Pelayo Peñarroya Rodríguez

Investigation of Convex Residual Penalty Functions for Orbit Determination

MSc Thesis

Challenge the future

Investigation of convex residual penalty functions for orbit determination

MSc Thesis Report

By

Pelayo Peñarroya Rodríguez

4518810 – Space Exploration

in partial fulfilment of the requirements for the degree of

Master of Science

in Aerospace Engineering

at Delft University of Technology,

September 2017

Abstract

This thesis aims to assess how different convex penalty functions can be used in orbit determination methods and to design an algorithm to test them under different conditions.

Most traditional Precise Orbit Determination (POD) algorithms use the Least-Squares (LSQ) method to minimise the misfit between a set of modelled and actual measurements. The approach followed in this research is to investigate other convex penalty functions for this purpose in order to achieve better results than the traditional LSQ method, while maintaining the overall quality and robustness of the former.

To simplify the application of convex optimisation methods, an external toolbox was used to implement the different convex cost functions. The testing environment included a Precise Orbit Propagator (POP), measurement generating and processing functions, the POD algorithm itself, and data processing functionalities for the representation of the results. All the different components integrated in the final algorithm were validated before their application.

Tests to assess different aspects of the implemented penalty functions were run, regarding both computational aspects and solution performance. The traditional method was observed to present suboptimal results when the noise present in the observations included unprocessed outliers. In addition, cases where the observations were highly sparse yielded a suboptimal estimation of the trajectory.

After that, the L1-Norm was implemented as penalty function, alongside with Huber’s penalty function, which represents a combination of both LSQ and L1-Norm.

The use of the L1-Norm in the orbit determination algorithm outperformed the traditional method in the cases where it lacked performance, such as in the presence of unprocessed outliers or sparse observation sets. However, in other tests run, the LSQ algorithm was able to reach higher accuracy levels than the L1-Norm. Huber’s penalty function, conversely, proved to be a great candidate for both purposes, closely resembling the results obtained by the best penalty function for each test and even improving it on occasions, at the cost of a higher computational effort.

Finally, the designed algorithms were applied to a real-world study-case making use of GOCE data provided by TU Delft. These applications demonstrated satisfactory performance for each of the methods that were implemented and provided an important validation of the work.

Keywords: convex functions, penalty functions, robustness, observations, orbit determination.

Contents

LIST OF SYMBOLS

$\boldsymbol{a}$	Acceleration $[m/s^2]$
ω	Argument of the Perigee [°]
$\boldsymbol{x}$	Cartesian State Vector
$(\boldsymbol{r}, \boldsymbol{v})$	Components of $\boldsymbol{x}$: Position and Velocity $[m, m/s]$
Q_y	Covariance Matrix of the Observations
Q_x	Covariance Matrix of the Parameters
S	Cross-Section or Wet-Area $[m^2]$
ρ_∞	Density of the Air at Rest $[kg/m^3]$
H	Design Matrix
c_D	Drag Coefficient [-]
e	Eccentricity [-]
e_r, e_t	Errors in the Radial and Tangential Directions
X^*	Estimated Parameters
$\boldsymbol{F_{net}}$	External Force $[N]$
$J_n, J_{n,m}, \phi_{n,m}$	Gravitational Model Parameters [- , °]
μ	Gravitational Parameter $[m^3/s^2]$
G	Grubbs' parameter
M	Huber´s Penalty Function´s Adjusting Parameter [-]
i	Inclination [°]
m	Mass $[kg]$
ρ	Misfit
A	Model Matrix
Y	Observations
y	Observations´ Residual
T	Orbital Period $[s]$
x	Parameters´ Residual
Φ	Penalty Function
$\boldsymbol{\Delta X_0}$	Perturbation added onto the Initial State (Validation)
W	Radiation Flux $[kg/s^3]$
R_i	Radius of i. Usually Earth (R_e), or Sun (R_s) $[m]$
X	Reference Parameters
c_R	Reflectivity Coefficient [-]
Ω	Right Ascension of the Ascending Node [°]
$S, N_{obs}, Pf.$	Scaling Factors
a	Semi-Major Axis $[m]$
I_{sp}	Specific Impulse $[s]$
(r, ϕ, λ)	Spherical Coordinates: Radius, Latitude, and Longitude $[m, °, °]$
σ	Standard Deviation
$\boldsymbol{X_0}$	Trajectory from Initial State (Validation)
$\boldsymbol{X_{0,pert}}$	Trajectory from Perturbed Initial State (Validation)
θ, E, M	True, Eccentric or Mean Anomaly [°]

LIST OF ACRONYMS

AE	Azimuth-Elevation
BCRF	Barycentre-Centred Reference Frame
CO	Convex Optimisation
CP	Conic Programming
DORIS	Doppler Orbitography and Radiopositioning Integrated by Satellite
ECEF	Earth-Centred Earth-Fixed
EGM	Earth Gravity Model
EoM	Equations of Motion
EOP	Earth Orientation Parameter
ERS	European Remote Sensing
ESA	European Space Agency
GEO	Geostationary Earth Orbit
GESD	Generalized Extreme Studentized Deviate
GLONASS	Global Navigation Satellite System
GNSS	Global Navigation Satellite Systems
GOCE	Gravity Field and Steady-State Ocean Circular Explorer
GPS	Global Positioning System
GRACE	Gravity Recovery and Climate Experiment
GTO	Geostationary Transfer Orbit
HPF	Huber´s Penalty Function
IAU	International Astronomical Union
ICRF	International Celestial Reference Frame
ICRS	International Celestial Reference System
IERS	International Earth Rotations and Reference Systems Service
IGLOS	International GLONASS Service Pilot Project
ILRS	International Laser Ranging Service
ITRF	International Terrestrial Reference Frame
JD	Julian Date
JPL	Jet Propulsion Laboratory
KKT	Karush-Kuhn-Tucker
LAGEOS	LAser GEOdynamic Satellite
LEO	Low Earth Orbit
LP	Linear Programming
LSQ	Least-Squares
MEO	Medium Earth Orbit
NASA	National Aeronautics and Space Administration
NGA	National Geospatial-Intelligence Agency
NOVAS	Naval Observatory Vector Astrometry Software
NRL	US Naval Research Laboratory
POD	Precise Orbit Determination
POP	Precise Orbit Propagator
PVT	Position, Velocity, Time
QP	Quadratic Programming
RA	Right Ascension
RDOD	Reduced-Dynamics Orbit Determination
RK	Runge-Kutta
RMS	Root Mean Square
RTN	Radial, Tangential, Normal
SLR	Satellite Laser Ranging

SM	Sensitivity Matrix
SOCP	Second-Order Conic Programming
RSO	Resident Space Object
SRP	Solar Radiation Pressure
STM	State Transition Matrix

PREFACE

Before you lies the thesis "*Investigation of convex residual penalty functions for orbit determination*", a research focused on assessing how different convex penalty functions can be used in orbit determination methods and designing, if feasible, an algorithm to test them under different conditions.This research, spanning a duration of seven months starting from February 2017 and finishing in August 2017; was conducted to fulfil the graduation requirements for the MSc programme in Aerospace Engineering offered by TU Delft.

This MSc thesis project took place in Bremen, by means of a collaboration agreement between TU Delft and OHB System, company in which I had already spent almost six months for an internship period. During this internship, the topic was proposed within the same team where I was allocated (Mission Analysis) and I decided to take part in this exciting trip almost immediately. The line of research was practically unexplored at that time, what opened windows for very important results, but was also reasons for doubt.

Fortunately, my tutors from TU Delft and OHB were really helpful and perfectly knew how to guide me when I was lost. Thus, I would like to thank Pieter Visser for the patience during the phone conferences and for providing me with excellent pieces of advice (as well as many papers in the past years) on the topic. Also the Mission Analysis team for always helping when needed, and especially, my supervisor at OHB: Ferdi de Bruijn. I really look up to him after this period in the company (including the internship), and I hope I could help in the professional development of someone in the future as much as he has in my case. This thesis would not be what it is without his guidance.

I would also like to thank my family for providing for me during these years of academic training and for their overwhelming support. Especially to my parents, I know sometimes I am difficult to counsel and stubborn but I could not be more grateful for what I am today, and it is all thanks to you. You are the best.

Last but not least, I would like to thank the person who joined me in this adventure almost one year ago but feels like an extension of myself today, my girlfriend Inés. Without her listening to all my frustrations and supporting me in every step of the road, I am certain this would have never happened. You make me better every day.

I hope you enjoy your reading.

Pelayo Peñarroya Rodríguez

Bremen, September 2017

“Scientific truth is beyond loyalty and disloyalty.”

Isaac Asimov, Foundation

1 Introduction

Precise Orbit Determination (POD) is currently one of the most used technologies in the satellite industry and, even if it has been used for many decades already, it keeps evolving and improving its results. Following this line, and including newer concepts like Convex Optimisation (CO), this research is meant to work towards a further development in such an important field by combining POD and CO. In this chapter, these two fields will be introduced and, afterwards, their research opportunities will be explained.

1.1 Previous Work

The content of this thesis will be based on the current state of the technology involved, and will build new concepts upon that. A literature study was carried at the beginning of the research and its main conclusions are summed up in this section, classified in two parts: POD and CO.

1.1.1 Precise Orbit Determination

POD is based on the combination of dynamics and orbital data to obtain a statistical orbit determination leading to very high accuracy levels. Once these data are gathered, there is usually a ground segment (sometimes on-board) post-processing that yields the desired results, which are combined once again with the rest of the operational instrumentation of the satellite -such as altimeters, cameras, etc. – to refine and produce more accurate final products. POD was one of the first activities in which the Global Navigation Satellite Systems (GNSS or GLONASS[1]) were involved (Dow, Neilan, Weber, & Gendt, 2007). These navigation systems provide a highly valuable source of (pseudo-)range observations for satellite orbiting the Earth in a Low Earth Orbit (LEO), for instance (Cerri, et al., 2010) (Visser P. N., 2007). With this information, the quality of the final products given by Earth observation satellites increased abruptly, meaning enhancements in the geodetic models, the altimetry data, or the imagery products, among others.

POD has been applied to many missions during the last decades including the ERS (European Remote Sensing) satellites (Visser & Scharroo, 1998), the GRACE ((Gravity Recovery and Climate Experiment) mission (Lemoine, et al., 2007), GOCE (Gravity Field and Steady-State Ocean Circular Explorer) mission (Visser P. N., 2007) (Bock, et al., 2007), or the LAGEOS (LAser GEOdynamic Satellite) satellites (Lucchesi, 2007), among others. However, in a more general sense, almost every mission needs POD (or at least orbit determination).

One of the first applications was the TOPEX/POSEIDON Mission (Tapley, et al., 1994), which was an oceanographic mission intended to improve the level of accuracy to which the geographical accidents underneath the oceans were known. It carried two experimental instruments: a single-frequency altimeter and a GPS (Global Positioning System) receiver (Fu, et al., 1994). The mission objective was to know the radial position of the spacecraft within an error of 13 cm RMS (Root Mean Square), but, thanks to the effort invested by the POD team composed by the Universities of Texas (Austin) and Colorado, the NASA (National Aeronautics and Space Administration) Goddard Space Flight centre, and the JPL (Jet Propulsion Laboratory), it was possible, not only to fulfil the latter, but to achieve an accuracy of 4.7cm and 5.1cm for TOPEX and POSEIDON (for singles passes), respectively. Since the requirement was to reduce the error to 13.7cm, the technology was validated and adopted by many missions after TOPEX/POSEIDON.

[1] Note that when the acronym GLONASS is used, it refers to the Russian alternative to GPS.

JASON-1 was the heir of the TOPEX/POSEIDON's legacy. Again, POD was vital to achieve the accuracy requirements, which were set to 1 to 1.5 cm RMS (Cerri, et al., 2010). JASON-1, and its follow-on mission JASON-2/OSTM, carried on-board three state-of-the-art tracking systems: DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite), GPS, and SLR (Satellite Laser Ranging); in order to achieve the mentioned accuracy. Additionally, altimeter crossover measurements (Tapley, Schutz, & Born, Differenced Altimeter Data, 2004) were taken to allow for more accurate results. The combination of these measurements led to a reduced-dynamics solution that took the accuracy towards the 1 cm level (Luthcke, Zelensky, Rowlands, Lemoine, & Williams, 2010). This exemplifies the narrow relation between the positioning technologies and the effectiveness of the POD techniques.

POD techniques continued spreading and (Dow, Neilan, Weber, & Gendt, 2007) explained how they were planning to include the benefits of POD inside the Galileo mission frame. According to their research, the Galileo constellation would be useful for Earth Orientation Parameter (EOP) science (Altamimi, Boucher, & Willis, 2005), in the International GLONASS Service Pilot Project (IGLOS), see (Springer, Gendt, & Dow, 2007) or (Weber, Slater, Fragner, & et al., 2005); or in projects including the International Laser Ranging Service (ILRS) (Pearlman, Degnan, & Bosworth, 2002). At the same time, the fundamental geodetic network in Russia was being built up using GPS. GLONASS (Sergey, Sergey, & Suriya, 2007) utilized this technique to refine the satellite data they already had using the GIPSY-OASIS II software provided by (Webb & Zumberge, 1995).

Apart from the mathematical methods to be used when it comes to data processing, there have been also multiple studies about how to use POD techniques for specific cases, for instance, orbital prediction for Resident Space Objects (RSO) near Geostationary Earth Orbits (GEO) (DeMars, Jah, Giza, & Kelecy, 2010). These researches would have to be taken in account because, as they explain, different case studies require different uses of POD techniques, depending on the differencing of the GPS carrier-phase or the use (or not) of the ambiguity resolution (comparisons are shown in (Svehla & Rothacher, 2003) or (Jäggi, Hugentobler, Bock, & Beutler, 2007)).

1.1.2 Convex Optimisation

CO is a subfield of general optimisation that deals with convex problems. This kind of problems is "easier" to solve because, for instance, any local minimum must be a global minimum, making the convergence of the algorithm much more robust and reliable. Most of the effort put in CO problems lies in translating general and complex problems into convex and smoother ones. Once this process has taken place, the problem can usually be solved without major hindrances (Boyd & Vandenberghe, 2009).

CO techniques are on the rise these days and their implementations to multiple fields have yielded excellent results in several topics such as *Rendez-Vous* (Bhagat, 2016), re-entry trajectories (Xinfu, Shen, & Lu, 2015), or trajectory optimisation (Liu, 2013), among others in space sciences. They have been proven to be a superb tool for many different applications, letting highly complex problems transform into much more approachable cases, whose final solution is also a solution for the original case (Boyd & Vandenberghe, 2009).

Most recently, (de Bruijn, 2017) carried a research on how CO techniques could be applied to guide and control geostationary satellites and on how such a method could be used to collocate a number of satellites under geometric constraints. The results were very satisfactory, serving as another example of CO techniques being applied to space-related problems.

1.2 Motivation

Nowadays, POD applications are still on the verge of technology and space missions, and their use is practically required for any Earth-sensing missions or other accuracy-based applications (Cerri, et al., 2010). Because of that, this research represents an excellent chance both for academic and industry-related fields, lending an opportunity to include CO techniques in the POD methods. The results would offer new insight into this theory and, perhaps, the possibility of an improved implementation and outcome for future space missions, and for the data processing of currently operated spacecraft.

The goal of this project is to develop and evaluate new methods that, combining POD with CO techniques, could lead to more robust and/or efficient algorithms compared to traditional methods. For this, the analysis will be focused on the cost (or penalty) functions involved in orbit determination algorithms.

Summing up, in this work, the starting point is set on top of all the previous developments gathered in the referenced bibliography. From there, an investigation on how CO functions and techniques could be applied to the penalty functions involved in many data processing methods will be conducted. This approach differs to what most authors have investigated in the previous years, who were mainly focused on the final results or on the statistical method involved. Here, a comparison will be offered between different penalty functions and how can they affect the robustness and performance of POD algorithms.

1.2.1 Research Questions and Methodology

Now, the starting point has been identified and the environment around the fields involved has been set up, in this section, a representative research question will be introduced and broken down into smaller sub-questions that will help distributing the different tasks to be undertaken during the investigation as well as establishing priorities for them.

The research question could be posed as:

Can CO techniques, when applied to POD algorithms' penalty functions, lead to an enhancement over the current technology and offer more robust and/or efficient results? If so, how can this be done?

Or alternatively:

How can convex optimisation techniques be applied to POD problems and what are the merits of the respective solutions in comparison to solutions resulting from conventional weighted least-squares methods in terms of efficiency, accuracy and robustness?

From this question, some more specific sub-questions can be developed:

- What are the main drawbacks of the currently used penalty functions in POD?
- How can convex cost functions be applied to POD algorithms in an efficient and robust way?
- If feasible, how do these penalty functions perform in comparison to the current state of the art?

By answering these three sub-questions, it will be possible to determine which aspects are to be improved and whether CO techniques are the way to do it.

The research question poses the challenge not only to demonstrate whether the application of CO techniques to POD problems is feasible, but also to find how this can be done. The development of such a method would represent truly valuable tools that could be directly applied to the space industry.

The objective of this research is to investigate whether the use of CO techniques in POD penalty functions can outperform in terms of accuracy, robustness, and/or efficiency the current technologies used by the industry and develop (if feasible) such a tool within the time-frame of this MSc Thesis.

With this objective, the workload can be distributed into different tasks to be undertaken. These are shown in Table 1.1. At the end of the project, if the feasibility analysis is positive, the creation and validation of a computational tool should be regarded.

Table 1.1 - Tasks to be completed during the thesis research.

Kinematics and Dynamics	• Development of a Precise Orbit Propagator (POP).
POD	• Implementation of measurement simulation and measurement processing function. • Implementation of relevant measurements models. • Implementation of weighting on measurements accounting for the confidence on the measurements.
CO	• Study CO applications to POD. • Implementation and comparison of orbit determination algorithms with different residuals penalty functions. • Evaluation and comparison of robustness and accuracy.

About the methodology involved in this research, a hybrid structure will be used, combining different resources depending on the current step of the research. First of all, the link between POD and CO will be identified, mainly by desk-researching to establish the starting point. Afterwards, using an experimental approach, the different CO functions that may be available will be compared and the results will be assessed. If their feasibility is proved, the use of different case studies will be necessary to validate the developed theory and proof its robustness.

1.3 Roadmap

To provide a structured and clear path to the reader, this document is organised in several chapters, ordered content- and time-wise.

Chapter 2 is included to serve as a brief introduction of the theoretical background, as well as the environmental set-up used along this thesis. The reader may refer to it for simple clarifications but further doubts should be checked in the referred bibliography.

After that, Chapter 3 will offer the first testing results, which will not be so focused on the final trajectory estimations from a realistic perspective but from a more computational sense. Aspects such as the validation of the computational capabilities of the tools involved or the proper understanding of the included technology will be the focus in this chapter.

Chapter 4 will contain more realistic tests with more practical applications. The algorithmic and computational part is considered validated at this point, and emphasis is made on the accuracy of the results and on how well the latter fit the expectations. The outcome of these tests is expected to be a relatively good overview of how the different penalty functions face different typical real-world situations.

Once all these tests have been run, Chapter 5 will serve as a proof of all the conclusions gathered by implementing real tracking data (from the GOCE mission) in the developed algorithm. The solutions using different CO methods will be compared with each other.

Finally, a last chapter including the conclusions obtained and the possible improvements will be added as the final point of this report. References and annexes can be found after that.

It is recommended that the Glossary is read before going through the body of the report.

2 Theoretical & Technical Environment

As it was already introduced, this research includes terms from different fields; mainly POD and CO, but also general concepts from orbital mechanics, for instance. In addition, for the different validation tests performed, a basic knowledge about reference systems is required. Also, information about the simulation environment and conditions could be of interest for the reader. Finally, a rationale for the tests to be run in future chapters will be included.

2.1 Reference Systems

For any kind of positioning method there is always a need for a reference in order to establish consistent and logic spatial distributions. A reference frame, in other words, is the abstract coordinate system and the physical references that fix its location and orientation in order to standardise measurements.

A reference system, differently from a reference frame, is the set of physical definitions that ideally define a coordinate system. As the human knowledge about the universe and its motion becomes larger, the definition of such systems varies in order to fit the best basis.

On the other hand, a reference frame is the realization of a certain reference system, so to say, the best representation of the ideal reference system that the current technology is able to obtain.

Once a reference system has been adopted, it is possible to classify its reference frames into two main types: (pseudo-)inertial and non-inertial.

In the following, some of the reference frames used throughout this document will be briefly described.

2.1.1 Inertial Reference Frames

An inertial or pseudo-inertial reference frame is the one that is defined to be under the influence of no other accelerations and is at rest or moves along a straight line. Theoretically there is no inertial reference frame, since there will always be some kind of movement (NASA, 2016) in the reference points no matter the further away they are. However, quasi-inertial reference frames are those whose apparent accelerations can be completely ignored for human purposes and timescales. This is reflected in the evolution of the reference systems mentioned before; as time goes by, technology grants higher accuracies and what used to be a quasi-inertial reference system, is not anymore. From this point on, when the term inertial reference frame, this concept should be taken into consideration.

2.1.1.1 International Celestial Reference Frame

The International Celestial Reference Frame (ICRF) is the realization of the ICRS (International Celestial Reference System), which was adopted as the official system to follow in 1997 by the International Astronomical Union (IAU), and based on its first realization of 1995. After that, the maintenance process has led to continuous corrections meant to make the realization closer to the system definition.

This reference frame is centred in the barycentre of the Solar System, this is the reason why is also known as Barycentre Centred Reference Frame (BCRF), with its axes fixed with respect to space. Its principal plane was defined by the International Earth Rotations and Reference Systems Service (IERS) to be *"close to the mean dynamical equator at J2000.0 epoch"* (IERS, 2010) (1st January 2000 at 12.0h, or Julian Date 2451545.0). This definition gives the Z-axis direction. Regarding the X-axis, the IAU recommends that "*the origin of the right ascension (RA) should be close to the dynamical equinox of J2000.0*" (IERS, 2010), what means that the axis should be pointing to the vernal equinox of J2000.0, which is the intersection of the celestial equator (see Section 2.1.2) and the solar ecliptic, i.e., the line projected onto the celestial sphere that describes the plane that contains the Earth and the apparent trajectory of the Sun around it. The Y-axis completes the orthonormal trihedron.

Because the distance from the Earth to the barycentre of the Solar System is tiny compared to the distance to the quasar radio sources, no rotation is needed to translate this system towards and Earth-centred definition.

The difference between this system and the dynamical references, which the ICRS is recommended to be close to, is that the ICRS is defined with respect to quasars instead of stars as the dynamical frame (a.k.a. J2000) was.

The last realization of this system (called ICRF-2) is defined by more than 3400 compact radio sources and maintained by a special set of 295 sources with good positional stability and unambiguous spatial structure. This results in a noise floor of only 40 μas and an axis stability of 10 μas (much better than the previous realization, ICRF-1, that had 250 and 20 μas respectively). This realization allows great improvements in spacecraft navigation and a better control of the EOPs.

2.1.2 Non-inertial Reference Frames

On the other hand, non-inertial reference frames are those that show accelerations acting on them. These accelerations have to be expressed as apparent forces and included in the equations for accelerations acting in this frame. On Earth, there are some of these auxiliary forces that have to be taken into account. The most commonly applied are Coriolis, Euler, or the centrifugal acceleration (Arnol'd, 1997) (Taylor, 2005).

2.1.2.1 Earth Centred Earth Fixed

The Earth Centred Earth Fixed (ECEF) is the most frequently used non-inertial reference frame for Earth-related purposes. This frame, a.k.a. International Terrestrial Reference Frame (ITRF), represents a right-handed Cartesian reference system. Its origin is placed in the Earth's centre of mass and is called geocentre. From that point, the X-axis points towards the Greenwich Meridian, which represents 0° longitude on Earth coordinates. The Z-axis is directed along the Earth's mean rotational axis of the planet, this is, it points to the mean pole of the Earth's rotation. The need for such a designation is impelled by the existence of periodical perturbations to the Earth's rotational axis direction. These perturbations, among which nutation, and precession can be found, are explained more in detail in (Wakker, 2015) and (de Pater & Lissauer, 2007). The Y-axis is directed to complete the right-handed system.

The plane formed by the XY directions is called mean equatorial plane and the XZ-plane is called the mean zero-meridian. The projection of the mean equatorial plane onto the celestial sphere is called celestial equator, and the intersection of this point with the Solar ecliptic plane is called Vernal Equinox, which has been addressed before as the reference point for the X-axis in some inertial frames.

Transformations between this frame and the ICRF need to account for different effects such as nutation, precession, or polar wander (NOVAS, 2011) (IERS, 2010).

2.1.2.2 Spherical Coordinates

This coordinate system is only a different representation of the ECEF frame. In order to represent the position of a point in spherical coordinates, three parameters for the allocation need to be introduced:

- r: radius of the point (x, y, z), i.e., the norm of the position vector.
- ϕ: geocentric latitude, i.e., the angle that the position vector forms with the mean equatorial plane.
- λ: geocentric longitude, i.e., the angle that the position vector forms with the mean zero-meridian plane.

These parameters are referred to the centre of the sphere and are denominated geocentric coordinates (for the Earth).

2.1.2.1 Radial, Tangential, Normal

The Radial, Tangential, Normal reference frame (RTN) is centred on the centre of mass of the spacecraft and is composed by a radial vector (pointing outwards the Earth) in the X-axis, the angular momentum vector of the orbit in the Z-axis, and the Y-axis in the resulting cross product (in the direction of the velocity for circular orbits). The name stands for Radial, Tangential and Normal, but is also usually known as *Hill* or *Orbital Frame*.

2.2 Orbital Mechanics

In this section, a brief introduction of the orbital dynamics used throughout this thesis will be given. It will be classified in two sections regarding the formulation and the content of the different perturbation modules used: equations of motion and force models.

