}_k + \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\\ &= \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\left[\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T\mathbf{W}_k + I\right]\\ &= \mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k\left[\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{W}_k^{-1}\right]\mathbf{W}_k,\end{aligned} \tag{5.1.11}$$

and solving for $\mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k$ gives

$$\mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k = \bar{\mathbf{P}}_k\mathbf{H}_k^T\left[\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{W}_k^{-1}\right]^{-1}, \tag{5.1.12}$$

and substituting the expression just obtained into equation (5.1.10) to eliminate the term $\mathbf{P}_k\mathbf{H}_k^T\mathbf{W}_k$, results into

$$\mathbf{P}_k = \bar{\mathbf{P}}_k - \bar{\mathbf{P}}_k\mathbf{H}_k^T\left[\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{W}_k^{-1}\right]^{-1}\mathbf{H}_k\bar{\mathbf{P}}_k. \tag{5.1.13}$$

Equation (5.1.13), relates the a priori covariance $\bar{\mathbf{P}}_k$ to the a posteriori covariance $\mathbf{P}_k$, similarly to what equation (5.1.7) does for the batch filter. In fact, equations (5.1.13) and (5.1.7) are two alternative way of computing the *a posteriori* solution covariance matrix $\mathbf{P}_k$. However, the matrix to be inverted in equation (5.1.13) is of dimension $p \times p$ instead of $n \times n$ as in equation (5.1.7), meaning that if at time t_k there is only one observation, or if all the observations are processed as scalar quantities, matrix inversion becomes scalar division, which is a remarkable computational convenience.

In equation (5.1.13) the following definition is used

$$\mathbf{K}_k \equiv \bar{\mathbf{P}}_k \mathbf{H}_k^T \left[\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k + \mathbf{R}_k\right]^{-1}, \tag{5.1.14}$$

and the weighting matrix $\mathbf{K}_k$ is referred to as the *Kalman gain.* Then, equation (5.1.13), can be expressed in a much more compact form as

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \bar{\mathbf{P}}_k. \tag{5.1.15}$$

Now, considering that equation (5.1.5) can be written as

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathbf{P}_k \left(\mathbf{H}_k^T \mathbf{W}_k \mathbf{y}_k + \bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}}_k\right) \\ &= \mathbf{P}_k \mathbf{H}_k^T \mathbf{W}_k \mathbf{y}_k + \mathbf{P}_k \bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}},\end{aligned} \tag{5.1.16}$$

and recognizing from equations (5.1.12) and (5.1.14) that $\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{W}_k$, (5.1.16) becomes

$$\hat{\mathbf{x}}_k = \mathbf{K}_k \mathbf{y}_k + \mathbf{P}_k \bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}}, \tag{5.1.17}$$

then, using equation (5.1.15) into (5.1.17) leads to

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathbf{K}_k \mathbf{y}_k + [\mathbf{I} + \mathbf{K}_k \mathbf{H}_k] \bar{\mathbf{P}}_k \bar{\mathbf{P}}_k^{-1} \bar{\mathbf{x}}_k \\ &= \mathbf{K}_k \mathbf{y}_k + \bar{\mathbf{x}}_k + \mathbf{K}_k \mathbf{H}_k \bar{\mathbf{x}}_k,\end{aligned} \tag{5.1.18}$$

and finally, collecting terms

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k [\mathbf{y}_k - \mathbf{H}_k \bar{\mathbf{x}}_k]. \tag{5.1.19}$$

Equation (5.1.19) along with (5.1.14), (5.1.1), (5.1.2) and (5.1.15), can be used in a sequential algorithm to obtain a best estimate at time t_k, having a priori information at time t_{k-1} incorporated with observations $\mathbf{y}_k$.

5.1.1 The Kalman Filter Algorithm

In this Section a brief summary of the Kalman filter algorithm and its flow chart are outlined.
The problem statement in this case goes as follows: given the estimates for the state deviations $\hat{\mathbf{x}}_{k-1}$, solution covariance matrix $\mathbf{P}_{k-1}$, the observation $\mathbf{Y}_k$ at time t_k and associated measurement error covariance matrix $\mathbf{R}_k$, find the best estimate of the state at time t_k.

First, the equations of motions are integrated along with the state transition matrix equations from t_{k-1} to t_k, where $\mathbf{X}^*$ still indicates the reference trajectory about which the linearization is made,

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}^*, \mathbf{P}^*, t), \qquad \mathbf{X}^*(t_{k-1}) = \mathbf{X}^*_{k-1} \tag{5.1.20}$$

$$\boldsymbol{\Phi}_{xx}(t, t_{k-1}) = \mathbf{A}(t) \boldsymbol{\Phi}_{xx}(t, t_{k-1}), \qquad \boldsymbol{\Phi}_{xx}(t_{k-1}, t_{k-1}) = I \tag{5.1.21}$$

$$\boldsymbol{\Phi}_{xp}(t, t_{k-1}) = \mathbf{A}(t) \boldsymbol{\Phi}_{xp}(t, t_{k-1}) + \mathbf{B}(t), \qquad \boldsymbol{\Phi}_{xp}(t_{k-1}, t_{k-1}) = \mathbf{0}. \tag{5.1.22}$$

Then, the estimates at time t_{k-1} are propagated to time t_k through the state transition matrix, and become a priori values

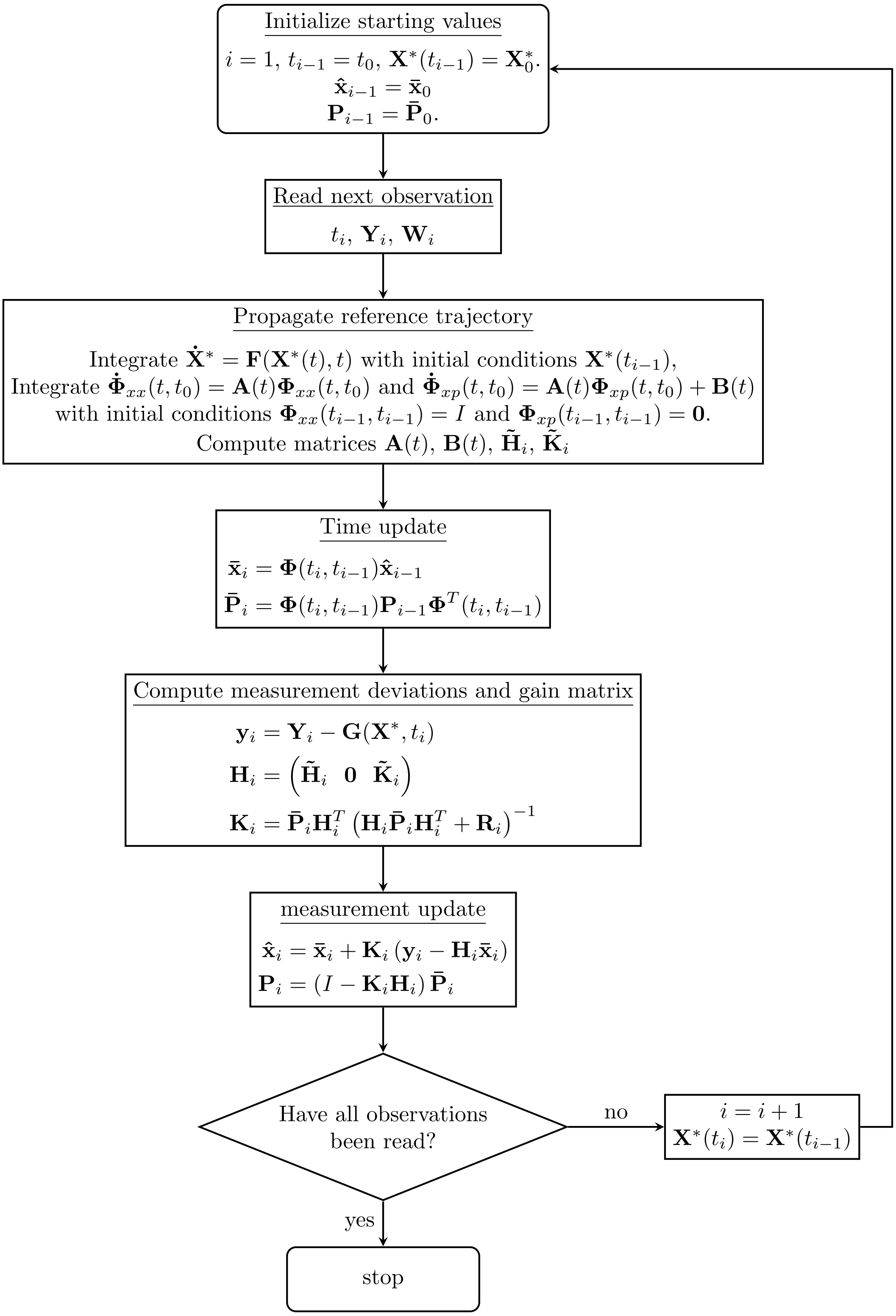
Initialize starting values
$i = 1$, $t_{i-1} = t_0$, $\mathbf{X}^*(t_{i-1}) = \mathbf{X}^*_0$.
$\hat{\mathbf{x}}_{i-1} = \bar{\mathbf{x}}_0$
$\mathbf{P}_{i-1} = \bar{\mathbf{P}}_0$.
Read next observation
t_i, $\mathbf{Y}_i$, $\mathbf{W}_i$
Propagate reference trajectory
Integrate $\dot{\mathbf{X}}^* = \mathbf{F}(\mathbf{X}^*(t), t)$ with initial conditions $\mathbf{X}^*(t_{i-1})$,
Integrate $\dot{\mathbf{\Phi}}_{xx}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xx}(t, t_0)$ and $\dot{\mathbf{\Phi}}_{xp}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t, t_0) + \mathbf{B}(t)$
with initial conditions $\mathbf{\Phi}_{xx}(t_{i-1}, t_{i-1}) = I$ and $\mathbf{\Phi}_{xp}(t_{i-1}, t_{i-1}) = \mathbf{0}$.
Compute matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\tilde{\mathbf{H}}_i$, $\tilde{\mathbf{K}}_i$
Time update
$\bar{\mathbf{x}}_i = \mathbf{\Phi}(t_i, t_{i-1})\hat{\mathbf{x}}_{i-1}$
$\bar{\mathbf{P}}_i = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}_{i-1}\mathbf{\Phi}^T(t_i, t_{i-1})$
Compute measurement deviations and gain matrix
$\mathbf{y}_i = \mathbf{Y}_i - \mathbf{G}(\mathbf{X}^*, t_i)$
$\mathbf{H}_i = \begin{pmatrix} \tilde{\mathbf{H}}_i & \mathbf{0} & \tilde{\mathbf{K}}_i \end{pmatrix}$
$\mathbf{K}_i = \bar{\mathbf{P}}_i\mathbf{H}_i^T \left(\mathbf{H}_i\bar{\mathbf{P}}_i\mathbf{H}_i^T + \mathbf{R}_i\right)^{-1}$
measurement update
$\hat{\mathbf{x}}_i = \bar{\mathbf{x}}_i + \mathbf{K}_i\left(\mathbf{y}_i - \mathbf{H}_i\bar{\mathbf{x}}_i\right)$
$\mathbf{P}_i = \left(I - \mathbf{K}_i\mathbf{H}_i\right)\bar{\mathbf{P}}_i$
Have all observations been read?
no
$i = i + 1$
$\mathbf{X}^*(t_i) = \mathbf{X}^*(t_{i-1})$
yes
stop

Figure 5.1.1: Flow chart for the Kalman filter algorithm.

$$\begin{aligned}\bar{\mathbf{x}}_k &= \mathbf{\Phi}(t_k, t_{k-1})\hat{\mathbf{x}}_{k-1}\\ \bar{\mathbf{P}}_k &= \mathbf{\Phi}(t_k, t_{k-1})\mathbf{P}_{k-1}\mathbf{\Phi}(t_k, t_{k-1}).\end{aligned} \tag{5.1.23}$$

Next, the measurement deviation vector and observation-state mapping matrices are computed

$$\mathbf{y}_k = \mathbf{Y}_k - \mathbf{G}(\mathbf{X}^*, t_k), \qquad \tilde{\mathbf{H}}_k = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right]^*, \qquad \tilde{\mathbf{K}}_k = \left[\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]^*, \tag{5.1.24}$$

and the Kalman gain, state deviations estimate and solution covariance matrix at time t_k can be calculated

$$\begin{aligned}&\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}_k^T \left[\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_K^T + \mathbf{R}_k\right]^{-1}\\ &\hat{\mathbf{x}}_k + \mathbf{K}_k \left[\mathbf{y}_k - \mathbf{H}_k \bar{\mathbf{x}}_k\right]\\ &\mathbf{P}_k = \left[I - \mathbf{K_k H_k}\right] \bar{\mathbf{P}}_k.\end{aligned} \tag{5.1.25}$$

Finally, the process is repeated by replacing k with $k+1$, so that the estimate becomes the a priori value for the next observation that is being read, and the estimate of the state at t_k is give by

$$\hat{\mathbf{X}}_k = \mathbf{X}_k^* + \hat{\mathbf{x}}_k. \tag{5.1.26}$$

The phase described by equations (5.1.23) is called the *time update*, where the state estimate and its covariance matrix are updated to the next observation time, while equations (5.1.25) are known as the *measurement update*. One major characteristic that distinguish this filter from the batch processor lies in the way the state transition matrix is handled. Unlike the batch filter, here, the state transition matrix is the mean by which the estimate and covariance matrix are propagated one step forward to the next observation, thus, after every *measurement update*, the state transition matrix is reinitialized. This translates into the fact that if every measurement is processed as a scalar, regardless of the size of $\mathbf{y}_k$, the state transition matrix is set to $\mathbf{\Phi}(t_k, t_k) = I$ after reading the first measurement at each time of observation.

Another main difference from the batch filter is that the Kalman filter is more sensitive to computer round-off errors and this sensitivity is found especially in equation (5.1.15), where the solution covariance matrix $\mathbf{P}_k$ is computed from the a priori covariance matrix $\bar{\mathbf{P}}_k$. In finite digit arithmetic this may lead to the loss of symmetry and positive definiteness of $\mathbf{P}_k$, especially when a large a priori covariance meets very accurate measurements, that is small values for $\mathbf{R}$ and large values for $\bar{\mathbf{P}}$. In other words the term $\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R}$ in the Kalman filter, eq. (5.1.14), will tend to ignore the contribution of new data, while the batch filter will tend to ignore $\bar{\mathbf{P}}^{-1}$ and give more relevance to the strength of new observations in $\mathbf{H}^T\mathbf{W}\mathbf{H} + \bar{\mathbf{P}}^{-1}$. A common alternative to equation (5.1.15), that deals with this problem is given by Bucy and Joseph,

$$\mathbf{P}_k = \left(I - \mathbf{K}_k \mathbf{H}_k\right) \bar{\mathbf{P}}_k \left(I - \mathbf{K}_k \mathbf{H}_k\right)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T, \tag{5.1.27}$$

that clearly always yields a symmetric matrix, and ensures the positive definiteness for well observed problems.

Moreover, an unfavorable characteristic of the Kalman filter described so far, is that the state estimation error covariance matrix may approach zero as the number of observations processed increases. By analyzing equation (5.1.15), we can see how the quantity $\bar{\mathbf{P}}$ is reduced by $\mathbf{K}\mathbf{H}\bar{\mathbf{P}}$ at every measurement update, and this decrease is governed by the density, information content and accuracy of the observations. The consequence of the matrix $\mathbf{P}_k$ approaching zero is that the kalman gain will also approach zero, making the filter insensitive to new observations and eventually the algorithm will diverge due to the approximations in the linearization process, or the incompleteness of the mathematical model. In order to overcome this problem the addition of *process noise* is often used to compensate for approximations, and will be discussed in Section 5.3. A flow chart for the conventional Kalman filter algorithm is presented in figure 5.1.1.

5.2 Extended Kalman Filter

One shortcoming that is common both to the *batch* and *conventional Kalman* filters described previously, arises from the fact that divergence can be encountered if the reference trajectory used for linearization isn't sufficiently near the true trajectory, making the assumption of small deviations used in Chapter 3 no more valid. This, moreover, is the reason why the batch filter needs to be iterated and the Kalman filter may take some time before converging to a satisfactory solution. The sequential filter is well suited for a formulation that helps overcoming this neglect of higher order terms, and this modified version is referred to as the *Extended Kalman Filter* (EKF), which nowadays is much more used over the conventional formulation.

The peculiarity lies in the fact that the reference trajectory is not held constant for the entire estimation process, but it is updated after each observation has been processed, so that the effects of the best estimates of the state deviations are reflected on the estimate of the true trajectory, that is

$$\mathbf{X}^*_{k_{new}} = \hat{\mathbf{X}}_k = \mathbf{X}^*_k + \hat{\mathbf{x}}_k. \tag{5.2.1}$$

Notice that using $\hat{\mathbf{X}}_k$ for the reference trajectory implies that $\hat{\mathbf{x}}_k = 0$, since the previous trajectory has been corrected by the very estimation $\hat{\mathbf{x}}_k$ at the same time t_k, and this results in having $\bar{\mathbf{x}}_{k+1} = 0$ which is clear from equation (5.1.23). All of this leads to the estimate $\hat{\mathbf{x}}_{k+1}$ being simply:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{K}_k \mathbf{y}_{k+1}, \tag{5.2.2}$$

where the Kalman gain and measurement deviations are computed through the updated reference trajectory. The rest of the algorithm remains similar to the conventional Kalman filter discussed in Section 5.1, but convergence is more rapid thanks to the reduction in linearization errors.

The only major drawback of the EKF algorithm is that the differential equation for the orbit propagation must be reinitialized after each observation time has been processed, and the considerations about *process noise* and convergence, previously made for the conventional Kalman filter, still hold true. A flow chart for the Extended Kalman Filter is shown in figure 5.2.1.

5.3 Process Noise for Kalman Filter

As previously stated, the Kalman filter (also known as sequential filter) processes new observations as soon as they are available, making it a real-time estimator rather than an offline estimator like the batch processor. One major problem that can be encountered when applying a Kalman filter to a non-linear physical system is due to the incompleteness or approximation of the mathematical model that is being used to linearize the equations, leading to filter saturation and divergence.

In other words, in the absence of process noise, the state estimation error covariance matrix $\mathbf{P}_k$ becomes smaller and smaller as new data are processed in the *measurement update* phase:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\,\bar{\mathbf{P}}_k. \tag{5.3.1}$$

Equation (5.3.1) clearly shows how the covariance matrix is reduced by the quantity $\mathbf{K}_k\mathbf{H}_k\bar{\mathbf{P}}_k$ at each measurement update depending on the density, information content and accuracy of the observations, which in turn makes the Kalman gain approach zero:

$$\mathbf{K}_k = \bar{\mathbf{P}}_k\mathbf{H}_k^T\left[\mathbf{H}_k\bar{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \to 0, \tag{5.3.2}$$

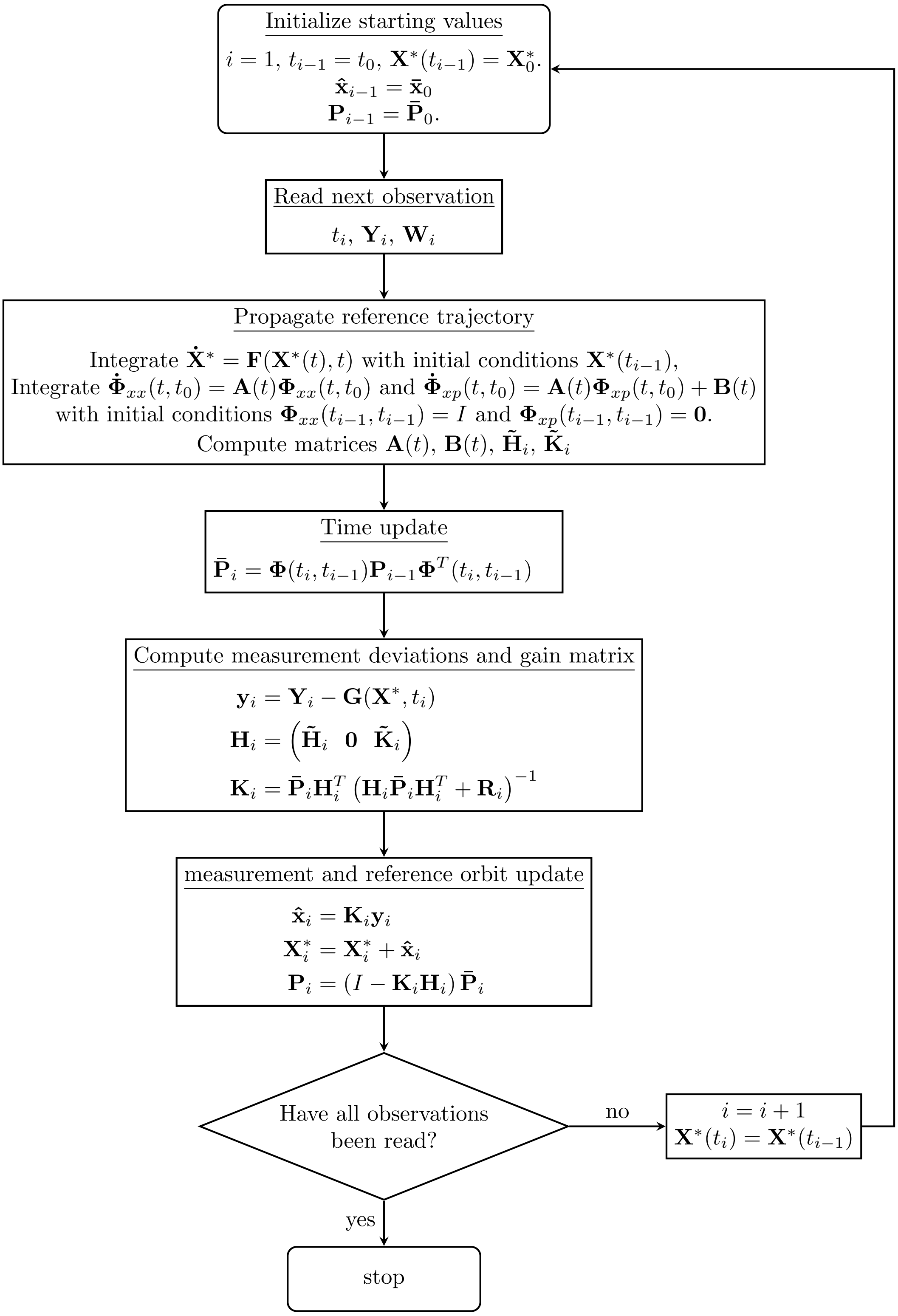
Initialize starting values
$i = 1$, $t_{i-1} = t_0$, $\mathbf{X}^*(t_{i-1}) = \mathbf{X}_0^*$.
$\hat{\mathbf{x}}_{i-1} = \bar{\mathbf{x}}_0$
$\mathbf{P}_{i-1} = \bar{\mathbf{P}}_0$.
Read next observation
t_i, $\mathbf{Y}_i$, $\mathbf{W}_i$
Propagate reference trajectory
Integrate $\dot{\mathbf{X}}^* = \mathbf{F}(\mathbf{X}^*(t), t)$ with initial conditions $\mathbf{X}^*(t_{i-1})$,
Integrate $\dot{\mathbf{\Phi}}_{xx}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xx}(t, t_0)$ and $\dot{\mathbf{\Phi}}_{xp}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}_{xp}(t, t_0) + \mathbf{B}(t)$
with initial conditions $\mathbf{\Phi}_{xx}(t_{i-1}, t_{i-1}) = I$ and $\mathbf{\Phi}_{xp}(t_{i-1}, t_{i-1}) = \mathbf{0}$.
Compute matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\tilde{\mathbf{H}}_i$, $\tilde{\mathbf{K}}_i$
Time update
$\bar{\mathbf{P}}_i = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}_{i-1}\mathbf{\Phi}^T(t_i, t_{i-1})$
Compute measurement deviations and gain matrix
$\mathbf{y}_i = \mathbf{Y}_i - \mathbf{G}(\mathbf{X}^*, t_i)$
$\mathbf{H}_i = \begin{pmatrix} \tilde{\mathbf{H}}_i & \mathbf{0} & \tilde{\mathbf{K}}_i \end{pmatrix}$
$\mathbf{K}_i = \bar{\mathbf{P}}_i\mathbf{H}_i^T \left(\mathbf{H}_i\bar{\mathbf{P}}_i\mathbf{H}_i^T + \mathbf{R}_i\right)^{-1}$
measurement and reference orbit update
$\hat{\mathbf{x}}_i = \mathbf{K}_i\mathbf{y}_i$
$\mathbf{X}_i^* = \mathbf{X}_i^* + \hat{\mathbf{x}}_i$
$\mathbf{P}_i = (I - \mathbf{K}_i\mathbf{H}_i)\bar{\mathbf{P}}_i$
Have all observations been read?
no
$i = i + 1$
$\mathbf{X}^*(t_i) = \mathbf{X}^*(t_{i-1})$
yes
stop

Figure 5.2.1: Flow chart for the Extended Kalman Filter (EKF) algorithm.

and it is in this situation that the filter becomes insensitive to new observations. This can be further seen by looking at the measurement update equation for the state estimation, where as $\mathbf{K}_k$ becomes smaller the state deviation estimate $\hat{\mathbf{x}}$ remains the same as the a priori state deviation $\bar{\mathbf{x}}$:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k \left[\mathbf{y}_k - \mathbf{H}_k \bar{\mathbf{x}}_k\right] \simeq \bar{\mathbf{x}}, \tag{5.3.3}$$

here the filter behaves like it reached a steady-state solution by ignoring new information and simply propagating the reference trajectory from one time to another following its equations of motion. Eventually, the estimated trajectory diverges from the true solution due to the aforementioned non-linearities, model approximations or incompleteness.

To overcome this problem, after recognizing that the linearized equations used for the propagation of the state estimate will always be in error to some extent, *process noise* is added to account for the effects of unmodeled or inaccurately modeled accelerations acting on the system, and it is in this context that various techniques have been developed. These techniques differentiate for their complexity and adaptability, but in every case their main purpose is to avoid filter saturation by ensuring that the solution covariance matrix $\mathbf{P}_k$ will never approach zero at least on its diagonal. This is accomplished with the determination of a proper *process noise covariance matrix* $\mathbf{Q}_k$ that is added to the second of time update equations in (5.1.23), that is

$$\bar{\mathbf{P}}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k) P_k \mathbf{\Phi}(t_{k+1}, t_k) + \mathbf{Q}_k, \tag{5.3.4}$$

where the matrix $\mathbf{Q}_k$ is properly determined according to the particular mathematical model that is chosen to represent process noise. This addition allows to compensate the reduction in $\mathbf{P}_k$ that arises from the information content of new observations during the measurement update described before, and figure 5.3.1, schematically shows the trend of the trace of $\mathbf{P}$ over time.

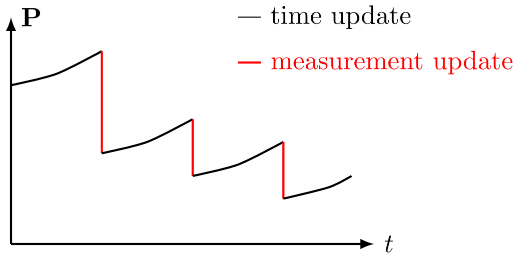

Figure 5.3.1: Schematic representation of the reduction of $\mathbf{P}$ values that occurs at every measurement update in the absence of process noise.

5.3.1 State Noise Compensation

A first and simple possibility is to assume that these accelerations can be described by a Gaussian white noise, and we refer to this method as the *State Noise Compensation* (SNC) which has been investigated by Myers (1974) [21] and is also widely described by Tapley (2004) [24]. Here, the dynamics of an Earth-orbiting satellite are assumed to be governed by the following set of equations:

$$\dot{\mathbf{r}} = \mathbf{v} \tag{5.3.5}$$

$$\dot{\mathbf{v}} = \mathbf{a}_m(\mathbf{r}, \mathbf{v}, t) + \mathbf{w}(t), \tag{5.3.6}$$

where $\mathbf{a}_m$ is a three dimensional functional representing modeled accelerations, while $\mathbf{w}(t)$ is the three-dimensional vector of white Gaussian noise representing unknown or poorly modeled accelerations that satisfies the following properties

$$E[\mathbf{w}(t)] = 0$$
$$E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{Q}'\delta(t-\tau),$$

meaning that it is unbiased (zero mean) and its covariance matrix $\mathbf{Q}'$ is diagonal and has no correlation in time.

In order to determine the process noise covariance matrix $\mathbf{Q}$ to be used in equation (5.3.4) we need to compute the integrated effect of $\mathbf{w}(t)$ in the velocity and position components, to do so, let us rewrite the equations of motion in state vector form considering the propagation from t_{k-1}, namely

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}, \mathbf{w}, t), \qquad \mathbf{X}(t_{k-1}) = \mathbf{X}_{k-1}, \tag{5.3.7}$$

where $\mathbf{X}$ contains all the n parameters to be estimated and note how these equations contain a stochastic term. In general, we can expect an integral form of (5.3.7) to be like the following

$$\mathbf{X}(t) = \mathbf{\Pi}(\mathbf{X}_{k-1}, t_{k-1}, t) + \boldsymbol{\eta}_{k-1}, \tag{5.3.8}$$

where $\mathbf{\Pi}$ is an n-dimensional deterministic functional, while $\boldsymbol{\eta}_{k-1}$ represents the integrated effect of the white process noise $\mathbf{w}(t)$ from time t_{k-1} to t or, in other words, it is the random portion of the integral solution to the equations of motion. Nonetheless, in the context of sequential filters for orbit determination a linearized solution is adopted rather than an actual integral formulation, and we saw how linearization occurs in terms of state deviations from the true trajectory, that is, we can linearize (5.3.8) about a reference orbit $\mathbf{X}^*$:

$$\mathbf{x}(t) = \left[\frac{\partial \mathbf{\Pi}}{\partial \mathbf{X}}\right]^* \mathbf{x}(t_{k-1}) + \boldsymbol{\eta}_{k-1} \tag{5.3.9}$$

where again, the superscript $*$ indicates evaluation along the reference trajectory. We can then recognize that the partial derivative of $\mathbf{\Pi}$ with respect to $\mathbf{X}$ is a state transition matrix $\mathbf{\Phi}(t, t_{k-1})$ by definition, so we can write the state deviation at time t as

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_{k-1})\mathbf{x}_{k-1} + \boldsymbol{\eta}_{k-1}, \tag{5.3.10}$$

and its associated covariance becomes

$$\begin{aligned} \bar{\mathbf{P}}(t) &= E\left\{\left[\mathbf{\Phi}(t, t_{k-1}) + \boldsymbol{\eta}_{k-1}\right]\left[\mathbf{\Phi}(t, t_{k-1}) + \boldsymbol{\eta}_{k-1}\right]^T\right\} \\ &= \mathbf{\Phi}(t, t_{k-1})\mathbf{P}_{k-1}\mathbf{\Phi}^T(t, t_{k-1}) + E[\boldsymbol{\eta}_{k-1}\boldsymbol{\eta}_{k-1}^T], \end{aligned} \tag{5.3.11}$$

where we have taken advantage of the fact that $\mathbf{x}_{k-1}$ and $\boldsymbol{\eta}_{k-1}$ are independent.

At this point we can note how arriving at (5.3.11) we actually found the expression of the covariance time update in the Kalman filter, and by comparing it with (5.3.4) we conclude that the process noise covariance matrix can be computed as

$$\mathbf{Q}_{k-1} = E[\boldsymbol{\eta}_{k-1}\boldsymbol{\eta}_{k-1}^T]. \tag{5.3.12}$$

Note that the result we have arrived at is general meaning that it is not valid for SNC only, and it tells that the process noise covariance can be obtained from $\boldsymbol{\eta}$, the random portion of the integrated equations. Another general result that is useful in determining a proper expression for $\boldsymbol{\eta}$ is shown by Ingram (1970) [8], where the equations of motion of a dynamical system of the form

$$\dot{\mathbf{r}} = \mathbf{v} \tag{5.3.13}$$
$$\dot{\mathbf{v}} = \mathbf{a}(\mathbf{r}, \mathbf{v}, t), \tag{5.3.14}$$

can be integrated by parts to obtain

$$\mathbf{r}(t) = \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t + \int_{t_{k-1}}^{t} \mathbf{a}(\tau)(t-\tau)\mathrm{d}\tau \tag{5.3.15}$$

$$\mathbf{v}(t) = \mathbf{v}_{k-1} + \int_{t_{k-1}}^{t} \mathbf{a}(\tau)\mathrm{d}\tau. \tag{5.3.16}$$

Now, for the case of SNC the expression for the acceleration we simply apply the results of (5.3.15) and (5.3.16) to equations (5.3.5) and (5.3.6) which gives

$$\mathbf{r}(t) = \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t + \int_{t_{k-1}}^{t} \mathbf{a}_m(\tau)(t-\tau)\mathrm{d}\tau + \int_{t_{k-1}}^{t} \mathbf{w}(\tau)(t-\tau)\mathrm{d}\tau \tag{5.3.17}$$

$$\mathbf{v}(t) = \mathbf{v}_{k-1} + \int_{t_{k-1}}^{t} \mathbf{a}_m(\tau)\mathrm{d}\tau + \int_{t_{k-1}}^{t} \mathbf{w}(\tau)\mathrm{d}\tau, \tag{5.3.18}$$

then, recognizing that the rightmost integrals in the above equations are indeed the components of $\boldsymbol{\eta}$ needed to compute $\mathbf{Q}$, i.e.,

$$\boldsymbol{\eta}_{k-1} = \begin{bmatrix} \int_{t_{k-1}}^{t} \mathbf{w}(\tau)(t-\tau)\mathrm{d}\tau \\ \int_{t_{k-1}}^{t} \mathbf{w}(\tau)\mathrm{d}\tau \end{bmatrix}, \tag{5.3.19}$$

and if we partition $\mathbf{Q}_{k-1}$ into four 3×3 submatrices as

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}, \tag{5.3.20}$$

equation (5.3.12) leads to the following results:

$$\mathbf{Q}_{11} = E\left[\int_{t_{k-1}}^{t}\int_{t_{k-1}}^{\tau} \mathbf{w}(\tau)\mathbf{w}^T(\nu)(t-\tau)(t-\nu)\mathrm{d}\nu\mathrm{d}\tau\right] \tag{5.3.21}$$

$$\mathbf{Q}_{22} = E\left[\int_{t_{k-1}}^{t}\int_{t_{k-1}}^{\tau} \mathbf{w}(\tau)\mathbf{w}^T(\nu)\mathrm{d}\nu\mathrm{d}\tau\right] \tag{5.3.22}$$

$$\mathbf{Q}_{12} = \mathbf{Q}_{21} = E\left[\int_{t_{k-1}}^{t}\int_{t_{k-1}}^{\tau} \mathbf{w}(\tau)\mathbf{w}^T(\nu)(t-\tau)\mathrm{d}\nu\mathrm{d}\tau\right]. \tag{5.3.23}$$

We now recall that $\mathbf{w}(t)$ has been defined as a stationary white noise process, so the expectation operator is brought inside the integrals, and given that this process is also uncorrelated in time, that is $E[\mathbf{w}(\tau)\mathbf{w}^T(\nu)] = \mathbf{Q}'\delta(\tau-\nu)$, we arrive at the following final result for the process noise covariance matrix for the SNC algorithm

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \frac{1}{3}\mathbf{Q}'\Delta t^3 & \frac{1}{2}\mathbf{Q}'\Delta t^2 \\ \frac{1}{2}\mathbf{Q}'\Delta t^2 & \mathbf{Q}'\Delta t \end{bmatrix}, \tag{5.3.24}$$

where Δt is the time interval between two subsequent observations.

The SNC algorithm described here is a simple and straightforward method to avoid filter saturation and divergence in the presence of unknown or unmodeled accelerations. By properly tuning the white noise process, through the choice of its variance $\mathbf{Q}'$, it is possible to set a lower bound for the solution covariance matrix $\mathbf{P}$ and obtain good estimates. Myers and Tapley [22] show an application where the SNC algorithm is applied with a “controlled environment”. Here, measurements are generated through the propagation of a true orbit, with a fairly refined force model, and then the estimation is performed with a reduced model containing the Newtonian and J_2 terms only. This is done in order to simulate the effects of unknown accelerations and they emphasize how in the case of no process noise ($\mathbf{Q}' = 0$) estimated results rapidly diverge to large values, while the solution covariance approaches zero. On the other hand, when process noise is included, the solution is stable and bounded with average root sum square position errors of a few tens of meters, showing that SNC is able to counteract the hundred-meters deviations that occur during propagation periods of absent visibility.