2.2.1 Equations of Motion

The Equations of Motion (EoM) are the mathematical reflection of the trajectories and states that a certain body will have throughout time. There are several state representations in which they can be described. Two of the more usual ones are Cartesian and Keplerian forms. The Cartesian form will express the state of the spacecraft (or any other celestial body) in terms of its coordinates according to a Cartesian reference frame (see Section 2.1). The Keplerian form, on the other hand, will represent the orbital state in terms of orbital elements of the body at that precise moment in time. The unperturbed, or *Keplerian,* orbits will be treated first. How the perturbing forces are introduced in the equations of motion will follow afterwards.

2.2.1.1 Keplerian Orbits

Using the Cartesian form, the state vector is defined by

$$\boldsymbol{x} = \left(x, y, z, v_x, v_y, v_z\right) = (\boldsymbol{r}, \boldsymbol{v})$$

where the first three elements ($\boldsymbol{r}$) represent the position and the three last ones the velocity ($\boldsymbol{v}$).

Taking the derivative of the state vector with respect to time leads to expressions for velocity and acceleration. As velocity is already the derivative with respect to time of position, the acceleration can be written as:

$$\boldsymbol{a} = \frac{d^2\boldsymbol{r}}{dt^2} = \frac{\boldsymbol{F_{net}}}{m} \tag{2.1}$$

where $\boldsymbol{F_{net}}$ is the net external force on the body and m its instantaneous mass. It should be highlighted here that this equation holds for inertial reference frames, where no apparent forces appear.

The model for gravitational accelerations in unperturbed orbits is given by:

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{2.2}$$

which is based on the potential model

$$U = -\frac{\mu}{r} \qquad\qquad \ddot{\boldsymbol{r}} = -\nabla U \tag{2.3}$$

In the equations above, μ is the gravitational parameter of the orbited body and $\ddot{\boldsymbol{r}}$ is the second time-derivative of the position, i.e., the acceleration.

On the other hand, the Keplerian approach is a much more geometric way to identify the state of an orbiting body. It is based on the polar equation for position

$$r = \frac{a(1-e^2)}{1+e\cos\theta} \tag{2.4}$$

where the orbital elements have already been implemented. A derivation for this can be found in (Wakker, 2015). In a similar fashion to what was explained in the previous section, the velocity and acceleration expressions can be obtained by taking the derivative of (2.4) with respect to time. For this derivative the only time-dependent variable is θ, since these orbits are unperturbed, where none of the orbital elements changes along the trajectory.

From (2.4), it is simple to obtain other relevant orbital equations such as the orbital period or the Vis-Viva equation, which is a reflection of the equation of energy (Wakker, 2015):

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \qquad\qquad v^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{2.5}$$

2.2.1.2 Perturbed Orbits

Perturbed orbits are those that take into account not only the acceleration of the main body as a point-mass but other perturbing forces that act upon the body. This can be expressed, from (2.2), as

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{f} \tag{2.6}$$

where $\boldsymbol{f}$ represents the perturbations that appear in the trajectory of the body. In the scope of this document, perturbation refers to any force, apart from the point-mass acceleration of the main orbited body that acts upon the orbiting body. The next section will deal with the most relevant perturbations that should be modelled in order for the results to be as close as possible to reality.

2.2.2 Force Models

This section will introduce some of the most important perturbations that should be taken into account when modelling an orbital trajectory. It is relevant to remark that these accelerations are given by equations that try to model the actual perturbations and, thus, they give (within a very high limit of accuracy) estimations of these.

Relativistic effects have been omitted in this section, as assumed also in (Wakker, 2015). This is due to the fact that the values for these perturbations, that affect mainly the argument of the pericentre, are in the order of magnitude of seconds of arc per century. This value is way below the errors that the other force models present, and, therefore, their effects could actually not be distinguished from the errors introduced by other (more relevant) models. The perturbations that are considered will be briefly explained in the following.

2.2.2.1 Irregular gravity

Gravity is the main driver of the orbital accelerations. As it can be observed in (2.2), the main force acting upon the orbiting the body is the gravitational pull that the orbited body exerts. This force can be modelled according to different levels of resolution. Initially, the model shown in (2.2) could be used, which corresponds to a Newtonian inverse-square expression that represents a perfectly uniform and spherical gravity force emanating from a single point. But as the accuracy requirements become more demanding, the Earth cannot be considered as a point-mass.

For this reason, models are created to try to model a geometry that could fit as close as possible the actual mass distribution of the Earth. These models are called *spherical harmonics* and consist of a list of coefficients that are combined with sinusoidal expressions to obtain a better approximation of the mass distribution of the Earth.

$$U = -\frac{\mu}{r}\left[1 + \sum_{n=2}^{\infty}\left(\frac{R}{r}\right)^{n} J_n P_n(\sin\theta) + \sum_{n=2}^{\infty}\sum_{m=1}^{n}\left(\frac{R}{r}\right)^{n} J_{n,m} P_{n,m}(\sin\theta)\cos m\big(\phi - \phi_{n,m}\big)\right] \qquad (2.7)$$

The equation above can be compared to that of the potential for point-mass distribution (2.3) shown before to observe that it keeps that term and adds a series of variations up to a certain degree and order (*n, m*). In this equation, (r, ϕ, θ) are the spherical coordinates of the point under consideration, R is the mean equatorial radius of the body (Earth, usually), and J_n, $J_{n,m}$ and $\phi_{n,m}$ are model parameters. Also, $P_n(x)$ and $P_{n,m}(x)$ are Legendre functions of the first kind defined as:

$$\begin{aligned} P_n(x) &= \frac{1}{(-2)^n n!}\frac{d^n}{dx^n}\,(1-x^2)^n \\ P_{n,m}(x) &= (1-x^2)^{m/2}\,\frac{d^m}{dx^m}\big(P_n(x)\big) \end{aligned} \qquad (2.8)$$

As (Wertz, 2009) states, when applied to Earth irregular gravity field, there are some considerations to be taken with these equations:

- Equation (2.7) is only valid for $r > R_e$, i.e., points outside the Earth's surface.
- The index n starts at 2 because the system is defined such that the centre of coordinates is the centre of mass and therefore both J_1 and $J_{1,1}$ are null (Wakker, 2015) (Montenbrück & Gill, 2000).
- The term J_2 is three orders of magnitude larger than the rest and sometimes the effects of the others can be disregarded.
- The second and third terms of equation (2.7) refer to north-south direction (zonal harmonics) and to both east-west and north-south direction (tesseral harmonics). When $n = m$ the third term only represents east-west harmonics (sectorial harmonics). This is pictured in figure below.

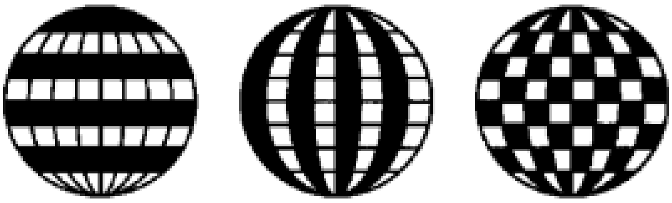


Figure 2.1 - Zonal (left), sectorial (centre), and tesseral (right) harmonics. (Wakker, 2015)

For other planets or bodies, the coefficients change but the method can be used in a similar fashion. In this thesis, the model used for Earth is the EGM2008 (Earth Gravity Model 2008), defined in (Pavlis, Holmes, Kenyon, & Factor, 2012).

2.2.2.2 N-body gravitation

It has already been seen that gravitational acceleration is actually not a point-mass force exerted on the orbiting body but an integral of all the mass particles of such a main body exerting their individual pull over the orbiting body. This concept, if extended, leads to the N-body gravitational acceleration. Every massive body exerts acceleration over the bodies around it. It has been shown in the previous sections that the further away from the centre of mass of a body, the smaller the effect in the perturbing acceleration (at inverse-quadratic proportion). On the other hand, the higher the mass (contained in the gravitational parameter) the higher this effect (directly proportional). For this reason, when considering an orbiting body, other massive bodies other than the main orbited body have to be taken into account. For example, in the case of the Earth, a spacecraft orbiting our planet will be perturbed by the Sun and the Moon mainly, but also by other Solar System bodies.

These additional gravitational forces are part of what is called the *N-body* problem, as a reference to the *N* bodies involved in it. As *N* gets higher, the problem becomes more and more complicated, and, therefore, a solution is not always directly obtained.

Up to $N = 2$, the problem has an analytical solution, first solved by J. Bernoulli in 1710 (Wakker, 2015). This is, however, the unperturbed case and is not applicable in this section. When $N > 2$, there is no analytical solution for the general case. Nevertheless, if some restrictions are applied to the statement of the problem, it is possible to get to an analytical solution. Example of this is the *circular restricted three body problem*, which is widely described in section 3.3 from (Wakker, 2015).

This situation is usually solved treating the N bodies as point-masses, calculating their acceleration separately and integrating all the components via numerical integrators. The positions of the planets will be retrieved using (NOVAS, 2011).

2.2.2.3 Thrust

Thrust represents the main source of acceleration that a satellite may use to face the rest of the perturbations that act upon the trajectory. Its effect is non negligible, precisely because it has been designed to change the trajectory of the orbit or to compensate for the perturbing forces that modify the latter.

When in an atmospheric environment, the force that the thrusters generate can be expressed by this simple equation:

$$T = \dot{m}v_e + (p_e - p_{amb})A_e \tag{2.9}$$

where T is the total thrust on the spacecraft, $\dot{m}$ is the mass rate, v_e is the exhaust velocity of the propellant out of the nozzle, p_e and p_{amb} are the exhaust (just inside the nozzle) and ambient pressure, and A_e is the exhaust section of the rocket engine.

On the other hand, when in vacuum, $p_{amb} \approx 0$ and the last term is substituted by a constant term called *specific impulse* (I_{sp}) that is defined as the impulse produced per unit of weight of propellant. It reflects the efficiency of the thrusting system. With this parameter, the previous equation can be re-written as:

$$T = \dot{m} I_{sp} g_0 \qquad I_{sp} = \frac{\int T\, dt}{\int \dot{m} g_0\, dt} \tag{2.10}$$

with g_0 is the gravitational acceleration at sea level on Earth.

From (2.9), it is easy to see that, as the exit velocity is only dependant on the design of the rocket engine, the thrust will be higher at higher altitudes. Considering I_{sp} in (2.10) to be variable (not constant), it can be said, in order to agree with (2.9), that the optimal specific impulse is reached in vacuum, so the propulsion is optimal in outer space conditions.

2.2.2.4 Atmospheric Drag

Atmospheric drag constitutes a highly relevant perturbation for satellites in low altitude orbits. It is due to the layer of gases that surround some planets, such as the Earth, for example. For this reason, using an accurate atmospheric drag model is extremely important for orbital propagations.

The force that the aerodynamic drag induces on the spacecraft is related to the composition of the atmosphere, the shape and dimensions of the satellite and the instantaneous speed at which it flies. With this information, the drag acceleration can be modelled as:

$$\boldsymbol{a_{drag}} = -\frac{1}{2}\left(\frac{c_D\, S}{m}\right)\rho_\infty\, \boldsymbol{v_\infty^2} \tag{2.11}$$

where c_D is a coefficient that accounts for the aerodynamic efficiency of the satellite (drag coefficient), S is a representative cross-section or wet area, m is the mass of the spacecraft, and v_∞ and ρ_∞ are the velocity (relative to the motion of the atmosphere) and the density of the medium in an unaltered point at the same altitude and conditions as the rest of points under study.

Equation (2.11) models the drag acceleration in a very precise way, when all parameters are known. However, the main problem with this perturbation is the value of ρ. The density of the atmosphere is not homogeneous throughout its extension nor varies in a predictable manner. It is affected by atmospheric streams, solar flux and tidal effects, among other perturbations. These effects can sometimes be extremely difficult to predict, and, thus, the values that the propagator gives will include an error with respect to reality. For this error to be as small as possible, numerical and empirical models for the atmosphere are used. In particular, the NRLMSISE00 model was adopted for this study.

As for magnitude of the perturbation, atmospheric drag is the most notable perturbation under 200 km of altitude. On the other hand, and as specified by the model used in this thesis (NRL, 2016), it can be disregarded above 1000 km (Wertz, 2009).

2.2.2.5 Radiation Pressure

Radiation pressure is the perturbation arising from radiated particles hitting the exposed surfaces of an orbiting body. Such particles generate a small but continuous acceleration directed against the source of radiation that could, throughout long periods of time, significantly deviate the spacecraft from its original trajectory.

On the terrestrial surface, the Solar Radiation Pressure (SRP) is extremely low because of the atmosphere, which offers a layer of protection. However, for satellite orbits, this effect has to be taken into account for accurate predictions. Earth-orbiting satellites might need to take Earth's Albedo, with a value of 0.3 (de Pater & Lissauer, 2007), into account also, in order for the propagation to be within the required tolerances.

It is possible to model this force in a simple way by using

$$\mathrm{a_{rad}} = c_R \frac{W\,S}{m\,c}, \quad with \quad 0 \le c_R \le 2 \tag{2.12}$$

where W represents the incoming radiation flux, S is the cross-sectional area, c is the speed of light, and c_R is a factor that takes into account the satellite's reflectivity. In (2.12), W is not a constant value and varies according to the proportional relation

$$W_{surface}\, r_{surface}^2 = W_1\, r_1^2 = \cdots = W\, r^2 \rightarrow W = \left(\frac{r_{surface}}{r}\right)^2 W_{surface} \tag{2.13}$$

And, thus, the model shown in (2.12), when applied to the Solar System, can be rewritten as:

$$\mathrm{a_{rad}} = c_R \frac{S}{m\,c}\left(\frac{R_S}{r}\right)^2 W_S \tag{2.14}$$

where R_S is the radius of the Sun, r is the distance from the Sun to the point of interest (usually spacecraft) and W_S is the radiation flux at the surface of the Sun.

Close to the Earth, $W_S \approx 1350\ \mathrm{W/m^2}$ and the radiation due to the Earth Albedo (including infrared radiations) is about $300 - 400\ \mathrm{W/m^2}$. These values, of course, are affected by the solar activity or time-varying coefficients of reflectivity. For this reason, a precise model of the solar activity is required for higher accuracies.

Figure 2.2 offers a great and general overview of how much the various forces mentioned above influence the orbiting body when this trajectory takes place around the Earth.

2.3 Precise Orbit Determination

As it was introduced in Chapter 1, POD is based on the combination of dynamics and orbital data to obtain a statistical orbit prediction leading to very high accuracy levels. In this section, a more detailed definition will be elaborated, including the resources needed, the theory involved, and its application to space related problems.

2.3.1 Data Acquisition

In order to make use of a POD method, a set of observations is needed. These observations can come from different sources, depending on the instrument used for the acquisition. Nowadays, two of the most frequently used techniques are GNSS positioning and Azimuth-Elevation (AE) tracking.

GNSS measurements are taken using satellite networks such as GPS, Galileo, or GLONASS services. They allow for POD without ground tracking facilities, which is vital for autonomous spacecraft navigation or formation flying. They are taken as pseudo-range and carrier phase measurements and afterwards transformed to Position, Velocity, Time (PVT). Examples of its application can be found in (Bock, et al., 2007). The accuracy of these instruments is usually in the order of $1 - 10\ m$.

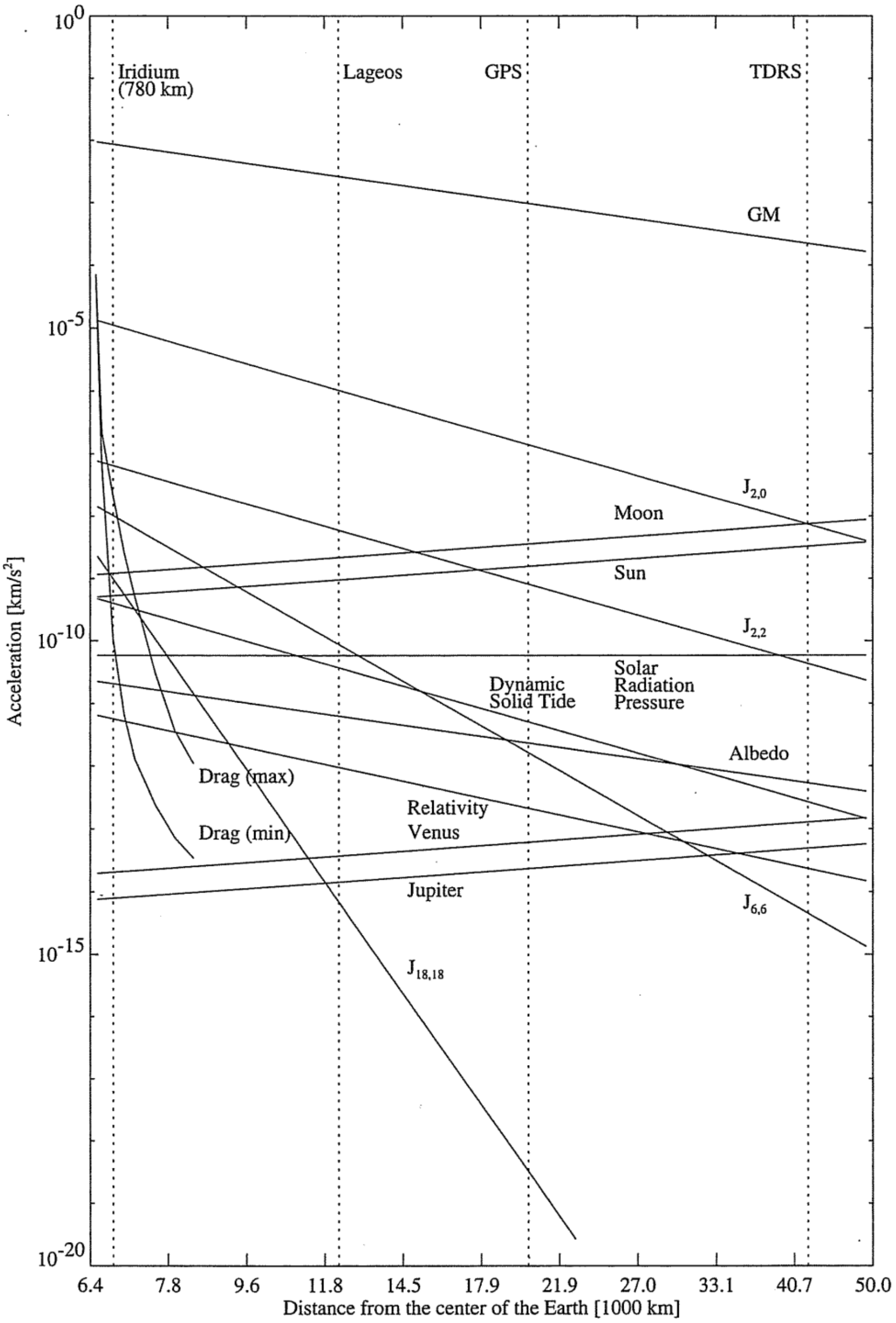


Figure 2.2 - Magnitude of the different forces involved in Earth-centred orbits. (Montenbrück & Gill, 2000)

These measurements will be used throughout this document represented as PVT values. Some of the advantages inherent to GNSS measurements are continuous tracking and high accuracy results, but requires the satellite to carry a GPS receiver and is dependent on the GNSS network involved.

On the other hand, AE data are taken using ground facilities that track arcs of the satellite's trajectory yielding data in the form of two angles: azimuth, which is the angle that the projection of the station-satellite line onto the Earth's surface forms with the direction of the Earth's north pole; and elevation, which is the angle that the station-satellite line forms w.r.t. to the local horizon (more information in (Montenbrück & Gill, 2000)). The main benefit of these observations is that they require a much simpler setup for the satellite to be tracked. However, they do not offer continuous data (sometimes multiple ground stations are required) and present lower accuracy (especially for lower elevations, where the atmospheric distortions are larger). For the purposes of this thesis, this accuracy has been assumed to result in position errors of around $10 - 100\ m$.

In any case, both acquisition techniques are relevant in the current space industry and, thus, tests involving their application will be run to assess whether the results given by the different penalty functions studied vary from one to another.

2.3.1.1 Covariance Matrixes

Covariance matrixes are used to collect, in a numerical way, information about the confidence level of a certain batch of measurements or the correlations between these measurements. Typically, they are used in weighted POD methods and play relevant roles in their application. In this thesis, the covariance matrixes involved have been set to be diagonal, meaning that no correlation is expected between the different observations. In reality, this is far from true, but represents a fair approximation when it comes to simulation environments.

2.3.2 Batch Methods vs Sequential Methods

Regarding the application of the POD algorithms, there are two possible methodologies: batch methods and sequential methods.

Batch methods make use of a set (or "batch") of measurements taken prior to the orbit determination process. Because of that, their ideal application lies on the ground segment. These methods make use of all the observations at once to compute the estimation of the trajectory. On the other hand, sequential methods provide an estimation of the trajectory that is updated every time an observation is made, what makes it perfect for real time applications such as on-board orbit determination.

Even though differences exist between the applications of both estimators (Tapley, Schutz, & Born, 2004, p. 237), their results show a very close agreement. Because the focus of this research lies on the comparison of the different penalty functions involved and not on a particular application of the tool, batch methods will be used for the tests included here. This is also supported by the possibility to apply the developed algorithm to real-world applications from a ground-segment perspective.

2.3.3 General Formulation of the Batch Methods

The general formulation of a batch method is based on a formulation where a set of parameters to be estimated has a linear (or linearised) relation with the measurements taken. For this simple formulation, the problem is stated using three different components:

- The observations, Y, grouped in a column vector containing the different values for the measured parameters, e.g., position, angle, range…
- The estimated parameters, X, grouped as well in a column vector and representing the final outcome of the estimation process.
- The model matrix, A, represented by a matrix expressing the partial derivatives of the observations w.r.t. the estimated parameters. It relates the two latter components such that:

$$Y = AX \tag{2.15}$$

Naturally, this expression is never exact because of imperfections in the models or noise in the observations, so a misfit is left. This misfit, ρ, is expressed as

$$\rho = Y - AX \tag{2.16}$$

and ρ is the quantity to be minimised so that X is the best possible fit for the cloud of observations Y.

To minimise ρ, a penalty function, Φ, is used to express the final form of the determination problem as shown in Equation (2.17). The optimisation and the penalty functions involved are further explained in Sections 2.4 and 2.5, respectively.

$$minimize \quad \Phi(\rho) \tag{2.17}$$

2.3.3.1 *Linearised Batch Methods*

If X is linearly related to Y, then the problem is straight-forward and a solution is easily found. However, in space applications, as well as in many other fields, the relation between the orbital state and the observations taken is not linear, so a linearised approach needs to be taken (Tapley, et al., 1994). Then, an initial guess is necessary and the components of the problem are changed to:

- The observation residual, y. It represents the difference between the real observations, Y, and the estimated observations, Y^*, that have to be provided as an initial guess.[2]
- The parameter residual, x. It represents the increment that has to be added to the first guess of X^* to reach the next estimation (and eventually the final solution).
- The model matrix, A, is again composed of the partial derivatives of the observations w.r.t. the estimated parameters. For space applications, this matrix is usually combined with the State Transition Matrix (STM), accounting for the dynamical evolution of the state; and obtaining, thus, the design matrix, H.

The problem is finally formulated as:

$$minimize \quad \Phi(\rho) = \Phi(y - Hx) \tag{2.18}$$

When the process is linearised, an iterative process needs to be adopted until convergence is reached. Usually, a tolerance level is set and the change on the value of X (i.e., x) is used as convergence criteria. This process is accurately detailed in Section 4.6 from (Tapley, Schutz, & Born, 2004) and will not be further discussed here.

2.3.4 State Transition Matrix

As mentioned before, the STM is used in orbit determination problems to serve as a way to integrate the evolution of the orbital state, when a relatively small perturbation, represented by ΔX_0 is introduced. The perturbation needs to be small because if its magnitude is too large, the process takes place out of the confidence region for the linearization of the model, and the result may not be accurate enough for the iterative process to find convergence.

The STM satisfies the relation

[2] The (*) superindex is adopted following the referred bibliography and represents estimated values, as opposed to measurements or other constant data. Further explanations can be read in the literature and/or in the Glossary Annex.

$$\Delta X_k = STM(t_k, t_0)\ \Delta X_0 \tag{2.19}$$

and always needs to be referred to an initial epoch (t_0). Detailed explanations about how the STM is obtained or its fundamental concepts are included in the literature, mainly in (Tapley, Schutz, & Born, 2004) and (Montenbrück & Gill, 2000). The latter, additionally, recommends that, as the calculation of the STM is a computationally demanding operation, reduced-accuracy models are utilised when the STM is generated. The differences in the results have been proved by the author (and throughout this thesis) to be completely dismissible and the savings in computational efforts really are worth the assumption.

2.3.4.1 The Sensitivity Matrix

The Sensitivity Matrix (SM) can be used for the estimation of parameters other than the components of the state vector, e.g., manoeuvres, coefficients, or satellite`s dimensions.

This matrix adds extra columns to the STM accounting for the partial derivatives of the accelerations involved in the propagation of the trajectory w.r.t. the selected parameters. Most typically, the drag coefficient, c_D, the radiation coefficient, c_R, or the manoeuvre commands are estimated to compensate for errors in the modelling of the satellite or in the models used for its propagation.

Again, more detailed information can be found in the literature.

2.4 Optimisation Theory

In this section, CO techniques will be introduced. As a starting point, conventional optimisation techniques will be briefly explained, to serve as reference point.