Despite this, the SNC method is seldom used nowadays, particularly when orbit determination requirements are more stringent. Its main disadvantage lies in its inability to capture unknown or unpredictable dynamics that exhibit some degree of correlation over time. Additionally, being a purely random process, its parameters cannot be estimated during filtering, meaning that it lacks any sort of adaptive capabilities. For this reasons different process noise structures in the form of *colored noise* are frequently adopted and the simplest of these, the first-order Gauss-Markov process, is presented in the Section below.

5.3.2 Dynamic Model Compensation - Estimating Unmodeled Accelerations

Dynamical Model Compensations algorithms (DMC), also known as *structural* methods, differentiate from SNC for their ability to capture and estimate time-correlated unmodeled accelerations during filtering. This is accomplished by assuming that unknown dynamics can be described by a certain structure modeled as a *colored noise process*, and through the estimation of its characterizing parameters, adaptive capabilities are enabled allowing not only to avoid filter divergence but also to quantitatively retrieve the collective effects of unknown accelerations or unpredictable events, like spacecraft venting and unbalanced attitude control torques. Many different structures have been studied, but here, we will focus on one of the most used in the orbit determination problem, the *first-order Gauss-Markov process.* Investigations by Ingram and Tapley [9] demonstrated its successful performance for both Earth and Lunar orbiters.

The first-order Gauss-Markov process

The first-order Gauss-Markov process is so called because it obeys a Gaussian probability density function and displays the Markov property. This latter aspect tells that the conditional probability of its future states depends only on the present state X, i.e.,

$$pr\left(X_{k+1}|X_k, X_{k-1}, \dots, X_{k-n}\right) = pr\left(X_{k+1}|X_k\right), \tag{5.3.25}$$

which also underlines its *first-order* characteristic, whereas for a second order process the conditional probability of future states would depend not only on the present but also on the previous state, and this pattern repeats for higher orders. If we consider a one-dimensional system, the differential equation governing the first-order Gauss-Markov process, also known as Langevin equation, is

$$\dot{\varepsilon}(t) = -\beta\varepsilon(t) + u(t), \tag{5.3.26}$$

where $u(t)$ is a white Gaussian noise with

$$\begin{aligned} E[u(t)] &= 0 \\ E[u(t)u(\tau)] &= \sigma^2\delta(t-\tau), \end{aligned}$$

and β is related to the correlation time constant τ as $\beta = \tau^{-1}$. Here we can clearly see how the dynamics of this process are regulated by a deterministic component and a purely random part, which indeed makes the process correlated in time as we demonstrate below.

The solution of equation (5.3.26) can be found using the method of variation of parameters [24], from which we get

$$\varepsilon(t) = \varepsilon(t_0)e^{-\beta(t-t_0)} + \int_{t_0}^{t} e^{-\beta(t-\tau)}u(\tau)\,d\tau, \tag{5.3.27}$$

where again, the right hand side is composed of a deterministic integrated part plus a stochastic integral. The salient feature of the Gauss-Markov process can be highlighted by computing its autocorrelation function:

$$\begin{aligned} E[\varepsilon(t_i)\varepsilon(t_j)] &= E\left[\varepsilon(t_i)\left(\varepsilon(t_i)e^{-\beta(tj-t_i)} + \int_{t_i}^{t_j} e^{-\beta(t_j-\tau)}u(\tau)\,d\tau\right)\right] \\ &= E[\varepsilon(t_i)\varepsilon(t_i)]e^{-\beta(t_j-t_i)} + E\left[\left(\int_{t_i}^{t_j} e^{-\beta(t_j-\tau)}u(\tau)\,d\tau\right)\varepsilon(t_i)\right] \\ &= E[\varepsilon(t_i)\varepsilon(t_i)]e^{-\beta(t_j-t_i)}, \end{aligned} \tag{5.3.28}$$

where the expectation containing the stochastic integral is zero since it represents independent increments, that is

$$E\left[u(t)\varepsilon(t)\right] = 0,$$

because the noise $u(t)$ has no instantaneous effect on the process $\varepsilon(t)$ for the same time t. The remaining term contains the autocorrelation function of $\varepsilon(t)$ at t_i that can be computed as:

$$\begin{aligned} E[\varepsilon(t_i)\varepsilon(t_i)] &= \Psi(t_i, t_i) \\ &= E\left[\varepsilon(t_0)^2 e^{-2\beta(t_i-t_0)} + 2\varepsilon(t_0)\int_{t_0}^{t_i} e^{-\beta(t_i-\tau)}u(\tau)\,d\tau + \int_{t_0}^{t_i} e^{-2\beta(t_i-\tau)}u(\tau)^2\,d\tau\right] \\ &= \varepsilon(t_0)^2 e^{-2\beta(t_i-t_0)} + \sigma^2 \int_{t_0}^{t_i} e^{-2\beta(t_i-\tau)\,d\tau} \\ &= \varepsilon(t_0)^2 e^{-2\beta(t_i-t_0)} + \frac{\sigma^2}{2\beta}\left[1 - e^{-2\beta(t_i-t_0)}\right]. \end{aligned} \tag{5.3.29}$$

Equations (5.3.28) and (5.3.29) show the main characteristic of the Gauss-Markov process for which its autocorrelation function fades exponentially with time, and the rate of fade is guided by the correlation time constant $\tau = 1/\beta$.

Once the process noise characteristics are clear we can properly incorporate it in the orbit estimation procedure as a means of representing unmodeled accelerations, and here, two main procedures are discussed. The first is based mainly on the work from Ingram [8] and is thoroughly discussed also by Myers and Tapley [24, 21] which follows from the same general results obtained in Section 5.3.1, while the second derives from linear dynamical systems theory [26, 7].

Equivalent stochastic process

The main difficulty encountered in using a first-order Gauss-Markov process lies in the presence of a stochastic integral in equation (5.3.27) which cannot be solved with conventional deterministic methods. The solution proposed by Ingram exploits the fact that this integral is Gaussian, so it can be replaced by and equivalent same mean and same variance process. In particular, this equivalent process if found by assuming that the white noise process $u(t)$ is constant over each time interval $\Delta t = t_j - t_i$, and if we define

$$L_k = \int_{t_i}^{t} e^{-\beta(t-\tau)} u(\tau)\, d\tau,$$

we have

$$E[L_k] = E\left[\int_{t_i}^{t} e^{-\beta(t-\tau)} u(\tau)\, d\tau\right] = 0 \tag{5.3.30}$$

by definition, while its variance is [8, 10]

$$\begin{aligned} E\left[L_k^2\right] &= E\left[\int_{t_i}^{t}\int_{t_i}^{\tau} e^{-\beta(2t-\tau-\nu)} u(\tau)u(\nu)\mathrm{d}\tau\mathrm{d}\nu\right] \\ &= \frac{\sigma^2}{2\beta}\left[1 - e^{-2\beta(t_j-t_i)}\right], \end{aligned} \tag{5.3.31}$$

thus, for the reasons previously stated, the integral L_k can be replaced by the following equivalent stochastic process:

$$L_k(t_i) = u_k\sqrt{\frac{\sigma^2}{2\beta}\left[1 - e^{-2\beta(t_j-t_i)}\right]}, \tag{5.3.32}$$

where u_k is a random *sequence* with zero mean and unitary variance,

$$E[u_k] = 0, \quad E[u_{ki}u_{kj}] = \delta_{ij}.$$

From this result, we can write rewrite (5.3.27), describing the integral solution of the Langevin equation, as:

$$\begin{aligned} \varepsilon(t) &= \varepsilon(t_i)e^{-\beta(t_j-t_i)} + L_k(t_i) \\ &= \varepsilon(t_i)e^{-\beta(t_j-t_i)} + u_k(t_i)\sqrt{\frac{\sigma^2}{2\beta}\left[1 - e^{-2\beta(t_j-t_i)}\right]}, \end{aligned} \tag{5.3.33}$$

and the random component will contribute to the definition of the process noise covariance matrix **Q**. The value of $\Delta t = t_j - t_i$ should be restricted to a sufficiently small and finite interval, so that the equivalent process is a good approximation of the actual one. In any case, it is desirable for $\varepsilon(t)$ to be as constant as possible over this interval and this is accomplished by ensuring that

$$e^{-\beta(t_j-t_i)} \approx 0.9. \tag{5.3.34}$$

Adding a first-order Gauss-Markov process to the accelerations acting on a spacecraft in order to compensate for unmodeled forces leads to the following set of differential equations governing the spacecraft motion:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a_m} + \boldsymbol{\varepsilon} \\ \dot{\boldsymbol{\varepsilon}} &= \mathbf{B}\boldsymbol{\varepsilon} + \mathbf{G}\mathbf{u}, \end{aligned} \tag{5.3.35}$$

where $\mathbf{a_m}$ are the modeled accelerations and $\boldsymbol{\varepsilon}$ is a three-dimensional Gauss-Markov process representing unmodeled accelerations. Here, **B** and **G** are 3×3 diagonal matrices defined as:

$$\mathbf{B} = \begin{bmatrix} -\beta_x & 0 & 0 \\ 0 & -\beta_y & 0 \\ 0 & 0 & -\beta_z \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} q_x & 0 & 0 \\ 0 & q_y & 0 \\ 0 & 0 & q_z \end{bmatrix},$$

where $q_x = \sigma_x/\sqrt{2\beta_x}$ and $\beta_x = 1/\tau_x$ with similar definitions for $\beta_y, \beta_z, q_y, q_z$ and, $\mathbf{u}(t)$ is a three components white Gaussian noise with $E[\mathbf{u}(t)] = 0$ and $E[\mathrm{G}\mathbf{u}(t)\mathbf{u}^T(\tau)\mathrm{G}^T] = \mathbf{Q}'\delta(t-\tau)$.

Making use of the equivalent stochastic process described in (5.3.32) we note that an integral solution for the third of equations (5.3.35) can be written as

$$\boldsymbol{\varepsilon} = \mathbf{E}(t)\boldsymbol{\varepsilon}_{k-1} + \mathbf{L}_{k-1}, \tag{5.3.36}$$

where

$$\mathbf{E} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{bmatrix} = \begin{bmatrix} e^{-\beta_x \Delta t} & 0 & 0 \\ 0 & e^{-\beta_y \Delta t} & 0 \\ 0 & 0 & e^{-\beta_z \Delta t} \end{bmatrix}, \tag{5.3.37}$$

and

$$\mathbf{L}_{k-1} = \begin{bmatrix} q_x\sqrt{1-\alpha_x^2}u_{x(k-1)} \\ q_y\sqrt{1-\alpha_y^2}u_{y(k-1)} \\ q_z\sqrt{1-\alpha_z^2}u_{z(k-1)} \end{bmatrix}, \tag{5.3.38}$$

then, following the same procedure adopted in Section 5.3.1, to find a proper formulation for the process noise covariance matrix $\mathbf{Q}_{k-1}$, we start by writing an integral solution for equations (5.3.35), i.e.,

$$\mathbf{r}(t) = \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t + \int_{t_{k-1}}^{t} \big[\mathbf{a}_m(\tau) + \mathbf{E}(\tau)\boldsymbol{\varepsilon}_{k-1}\big](t-\tau)\mathrm{d}\tau + \int_{t_{k-1}}^{t} \mathbf{L}_{k-1}(t-\tau)\mathrm{d}\tau \tag{5.3.39}$$

$$\mathbf{v}(t) = \mathbf{v}_{k-1} + \int_{t_{k-1}}^{t} \big[\mathbf{a}_m + \mathbf{E}(\tau)\boldsymbol{\varepsilon}_{k-1}\big]\mathrm{d}\tau + \int_{t_{k-1}}^{t} \mathbf{L}_{k-1}\mathrm{d}\tau \tag{5.3.40}$$

$$\boldsymbol{\varepsilon}(t) = \mathbf{E}(t)\boldsymbol{\varepsilon}_{k-1} + \mathbf{L}_{k-1}. \tag{5.3.41}$$

As for SNC, only the stochastic portions of the above equations will contribute to the determination od the process noise covariance, and referring to equation (5.3.8) we have that

$$\boldsymbol{\eta}_{k-1} = \begin{bmatrix} \int_{t_{k-1}}^{t} \mathbf{L}_{k-1}(t-\tau)\mathrm{d}\tau \\ \int_{t_{k-1}}^{t} \mathbf{L}_{k-1}\mathrm{d}\tau \\ \mathbf{L}_{k-1} \end{bmatrix}, \tag{5.3.42}$$

then, we can solve the integrals considering that $\mathbf{L}_{k-1}$ is constant over the time interval Δt arriving at

$$\boldsymbol{\eta}_{k-1} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \mathbf{L}_{k-1} \\ \Delta t \mathbf{L}_{k-1} \\ \mathbf{L}_{k-1} \end{bmatrix}. \tag{5.3.43}$$

At this point, the definition of the process noise covariance matrix $\mathbf{Q}_{k-1}$ for the DMC algorithm still follows from equation (5.3.12), which gives:

$$\mathbf{Q}_{11} = \frac{1}{4}\Delta t^4 E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \tag{5.3.44}$$

$$\mathbf{Q}_{12} = \mathbf{Q}_{21} = \frac{1}{2}\Delta t^3 E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \tag{5.3.45}$$

$$\mathbf{Q}_{13} = \mathbf{Q}_{31} = \frac{1}{2}\Delta t^2 E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \tag{5.3.46}$$

$$\mathbf{Q}_{22} = \Delta t^2 E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \tag{5.3.47}$$

$$\mathbf{Q}_{23} = \mathbf{Q}_{32} = \Delta t E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \tag{5.3.48}$$

$$\mathbf{Q}_{33} = E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\}, \tag{5.3.49}$$

and if we define the 3-dimensional diagonal covariance matrix $\mathbf{\Lambda}$, following equation (5.3.31), as

$$\begin{aligned}\mathbf{\Lambda} &= E\left\{\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T\right\} \\ &= \begin{bmatrix} q_x^2\left(1-\alpha_x^2\right) & 0 & 0 \\ 0 & q_y^2\left(1-\alpha_y^2\right) & 0 \\ 0 & 0 & q_z^2\left(1-\alpha_z^2\right)\end{bmatrix},\end{aligned} \tag{5.3.50}$$

the process noise covariance matrix for the DMC algorithm can be written as follows

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \mathbf{Q}_{13} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} & \mathbf{Q}_{23} \\ \mathbf{Q}_{31} & \mathbf{Q}_{32} & \mathbf{Q}_{33}\end{bmatrix} = \begin{bmatrix} \frac{1}{4}\mathbf{\Lambda}\Delta t^4 & \frac{1}{2}\mathbf{\Lambda}\Delta t^3 & \frac{1}{2}\mathbf{\Lambda}\Delta t^2 \\ \frac{1}{2}\mathbf{\Lambda}\Delta t^3 & \mathbf{\Lambda}\Delta t^2 & \mathbf{\Lambda}\Delta t \\ \frac{1}{2}\mathbf{\Lambda}\Delta t^2 & \mathbf{\Lambda}\Delta t & \mathbf{\Lambda}\end{bmatrix}. \tag{5.3.51}$$

Estimating Unmodeled Accelerations

As opposed to the SNC algorithm, where unknown accelerations are purely random, the DMC algorithm enables the estimation of these unmodeled effects through the use of a time-correlated model and the estimation of its parameters. In this Section, we discuss this aspect from a computational point of view, considering the main implications it has on the structure of the equations involved, both for the state propagation and estimation with an extended Kalman filter.

Let us begin by assuming that in a typical orbit determination problem we want to estimate the satellite's position $\mathbf{r}$ and velocity $\mathbf{v}$, as well as different geometrical and dynamical parameters that here will be indicated by the p-dimensional vector $\mathbf{P}$. In this case, the n-dimensional state vector $\mathbf{X}$ would be:

$$\mathbf{X} = \begin{Bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{P} \end{Bmatrix}. \tag{5.3.52}$$

The estimation of unmodeled accelerations is performed by including the process noise parameters inside the state vector, which means that both $\boldsymbol{\varepsilon}$ and the time correlation coefficients β_x, β_y, β_z may be added. However, for this study we consider the inclusion of $\boldsymbol{\varepsilon}$ only since it is the parameter that actually represents accelerations, whereas in many studies for Earth-orbiting satellites the β_i parameters are fixed to a proper value. For example, Darugna et al. (2022) [5] use a fixed value of

$\beta_i = 600\,\mathrm{s}$ for the orbit determination of Sentinel-6A. In view of this, the estimation state vector is extended as follows

$$\mathbf{X} = \begin{Bmatrix} \mathbf{r} \\ \mathbf{v} \\ \boldsymbol{\varepsilon} \\ \mathbf{P} \end{Bmatrix}, \tag{5.3.53}$$

and now comprises 9×1 components for $\mathbf{r}$, $\mathbf{v}$ and $\boldsymbol{\varepsilon}$, while the last $p \times 1$ part contains the estimated model parameters $\mathbf{P}$. The dimension of $\mathbf{X}$ will still be referred to as n. To apply the Kalman filter algorithm in this case, we need to integrate a system of 9×1 equation of motion plus $n \times n$ variational equations for the state transition matrix, that is

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a_m} + \boldsymbol{\varepsilon} \\ \dot{\boldsymbol{\varepsilon}} &= \mathbf{B}\boldsymbol{\varepsilon} + \mathbf{G}\mathbf{u} \\ \boldsymbol{\Phi}(t, t_0) &= \mathbf{A}(t)\boldsymbol{\Phi}(t, t_0), \end{aligned} \tag{5.3.54}$$

with initial conditions

$$\begin{aligned} \mathbf{X}(t_0) &= \mathbf{X}_0 \\ \boldsymbol{\Phi}(t_0, t_0) &= \mathbf{I}. \end{aligned}$$

Regarding the first 9 equations of motion we can see how the differential equation for $\boldsymbol{\varepsilon}$ contains a random term $\mathbf{u}$ (see equation (5.3.35)) that cannot be integrated with conventional methods, however, we can consider the deterministic portion only, whereas the random effects will be taken into account in the process noise covariance matrix $\mathbf{Q}$ as was shown in the previous Section. This consideration leads to a new form for the equation of motions that contains only the time-correlated portion of $\boldsymbol{\varepsilon}$ that we indicate with $\boldsymbol{\varepsilon}'$, i.e.,

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a_m} + \boldsymbol{\varepsilon}' \\ \dot{\boldsymbol{\varepsilon}}' &= \mathbf{B}\boldsymbol{\varepsilon}' \end{aligned} \tag{5.3.55}$$

and these are often known as the *filter-world* equations. Note that actually, the equation for $\boldsymbol{\varepsilon}'$ has its own analytical solution given by

$$\boldsymbol{\varepsilon}'_{k+1} = \mathbf{E}(\Delta t)\boldsymbol{\varepsilon}'_k$$

where $\mathbf{E}$ defined in (5.3.37) acts as a state transition matrix. Nonetheless, its numerical integration is simple and does not drastically compromise the computational cost, for this reason we choose to follow this latter approach.

For what concerns the variational equations for the state transition matrix we proceed by examining the structure of $\mathbf{A}(t)$ and $\boldsymbol{\Phi}(t, t_0)$. In terms of $\mathbf{r}$, $\mathbf{v}$, $\boldsymbol{\varepsilon}$ and $\mathbf{P}$ these matrices can be written as

$$\mathbf{A}(t) = \left[\frac{\partial \dot{\mathbf{X}}}{\partial \mathbf{X}}\right]^* = \begin{bmatrix} \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{r}} & \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{v}} & \frac{\partial \dot{\mathbf{r}}}{\partial \boldsymbol{\varepsilon}'} & \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{P}} \\ \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{r}} & \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{v}} & \frac{\partial \dot{\mathbf{v}}}{\partial \boldsymbol{\varepsilon}'} & \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{P}} \\ \frac{\partial \dot{\boldsymbol{\varepsilon}}'}{\partial \mathbf{r}} & \frac{\partial \dot{\boldsymbol{\varepsilon}}'}{\partial \mathbf{v}} & \frac{\partial \dot{\boldsymbol{\varepsilon}}'}{\partial \boldsymbol{\varepsilon}'} & \frac{\partial \dot{\boldsymbol{\varepsilon}}'}{\partial \mathbf{P}} \\ \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{r}} & \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{v}} & \frac{\partial \dot{\mathbf{P}}}{\partial \boldsymbol{\varepsilon}'} & \frac{\partial \dot{\mathbf{P}}}{\partial \mathbf{P}} \end{bmatrix}, \tag{5.3.56}$$

where again the superscript $*$ indicates evaluation along the reference trajectory, and

$$\mathbf{\Phi}(t,t_0)=\frac{\partial \mathbf{X}}{\partial \mathbf{X}_0}=\begin{bmatrix}\mathbf{\Phi_{rr}} & \mathbf{\Phi_{rv}} & \mathbf{\Phi}_{\mathbf{r}\varepsilon} & \mathbf{\Phi_{rp}}\\ \mathbf{\Phi_{vr}} & \mathbf{\Phi_{vv}} & \mathbf{\Phi}_{\mathbf{v}\varepsilon} & \mathbf{\Phi_{vp}}\\ \mathbf{\Phi}_{\varepsilon\mathbf{r}} & \mathbf{\Phi}_{\varepsilon\mathbf{v}} & \mathbf{\Phi}_{\varepsilon\varepsilon} & \mathbf{\Phi}_{\varepsilon\mathbf{p}}\\ \mathbf{\Phi_{pr}} & \mathbf{\Phi_{pv}} & \mathbf{\Phi}_{\mathbf{p}\varepsilon} & \mathbf{\Phi_{pp}}\end{bmatrix}. \tag{5.3.57}$$

From the way these matrices are written, we see how they form a system of $n \times n$ equations. Nevertheless, we shall not actually perform $n \times n$ integrations since both $\mathbf{A}$ and $\mathbf{\Phi}$ contain some null entries. In fact, for what concerns $\mathbf{\Phi}$ we have that

$$\mathbf{\Phi}_{\varepsilon\mathbf{r}}=\mathbf{\Phi}_{\varepsilon\mathbf{v}}=\mathbf{\Phi}_{\varepsilon\mathbf{p}}=\mathbf{\Phi_{pr}}=\mathbf{\Phi_{pv}}=\mathbf{\Phi}_{\mathbf{p}\varepsilon}=\mathbf{0}, \qquad \mathbf{\Phi_{pp}}=\mathbf{I},$$

while for matrix $\mathbf{A}(t)$

$$\frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{r}}=\frac{\partial\dot{\mathbf{r}}}{\partial\varepsilon'}=\frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{P}}=\frac{\partial\dot{\varepsilon}'}{\partial\mathbf{r}}=\frac{\partial\dot{\varepsilon}'}{\partial\mathbf{v}}=\frac{\partial\dot{\varepsilon}'}{\partial\mathbf{P}}=\frac{\partial\dot{\mathbf{P}}}{\partial\mathbf{r}}=\frac{\partial\dot{\mathbf{P}}}{\partial\mathbf{v}}=\frac{\partial\dot{\mathbf{P}}}{\partial\varepsilon'}=\frac{\partial\dot{\mathbf{P}}}{\partial\mathbf{P}}=\mathbf{0}$$

and

$$\begin{aligned}&\frac{\partial\dot{\mathbf{r}}}{\partial\mathbf{v}}=\frac{\partial\dot{\mathbf{v}}}{\partial\varepsilon'}=\mathbf{I}\\ &\frac{\partial\dot{\varepsilon}'}{\partial\varepsilon'}=\mathbf{B}.\end{aligned}$$

This means that $\mathbf{A}(t)$ and $\mathbf{\Phi}$ reduce to

$$\mathbf{A}(t)=\left[\frac{\partial\dot{\mathbf{X}}}{\partial\mathbf{X}}\right]^*=\begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0}\\ \frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{r}} & \frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{v}} & \mathbf{I} & \frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{P}}\\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}, \qquad \mathbf{\Phi}(t,t_0)=\begin{bmatrix}\mathbf{\Phi_{rr}} & \mathbf{\Phi_{rv}} & \mathbf{\Phi}_{\mathbf{r}\varepsilon} & \mathbf{\Phi_{rp}}\\ \mathbf{\Phi_{vr}} & \mathbf{\Phi_{vv}} & \mathbf{\Phi}_{\mathbf{v}\varepsilon} & \mathbf{\Phi_{vp}}\\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_{\varepsilon\varepsilon} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}, \tag{5.3.58}$$

and the system of variational equations for computing the state transition matrix in the DMC algorithm, following le last of equations (5.3.54), becomes:

$$\begin{aligned}
\dot{\mathbf{\Phi}}_{\mathbf{rr}}&=\mathbf{\Phi_{vr}}\\
\dot{\mathbf{\Phi}}_{\mathbf{rv}}&=\mathbf{\Phi_{vv}}\\
\dot{\mathbf{\Phi}}_{\mathbf{vr}}&=\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{r}}\mathbf{\Phi_{rr}}+\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{v}}\mathbf{\Phi_{vr}}\\
\dot{\mathbf{\Phi}}_{\mathbf{vv}}&=\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{r}}\mathbf{\Phi_{rv}}+\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{v}}\mathbf{\Phi_{vv}}\\
\dot{\mathbf{\Phi}}_{\mathbf{r}\varepsilon}&=\mathbf{\Phi}_{\mathbf{v}\varepsilon}\\
\dot{\mathbf{\Phi}}_{\mathbf{v}\varepsilon}&=\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{r}}\mathbf{\Phi}_{\mathbf{r}\varepsilon}+\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{v}}\mathbf{\Phi}_{\mathbf{v}\varepsilon}+\mathbf{\Phi}_{\varepsilon\varepsilon}\\
\dot{\mathbf{\Phi}}_{\varepsilon\varepsilon}&=\mathbf{B}\mathbf{\Phi}_{\varepsilon\varepsilon}\\
\dot{\mathbf{\Phi}}_{\mathbf{rp}}&=\mathbf{\Phi_{vp}}\\
\dot{\mathbf{\Phi}}_{\mathbf{vp}}&=\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{r}}\mathbf{\Phi_{rp}}+\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{v}}\mathbf{\Phi_{vp}}+\frac{\partial\dot{\mathbf{v}}}{\partial\mathbf{p}}.
\end{aligned} \tag{5.3.59}$$

By properly examining the structure of the matrices involved, the system of equations to be solved for the DMC algorithm has been reduced from $9+n\times n$ equations to just $9+9\times 7+6\times p$. Other than these considerations for the propagation of the state equations, the filter structure involved for the parameters estimations remain essentially the same described for the extended Kalman filter in Section 5.2, the only necessary precaution concerns the formulation of the process noise covariance $\mathbf{Q}$. In Section 5.3.2, we arrived at an expression for $\mathbf{Q}$ that is of dimension 9×9, however, for a more general case, as the one we just discussed, the state vector may also include constant geometrical or dynamical parameters represented by $\mathbf{p}$, and to maintain dimensional consistency with the solution covariance $\mathbf{P}$, as dictated by the time-update equation (5.3.4), the matrix $\mathbf{Q}$ must be extended from 9×9 to $n\times n$, thus accounting for the parameters vector size $\mathbf{p}$. This is achieved by considering that these parameters are constant for every propagation interval between two subsequent observations, so, the process noise has no direct effect on them and they are treated as *bias states* [7]. In other words, the entries in $\mathbf{Q}$ associated to these constant states are simply null and, referring to equation (5.3.51), the process noise covariance for the time updated of $\mathbf{P}$ will have only the upper left portion populated, i.e.,

$$\mathbf{Q}_{k-1}=\begin{bmatrix}\frac{1}{4}\mathbf{\Lambda}\Delta t^4 & \frac{1}{2}\mathbf{\Lambda}\Delta t^3 & \frac{1}{2}\mathbf{\Lambda}\Delta t^2 & \mathbf{0}\\ \frac{1}{2}\mathbf{\Lambda}\Delta t^3 & \mathbf{\Lambda}\Delta t^2 & \mathbf{\Lambda}\Delta t & \mathbf{0}\\ \frac{1}{2}\mathbf{\Lambda}\Delta t^2 & \mathbf{\Lambda}\Delta t & \mathbf{\Lambda} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}_{n\times n}, \tag{5.3.60}$$

where every $\mathbf{0}$ entry is adequately added to satisfy dimensional consistency.

5.3.3 Computing matrix Q through linear systems theory

In this approach, the process noise is still assumed to obey the Lagevin equation (5.3.26), but this time, the covariance matrix $\mathbf{Q}$ is computed through the similarity relation that involves the state transition matrix of the process noise itself.

Recalling that a linearized system without control but subjected to a continuous random disturbance $\mathbf{q}_C(t)$ can be described by the following equation:

$$\dot{\mathbf{x}}=\mathbf{F}(t)\mathbf{x}(t)+\mathbf{G}(t)\mathbf{q}_C(t), \tag{5.3.61}$$

where $\mathbf{q}_C(t)$ is assumed to be a white random noise with zero mean and variance-covariance $E[\mathbf{q}_C(t)\mathbf{q}_C^T(\tau)]=\mathbf{Q}_s\delta(t-\tau)$. The solution of equation (5.3.61) at discrete measurement times can be obtained as the sum of the *homogeneous* solution and a *particular* or *forced* solution,

$$\mathbf{x}(t_{i+1})=\mathbf{\Phi}(t_{i+1},t_i)\mathbf{x}(t_i)+\mathbf{q}_D(t_{i+1},t_i), \tag{5.3.62}$$

where $\mathbf{\Phi}(t_{i+1},t_i)$ is the state transition matrix of the system, and $\mathbf{q}_D(t_{i+1},t_i)$ is the discrete form of the process noise given by:

$$\mathbf{q}_D(t_{i+1},t_i)=\int_{t_i}^{t_{i+1}}\mathbf{\Phi}(t_{i+1},\tau)\mathbf{G}(\tau)\mathbf{q}_C(\tau)\,d\tau. \tag{5.3.63}$$

Now, the process noise covariance matrix $\mathbf{Q}_D(t_{i+1},t_i)$ can be computed as [7]:

$$\begin{aligned}\mathbf{Q}_D(t_{i+1},t_i) &= E[\mathbf{q}_D(t)\mathbf{q}_D^T(\tau)] \\ &= E\left(\left[\int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1},\tau)\mathbf{G}(\tau)\mathbf{q}_C(\tau)\,d\tau\right]\left[\int_{t_i}^{t_{i+1}} \mathbf{q}_C^T(\tau)\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t_{i+1},\tau)\,d\tau\right]\right) \\ &= \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1},\tau)\mathbf{G}(\tau)E\left[\mathbf{q}_C(\tau)\mathbf{q}_C(\tau)^T\right]\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t_{i+1},\tau)\,d\tau \\ &= \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1},\tau)\mathbf{G}(\tau)\mathbf{Q}_s\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t_{i+1},\tau)\,d\tau,\end{aligned} \tag{5.3.64}$$

where the expectation was moved within a single integral since the only random variable is $\mathbf{q}_C(t)$, and $E[\mathbf{q}_C(t)\mathbf{q}_C^T(\tau)] = 0$ for $t \neq \tau$.

To compute the elements of the process noise covariance matrix needed for the orbit estimation problem, we can consider a simple position-velocity-acceleration system in which the acceleration is modelled as a first-order Gauss-Markov process. In this case the system of equations describing the dynamical system is:

$$\begin{Bmatrix}\dot{r}\\ \dot{v}\\ \dot{a}\end{Bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & -\beta\end{bmatrix}\begin{Bmatrix}r\\ v\\ a\end{Bmatrix} + \begin{Bmatrix}0\\ 0\\ q_C(t)\end{Bmatrix}. \tag{5.3.65}$$

Note that for the sake of simplicity this equation is written for a unidimensional system. Equation (5.3.65) clearly represents a linear time-invariant system in the form

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{q}_C(t), \tag{5.3.66}$$

with $\mathbf{F}$ and $\mathbf{G}$ being constant matrices, in particular for this case

$$\mathbf{F} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & -\beta\end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}, \tag{5.3.67}$$

and $\mathbf{x}$ and $\mathbf{q}_C$ are the time dependent state vector and noise vector respectively

$$\mathbf{x}(t) = \begin{Bmatrix}r(t)\\ v(t)\\ a(t)\end{Bmatrix}, \qquad \mathbf{q}_C(t) = \begin{Bmatrix}0\\ 0\\ q_C(t)\end{Bmatrix}. \tag{5.3.68}$$

Now, in order to obtain the state transition matrix $\mathbf{\Phi}$ needed in equation (5.3.64), we make use of the *Laplace transform* and its main properties, that allows the reformulation of equation (5.3.66) in the frequency domain, that is, we change from the time variable t to the complex Laplace variable $s = \sigma + j\omega$. The transform for a scalar variable is defined as:

$$\mathcal{L}\left[x(t)\right] = x(s) = \int_0^\infty x(t)e^{-st}\,dt \tag{5.3.69}$$

while the inverse Laplace transform is given by

$$\mathcal{L}^{-1}\left[x(s)\right] = x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} x(s)e^{st}\,ds. \tag{5.3.70}$$

In linear system theory, rather than the direct calculation of these integrals, we are more interested in selected applications of this transform, such as the fact that it is a linear operator, then

$$\begin{aligned} \mathcal{L}\left[x_1(t)+x_2(t)\right] &= \mathcal{L}\left[x_1(t)\right]+\mathcal{L}\left[x_2(t)\right]=x_1(s)+x_2(s) \\ \mathcal{L}\left[ax(t)\right] &= a\mathcal{L}\left[x(t)\right]=ax(s), \end{aligned} \tag{5.3.71}$$

and the Laplace transform of the time derivative of $x(t)$

$$\mathcal{L}\left[\dot{x}(t)\right] = sx(s) - x(0), \tag{5.3.72}$$

where $x(0)$ is the numerical initial value of $x(t)$ immediately before any input acts on it. Moreover, Laplace transforms of vectors and matrices can be defined similarly, for the n-vector $\mathbf{v}$ and the $n \times m$ matrix $\mathbf{M}$ we can write

$$\mathcal{L}\left[\mathbf{v}(\mathbf{t})\right] = \mathbf{v}(s) = \begin{Bmatrix} v_1(s) \\ v_2(s) \\ \vdots \\ v_n(s) \end{Bmatrix}, \qquad \mathcal{L}\left[\mathbf{M}(\mathbf{t})\right] = \mathbf{M}(s) = \begin{bmatrix} M_{11}(s) & M_{12}(s) & \dots & M_{1m}(s) \\ M_{21}(s) & M_{22}(s) & & \vdots \\ \vdots & & \ddots & \vdots \\ M_{n1}(s) & \dots & \dots & M_{nm}(s) \end{bmatrix}. \tag{5.3.73}$$

Making use of equations (5.3.69) and (5.3.72), the system differential equation (5.3.66) can be rewritten as

$$s\mathbf{x}(s) - \mathbf{x}(0) = \mathbf{F}\mathbf{x}(s) + \mathbf{G}\mathbf{q}_C(s) \tag{5.3.74}$$

that can be rearranged to give

$$\begin{aligned} s\mathbf{x}(s) - \mathbf{F}\mathbf{x}(s) &= \mathbf{x}(0) + \mathbf{G}\mathbf{q}_C(s) \\ \left[s\mathbf{I} - \mathbf{F}\right]\mathbf{x}(s) &= \mathbf{x}(0) + \mathbf{G}\mathbf{q}_C(s) \\ \mathbf{x}(s) &= \left[s\mathbf{I} - \mathbf{F}\right]^{-1}\left(\mathbf{x}(0) + \mathbf{G}\mathbf{q}_C(s)\right), \end{aligned} \tag{5.3.75}$$

and if we consider the initial condition response only we get

$$\mathbf{x}(s) = \left[s\mathbf{I} - \mathbf{F}\right]^{-1}\mathbf{x}(0). \tag{5.3.76}$$

Finally, given that $\mathbf{x}(0)$ is a constant we can take the Laplace antitransform of equation (5.3.76), that gives

$$\mathbf{x}(t) = \mathcal{L}^{-1}\left[\left(s\mathbf{I} - \mathbf{F}\right)^{-1}\right]\mathbf{x}(0) \tag{5.3.77}$$

from which it is clear that

$$\mathbf{\Phi}(t) = \mathcal{L}^{-1}\left[\left(s\mathbf{I} - \mathbf{F}\right)^{-1}\right]. \tag{5.3.78}$$

The matrix inverse $(s\mathbf{I} - \mathbf{F})^{-1}$ can be easily found to be:

$$\left(s\mathbf{I} - \mathbf{F}\right)^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} & \frac{1}{s^2(s+\beta)} \\ 0 & \frac{1}{s} & \frac{1}{s(s+\beta)} \\ 0 & 0 & \frac{1}{s+\beta} \end{bmatrix}, \tag{5.3.79}$$

then, by using the following common Laplace antitransforms

$$\mathcal{L}^{-1}\left[\frac{1}{s^n}\right] = \frac{t^{n-1}}{(n-1)!} \tag{5.3.80}$$

$$\mathcal{L}^{-1}\left[\frac{1}{(s-a)}\right] = e^{at}, \tag{5.3.81}$$

the following expression for the state transition matrix for the system described in equation (5.3.65) is obtained:

$$\mathbf{\Phi}(\Delta t) = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right) \\ 0 & 1 & \frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right) \\ 0 & 0 & e^{-\beta\Delta t} \end{bmatrix}, \tag{5.3.82}$$

and here, Δt is the time interval between subsequent discrete samples of the continuous dynamical model.