2.4.1 General Optimisation

The general optimisation problem is of the form:

$$\begin{aligned} &minimize \quad f_0(\boldsymbol{x}) \\ &subject\ to \quad f_i(\boldsymbol{x}) \leq b_i, \quad i = 1, \dots, m \end{aligned} \tag{2.20}$$

where the vector $\boldsymbol{x}$ is composed of the optimisation variables, $f_0\colon \mathbb{R}^n \to \mathbb{R}$ is the objective function (a.k.a. cost function), and $f_i\colon \mathbb{R}^n \to \mathbb{R}$ are the m (in)equality constraints. The optimum vector that would minimise the objective function shown in Equation (2.20) is called $\boldsymbol{x}^*$. In the same fashion, the mentioned problem can be maximised if $-f_0(\boldsymbol{x})$ is chosen as the objective or cost function.

The general optimisation problems can turn out to be extremely hard to resolve as the optimisation variable increases the number of elements that have to be optimised. According to (Hindi, 2004), the reasons for that are:

1. Local optima may blur the optimum point search.
2. There might even not exist a set of feasible points (due to constraints, for example).
3. The stopping criteria is often arbitrary.
4. The convergence rates could be very poor.
5. Numerical problems could cause the algorithm to stop or wander.

On the other hand, not all the optimisation problems are hard to solve and the ones that fit the requirements for an algorithm to find a proper optimal point are quite efficiently solved. Here, the selected constraints play an extremely relevant role because, sometimes, functions that are bounded within a certain range have the minimum precisely in the limit of their domain. Linear programming or the well-known Least Squares method (LSQ) are also easily and reliably solved. They are nowadays considered as, in words of Stephen Boyd (Boyd & Vandenberghe, 2009), a "*mature technology*", due to the extensive research they have been the object of. These two types of problems are comprised by a larger set of problems called "*Convex Problems"*.

2.4.2 Convex Theory

In this section some important (and also basic) concepts about CO theory will be introduced. This will be mainly taken from (Boyd & Vandenberghe, 2009).

Convex Sets

A set *C* is **affine** if the linear combination of any points in the set is also in *C*. This idea can be extended from two points to *n* points, being the linear combination (equation of a line):

$$\theta_1 x_1 + \cdots + \theta_n x_n \tag{2.21}$$

where $\theta_1 + \cdots + \theta_n = 1$.

Similarly, a certain set is **convex** if the line segment (and not the whole line, as happened with affine sets) between any two points in *C* lies in the set as well, i.e., if for any $x_1, x_2 \in C$ and any θ such that $0 \leq \theta \leq 1$:

$$\theta x_1 + (1 - \theta) x_2 \in C \tag{2.22}$$

In plain words, a set is convex if a line connecting two points in it can be made passing only through points that lie in the set. Convex sets must, therefore, have a non-empty interior.

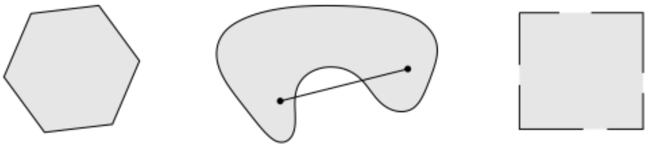


Figure 2.3 - Some simple examples. Left: a hexagonal convex set. Centre and right: two non-convex sets (Boyd & Vandenberghe, 2009).

Something that is really useful when dealing with (potentially) convex sets is to know how to determine whether they are or not actually convex sets. There are three main ways to assess that:

- **Analytical Evaluation**: if the function that generates the set is convex, it can be decomposed into convex subfunctions and/or it has been formed with operations that preserve convexity (see Section 2.4.3), then the set will be convex. This is not very applicable to computational techniques and has a much more theoretical and didactic basis.
- **Restriction of a convex function to a line**: if a function is restricted to a line (making it intersect with a plane, for example), it can be said that the function is convex (and thus the set it generates) if any of these lines is also convex. This generates infinite lines, but conceptually is a great way to understand the way the set is formed.
- **Random point generation**: if a large enough number of pair of points is generated and each segment connecting all of these pairs is evaluated, then, after a number of iterations, it can be concluded with a relatively high degree of certainty whether the set is convex or not.

The convex **Hull** of a set *C* is the smallest volume that makes a set of points convex. This can be clearly understood in Figure 2.4

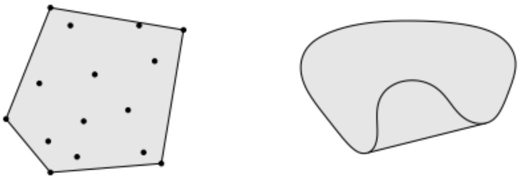


Figure 2.4 - Convex Hulls of simple sets (Boyd & Vandenberghe, 2009).

Another type of set is what is called a **cone**. A set *C* is a cone if for every $x \in C$ and any θ such that $0 \leq \theta$, it applies that $\theta x \in C$. Moreover, this cone is convex if:

$$\theta_1 x_1 + \theta_2 x_2 \in C \tag{2.23}$$

for any $x_1, x_2 \ \in C$ and $\theta_1, \theta_2 \geq 0$.

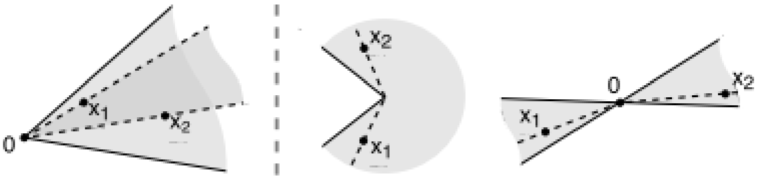


Figure 2.5 - Convex cone (left) and non-convex cones (right) (Bhagat, 2016).

Note that the fact that θ_i is always non-negative defines all the points of a convex cone at one side of the apex (which is usually considered to be placed at the origin). In the same fashion as before, given an apex point, conic hulls may be defined as can be seen in Figure 2.6.

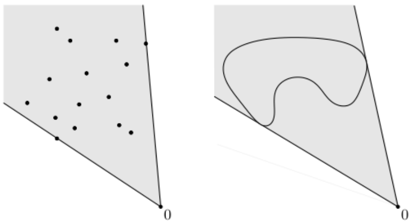


Figure 2.6 - Conic Hulls (Boyd & Vandenberghe, 2009).

A conic set can be constrained to a certain dimension by applying a certain norm. One of the most famous ones is the *second-order cone*, which results from applying the Euclidean norm to a certain conic set. It is defined as:

$$C = \{(\boldsymbol{x}, t) \mid \|x\|_2 \leq t\} \tag{2.24}$$

If applied to $\mathbb{R}^3$, it is known as the *ice cream cone* or *Lorentz cone* (see Figure 2.7).

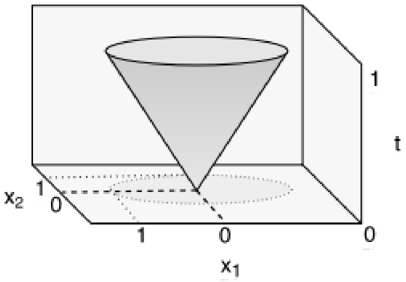


Figure 2.7 - Lorentz cone (Bhagat, 2016).

2.4.3 Convex Functions

A function f: $\mathbb{R}^n \rightarrow \mathbb{R}$ is said to be convex if $\boldsymbol{dom}\, f$ is a convex set for all $x, y \in \boldsymbol{dom}\, f$, and $0 \leq \theta \leq 1$:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \qquad (2.25)$$

Equation (2.25) is also known as the Jensen inequality (Boyd & Vandenberghe, 2009). From a geometrical point of view, Equation (2.25) can be interpreted as:

A function f is convex if the line segment between $(x, f(x))$ and $(y, f(y))$ lies above the graph of f in the range $[x, y]$.

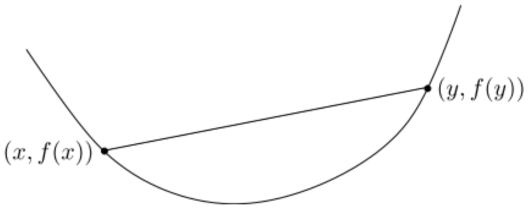


Figure 2.8 - Geometrical interpretation of a convex function (Boyd & Vandenberghe, 2009).

The same can be said about a concave function, with the difference that in a concave function, the chord between these two points lies below the graph.

Sometimes, some functions present problems when defining their domain, and thus, the resolution of these becomes more difficult. For this sake, there is a technique known as **extended-value extension**. In this technique a value for the function is assigned when the function is evaluated out of its domain. This is:

$$\tilde{f}(x) = \begin{cases} f(x) & x \in \boldsymbol{dom}\, f \\ \infty & x \notin \boldsymbol{dom}\, f \end{cases} \qquad (2.26)$$

In this way, when the function is not defined, it is ensured that its value is higher (or at least equal) than anywhere else in the domain.

The Jensen inequality (Equation (2.25)) and the extended-value method do not always serve as a proof of the (non-)convexity of a certain function. Luckily, there are other methods that are used to find out whether a function is convex or not. The two main methods are grouped as first and second-order conditions for convexity:

First-order condition: For f differentiable, it is convex if and only if $\boldsymbol{dom}\, f$ is convex and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{2.27}$$

in all $x, y \in \boldsymbol{dom}\, f$. When observed carefully, it is clear that Equation (2.27) is a first-order Taylor approximation of the function itself, thus, what this expression reflects is that the Taylor expansion of a function is a global underestimator of its behaviour. This implies a concept of vital importance: it is possible to obtain global information of a convex function from local information. Proof of this very important property is given in (Boyd & Vandenberghe, 2009).

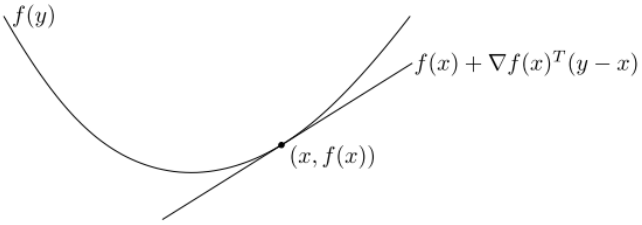


Figure 2.9 - First-order condition for convexity (Boyd & Vandenberghe, 2009).

Second-order condition: For f twice differentiable, it is convex if and only if $\boldsymbol{dom}\, f$ is convex and its Hessian (second derivative) at any point is positive-semidefinite:

$$\nabla^2 f(x) \succcurlyeq 0 \tag{2.28}$$

where $\succcurlyeq$ refers to a matrix inequality. For a function in $\boldsymbol{R}$, this condition basically says that a function is convex if its derivative, i.e., its growth, is always non-decreasing, $f''(x) \geq 0$.

These two conditions can be easily reversed for the concave case by just changing the signs in the inequalities.

Examples of Convex Functions

Some of the most basic and known examples of convex functions are:

1. Exponential functions: e^{ax} is convex in $\mathbb{R}$, for any $a \in \mathbb{R}$.
2. Powers: x^a is convex in $\mathbb{R}_{++}$ when $a \geq 1$ or $a \leq 0$ and concave otherwise.
3. Powers of absolute value: $|x|^p$, for any $p \geq 1$, is convex in $\mathbb{R}$.
4. Logarithm: concave in $\mathbb{R}_{++}$.
5. Negative entropy: $x \cdot \log(x)$ is convex in $\mathbb{R}_{+}$if the function is defined as 0 when $x = 0$.
6. Norms: every norm in $\mathbb{R}^n$ is convex.
7. Max functions: always convex.
8. Quadratic-over-linear functions: x^2/y is convex when $\boldsymbol{dom}\, f = \mathbb{R} \times \mathbb{R}_{++}$.
9. Log-determinant: $f(\boldsymbol{X}) = \log(\det \boldsymbol{X})$ is concave in $\boldsymbol{dom}\, f = S^n_{++}$.

Operations that Preserve Convexity

Also important is to take into account the operations that allow to preserve convexity when they are applied:

1. Non-negative scaling.
2. Addition.
3. Non-negative weighted sum (from two previous ones).
4. Composition with affine mapping: if $f\colon \mathbb{R}^n \to \mathbb{R}, \boldsymbol{A} \in \mathbb{R}^{n \times m}$, and $\boldsymbol{b} \in \mathbb{R}^n$:

$$g(\boldsymbol{x}) = f(\boldsymbol{Ax} + \boldsymbol{b})$$

where $g: \mathbb{R}^m \rightarrow \mathbb{R}$, and $\boldsymbol{dom}\ g = \{\boldsymbol{x} \mid \boldsymbol{Ax} + \boldsymbol{b} \in \boldsymbol{dom}\ f\}$, then if f is convex, g is also convex.

5. Point-wise maximum.
6. (Point-wise) Minimization.

All these cases are further explained in (Boyd & Vandenberghe, 2009).

2.4.4 Optimisation Problem

First of all, some basic terminology used in general optimisation problems will be defined. The problem notation is written as:

$$\begin{aligned} &minimize \quad f_0(\boldsymbol{x}) \\ &subject\ to \quad f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \dots, m \\ &\qquad\qquad h_i(\boldsymbol{x}) = 0_i, \quad i = 1, \dots, p \end{aligned} \tag{2.29}$$

where all components are the same as described in Equation (2.20) except $h_i(\boldsymbol{x})$, which are the p equality constraints. Also note that all constrains are referred to 0 and are set as minor or equal. If $m = p = 0$, the problem is *unconstrained*.

The domain (D) of the convex problem is defined as:

$$D = \bigcap_{i=0}^{m} \boldsymbol{dom}\ \boldsymbol{f_i} \cap \bigcap_{i=1}^{p} \boldsymbol{dom}\ h_i \tag{2.30}$$

A problem is feasible if there is at least one point contained in the domain D. Moreover, an optimal value (p^*) is defined by:

$$p^* = min\{f_0(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) \leq 0, i = 1, \dots, m, h_i(\boldsymbol{x}) = 0_i, i = 1, \dots, p\} \tag{2.31}$$

In other words, p^* is the minimal point that fulfils the constraints imposed in Equation (2.30). Note that p^* is used as well when working in the extended-value case. There, if the problem is infeasible, $p^* = \infty$. The optimal point $\mathbf{x}^*$ is the point that makes $f_0(\boldsymbol{x}^*) = \mathrm{p}^*$.

With this set up, there are several characteristics that could be further explained, such as changes of variables, function transformations, introduction of slack variables or elimination/introduction of equality constraints. These cases are explained in detail in (Boyd & Vandenberghe, 2009).

2.4.4.1 Convex Optimisation Problem

The convex optimisation problem has the form:

$$\begin{aligned} &minimize \quad f_0(\boldsymbol{x}) \\ &subject\ to \quad f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \dots, m \\ &\qquad\qquad a_i^T \boldsymbol{x} = b_i, \quad i = 1, \dots, p \end{aligned} \tag{2.32}$$

where $\mathrm{f}_0, \dots, f_m$ are convex functions. Three points must be remarked with respect to the general optimisation problem:

1. The objective function must be convex.
2. The inequality constraints must also be convex.
3. The equality constraints must be affine.

This immediately remarks an important property: the feasible set of a convex optimisation problem is convex.

$$D = \bigcap_{i=0}^{m} \boldsymbol{dom}\, \boldsymbol{f_i} \tag{2.33}$$

As mentioned before in Section 2.4.3, the first-order condition for convexity allows to obtain global information from local information. In that case, for any differentiable f and $x, y \in \boldsymbol{dom}\, f_0$:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

and, thus, x is only optimal if and only if it belongs to the set of feasible points for the problem and $\nabla f(x)^T (y - x) \geq 0$. Proof of this fact can be found in (Boyd & Vandenberghe, 2009, p. 139).

2.4.4.2 Special Cases

The convex optimisation problems may appear in special or characteristic forms that allow for a simpler resolution. Three very well-known cases will be briefly commented in the next paragraphs.

1. Linear Programming (LP): The special case when both the objective function and the constraints are affine. This is, the problem has the form:

$$\begin{aligned} &minimize \quad c^T \boldsymbol{x} + d \\ &subject\ to \quad G\boldsymbol{x} \leq h \\ &\qquad\qquad\quad \mathrm{A}\mathbf{x} = \mathrm{b} \end{aligned} \tag{2.34}$$

   where *d* is usually omitted, since it does not change the outcome of the optimisation.

2. Quadratic Programming (QP): When the previous LP problem includes a quadratic term in the objective function, while the constraints are preserved affine.

$$\begin{aligned} &minimize \quad \frac{1}{2} \boldsymbol{x}^T \boldsymbol{P} \boldsymbol{x} + \boldsymbol{q}^T \boldsymbol{x} + r \\ &subject\ to \quad G\boldsymbol{x} \leq h \\ &\qquad\qquad\quad \mathrm{A}\mathbf{x} = \mathrm{b} \end{aligned} \tag{2.35}$$

   When, additionally, one or more constraints are also quadratic, the problem is called Quadratically Constrained Quadratic Problem (QCQP). The famous LSQ problem, is an example of unconstrained QP problem, and LP problems are a special case where $\boldsymbol{P_i} = 0$, thus, QP problems are more general than LP.

3. Second-Order Conic Programming (SOCP): This is a more general problem in which one of the constraints is a second-order conic constraint:

$$\begin{aligned} &minimize \quad f^T \boldsymbol{x} \\ &subject\ to \quad \|A_i \boldsymbol{x} + b_i\|_2 \leq c_i^T \boldsymbol{x} + d_i, \qquad i = 1, \dots, m \\ &\qquad\qquad\quad \mathrm{F}\mathbf{x} = \mathrm{g} \end{aligned} \tag{2.36}$$

   In this case, for example, if $\mathrm{A_i} = 0$ a general LP problem is posed. It is clear from Equation (2.36) that this form is more generic. There is, however, an even more general form of a convex problem:

4. Conic Programming (CP): The difference with respect to the SOCP is that it contains a *generalised inequality* constraint ($\mathrm{M}\boldsymbol{x} + \mathbf{p} \preccurlyeq_{\mathrm{k}} 0$) over a certain cone $K \subseteq \mathbb{R}^m$.

$$minimize \quad f^T \boldsymbol{x} \tag{2.37}$$

$$
\begin{aligned}
subject\ to\quad & M\boldsymbol{x} + \boldsymbol{p} \preccurlyeq_k 0 \\
& \mathrm{F}\mathbf{x} = \mathrm{g}
\end{aligned}
$$

The term generalized inequality means that the point in discussion is contained in a certain cone, component-wise. If write $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$**,** that means that $x_i \leq y_i$, for $i = 1, \dots, m$. As it is still an inequality, it preserves its natural properties:

$$
\begin{aligned}
\boldsymbol{x} \preccurlyeq_K \boldsymbol{y} \rightarrow \boldsymbol{\alpha x} \preccurlyeq \boldsymbol{\alpha y} \qquad & \forall \boldsymbol{\alpha} > \mathbf{0} \\
\boldsymbol{x} \preccurlyeq_K \boldsymbol{y} \rightarrow \boldsymbol{\alpha x} \succcurlyeq \boldsymbol{\alpha y} \qquad & \forall \boldsymbol{\alpha} < \mathbf{0} \\
\boldsymbol{x} \preccurlyeq_K \boldsymbol{y} \quad \boldsymbol{x} \succcurlyeq_{\boldsymbol{K}} \boldsymbol{y} \qquad & \quad \boldsymbol{x} = \boldsymbol{y}
\end{aligned}
\tag{2.38}
$$

The expression $M\boldsymbol{x} + \boldsymbol{p} \preccurlyeq_k 0$, means that $M\boldsymbol{x} + \boldsymbol{p} \in K.$ This information is further completed in (Bhagat, 2016) and (Boyd & Vandenberghe, 2009).

It can be observed that, if a CP solver is coded, all the other type of problems would be solved, because they are contained in the formulation and to reach them would be only a matter of setting determined coefficients to zero. However, there is a flaw to this simple idea: the generalised inequalities. They apply an equal concept in different ways and it is difficult to define a generalised inequality in an efficient way in a computational solver. The solution for this, is explained in Section 4.4.2 of (Bhagat, 2016) and deals with this problem based on the fact that, for the SOCP, the set of inequality constraints are defined by m cones (one for each $A_i, b_i, c_i,$ and d_i). If all the sets of feasible points for each cone are obtained, m cones that define the problem will be found. The intersection of all these cones forms the set of feasible points of the whole problem, K, and, thus:

$$
K = k^1 \times k^2 \times \dots \times k^m \tag{2.39}
$$

where k^i represents the cone for the i^{th} constraint and K is the union of all of them.

With the help of the formulation of the norm-cone shown in Equation (2.23), it is possible to define the components M and p shown in Equation (2.38) as:

$$
\begin{gathered}
k_i = \left\{ \left(A_i\boldsymbol{x} + b_i, c_i^T\boldsymbol{x} + d_i \right) \middle| \, \| A_i\boldsymbol{x} + b_i \|_2 \leq c_i^T\boldsymbol{x} + d_i \right\} \\
\rightarrow \begin{pmatrix} A_i\boldsymbol{x} + b_i \\ c_i^T\boldsymbol{x} + d_i \end{pmatrix} = \begin{pmatrix} A_i \\ c_i^T \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} b_i \\ d_i \end{pmatrix} \succcurlyeq_{k^i} 0
\end{gathered}
\tag{2.40}
$$

and thus:

$$
\mathrm{M} = -\begin{pmatrix} A_1 \\ c_1 \\ A_2 \\ c_2 \\ \vdots \\ A_m \\ c_m \end{pmatrix} \qquad p = -\begin{pmatrix} b_1 \\ d_1 \\ b_2 \\ d_2 \\ \vdots \\ b_m \\ d_m \end{pmatrix}
\tag{2.41}
$$

where the negative sign helps the $\succcurlyeq$ shown in Equation (2.40) transform to a $\preccurlyeq$ as Equation (2.38) requires.

2.5 Penalty Functions

In this section, the different penalty functions, Φ, to be compared throughout this thesis will be introduced. A graphical representation of these functions is depicted in Figure 2.10. It is worth noting that all of the cost functions studied in this report, including the quadratic cost function used in the traditional LSQ method, are convex, and, thus, all the theory explained above can be applied to them.

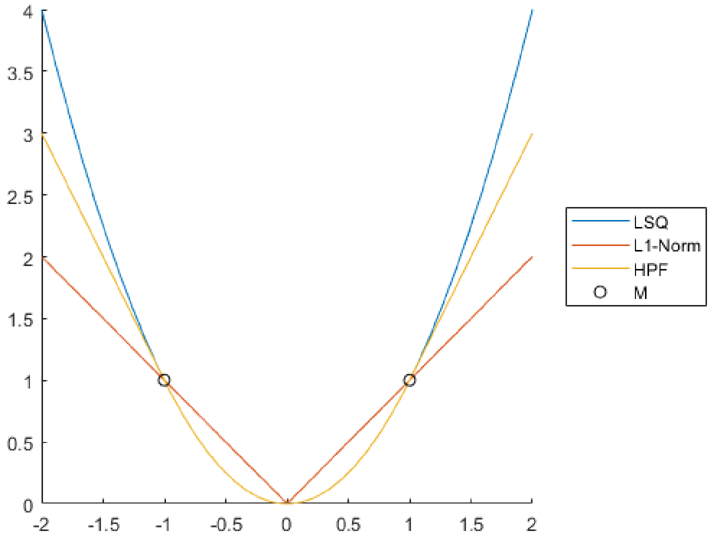


Figure 2.10 - Graphical representation of the different penalty functions studied.

2.5.1 Least-Squares

The LSQ parameter estimation, based on the L2-Norm, is used to obtain reliable solutions for a certain set of parameters given a larger set of measurements and a model relating the parameters and the measurements. The LSQ penalty function can be described as:

$$\Phi^{\mathrm{LSQ}}(\rho) = \rho^T \rho \tag{2.42}$$

The LSQ batch method fits linear problems, so its application to problems with a non-linear behaviour would require the linearization of these. Thus, starting from a relation between the observations, y_i $(i = 1, 2, \dots, m)$, and the parameters, x_j $(j = 1, 2, \dots, n)$

$$y_i = a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,n}x_n \tag{2.43}$$

It is possible to set this relation in matrix form as:

$$A\ x^* = y^* \tag{2.44}$$

where x^* and y^* represent the estimated parameters and observations, respectively, and A is called the design matrix.

However, if the relation between x^* and $y^*$is not perfectly linear or noise is present in the observations, both sides of Equation (2.44) will not be equal, and residuals will appear:

$$y^* - Ax^* = \rho^* \tag{2.45}$$

The residuals, ρ_i^* $(i = 1, 2, \dots, m)$ represent the misfit vector between the parameters (combined with the design matrix) and the observations.

The LSQ method tries to minimise the sum of the squares of this misfit:

$$\min\{{\rho^*}^T \rho^*\} \tag{2.46}$$

$$\min\{(y^* - Ax^*)^T (y^* - Ax^*)\}$$

which leads to the solution

$$x^* = \left(A^T A\right)^{-1} A^T y^* \tag{2.47}$$

The product $A^T A$ is called normal matrix. This product is symmetric and positive definite if not rank deficient.

When not all the observations have the same quality, a weighting factor can be included in the method. This weighting, w, is defined as the inverse of the assumed accuracy level, usually represented by the standard deviation, σ.

$$w_i = \frac{1}{\sigma_i}$$

$$W = \begin{pmatrix} w_1 & 0 & \cdots & \cdots & 0 \\ 0 & w_2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & w_m \end{pmatrix} \tag{2.48}$$

Repeating the minimisation procedure used in Equation (2.46), the weighted solution of the LSQ method is obtained:

$$x^* = \left(A^T W^T W A\right)^{-1} A^T W^T W y^* \tag{2.49}$$

The product $A^T W^T W A$ is called the weighted normal matrix. More specifically, the product $W^T W$ is usually written as Q_y^{-1}, which is the inverse of the observation error covariance matrix (which needs not to be diagonal).

$$Q_y = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \sigma_m^2 \end{pmatrix} = \begin{pmatrix} 1/w_1^2 & 0 & \cdots & \cdots & 0 \\ 0 & 1/w_2^2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1/w_m^2 \end{pmatrix} \tag{2.50}$$

So the weighted solution could be written as

$$x^* = \left(A^T Q_y^{-1} A\right)^{-1} A^T Q_y^{-1} y^* \tag{2.51}$$

The use of a weighted method, allows to also obtain information about the errors in the estimation of the parameters. This is known as quality assessment. From Q_y and A, it is possible to obtain the parameter covariance matrix (demonstration can be found in (Visser P. d., 2017, p. 7)) Q_x:

$$Q_x = \left(A^T Q_y^{-1} A\right)^{-1} \tag{2.52}$$

The values in the diagonals $Q_{x_{jj}} = \sigma_{x_j}^2$ are estimates of the variance of the estimated parameters, where σ_{x_j} is known as the formal error. On the other hand, the off-diagonal values provide estimates for the covariances of the parameters, i.e., the dependency existing between each of them and the stability of the estimation process. This correlation is usually represented as

$$\rho_{x_{ij}} = \frac{Q_{x_{ij}}}{\sqrt{Q_{x_{ii}}\, Q_{x_{jj}}}} \qquad with \qquad -1 \leq \rho_{x_{ij}} \leq 1 \tag{2.53}$$

Correlations close to 1 or -1 imply that the set of observations is not good enough to estimate parameters x_i and x_j simultaneously.

The formal error is a very insightful parameter that will provide an estimation of how accurate the determination of the parameters is, without any information about the true values of the modelled function. However, this error only depends on modelling parameters (design matrix and covariance matrix of the observations). For this reason, if the model's accuracy or the measurements are not as close to reality as expected, the estimated error would be much smaller than the actual error, leading to false conclusions about the quality of the results. This is exemplified in Section 3.4.

2.5.2 L1-Norm

Even if its use is not as common as the LSQ method, the L1-Norm has been used and proved to be less sensitive against certain observation errors such as outliers (Bassett & Koenker, 1978). It can be represented by

$$\Phi^{\mathrm{L1}}(\rho) = \|\rho\|_1 \tag{2.54}$$

and, for a simpler interpretation when using CVX (see Section 2.6.2), it will be implemented as:

$$\begin{aligned} & r = Hx - y \\ & minimise\ z \\ & subject\ to: \\ & \qquad z \geq r \\ & \qquad -z \geq -r \end{aligned} \tag{2.55}$$

This formulation was used due to convergence issues encountered during the testing phase in this thesis. It was defined following the rules and concepts previously introduced in sections 2.4.2 and 2.4.3.

2.5.3 Huber's Penalty Function

Huber's Penalty Function (HPF) is a hybrid between the two previously introduced norms that tries to make use of the robust qualities of the L1-Norm for outlying measurements, and the accuracy level of the LSQ method for the rest. It is described as

$$\Phi^{HPF}(x, M) = \begin{cases} x^2, & |x| \leq M \\ 2M|x| - M^2, & |x| \geq M \end{cases} \tag{2.56}$$

where the M factor represents the threshold where the Huber´s function changes its behaviour. Conceptually, its value means that the cost function will change if the residual is beyond (or within) M times the standard deviation (Geebelen, Wagner, Gros, Swevers, & Diehl, 2017). If nothing else is specified, it will be set to $M = 1.35$, according to the recommendation found in (Owen, 2006).

It represents the most interesting study case of all the penalty functions introduced, since its application to space-related problems is yet to be assessed.

2.6 Simulation Environment

In this section, the simulation environment used to run the tests will be introduced, ranging from the platform used for scripting, to the engines responsible for the propagation of the trajectories, or the optimisation of the penalty functions.

2.6.1 Scripting Frame

All the tests run throughout this document are set up using a script written in MATLAB. Figure 2.11 shows a simplified representation of the organisation of the simulation environment. The outermost box represents the main MATLAB script, in charge of setting the different conditions for the tests, such as the penalty functions involved, the type of test to be run, or the initialisation of the variables used in the POD process. After that, and if no real measurements are being used, a scenario is generated and a true trajectory is obtained. A layer of noise is applied on top of that true trajectory, and the measurements that the batch method will receive as input are generated using PVT format nominally (Section 4.4 will also treat different formats).

Once the input is ready, each configured condition is run in a loop and results for each of these conditions are saved and plotted. Each loop consists of a brief initialisation, a call to the function that runs the selected POD method, and a results processing section.

Finally, all the results are saved and the script closes up.

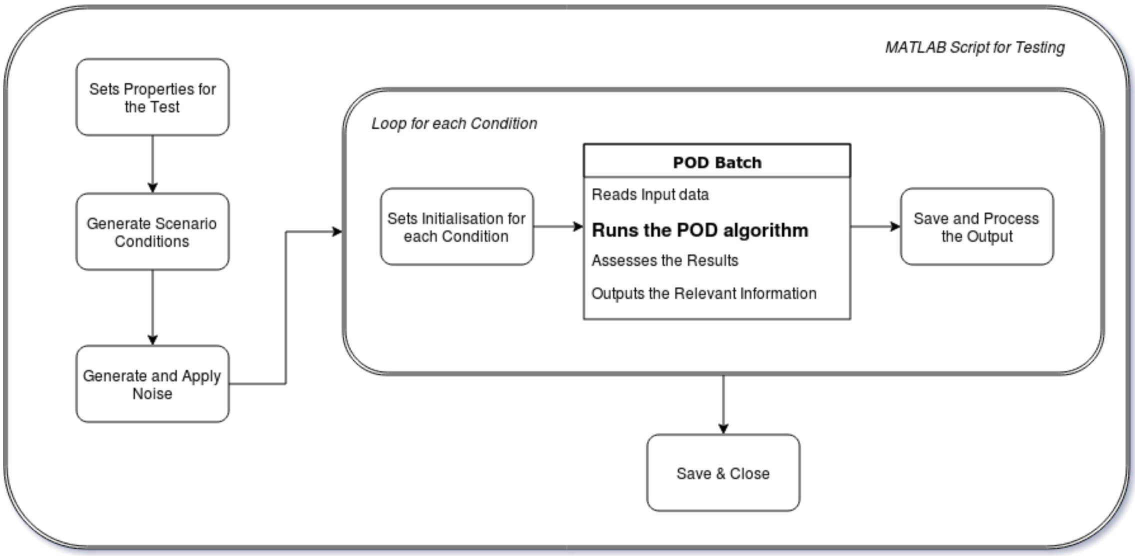

MATLAB Script for Testing
Sets Properties for the Test
Generate Scenario Conditions
Generate and Apply Noise
Loop for each Condition
Sets Initialisation for each Condition
POD Batch
Reads Input data
Runs the POD algorithm
Assesses the Results
Outputs the Relevant Information
Save and Process the Output
Save & Close

Figure 2.11 - Schematic overview of the testing environment.

2.6.2 CVX

CVX is a toolbox of solvers designed to deal with convex problems in MATLAB interface in a very user-friendly fashion.

For the developed algorithm, the CVX tool was applied to the optimisation of the penalty functions to be assessed. All of the penalty functions were initially included in the manual provided (Grant & Boyd, 2017), although the L1-Norm needed a reformulation for a better numerical assimilation (shown in Section 2.5.2).

The basic package of the CVX tool offers two solvers: SDPT3 and SeDuMi (further detailed in (Grant & Boyd, 2017)). After thorough testing, it was found that the SDPT3 solver is more robust regarding numerical problems, but SeDuMi is able to work using less memory. Due to this, the L1- and L2-Norm are minimised using the former while HPF is minimised using the latter.

2.6.2.1 Validation of the tool

The implementation of the CVX toolbox should be tested against the conventional LSQ algorithm to check whether the results match. And if they do, it will be used for all the penalty functions to be studied, including the LSQ method. The scenario used can be seen in Table 2.3 (on page 36).

The results of this test can be observed in Figure 2.12. The nominal noise layer was used for these tests (see Section 2.7.1.5) and only position measurements were considered.

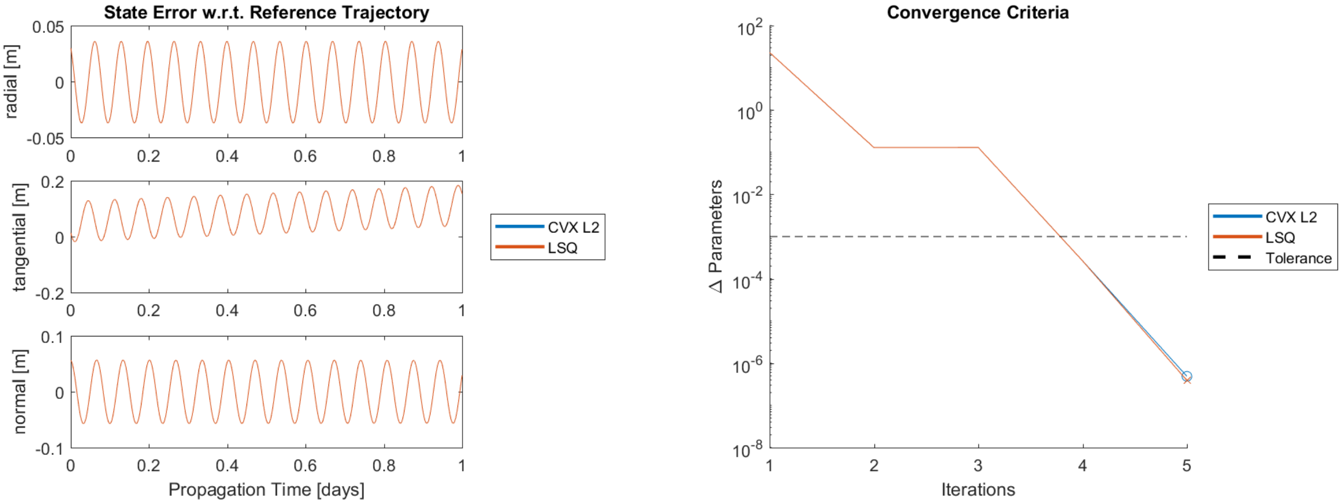


Figure 2.12 - Comparison of the results obtained from the conventional LSQ algorithm and from the CVX toolbox.

It is clear that the toolbox fits perfectly the results obtained by the conventional LSQ algorithm in both accuracy and convergence rates (so perfectly that the blue line cannot be distinguished under the red one without zooming). The only difference, was found in the computational effort, where the conventional algorithm resulted to be around 7% faster ($50{,}34\ s$ against $54{,}51\ s$).

This increase in time was expected, since the toolbox uses a general approach for convex problems, which will always be slower than a specific method. Nevertheless, the higher flexibility of the CVX tool and the savings in time coming from the problem statement largely compensate this drawback.

These results prove the CVX toolbox to be valid. Thus, from now on, when any L2-Norm (LSQ) method is applied, it will be accounting for the CVX implementation.

2.6.3 MyProp

The trajectories and the STM need to be computed using an orbital propagator. Such a propagator was the objective of the internship of the author of this thesis before the MSc thesis. The developed (and hereby implemented) tool is a Precise Orbit Propagator (POP) consisting on a numerical integrator able to generate a trajectory from initial scenario conditions.

This POP, named *MyProp*, includes all the forces introduced in Section 2.2.2 and some additional features such as STM and SM computation, or eclipse detection algorithm. All its different modules have been validated in (Peñarroya, 2017), and the tool is ready to be used in a professional environment.

MyProp is coded in C and a *mex* function was compiled from MATLAB to ease the interaction between the two platforms. This allows very fast propagations, combined with a more user-friendly result processing capabilities.

Table 2.1 – Nominal perturbations included in the simulations (unless specified change)[3].

Perturbation	Implementation
Geopotential	Degree and order 20
Sun Gravity	DE405
Moon Gravity	DE405
SRP	Spherical Model
Atmospheric Model	NRLMSISE00

The nominal models used in the trajectory propagations can be observed in Table 2.1. If this configuration is changed for any specific test, it will be mentioned.

2.7 Modelling of Errors

The different techniques used in POD algorithms usually contain an optimisation problem within their structure. Regardless of the nature of the objective function to be minimised, the solution obtained is the best fit to the observations, according to a certain penalty function. However, this fit is not perfect and there will always be a residual.

This residual is the numerical representation of the different inaccuracies that a certain estimation could have. Mainly, they can be grouped into three different types: errors in the measurements, in the models, or in the orbit determination problem. The latter refers to the final error of the estimation w.r.t. the real solution, and represents the main discussion of this thesis. Thus, it will be treated afterwards in this document, in a more extensive way. The errors in the models will be specifically tested in Section 4.3.

2.7.1 Errors in Measurements

Depending on the measurement source used, the way in which data is collected varies. Even within similar measuring technologies, the quality of the data varies with respect to the accuracy of the instrument, the environmental conditions, or the orbital characteristics, for instance.

In order to execute a POD process, a set of observations is needed. Until the software is ready to handle real satellite data, a measurement generator will cover this necessity. This generator will include different sources of error to the true trajectory, propagated using MyProp. These sources are described in this section, as well as the way in which the developed generator deals with them, if applicable.

2.7.1.1 Stochastic Errors

Stochastic errors are due to random noise in the measurements that is usually fitted by a certain Gaussian distribution. To simulate this, a low amplitude noise around the true trajectory is generated, giving place to the observations. This noise is generated using a normal distribution, $N{\sim}(0,\sigma)$, for the nominal noise layer, although different distributions will be tested (read Section 3.5.3). It could be defined as:

$$observation^{k} = trueState^{k} + N(0,\sigma) \tag{2.57}$$

[3] See (Peñarroya, 2017) for more information about the used implementations.

where σ is the standard deviation of the noise for each component of the state, i.e., positions and velocities.

This error is processed to a large extent during the POD algorithm, which, by definition, finds the trajectory that best fits the cloud of measurements.

2.7.1.2 Systematic Errors

This type of errors are usually identified by biases or offsets in the measurements and might be difficult to detect. They can arise from problems with the measuring instrument or inaccuracies in the modelling parameters. A very typical case for this error could appear when using GPS measurements. If the antenna is not exactly in the centre of mass of the satellite, there will be an offset between the positions of the satellite (i.e., its centre of mass) and the position of the receiver, which is what the measurements reflect.

These errors are not identified by the POD method and have to be foreseen or processed afterwards. For instance, in the GPS case, it is possible to include an estimated offset in the measurements to compensate for the location of the GPS receiver within the satellite.

These systematic errors are currently not included in the generator.

2.7.1.3 Data Gaps

Data gaps are very usual problems in measurements that can appear due to the nature of the measurements or some temporal impediment, for instance. When a satellite does not have full tracking throughout its entire orbit, there will be gaps in the data collected. Depending on how large the gaps are, the number of measurements needed to arrive to a certain accuracy will vary. The most typical case is range-angle measurements.

2.7.1.4 Outliers

Outliers are errors in measurements characterised by a large difference w.r.t. the trend of the data. They are very common and have been included in the nominal noise layer that will be introduced in Subsection 2.7.1.5. The way in which these measurements are generated in this algorithm is based on probabilities and magnitudes.

The reference trajectory is given to the noise generation function as an input, along with the standard deviations, σ, of the noise for each component of the state vector, i.e., three positions and three velocities. Based on a pre-established probability, a certain number of observations will be flagged as outliers. When this happens, an outlying component will be added, i.e., an extra value within the outlier´s magnitude range (pre-established by the user) multiplied by σ.

Nominally, the configuration settings are adjusted as it is shown in Table 2.2. In this case, the observations will have the normally distributed noise, $N\sim(0,\sigma)$, i.e., approximately a 5% of the state components of all the observations will include a state component defined as:

$$outlierMag = \pm\, \sigma * rand([10-100]) \tag{2.58}$$

where, the ± sign means sign of this component is selected randomly. Summing up, the formula used to assign noise to an outlying observation, k, is:

$$observation^{k} = trueState^{k} \pm\ \sigma * rand([10-100]) \tag{2.59}$$

It is important to remark that this probability is applied to the components. Thus, the final number of outlying measurements will correspond approximately to the 5% of six times the number of observations.

There are many ways to identify them, but in this case, an approach based on the Grubbs´ test (Grubbs, 1969) was followed, and extended for multiple outliers; named Generalized Extreme Studentized Deviate (GESD) test. This method, chosen for its simplicity and effectiveness, calculates statistical parameters about the collection of measurements and looks for values in dissonance with the rest. These values are eliminated, leaving a signal free of outliers.

In this method, an a priori estimation of the number of outliers needs to be given, to serve as upper limit. If this number is higher than half the number of measurements, the maximum number of outliers can be larger than the number of non-outliers. This is far from acceptable for measurement data and the algorithm finds trouble in deciding which of the values are outliers and which are not. Therefore, the upper limit for the number of outliers is half of the observations, with a recommended value of a third of the data sample.

In addition, the algorithm needs a reference trajectory to compare the measurements to. This reference trajectory is something that, if not given, is completely unknown at the beginning of the processing. Thus, the first estimation of the trajectory will include the totality of the outlying measurements. After that, every iteration will process the observations using as a reference the current estimation of the trajectory, until convergence is found regarding the number of outliers.

The Grubbs' test uses this reference data to process the outliers. To do that, it finds the point that differs the most from the trend, $\Delta^{\max} meas$, and computes a statistical parameter, G:

$$G = \frac{|\Delta^{\max} meas|}{\sigma} \qquad (2.60)$$

where σ is the standard deviation of the input data, computed from the sample.

In parallel, the Student distribution is used to compute the t_{crit} of the distribution $T(n-2)$, with significance level α/n, where n is the number of samples in the data, and α is the nominal significance level. The critical value, above which, the chosen point is considered an outlier is given by:

$$G_{crit} = \frac{(n-1)t_{crit}}{\sqrt{n\left(n-2+t_{crit}^2\right)}} \qquad (2.61)$$

If $G_{crit} < G$, the value is considered an outlier and is discarded; if not, the algorithms continues with the next value, i.e., the second furthest away from the trend. This process continues until the maximum number of possible outliers is reached. It is important to remark that the only inputs given for this test are the observations, the reference trajectory, the significance level, which is related to the estimated distribution of the outliers; and an upper boundary for the number of outliers.

An interesting case is possible: if the number of outliers is high enough, the σ and the trend, $\bar{x}$, can be biased. Thus, the first considered points may not be evaluated as outliers. As more points are studied, the $\bar{x}$ and the σ will vary, and, at a certain point, one of the "hidden" outlying points will be discovered as such. Now, what would happen with the rest of the non-detected outliers? Will they be omitted by the algorithm and, therefore, perturb the accuracy of the POD algorithm? In preparation for this, the algorithm has been programmed to identify this situation and re-evaluate the previous points as outliers. In this way, even very outlier-rich data can be successfully processed and the quality of the results will not be compromised.

After that, and in order to omit the detected outliers for the POD algorithm, the measurements where an outlier is found are set to 0 (every component), as well as the corresponding rows of the design matrix that models the dynamics (see Section 2.5.1). Thus, the outlying measurements will have no contribution to the residual and the estimated value for the initial position is calculated in a clearer way.

This fact, however, has an impact on the calculation of the estimated errors, which are calculated from statistical information and, thus, are affected by the latter.

Table 2.2 - Main characteristics of the nominal noise layer.

Outlier Probability	5 %
$\boldsymbol{\sigma}$	$1\,m\,//\,1\frac{mm}{s}$
Outlier Magnitude	$10\sigma - 100\sigma$

For instance, in the case of the LSQ method, the formal error was defined in section 2.5.1 as the square root of the diagonal values contained in the parameter covariance matrix (see Equation (2.52)). Because this matrix only depends on the design matrix and the observations covariance matrix, the calculated value for the formal error will vary if the rows of the design matrix corresponding to the outlying measurements are dismissed. The lesser the number of points in a sample (matrix rows in this case) the larger the value of the standard deviation. This can be observed in Equation (2.52), where the inverse of something that becomes smaller (since the design matrix reduces its dimension after the outliers are processed) yields larger values.

Summing up, an increase is expected in the formal error after the outliers are processed. This increase, in turn, will be approximately proportional to the square root of the ratio between the original number of observations over the number of observations containing outliers. Thus, if one third of the observations are identified as outliers, the new formal error will increase by a factor around $\sqrt{3}$.

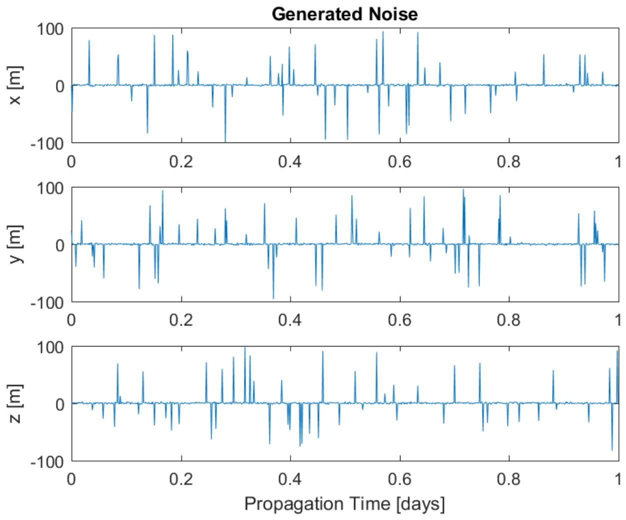


Figure 2.13 - Nominal Noise Layer.

2.7.1.5 The noise layer

When the reference trajectory used needs to be generated, i.e., no real data is being used; noise must be introduced to generate measurements that the POD method could take as an input. A noise generation is, thus, vital and has been applied to the reference trajectory computed by MyProp as explained in the previous subsections.

Depending on the test configuration, some parameters are modified. However, all these modifications take place upon a basic noise that will be referred to as the "nominal noise layer". This layer represents a combination of stochastic noise and outliers and its main characteristics are gathered in Table 2.2.

An outlier probability of $5\ \%$ is higher than the actual behaviour of current instruments. However, this was selected to keep a conservative approach and make the outlier detection easier to evaluate. This philosophy was also followed for the selection of the other parameters represented in Table 2.2.

This noise is applied equally to all scenarios (if no modifications are mentioned) so that the comparisons to be made could be more reliable. Figure 2.13 illustrates the generated nominal noise.

Table 2.3 – Scenario conditions for POD algorithm testing.

$\boldsymbol{a}\ [\boldsymbol{km}]$	7000.0
$\boldsymbol{e}\ [-]$	0.01
$\boldsymbol{i}\ [°]$	45.0
$\boldsymbol{\Omega}\ [°]$	10.0
$\boldsymbol{\omega}\ [°]$	50.0
$\boldsymbol{\theta}\ [°]$	35.0
Epoch	10th August 1993 08:00:00.00 UTC
dt [s] / timespan [days]	100 / 1

2.8 Scenario Conditions

The scenario conditions, similarly as the noise layer, are also gathered on a general set of parameters that will be equally applied to all the tests. When these conditions are changed, it will be mentioned in the corresponding section.

Selecting a single initial scenario for a relatively large number of tests is not straight forward. As many perturbations as possible should be included to enrich the results. For that sake, a LEO was chosen, so that drag acceleration could be included. The orbit was also chose to be circular to further simplify the interpretation of the results. The rest of the orbital elements, as well as the epoch, the timespan, and the time-step used to propagate the trajectory (acting as sampling-frequency) are described in Table 2.3.

Regarding the satellite's configuration, a medium size satellite was considered. It is assumed to have a wet surface (for both drag and SRP purposes) of $20\ m^2$, a drag coefficient $\mathrm{c_D}$ of 2.2 and a SRP coefficient $\mathrm{c_R}$ of 1.2. Its shape can be interpreted as a cannon ball. The in-orbit mass is $2000\ kg$. This information should suffice for the purposes of this thesis, since the attitude of the satellite is not considered in the propagations.

2.9 Testing Rationale

At this point, all the information needed for the proper understanding of the aim of this research has been presented and is, thus, finally time to start with the research. For that, and as it was introduced in Chapter 1, a series of tests will be run. It is the aim of this section to justify their application and serve as a preparation for the reader for the following chapters.

There are basically two different sides to the testing: the testing environment and the application of the algorithm to actual scenarios, i.e., the performance of the algorithm in different situations. The former will be presented in Chapter 3, and the latter distributed among chapters 4 and 5.

2.9.1 Simulation Environment Tests

These tests refer to the tests that need to be made to warrant not only the validity of the results but also of the tools used and the configurations of the simulation environment.

The core of the POP in charge of the trajectory propagations has already been validated in (Peñarroya, 2017). However, further validations need to be run for the STM, SM, and manoeuvre application, gathered in sections 3.1 and 3.2.

Independently, the CVX toolbox has also been implemented, as explained in Section 2.6.2. There, it was proven that the results are identical to those obtained from the usual formulation of the LSQ problem. It is not the goal here to validate the toolbox but the application of the latter to the stated problem and, if needed, to figure out a way to improve the convergence of the minimisation process for each penalty function, since each definition varies the optimal values of the problem and, thus, the resolution of the latter. This very important test will be shown in Section 3.3.

Once all the components are proven to work as expected, it is important to validate the methodology to be followed during the tests. In principle, as the true trajectory is generated within the simulation environment, the results could be easily compared to the true value to assess how well each penalty function is able to deal with the introduced noise. However, it would be also attractive to explore the possibility of an error estimation assuming no perfect knowledge of the true trajectory, as introduced in Section 2.5.1 with the formal error, but for all penalty functions. Section 3.4 will take care of this.

After that, and probably most importantly, the errors in the observations need to be tested. Not only its generation but also how much and in which way they affect the results. The developed outlier detection function will also be tested in Section 3.5. This test serves also as a first comparison between the penalty functions studied in this document, assessing how well they can estimate a reference for the data processing function, or, more directly, to assess the performances of the different penalty functions when using unprocessed outliers.