From equation (5.3.65) we can see that $\mathbf{G}(t) = \mathbf{I}$, and since the white random noise is acting only on the acceleration component,

$$\mathbf{Q}_s = E\left[\mathbf{q}_C(t)\mathbf{q}_C^T(t)\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & Q_s \end{bmatrix}, \tag{5.3.83}$$

and we can compute the process noise covariance matrix using equation (5.3.64) as:

$$\begin{aligned} \mathbf{Q}_D(t_{i+1}, t_i) &= \int_0^T \mathbf{\Phi}(\Delta t)\mathbf{Q}_s\mathbf{\Phi}^T(\Delta t)\, d(\Delta t) \\ &= Q_s \int_0^T \begin{bmatrix} \frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right) \\ \frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right) \\ e^{-\beta\Delta t} \end{bmatrix} \begin{bmatrix} \frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right) & \frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right) & e^{-\beta\Delta t} \end{bmatrix} d(\Delta t). \end{aligned} \tag{5.3.84}$$

Equation (5.3.84) allows us to compute every component of the matrix $\mathbf{Q}_D$ as a function of the time interval Δt, the time constant of the process noise $\tau = 1/\beta$, and the strength of the driving white noise represented by Q_s.

Hereafter, the computations of every matrix element following the integrals that derive from equation (5.3.84) are reported. Notice that integration by parts is required.

$$\begin{aligned} \mathbf{Q}_D(T)_{11} &= Q_s \int_0^T \left(\frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right)\right)^2 d(\Delta t) \\ &= \frac{Q_s}{\beta^2}\left[\frac{\Delta t^3}{3} - \frac{\Delta t^2}{\beta} + \frac{\Delta t}{\beta^2} - \frac{1}{2\beta^3}e^{-2\beta\Delta t} + \frac{2}{\beta^3}e^{-\beta\Delta t} - \frac{2}{\beta^2}e^{-\beta\Delta t}\left(\Delta t + \frac{1}{\beta}\right)\right]_{\Delta t=0}^{T} \\ &= \frac{Q_s}{\beta^2}\left[\frac{T^3}{3} - \frac{T^2}{\beta} + \frac{T}{\beta^2} - \frac{1}{2\beta^3}\left(e^{-2\beta T} - 1\right) - \frac{2}{\beta^2}Te^{-\beta T}\right]; \end{aligned} \tag{5.3.85}$$

$$\begin{aligned}\mathbf{Q}_D(T)_{12} &= Q_s \int_0^T \left[\frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right)\right] \frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right)\, d(\Delta t) \\ &= \frac{Q_s}{\beta^2}\left[\frac{\Delta t^2}{2} + \frac{\Delta t}{\beta}e^{-\beta\Delta t} - \frac{\Delta t}{\beta} + \frac{1}{2\beta^2}e^{-2\beta\Delta t} - \frac{1}{\beta^2}e^{-\beta\Delta t}\right]_{\Delta t=0}^{T} \\ &= \frac{Q_s}{\beta^2}\left[\frac{T^2}{2} + \left(\frac{T}{\beta} - \frac{1}{\beta^2}\right)\left(e^{-\beta T} - 1\right) + \frac{1}{2\beta^2}\left(e^{-2\beta T} - 1\right)\right];\end{aligned} \tag{5.3.86}$$

$$\begin{aligned}\mathbf{Q}_D(T)_{13} &= Q_s \int_0^T \left[\frac{\Delta t}{\beta} + \frac{1}{\beta^2}\left(e^{-\beta\Delta t} - 1\right)\right] e^{-\beta\Delta t}\, d(\Delta t) \\ &= \frac{Q_s}{\beta^2}\left[-\Delta t e^{-\beta\Delta t} - \frac{1}{2\beta}e^{-2\beta\Delta t}\right]_{\Delta t=0}^{T} \\ &= \frac{Q_s}{\beta^2}\left[-T e^{-\beta T} - \frac{1}{2\beta}\left(e^{-2\beta T} - 1\right)\right];\end{aligned} \tag{5.3.87}$$

$$\begin{aligned}\mathbf{Q}_D(T)_{22} &= Q_s \int_0^T \left(\frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right)\right)^2 d(\Delta t) \\ &= \frac{Q_s}{\beta^2}\left[\Delta t + \frac{2}{\beta}e^{-2\beta\Delta t} - \frac{1}{2\beta}e^{-2\beta\Delta t}\right]_{\Delta t=0}^{T} \\ &= \frac{Q_s}{\beta^2}\left[T + \frac{2}{\beta}\left(e^{-\beta T} - 1\right) - \frac{1}{2\beta}\left(e^{-2\beta T} - 1\right)\right];\end{aligned} \tag{5.3.88}$$

$$\begin{aligned}\mathbf{Q}_D(T)_{23} &= Q_s \int_0^T \frac{1}{\beta}\left(1 - e^{-\beta\Delta t}\right) e^{-\beta\Delta t}\, d(\Delta t) \\ &= \frac{Q_s}{\beta^2}\left[-e^{-\beta\Delta t} + \frac{1}{2\beta}e^{-2\beta\Delta t}\right]_{\Delta t=0}^{T} \\ &= \frac{Q_s}{\beta^2}\left[1 - e^{-\beta T} + \frac{1}{2}\left(e^{-2\beta T} - 1\right)\right];\end{aligned} \tag{5.3.89}$$

$$\begin{aligned}\mathbf{Q}_D(T)_{33} &= Q_s \int_0^T e^{-2\beta\Delta t}\, d(\Delta t) \\ &= \frac{Q_s}{2\beta}\left(1 - e^{-2\beta T}\right);\end{aligned} \tag{5.3.90}$$

$\mathbf{Q}_D(T)_{21} = \mathbf{Q}_D(T)_{12}$, $\mathbf{Q}_D(T)_{31} = \mathbf{Q}_D(T)_{13}$, $\mathbf{Q}_D(T)_{32} = \mathbf{Q}_D(T)_{23}$ for simmetry.

So far, the process noise covariance matrix has been constructed considering a unidimensional poistion-velocity-acceleration dynamical system, eq. (5.3.65), and therefore its dimension is 3 × 3. We now need to extend these results to the orbit estimation problem in which the dimension of the system of equations may be as large as the estimation requires, considered that we may have a large number of geometrical and dynamical parameters we want to estimate. We recall that for the general case, the dimension of $\mathbf{Q}_D(T)$ will be $n \times n$, where $n = 6 + p + q$, that is 6 components comprising position and velocity, p dynamical and q geometrical parameters. Since the process noise is added to the system as an unmodelled acceleration it will affect only position and velocity components, meaning that the same considerations made in Section 5.3.2 about the structure of the matrix $\mathbf{Q}$ still holds here. Specifically, the matrix $\mathbf{Q}_D(T)$ for the orbit estimation problem will only have non-zero elements in the upper-left 6×6 or 9×9 portion, depending on whether we are also estimating the deterministic portion of the Gauss-Markov process or not. In the case we are not estimating the time constants of the process, nor its deterministic part, the process noise covariance matrix will contain components related to position and velocity only, in which the upper-left 6×6 matrix is composed of four 3×3 diagonal sub-matrices, each associated to one of the elements from equations (5.3.85) to (5.3.90). Namely,

$$\mathbf{Q}_D(T) = \begin{bmatrix} \mathbf{Q}_{rr} & \mathbf{Q}_{rv} & 0 & \dots & 0 \\ \mathbf{Q}_{vr} & \mathbf{Q}_{vv} & 0 & \dots & 0 \\ 0 & 0 & 0 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \dots & 0 \end{bmatrix}_{n\times n}, \tag{5.3.91}$$

where $\mathbf{Q}_{rr}, \mathbf{Q}_{rv}, \mathbf{Q}_{vr}, \mathbf{Q}_{vv}$ are 3×3 diagonal matrices:

$$\mathbf{Q}_{rr} = \begin{bmatrix} \mathbf{Q}_D(T)^x_{11} & 0 & 0 \\ 0 & \mathbf{Q}_D(T)^y_{11} & 0 \\ 0 & 0 & \mathbf{Q}_D(T)^z_{11} \end{bmatrix},$$

and the superscripts x, y, z indicate that the element is associated with a particular component of the position or velocity vector, while the subscripts indicate which element we are considering from equations (5.3.85) - (5.3.90). Similarly:

$$\mathbf{Q}_{rv} = \mathbf{Q}_{vr} = \begin{bmatrix} \mathbf{Q}_D(T)^x_{12} & 0 & 0 \\ 0 & \mathbf{Q}_D(T)^y_{12} & 0 \\ 0 & 0 & \mathbf{Q}_D(T)^z_{12} \end{bmatrix},$$

$$\mathbf{Q}_{vv} = \begin{bmatrix} \mathbf{Q}_D(T)^x_{22} & 0 & 0 \\ 0 & \mathbf{Q}_D(T)^y_{22} & 0 \\ 0 & 0 & \mathbf{Q}_D(T)^z_{22} \end{bmatrix},$$

and once computed, $\mathbf{Q}_D(T)$ can be used in equation (5.3.4) to propagate the solution covariance matrix accounting for the influence of process noise, allowing to overcome filter divergence.

Chapter 6

Optimal Smoothing

In the previous Sections we discussed how the Kalman filter, or sequential filter, is not only an optimal real-time estimator of the state at every time of observation, but it is also a *predictor* that computes optimal estimates at future times by propagating the last a posteriori values. In particular, the *a priori* estimate of the state at time t_i is the estimate given all observations from t_0 up to t_{i-1}, meaning that information in the observation $\mathbf{y}_i$ has not been included yet, while the *a posteriori* estimate at t_i, is the one given all measurements up to time t_i, thus including $\mathbf{y}_i$, namely

$$\bar{\mathbf{x}}_i = E\left(\mathbf{x}_i | \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}\right) \tag{6.0.1}$$

$$\hat{\mathbf{x}}_i = E\left(\mathbf{x}_i | \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{i}\right). \tag{6.0.2}$$

It may happen however, that at time t_i we have availability of measurements up to a later time, say for example $t_N > t_i$, and in this situation it is reasonable to think that we should obtain an even better estimate of the state $\mathbf{x}_i$, compared to the estimate given by (6.0.2), by including these additional observations. Specifically, we want to find an optimal way of computing the estimate

$$\hat{\mathbf{x}}_{t_i|t_N} = E\left(\mathbf{x}_i | \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_i, \ldots, \mathbf{y}_N\right), \quad t_N > t_i, \tag{6.0.3}$$

where the subscript $t_i|t_N$ is used precisely to indicate that the estimate occurs at time t_i considering all measurements from t_0 to t_N.

Depending on the availability of measurement data and how they are handled, three different classes of smoothing techniques can be identified:

- *Fixed Point Smoother*: in this type of smoother the estimate of the state is obtained at a fixed time t_i or several fixed times, meaning that as time passes by and new measurements are available, they are only used to keep improving the estimate of $\mathbf{x}_i$. This technique can be implemented as a real-time estimator for applications in which the system state at a particular time is considered critical, such as engine burnout for rocket boosters or orbital maneuvers for spacecrafts.

- *Fixed Lag Smoother*: here, the idea is that measurements are still acquired in a real-time fashion, but it is not crucial to obtain the estimate of $\mathbf{x}_i$ exactly at time t_i. Instead, a certain delay in estimation is admitted, so that $\mathbf{x}_i$ can be estimated using all measurements up to time t_{i+N}, with N being a fixed integer, thus taking advantage of the N additional observations. This technique is particularly suitable for communications and telemetry data reduction.

- *Fixed Interval Smoother*: in this case, measurements are collected throughout a fixed time interval $[t_0, t_N]$, and are subsequently processed to obtain a smoothed estimate of the state at every observation time, meaning that every estimate is based on information contained in all measurements inside this particular interval. This is inherently an offline technique, particularly used for post-flight analysis to obtain estimates of trajectory parameters and to assess the validity of control inputs.

For precise orbit determination, the smoothing technique of greater interest is the *fixed-interval* formulation that is described hereafter.

6.1 Fixed-Interval Forward-Backward Smoother

Given a time interval $[t_0, t_N]$, in fixed-interval smoothing we search for the estimate of the state $\hat{\mathbf{x}}_{t_i|t_N}$ at every time $t_0 \leq t_i \leq t_N$ based on the knowledge of all observations from t_0 up to t_f, that is

$$\hat{\mathbf{x}}_{t_i|t_N} = E\left(\mathbf{x}_i|\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_N\right). \tag{6.1.1}$$

In the *forward-backward* formulation, this estimate is obtained as a suitable combination of two optimal filters, one that operates in a forward manner from t_0 to the time of estimation t_i, and the other going backward in time from t_N to t_i. In particular, the forward filter at time t_i will give

$$\hat{\mathbf{x}}_{if} = E\left(\mathbf{x}_i|\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_i\right), \tag{6.1.2}$$

that is the a posteriori estimate of $\mathbf{x}_i$ with associated covariance matrix $\mathbf{P}_{if}$, meaning that a measurement update at t_i is performed, while the backward filter gives

$$\bar{\mathbf{x}}_{ib} = E\left(\mathbf{x}_i|\mathbf{y}_{i+1}, \mathbf{y}_{i+2}, \ldots, \mathbf{y}_N\right), \tag{6.1.3}$$

which is clearly an a priori estimate of $\mathbf{x}_i$ since the measurement y_i at time t_i has not been included, and its associated covariance matrix is indicated by $\bar{\mathbf{P}}_{ib}$. Moreover, the fact that the two filters always consider different sets of measurements ensures that the forward a posteriori and backward a priori estimates are independent, and no correlation will be present between forward and backward filter errors

$$E\left(\hat{\mathbf{x}}_{if}\bar{\mathbf{x}}_{ib}^T\right) = 0. \tag{6.1.4}$$

The smoothed estimate is then obtained as an optimal weighted combination of the output of these two filters at time t_i,

$$\hat{\mathbf{x}}_{t_i|t_N} = \mathbf{K}_f\hat{\mathbf{x}}_{if} + \mathbf{K}_b\bar{\mathbf{x}}_{ib} \tag{6.1.5}$$

with $\mathbf{K}_f$ and $\mathbf{K}_b$ being the weighting matrices for the forward and backward estimates respectively. If $\hat{\mathbf{x}}_{if}$ and $\bar{\mathbf{x}}_{ib}$ are obtained from optimal Kalman filters they must be unbiased estimates, and if we wish the smoothed estimate to be also *unbiased*, the weighting matrices must satisfy the following condition:

$$\mathbf{K}_f + \mathbf{K}_b = \mathbf{I}, \tag{6.1.6}$$

from which

$$\hat{\mathbf{x}}_{t_i|t_N} = \mathbf{K}_f\hat{\mathbf{x}}_{if} + (\mathbf{I} - \mathbf{K}_f)\bar{\mathbf{x}}_{ib}, \tag{6.1.7}$$

that clearly corresponds to an unbiased smoothed estimate given that

$$\begin{aligned} E\left[\hat{\mathbf{x}}_{t_i|t_N}\right] &= E\left[\mathbf{K}_f\hat{\mathbf{x}}_{if} + (\mathbf{I}-\mathbf{K}_f)\bar{\mathbf{x}}_{ib}\right] \\ &= \mathbf{K}_f E\left[\hat{\mathbf{x}}_{if}\right] + (\mathbf{I}-\mathbf{K}_f)E\left[\bar{\mathbf{x}}_{ib}\right] \\ &= \mathbf{K}_f\mathbf{x}_i + \mathbf{x}_i - \mathbf{K}_f\mathbf{x}_i \\ &= \mathbf{x}_i. \end{aligned} \tag{6.1.8}$$

Now, in order for the smoothed estimate to be optimal besides being unbiased, it should correspond to a *minimum variance* condition. We begin by computing the covariance matrix of the smoothed estimate as [17]:

$$\begin{aligned} \mathbf{P}_{t_i|t_N} &= E\left[(\mathbf{x}_i - \hat{\mathbf{x}}_{t_i|t_N})(\mathbf{x}_i - \hat{\mathbf{x}}_{t_i|t_N})^T\right] \\ &= E\left[(\mathbf{x}_i - \mathbf{K}_f\hat{\mathbf{x}}_{if} - (\mathbf{I}-\mathbf{K}_f)\bar{\mathbf{x}}_{ib})(\mathbf{x}_i - \mathbf{K}_f\hat{\mathbf{x}}_{if} - (\mathbf{I}-\mathbf{K}_f)\bar{\mathbf{x}}_{ib})^T\right] \\ &= E\left[((\mathbf{x}_i - \bar{\mathbf{x}}_{ib} + \mathbf{K}_f(\mathbf{x}_i - \hat{\mathbf{x}}_{if} - \mathbf{x}_i + \bar{\mathbf{x}}_{ib}))\,((\mathbf{x}_i - \bar{\mathbf{x}}_{ib} + \mathbf{K}_f(\mathbf{x}_i - \hat{\mathbf{x}}_{if} - \mathbf{x}_i + \bar{\mathbf{x}}_{ib}))^T\right] \\ &= E\left[(\boldsymbol{\varepsilon}_b + \mathbf{K}_f(\boldsymbol{\varepsilon}_f - \boldsymbol{\varepsilon}_b))(\boldsymbol{\varepsilon}_b + \mathbf{K}_f(\boldsymbol{\varepsilon}_f - \boldsymbol{\varepsilon}_b))^T\right] \\ &= E\left[\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T - \mathbf{K}_f\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T - \boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\mathbf{K}_f^T + \mathbf{K}_f(\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T + \boldsymbol{\varepsilon}_f\boldsymbol{\varepsilon}_f^T)\mathbf{K}_f^T\right] \\ &= \mathbf{P}_b - \mathbf{K}_f\mathbf{P}_b - \mathbf{P}_b\mathbf{K}_f^T + \mathbf{K}_f(\mathbf{P}_b + \mathbf{P}_f)\mathbf{K}_f^T, \end{aligned} \tag{6.1.9}$$

where $\boldsymbol{\varepsilon}_b = \mathbf{x}_i - \bar{\mathbf{x}}_{ib}$ and $\boldsymbol{\varepsilon}_f = \mathbf{x}_i - \hat{\mathbf{x}}_{if}$ represent estimation errors for the backward and forward filter, while $\mathbf{P}_b = E\left[\boldsymbol{\varepsilon}_b\boldsymbol{\varepsilon}_b^T\right]$ and $\mathbf{P}_f = E\left[\boldsymbol{\varepsilon}_f\boldsymbol{\varepsilon}_f^T\right]$ are the related covariance matrices.

The minimum variance smoother solution is found minimizing the trace of the covariance matrix for $\hat{\mathbf{x}}_{t_i|t_N}$ described by equation (6.1.9), in other words, we need to minimize the scalar performance index J defined by:

$$\begin{aligned} J &= tr(\mathbf{P}_{t_i|t_N}) \\ &= tr(\mathbf{P}_b - \mathbf{K}_f\mathbf{P}_b - \mathbf{P}_b\mathbf{K}_f^T + \mathbf{K}_f(\mathbf{P}_b + \mathbf{P}_f)\mathbf{K}_f^T), \end{aligned} \tag{6.1.10}$$

with respect to the weighting matrix $\mathbf{K}_f$, that translates into taking the following derivative

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{K}_f} &= \frac{\partial tr(\mathbf{P}_{t_i|t_N})}{\partial \mathbf{K}_f} \\ &= -2\mathbf{P}_b + 2\mathbf{K}_f(\mathbf{P}_b + \mathbf{P}_f), \end{aligned} \tag{6.1.11}$$

then, by setting equation (6.1.11) equal to zero, the value of $\mathbf{K}_f$ that minimizes the trace of the covariance matrix $\mathbf{P}_{t_i|t_N}$, and the subsequent value of $\mathbf{K}_b$ can be computed:

$$\mathbf{K}_f = \mathbf{P}_b(\mathbf{P}_b + \mathbf{P}_f)^{-1} \tag{6.1.12}$$

$$\mathbf{K}_b = \mathbf{P}_f(\mathbf{P}_b + \mathbf{P}_f)^{-1}. \tag{6.1.13}$$

Taking the value for $\mathbf{K}_f$ in equation (6.1.12) and substituting it into equation (6.1.9), will give the expression for the covariance matrix of the minimum variance smoothed estimate, namely

$$\begin{aligned} \mathbf{P}_{t_i|t_N} = \mathbf{P}_b &- \mathbf{P}_b(\mathbf{P}_b + \mathbf{P}_f)^{-1}\mathbf{P}_b - \mathbf{P}_b(\mathbf{P}_b + \mathbf{P}_f)^{-1}\mathbf{P}_b + \\ &+ \mathbf{P}_b(\mathbf{P}_b + \mathbf{P}_f)^{-1}(\mathbf{P}_b + \mathbf{P}_f)(\mathbf{P}_b + \mathbf{P}_f)^{-1}\mathbf{P}_b, \end{aligned} \tag{6.1.14}$$

where the symmetry of the matrices $\mathbf{P}_b$ and $\mathbf{P}_f$ was considered. Now, multiplying out the last term, and using the following matrix inversion identity $(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{B}^{-1}(\mathbf{A}\mathbf{B}+\mathbf{I})^{-1}$ for $(\mathbf{P}_b+\mathbf{P}_f)^{-1}$ leads to

$$\begin{aligned}\mathbf{P}_{t_i|t_N} &= \mathbf{P}_b - (\mathbf{P}_f\mathbf{P}_b^{-1}+\mathbf{I})^{-1}\mathbf{P}_b - (\mathbf{P}_f\mathbf{P}_b^{-1}+\mathbf{I})^{-1}\mathbf{P}_b + (\mathbf{P}_f\mathbf{P}_b^{-1}+\mathbf{I})^{-1}\mathbf{P}_b \\ &= \mathbf{P}_b - (\mathbf{P}_f\mathbf{P}_b^{-1}+\mathbf{I})^{-1}\mathbf{P}_b,\end{aligned} \tag{6.1.15}$$

then, for the last term on the RHS of equation (6.1.15) we can apply the matrix inversion lemma $(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}$ to finally obtain

$$\begin{aligned}\mathbf{P}_{t_i|t_N} &= \mathbf{P}_b - \mathbf{P}_b + (\mathbf{P}_f^{-1}+\mathbf{P}_b^{-1})^{-1} \\ &= (\mathbf{P}_f^{-1}+\mathbf{P}_b^{-1})^{-1},\end{aligned} \tag{6.1.16}$$

that is the expression for the covariance matrix of the *minimum variance unbiased smoothed estimate* of $\hat{\mathbf{x}}_{t_i|t_N}$. Moreover, combining equations (6.1.7), (6.1.12) and (6.1.13) gives the following expression for $\hat{\mathbf{x}}_{t_i|t_N}$

$$\begin{aligned}\hat{\mathbf{x}}_{t_i|t_N} &= \mathbf{P}_b(\mathbf{P}_b+\mathbf{P}_f)^{-1}\hat{\mathbf{x}}_{if} + \mathbf{P}_f(\mathbf{P}_b+\mathbf{P}_f)^{-1}\bar{\mathbf{x}}_{ib} \\ &= (\mathbf{I}+\mathbf{P}_f\mathbf{P}_b^{-1})^{-1}\hat{\mathbf{x}}_{if} + (\mathbf{P}_b\mathbf{P}_f^{-1}+\mathbf{I})^{-1}\bar{\mathbf{x}}_{ib} \\ &= (\mathbf{P}_f^{-1}+\mathbf{P}_b^{-1})^{-1}\mathbf{P}_f^{-1}\hat{\mathbf{x}}_{if} + (\mathbf{P}_f^{-1}+\mathbf{P}_b^{-1})^{-1}\mathbf{P}_b^{-1}\bar{\mathbf{x}}_{ib} \\ &= (\mathbf{P}_f^{-1}+\mathbf{P}_b^{-1})^{-1}\left[\mathbf{P}_f^{-1}\hat{\mathbf{x}}_{if} + \mathbf{P}_b^{-1}\bar{\mathbf{x}}_{ib}\right] \\ &= \mathbf{P}_{t_i|t_N}\left[\mathbf{P}_f^{-1}\hat{\mathbf{x}}_{if} + \mathbf{P}_b^{-1}\bar{\mathbf{x}}_{ib}\right].\end{aligned} \tag{6.1.17}$$

An alternative simple derivation of equations (6.1.16) and (6.1.17), can be achieved by analogy with previous discussed filters. Recalling that equation (6.1.2) for the forward filter is equivalent to

$$\begin{aligned}\hat{\mathbf{x}}_{if} &= \mathbf{P}_{if}\left(\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i + \bar{\mathbf{P}}_{if}^{-1}\bar{\mathbf{x}}_{if}\right) \\ &= \left(\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i + \bar{\mathbf{P}}_{if}^{-1}\right)^{-1}\left(\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i + \bar{\mathbf{P}}_{if}^{-1}\bar{\mathbf{x}}_{if}\right),\end{aligned} \tag{6.1.18}$$

which, as stated before, is the a posteriori estimate of the state at time t_i coming from the forward filter, and this, should be properly combined with $\bar{\mathbf{x}}_{ib}$ from the backward filter to obtain the smoothed estimate. To do so, we can see the backward filter output as an additional "measurement" of the state at t_i, with associated uncertainty described by its matrix $\bar{\mathbf{P}}_{ib}$, thus equation (6.1.18) can be modified to include this new "measurement" in the following way:

- $\bar{\mathbf{x}}_{ib}$ is seen as a new measurement and replaces $\mathbf{y}_i$;
- $\bar{\mathbf{P}}_{ib}$ takes the place of $\mathbf{R}_i$ since it represents errors associated with the measurement $\bar{\mathbf{x}}_{ib}$;
- $\mathbf{H}_i = \mathbf{I}$ because $\bar{\mathbf{x}}_{ib}$ can be treated as a *direct* measurement of the state, so there is no need for an observation-state mapping matrix;
- $\hat{\mathbf{x}}_{if}$ and $\mathbf{P}_{if}$ become a priori information for the smoothed estimate;

from these considerations equation (6.1.18) becomes

$$\begin{aligned}\hat{\mathbf{x}}_{t_i|t_N} &= \left(\bar{\mathbf{P}}_{ib}^{-1} + \mathbf{P}_{if}^{-1}\right)^{-1} \left(\bar{\mathbf{P}}_{ib}^{-1}\bar{\mathbf{x}}_{ib} + \mathbf{P}_{if}^{-1}\hat{\mathbf{x}}_{if}\right) \\ &= \mathbf{P}_{t_i|t_N}\left(\bar{\mathbf{P}}_{ib}^{-1}\bar{\mathbf{x}}_{ib} + \mathbf{P}_{if}^{-1}\hat{\mathbf{x}}_{if}\right),\end{aligned} \tag{6.1.19}$$

from which we clearly see the smoothed estimate as a weighted combination of the forward and backward filter outputs. Moreover, the covariance matrix associated with the smoothed estimate is given by

$$\mathbf{P}_{t_i|t_N} = \left(\bar{\mathbf{P}}_{ib}^{-1} + \mathbf{P}_{if}^{-1}\right)^{-1}, \tag{6.1.20}$$

that if rewritten as

$$\mathbf{P}_{t_i|t_N}^{-1} = \left(\bar{\mathbf{P}}_{ib}^{-1} + \mathbf{P}_{if}^{-1}\right), \tag{6.1.21}$$

shows that the smoothed estimate is at least as good as the estimate obtained by conventional filtering. In fact it will generally improve estimation accuracy except for the final time t_N, where by definition the smoothed estimate is the same as the filtered estimate, which is clear from equations (6.1.1) and (6.1.2) since at t_N,

$$\hat{\mathbf{x}}_{t_N|t_N} = \hat{\mathbf{x}}_{Nf} = E\left(\mathbf{x}_N|\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_N\right). \tag{6.1.22}$$

6.1.1 Forward-Backward smoother algorithm

The forward filter is usually formulated as a conventional Kalman filter, that is initialized with the known a priori values

$$\bar{\mathbf{x}}_f(t_0) = \bar{\mathbf{x}}_{0f} \tag{6.1.23}$$
$$\bar{\mathbf{P}}_f(t_0) = \bar{\mathbf{P}}_{0f}. \tag{6.1.24}$$

Then, for every time t_i a measurement update is performed as

$$\mathbf{K}_i = \bar{\mathbf{P}}_{if}\mathbf{H}_i^T\left[\mathbf{H}_i\bar{\mathbf{P}}_{if}\mathbf{H}_i^T + \mathbf{R}_i\right]^{-1} \tag{6.1.25}$$
$$\hat{\mathbf{x}}_{if} = \bar{\mathbf{x}}_{if} + \mathbf{K}_i\left(\mathbf{y}_i - \mathbf{H}_i\bar{\mathbf{x}}_{if}\right) \tag{6.1.26}$$
$$\mathbf{P}_{if} = \left[\mathbf{I} - \mathbf{K}_i\mathbf{H}_i\right]\bar{\mathbf{P}}_{if}, \tag{6.1.27}$$

and the subsequent time update as

$$\bar{\mathbf{x}}_{i+1f} = \mathbf{\Phi}(t_{i+1}, t_i)\hat{\mathbf{x}}_{if} \tag{6.1.28}$$
$$\bar{\mathbf{P}}_{i+1f} = \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_{if}\mathbf{\Phi}^T(t_{i+1}, t_i) + \mathbf{Q}_{if}. \tag{6.1.29}$$

At every time t_i the quantities $\hat{\mathbf{x}}_{if}$ and $\mathbf{P}_{if}$ from equations (6.1.26) and (6.1.27) are stored, and will be later combined with the backward filter results so, computationally speaking, the smoothing process consists of two subsequent data sweeps, one forward and one backwards.

The backward filter runs backwards in time, and requires some considerations on its starting conditions at t_N. Theoretically, we have no a priori information for the backward filter at the final time t_N, reflecting the fact that no information used in the forward filter is allowed to enter the backward filter, thus making them independent. The absence of a priori information translates into a covariance matrix being infinite

$$\bar{\mathbf{P}}_{Nb} \rightarrow \infty, \tag{6.1.30}$$

which clearly is inconvenient to handle. For this reason, the backward filter is usually formulated as an *information filter*, where the quantity to be propagated is not the estimate of the state $\hat{\mathbf{x}}$, but rather

$$\hat{\mathbf{D}} = \mathbf{P}^{-1}\hat{\mathbf{x}} = \hat{\mathbf{\Lambda}}\hat{\mathbf{x}}. \tag{6.1.31}$$

The fact that we are considering the *information matrix* $\mathbf{\Lambda}$ instead of the covariance matrix, and considering equation (6.1.30), leads to a zero a priori value rather than an infinite value

$$\bar{\mathbf{\Lambda}}_{Nb} = \bar{\mathbf{P}}_{Nb}^{-1} \rightarrow 0, \tag{6.1.32}$$

which then gives the following starting conditions for the backward information filter:

$$\bar{\mathbf{P}}_{Nb}^{-1} = 0 \tag{6.1.33}$$
$$\bar{\mathbf{D}}_{Nb} = 0. \tag{6.1.34}$$

So, for every t_i going from t_N to t_0, the backward measurement update is performed as follows,

$$\hat{\mathbf{D}}_i = \bar{\mathbf{D}}_i + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i \tag{6.1.35}$$
$$\hat{\mathbf{\Lambda}}_i = \bar{\mathbf{\Lambda}}_i + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i \tag{6.1.36}$$
$$\hat{\mathbf{x}}_i = \hat{\mathbf{\Lambda}}_i^{-1}\hat{\mathbf{D}}_i, \tag{6.1.37}$$

where actually, equation (6.1.37) cannot be used until $\hat{\mathbf{\Lambda}}_i$ becomes non-singular. This may look like a serious flaw, but in reality the information filter propagates $\hat{\mathbf{D}}$ instead of $\hat{\mathbf{x}}$, making equation (6.1.37) somewhat useless. Then, the time update is given by the following equations considering that we are propagating backwards from t_{i+1} to t_i,

$$\mathbf{M}_i = \mathbf{\Phi}^T(t_{i+1}, t_i)\mathbf{\Lambda}_{i+1}\mathbf{\Phi}(t_{i+1}, t_i) \tag{6.1.38}$$
$$\mathbf{L}_i = \mathbf{M}_i\mathbf{\Gamma}(t_{i+1}, t_i)\left[\mathbf{Q}_{i+1}^{-1} + \mathbf{\Gamma}^T(t_i, t_{i+1})\mathbf{M}_i\mathbf{\Gamma}(t_i, t_{i+1})\right]^{-1} \tag{6.1.39}$$
$$\bar{\mathbf{D}}_i = \left[\mathbf{I} - \mathbf{L}_k\mathbf{\Gamma}^T(t_i, t_{i+1})\right]\mathbf{\Phi}^T(t_{i+1}, t_i)\hat{\mathbf{D}}_{i+1} \tag{6.1.40}$$
$$\bar{\mathbf{\Lambda}}_i = \left[\mathbf{I} - \mathbf{L}_i\mathbf{\Gamma}^T(t_i, t_{i+1})\right]\mathbf{M}_i, \tag{6.1.41}$$

where $\mathbf{Q}_i$ is the *process noise covariance matrix* and $\mathbf{\Gamma}(t_i, t_{i+1})$ is the *process noise transition matrix.*

The process described by equations (6.1.35) to (6.1.41) is iterated for every observation time from t_N to t_0, and at every t_i, $\hat{\mathbf{x}}_i$ and $\mathbf{P}_i$ (previously stored from the forward filter) are combined with $\bar{\mathbf{\Lambda}}_i$ and $\bar{\mathbf{D}}_i$ from the backward filter to obtain the smoothed estimate as:

$$\mathbf{X}_i = \left[\mathbf{I} + \mathbf{P}_i\bar{\mathbf{\Lambda}}_i\right] \tag{6.1.42}$$
$$\mathbf{W}_i = \mathbf{P}_i\mathbf{X}_i \tag{6.1.43}$$
$$\mathbf{Y}_i = \mathbf{I} - \mathbf{W}_i\bar{\mathbf{\Lambda}}_i \tag{6.1.44}$$
$$\mathbf{P}_{t_i|t_N} = \mathbf{Y}_i\mathbf{P}_i\mathbf{T}_i^T + \mathbf{W}_i\bar{\mathbf{\Lambda}}_i\mathbf{W}_i^T \tag{6.1.45}$$
$$\hat{\mathbf{x}}_{t_i|t_N} = \mathbf{X}_i\hat{\mathbf{x}}_i + \mathbf{P}_{t_i|t_f}\bar{\mathbf{D}}_i. \tag{6.1.46}$$

In equation (6.1.45), we can notice how the formulation for the backward information filter described above does not require the inversion of the matrix $\mathbf{P}_i$ coming from the forward filter, unlike the computation for $\mathbf{P}_{t_i|t_N}$ in euquation (6.1.20). However, equation (6.1.39) still requires $\mathbf{Q}_i$ to be non-singular, otherwise $\mathbf{L}_i$ will not be computable, and in practice this problem is worked around making small modification to $\mathbf{Q}_i$ to ensure invertibility.

As we saw earlier, the backward filter should be implemented in the form of an *information filter*, due to the fact that no a priori information nor measurements processed during the forward sweep should enter in the backward process, thus ensuring the absence of correlation between the two phases. In this regard a relevant consideration on the reference trajectory is necessary, since we should not forget that in reality we are not dealing with a linear system, but rather with a linearized orbit around a reference solution that is fairly near to the actual trajectory, meaning that besides discussing the a priori information the backward filter starts with, we must also discuss the initial values for the backward reference trajectory propagation. If we were to use and extended Kalman filter in the forward sweep and then use the final updated value as the starting condition for the backward reference trajectory, correlation between the two filters will surely be present because the updated forward trajectory not only contains a priori information from the forward sweep, it contains the information from the entire set of measurements in the fixed interval, making the two filters strongly dependent. To avoid this, the reference trajectory for the forward Kalman filter should never be updated with observational data, instead it should be simply propagated up to the final time t_N using only the available a priori values at the initial time t_0, then, the backward trajectory will start from the forward final value.

6.2 Fixed-Interval RTS Smoother

Several alternatives for the fixed-interval smoother have been developed, and one of the most common is named after Rauch, Tung and Striebel usually called the *RTS smoother* [25]. This formulation is attractive due to its greater computational efficiency compared to the forward-backward form discussed in the previous Section, since it doesn't require the direct computation of the backward estimate and covariance in order to obtain the smoothed values. In particular, the RTS smoother stores values coming from the forward filter only, and then applies a backward recursion starting from the last estimate and covariance obtained at the end of the forward sweep. It should be noted from the beginning that the forward Kalman filter, used to generate values that are stored for later backward recursion, must be optimal, thus, a consider filter cannot be used since it gives suboptimal estimates, introducing correlations between forward and backward solutions.

From a *Maximum Likelihood* standpoint we wish to find the smoothed estimate of $\mathbf{x}_k$ at time t_k as a backward recursive expression in terms of $\hat{\mathbf{x}}_{k+1}^N$, where $\hat{\mathbf{x}}_{k+1}^N$ is the smoothed estimate, that is the estimate of the state at time t_{k+1} given all measurements up to the final time t_N [24]:

$$\hat{\mathbf{x}}_{k+1}^N = E(\mathbf{x}_{k+1}|\mathbf{Y}_N). \tag{6.2.1}$$

In other words, we need to maximize the following conditional probability density function:

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Y}_N), \quad \mathbf{Y}_N = \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_N. \tag{6.2.2}$$

Form Bayes rule we can write (6.2.2) as

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Y}_N) &= \frac{p(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{Y}_N)}{p(\mathbf{Y}_N)} \\ &= \frac{p(\mathbf{y}_{k+1}, \ldots, \mathbf{y}_N|\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{Y}_k)}{p(\mathbf{Y}_N)} p(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{Y}_k) \\ &= \frac{p(\mathbf{Y}_k)}{p(\mathbf{Y}_N)} p(\mathbf{y}_{k+1}, \ldots, \mathbf{y}_N|\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{Y}_k) p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Y}_k), \end{aligned} \tag{6.2.3}$$

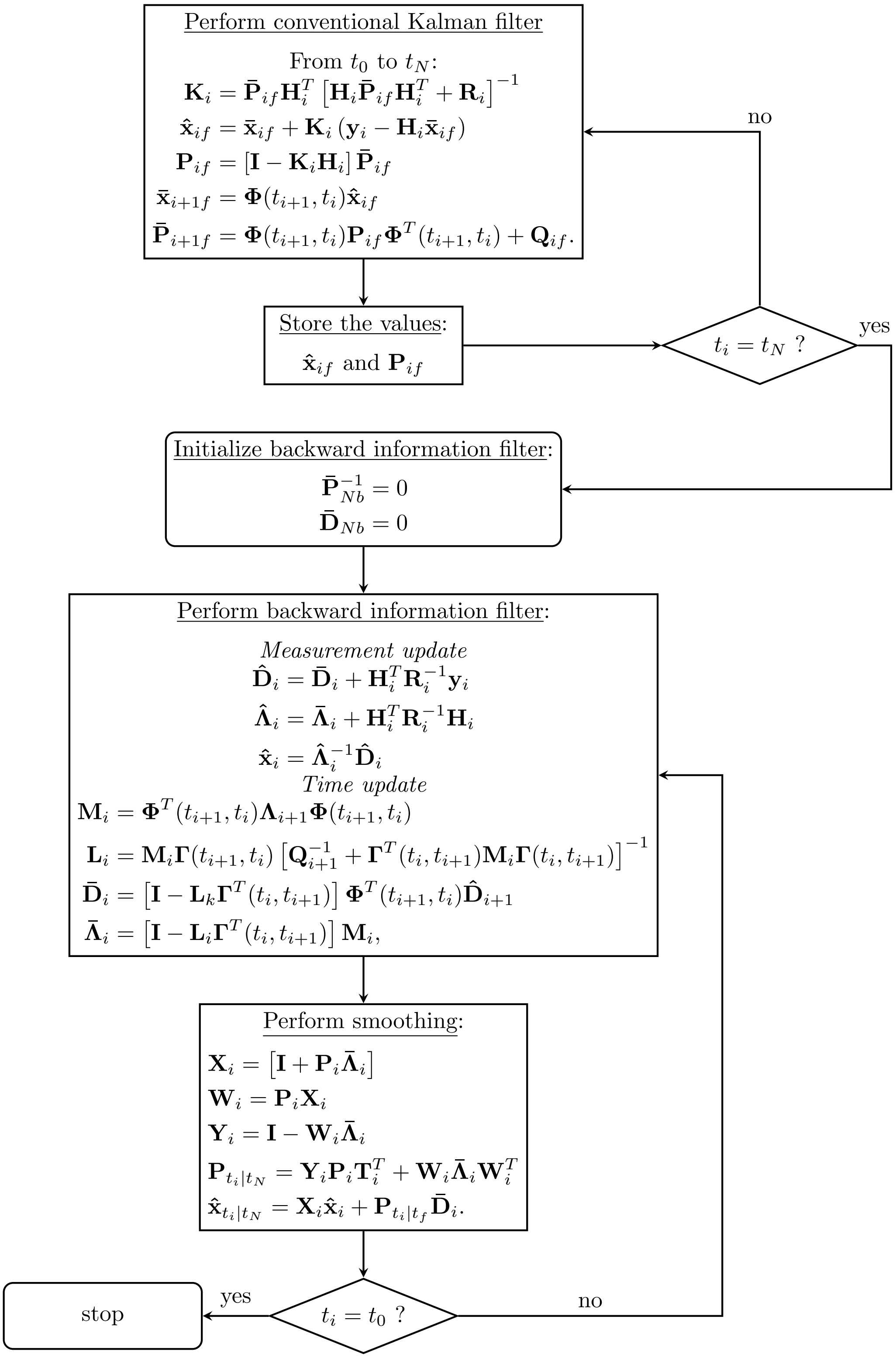
Perform conventional Kalman filter
From t_0 to t_N:
$\mathbf{K}_i = \bar{\mathbf{P}}_{if}\mathbf{H}_i^T\left[\mathbf{H}_i\bar{\mathbf{P}}_{if}\mathbf{H}_i^T + \mathbf{R}_i\right]^{-1}$
$\hat{\mathbf{x}}_{if} = \bar{\mathbf{x}}_{if} + \mathbf{K}_i\left(\mathbf{y}_i - \mathbf{H}_i\bar{\mathbf{x}}_{if}\right)$
$\mathbf{P}_{if} = \left[\mathbf{I} - \mathbf{K}_i\mathbf{H}_i\right]\bar{\mathbf{P}}_{if}$
$\bar{\mathbf{x}}_{i+1f} = \mathbf{\Phi}(t_{i+1}, t_i)\hat{\mathbf{x}}_{if}$
$\bar{\mathbf{P}}_{i+1f} = \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_{if}\mathbf{\Phi}^T(t_{i+1}, t_i) + \mathbf{Q}_{if}.$
no
Store the values:
$\hat{\mathbf{x}}_{if}$ and $\mathbf{P}_{if}$
$t_i = t_N$?
yes
Initialize backward information filter:
$\bar{\mathbf{P}}_{Nb}^{-1} = 0$
$\bar{\mathbf{D}}_{Nb} = 0$
Perform backward information filter:
Measurement update
$\hat{\mathbf{D}}_i = \bar{\mathbf{D}}_i + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{y}_i$
$\hat{\mathbf{\Lambda}}_i = \bar{\mathbf{\Lambda}}_i + \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i$
$\hat{\mathbf{x}}_i = \hat{\mathbf{\Lambda}}_i^{-1}\hat{\mathbf{D}}_i$
Time update
$\mathbf{M}_i = \mathbf{\Phi}^T(t_{i+1}, t_i)\mathbf{\Lambda}_{i+1}\mathbf{\Phi}(t_{i+1}, t_i)$
$\mathbf{L}_i = \mathbf{M}_i\mathbf{\Gamma}(t_{i+1}, t_i)\left[\mathbf{Q}_{i+1}^{-1} + \mathbf{\Gamma}^T(t_i, t_{i+1})\mathbf{M}_i\mathbf{\Gamma}(t_i, t_{i+1})\right]^{-1}$
$\bar{\mathbf{D}}_i = \left[\mathbf{I} - \mathbf{L}_k\mathbf{\Gamma}^T(t_i, t_{i+1})\right]\mathbf{\Phi}^T(t_{i+1}, t_i)\hat{\mathbf{D}}_{i+1}$
$\bar{\mathbf{\Lambda}}_i = \left[\mathbf{I} - \mathbf{L}_i\mathbf{\Gamma}^T(t_i, t_{i+1})\right]\mathbf{M}_i,$
Perform smoothing:
$\mathbf{X}_i = \left[\mathbf{I} + \mathbf{P}_i\bar{\mathbf{\Lambda}}_i\right]$
$\mathbf{W}_i = \mathbf{P}_i\mathbf{X}_i$
$\mathbf{Y}_i = \mathbf{I} - \mathbf{W}_i\bar{\mathbf{\Lambda}}_i$
$\mathbf{P}_{t_i|t_N} = \mathbf{Y}_i\mathbf{P}_i\mathbf{T}_i^T + \mathbf{W}_i\bar{\mathbf{\Lambda}}_i\mathbf{W}_i^T$
$\hat{\mathbf{x}}_{t_i|t_N} = \mathbf{X}_i\hat{\mathbf{x}}_i + \mathbf{P}_{t_i|t_f}\bar{\mathbf{D}}_i.$
stop
yes
$t_i = t_0$?
no

Figure 6.1.1: Flow chart for the forward-backward smoother algorithm.

But

$$p(\mathbf{y}_{k+1}, \ldots, \mathbf{Y}_N | \mathbf{x}_k \mathbf{x}_{k+1}, \mathbf{Y}_k) = p(\mathbf{y}_{k+1}, \ldots, \mathbf{y}_N | \mathbf{x}_{k+1}), \tag{6.2.4}$$

because measurements from time t_{k+1} onwards are independent from all previous measurements $\mathbf{Y}_k$ and $\mathbf{x}_k$, also

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{Y}_k) &= \frac{p(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{Y}_k)}{p(\mathbf{Y}_k)} \\ &= \frac{p(\mathbf{x}_{k+1} | \mathbf{x}_k, \mathbf{Y}_k)}{p(\mathbf{Y}_k)} p(\mathbf{x}_k, \mathbf{Y}_k) \\ &= p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_k), \end{aligned} \tag{6.2.5}$$

and using (6.2.4) and (6.2.5) into (6.2.3) gives

$$p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{Y}_N) = \frac{p(\mathbf{Y}_k)}{p(\mathbf{Y}_N)} p(\mathbf{y}_{k+1}, \ldots, \mathbf{y}_N | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Y}_k). \tag{6.2.6}$$

Note that in equation (6.2.6) only the last two pdf in the RHS are dependent of $\mathbf{x}_k$, thus in the maximization process we are concerned only with $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{x}_k|\mathbf{Y}_k)$. Under the hypothesis that these probability density functions are Gaussian and the process noise is zero mean, the conditional distributions are completely characterized by their mean and variance-covariance, namely

$$p(\mathbf{x}_{k+1} | \mathbf{x}_k) \sim N\left(\mathbf{\Phi}(t_{k+1}, t_k)\mathbf{x}_k, \mathbf{\Gamma}(t_{k+1,t_k})\mathbf{Q}_k \mathbf{\Gamma}^T(t_{k+1}, t_k)\right) \tag{6.2.7}$$

$$p(\mathbf{x}_k | \mathbf{Y}_k) \sim N\left(\hat{\mathbf{x}}_k, \mathbf{P}_k\right), \tag{6.2.8}$$

and notice how the covariance in (6.2.8) does not contain the term $\mathbf{\Phi}(t_{k+1}, t_k)\mathbf{P}_k\mathbf{\Phi}^T(t_{k+1}, t_k)$ since $\mathbf{P}_k$ is associated to errors in the estimate $\hat{\mathbf{x}}_k$, but here in terms of conditional probability, $\mathbf{x}_k$ has occurred and it's known, so its covariance must be a null matrix. In other words, errors in $\mathbf{x}_{k+1}$ are due to process noise only.

Now that the normal distributions of interest are completely defined, we can proceed to maximize the probability density function (6.2.6) by maximizing its logarithm, since the logarithm is a strictly increasing function. Recalling that for a random vector $\mathbf{v}$ of dimension x with associated $n \times n$ variance-covariance matrix $\mathbf{P}$ and mean vector $\bar{\mathbf{x}}$, its *multivariate normal probability density function* is given by

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{P}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}})}, \tag{6.2.9}$$

with $|\mathbf{P}|$ indicating the determinant of $\mathbf{P}$, taking the logarithm of $p(\mathbf{x})$ gives

$$\ln p(\mathbf{x}) = -\frac{1}{2} \ln\left((2\pi)^n |\mathbf{P}|\right) - \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{P}^{-1} (\mathbf{x} - \bar{\mathbf{x}}). \tag{6.2.10}$$

Now, as mentioned before, to maximize (6.2.6) we need to consider only the terms that depends on $\mathbf{x}_k$, so we need to maximize the probability density function

$$L = p(\mathbf{x}_{k+1}|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Y}_k), \tag{6.2.11}$$

taking the logarithm of L following equations (6.2.7), (6.2.8) and (6.2.10) leads to

$$
\begin{aligned}
\ln L = -\frac{1}{2}\ln\left\{(2\pi)^n|\mathbf{\Gamma}(t_{k+1},t_k)\mathbf{Q}_k\mathbf{\Gamma}(t_{k+1},t_k)|\right\}+ \\
-\frac{1}{2}\left(\mathbf{x}_{k+1}-\mathbf{\Phi}(t_{k+1},t_k)\mathbf{x}_k\right)^T\mathbf{\Gamma}(t_{k+1},t_k)\mathbf{Q}_k\mathbf{\Gamma}^T(t_{k+1},t_k)\left(\mathbf{x}_{k+1}-\mathbf{\Phi}(t_{k+1},t_k)\mathbf{x}_k\right)+ \\
-\frac{1}{2}\ln\left\{(2\pi)^n|\mathbf{P}_k|\right\}-\frac{1}{2}(\mathbf{x}_k-\hat{\mathbf{x}}_k)^T\mathbf{P}_k^{-1}(\mathbf{x}_k-\hat{\mathbf{x}}_k),
\end{aligned}
\quad (6.2.12)
$$

then, differentiating with respect to $\mathbf{x}_k$ ignoring terms that are independent of $\mathbf{x}_k$, and assuming that $\hat{\mathbf{x}}_{k+1}^N$ is known from previous backward recursion (time identifiers for $\mathbf{\Phi}(t_{k+1},t_k)$ and $\mathbf{\Gamma}(t_{k+1},t_k)$ are dropped for the sake of compactness)

$$
\begin{aligned}
\frac{\partial\ln L}{\partial\mathbf{x}_k} &= \frac{\partial}{\partial\mathbf{x}_k}\left[\mathbf{x}_k^T\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\mathbf{\Phi}\mathbf{x}_k-2\mathbf{x}_k^T\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\mathbf{x}_{k+1}+\mathbf{x}_k^T\mathbf{P}_k^{-1}\mathbf{x}_k-2\mathbf{x}_k^T\mathbf{P}_k^{-1}\hat{\mathbf{x}}_k\right] \\
&= 2\mathbf{x}_k^T\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\mathbf{\Phi}-2\mathbf{x}_{k+1}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}+2\mathbf{x}_k^T\mathbf{P}_k^{-1}-2\hat{\mathbf{x}}_k^T\mathbf{P}_k^{-1},
\end{aligned}
\quad (6.2.13)
$$

finally, using proper notation and setting (6.2.13) equal to zero performing appropriate transpositions,

$$
\left[\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\mathbf{\Phi}+(\mathbf{P}_k^k)^{-1}\right]\mathbf{x}_k=\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^{\mathbf{T}})^{-1}\hat{\mathbf{x}}_{k+1}^N+(\mathbf{P}_k^k)^{-1}\hat{\mathbf{x}}_k^k, \quad (6.2.14)
$$

gives the following expression for the smoothed estimate $\hat{\mathbf{x}}_k^N$

$$
\hat{\mathbf{x}}_k^N=\left[\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\mathbf{\Phi}+(\mathbf{P}_k^k)^{-1}\right]^{-1}\left[\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\hat{\mathbf{x}}_{k+1}^N+(\mathbf{P}_k^k)^{-1}\hat{\mathbf{x}}_k^k\right]. \quad (6.2.15)
$$

However, the expression obtained above is not quite common. A more conventional form can be obtained using the following matrix identity for the inverse in (6.2.15)

$$
(\mathbf{A}+\mathbf{B}\mathbf{C}\mathbf{D})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}, \quad (6.2.16)
$$

with

$$
\begin{aligned}
\mathbf{A} &= (\mathbf{P}_k^k)^{-1} \\
\mathbf{B} &= \mathbf{\Phi}^T \\
\mathbf{C} &= (\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1} \\
\mathbf{D} &= \mathbf{\Phi},
\end{aligned}
$$

thus obtaining

$$
\hat{\mathbf{x}}_k^N=\left[\mathbf{P}_k^k-\mathbf{P}_k^k\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T+\mathbf{\Phi}\mathbf{P}_k^k\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{P}_k^k\right]\left[\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\hat{\mathbf{x}}_{k+1}^N+(\mathbf{P}_k^k)^{-1}\hat{\mathbf{x}}_k^k\right], \quad (6.2.17)
$$

recognising that

$$
(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T+\mathbf{\Phi}\mathbf{P}_k^k\mathbf{\Phi}^T)=\mathbf{P}_{k+1}^k \quad (6.2.18)
$$

is the propagation of the estimation error covariance matrix $\mathbf{P}_k^k$ up to time t_{k+1} under the influence of process noise, that is the a priori covariance at t_{k+1}, and performing multiplications we can write

$$
\begin{aligned}
\hat{\mathbf{x}}_k^N =&\hat{\mathbf{x}}_k^k-\mathbf{P}_k^k\mathbf{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\mathbf{\Phi}\hat{\mathbf{x}}_k^k+\mathbf{P}_k^k\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\hat{\mathbf{x}}_{k+1}^N+ \\
&-\mathbf{P}_k^k\mathbf{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\mathbf{\Phi}\mathbf{P}_k^k\mathbf{\Phi}^T(\mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T)^{-1}\hat{\mathbf{x}}_{k+1}^N,
\end{aligned}
\quad (6.2.19)
$$

or

$$\begin{aligned}\hat{\mathbf{x}}_k^N =&\hat{\mathbf{x}}_k^k - \mathbf{P}_k^k\boldsymbol{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\boldsymbol{\Phi}\hat{\mathbf{x}}_k^k + \\ &+ \mathbf{P}_k^k\boldsymbol{\Phi}^T\left[(\boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T)^{-1} - (\mathbf{P}_{k+1}^k)^{-1}\boldsymbol{\Phi}\mathbf{P}_k^k\boldsymbol{\Phi}^T(\boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T)^{-1}\right]\hat{\mathbf{x}}_{k+1}^N.\end{aligned} \tag{6.2.20}$$

Now, the term in brackets in the RHS of the above equation can be rewritten considering that from

$$\mathbf{P}_{k+1}^k = \boldsymbol{\Phi}(t_{k+1},t_k)\mathbf{P}_k^k\boldsymbol{\Phi}^T(t_{k+1},t_k) + \boldsymbol{\Gamma}(t_{k+1},t_k)\mathbf{Q}_k\boldsymbol{\Gamma}^T(t_{k+1},t_k) \tag{6.2.21}$$

we can obtain

$$\mathbf{P}_k^k = \boldsymbol{\Phi}^{-1}(t_{k+1},t_k)\left[\mathbf{P}_{k+1}^k - \boldsymbol{\Gamma}(t_{k+1},t_k)\mathbf{Q}_k\boldsymbol{\Gamma}^T(t_{k+1},t_k)\right]\boldsymbol{\Phi}^{-T}(t_{k+1},t_k), \tag{6.2.22}$$

so, using (6.2.22) inside the square brackets in equation (6.2.20) gives

$$\begin{aligned}\hat{\mathbf{x}}_k^N &= \hat{\mathbf{x}}_k^k - \mathbf{P}_k^k\boldsymbol{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\boldsymbol{\Phi}\hat{\mathbf{x}}_k^k + \\ &+ \mathbf{P}_k^k\boldsymbol{\Phi}^T\left[(\boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T)^{-1} - (\mathbf{P}_{k+1}^k)^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{-1}(\mathbf{P}_{k+1}^k - \boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T)\boldsymbol{\Phi}^{-T}\boldsymbol{\Phi}^T(\boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T)^{-1}\right]\hat{\mathbf{x}}_{k+1}^N,\end{aligned} \tag{6.2.23}$$

and performing multiplications leads to

$$\hat{\mathbf{x}}_k^N = \hat{\mathbf{x}}_k^k - \mathbf{P}_k^k\boldsymbol{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\boldsymbol{\Phi}\hat{\mathbf{x}}_k^k + \mathbf{P}_k^k\boldsymbol{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}\hat{\mathbf{x}}_{k+1}^N, \tag{6.2.24}$$

finally, collecting terms we arrive at the conventional *RTS smoother* algorithm

$$\hat{\mathbf{x}}_k^N = \hat{\mathbf{x}}_k^k + \mathbf{S}_k\left[\hat{\mathbf{x}}_{k+1}^N - \boldsymbol{\Phi}\hat{\mathbf{x}}_k^k\right], \tag{6.2.25}$$

with

$$\mathbf{S}_k = \mathbf{P}_k^k\boldsymbol{\Phi}^T(\mathbf{P}_{k+1}^k)^{-1}, \tag{6.2.26}$$

being the *smoothing estimator gain matrix.*

Equation (6.2.25) clearly shows how the RTS algorithm is a backward recursion since every smoothed estimate at t_k is based on the knowledge of the future estimate and covariance at t_{k+1}, and the starting value is the final value obtained at the end of the forward sweep, confirming the fact that the smoothed solution is generally better than the filtered solution, except for the final time of the fixed interval.

For the propagation of the smoothed covariance we begin by showing how, in a general sense, it can be computed as

$$\mathbf{P}_k^N = E\left[(\hat{\mathbf{x}}_k^N - \mathbf{x}_k)(\hat{\mathbf{x}}_k^N - \mathbf{x}_k)^T\right], \tag{6.2.27}$$

where $\mathbf{x}_k$ represents actual state and $\hat{\mathbf{x}}_k^N$ is the smoothed estimate of $\mathbf{x}_k$. Now, taking equation (6.2.25) and subtracting $\mathbf{x}_k$ from both sides yields:

$$\hat{\mathbf{x}}_k^N - \mathbf{x}_k = \hat{\mathbf{x}}_k^k - \mathbf{x}_k + \mathbf{S}_k\hat{\mathbf{x}}_{k+1}^N - \mathbf{S}_k\boldsymbol{\Phi}\hat{\mathbf{x}}_k^k, \tag{6.2.28}$$

then, rearranging terms and using the definitions

$$\begin{aligned}\tilde{\mathbf{x}}_k^N &\equiv \hat{\mathbf{x}}_k^N - \mathbf{x}_k \\ \tilde{\mathbf{x}}_k^k &\equiv \hat{\mathbf{x}}_k^k - \mathbf{x}_k,\end{aligned}$$

gives

$$\tilde{\mathbf{x}}_k^N - \mathbf{S}_k\hat{\mathbf{x}}_{k+1}^N = \tilde{\mathbf{x}}_k^k - \mathbf{S}_k\mathbf{\Phi}\hat{\mathbf{x}}_k^k. \tag{6.2.29}$$

In order to write everything in terms of covariances we multiply both sides of equation (6.2.29) by the respective transposes and take expected values:

$$E\left[(\tilde{\mathbf{x}}_k^N - \mathbf{S}_k\hat{\mathbf{x}}_{k+1}^N)(\tilde{\mathbf{x}}_k^N - \mathbf{S}_k\hat{\mathbf{x}}_{k+1}^N)^T\right] = E\left[(\tilde{\mathbf{x}}_k^k - \mathbf{S}_k\mathbf{\Phi}\hat{\mathbf{x}}_k^k)(\tilde{\mathbf{x}}_k^k - \mathbf{S}_k\mathbf{\Phi}\hat{\mathbf{x}}_k^k)^T\right], \tag{6.2.30}$$

and for more clarity in the passages, we will address the two expectations above one by one. Let's start by considering the LHS of the above equation that can be expanded as:

$$E\left[\tilde{\mathbf{x}}_k^N\tilde{\mathbf{x}}_k^{N^T}\right] - \mathbf{S}_k E\left[\hat{\mathbf{x}}_{k+1}^N\tilde{\mathbf{x}}_k^{N^T}\right] - E\left[\tilde{\mathbf{x}}_k^N\hat{\mathbf{x}}_{k+1}^{N^T}\right]\mathbf{S}_k^T + \mathbf{S}_k E\left[\hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T}\right]\mathbf{S}_k^T, \tag{6.2.31}$$

here, the firs expectation is clearly the smoothed covariance at time t_k

$$E\left[\tilde{\mathbf{x}}_k^N\tilde{\mathbf{x}}_k^{N^T}\right] = \mathbf{P}_k^N, \tag{6.2.32}$$

while for the second and third expectations coming from the cross product

$$\begin{aligned} E\left[\hat{\mathbf{x}}_{k+1}^N\tilde{\mathbf{x}}_k^{N^T}\right] &= \hat{\mathbf{x}}_{k+1}^N E\left[\hat{\mathbf{x}}_k^N - \mathbf{x}_k\right] \\ &= \hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_k^N - \hat{\mathbf{x}}_{k+1}^N E\left[\mathbf{x}_k|\mathbf{Y}_N\right] \\ &= \hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_k^N - \hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_k^N = 0 \end{aligned} \tag{6.2.33}$$

where we have considered the fact that $\hat{\mathbf{x}}$ is not a random quantity, but it expresses a conditional mean coming from properly assumed probability density functions (see (6.2.7) and (6.2.8)), and the expectation is conditioned on the knowledge of all measurements up to time t_N, that is $\mathbf{Y}_N$. Accordingly, the fourth expectation gives

$$E\left[\hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T}\right] = \hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T}, \tag{6.2.34}$$

however, it is more useful for later to consider the following expression:

$$\begin{aligned} \mathbf{P}_{k+1}^N &= E\left[(\hat{\mathbf{x}}_{k+1}^N - \mathbf{x}_{k+1})(\hat{\mathbf{x}}_{k+1}^N - \mathbf{x}_{k+1})^T\right] \\ &= \hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T} - \hat{\mathbf{x}}_{k+1}^N E\left[\mathbf{x}_{k+1}^T|\mathbf{Y}_N\right] - E\left[\mathbf{x}_{k+1}|\mathbf{Y}_N\right]\hat{\mathbf{x}}_{k+1}^{N^T} + E\left[\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T\right] \\ &= -\hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T} + E\left[\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T\right] \end{aligned} \tag{6.2.35}$$

so that

$$\hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T} = E\left[\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T\right] - \mathbf{P}_{k+1}^N. \tag{6.2.36}$$

Moreover, considering that $\mathbf{x}_{k+1}$ is the propagation of the actual state $\mathbf{x}_k$ to time t_{k+1} under the influence of process noise $\mathbf{u}_k$, we can write

$$\mathbf{x}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{x}_k + \mathbf{\Gamma}(t_{k+1}, t_k)\mathbf{u}_k, \tag{6.2.37}$$

and substituting in equation (6.2.36) yields

$$\hat{\mathbf{x}}_{k+1}^N\hat{\mathbf{x}}_{k+1}^{N^T} = \mathbf{\Phi}E\left[\mathbf{x}_k\mathbf{x}_k^T\right]\mathbf{\Phi}^T + \mathbf{\Gamma}\mathbf{Q}_k\mathbf{\Gamma}^T - \mathbf{P}_{k+1}^N. \tag{6.2.38}$$

Now that the first expectation in equation (6.2.30) has been completely expanded, it is time to perform similar passages for the expectation in the right hand side, so we begin by performing multiplications:

$$E\left[\tilde{\mathbf{x}}_k^k\tilde{\mathbf{x}}_k^{k^T}\right] - \mathbf{S}_k\boldsymbol{\Phi}E\left[\hat{\mathbf{x}}_k^k\tilde{\mathbf{x}}_k^{k^T}\right] - E\left[\tilde{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T}\right]\boldsymbol{\Phi}^T\mathbf{S}_k^T + \mathbf{S}_k\boldsymbol{\Phi}E\left[\hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T}\right]\boldsymbol{\Phi}^T\mathbf{S}_k^T, \tag{6.2.39}$$

then, for the first expectation we have

$$E\left[\tilde{\mathbf{x}}_k^k\tilde{\mathbf{x}}_k^{k^T}\right] = \mathbf{P}_k^k, \tag{6.2.40}$$

the second and third expectations are null, similarly to equation (6.2.33)

$$\begin{aligned} E\left[\hat{\mathbf{x}}_k^k\tilde{\mathbf{x}}_k^{k^T}\right] &= \hat{\mathbf{x}}_k^k E\left[(\hat{\mathbf{x}}_k^k - \mathbf{x}_k)^T\right] \\ &= \hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} - \hat{\mathbf{x}}_k^k E\left[\mathbf{x}_k^T|\mathbf{Y}_k\right] \\ &= \hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} - \hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} = 0 \end{aligned} \tag{6.2.41}$$

where again $\hat{\mathbf{x}}$ is not a random quantity but it is a conditional mean, and the expectations are conditioned on thre knowledge of all measurements up to time t_k, that is $\mathbf{Y}_k$. The last expectation in (6.2.40) is

$$E\left[\hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T}\right] = \hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} \tag{6.2.42}$$

but again, we should consider a more convenient expression for later computations given by

$$\begin{aligned} \mathbf{P}_k^k &= E\left[(\hat{\mathbf{x}}_k^k - \mathbf{x}_k)(\hat{\mathbf{x}}_k^k - \mathbf{x}_k)^T\right] \\ &= \hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} - \hat{\mathbf{x}}_k^k E\left[\mathbf{x}_k^T|\mathbf{Y}_k\right] - E\left[\mathbf{x}_k|\mathbf{Y}_k\right]\hat{\mathbf{x}}_k^{k^T} + E\left[\mathbf{x}_k\mathbf{x}_k^T\right] \\ &= -\hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} + E\left[\mathbf{x}_k\mathbf{x}_k^T\right], \end{aligned} \tag{6.2.43}$$

from which we get

$$\hat{\mathbf{x}}_k^k\hat{\mathbf{x}}_k^{k^T} = E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \mathbf{P}_k^k. \tag{6.2.44}$$

At this point, using equations (6.2.44) and (6.2.40) inside (6.2.39), and equations (6.2.38) and (6.2.32) inside (6.2.31) to rewrite equation (6.2.30), we arrive at

$$\begin{aligned} &\mathbf{P}_k^N + \mathbf{S}_k\left(\boldsymbol{\Phi}E\left[\mathbf{x}_k\mathbf{x}_k^T\right]\boldsymbol{\Phi}^{\mathbf{T}} + \boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T - \mathbf{P}_{k+1}^N\right)\mathbf{S}_k^T = \mathbf{P}_k^k + \mathbf{S}_k\boldsymbol{\Phi}\left(E\left[\mathbf{x}_k\mathbf{x}_k^T\right] - \mathbf{P}_k^k\right)\boldsymbol{\Phi}^T\mathbf{S}_k^T, \\ &\mathbf{P}_k^N + \mathbf{S}_k\left[\boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T - \mathbf{P}_{k+1}^k\right]\mathbf{S}_k^T = \mathbf{P}_k^k - \mathbf{S}_k\boldsymbol{\Phi}\mathbf{P}_k^k\boldsymbol{\Phi}^T\mathbf{S}_k^T, \\ &\mathbf{P}_k^N = \mathbf{P}_k^k + \mathbf{S}_k\left[\mathbf{P}_{k+1}^N - \boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T - \boldsymbol{\Phi}\mathbf{P}_k^k\boldsymbol{\Phi}^T\right]\mathbf{S}_k^T, \end{aligned} \tag{6.2.45}$$

and recognising that the square brackets contain $\mathbf{P}_{k+1}^k$ given that

$$\boldsymbol{\Phi}\mathbf{P}_k^k\boldsymbol{\Phi}^T + \boldsymbol{\Gamma}\mathbf{Q}_k\boldsymbol{\Gamma}^T = \mathbf{P}_{k+1}^k, \tag{6.2.46}$$

finally yields:

$$\mathbf{P}_k^N = \mathbf{P}_k^k + \mathbf{S}_k\left[\mathbf{P}_{k+1}^N - \mathbf{P}_{k+1}^k\right]\mathbf{S}_k^T. \tag{6.2.47}$$

Equations (6.2.47) and (6.2.25) are used together in the backward recursion to compute the smoothed estimates and covariances at every observation time t_k.
One interesting feature of the RTS smoother algorithm is the fact that unlike the forward-backward smoother, during the backward phase, no information from measurements seems to appear, which is also suggested by the absence of any observation-state mapping matrix $\mathbf{H}_i$. Nevertheless, this absence of information is only apparent, since it is implicitly contained is the probability density functions (6.2.2) - (6.2.6) and the backward RTS recursive algorithm is a proper combination

of *a priori* and *a posteriori* information from the forward filter, which clearly, already contains information from the entire set of measurements. On the other hand, the backward phase of the forward-backward smoother, explicitly processes information from measurements one at a time just like the forward filter, in fact, here the smoother is a proper combination of the results from two independent filters rather than a recursive combination of results from one forward filter.