Finally, the value of the M parameter of HPF will be also investigated, since it is not clear yet how this parameter affects the performances of the function. These results will be presented in Section 3.6.

2.9.2 Performance Tests

On the other hand, and once the more computational side of the errors is properly observed and assessed, tests related to the performance of the developed algorithm regarding the environmental conditions will be run.

The first variable that comes to mind is the scenario. In Section 2.8, a nominal scenario was selected. However, different orbits might yield different results, and that is what Section 4.1 will investigate.

As it was introduced in Section 2.3, one of the most used applications for POD methods is parameter estimation. Section 4.2 will collect some results, supported by the validations offered in sections 3.1 and 3.2.

The next set of tests to be run should be focused on simulating the gap existing between the models used in any simulation and reality. These gaps can be closely approximated but, for some perturbations, such as irregular gravity or atmospheric model mainly, there is always a gap that causes errors in the dynamics that needs to be, at least partially, fixed by the POD process. This very important set of tests will be included in Section 4.3.

Another perspective could be taken if, instead of thinking about the scenario configuration or the dynamic models, more practical aspects are considered. Up to this point, all the proposed tests will be run using PVT measurements (as mentioned in 2.6), but other measurements could also be given as an input representing data with different characteristics. This will be treated in Section 4.4 with the case study of angle measurements.

Finally, Chapter 5 will apply the developed algorithm to a real-data case (GOCE mission), to serve as further comparison of the penalty functions, as well as a final validation of the developed POD process.

3 Sensitivity Analysis and Technology Selection

This chapter starts from the first version of the POD algorithm to be used during this thesis. In this stage, the concepts introduced throughout Chapter 2 are synthesised in a real simulation environment and the first tests are run.

However, before starting with the core of the tests (performed in Chapter 4) a set of validations needs to be made to ensure that the implementation of the software is correct and that a deep understanding of the behaviour of the algorithm is gained. Due to this, tests regarding convergence capabilities, error estimation, errors in the observations, or a study about how does the M parameter actually affect HPF will take place in this chapter.

3.1 Manoeuvres Validation

The validation of the manoeuvres will be based on the Gauss' form of Lagrange's planetary equations (Wakker, 2015, p. 601). These equations calculate the variation of each of the orbital elements accordingly to the state of the satellite and the perturbations involved. In this case, the perturbation will only be the manoeuvres. A close correlation is expected between the outcome of MyProp and the Gauss equations.

For the equations not to yield any singularity problem, scenario conditions will be set such that none of the critical orbital elements, i.e., e and i, is 0. Thus, all the tests are run in equal scenario conditions, characterised by the following elements:

Table 3.1 - Scenario conditions for manoeuvre validation

$\boldsymbol{a}\ [\boldsymbol{km}]$	10000.0
$\boldsymbol{e}\ [-]$	0.01
$\boldsymbol{i}\ [°]$	20.0
$\boldsymbol{\Omega}\ [°]$	10.0
$\boldsymbol{\omega}\ [°]$	20.0
$\boldsymbol{\theta}\ [°]$	150.0
Epoch	10th August 1993 08:00:00.00 UTC

Also, it will be assumed that $M \approx \theta$ based on $e \downarrow 0$ (Gauss' form of Lagrange's planetary equations uses M instead of θ). All scenarios would be considered to be Keplerian elliptic orbits only perturbed by the thrusting manoeuvre. These manoeuvres will be applied in each of the frame axes, in positive and negative directions. This makes for a total of 6 scenarios with identical initial conditions and varying manoeuvres.

$$manoeuvres = \begin{pmatrix} +1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & +1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & +1 \\ 0 & 0 & -1 \end{pmatrix} \frac{m}{s^2} \tag{3.1}$$

The loop runs *MyProp* for the established manoeuvre timespan, collects the ephemeris, and calculates the change in each of the classical orbital elements. After that, the Gauss´ equations are applied to compute what the theoretical change in each element should be. The Gauss formulation of these equations is used because the manoeuvre commands are given in orbital frame, which is the frame in which they are based.

Something to be taken into account is that these equations are linearized in the point where they are applied, so the value of each orbital element is included in the computation of its own rate of change. For this reason, higher accuracy requires an update of these elements within each time step. A very simple Euler-based numerical method is used to obtain this accuracy level.

After that, the results are compared and the relative error is computed, defined as the difference between both methods w.r.t. the change in the elements according to MyProp.

The results obtained for these tests are shown in this chapter in terms of relative errors. Additionally, specific cases are included where relevant behaviours that might be of interest for the reader will be pointed out.

3.1.1 Relative Error

When the set of manoeuvres shown in Equation (3.1) is run, the results shown in Table 3.2 are obtained.

Table 3.2 - Relative errors in each of the implemented manoeuvres.

manoeuvre	$\boldsymbol{a}$	$\boldsymbol{e}$	$\boldsymbol{i}$	$\boldsymbol{\Omega}$	$\boldsymbol{\omega}$	$\boldsymbol{\theta}$
1	0,0022%	0,0021%	0,0000%	0,0000%	0,0010%	0,0245%
2	0,0032%	0,0034%	0,0000%	0,0000%	0,0006%	0,0220%
3	0,0000%	0,0001%	0,0000%	0,0000%	0,0004%	0,0009%
4	0,0000%	0,0000%	0,0000%	0,0000%	0,0001%	0,0010%
5	0,0019%	0,0000%	0,0000%	0,0003%	0,0003%	0,0174%
6	0,0019%	0,0000%	0,0000%	0,0003%	0,0003%	0,0174%

The only comment worth making here is that the largest errors belong to the true anomaly (θ), what can be explained by recalling that it has been assumed that $M \approx \theta$ based on $e \downarrow 0$. Otherwise, the results are excellent.

3.1.2 Absolute Error

In order to get a bit more of insight, one of the scenarios is selected to show how the absolute results are like. All the cases show similar levels of accuracy, so scenario number 1 is selected as an example, but the conclusions can be extrapolated to any of the others.

In this example, a positive acceleration is applied onto the radial axis. According to Gauss´ equations (Wakker, 2015, p. 601), this should generate variations in every orbital element except in the inclination and the RAAN. When the plots shown in Figure 3.4 are observed, it is clear that this is exactly what happens, supporting the validation of the manoeuvres.

It is also interesting to see how the Gauss equation´s value for the orbital elements shows a slight curvature, based on the Euler-based integration approach used. On the other hand, the data collected from MyProp is a straight line. This is because, although a Runge-Kutta (RK) method is implemented within the software, the ephemeris obtained as an output are referenced to each time step, and the intermediate values are not registered. It has been explained that, in order for the Euler method to match the integration accuracy level of MyProp, around 100 steps are needed (while the RK method inside the propagator uses only 11 for the RK89 Verner´s model).

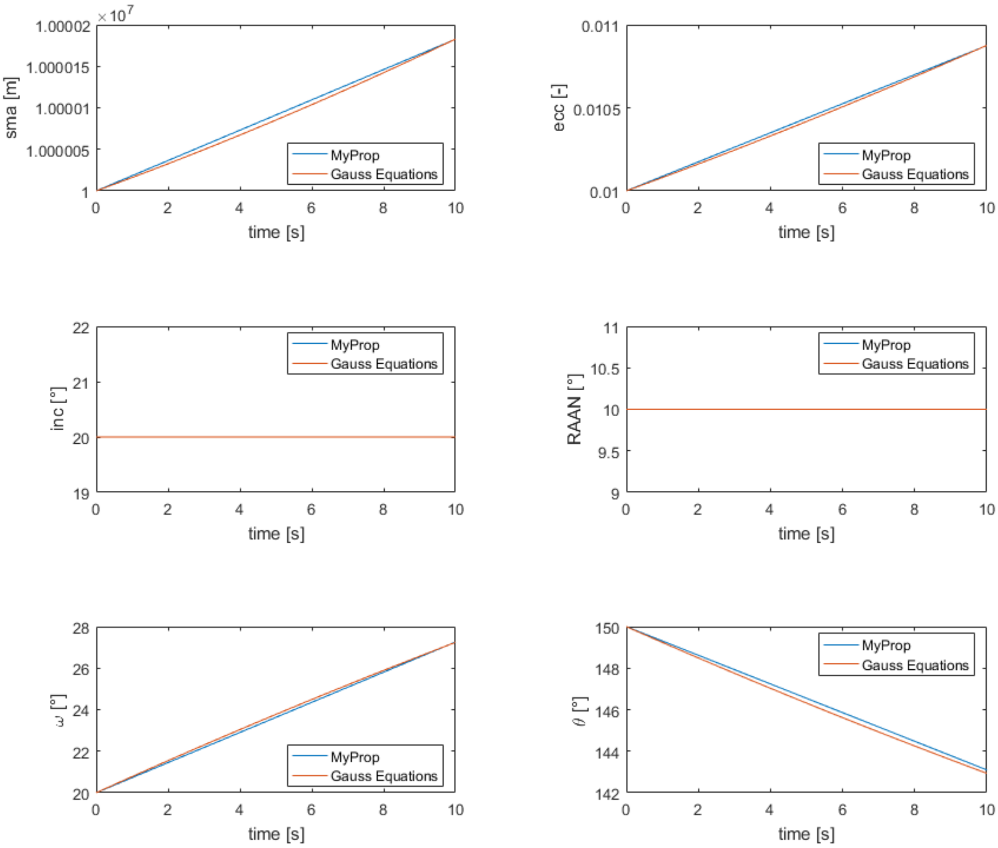


Figure 3.1 - Evolution of the classical orbital elements.

Based on these results, the implementation of the manoeuvres is considered validated.

3.2 State Transition and Sensitivity Matrixes Validation

In this section the validation process for the STM and SM will be shown. The methodology utilised is explained before the results are shown.

3.2.1 Methodology

First of all, a number of random scenarios are defined and saved so that the different validations are run for the same initial conditions. Although the scenarios are set to be random, there are some parameters that have been limited:

- timespan is fixed to 5 solar days,
- altitude ranges from $400-1000\ km$ (for the atmospheric models to be effective),
- eccentricity is set to be minor than $0{,}01$ (to avoid impacts with the Earth’s surface),

- inclination $\leq 45^o$ (explained in Section 3.2.2),
- ICRF, and
- epochs between 1970 and 2015 (limited by the input files for EOP or Space Weather).

A loop is run for each different scenario. At each iteration, the implemented POP is called, using the corresponding scenario conditions, and the trajectory, X_0, the STM, and the SM are obtained and stored. After that, a small perturbation to the initial state or the parameters (for STM or SM respectively) is applied and, again, the perturbed initial state is propagated using the POP to obtain $X_{0,pert}$. In parallel, the introduced perturbation is also propagated using the stored STM (and SM, if applicable) and its evolution, ΔX_0, is recorded.

After that, the outcome of the propagation (POP) of the perturbed state, and the sum of the propagation (POP) of the non-perturbed state and the propagated (STM or SM) initial perturbation are compared. This is:

$$abs\left(\boldsymbol{X_{0,pert}} - (\boldsymbol{X_0} + \boldsymbol{\Delta X_0})\right) \tag{3.2}$$

With this, absolute and relative information about the error can be easily obtained. The relative error is computed as:

$$relError = \frac{abs\left(\boldsymbol{X_{0,pert}} - (\boldsymbol{X_0} + \boldsymbol{\Delta X_0})\right)}{abs\left(\boldsymbol{X_{0,pert}} - \boldsymbol{X_0}\right)}\% \tag{3.3}$$

All data are stored and the plots shown in the following section are generated.

For the STM, twelve different scenarios were run for each perturbation. The reason for this number is that a perturbation will be introduced in the initial state in each of the components of the state vector (three for position and three for velocity) in each of the directions (positive or negative axis). Thus, twelve cases are needed.

Similarly, an additional set of tests was generated, based on the same scenario conditions, for the validation of the SM. The parameters used for the SM are c_D, c_R, and manoeuvres. The scenario used will be perturbed by atmospheric drag, solar radiation, and a single manoeuvre; so that any further variation of the involved parameters will have a repercussion in the trajectory.

This process is used for each of the different modules involved in the computation of the STM within the POP, i.e., Keplerian, irregular gravity, Sun and Moon gravitational pull (separately), SRP, and atmospheric drag. Also, tests for all together was run. Defining a threshold as acceptance criteria is not simple for this validation, as it would depend on the application case and the complexity of the problem.

3.2.2 State Transition Matrix

The magnitude of the perturbations introduced for this validation test are

$$\Delta \mathrm{X}_0^{\text{initial state}} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \mathrm{v_x} \\ \Delta \mathrm{v}_y \\ \Delta \mathrm{v}_z \end{bmatrix}^{ICRF} = \begin{bmatrix} \pm 10\,m \\ \pm 10\,m \\ \pm 10\,m \\ \pm 0.01\frac{m}{s} \\ \pm 0.01\frac{m}{s} \\ \pm 0.01\frac{m}{s} \end{bmatrix}^{ICRF}, \tag{3.4}$$

and the results are shown in the following subsections. The perturbations for the velocities were set to be 10^{-3} times smaller than the ones for positions.

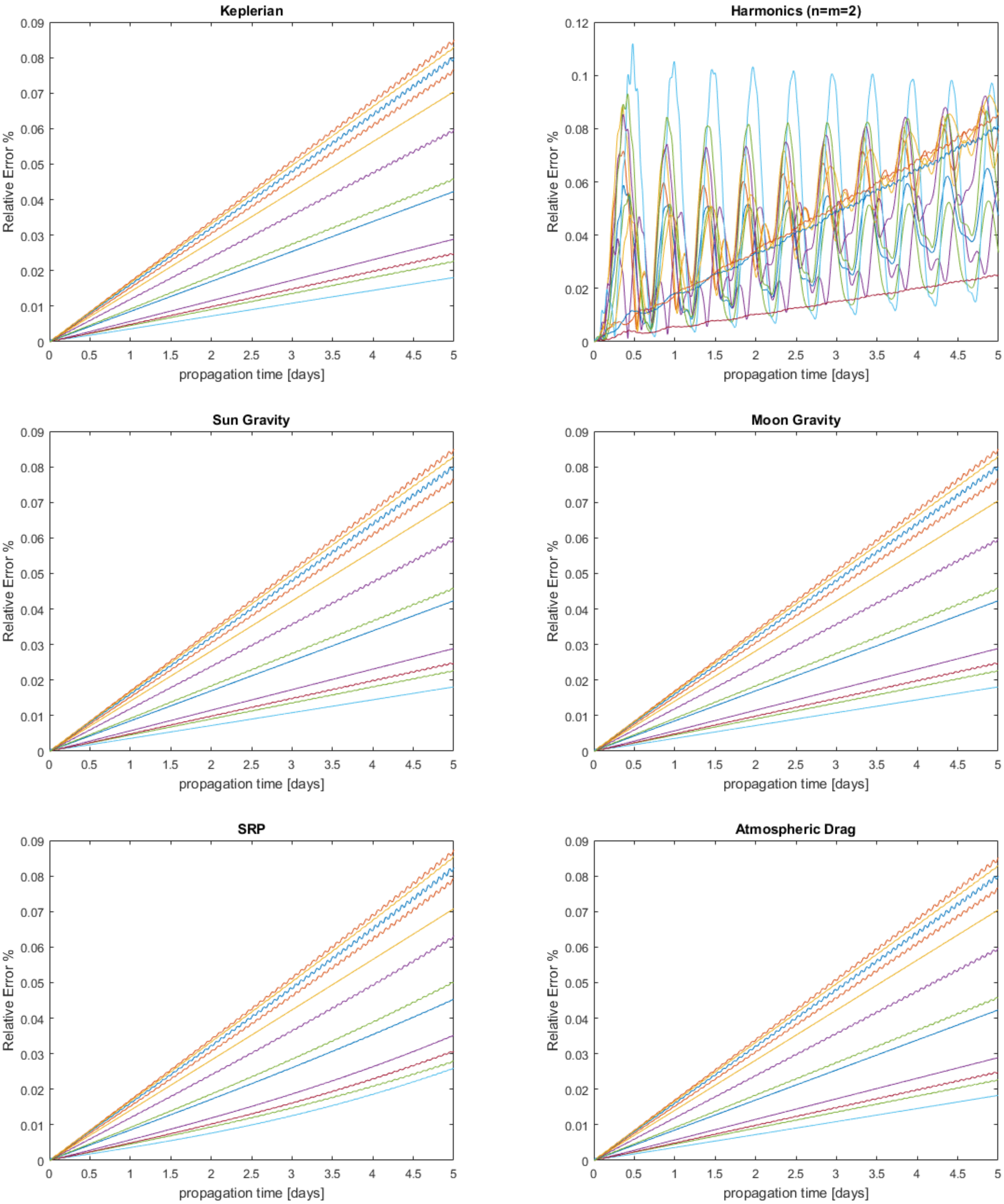


Figure 3.2 - Relative errors for each individual perturbation applied.

3.2.2.1 Relative Error

When the tests for each perturbation are run, the results shown in Figure 3.2 are obtained. There, it can be observed that the relative errors do not exceed 0.12% for any of the cases or perturbations applied. Moreover, a linear evolution of the relative error is clearly appreciated in most of the force models. This means there is a linear increment in the error of the propagated perturbation. A relatively large timespan was selected for the propagation to evaluate whether there was a certain timeframe after which it is not recommended to use a STM-based approach for the orbital propagation. However the results shown offer very accurate results in this sense, and it can be concluded that timespans over 5 days in a LEO would not exceed a 0.15% relative error.

The tests for the Earth´s irregular gravity field, nevertheless, show a particularly strange behaviour. Compared with the other plots, it is the only one where the relative error does not show a clear linear behaviour for every case. There are, however, three cases where the linear behaviour holds (coloured in red, blue, and orange from the bottom to the top).

These three cases represent the scenarios where the inclination of the orbit was lower. Moreover, if these scenarios are fetched in the other plots, for example in the Keplerian test, it can be seen that they remain practically unchanged (as they do for the rest of the perturbations).

Therefore, it is possible to conclude that inclination values close to 90° generate larger relative errors. Nevertheless, there is no clear proportionality between the value of the inclination and the magnitude of the relative error. With the original constraint on the inclination ($i \leq 90°$), and due to the random nature of the generated scenarios, all test cases showed the same behaviour. As it was also important to show the low inclination cases in the analysis, the inclination was constrained to be under 45°, so that more low inclination cases were generated to picture both behaviours.

Considering that an order and degree 2 for the harmonic gravity field was used (as recommended in (Montenbrück & Gill, 2000)), it makes sense that the orbits with lower inclination suffer a more continuous perturbation, i.e., they do not show the oscillations that are seen in the plot, since the mass underneath the satellite is more regularly distributed according to the harmonics used. Also, when the trend of each of the errors is analysed, a linear component can still be observed underneath the frequency oscillation (related to the relative motion of the surface of the Earth under the orbit).

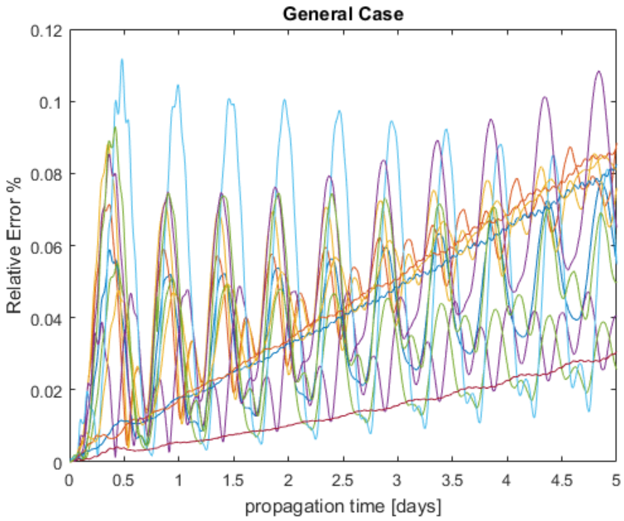


Figure 3.3 - General case validation

The magnitude of the relative error increases with respect to the other cases. This is due to the fact that the initial perturbation is propagated using a linearized procedure. The more irregular the perturbation is, the less valid the linearization becomes, and, thus, the higher the errors.

After each of the perturbations has been validated, a general case is run, where all the perturbations are run together. The results are shown in Figure 3.3, prove that the error due to the irregular gravity field overtakes the rest of the error and shapes the general relative error (mainly because of the sinusoidal layer that it adds).

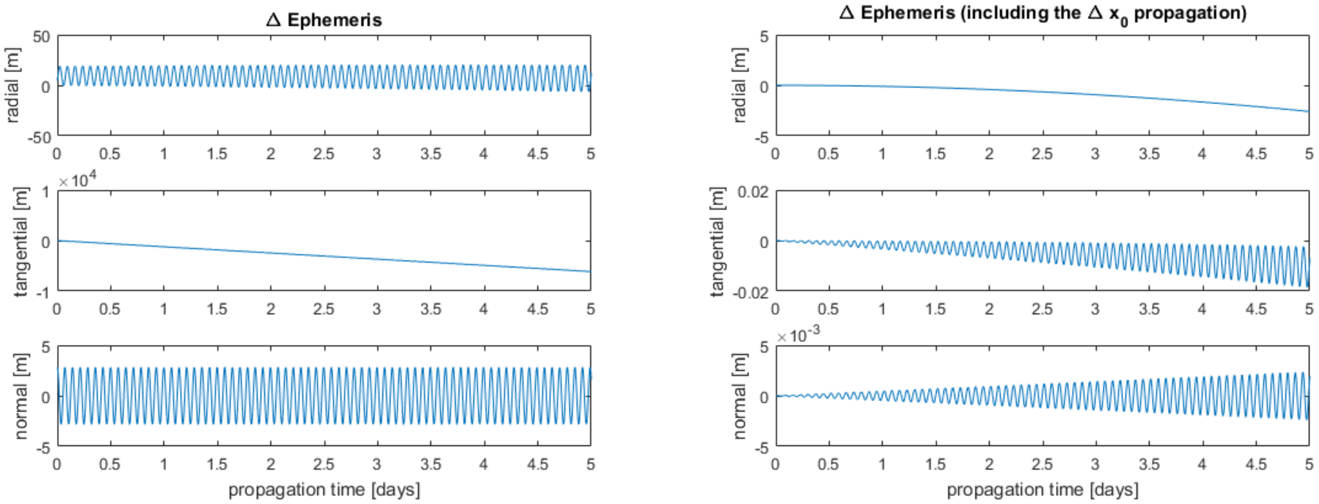


Figure 3.4 - Absolute error in orbit centred reference frame.

3.2.2.2 Absolute Error

It is also very insightful to see how the components of the state evolve. In the right plot of Figure 3.4, it can be seen that the errors are maintained well below the centimetre level for the tangential and normal directions, whereas the radial component shows the largest error. This kind of behaviour is interesting, overall when the original difference between both ephemeris was much larger in the tangential component instead.

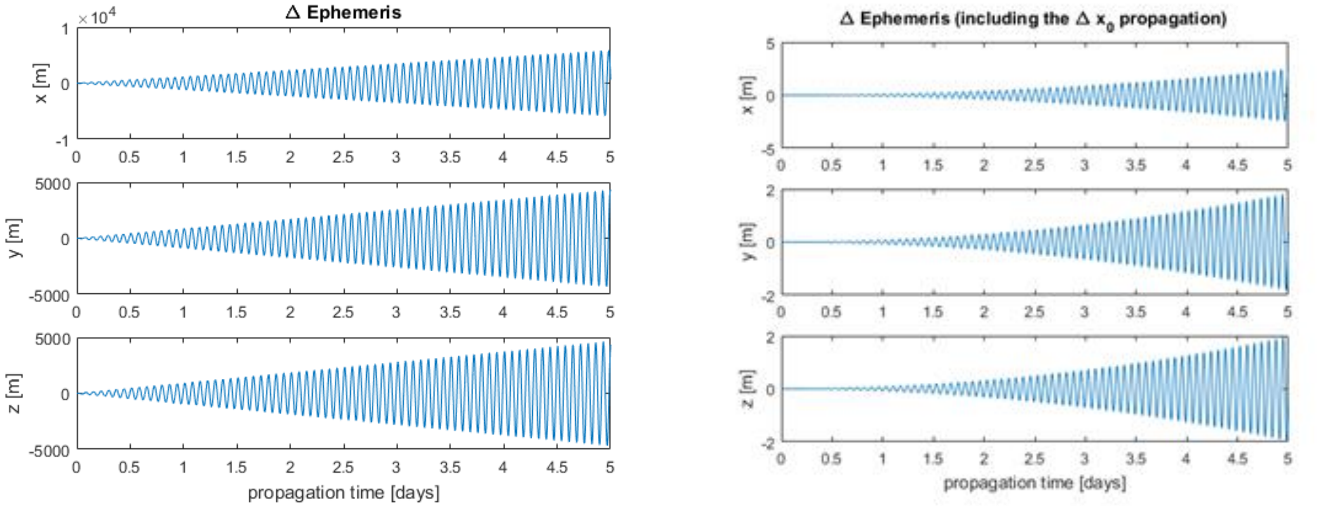
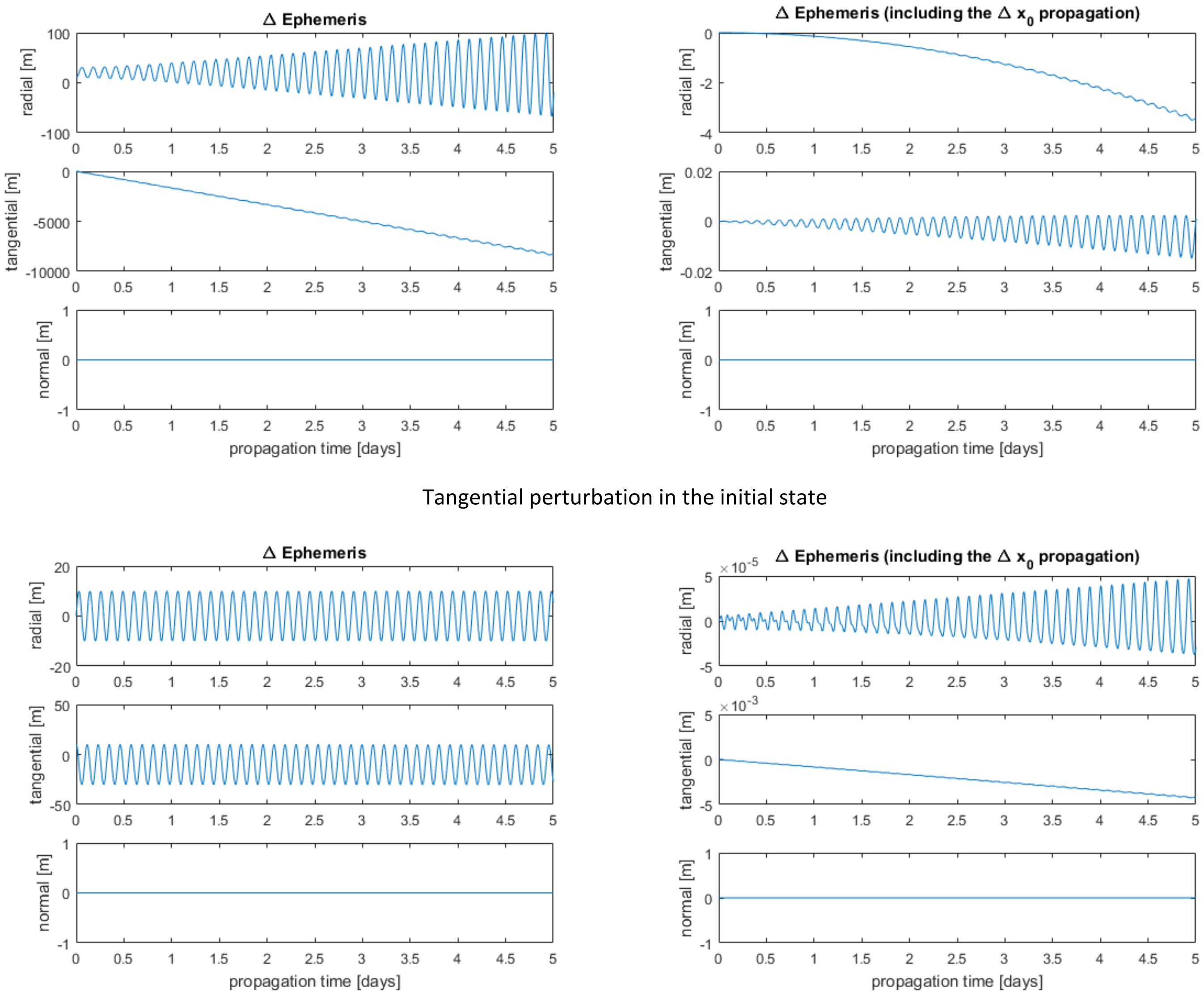


Figure 3.5 - Absolute error in ICRF.

The propagation of the Δx as a function of time, is linearly related to the STM values and the introduced Δx_0. This means that the accuracy will be affected by this linearization error. As it was explained in Section 2.3, the STM generation involves the calculation of the partial derivatives of the acceleration w.r.t. the satellite´s state. Because of this, the more the acceleration the satellite goes under varies, the less accurate a linearization would be around that point. In this case, the component in which this varies the most is the radial component, since the main acceleration is the gravity from the Earth. However, for sufficiently small Δx_0´s (as it is the one applied in these tests), this concept does not justify the magnitude of the error seen on Figure 3.4.


Radial perturbation in the initial state

Tangential perturbation in the initial state

Figure 3.6 - Absolute error in orbit centred reference frame for perturbation only in the radial (top) and tangential (bottom) direction. Study case with a=10 000 km, e=0.01, i=Ω= ω= θ=0°, and $\Delta x = 10\ m$.

A different possibility is that the transformation of the obtained ephemeris to the reference frame of use (orbital frame) is generating some sort of error. To check this, the same data are plotted in ICRF (see Figure 3.5) and it can be observed that the error remains constant. Thus, this hypothesis is discarded.

In order to study the relation between the initial state perturbations and the errors, a fixed scenario is selected (differently from the random set used in the validation) onto which perturbations will be applied in the radial, tangential, and normal directions, i.e., in orbital frame and not in ICRF axes anymore. The results for the radial and tangential perturbations in the initial state are shown in Figure 3.6.

From these results, it seems likely that a relation between the tangential error before the inclusion of the Δx propagation and the radial error after the latter (middle-left and upper-right subplots of Figure 3.6, respectively) exists. To obtain a better insight on the relative motion of both satellites, plots are generated representing the position of the trajectory starting from the perturbed initial position w.r.t. to the position of the original trajectory, and the results are shown in Figure 3.7.

When the initial perturbation is applied in the radial direction, a drift in the tangential direction is observed (as expected, according to orbital dynamics; see Clohessy-Wiltshire equations in (Wakker, 2015)). Nevertheless, a very interesting behaviour appears: the difference between the initial minimum radial distance $(10\ m)$ and the final minimum radial distance $(6.71\ m)$ coincides almost exactly with the magnitude of the error appearing in the radial component after the $\mathit{\Delta} x$ propagation was added ($3.29\ m$, see upper-right subplot of Figure 3.6).

On the other hand, the tangential $\mathit{\Delta} x_0$ shows no apparent drift in any component but only a periodic oscillation around satellite 1.

This test was repeated for different magnitudes of the $\mathit{\Delta} x_0$ and different scenario conditions, but the results were still the same. This proves that the error reflected in the radial upper-right plot of Figure 3.6 does not come from the STM propagation but from somewhere else.

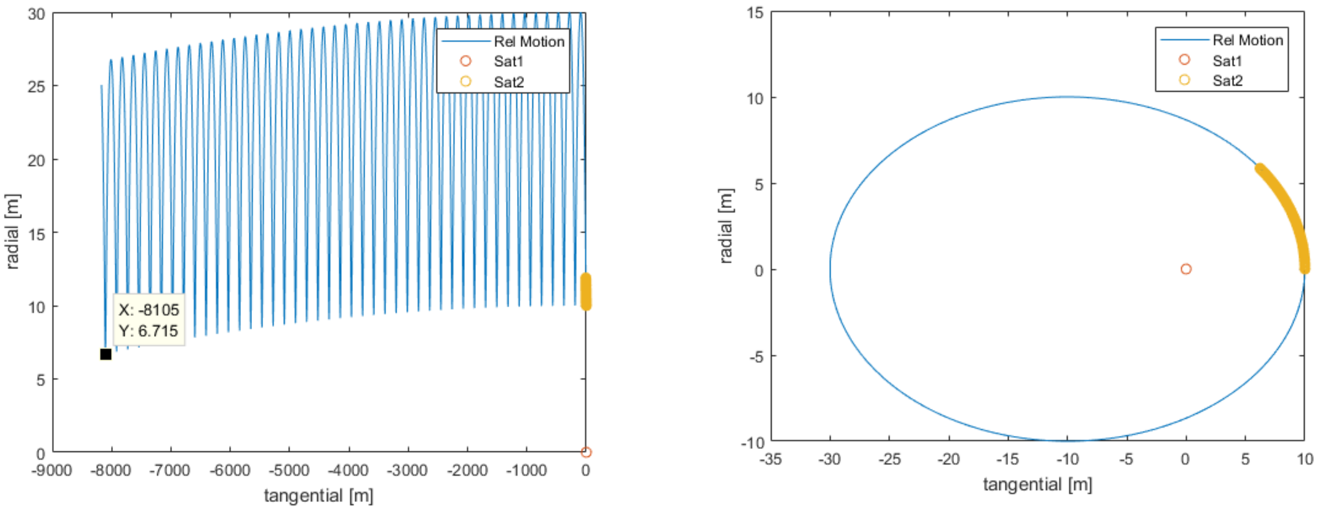


Figure 3.7 - Relative motion of the trajectory with the perturbed initial state w.r.t. the original trajectory. Left side for radial perturbation and right side for tangential perturbation.

It is quite intuitive that the arc shown in the left-hand side of Figure 3.7 resembles the curve that an orbit would follow throughout its trajectory. To check if this is the case, this hypothesis is tried estimating the error that a curved trajectory would generate when measured with respect to straight axes.

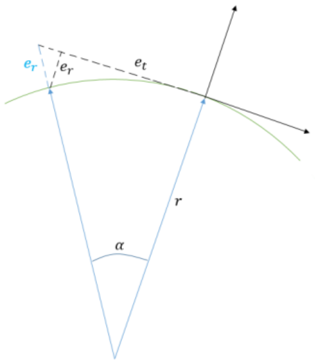


Figure 3.8 - Geometry of a circular trajectory with an orbital reference frame.

The aim is to compute e_r, for a given e_t and r. For that, and based on the scheme shown in Figure 3.8, a value of $\alpha \ll 1°$ is assumed:

$$\alpha \approx \tan\alpha = \frac{e_t}{r} \tag{3.5}$$

$$e_r = \frac{r}{\cos\alpha} - r$$

Applying these formulas to the case shown in Figure 3.6:

$$r = 10\,000\ km\,, e_t = 8{,}105\ km \rightarrow\ e_r = 3{,}29\ m$$

Thus, it is proved that the error in the radial component is only due to the nature of the curved (almost circular in this case) trajectory that a satellite follows and the fact that the reference frames use straight axes. It can also be said that the tangential distance will map the error to be expected in the radial component of the corrected trajectory.

Finally, it can be observed that, apart from this error, the propagation of Δx is able to handle the errors due to the orbital dynamics involved with almost perfect accuracy. This can be seen in the case where the perturbation in the initial position was only tangential, where the errors after the correction are well below the centimetre level (see bottom-right plot of Figure 3.6).

3.2.3 Sensitivity Matrix

The perturbations used for the validation are:

$$\Delta\text{parameters} = \begin{bmatrix} \Delta c_D \\ \Delta c_R \\ \Delta man \end{bmatrix} = \begin{bmatrix} 10^{-2} \\ 10^{-2} \\ 10^{-6}\ m/s^2 \end{bmatrix}. \tag{3.6}$$

The differences in the perturbations are due to the magnitude of the acceleration they are related to. While the accelerations coming from atmospheric drag and SRP are usually relatively small (see Figure 2.2), the manoeuvres' magnitude is quite high, in comparison. Additionally, the variations on the drag acceleration and SRP are introduced in their corresponding coefficients, whereas the variation for the manoeuvres is directly introduced in its value.

The initial values for the parameters are

$$\text{parameters} = \begin{bmatrix} c_D \\ c_R \\ man \end{bmatrix} = \begin{bmatrix} 2.2 \\ 1.2 \\ 0.001\ m/s^2 \end{bmatrix}, \tag{3.7}$$

where the perturbation for the manoeuvre is applied equally in all the components of the orbital frame (radial, tangential, and normal), after one day of propagation, and lasting $10\ s$.

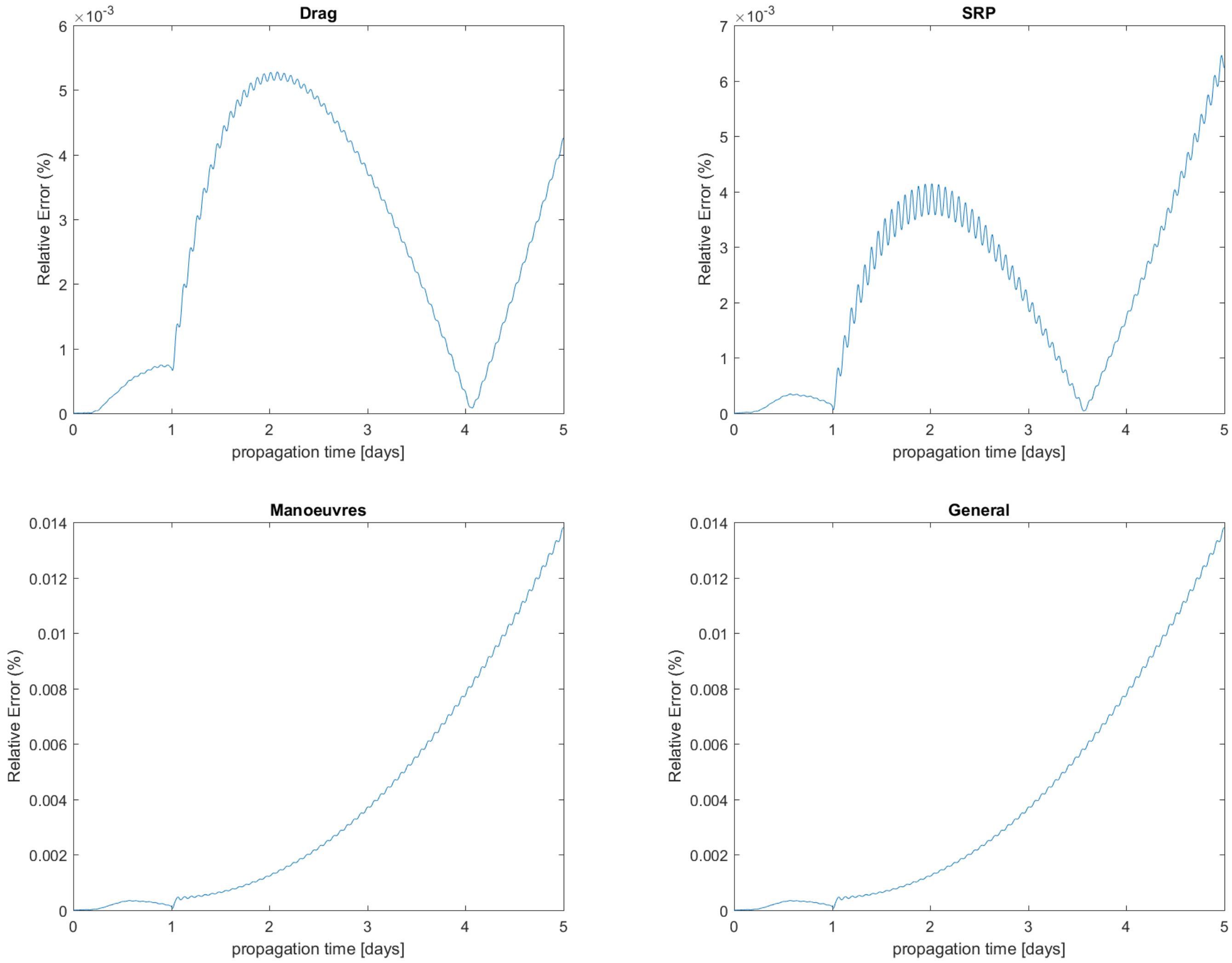


Figure 3.9 - Relative errors for each individual parameter perturbation applied[4]. Manoeuvres of 0.001 m/s^2.

3.2.3.1 Relative Error

The results for the relative error are shown in Figure 3.9.

It is clear that, for the magnitudes used for these tests, the manoeuvre acceleration is the most affected by a coefficient variation. Even if the change is still very small (below 0.02% relative error), it drives the general case, where all three perturbations in the coefficients are applied.

All the plots show a drastic change of behaviour after the manoeuvre is implemented (after one day). This is the clear reflection of the STM-SM propagation process being strictly linear. A manoeuvre creates an extremely abrupt and sudden change in the force that the satellite experiences, generating a huge increase of the relative error.

Because of that, it could be interesting to find out what happens when the acceleration is some orders of magnitude higher. The same process is repeated for manoeuvres of 0.01 and $0.1\ m/s^2$ to obtain the results shown in the figures below and below, respectively. The perturbations will

[4] Note that the changes in concavity/convexity of the plots of the error are due to the analysis being processed in absolute magnitudes.

be again three orders of magnitude smaller than the command, i.e., 10^{-5} and $10^{-4}\ m/s^2$ respectively.

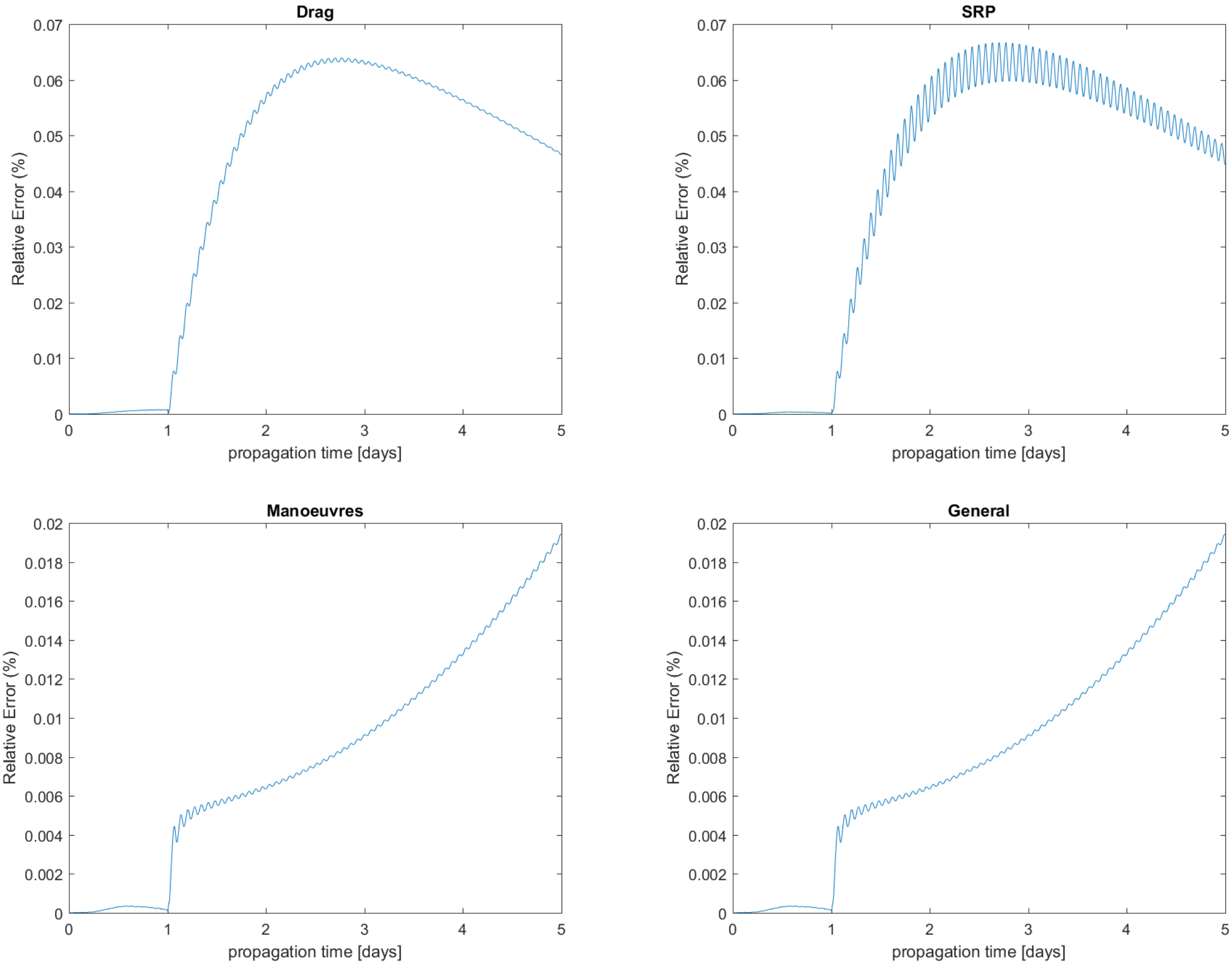


Figure 3.10 - Relative errors for each individual parameter perturbation applied. Manoeuvres of 0.01 m/s².

Both cases show similar behaviours. For perturbations in the drag or SRP coefficients, the relative error is multiplied by a factor 10 (Figure 3.10 and factor 100 (Figure 3.11), approximately, due to the dependence of both acceleration on the position of the body. The more this position is altered (with stronger manoeuvres), the further from the point of linearization for the STM/SM the satellite will be, and the higher the relative error will become.

It is worth remarking that the perturbation in the manoeuvres shows now a smaller relative error than in the other parameters. Also, although it grows as well, this relative error remains more stable, and keeps driving the general case. This is due to the perturbation in the manoeuvre instructions having a greater effect in the trajectory of the body than a change in its aerodynamic or reflective coefficients (this will become clearer in the next section, where the absolute error is shown).

3.2.3.2 Absolute Error

As expected, the larger absolute error is given by the case where the manoeuvres are perturbed, as observed in Figure 3.12). It clearly dominates the general case and explains why the relative errors shown before follow the manoeuvre perturbation behaviour, as it was introduced before.

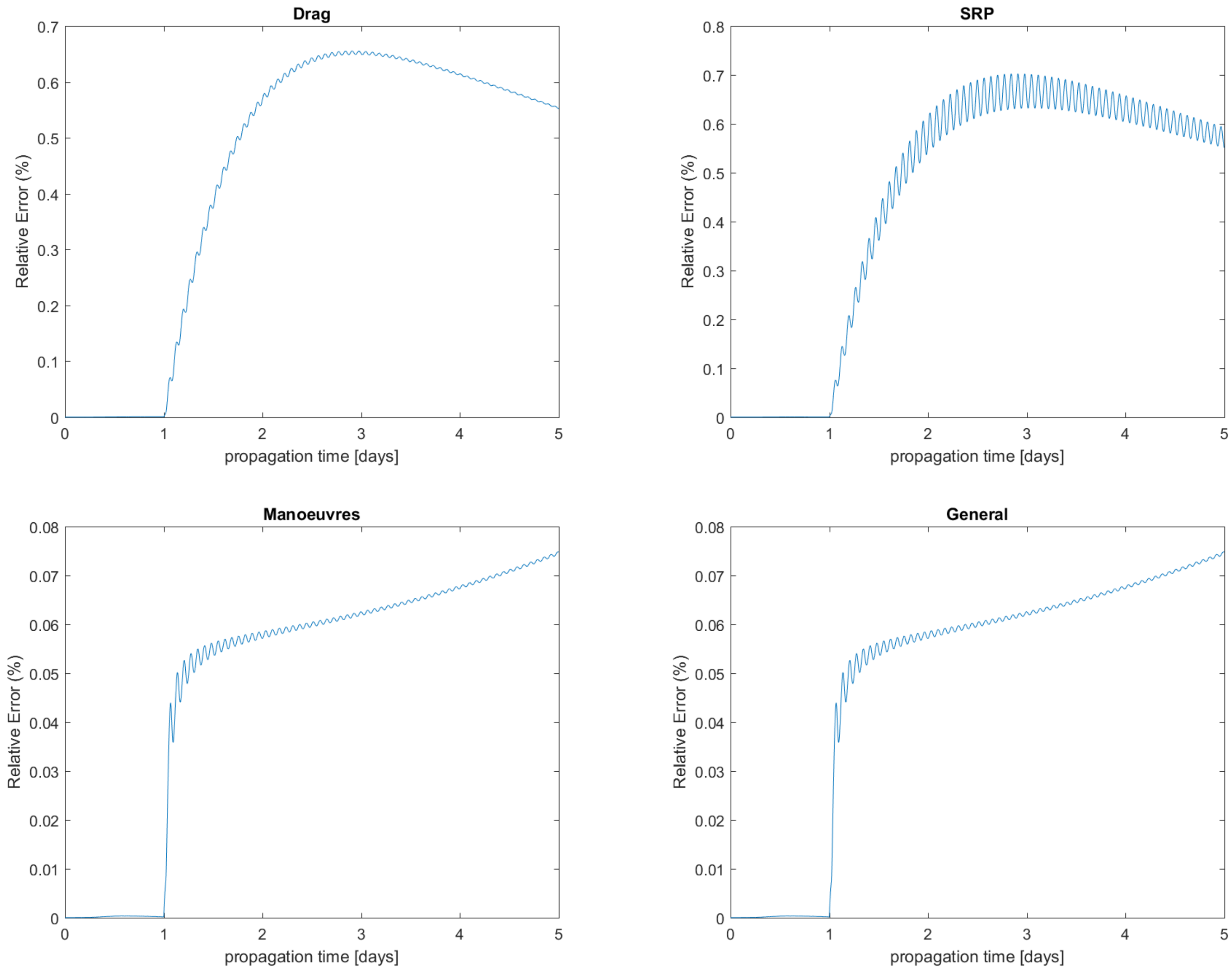


Figure 3.11 - Relative errors for each individual parameter perturbation applied. Manoeuvres of 0.1 m/s².

3.2.4 Conclusions of the Validation

Even though the results are extremely satisfactory and the errors are low enough to warrant a high accuracy for long timespans and relatively large perturbations, care must be taken regarding the magnitudes and behaviours of the forces involved in the orbital problem. The observed errors are mainly inconsistencies related to the tangential drift between both trajectories, because of curved nature of the orbital trajectories. Also, the errors grow linearly with the magnitude of the Δx_0.

In case manoeuvres are applied, it has been shown that, even if they are extremely accurate when used within MyProp (see Section 3.1), they introduce a highly non-linear step in the accelerations that the satellite undergoes and increase the error of the STM-/SM-based propagation.

With these two main points in mind, and based on the excellent results shown and explained in this section, it is possible to ensure that the STM and the SM are validated, and ready to be used in a POD process.