6.2.1 RTS Smoother Algorithm

The RTS smoother algorithm begins with a conventional Kalman filter that processes all observations from t_0 to t_N. Thus, the equations for this phase remain the same as the ones described in (6.1.25) up to (6.1.29), however, using current notation where the subscript indicates the time of interest and the superscript indicates the conditioning measurements, we can rewrite them as

$$\mathbf{K}_i = \mathbf{P}_i^{(i-1)}\mathbf{H}_i^T\left[\mathbf{H}_i\mathbf{P}_i^{(i-1)}\mathbf{H}_i^T + \mathbf{R}_i\right]^{-1} \tag{6.2.48}$$

$$\hat{\mathbf{x}}_i^i = \hat{\mathbf{x}}_i^{i-1} + \mathbf{K}_i\left[\mathbf{y}_i - \mathbf{H}_i\hat{\mathbf{x}}_i^{(i-1)}\right] \tag{6.2.49}$$

$$\mathbf{P}_i^i = \left[\mathbf{I} - \mathbf{K}_i\mathbf{H}_i\right]\mathbf{P}_i^{(i-1)} \tag{6.2.50}$$

$$\hat{\mathbf{x}}_{i+1}^i = \mathbf{\Phi}(t_{i+1}, t_i)\hat{\mathbf{x}}_i^i \tag{6.2.51}$$

$$\mathbf{P}_{i+1}^i = \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_i^i\mathbf{\Phi}^T(t_{i+1}, t_i), \tag{6.2.52}$$

and now, the quantities $\hat{\mathbf{x}}_{i+1}^i$, $\mathbf{P}_{i+1}^i$, $\hat{\mathbf{x}}_i^i$ and $\mathbf{P}_i^i$ are stored. As previously discussed, the forward filter ends at time t_N giving the following a posteriori values after incorporating measurements at the final time,

$$\begin{gathered}\hat{\mathbf{x}}_N^N \\ \mathbf{P}_N^N,\end{gathered} \tag{6.2.53}$$

and this values are used as the starting conditions for the backward smoothing from t_N to t_0, that is performed as

$$\hat{\mathbf{x}}_i^N = \hat{\mathbf{x}}_i^i + \mathbf{S}_i\left[\hat{\mathbf{x}}_{i+1}^N - \mathbf{\Phi}\hat{\mathbf{x}}_i^i\right], \tag{6.2.54}$$

$$\mathbf{P}_i^N = \mathbf{P}_i^i + \mathbf{S}_i\left[\mathbf{P}_{i+1}^N - \mathbf{P}_{i+1}^i\right]\mathbf{S}_i^T. \tag{6.2.55}$$

where the *smoothing estimate gain matrix* $\mathbf{S}_i$ is

$$\mathbf{S}_i = \mathbf{P}_i^i\mathbf{\Phi}^T(t_{i+1}, t_i)\left(\mathbf{P}_{i+1}^i\right)^{-1}. \tag{6.2.56}$$

One relevant aspect on smoothing that emerge from equation (6.2.56) is the fact that in the absence of process noise, the smoothed estimate will be the same as the filtered one. In fact, we can rewrite $\mathbf{S}_i$ considering that

$$\mathbf{P}_i^i = \mathbf{\Phi}^{-1}(t_{i+1}, t_i)\left[\mathbf{P}_{i+1}^i - \mathbf{\Gamma}(t_{i+1}.i)\mathbf{Q}_i\mathbf{\Gamma}^T(t_{i+1}, i)\right]\mathbf{\Phi}^{-T}(t_{i+1}, t_i) \tag{6.2.57}$$

and substituting (6.2.57) into (6.2.56) gives

$$\mathbf{S}_i = \mathbf{\Phi}^{-1}(t_{i+1}, t_i)\left[\mathbf{I} - \mathbf{Q}_i\bar{\mathbf{P}}_{i+1}^{-1}\right], \tag{6.2.58}$$

which, for the case of $\mathbf{Q} = 0$ becomes

$$\mathbf{S} = \mathbf{\Phi}^{-1}(t_{i+1}, t_i), \tag{6.2.59}$$

leading to

$$\begin{aligned}\hat{\mathbf{x}}_i^N &= \mathbf{\Phi}^{-1}(t_{i+1}, t_i)\hat{\mathbf{x}}_{i+1}^N \\ &= \mathbf{\Phi}(t_k, t_{k+1})\hat{\mathbf{x}}_{k+1}^N,\end{aligned} \tag{6.2.60}$$

and

$$\begin{aligned}\mathbf{P}_i^N &= \mathbf{\Phi}^{-1}(t_{i+1}, t_i)\mathbf{P}_{i+1}^N\mathbf{\Phi}^{-T}(t_{i+1}, t_i) \\ &= \mathbf{\Phi}(t_i, t_{i+1})\mathbf{P}_{i+1}^N\mathbf{\Phi}^T(t_i, t_{i+1}),\end{aligned} \tag{6.2.61}$$

where we have used the property of the state transition matrix for which $\mathbf{\Phi}^{-1}(t_i, t_j) = \mathbf{\Phi}(t_j, t_i)$. Equations (6.2.60) and (6.2.61) show that in the absence of process noise, smoothing simply becomes backward propagation, and while additional measurements are still helpful to get better estimates of constant states, there is no point in using smoothing techniques in the absence of random components. This result can be extended to the case in which the state vector is composed of dynamical components affected by process noise and *bias* components that are not affected by process noise, here, the latter will not be smoothable. Moreover, the dynamical components in the state vector may not all be directly affected by process noise. For example one may introduce noise only in the acceleration component, meaning that velocity and position are still under the influence of noise but separated by integration, and in general the more a component is directly affected by noise the more it is smoothable. An important consequence of these considerations is the fact that in the absence of process noise, meaning that there is no random component in the dynamical model used for propagation, the batch estimate will actually give the smoothed solution. However, as noted before, the batch processor gives rise to important difficulties in trying to include the effects of process noise, and so, if one wish to consider its effects a smoothing algorithm is the right choice to overcome this trouble.

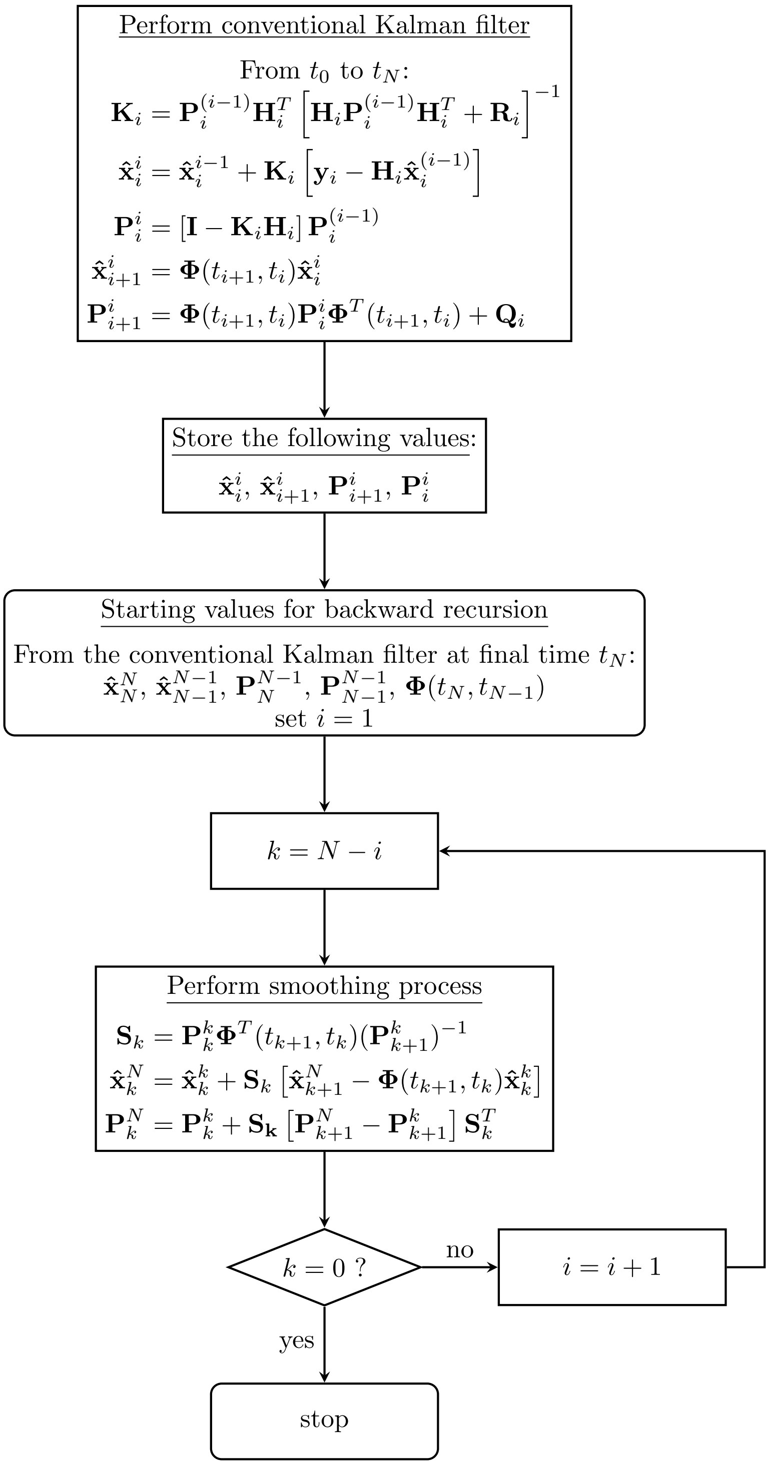
Perform conventional Kalman filter
From t_0 to t_N:
$\mathbf{K}_i = \mathbf{P}_i^{(i-1)}\mathbf{H}_i^T\left[\mathbf{H}_i\mathbf{P}_i^{(i-1)}\mathbf{H}_i^T + \mathbf{R}_i\right]^{-1}$
$\hat{\mathbf{x}}_i^i = \hat{\mathbf{x}}_i^{i-1} + \mathbf{K}_i\left[\mathbf{y}_i - \mathbf{H}_i\hat{\mathbf{x}}_i^{(i-1)}\right]$
$\mathbf{P}_i^i = [\mathbf{I} - \mathbf{K}_i\mathbf{H}_i]\,\mathbf{P}_i^{(i-1)}$
$\hat{\mathbf{x}}_{i+1}^i = \mathbf{\Phi}(t_{i+1}, t_i)\hat{\mathbf{x}}_i^i$
$\mathbf{P}_{i+1}^i = \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{P}_i^i\mathbf{\Phi}^T(t_{i+1}, t_i) + \mathbf{Q}_i$
Store the following values:
$\hat{\mathbf{x}}_i^i,\ \hat{\mathbf{x}}_{i+1}^i,\ \mathbf{P}_{i+1}^i,\ \mathbf{P}_i^i$
Starting values for backward recursion
From the conventional Kalman filter at final time t_N:
$\hat{\mathbf{x}}_N^N,\ \hat{\mathbf{x}}_{N-1}^{N-1},\ \mathbf{P}_N^{N-1},\ \mathbf{P}_{N-1}^{N-1},\ \mathbf{\Phi}(t_N, t_{N-1})$
set $i = 1$
$k = N - i$
Perform smoothing process
$\mathbf{S}_k = \mathbf{P}_k^k\mathbf{\Phi}^T(t_{k+1}, t_k)(\mathbf{P}_{k+1}^k)^{-1}$
$\hat{\mathbf{x}}_k^N = \hat{\mathbf{x}}_k^k + \mathbf{S}_k\left[\hat{\mathbf{x}}_{k+1}^N - \mathbf{\Phi}(t_{k+1}, t_k)\hat{\mathbf{x}}_k^k\right]$
$\mathbf{P}_k^N = \mathbf{P}_k^k + \mathbf{S_k}\left[\mathbf{P}_{k+1}^N - \mathbf{P}_{k+1}^k\right]\mathbf{S}_k^T$
$k = 0$?
no
$i = i + 1$
yes
stop

Figure 6.2.1: Flow chart for the RTS smoother algorithm.

Chapter 7

Simulations and Test Cases

In this Chapter, the results of the simulations conducted to test the various POD techniques discussed in this dissertation are presented and analyzed, in particular, all POD algorithms are tested on the Sentinel-3A satellite. The code used to generate these results have been written in Fortran, utilizing a variety of pre-existing libraries and subroutines developed, and made available for this work, by the *Navigation and Space Geodesy Group* led by Professor Casotto of the University of Padova. These libraries contain sophisticated subroutines that are able to model various forces acting on a satellite, as well as managing coordinate and time transformations which are crucial in precise orbit determination applications. Nonetheless, all the routines associated to the filtering techniques discussed in this thesis have been developed independently, along with an Earth radiation pressure force model and other various utilities. The post-processing phase, which involves managing the output data and formatting it into graphs, was instead carried out in MATLAB. As for the pre-processing phase, the official data for the Sentinel-3A satellite used as input for all the simulations are generated at ESOC and provided by ESA. Brief information about the satellite and its representation are given in table 7.1 and figure 7.0.1. These data are in the form of estimated inertial position and velocity every 10 seconds over and arc of 7 days starting from 2017-06-03 ,22:00:51 (TT) or 2 457 908.41725907 (JD). The following Section, after providing general information about Sentinel-3A, discusses the pre-processing phase of generating measurement data, while subsequent Sections delve into a detailed discussion and analysis of the results obtained.

Table 7.1: Sentinel-3A general information.

Item	**Description**
S3A orbit	near-polar sun-synchronous LEO orbit Repeat cycle: 27 days Period: 100.99 min Mean altitude: 814.5 km eccentricity: 0.000309 Inclination: 98.65 deg
S3A spacecraft parameters	Mass: 1128.0 kg CoM position: (1.533 m, -0.007 m, 0.037 m) Reference area Drag: 7 m^2 Reference area Rad: 12 m^2

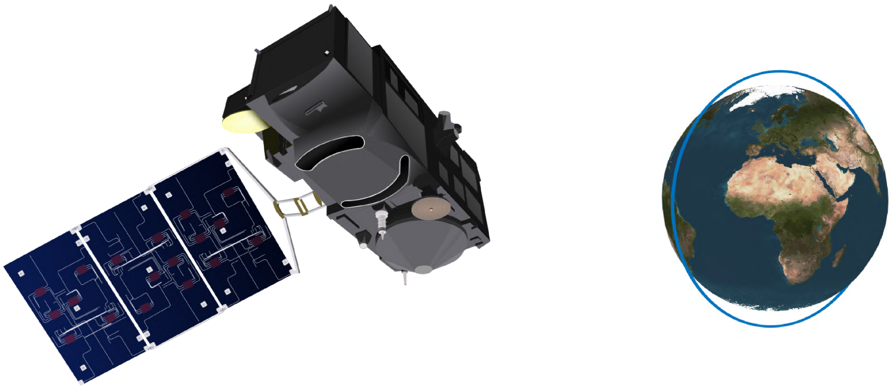

Figure 7.0.1: Sentinel-3A satellite and a representation of its orbit.

7.1 Generating Observations

Every orbit determination algorithm presented requires measurements of some sort in order to estimate the satellite's orbit. The results shown in the following Sections are all based on instantaneous ranges generated from the Sentinel-3A inertial positions provided by ESA, as well as the Earth-fixed coordinates of 50 ground stations available from the International GNSS Service (IGS) [1]. These measurements are computed using the instantaneous range equation (3.3.1) from Section 3.3.1 with the available data and a total of 21 217 ranges are obtained starting from the initial epoch of 2017-06-03 ,22:00:51 (TT) over an arc of 1 day from the 7 available days of information, thus, all simulation results will be presented in terms of *elapsed time* in seconds starting from 0.0 s to 86 400 s. The choice of a 1-day arc is motivated by the fact that Sentinel-3A is a low-Earth satellite that completes an orbit in roughly 100 minutes, or in other terms, orbits the Earth 14 times a day, thus providing an already challenging situation for every estimation algorithm without the need to consider data from the whole 7 days arc, which may noticeably increase the computational cost.

An important aspect when generating ranges, is the satellite's visibility. Clearly, the satellite is not continuously visible from all ground stations at the same time, and especially for LEO satellites, low observability may have a detrimental impact on the estimation results. For this reason the computation of measurements, with a simple spherical Earth model along with purely geometrical considerations on the satellite's visibility cone, has led to a ranges time-history with variable number of instantaneously active ground stations as well as non-observability periods of various extents. Figure 7.1.1 shows the number of visible ground stations over the whole estimation arc of 1 day, and as we can see, the 50 ground stations provide a fairly adequate coverage considering the satellite's altitude. Nonetheless, many periods on non-visibility can be identified in correspondence of the vertical gray bands, the longest of which is around 15 minutes, and on average, roughly 3 stations are in sight, with the most frequent occurrence being 2 visible stations.

Another common practice when testing POD algorithms with synthetic data is to add noise of some level and nature to the computed measurements. Here, a simple white noise approach is followed, where a random sequence with standard deviation of $\sigma = 1$ cm is added to the nominal synthetic ranges. This increases the level of realism, since real measurements are always subject to noise, and it also allows for a greater stress test of the estimation algorithms. It is clear that a more sophisticated and definitive test should handle real measurement data rather than synthetic computed observations. Nevertheless, the scope of this thesis in not the orbit determination of a real satellite itself, but instead the evaluation and comparison of different POD techniques performances, which becomes more clear and straightforward in a controlled environment of synthetic data, circumventing all the difficulties and unforeseen events of real-world dynamics.

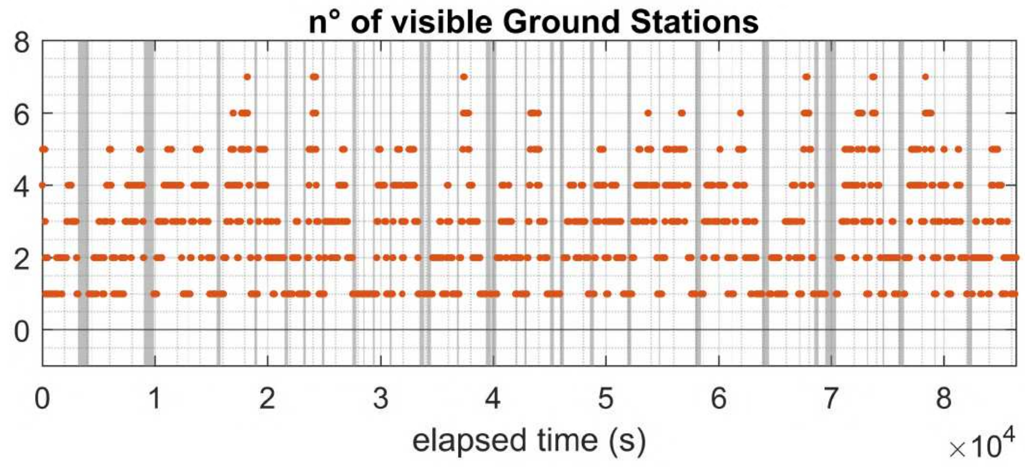

Figure 7.1.1: Number of visible ground stations vs time. As we can see observability is adequately distributed over the estimation arc, with a mode of 2 visible ground stations. The gray vertical bands indicate periods of non-observability.

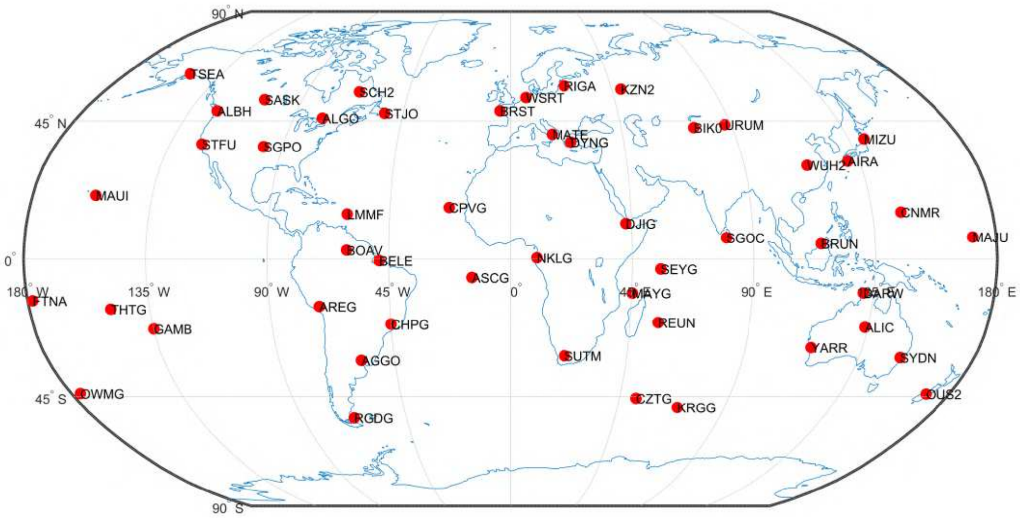

Figure 7.1.2: Locations and identifiers of the 50 ground stations used to generate observations.

7.2 Sentinel-3A Orbit Determination through Batch Processing

The first results we are going to discuss, are related to the batch processing technique which we have detailed in Chapter 4. We recall that the main peculiarity of this approach lies in the fact that measurements are processed after they have been collected during the whole observation arc, hence the name *batch* estimation. As previously stated, the observations are generated in the form of *instantaneous ranges* using the methods described in Section 3.3. These, are computed from an available estimated orbit of Sentinel-3A and the coordinates of fifty Ground Stations distributed over the Earth's surface (Figure 7.1.2). The general computational algorithm used in the Fortran code follows the structure shown in figure 4.3.1, where estimation is performed through non-linear least squares with a priori information in the form of an iterative *differential corrections* process. Here, every iterations computes corrections to state vector at epoch and we recall that the process terminates when convergence is achieved. In general, the initial guess for the state vector at epoch for every estimation process in this Section is perturbed to some extent in order to further stress the estimation algorithm and test its convergence. We now proceed to examine the results obtained for various formulations of the batch processor, where both the box-wing and cannon-ball satellite models are tested, and the discussion of the parameters segmentation procedure is studied as a means of improving the cannon-ball model estimation.

7.2.1 Box-Wing Model

The first results shown in this Section are related to the batch orbit determination using a so called *box-wing* model. In this case, when performing numerical propagation of the reference trajectory, it was possible to account for the satellite's attitude law and its cross-sectional variations over time thanks to the available Sentinel-3A box-wing model and attitude law in the form of Fortran libraries, provided by the Navigation and Space Geodesy Group of Professor Casotto at University of Padova.

When discussing results in this Section we will often refer to *initial values* and *nominal value*, where tables 7.2 and 7.3 provide values for the initial settings of every batch estimation that is discussed here. Note how for position and velocity we have an initial value, that is the one used as the initial guess of the state vector at epoch t_0 and a nominal value, that is the true dynamical state at epoch corresponding to the *true orbit model* from which observations are generated. Moreover, a priori information is represented by $\bar{\mathbf{x}}$ and $\bar{\boldsymbol{\sigma}}$, that is respectively the initial correction guess and its associated uncertainty, then, $\Delta\mathbf{X}$ is the magnitude of the aforementioned perturbation that is added to the nominal value to obtain the initial state with which the reference orbit is propagated. Similar considerations are valid for the parameters table, but here, only an initial value is present and $\bar{\mathbf{x}}$ is set to 0, knowing that this choices are less influential than the position and velocity initialization.

Table 7.2: Dynamical initial conditions and a priori information.

Parameter	**Initial Value**	**Nominal Value**	$\Delta\mathbf{X}$	$\bar{\mathbf{x}}$	$\bar{\boldsymbol{\sigma}}$
X (m)	$-3\,198\,972.3572464$	$-3\,199\,072.3572464$	100	10	1000
Y (m)	$-1\,651\,187.851021800$	$-1\,651\,087.8510218$	-100	-10	1000
Z (m)	$6\,205\,796.0563044$	$6\,205\,596.0563044$	200	20	1000
$\dot{X}$ (m/s)	$4\,459.970115317$	$4\,459.870115317$	0.1	0.01	1
$\dot{Y}$ (m/s)	$4\,790.627269888$	$4\,790.577269888$	0.05	-0.005	1
$\dot{Z}$ (m/s)	$3\,565.497695423$	$3\,565.427695423$	0.07	0.007	1

Table 7.3: Parameters initial values and a priori information.

Parameter	Initial Value	$\bar{\mathbf{x}}$	$\bar{\boldsymbol{\sigma}}$
C_R	1	0	0.5
C_D	2.2	0	0.5
A_{RTN} (m/s^2)	0	0	10^{-10}
D_{RTN_i} (m/s^2)	0	0	10^{-10}
S_{RTN_i} (m/s^2)	0	0	10^{-10}

First of all, we begin by considering the effects of the initial reference state $\mathbf{X}_0^*$ on the first batch iterations in order to assess the convergence capability of the algorithm. As the previous tables shown, the initial state value is perturbed by hundreds of meters in position and centimeter per second level in velocity. This disturbance might not seem so drastic, however, its effects become more appreciable as propagation evolves over time. Figure 7.2.1 shows the trend of the observation residuals (difference between observed and computed ranges) and the orbit residuals (difference between true and estimated orbit) over time for the first bacth iteration. Note how this apparently harmless initial perturbation leads to observation residuals of more than 60 km and orbit residuals of tens of kilometers in radial and transverse direction, which remarkably stresses the algorithm to some extent.

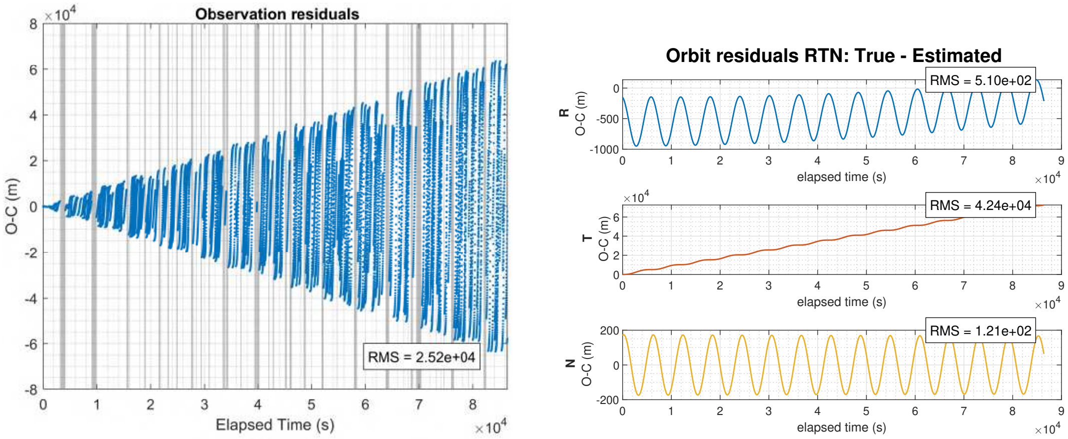

Figure 7.2.1: Ranges residuals (left) and difference between true and estimated orbit (right) at first iteration. Vertical gray bands represent periods of non-observability. Note how a perturbation on initial condition of hundreds of meters in position and few centimeters per second in velocity, lead to tens of kilometers in measurement residuals after one day.

On the other hand, figures 7.2.2 and 7.2.3 show the observation and orbit residuals at convergence. Here we note how the observation residuals have dropped down to few centimeters, 6.86 cm of RMS exactly, which is true also for the orbit residuals that now display centimeter-level values in radial and transverse direction, while the normal component shows an RMS value of 6.1 cm. These two figures alone demonstrate that the developed batch algorithm has good stability and convergence, where as few as 5 iterations lead to satisfying results in the face of the strong initial perturbations. In this regard, tables 7.4 and 7.5 illustrate the result obtained when a cannon-ball satellite model is used in batch estimation. Regarding the dynamical state, it is interesting to note how the difference between the initial value and the estimated value, indicated by $\Delta\mathbf{X}$, actually compensates for the initial added perturbation, which is a sign of good algorithm performance. In fact, since we have the *true orbit* at our disposal, it is reasonable to expect that an adequate estimation would eventually lead to an absorption of all the perturbing effects that has been superimposed to the true solution, net of force modeling shortcomings and measurement noise. With regard to the estimated parameters it is interesting to note how especially the drag coefficient C_D approaches the unitary value starting from the 2.2 drag coefficient of a sphere. Knowing that this parameter is a scale factor for the drag acceleration, this fact indicates that the modeled attitude law, together with the box-wing model, is able to properly emulate the true dynamics of the satellite interaction with the higher atmosphere. On the other hand, the C_R parameter goes from 1 to 1.38 and for the empirical accelerations only the bias and once-per-rev parameters are estimated, with the twice-per-rev coefficients being numerical zeros. Regarding these last parameters, we should underline how the empirical accelerations have been constrained with a properly low value of their $\bar{\sigma}$ to be lower than the lowest modeled acceleration. This is done to ensure that these "artificial" perturbation will not absorb effects related to already modeled force.

Figure 7.2.4 shows the modeled accelerations profile over time for the whole estimation arc of 1 day. These accelerations are computed from the orbit propagated with the estimated initial conditions after the batch algorithm has converged, an here, we focus on the SRP and drag profiles that will be a subject of discussion ih the next Section. In particular, note how the SRP acceleration exhibits periodical downward cusps which are the result of the satellite's attitude law, as well as gaps in between period of illumination caused by eclipses.

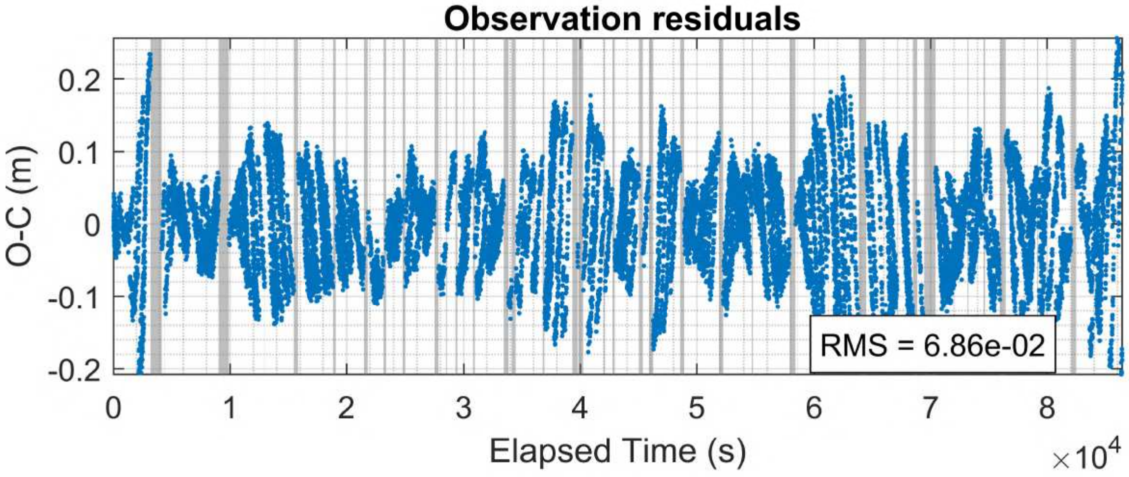

Figure 7.2.2: Difference between Observed and Computed ranges (O-C) at convergence (last iteration). vertical gray bands represent periods of non-observability. As we can see, the measurement residuals RMS value for this case is 6.86 cm against the 1 cm of σ error added when generating measurements.

One last aspect that must be addressed before proceeding with other batch estimation results, is related to the obtained orbit residuals in RTN components (figure 7.2.3). In general, their distribution and information content can be a good measure of the estimation algorithm performance. In fact, if one were able to perfectly model the real dynamics describing the satellite motion, the estimated residuals should behave as close as possible to a white random sequence with zero mean and standard deviation reflecting the measurements noise strength, however in this case, their profile clearly exhibit some information content. Figure 7.2.5 shows the amplitude spectral analysis of the orbit residuals, confirming that they all contain signals with characteristic frequencies in proximity first of all to the orbital rate, whereas radial and transverse components exhibit peaks

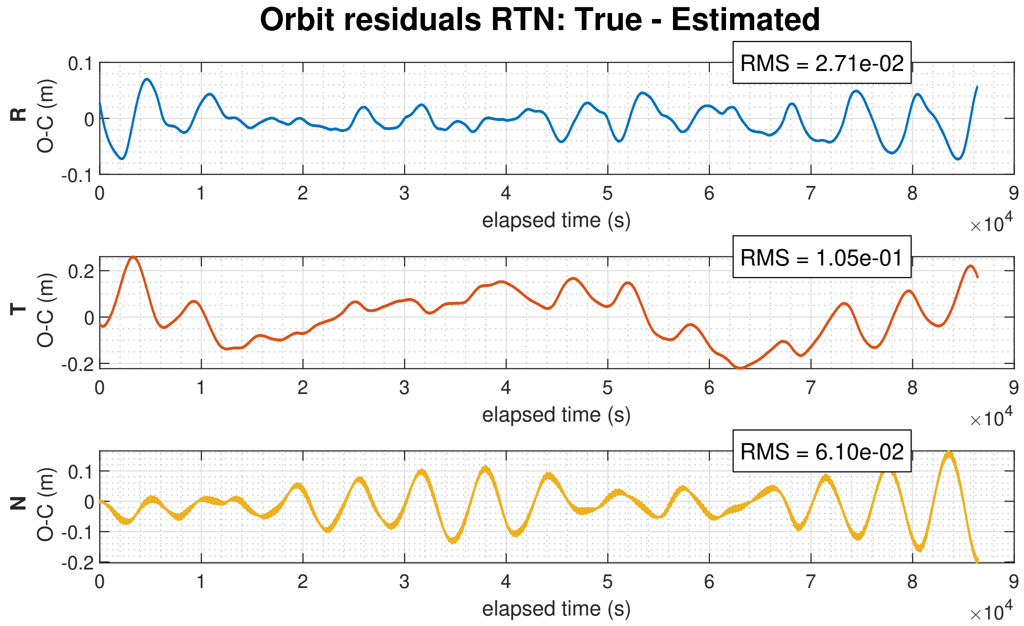

Figure 7.2.3: Difference between true and estimated orbit in RTN. Note how every component clearly contains signal, especially in the orbital period (once-per-rev), while the other evident frequency appears to be of near-diurnal nature.

also near the diurnal frequency. This behavior might find its explanation in the lack of a model for the ocean tides in the overall force model that is used to propagate the satellite's orbit, however, deeper investigation is needed to confirm this hypothesis, requiring a thorough study of current theories on ocean tides. In fact, even though we have a general knowledge of the frequency content, the orbit itself acts as a filter with its periodic motion, masking and distorting the true frequencies associated with a particular perturbing acceleration, making it extremely difficult to trace these signals back to the actual source.

Table 7.4: Batch Estimation results: position and velocity.

Parameter	**Initial Value $\bar{\mathbf{X}}_0$**	**Estimated Value $\hat{\mathbf{X}}_0$**	**ΔX**	$\hat{\boldsymbol{\sigma}}$
X (m)	$-3\,198\,972.3572464$	$-3\,199\,072.3572464$	-99.972	0.027
Y (m)	$-1\,651\,187.851021800$	$-1\,651\,087.8510218$	100.029	0.029
Z (m)	$6\,205\,796.0563044$	$6\,205\,596.0563044$	-200.009	0.025
$\dot{X}$ (m/s)	$4\,459.970115317$	$4\,459.870115317$	-0.10001	0.0000218
$\dot{Y}$ (m/s)	$4\,790.627269888$	$4\,790.577270892$	-0.04999	0.0000173
$\dot{Z}$ (m/s)	$3\,565.497695423$	$3\,565.427776769$	-0.06991	0.0000369

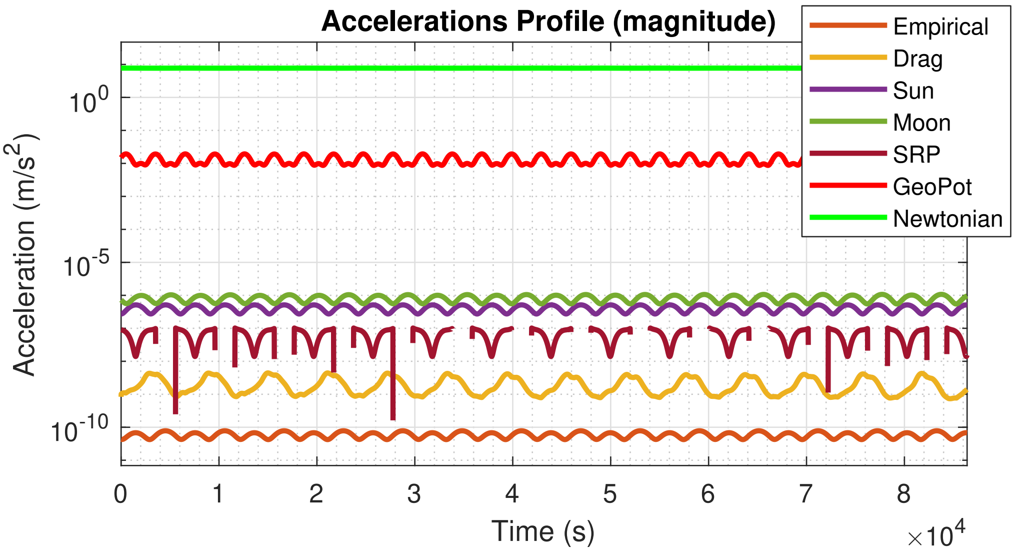

Figure 7.2.4: Trend of accelerations over time. Note the presence of the empirical accelerations which have been constrained to be weaker than every other modeled effect.

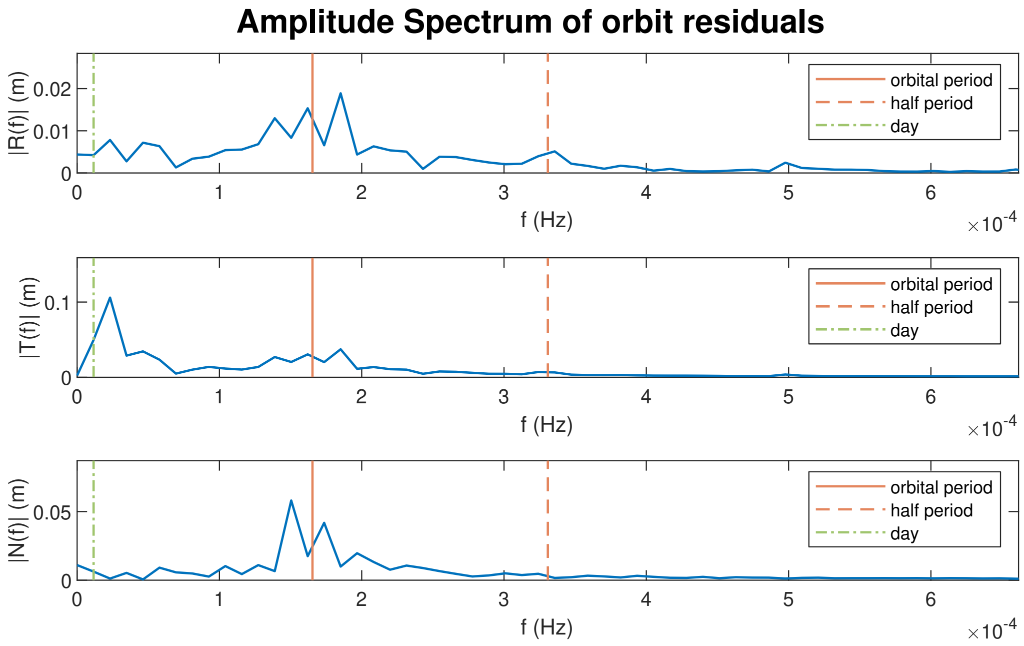

Figure 7.2.5: Amplitude spectral analysis of difference between true and estimated orbit in RTN coordinates. Note how every component exhibits peaks near the orbital frequency, whereas the transverse direction displays a strong peak in the near-diurnal frequency.

Table 7.5: Batch Estimation results: parameters.

Parameter	**Initial Value $\bar{\mathbf{X}}_0$**	**Estimated Value $\hat{\mathbf{X}}_0$**	$\Delta\mathbf{X}$	$\hat{\boldsymbol{\sigma}}$
C_R	1	1.38	0.38	0.0064
C_D	2.2	0.92	-1.28	0.0452
A_R (m/s^2)	0	0.1×10^{-12}	0.1×10^{-12}	10^{-12}
A_T (m/s^2)	0	1.59×10^{-11}	-6.7×10^{-12}	10^{-12}
A_N (m/s^2)	0	-0.43×10^{-12}	-2.1×10^{-12}	10^{-12}
D_{R_1} (m/s^2)	0	1.7×10^{-11}	1.7×10^{-11}	10^{-10}
S_{R_1} (m/s^2)	0	3.2×10^{-11}	3.2×10^{-11}	10^{-11}
D_{T_1} (m/s^2)	0	6.4×10^{-11}	6.4×10^{-11}	10^{-11}
S_{T_1} (m/s^2)	0	-3.6×10^{-11}	-3.6×10^{-11}	10^{-11}
D_{N_1} (m/s^2)	0	-1×10^{-11}	-1×10^{-11}	10^{-11}
S_{N_1} (m/s^2)	0	-2×10^{-11}	-2×10^{-11}	10^{-11}
D_{RTN_2} (m/s^2)	0	≈ 0	≈ 0	–
S_{RTN_2} (m/s^2)	0	≈ 0	≈ 0	–

7.2.2 Cannon Ball Model and Parameters Segmentation

This Section discusses and analyzes the results of batch estimation using a *cannon-ball* satellite model in contrast to the previously discussed *box-wing* model. In particular, the effects of parameter segmentation is addressed as a possible solution to improve the orbit determination when a box-wing model is not available.

The batch algorithm initialization for the cannon-ball case is identical to the one used in the previous Section, thus, we still refer to tables 7.2 and 7.3 for the initial values and a priori information. The only difference is that the C_R and C_D parameters now start from the estimated values previously obtained with the box-wing model. First, we consider the case in which the batch estimation is performed with single constant values for C_D and C_R, i.e., they are not segmented. Figure 7.2.6 and tables 7.6-7.7 show the resulting estimated orbit residuals and initial state vector, where again, convergence is achieved in 5 iterations. By examining the observation residuals we see how even for this case the RMS vale is under 10 cm, however, as can be expected, it has increased of about 2 cm. This quantity might seem irrelevant at first, but at these levels, it represents a remarkable worsening of the orbit determination performances. The behavior of the C_D and C_R parameters is also worth noting. We can see how, starting from the values estimated with the box-wing model, the SRP coefficient grows to 1.59, while the C_D parameter exhibits an exceptional increase from 0.92 to 2.83, reflecting the significant effect of properly modeling or not the actual satellite's shape and attitude law. Finally, figure 7.2.7 shows the accelerations profile for the cannon-ball satellite model. Here, in contrast to the box-wing case, we see how estimating a single C_R value does not allow the modeled SRP to capture the actual dynamics arising from the satellite motion. In fact, by comparing this results with figure 7.2.4, we can see how the SRP profile has flattened out and the characteristic cusps are not present. The Drag acceleration profile on the other hand, remains almost the same, indicating that a properly estimated single C_D value is able to capture the fundamental dynamics involved in the satellite-atmosphere interaction.

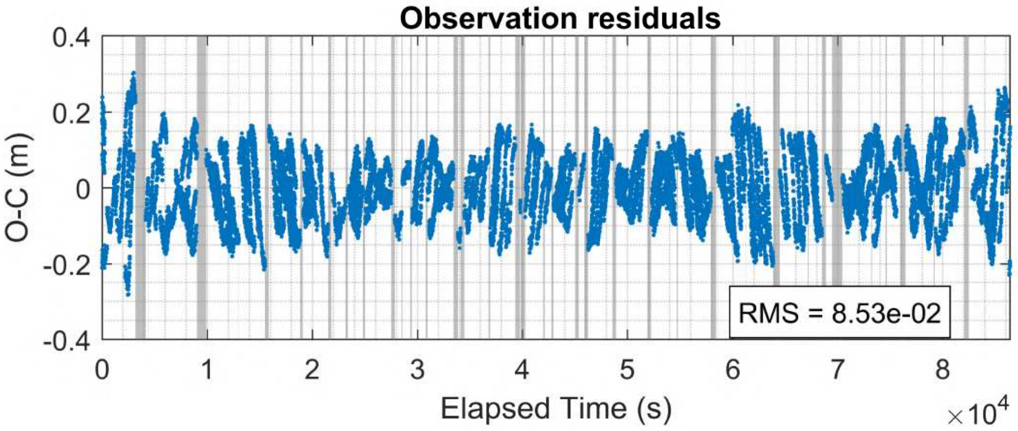

Figure 7.2.6: Observation residuals for the batch estimation with cannon-ball satellite model.

We now proceed to examine the result that are obtained with the segmentation of C_R and C_D parameters, which has been performed following the techniques described in Section 4.4. First, we can see in figure 7.2.8 how there is a conspicuous improvement in the observation residuals RMS value which has decreased to 4.73 cm, which is even lower that the 6.86 cm value for the bow-wing model. However, care is required when drawing conclusions from the observation residuals, in fact, a low RMS value is not always an indicator of good orbit determination since an instantaneous range measurement is geometrically "weak" under certain circumstances. As a matter of fact, having the "real" orbit at our disposal, allows for a better performance assessment. In confirmation of

Table 7.6: Batch Estimation results: position and velocity for the cannon-ball satellite model.

Parameter	Initial Value $\bar{\mathbf{X}}_0$	Estimated Value $\hat{\mathbf{X}}_0$	$\Delta\mathbf{X}$	$\hat{\boldsymbol{\sigma}}$
X (m)	−3 198 972.3572464	−3 199 072.2164357	−99.861	0.027
Y (m)	−1 651 187.8510218	−1 651 088.0261696	99.925	0.029
Z (m)	6 205 796.0563044	6 205 596.0571736	−199.992	0.025
$\dot{X}$ (m/s)	4 459.970115317	4 459.870082856	−0.09998	0.0000218
$\dot{Y}$ (m/s)	4 790.627269888	4 790.577278232	−0.04995	0.0000173
$\dot{Z}$ (m/s)	3 565.497695423	3 565.427799766	−0.06996	0.0000369

Table 7.7: Batch Estimation results: parameters for the cannon-ball satellite model.

Parameter	Initial Value $\bar{\mathbf{X}}_0$	Estimated Value $\hat{\mathbf{X}}_0$	$\Delta\mathbf{X}$	$\hat{\boldsymbol{\sigma}}$
C_R	1.30	1.59	0.292	0.0064
C_D	0.92	2.87	1.956	0.0452

this, a careful interpretation of figure 7.2.11, reveals that in terms of orbit residuals the cannon-ball model with segmented parameters has not led to an improvement in overall orbit determination if compared to the box-wing case, in fact, every RTN component displays a higher RMS value.

Nonetheless, the process of parameter segmentation has remarkably improved the orbit estimation if compared to the previous cannon-ball model with single parameters. This improvement is not only suggested by a decrease in residuals RMS values, but also from the amplitude spectral analysis shown in figure 7.2.12. Here, we compare the spectrum of the segmented and non-segmented case, and as we can see, the parameter segmentation process leads to an overall decrease in amplitude, especially near the orbital frequency, suggesting that segmentation is able to absorb effects that the cannon-ball model alone is too “rigid” to capture. This is further confirmed by examining the C_R and C_D estimated values, as well as the consequent acceleration profiles. Figures 7.2.9 and 7.2.10 demonstrate how these parameters are no longer constrained to a single vale that is valid throughout the whole estimation arc, now, they are a collection of independent parameters each valid during a certain period along the orbit. We can note how they exhibit a somewhat periodical behavior suggesting that the segmentation is trying to compensate both the lacking modeled attitude law and the variable cross-section associated to a box-wing model. As a consequence, from figures 7.2.13 and 7.2.14, we can appreciate how the segmented parameters, especially the SRP coefficient C_R, is approaching the acceleration profile that would be obtained with a box-wing model, further suggesting that parameter segmentation is a powerful tool for orbit determination when we lack of a detailed satellite geometry and attitude.

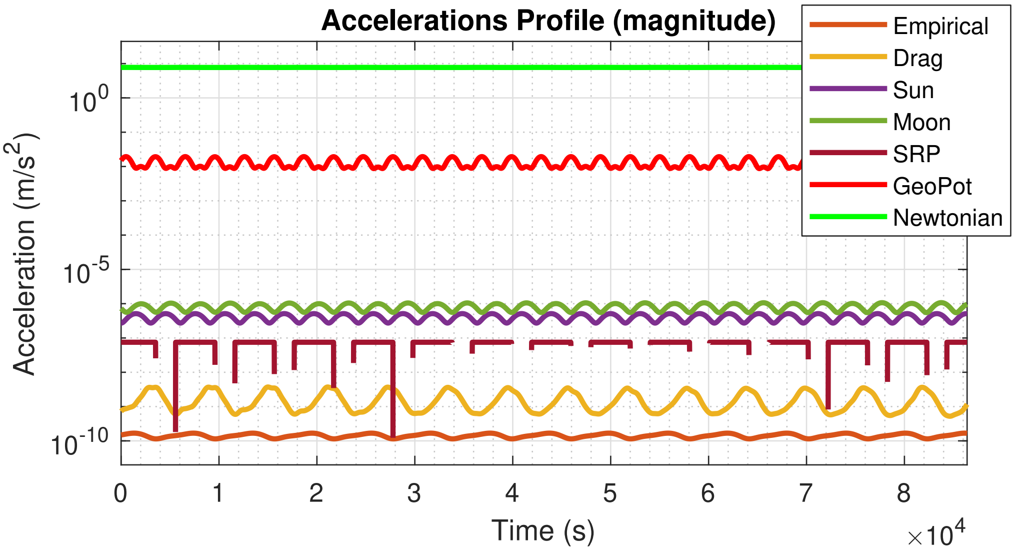

Figure 7.2.7: Trend of modeled acceleration vs time with cannon-ball satellite model. Note how the SRP profile is flat during period of illumination, in contrast to the acceleration profile obtained with the box-wing model.

Table 7.8: Batch Estimation results: position and velocity for the parameters segmentation case.

Parameter	**Initial Value $\bar{\mathbf{X}}_0$**	**Estimated Value $\hat{\mathbf{X}}_0$**	$\Delta\mathbf{X}$	$\hat{\boldsymbol{\sigma}}$
X (m)	−3 198 972.3572464	−3 199 072.2151322	−99.857	0.027
Y (m)	−1 651 187.8510218	−1 651 087.9043893	99.946	0.029
Z (m)	6 205 796.0563044	6 205 596.0641031	−199.992	0.025
$\dot{X}$ (m/s)	4 459.970115317	4 459.870131903	−0.09998	0.0000218
$\dot{Y}$ (m/s)	4 790.627269888	4 790.577313108	−0.04995	0.0000173
$\dot{Z}$ (m/s)	3 565.497695423	3 565.427731181	−0.06996	0.0000369

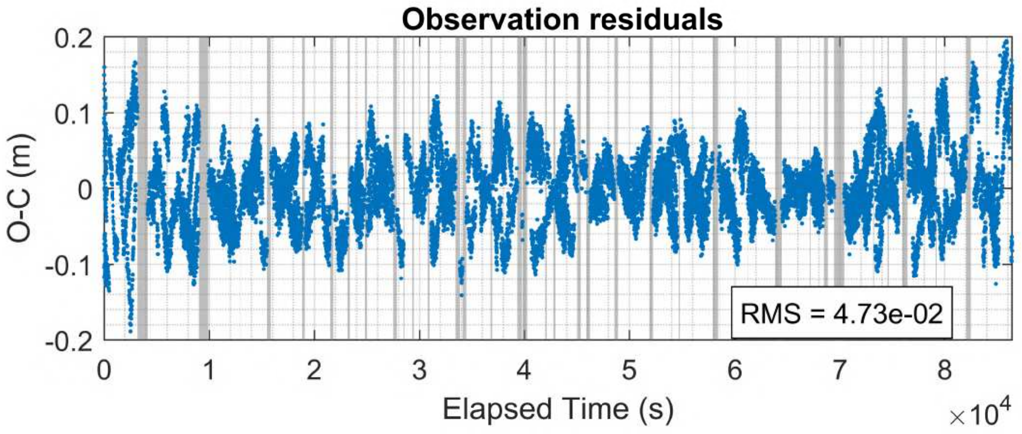

Figure 7.2.8: Observation residuals for the batch estimation through parameter segmentation.

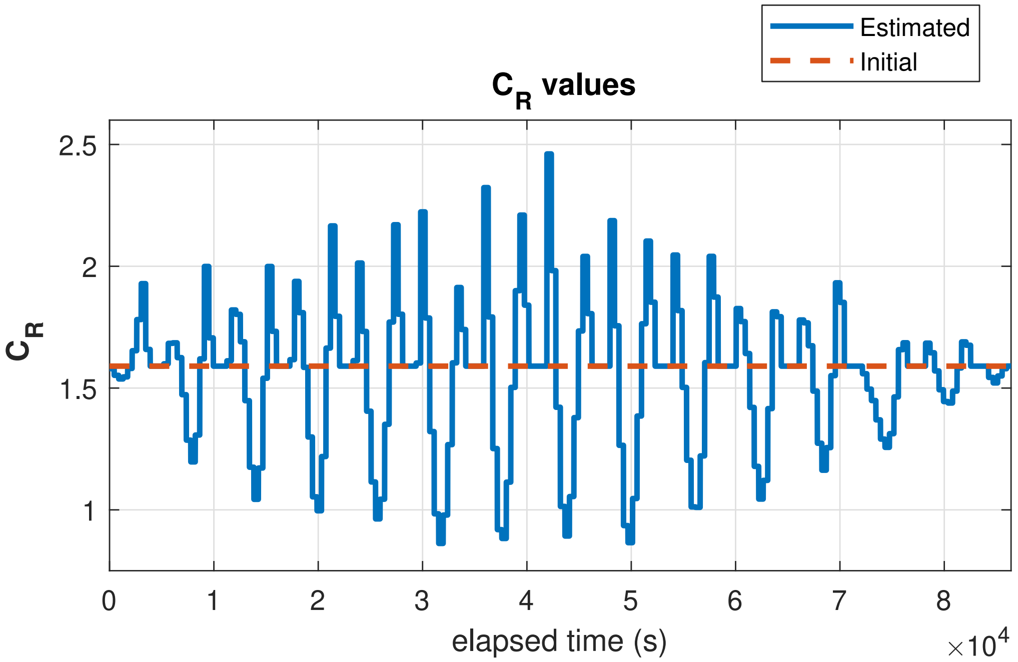

Figure 7.2.9: Estimated C_R values through parameter segmentation.

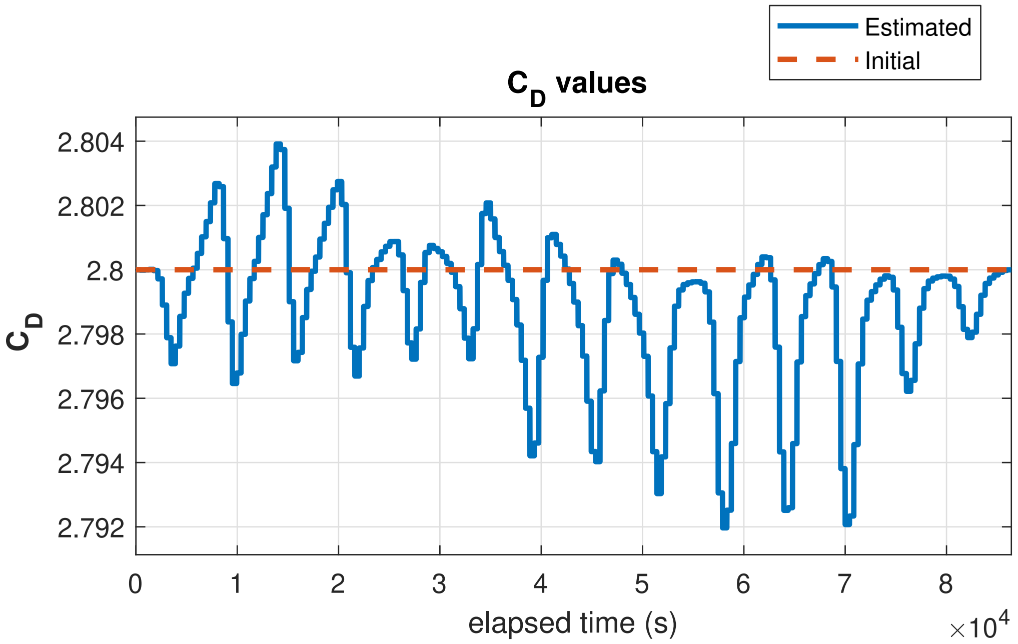

Figure 7.2.10: Estimated C_D values through parameter segmentation.

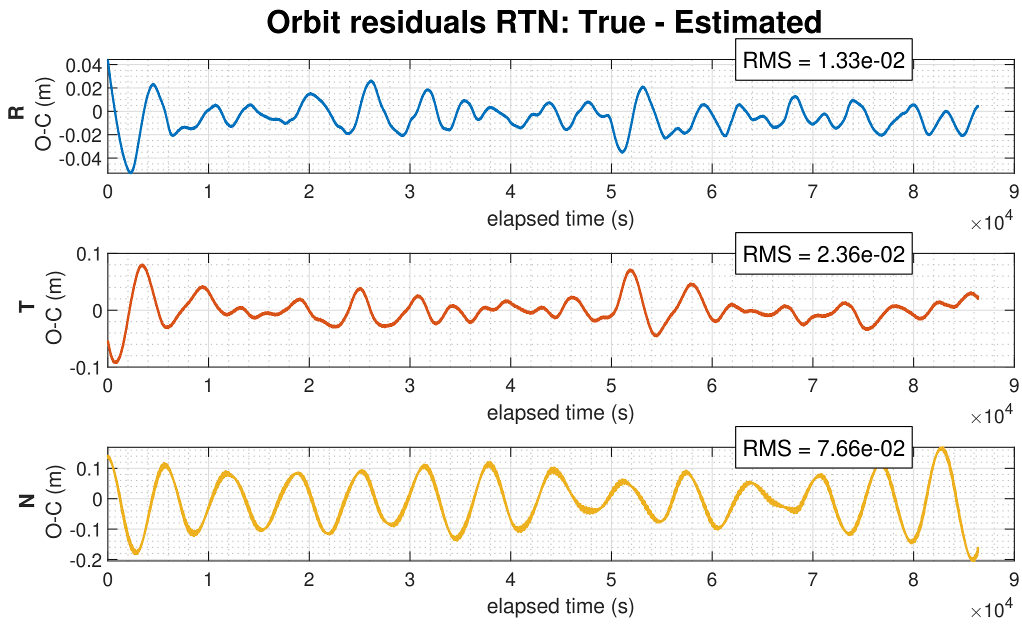

Figure 7.2.11: Orbit residuals for the batch estimation through parameter segmentation.

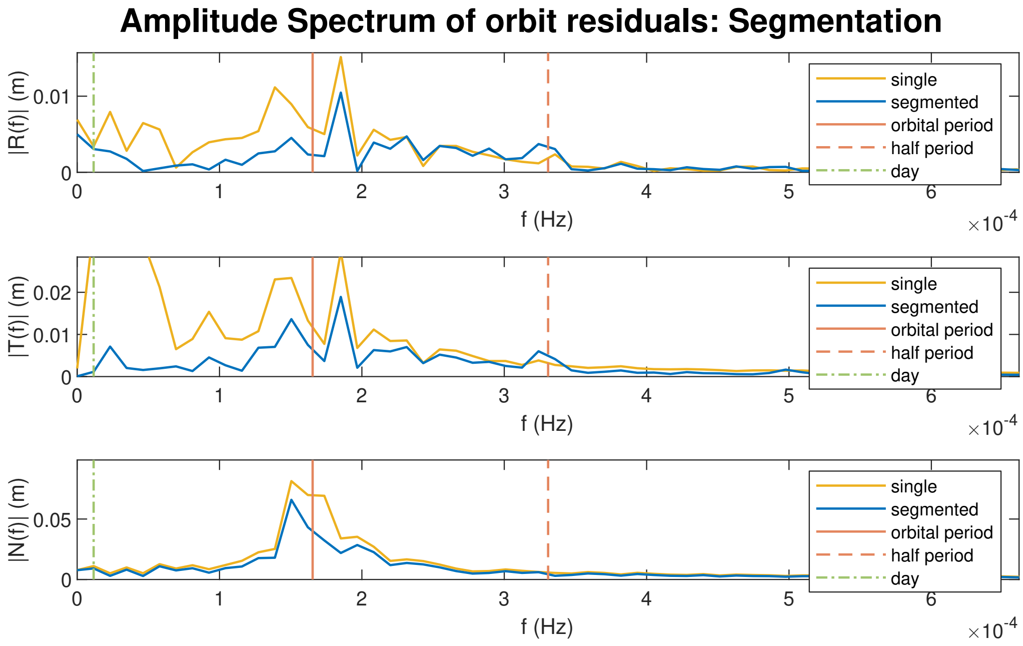

Figure 7.2.12: Orbit residuals spectrum for the batch estimation through parameter segmentation.

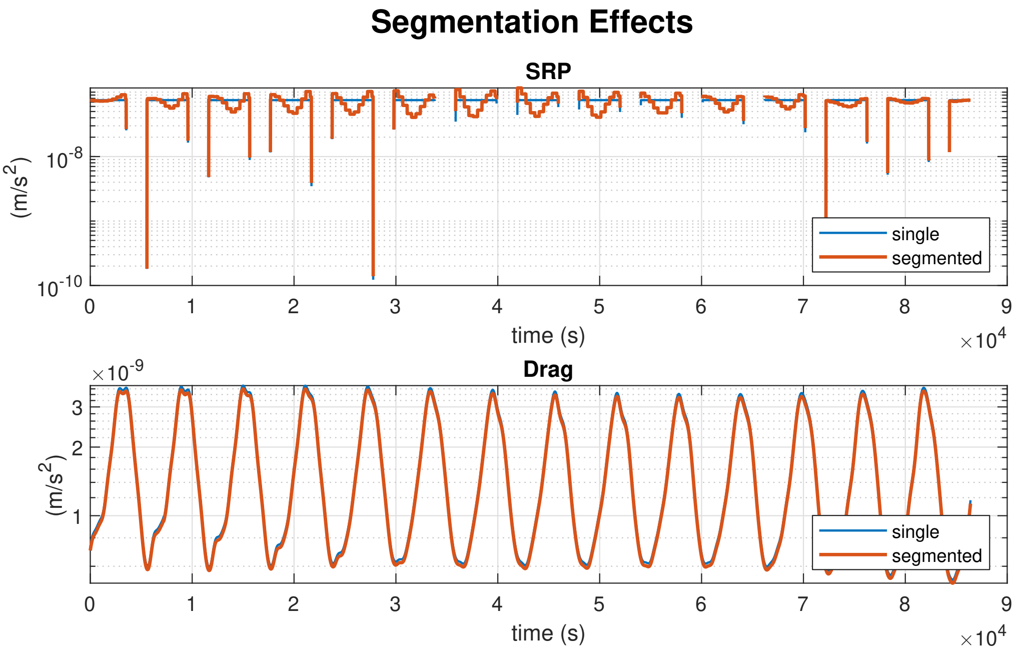

Figure 7.2.13: SRP and Drag acceleration profiles. Comparison between parameters segmentation and cannon-ball model with single parameters.

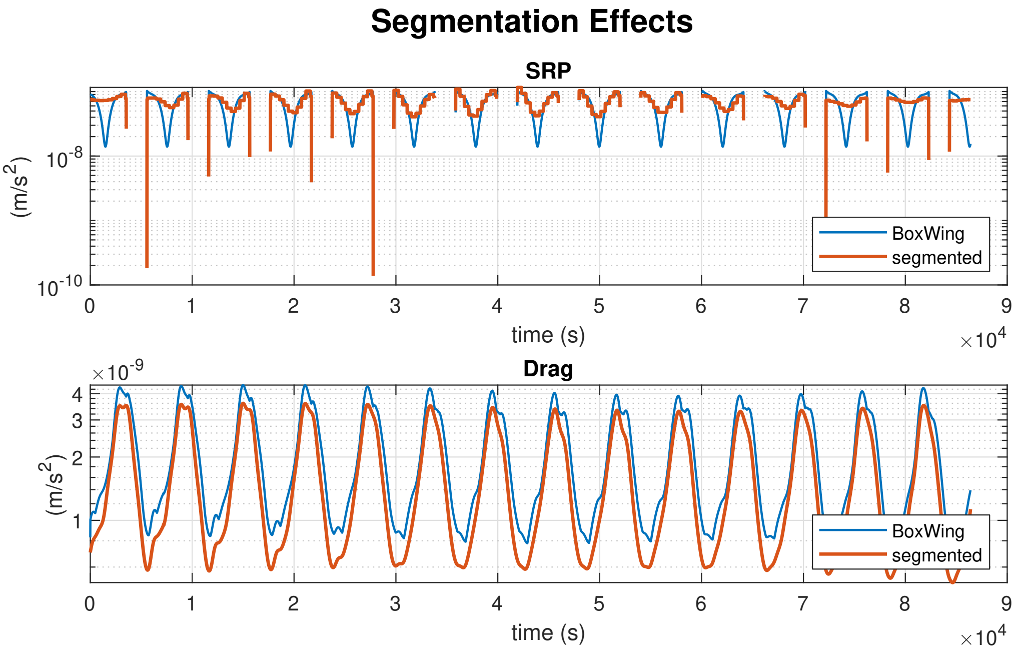

Figure 7.2.14: SRP and Drag acceleration profiles. Comparison between parameters segmentation and box-wing model with true attitude law. Note how the C_R segmentation enables the SRP acceleration to approach the true profile arising from the satellite's attitude motion.

7.2.3 Earth Radiation Pressure Effects

A good part of the research work carried out for this thesis has been dedicated to the analysis and development of an Earth Radiation Pressure model. This model is intended to expand the already available set of Fortran routines that compute the major accelerations acting on a satellite, and its details and algorithmic implementation has been discussed in Section 2.6. As explained there, the importance of being able to account for fine and weak accelerations associated to precision modeling, is a crucial aspect for a bacth estimation process rather than a real-time estimator. This is due to the fact that when dealing with a batch processor, we are estimating the state vector at a certain reference epoch, then, we want this single set of parameters to be able to fit an entire batch of measurements over a more or less extended temporal arc, making it particularly sensitive to model incompleteness and approximations. Accordingly, this Section investigates the effects of the inclusion of an ERP model, and the estimation of its parameters, when performing orbit determination through batch processing.

The simulations in this case are performed with a non-segmented box-wing model, in view of this, the ERP results will be compared to the ones obtained in Section 7.2.1, whereas the initialization remains the same. Here, however, the *satellite reflectivity* parameter η_E associated to the ERP model is added to the estimation state vector as shown in table 7.9. First, by examining the observation residuals (figure 7.2.15), we can see that there is a reduction in the RMS value of about 2 mm if compared to the box-wing case without ERP. This reduction has shown to be consistent during many different simulations. Moreover, figure 7.2.17 compares the spectrum of orbit residuals for the box-wing model with and without ERP, here, we can see how slight reductions in the peaks near the orbital frequency are visible, especially for the radial and transverse components, suggesting that the added acceleration is benefiting the orbit determination process, albeit only to a low extent.

Another aspect that suggests the correct modeling and implementation of the earth radiation pressure acceleration lies in its comparison with the SRP perturbation. As stated in Section 2.6, when well implemented, the ERP should be around 15% to 25% of the SRP acceleration, with exceptional values of 35% for very low satellites. Figure 7.2.19, shows that the estimated orbit gives an acceleration profile for the SRP with an RMS of 6.1×10^{-8} m/s^2, while the RMS for ERP is 1.23×10^{-8} m/s^2, whereas their ratio gives an approximate value of 20%, which is in good accordance with the expected result.

Finally, tables 7.10 and 7.11 show the results of the batch estimation with the inclusion of ERP. As we can see, the η_E parameter does not deviate much from its starting value, whereas the C_D and C_R parameters remains substantially the same as the ones estimated in section 7.2.1, as the ERP effect is too weak to cause substantial changes in parameters estimation.

Table 7.9: Parameters initial values and a priori information. ERP case

Parameter	**Initial Value**	$\bar{\mathbf{x}}$	$\bar{\boldsymbol{\sigma}}$
C_R	1	0	0.5
C_D	2.2	0	0.5
η_E	0.13	0	0.05

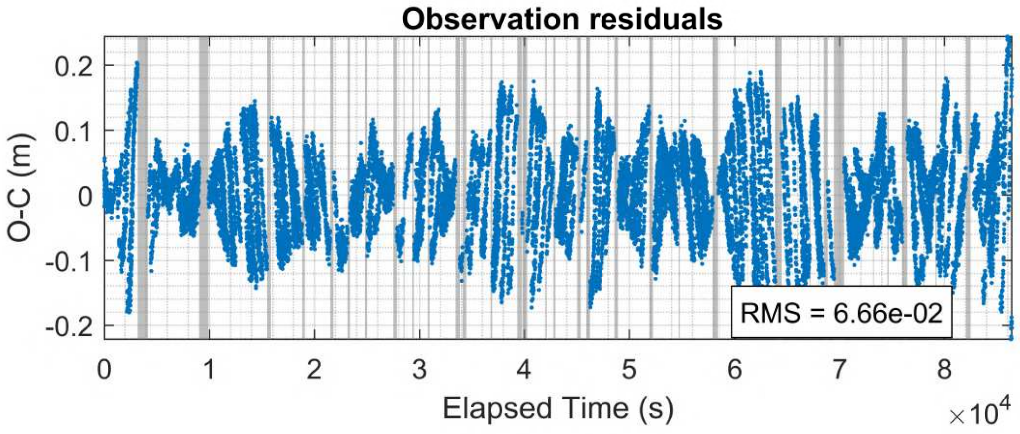

Figure 7.2.15: Observation residuals for the batch estimation when ERP is modeled and estimated.

Table 7.10: Batch Estimation results: position and velocity. ERP case.

Parameter	Initial Value $\bar{\mathbf{X}}_0$	Estimated Value $\hat{\mathbf{X}}_0$	$\Delta\mathbf{X}$	$\hat{\sigma}$
X (m)	$-3\,198\,972.3572464$	$-3\,199\,072.3448428$	-99.987	0.027
Y (m)	$-1\,651\,187.851021800$	$-1\,651\,087.8541227$	99.996	0.029
Z (m)	$6\,205\,796.0563044$	$6\,205\,596.0326844$	-200.023	0.025
$\dot{X}$ (m/s)	$4\,459.970115317$	$4\,459.870091958$	-0.10002	0.0000218
$\dot{Y}$ (m/s)	$4\,790.627269888$	$4\,790.577264603$	-0.05000	0.0000173
$\dot{Z}$ (m/s)	$3\,565.497695423$	$3\,565.427784770$	-0.06991	0.0000369

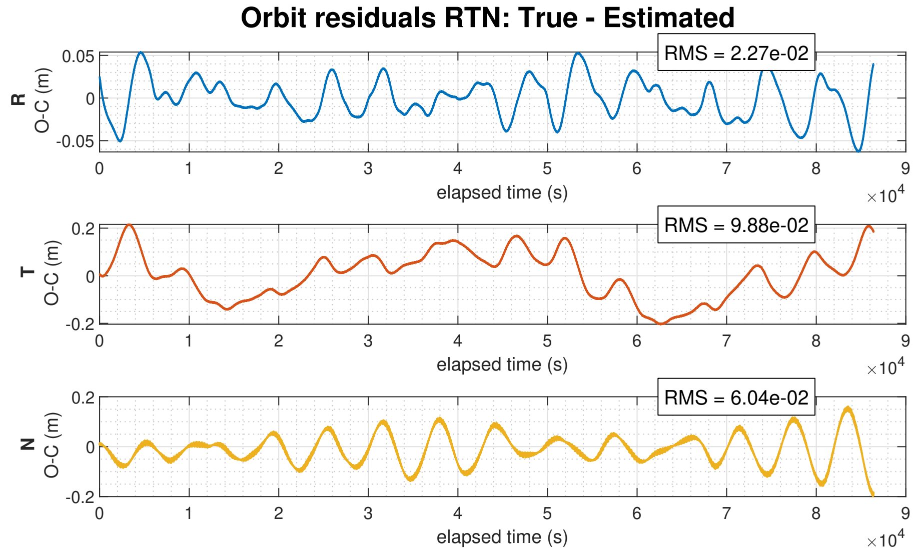

Figure 7.2.16: Orbit residuals in RTN components for the case of modeled ERP acceleration.

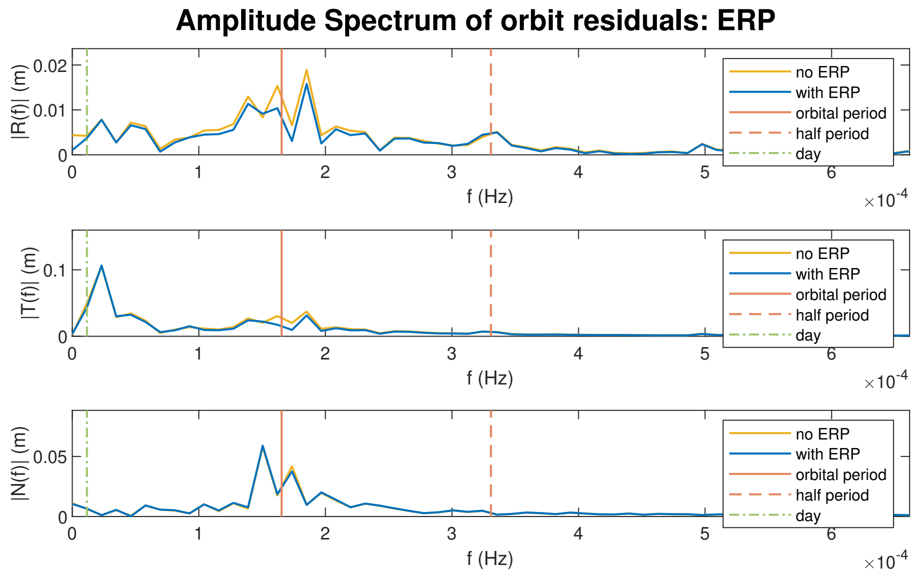

Figure 7.2.17: Comparison of orbit residuals spectrum with and without the modeled ERP acceleration. Note how there is a general reduction in amplitude around the orbital frequency, especially for the radial and transverse components.