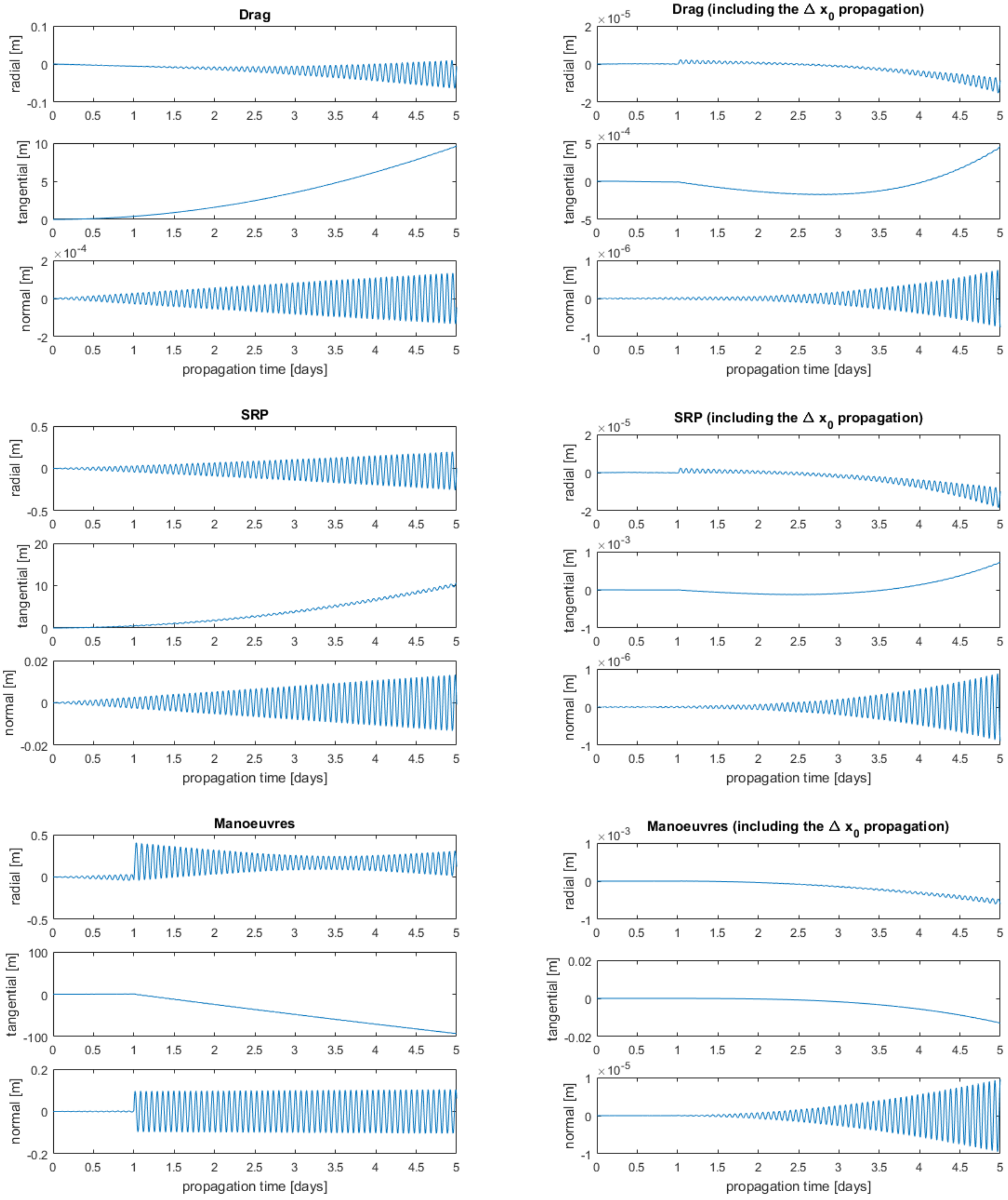


Figure 3.12 - Absolute error in orbit centred reference frame.

3.3 Convergence Capabilities

The first aspect that should be tested is the convergence of the optimisation toolbox itself, to make sure the penalty functions are correctly implemented. POD methods are based on solving minimisation problems and, if a solution for this minimisation problem cannot be found, there will be no trajectory estimation.

The cost function for POD applications, whose generic form was introduced in Section 2.3, is shown in Equation (2.17). The outcome of this equation is the value that the optimisation problem tries to minimise, i.e., the optimal value.

At this point it is very important to differentiate between the two convergence loops present in this algorithm: the inner loop and the outer loop.

The inner loop refers to the iterative process followed by the CVX tool to find a solution for the optimisation problem. The convergence of this method is based on the evolution of the optimal value throughout these iterations, and on certain threshold configurations of the CVX toolbox (see (Grant & Boyd, 2017)). The result obtained from this loop is conceptually equivalent to the solution obtained from the traditional LSQ solution, as represented in Equation (2.47). Once a solution is found, it is taken as the current estimate of the POD method, x^*, and the algorithm continues.

The outer loop is the classical iterative process used in batch methods, where the estimated solution is compared to the previous value, until the difference between both, x^*, is below a certain tolerance level. The exact value is taken using the infinity-norm for the residual of the parameters, x^*. Figure 3.13 offers a simplified view of the process.

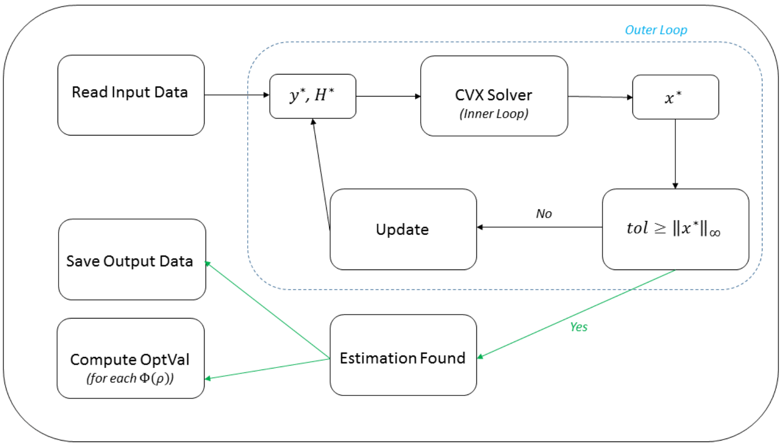


Figure 3.13 - Schematic overview of the POD batch algorithm.

As it is the object of this thesis to deal with convex functions, the optimisation problem stated in the inner loop will always be theoretically possible. However, numerical issues might occur due to the used toolbox not reaching the desired convergence, or to the evolution of the optimal value not being properly assessed (more details in (Grant & Boyd, 2017)). These numerical situations are usually triggered by conflicts between the tolerances of the toolbox and the outcome of the penalty function, errors in the measurements or the model that exceed the magnitudes that the optimisation method can handle, or a wrong problem set-up.

To evaluate convergence, a number of tests were run in this section to understand how different the optimal values for each penalty function are. The methodology consisted of running the POD algorithm using a different penalty function in the inner loop each time. The solution of each penalty function, was given to the other penalty functions to compute the "optimal" [5] value they would yield. The aim of this experiment is to evaluate how the same trajectory estimation can be represented by the different definitions of the penalty functions implemented and, thus, understand the order of magnitudes in which each of these functions usually works.

This set-up yields a total of nine values for each test configuration (three "optimal" values are computed when the LSQ method is active, another three when the L1-Norm is, and another three when HPF is).

The result of this optimal value depends on basically two factors: the number of measurements, and the residual in each measurement. The latter, in time, also depends on different aspects, such as, the quality of the model w.r.t. the observations, the noise in the observations, or the number of outliers, among others. From the mentioned sources, two are known to a much finer accuracy: the number of measurements, and the expected error in the observations (see sections 2.3.1 and 2.8). Proportional variations in these two factors should intuitively yield similar optimisation results.

The graphical representation of this methodology is rather dense and requires a detailed explanation. In the figures shown in this section, the colours labelled in the legend represent the penalty function that is active in the inner loop for each test, while the X-axis represents the function used to compute the corresponding optimal value (Y-axis) for each parameter estimation. This means, every group of three bars with the same colour are computed based on the same trajectory estimation. When convergence is not reached, the bars will not be displayed. Groups of nine bars (in red, green, and blue) refer to identical testing conditions, and when groups like these are concatenated, it means that the testing conditions where altered from one to another group, in order to obtain a comparative result.

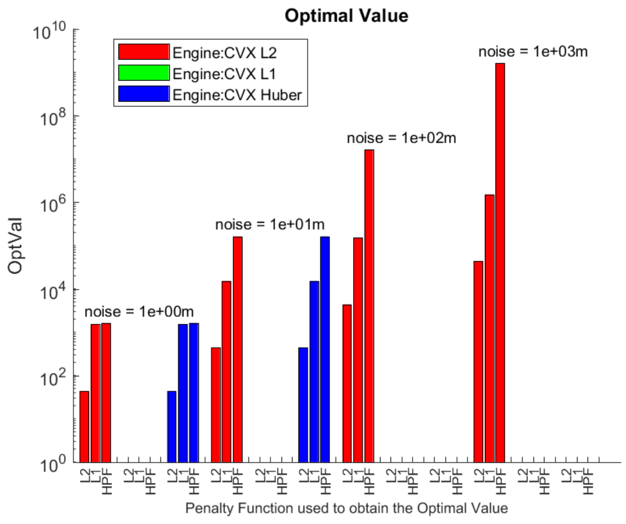


Figure 3.14 - Optimal Values for the different penalty functions with varying stochastic error magnitude.

First of all, the stochastic noise level is evaluated, and results are shown in Figure 3.14. It can be seen that, as the trajectories estimated using each penalty function are indeed quite similar (as it will be shown in Figure 3.25, on page 65), the optimal values obtained with each function (indicated in the X-axis) are also practically constant for each active CO method (indicated by the colour). However, each penalty model has its own definition, as introduced in Section 2.5,

[5] Bear in mind that, the optimal values calculated will not be exactly the ones that would appear if a different penalty function was used in the inner loop, because the estimated trajectory would be distinct.

and, therefore, the optimal values change from one function to another, even when the same trajectory estimation is used, e.g., HPF usually presents a higher optimal value than the rest of the penalty functions, for identic trajectory estimation results.

The σ value is varied for the noise. In these tests, the L2-Norm is the only penalty function able to converge always. HPF finds a solution for the first two cases, and the L1-Norm is not able to reach convergence from the beginning. Ironically, the test-cases where the noise magnitude is larger, and where a more robust penalty function would be of more use, only the L2-Norm, which is known for being largely affected by the presence of outlying measurements, is able to yield results. This fact represents a need for an improvement in the convergence capabilities of the penalty functions. It can be observed that the optimal value of HPF escalates at a faster rate than the others as the noise grows bigger.

If the stochastic noise level is maintained constant, but the number of measurements varies, the results shown in Figure 3.15 are obtained. There, five and ten times more observations are used.

It is important to remark that, in order to maintain the same conditions for the testing while increasing the number of measurements, it is the observation frequency and not the timespan that should be increased. Both can virtually generate the same extra number of observations, however, when the timespan is increased, residuals tend to increase more, due to the fact that the longer a propagation lasts, the larger the repercussions in the residuals from small errors in the initial state become. Because of this, an increment in the timespan of the trajectory, when trying to increase the number of observations, would not only increase the optimal value because of the studied parameters, but also due to error drifts in time originated by small disturbances in the initial state.

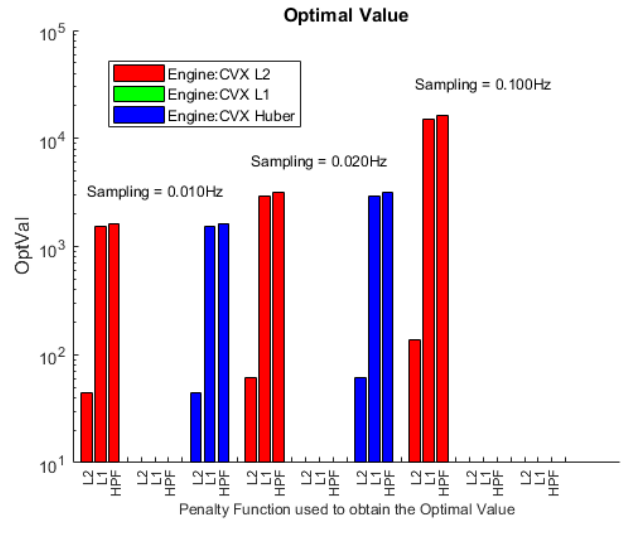


Figure 3.15 - Optimal Values for the different penalty functions with different number of observations.

In the previous figures, it can be observed that the L1-Norm already fails to find convergence for the nominal case ($\sigma = 1\ m$ and sampling frequency of $0.01\ Hz$). HPF performs better in this sense but the optimisation tool also fails to converge for the most demanding cases. The L2-Norm keeps yielding excellent results in this aspect.

3.3.1 Equivalent Problems: Scaling

With this information, a different approach is necessary to improve the convergence capabilities of the studied penalty functions while leaving the final estimation unchanged. In (Boyd & Vandenberghe, 2009), equivalent problems are introduced as a way to deal with issues like this. One the most frequent (and most simple) ways to handle this situation is scaling.

Scaling a minimization problem is nothing else than using non-negative scalar values to multiply the cost function and/or the constraints, to accomplish an easier resolution of the problem. Scaling is a way to deal with, for instance, large differences in the orders of magnitude between the optimised parameters and the solver's tolerance levels. If a scaling factor is not finely tuned, the sensitivity of the solver can be negatively affected and a solution might be found that satisfies the tolerances but does not solve the original problem. In fact, a trade-off solution should be found that satisfies not only the convergence of the CVX toolbox, but also of the POD algorithm itself. With this technique, the scaling parameters will be kept within the $(0, 1]$ range, since there will be no need for augmentation, 0 values yield invalid solutions, and negative scalars are not allowed if convexity is to be preserved.

Typically, space optimisation problems use scaling parameters to work with dimensionless quantities, such as the radius of the Earth, or the semi-major axis of the orbit; depending on the problem. In a POD algorithm, the problem to be solved is the linearization of the orbital dynamics that uses as parameters the residuals of an estimation and not the estimated values themselves (as explained in Section 2.3.3). For this reason, a proper value for the scaling process would not be the components of the state vector of the satellite but the initial estimation of the error existing in the observations. This concept is, in fact, related to another factor already discussed in this report: the covariance matrix (Section 2.3.1.1).

If the covariance matrix is used in the formulation of a cost function for a POD algorithm, the solution obtained is known as the weighted method (example for the LSQ method in Section 2.5.1). When a weighted method is used, convergence is improved, although some problems for large stochastic errors linger, mainly for the L1-Norm and HPF. Finally, and accounting for the number of observations, it has been observed that the number of observations (for a fixed timeframe) contributes to the optimal value as well.

Taking all these aspects into account, a scaling factor, S, is chosen:

$$S = \sqrt{Q_y}\, N_{obs} \tag{3.8}$$

where Q_y is the covariance matrix, defined by Equation (2.50); N_{obs} is a scalar accounting for the number of observations:

$$N_{obs} = \frac{1}{n_{obs}} \tag{3.9}$$

Using this scaling method, new results are offered for the previous tests shown in Figure 3.14 and Figure 3.15.

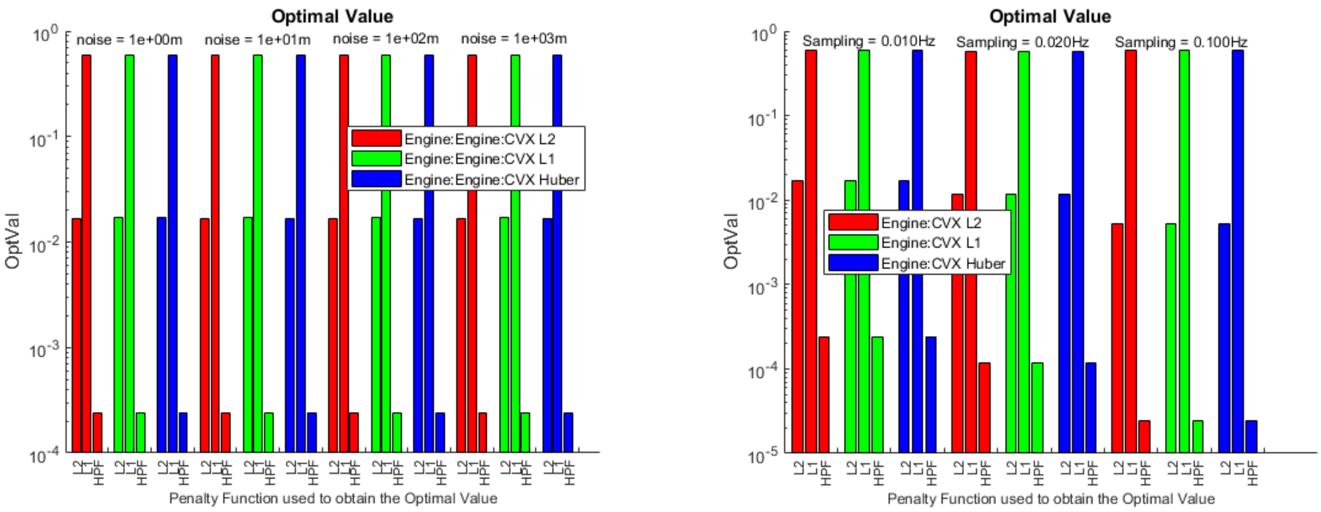


Figure 3.16 – Results for optimal values with scaling factors.

Figure 3.16 shows that the scaling parameters improve hugely the convergence capabilities of the different penalty function minimisations. It can be observed that, thanks to the covariance matrix scaling, the magnitude of the noise does not affect anymore the optimal value, since an inversely proportional factor is used. For the sampling frequency scaling, the utilised factor reflects a similar behaviour.

The last remarkable point is that, even though identical scaling parameters were applied to every penalty function, HPF experienced a much steeper reduction of its optimal value, mainly due to its mathematical definition and the way in which the scaling factors are applied.

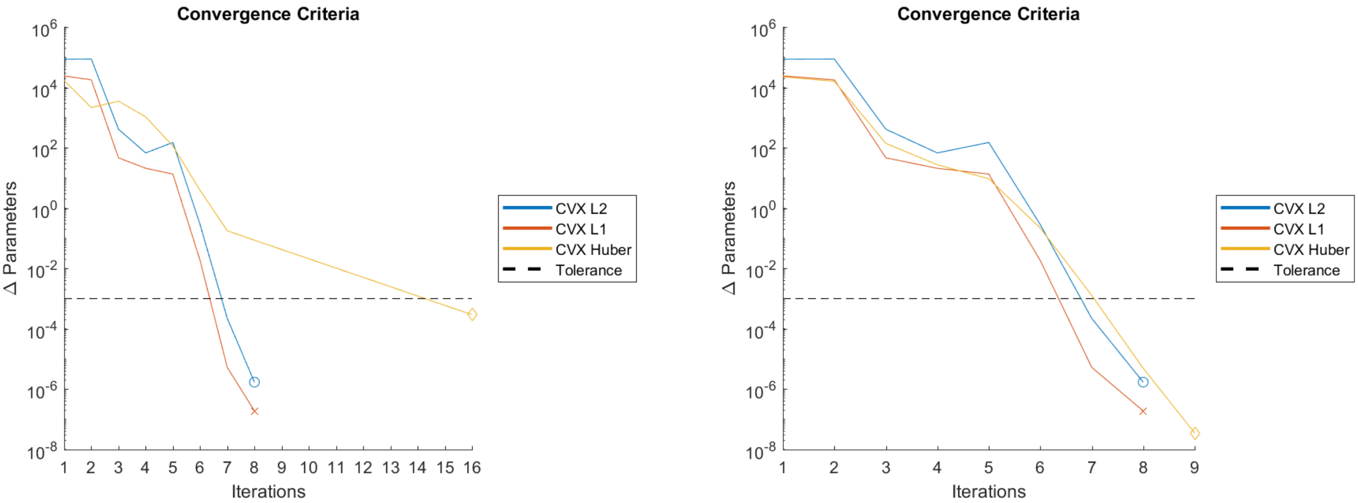


Figure 3.17 – Convergence criteria without (left) and with (right) the Pf scaling factor for $\sigma = 10^3 m$.

When the test results are analysed, it is observed that, for the most demanding cases (tests with high σ values), HPF converges with a slower rate than the other penalty functions (as shown in the left-hand plot of Figure 3.17); something that does not happen for other noise magnitudes. This can be related to the mentioned reduction of its optimal value, that may result in a loss of sensibility of the minimisation process and, thus, a reduction in the convergence rates of the POD.

To compensate for this, an additional scaling parameter is defined, accounting for the different penalty functions used (Pf). This parameter is empirically based on the behaviour shown above and is defined as

$$Pf = \begin{pmatrix} L2_{norm} \\ L1_{norm} \\ HPF \end{pmatrix} = \begin{pmatrix} 10^{0} \\ 10^{-1} \\ 10^{1} \end{pmatrix} \tag{3.10}$$

that would reshape the total scaling parameter to

$$S = \sqrt{Q_y}\, N_{obs}\, Pf. \tag{3.11}$$

With this extra parameter, the results for the optimal values are shown in Figure 3.18, where the expected effect is shown, resulting in an improvement on the convergence rates of the POD process (see the right-hand plot of Figure 3.17).

3.3.2 A Posteriori Changes

Further in this document, Section 4.2 will introduce the use of the SM as a way to estimate design parameters such as c_D, c_R, or manoeuvres[6]. When the tests were run, the CVX toolbox *failed* to find a solution for the minimisation of HPF.

The reason for that might actually not be the introduction of the SM itself, but the introduction of parameter variations in the initial state of the trajectory that differ from the ones set for reference trajectory, and, therefore, the measurements. This initial gap creates an initially higher error in the estimation that result in a higher optimal value. When this optimal value is not low enough, the CVX toolbox labels the result as *failed* (see (Grant & Boyd, 2017, p. 47)), because it considers the progress made towards a solution not adequate. This can be indirectly fixed by applying a modification on the scaling factor, S, that acts when the SM is active (even though its activation is not the immediate reason). The component of HPF in Pf needs to be reduced to make the optimal value fall within the same range it was falling in Figure 3.18.

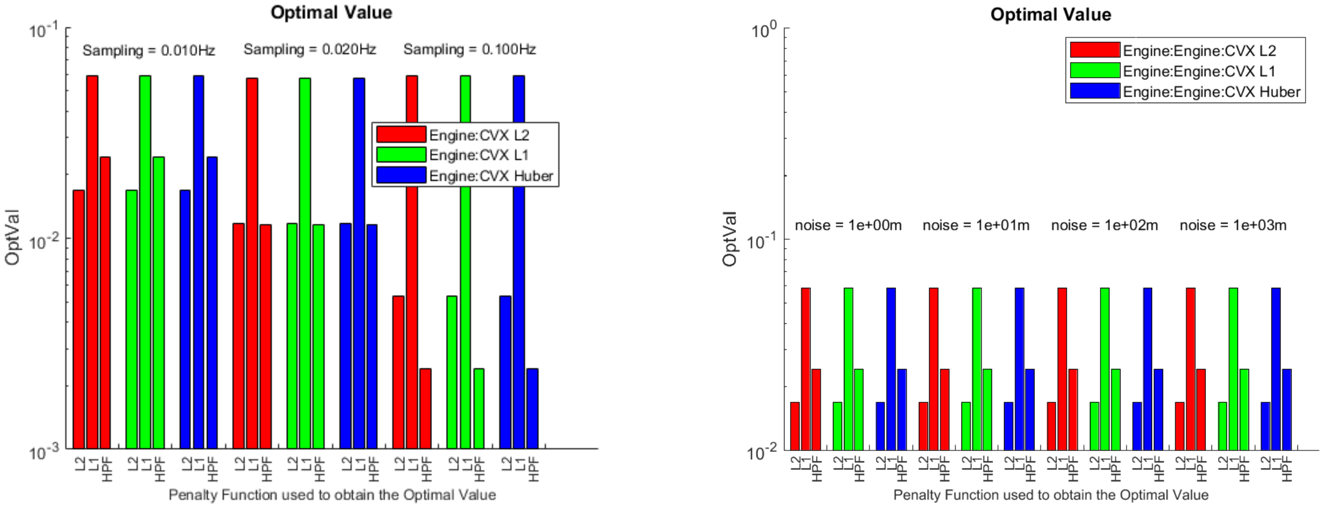


Figure 3.18 – Results for optimal values with scaling factors, including Pf.

[6] It is recommended to read that section before this one, to properly understand the reasons for the change. The reader may skip this subsection for now, and come back when mentioned further in the report.

The same phenomena occurs when the model tests are run, in Section 4.3. The convergence of HPF turns out to be notably slower, even if the CVX toolbox does not *fail* this time. The explanation is the same as above.

These facts raise a new consideration about the initially designed S factor, mainly regarding the Pf component. This component was adjusted so that HPF was able to reach fast convergence even for cases where the noise was extremely high, e.g., $\sigma = 10^3\ m$. Now, it has been observed that, when the initial parameters are not well defined, the optimal value for HPF grows too abruptly, causing the CVX toolbox to *fail* when minimising the cost function. To solve that, a reduction is needed. This is strictly opposite to the cases presented in Figure 3.16, where an augmentation in the scaling factor was required.

After considering both case studies, it is concluded that the most relevant scenario is the situation in which the parameters are not accurately known (e.g., they are estimated), and a larger optimal value is generated by the larger gap existing in the first iterations.

On the contrary, the scenarios where the noise magnitude rises up to $10^3\ m$ are not so frequent, and, even if they were, do not cause the CVX toolbox to *fail*, but only to lose sensitivity and converge at a slower pace.

According to this criteria, the S factor is redefined by modifying its Pf component to:

$$Pf = \begin{pmatrix} L2_{norm} \\ L1_{norm} \\ HPF \end{pmatrix} = \begin{pmatrix} 10^0 \\ 10^{-1} \\ 10^0 \end{pmatrix} \tag{3.12}$$

With that in mind, care must be taken if the noise magnitude is thought to be too large, to re-adjust the scaling factors to more accurate values. Thus, on top of that definition, a temporary correction factor is included for the cases where the noise magnitude is above $\sigma = 10^3\ m$, increasing the scaling for HPF by a factor 10, to improve its sensitivity.