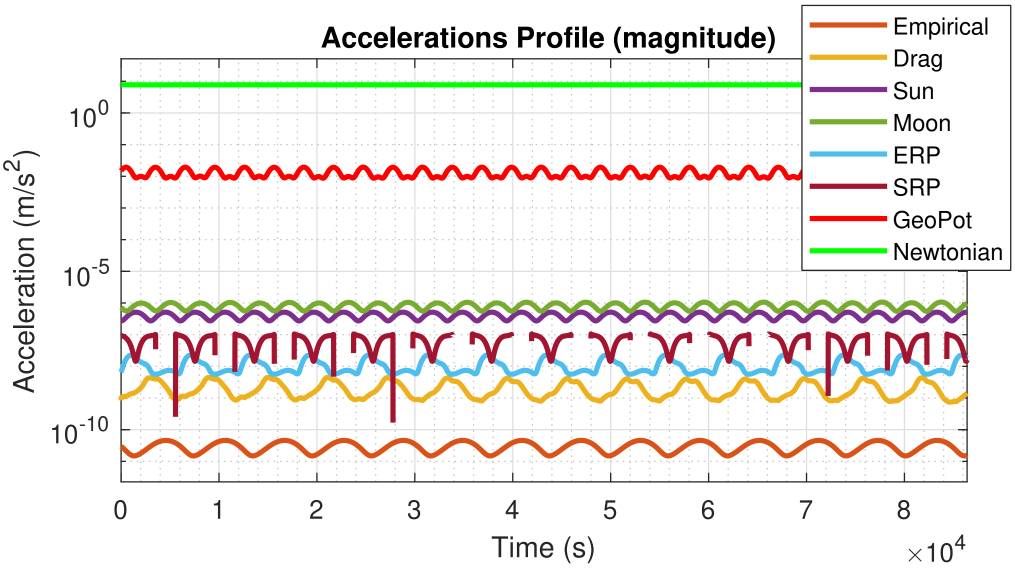

Figure 7.2.18: Trend of modeled acceleration vs time. Note the presence of the ERP acceleration below the SRP profile.

Table 7.11: Batch Estimation results: parameters. ERP case.

Parameter	Initial Value $\bar{\mathbf{X}}_0$	Estimated Value $\hat{\mathbf{X}}_0$	$\Delta\mathbf{X}$	$\hat{\boldsymbol{\sigma}}$
C_R	1	1.42	0.424	0.0064
C_D	2.2	0.93	−1.269	0.0452
η_E	0.13	0.134	0.004	0.0498

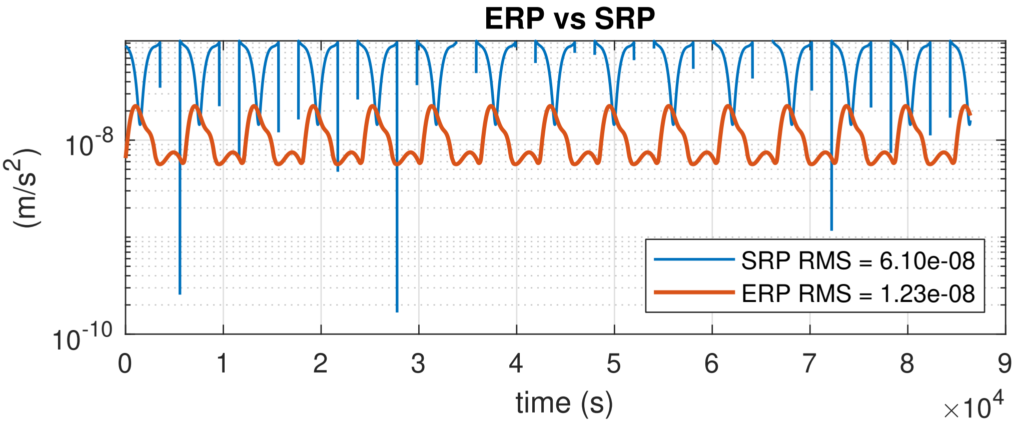

Figure 7.2.19: ERP vs SRP acceleration.

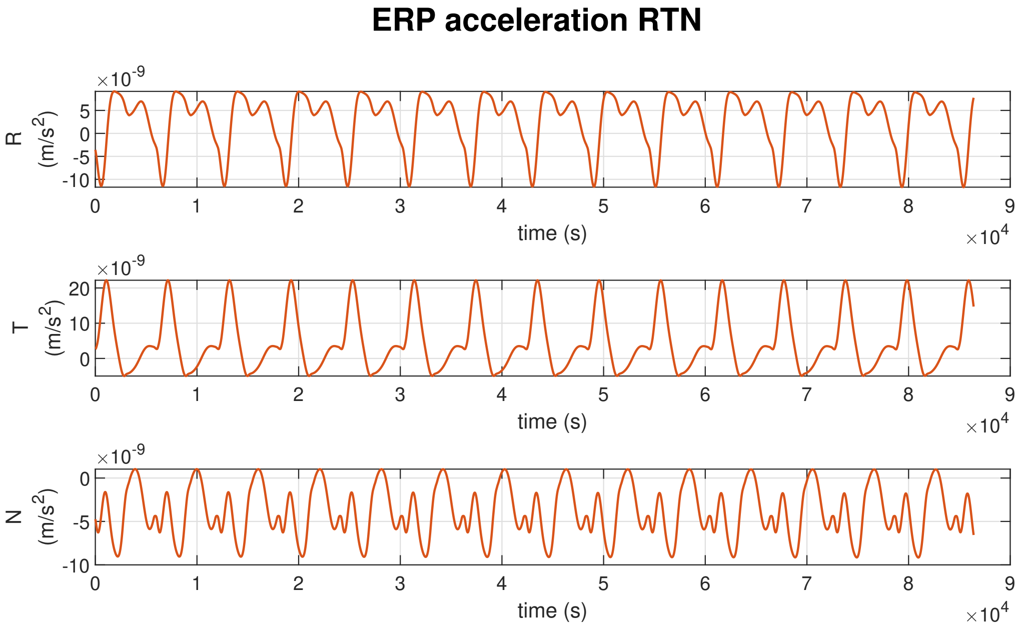

Figure 7.2.20: Estimated ERP acceleration in RTN components.

7.3 Sentinel-3A Real-Time Orbit Determination

In this section the orbit determination process is carried out with real-time estimators. In particular, the Extended Kalman Filter and the effect of process noise are the focus of the following presented results, which still come from the orbit estimation of Sentinel-3A with generated range measurements as was the case for the batch estimation. The difference here, is that observations are not processed all at once as a single batch of data to estimate the state at a reference epoch t_0, instead, measurements are read and processed sequentially one instant at a time, providing a state estimate at every time of observation. This characteristic makes the Kalman filter suitable for on-line applications such as the Precise Onboard Orbit Determination (P2OD), where the state estimate is carried out directly onboard the satellite and is continuously updated by periodic measurements.

7.3.1 The Extended Kalman Filter

Here we illustrate and discuss the results obtained with the EKF algorithm developed in Fortran as described in figure 5.2.1, with the addition of process noise and estimation of its parameters as illustrated in Section 5.3.2. We recall that the Extended formulation of the Kalman filter uses each new estimate as the new reference trajectory, while the estimation of process noise is peculiar to the Dynamic Model Compensation approach (DMC), where the unmodeled accelerations are represented by a first-order three-dimensional Gauss-Markov process. Before going on with the discussion of the results, we must underline how every simulation related to real time estimation, including the forward filter in the smoothing process discussed in the next Section, will use a reduced dynamical model (reduced-dynamic approach) compared to the case of batch estimation. The reason for this, is the already mentioned suitability of the EKF to onboard applications, whereas computational budget is always strictly limited to a low extent, meaning that there are not enough resources to compute complex and detailed force models. Moreover, as the state is periodically updated by new measurements, the Kalman filter is less subjected to long propagation periods as is the case for the batch estimator. This suggests that other than reducing the modeled dynamics, a less sophisticated integrator can be used, usually a Runge-Kutta-Hull(2)4 is sufficient for most applications.

Figures 7.3.1 and 7.3.2 show the observation and orbit residuals for the EKF algorithm applied to the Sentinel-3A satellite. At first, we immediately notice how in both cases the real-time filter outperforms the batch estimator with an RMS in observation residuals of 1.32 cm, and RMS in orbit residuals of 1.94 cm for position and 33.4 μm$/s$ for velocity. It is noted that here, the EKF begins with the same initial values and a priori information used in the previous batch estimates. this means that the filter exhibit very good stability and convergence, even in the face of numerous periods of low or absent visibility. With this regard it is interesting to note how a correlation can be identified between higher peaks in residuals and periods in which fewer or none stations are in sight, highlighting the real-time nature of the EKF algorithm whose estimation capability strongly depends on currently available information density and content (recall the measurement update phase).

Regarding the parameters estimation we refer to figures 7.3.3 and 7.3.4. In view of the reduced-dynamics approach and the reduced computational cost associated to real-time filters, the use of a box-wing model in an EKF approach is seldom implemented. For this reason, a cannon-ball model is used in every EKF simulation presented here and interestingly, after an initial transient, both the C_D and C_R approach values that are in good accordance with the estimated parameters obtained in batch estimation with the cannon-ball satellite model, thus suggesting a sort of consistency between both orbit determination procedures.

Finally, figure 7.3.5 shows the profile of the estimated process noise acceleration. Remember that this is the deterministic part of the process noise, whereas the random part contributes in preventing filter saturation as will be shown later on. Note how every estimated process noise component is on the order of few nm/s^2 except for the y component that reaches an RMS value of roughly 10 nm/s^2, which is still in good accordance with typical process noise value for this type of satellites.

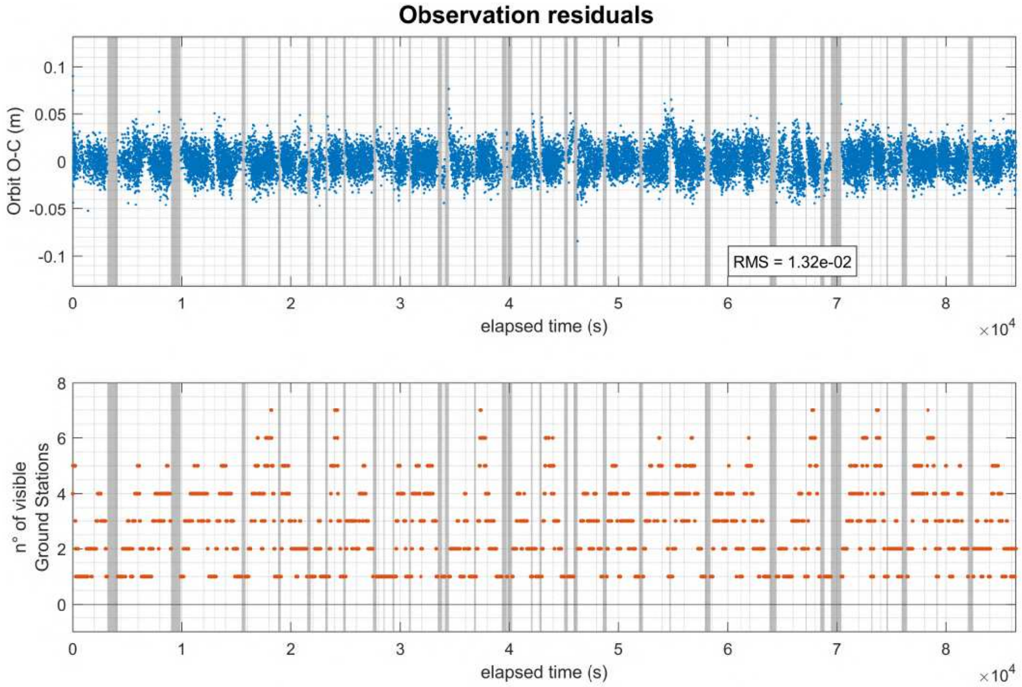

Figure 7.3.1: Observation residuals for the Extended Kalman Filter estimation (above) and number of visible round stations (below). Note how there is a correlations between high peaks in O-C values and periods of low or absent visibility.

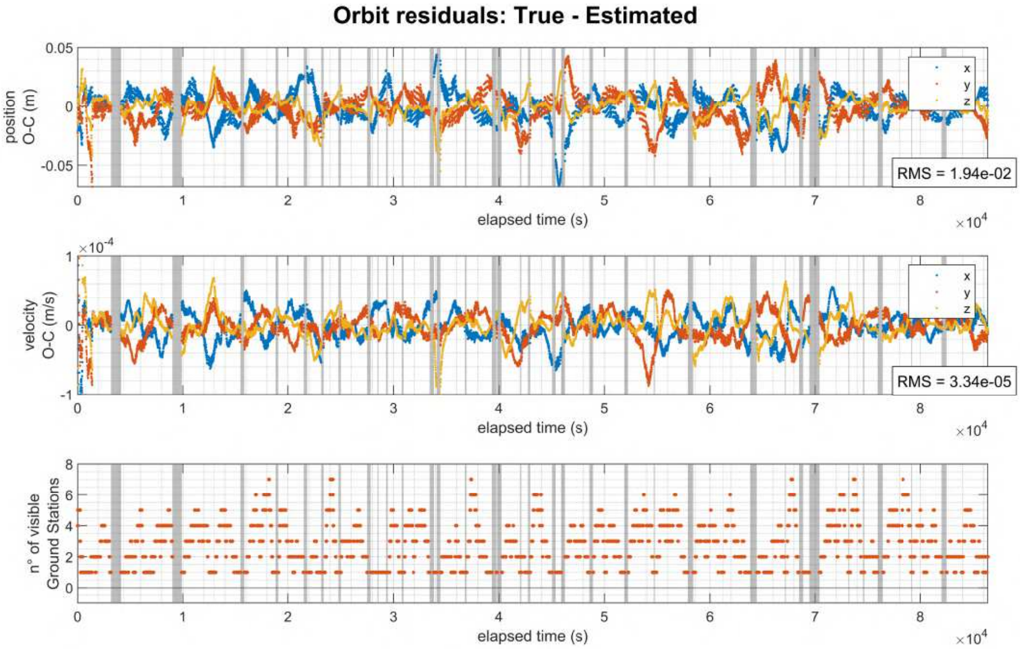

Figure 7.3.2: Orbit residuals (position and velocity) for EKF estimation. Note how there is a correlation between high peaks in residuals and periods of low or absent visibility.

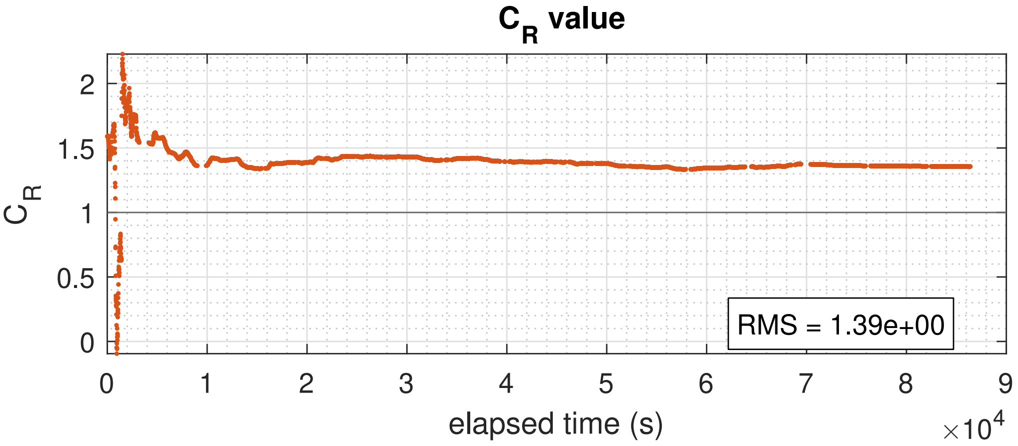

Figure 7.3.3: C_R parameter value over time during EKF estimation.

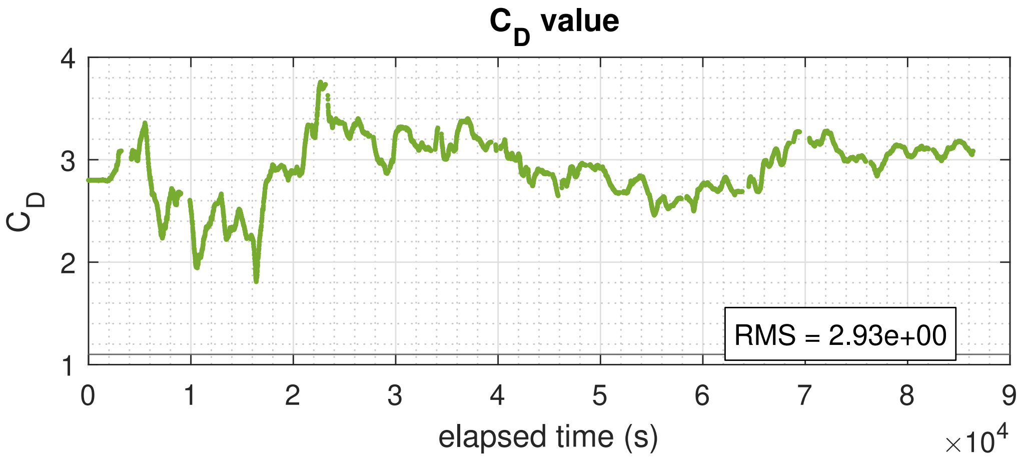

Figure 7.3.4: C_D parameter value over time during EKF estimation.

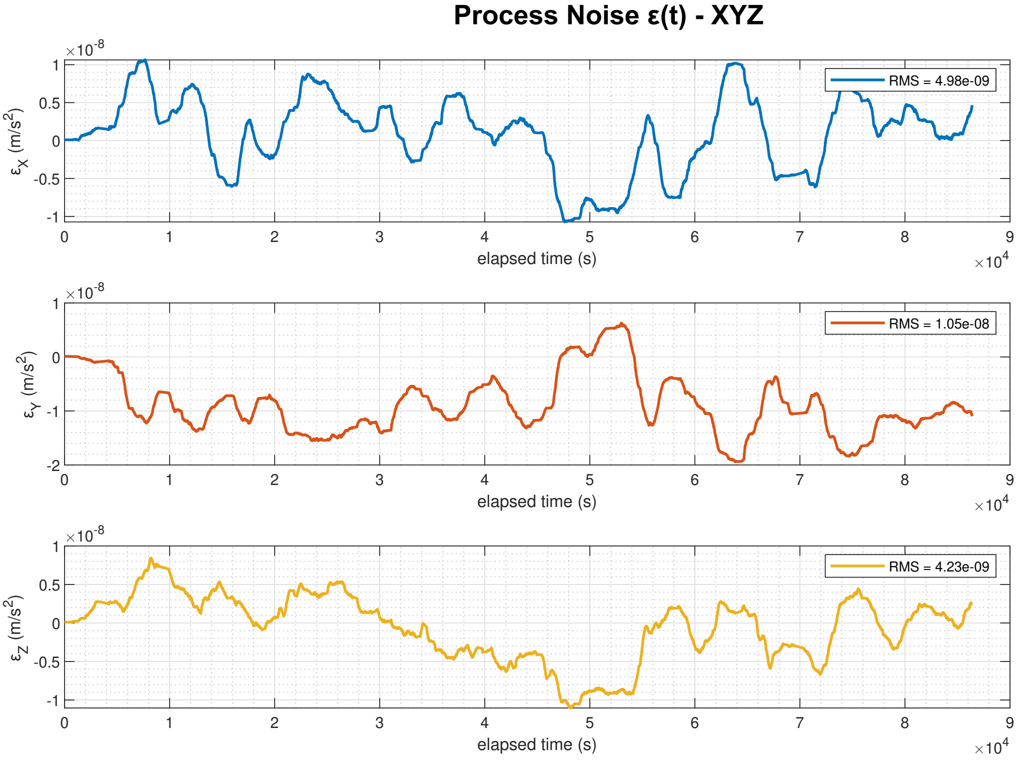

Figure 7.3.5: Deterministic process noise $\varepsilon(t)$ estimation with EKF.

7.3.2 The Effect of Process Noise

This Section demonstrates the crucial role of the process noise in a real-time filter like the EKF algorithm. As explained in Section 5.3 the importance of this added noise lies in its capacity to avoid the so called filter divergence. In the absence of process noise we have shown how the Kalman gain **K** approaches zero as time passes, meaning that the filter becomes insensitive to new observations and will solely rely on the a priori estimate $\bar{\mathbf{x}}$. This a priori value in turn, comes from the reference trajectory propagation which will always be in error to some extent, eventually leading to divergence and filter saturation. From another point of view, this divergence is caused by the the solution covariance matrix **P** approaching zero, which indicates that the computed estimate has no uncertainty, thus erroneously leading to the assumption that the estimated state represents the true exact solution. Here, the EKF algorithm results without the inclusion of process noise are shown, validating its vital importance when using a real-time approach to the orbit determination problem.

Figures 7.3.6 and 7.3.7 evidently demonstrate the detrimental effects arising from the lack of process noise in a Kalman filter. Both observation and orbit residuals exhibit a divergent trend, with values that are now of meter-level magnitude in contrast to the stable centimeter-level values obtained with the inclusion of process noise. In addition, if the estimation arc gets longer, higher residual values can be expected.

However, it is interesting to note how from figures 7.3.8 and 7.3.9 the trend for C_R and C_D parameters seems to stabilize somehow near the expected values after a long time, nonetheless, high magnitude initial transients still confirm the negative performance of EKF without the inclusion of process noise.

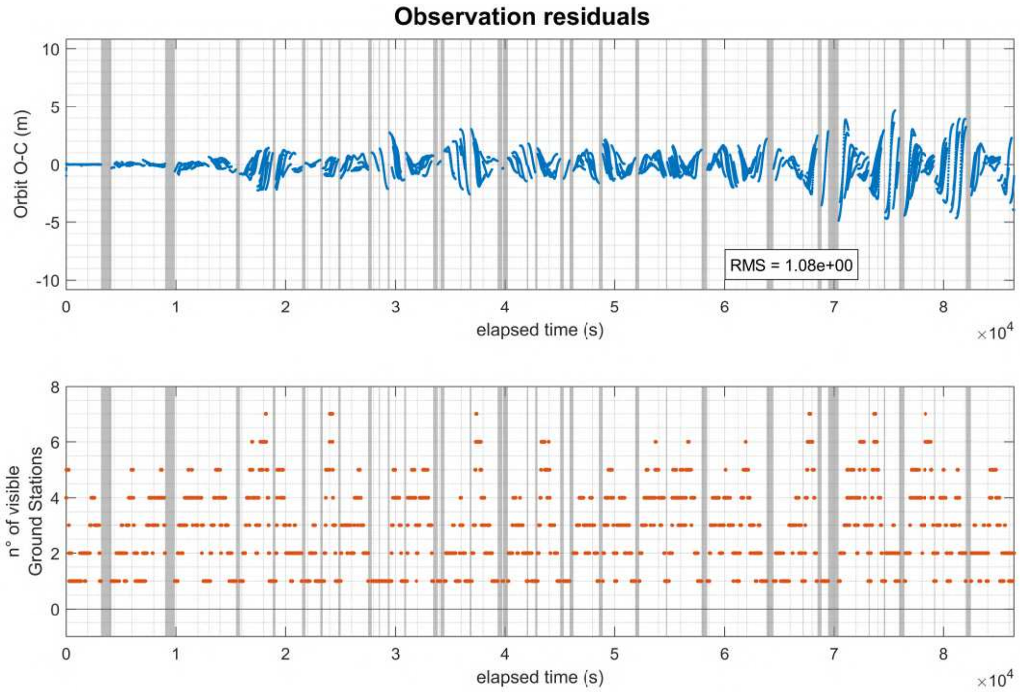

Figure 7.3.6: Observation residuals for the Extended Kalman Filter estimation (above) and number of visible round stations (below) without process noise. Note how there is a clear divergent trend over time.

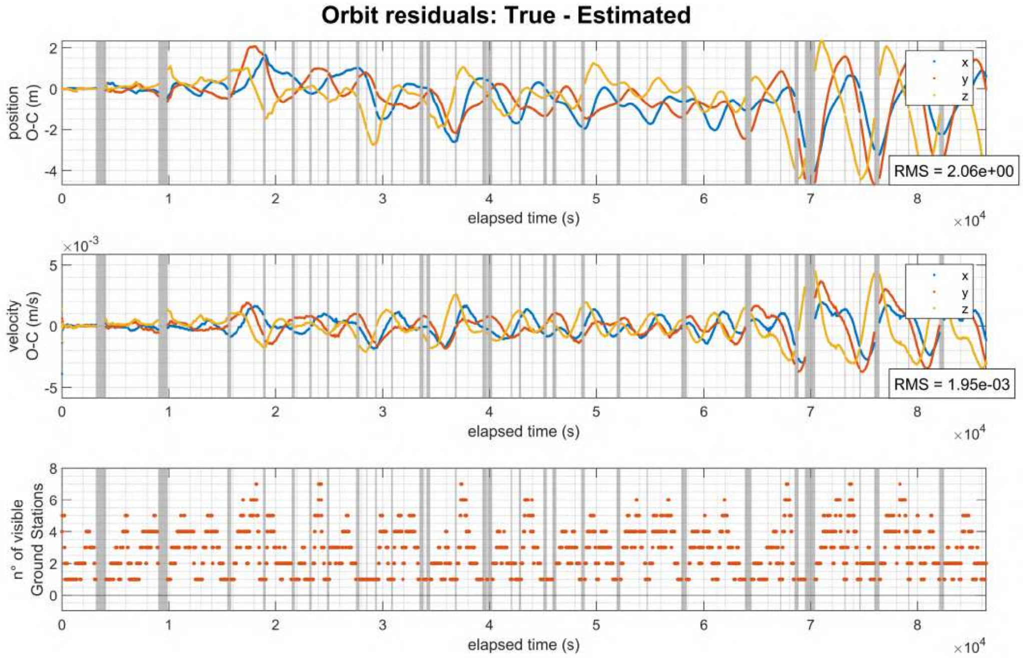

Figure 7.3.7: Orbit residuals (position and velocity) for EKF estimation without process noise. Note how there is a clear divergent trend over time.

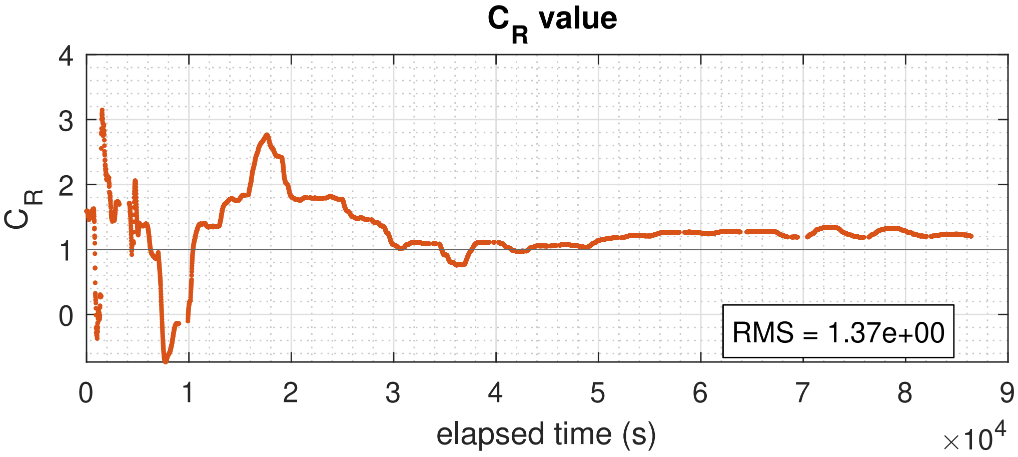

Figure 7.3.8: Estimated C_R parameter without process noise.

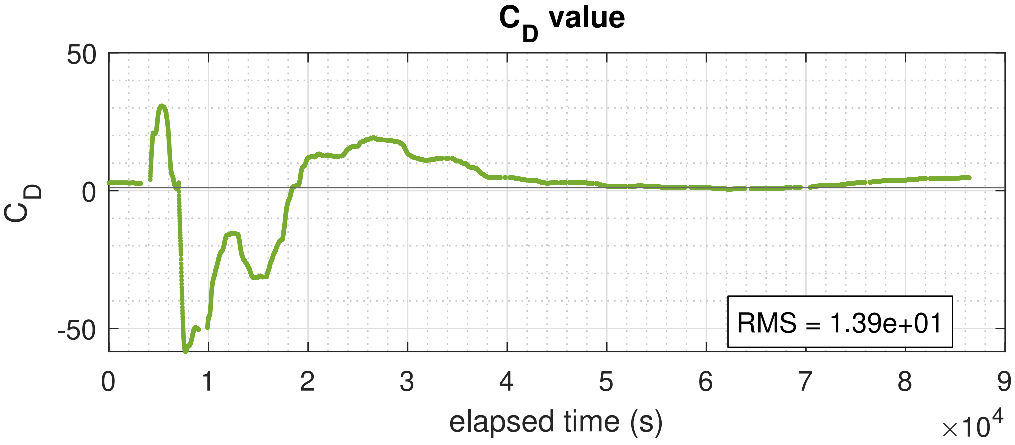

Figure 7.3.9: Estimated C_D parameter without process noise.

7.4 Sentinel-3A Orbit Determination with Smoothing

The last orbit determination technique that have been studied and implemented in Fortran for this thesis work is the smoother. Chapter 6 showed the algorithm derivation of two main types of fixed-interval smoothers: the *forward-backward* and the *RTS*. The latter has been chosen for the orbit determination of Sentinel-3A given its more simple and direct implementation that does not require the use of a backward information filter. For a complete comprehension of the results that will be shown here, we recall that the general structure of the RTS smoother algorithm consists of a forward Conventional Kalman Filter (CKF) and then a backward recursion based on stored values from the forward processing as detailed in figure 6.2.1, in fact, the majority of graphs in this section compare these two phases to highlight the beneficial effects of smoothing. Moreover, we remember that the CKF is a Kalman filter where the reference trajectory is not updated after processing each observation, but is simply propagated for the entire estimation arc. This is necessary to ensure that the backward recursion is completely independent from the forward filter, or in other words, it must be assured that observational information does not influence a priori values before the backward recursion begins. Hereafter, the smoothing process results are discussed and compared with the Kalman estimation, then, the effects of initial conditions on the smoother algorithm are shown.

7.4.1 The Smoothed Solution

As usual, we begin by assessing the orbit estimation performance by looking at the observation and orbit residuals. Figures 7.4.1, 7.4.2 and 7.4.3 demonstrate that a general improvement in the residuals RMS values is obtained with the smoothed solution when compared to the CKF results, even if slight (on the order of few millimeters), and note that this is also true when compared with the EKF results of the previous section (see figures 7.3.1 and 7.3.2). Besides the RMS values, by looking at these figures it is worth noting how the smoothing process has a greater impact on reducing the peaks exhibited by the forward filter solution, whereas peaks of 5 cm have been reduced to around 2 cm for position residuals, while in terms of velocity, peaks of more than 50 μm/s drop down to less than 40 μm/s. This results indicate that a first noticeable impact of smoothing is that of decreasing the dispersion of residuals around their mean value. As a matter of fact, the smoothing procedure does not necessarily provides more accurate results, but rather reduces their uncertainty thus increasing their accuracy, and it is by examining the solution covariance matrix $\mathbf{P}$ that the smoothing effects become more evident. Figures 7.4.8 and 7.4.9 show a comparison between the trace of $\mathbf{P}$ for the forward filter and the smoothed solution separated into position, velocity and parameters components to ensure dimensional consistency. For position and velocity we can see that the smoothed covariance is always lower than the filter covariance, but still follows the evident peaks pattern correlated to periods of poor observability as indicated by the usual gray vertical bands. This is true everywhere except for the end of the estimation arc when the backward recursion begins. Here, as explained in Chapter 6, the filter covariance and the smoothed covariance coincide by definition, see equation (6.1.22). It is also noteworthy how all the smoother solutions tend to be less jagged, more smooth indeed, which is highlighted in figure 7.4.10, where we see a detail of the comparison between the forward filter solution and smoother solution for the X position component, but is even more evident for the estimated process noise in figures 7.4.6 and 7.4.7.

The fact that smoothing seems to have the highest impact on the estimated process noise parameters actually reflects a crucial aspect that was discussed in Chapter 6 and is here validated by the results. When treating the theoretical derivation of the smoothing algorithm, we illustrated how a distinction between some estimation state vector components should be made based on their nature, in particular, we distinguished *dynamical states* and *bias states* as those components which are, or are not, subjected to the effect of process noise. This difference becomes meaningful in the current context of smoothing as we demonstrated how *bias states* are not smoothable, i.e., for these components the backward smoothing process become simple backward propagation and no beneficial effects can be appreciated. Actually, the more a state is subjected to process noise the more it is smoothable and this is the reason why the process noise itself is strongly smoothed, as well as position and velocity, which are subjected to the integrated effects of process noise. On the other hand, figures 7.4.5 and 7.4.9 show that for the bias states C_D and C_R, the smoothed solution is simply the last forward filter solution propagated backwards as a constant, both for their estimated values and covariances.

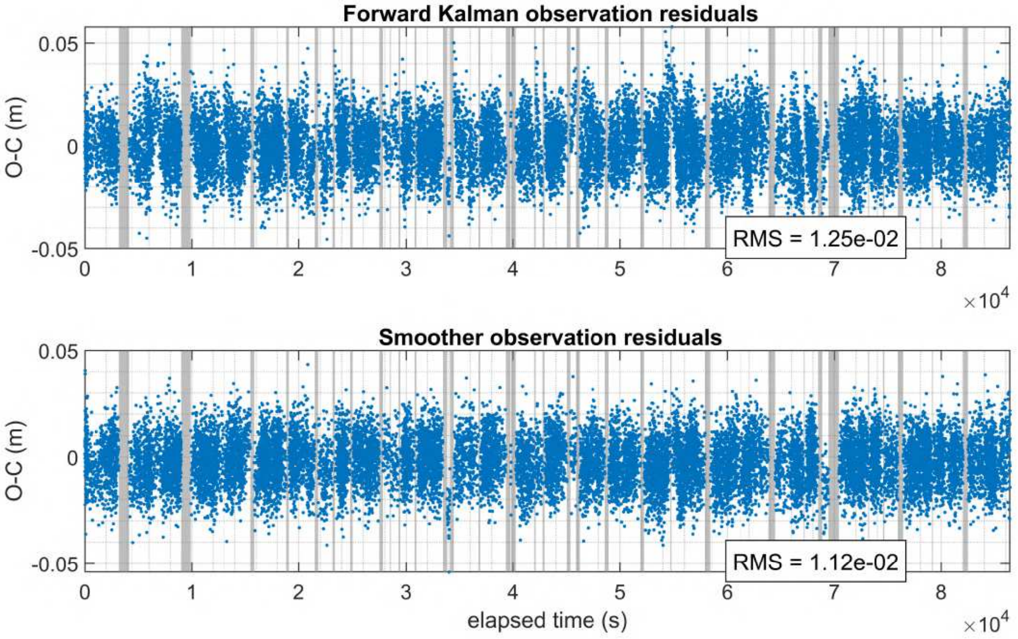

Figure 7.4.1: Observation residuals for the smoother algorithm. Above, the observation residuals for the forward filter are also displayed.

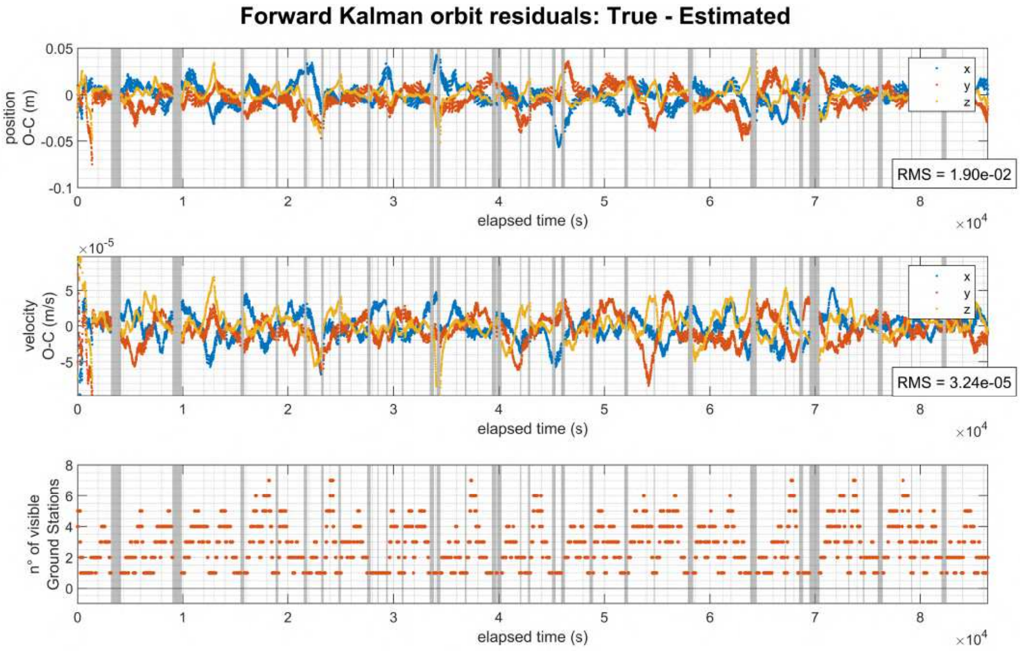

Figure 7.4.2: Orbit residuals for the forward conventional Kalman filter.

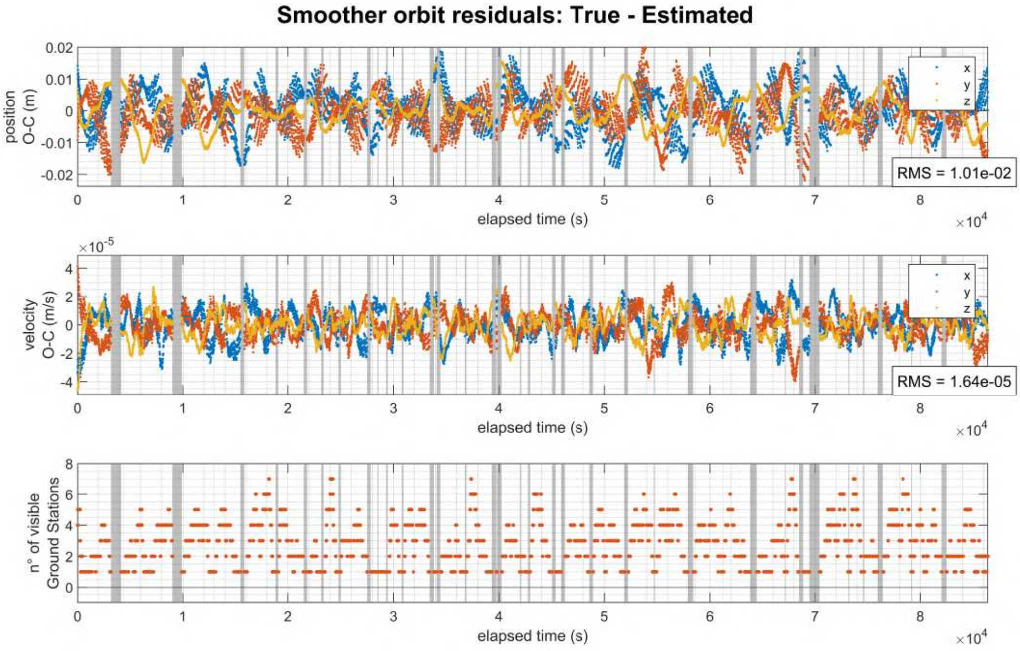

Figure 7.4.3: Orbit residuals for the backward RTS smoother.

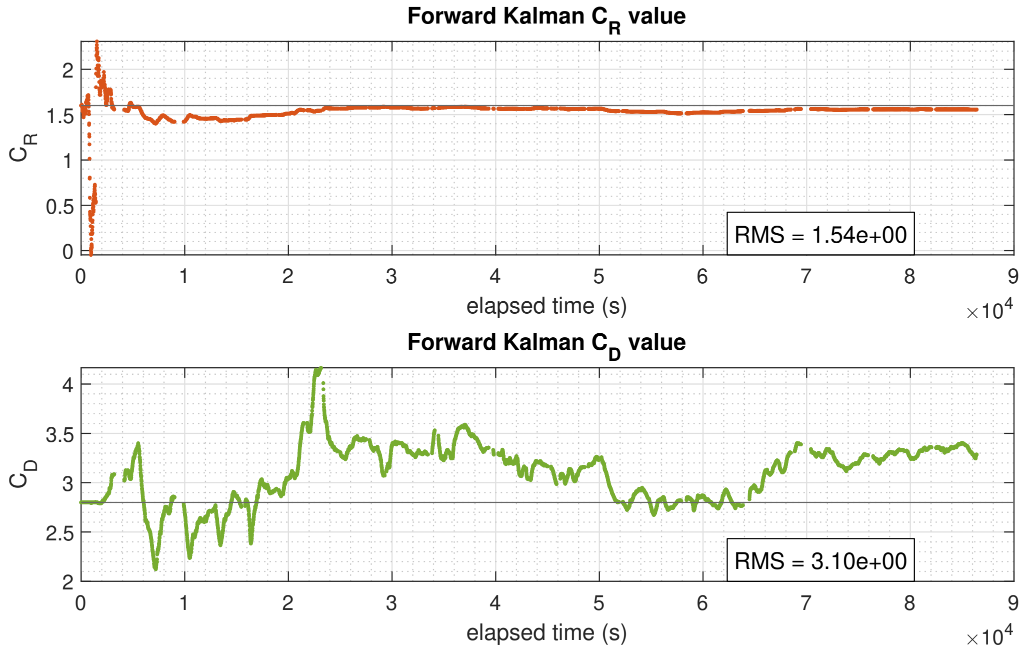

Figure 7.4.4: Estimated values of C_D and C_R parameters during the forward Kalman filter.

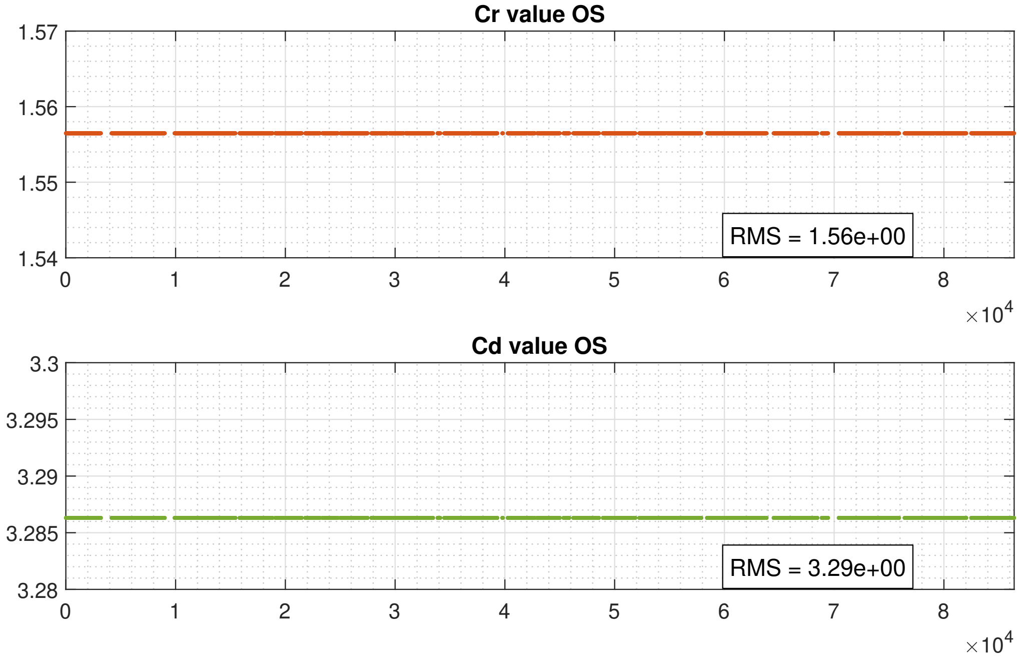

Figure 7.4.5: C_R and C_D parameters during backward smoothing. Since they are *bias states* they are not smoothed.

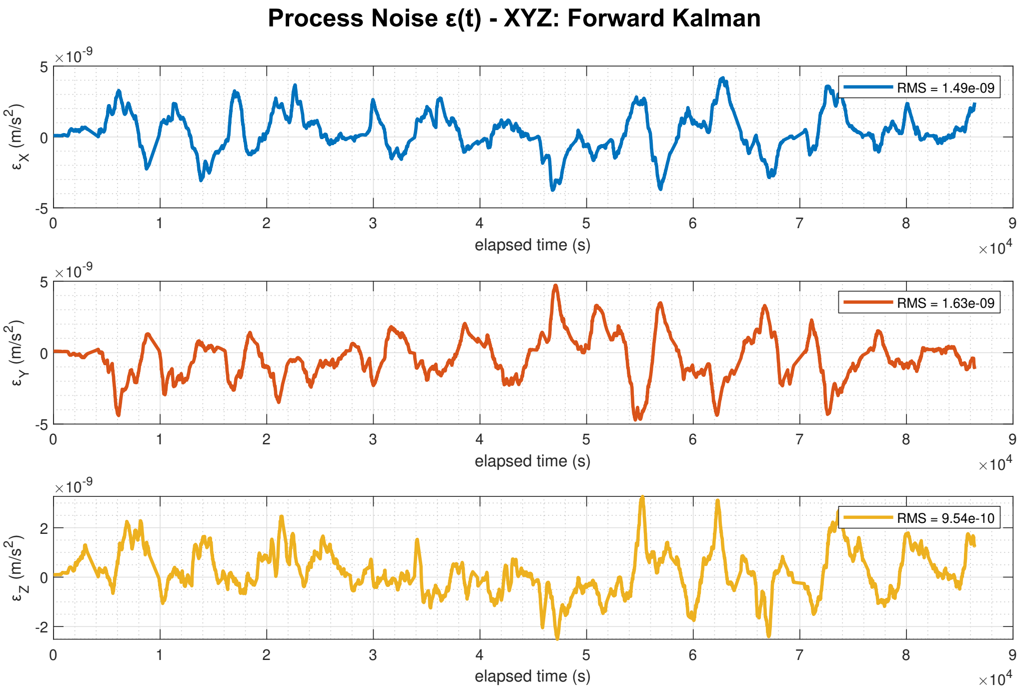

Figure 7.4.6: Estimated process noise parameters during the forward Kalman filter.

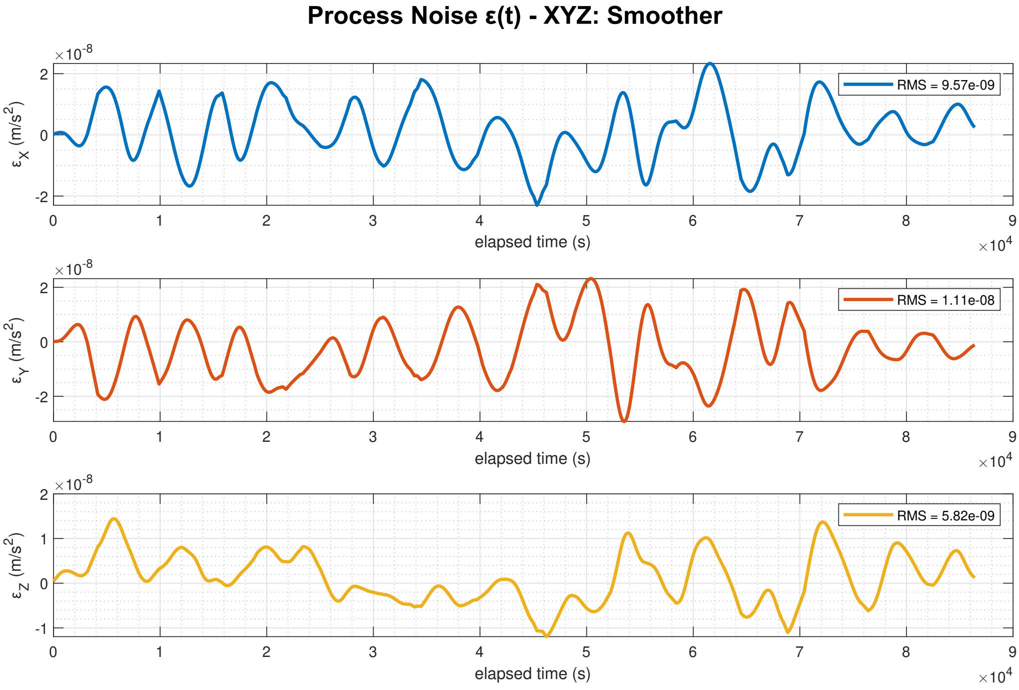

Figure 7.4.7: Estimated process noise parameters during the backward RTS smoother. Since the deterministic portion of the process noise $\boldsymbol{\varepsilon}(t)$ is subjected to process noise itself it is a smoothable *dynamical state*.

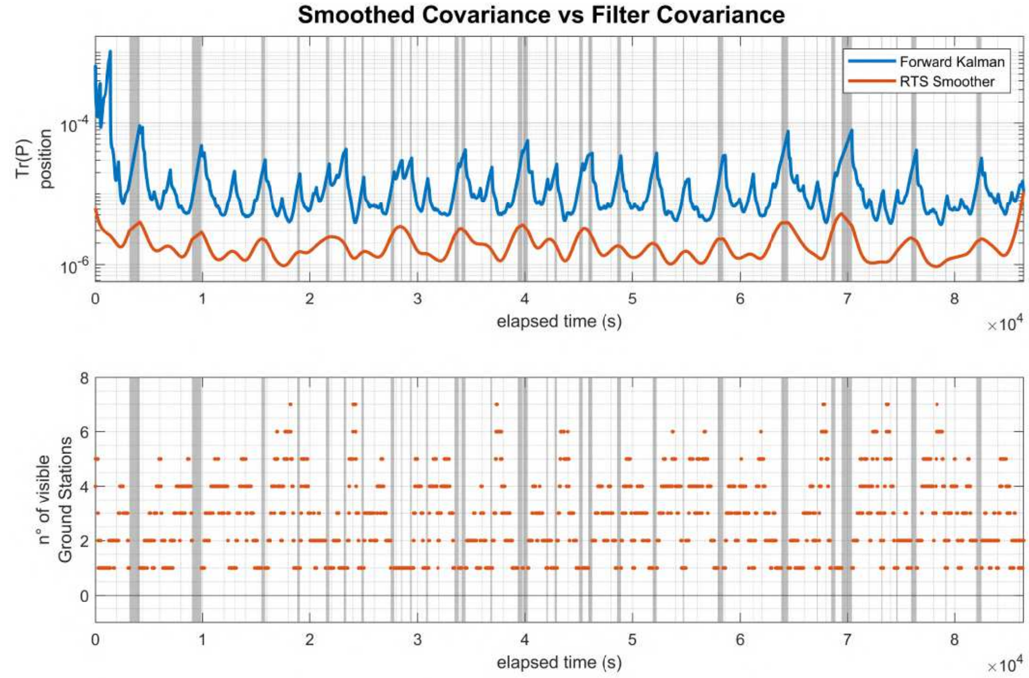

Figure 7.4.8: Effects of smoothing on the solution covariance matrix **P**. Here, the trace of the **P** matrix elements related to position is shown.

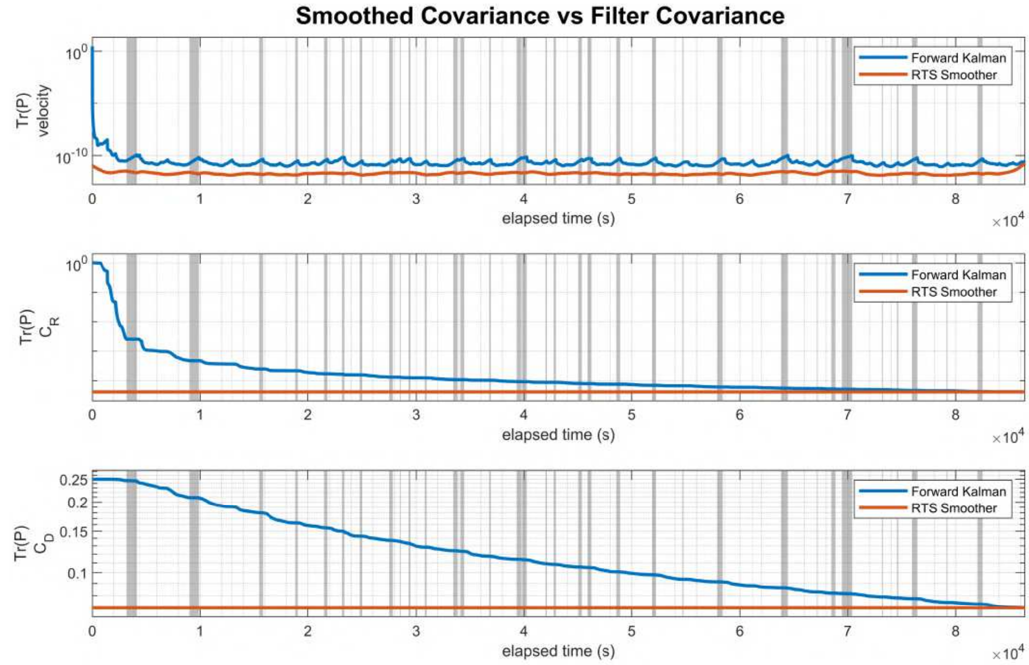

Figure 7.4.9: Effects of smoothing on the solution covariance matrix **P**. Here we can see the trace of the **P** matrix that are related to velocity and the parameters C_R and C_D. Note how the velocity is smoothable, being a dynamical state subjected to noise, while the other two parameters are not.

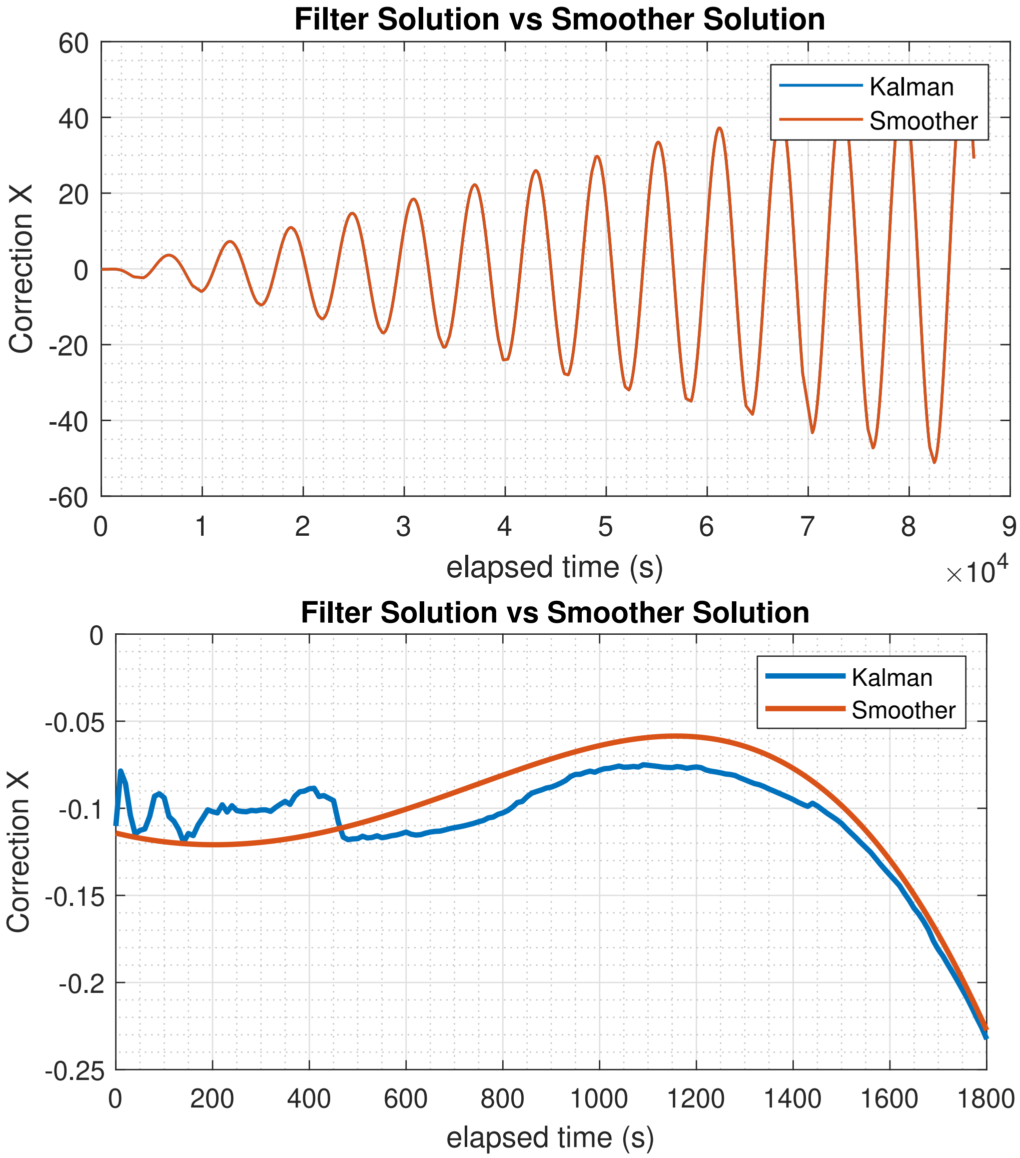

Figure 7.4.10: Filtered corrections vs Smoothed corrections. The figure on the bottom provides a detail of the whole image where it is possible to appreciate the effects of smoothing on the computed orbit corrections. Here, the X coordinate is considered as an example.

7.4.2 The Effect of Initial Conditions

The last relevant characteristic of the smoother estimate that is worth noting is actually a drawback inherently present due to the forward filter's nature. As has been well established, the forward Kalman filter is used in the *conventional* form, in contrast to the extended formulation where the reference orbit is update after each observation has been processed. Nonetheless, the extended Kalman filter has been described as a better alternative to the conventional one, as it is more resilient and stable in the presence of high nonlinearities in the mathematical model. This means that, since the various orbit determination techniques presented here are based on a first order linearization about a fairly near reference trajectory, one may expect the conventional Kalman filter's performances to worsen when initial conditions are highly perturbed or way off the true orbit. This is exactly what the following results show, where we can see that if the smoother was to be initialized with the same position and velocity perturbations used for the previous estimations, a clear divergence would arise over time. Figures 7.4.11, 7.4.12 and 7.4.13 show the observation and orbit residuals when the initial position and velocity perturbations are the same as the ones in table 7.2. The detrimental effect is clear as high residuals, on the order of kilometers, are present for the forward filter and even though the smoothed solution is still somehow better, it surely isn't satisfying since after one day it is still on the hundred-meters level. This consideration leads to the conclusion that smoothing generally improves the estimate in terms of precision, but requires the "guessed" initial conditions to be fairly near the actual ones. To generate the successful smoothed results in the previous section, the magnitude of both the position and velocity perturbations had to be decreased of about two orders of magnitude.

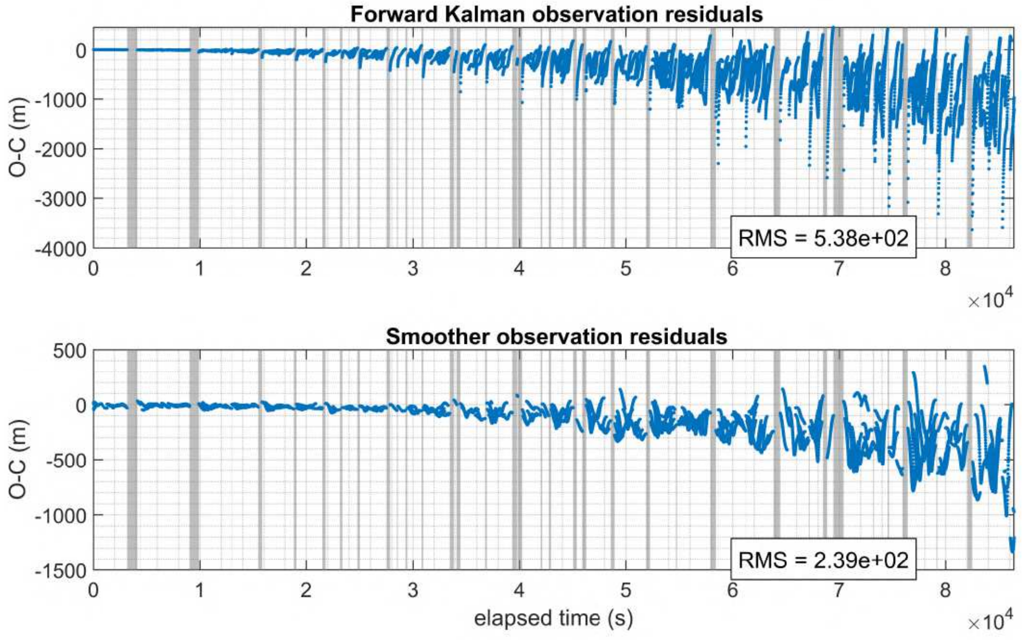

Figure 7.4.11: Effects of initial conditions on the forward Kalman filter and the backward recursive smoother. Clearly, deviations on the order of hundreds of meters in position cannot be handled by the forward conventional filter.

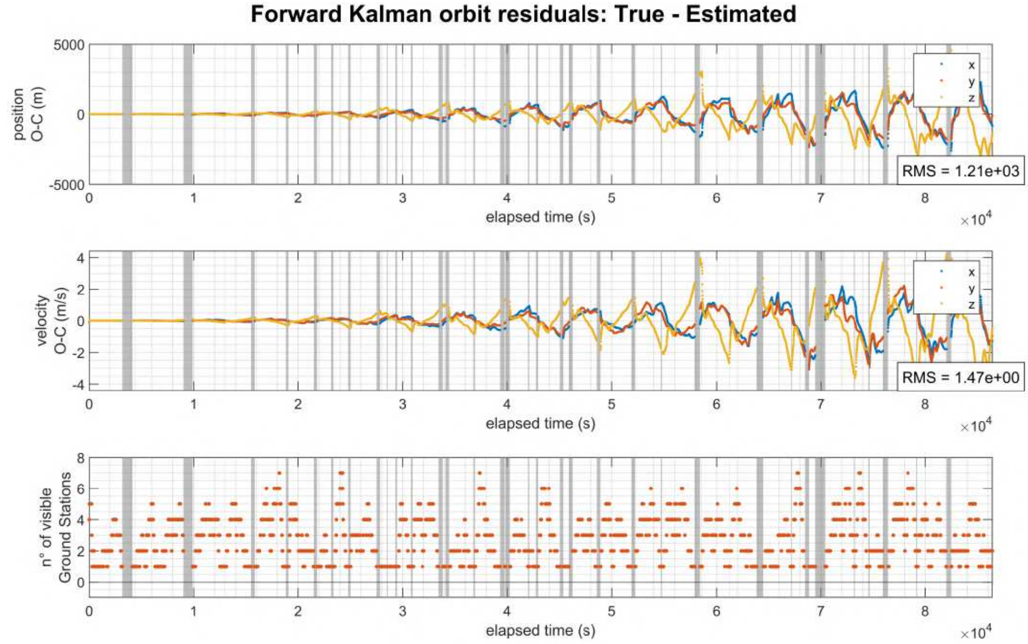

Figure 7.4.12: Effects of initial conditions on the orbit residuals of the forward conventional Kalman filter.

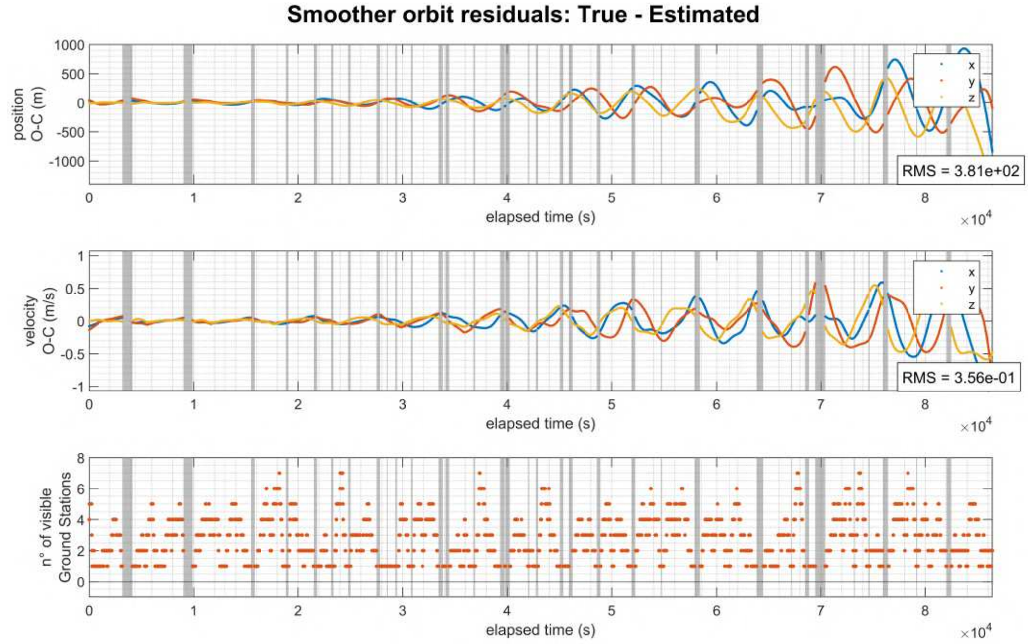

Figure 7.4.13: Effects of initial conditions on the orbit residuals of the backward RTS smoother.

Chapter 8

Conclusions and Future Work

This research work aimed to develop and test various algorithms for Precise Orbit Determination (POD) applications, specifically, the batch estimation, the Kalman filter, and the smoother. The results achieved are somewhat satisfactory as they clearly indicate that all the developed algorithms are able to perform orbit estimation to a sub-decimeter level, despite the force model incompleteness, the measurement noise and the perturbed a priori initial conditions. In particular, the batch estimation reaches observation residuals of about 6 cm, as well as few centimeters of RMS in orbital residuals. The Kalman filter and the smoother, on the other hand, arrive to the centimeter-level, outperforming the batch estimation in terms of observation and orbit residuals. In any case, all estimators show good stability as their solutions remain bounded during the estimation arc, and while convergence is excellent for both batch and EKF, that can overcome initial perturbation of many hundreds of meters, it is not entirely true for the smoother, where due to the presence of the forward conventional Kalman filter, perturbations to the initial conditions must be bounded to few meters in position and few millimeters per second in velocity. Nonetheless, the benefits of a smoothing procedure as a means of improving the estimate of a forward Kalman filter are undeniable, whereas the reduction in the solution covariance is evident as well as a slight reduction in residuals. These facts suggest that the force and kinematic models used to model the satellite dynamics are in good agreement with those that have been used to generate the available orbital data from which measurements are computed, although they are not yet perfectly coincident. In batch estimation, the parameters segmentation procedure showed promising results, improving the orbit determination in the case of a cannon-ball satellite model where the true attitude law is not modeled. It has been demonstrated that a good segmentation of the C_R parameter has resulted in a significant improvement in orbital determination performance, as well as a better representation of the accelerations acting on the satellite, with a visible similarity between the SRP profiles for the box-wing case and the segmented cannon-ball model. The simulations with the extended Kalman filter, in turn, have shown promising results. In particular, it was noted here how a good implementation of the process noise helps to avoid filter saturation and the consequent divergence of both observation and orbit residuals. Furthermore, the DMC approach allowed the estimation of the deterministic parameters of the process noise itself, recalling that for this case it is represented as a first-order Gauss-Markov process.

As for the batch estimation, future developments will certainly have to move towards the definition of a more refined force model, since it is clear that this type of estimator is particularly sensitive to even the smallest perturbation, especially when used on very wide estimation arcs. As a matter of fact, the weaker performance of the batch estimator in comparison to the filter and smoother might be mainly due to this fact alone. In this regard, it is first necessary to implement and appropriately calibrate the ocean tides model, which is currently not active in the various available Fortran subroutines. Furthermore, regarding the ERP modeling, the benefits of using a box-wing model for this acceleration should be investigated. From the point of view of parameter segmentation for the batch processor, it would be appropriate to evaluate the development of a piecewise linear model in addition to a piecewise constant one, whereas segmented parameters modeled as linear instead of constant can certainly improve the estimate performance without the need of a strong overparametrization. Other possible future developments should move towards a more realistic representation of the situation. In these terms, a *consider covariance* analysis approach should be deepened. The point is that the covariances obtained from an estimate carried out following the methods outlined in this research work are not a true index of the precision with which the state estimate is known, in most cases they are overly optimistic estimates. The consider covariance analysis not only admits that the estimation parameters are somehow in error, but assumes that even the non-estimated parameters are not known exactly. This ultimately leads to an increase in the covariance of the estimated parameters that is certainly more realistic.

In summary, future developments should move towards a more detailed representation of the forces acting on the satellite, moreover, the use of real measurements in contrast to synthetic measures could really give a decisive verdict on the performance of the developed Fortran routines for POD applications. However, it is not that simple and straightforward since real observational data significantly increase the level of difficulty and considerations. In any case, the results achieved here in a somewhat controlled environment of synthetic data, are motivating and have allowed to have a clear vision of the general picture on the precise orbit determination of Earth satellites.

Bibliography

[1] International gnss service (IGS). URL: `https://network.igs.org/`.

[2] S. Casotto. *Nominal ocean tide models for TOPEX precise orbit determination*. The University of Texas at Austin, 1989.

[3] S. Casotto. Introduzione alla meccanica celeste. Lecture Notes in Celestial Mechanics, 2012.

[4] M. Catania, C. Dietze, and J. Klein. Sentinel-3a flight dynamics LEOP operational experience.

[5] Francesco Darugna, Stefano Casotto, Massimo Bardella, Mauro Sciarratta, and Paolo Zoccarato. Sub-decimeter onboard orbit determination of LEO satellites using SSR corrections: A Galileo-based case study for the Sentinel-6a satellite. *Remote Sensing*, 14(23):6121, 2022.

[6] E. Fantino and S. Casotto. Methods of harmonic synthesis for global geopotential models and their first-, second and third-order gradients. *Journal of Geodesy*, 83:595–619, 2009.

[7] B. P. Gibbs. *Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook*. Wiley, 2011.

[8] D. S. Ingram. *Orbit determination in the presence of unmodeled accelerations*. The University of Texas at Austin, 1970.

[9] D. S. Ingram and B. D. Tapley. Lunar orbit determination in the presence of unmodeled accelerations. *Celestial mechanics*, 9(2):191–211, 1974.

[10] J. F. JORDAN JR. *Optimal stochastic control theory applied to interplanetary guidance*. The University of Texas at Austin, 1966.

[11] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[12] D. G. King-Hele. *Satellite Orbits in an Atmosphere: Theory and application*. Springer Netherlands, 1987.

[13] P. Knocke, J. Ries, and B. Tapley. Earth radiation pressure effects on satellites. In *Astrodynamics conference*, page 4292, 1988.

[14] R. Kroes. Precise relative positioning of formation flying spacecraft using GPS. *Publications on Geodesy, 61*, 61, 01 2006. `doi:10.1117/12.855923`.

[15] J. M. Longuski, F. R. Hoots, and G. E. Pollock. *Introduction to Orbital Perturbations*. Space Technology Library. Springer International Publishing, 2022.

[16] R. H. Lyon. *Geosynchronous orbit determination using space surveillance network observations and improved radiative force modeling*. PhD thesis, Massachusetts Institute of Technology, 2004.

[17] P.S. Maybeck. *Stochastic Models, Estimation, and Control*. ISSN. Elsevier Science, 1982.

[18] W. D. McClain and D. A. Vallado. *Fundamentals of Astrodynamics and Applications.* Space Technology Library. Springer Netherlands, 2001.

[19] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods, and Applications.* Physics and Astronomy online library. Springer Berlin Heidelberg, 2000.

[20] O. Montenbruck, P. Steigenberger, and U. Hugentobler. Enhanced solar radiation pressure modeling for Galileo satellites. *Journal of Geodesy*, 89:283–297, 2015.

[21] K. A. Myers. *Filtering theory methods and applications to the orbit determination problem for near-earth satellites.* The University of Texas at Austin, 1974.

[22] K. A. Myers and B. D. Tapley. Dynamical model compensation for near-earth satellite orbit determination. *AIAA Journal*, 13(3):343–349, 1975.

[23] C.J. Rodriguez-Solano, U. Hugentobler, P. Steigenberger, and S. Lutz. Impact of earth radiation pressure on GPS position estimates. *Journal of geodesy*, 86:309–317, 2012.

[24] B. Schutz, B. Tapley, and G. H. Born. *Statistical Orbit Determination.* Elsevier Science, 2004.

[25] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* Wiley, 2006.

[26] R. F. Stengel. *Optimal Control and Estimation.* Dover Books on Mathematics. Dover Publications, 2012.