3.4 Error Estimation

An aspect worth investigating is the error estimators available for each method. When out of the testing-shell, the “true” trajectory is not known and the observations are the only reference. From this information, it is not possible to calculate the actual error in the POD process for the estimated value of the parameters involved. However, if the expected accuracy of the measurements is known, statistical information can be obtained about the expected accuracy of the estimated parameters.

In this section, error estimation techniques will be tested to assess whether its use can be considered within the developed POD algorithm. The methodology in these tests will be to obtain a first-estimate of the error using observations with outliers and after the latter are processed, assess the expected error again. For each step, the results will be shown in terms of error with respect to the reference trajectory because real values are needed to validate an error estimator.

HPF will not be discussed, as it is a combination of both norms.

3.4.1 L2-Norm

As introduced in Section 2.5.1, the LSQ method allows for the use of an error estimation parameter referred to as formal error. This formal error, whose mathematical definition is given in Equation (2.52), depends on the covariance matrix of the observations, i.e., the expected accuracies in the measurements; on the model, and, indirectly, on the number of samples available.

In the developed algorithm, and as introduced in Section 3.3, the covariance matrix is initialised under the assumption of perfect awareness of the accuracy levels of the technology taking the measurements. Due to this, it is possible to apply the equation for the formal error with different noise magnitudes and test if the final errors still lie within this estimated range.

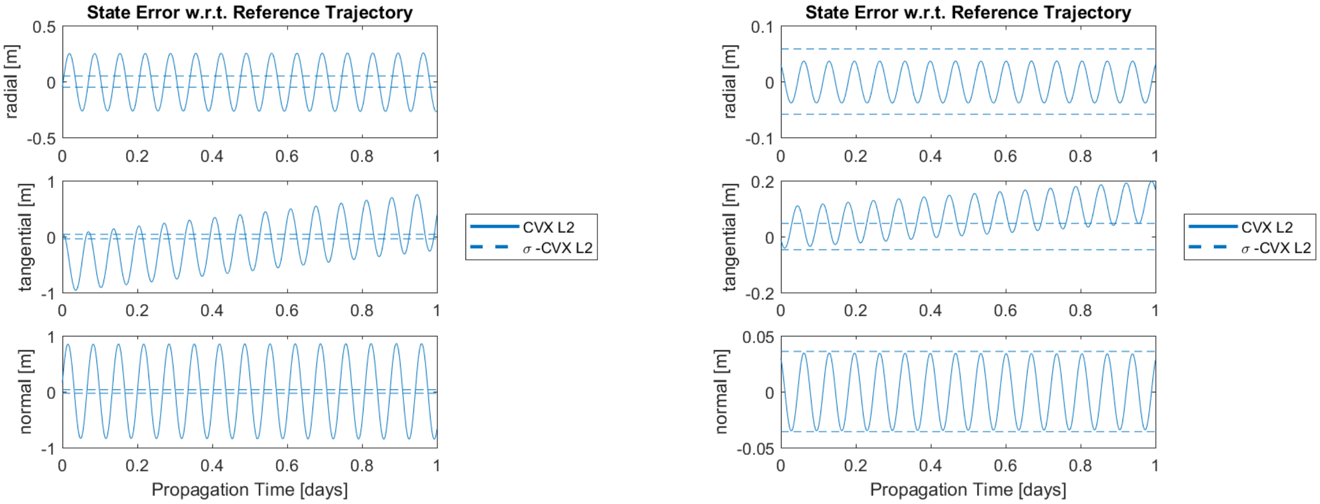


Figure 3.19 –Estimated trajectory for the L2-Norm penalty function with (right) and without (left) outleir processing.

The results are shown in Figure 3.19, where non-processed (left), i.e., observations including outliers, and processed (right) observations are used. While the processed measurements yield excellent results and error predictions that represent the actual errors (considering 1σ), the non-processed case presents errors that are considerably larger than their estimations. This is due to the fact that, for an error estimation to be accurate, the statistical information that is given as input has to be trust-worthy.

When the observation data contains outliers, the estimation of the error in the observation (included in the observation´s covariance matrix) is not as truthful as it should be. However, the algorithm has no information about this fact, and calculates the formal error considering the measurements to be perfectly accurate and, therefore, underestimating the final error.

Even though it is not visible in Figure 3.19, the value of the formal error increases after the measurements are processed. This implies that the predicted error increases after the outlying measurements are removed from the data. Although this could sound illogical at first glance, it is perfectly explained by the fact that when measurements are removed, the number of samples decreases, and, therefore, and based on the inversely proportional relation between the standard deviation of a distribution and the number of samples in the latter, the statistical information becomes poorer, yielding an increase in the expected error.

3.4.2 L1-Norm

Unlike the L2-Norm, the L1-Norm has no established error estimation method. In (Branham, 1986), an attempt is made to obtain a suitable solution for this problem, but the conclusions extracted are rather vague and offer no clear solution other than using the L2-Norm error estimation to raise the robustness of the analysis.

When the outliers are processed, the L1-Norm usually presents higher errors (w.r.t. the reference trajectory) than the L2-Norm. However, these are not so much higher under normal scenario conditions. Due to this, no specific L1-Norm error estimator was implemented and the formal error (from the LSQ method) would be used in case an estimator for the L1-Norm was needed.

3.5 Errors in the Observations

Whenever a measurement is taken, there is an inherent error that distorts the actual values of the desired parameter. This is actually the main reason why POD algorithms are extremely important for data processing utilities or flight dynamic problems. In this section, the different errors that are usually present in sets of measurements are studied and the effect that they provoke in the final results is discussed. Furthermore, the outlier detection functions implemented within the POD method itself are put to test and the behaviour of different penalty functions is analysed.

The tests run in this section will be compared to the reference trajectory computed from the scenario conditions shown in Table 2.3. This methodology applies as well for the results after the outliers are processed. The reason for that is that this section of tests has no interest in assessing the effectiveness of the algorithm but only in understanding how much different errors in the measurements distort reality and how much of this distortion can be avoided by processing the measurement errors, both through the POD algorithm and the outlier processing functions.

3.5.1 Outliers

In Section 2.7.1, it was mentioned that, for the processing of outlying measurements, a first estimation of the trajectory was needed. Of course, this first estimation of the trajectory depends on the penalty function to be optimised and, therefore, different methods will yield different results. Here, tests will be run to investigate how different cost functions generate a first estimation of the trajectory.

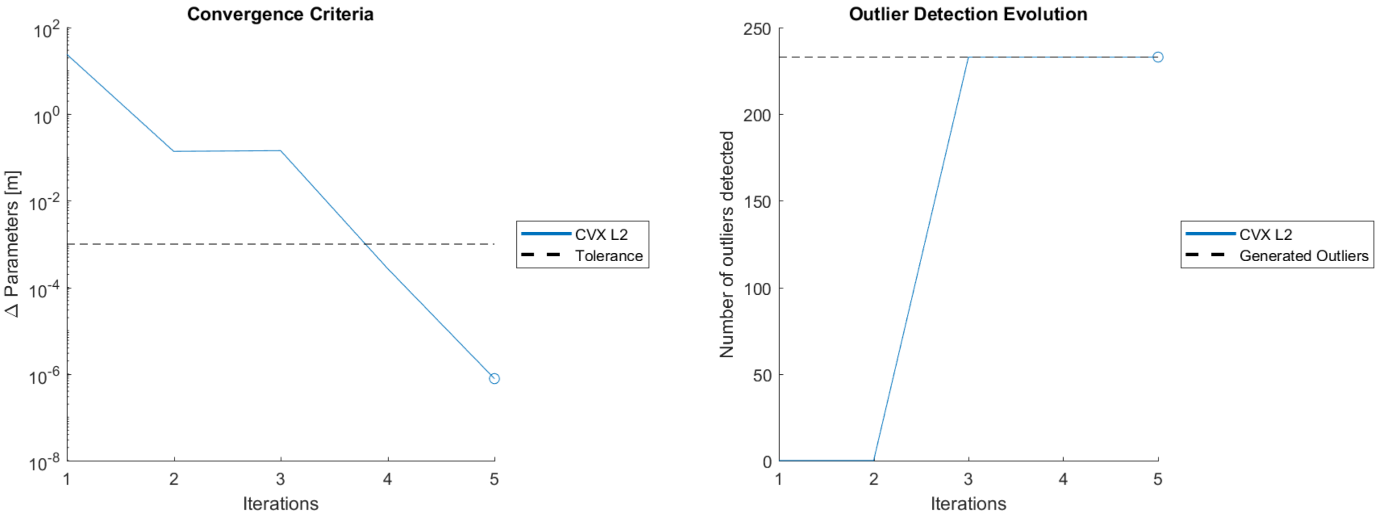


Figure 3.20 - Example of the POD results.

In the developed algorithm, the observations are processed at each iteration. To process the noise in the measurements, an initial estimation is computed and given as input to the outlier detection function, which identifies the outlying measurements and flags them as such. When the initial estimation is not accurate enough, the processing function will not be able to perform adequately, but, as iterations go on, this accuracy is improved and the number of identified outliers converges.

If the outliers found have not changed from the previous iteration and the improvement of the solution is below the required tolerance, an extra iteration is performed to ensure the results are stable. Else, the same iterative process is repeated until the processing function establishes the definitive number of outliers. This is exemplified in Figure 3.20, where convergence was found after five (four plus the final check) iterations.

3.5.1.1 Trajectory Estimation without Outlier Detection

First of all, the results obtained without processing outlying measurements will be discussed here, for each of the penalty functions. This is meant to give an idea about how accurate the first estimation is, and how dependent on that the outlier detection function is. The results for the nominal noise are shown in Figure 3.21, as a reference plot.

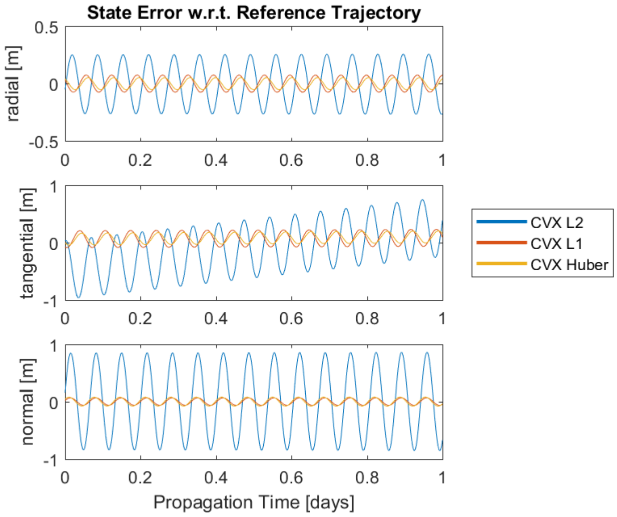


Figure 3.21 - First estimation w.r.t. the reference trajectory for the nominal noise.

It is clear that the LSQ method suffers much more from the existence of outliers, as it was expected. HPF, however, follows the L1-Norm solution and keeps the errors low.

3.5.1.1.1 Magnitude of the Stochastic Error

If the same test is repeated for noises one and two orders of magnitude larger, the same results are obtained, only scaled by a factor 10 and 100 respectively (see Figure 3.22).

3.5.1.1.2 Number of Outliers.

By varying the number of outliers from 233 to 413 and 630 (corresponding to probabilities of 10% and 20% respectively[7]), the results shown in Figure 3.23 are obtained.

As expected, the L1-Norm is able to handle non-detected outliers in a much more accurate way than the L2-Norm. Similarly, HPF mimics again the behaviour of the L1-Norm, which has much more relevance in this case, due to the presence of outliers.

[7] Recall that, as it was introduced in Subsection 2.7.1.4, the generation of an outlying measurement is based on a threshold probability, nominally set to 5%. This probability is evaluated for each of the six components of the state, and, therefore, the 233 outliers that are nominally introduced are not the 5% of the observations, but of the totality of the state components, i.e., six times the observations.

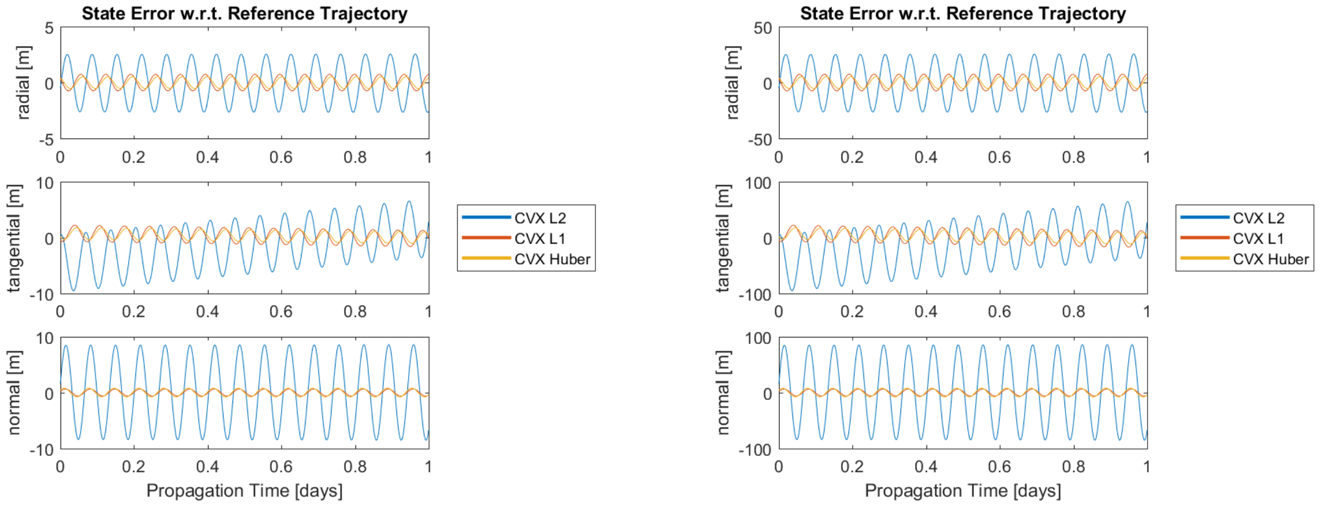


Figure 3.22 - First estimation w.r.t. the reference trajectory for different magnitudes for the stochastic error. Left, $\sigma = 10$; Right, $\sigma = 100$.

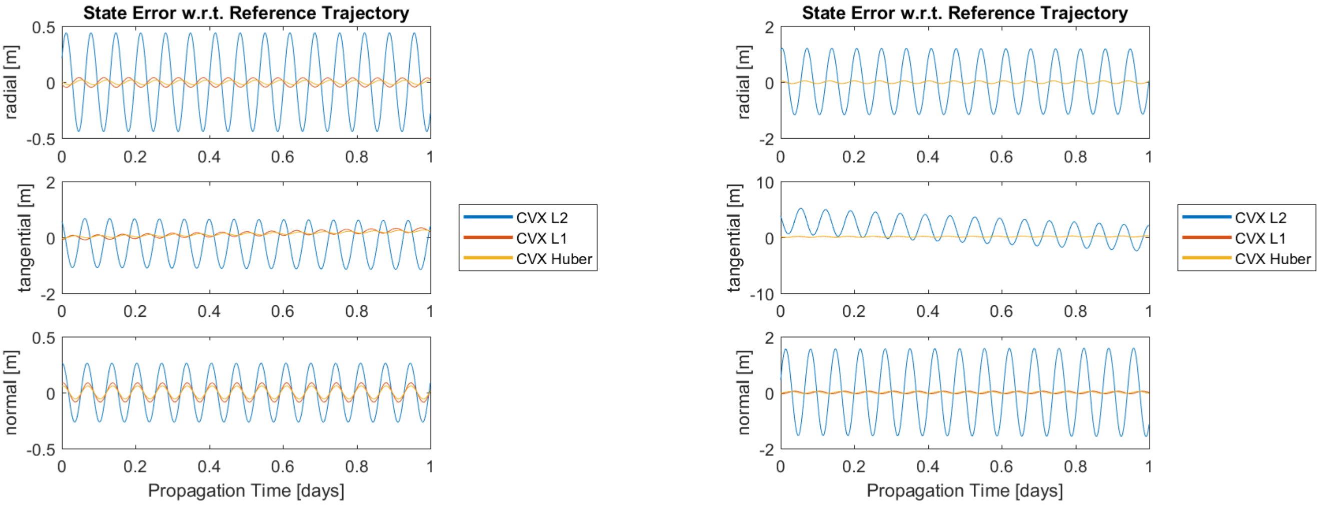


Figure 3.23 - First estimation w.r.t. the reference trajectory for different number of outliers. Left, 413 Outliers; Right, 630 Outliers.

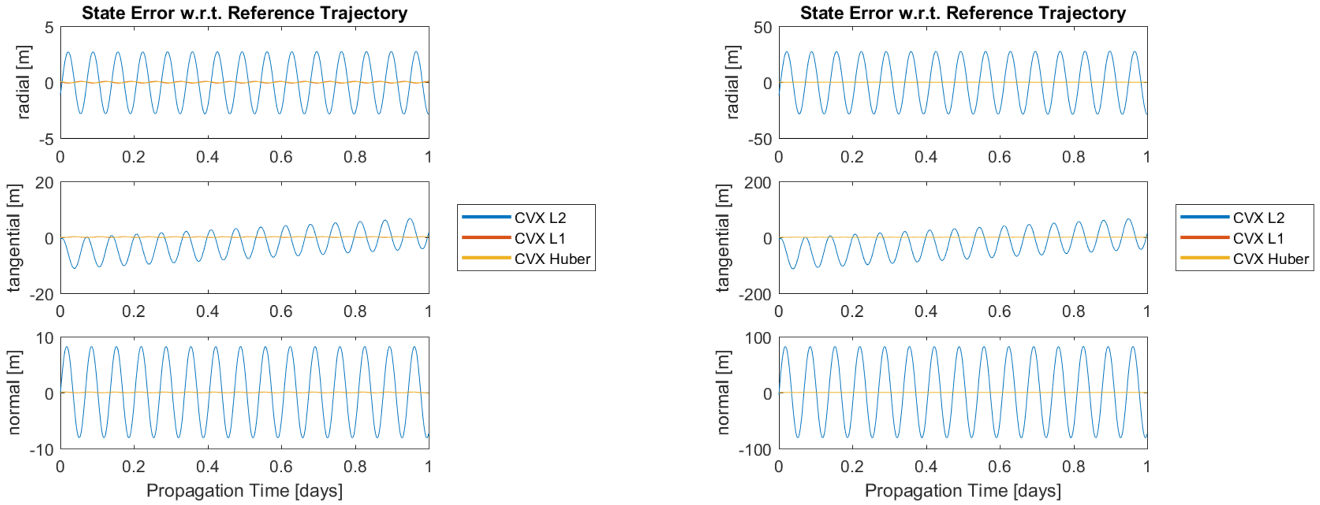


Figure 3.24 - First estimation w.r.t. the reference trajectory for different outlier magnitudes. Left, 10-1.000; Right, 10-10.000.

3.5.1.1.3 Magnitude of the Outliers

Varying the magnitude of the outliers yields the results shown in Figure 3.24.

The L2-Norm penalty function is greatly perturbed by the presence of outliers and by its magnitude. The L1-Norm and HPF remain more accurate.

3.5.1.2 Trajectory Estimation with Outlier Detection

After the first estimations of the trajectory are calculated, they are given to the outlier detection functions along with the measurements, so that outliers could be detected. To assess this detection, two criteria will considered: the number of detected outliers (at the end of the process), and the accuracy of the final solution. Outliers in both position and velocity measurements will be considered for these tests.

The nominal case yields the results shown in Figure 3.25, where all outliers are detected. This case, where it is possible to already see the huge improvement in the results coming from the L2-Norm when outliers are processed, shall be used as a reference solution.

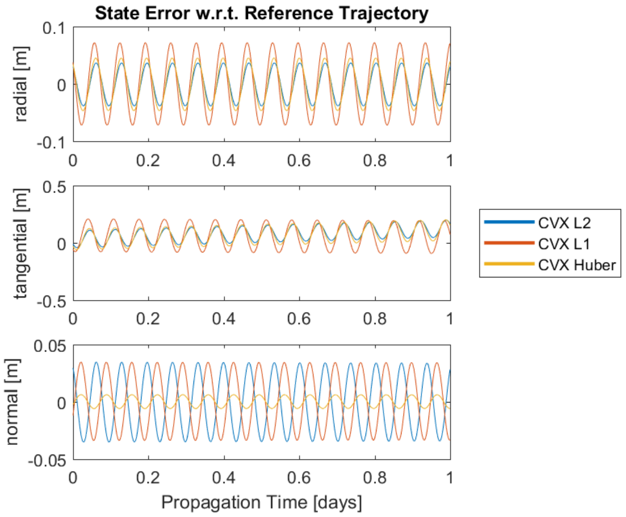


Figure 3.25 - Estimated trajectory w.r.t. the reference trajectory for the nominal noise.

3.5.1.2.1 Magnitude of the Gaussian Noise

In Table 3.3, it can be observed that every outlier is detected in each of the variations. After the outliers are detected, the trajectory is processed and the final results shown in Figure 3.26.

Table 3.3 - Detected outliers for different noise magnitudes.

	L2-Norm	L1-Norm	Huber's Function	Total Number
Detected with $\boldsymbol{noise = 1\,m\ /\ 1\frac{\mathrm{mm}}{s}}$	233	233	233	233
Detected with $\boldsymbol{noise = 10\,m\ /\ 10\frac{\mathrm{mm}}{s}}$	233	233	233	
Detected with $\boldsymbol{noise = 100\,m\ /\ 100\frac{\mathrm{mm}}{s}}$	233	233	233	

For the two variations of the noise layer, the final errors are exactly multiplied by a factor 10 and 100, respectively.

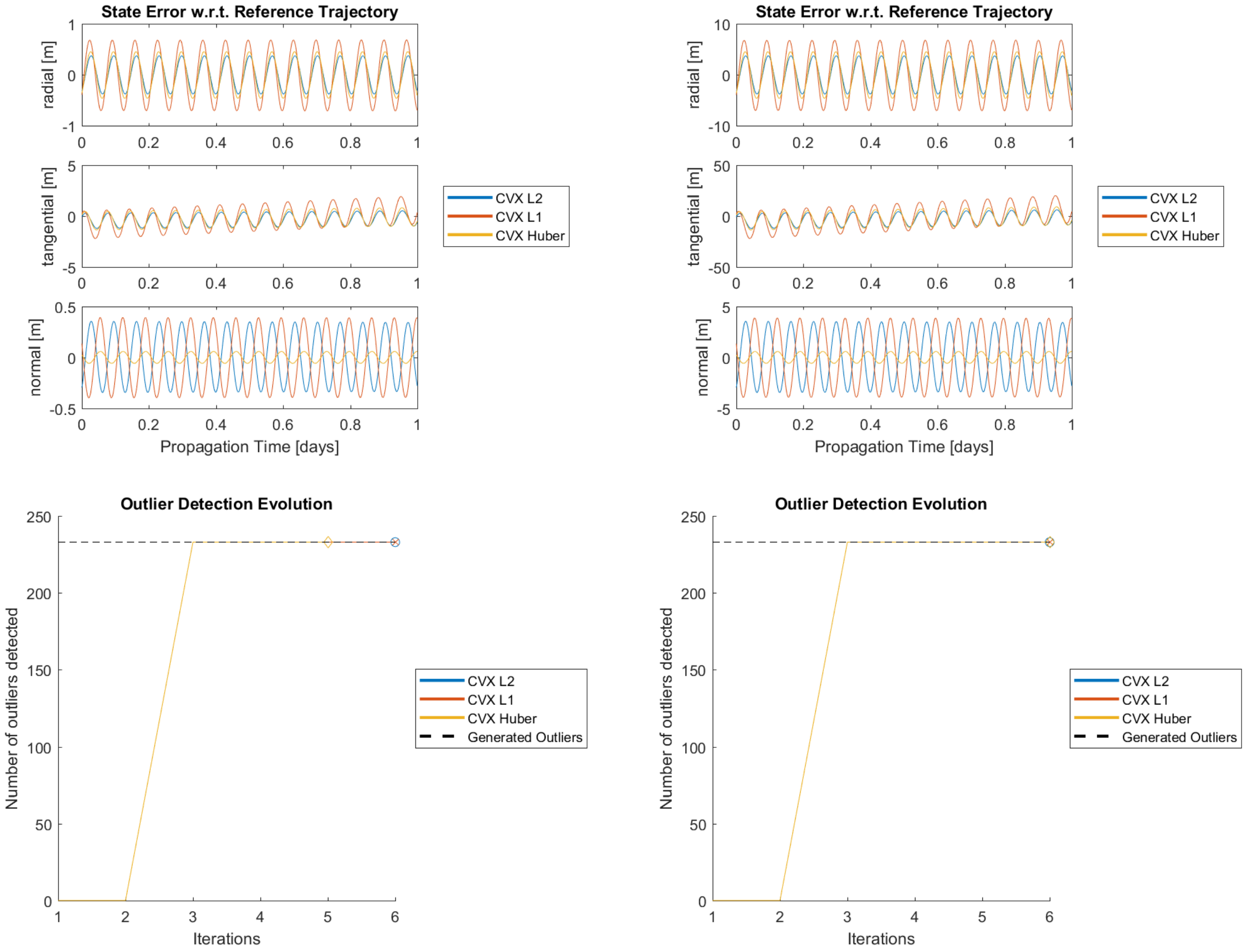


Figure 3.26 - Results for different magnitudes for the stochastic error. Left, $\sigma = 10$; Right, $\sigma = 100$.

3.5.1.2.2 Number of Outliers

In Table 3.4 the outlier detection performances are shown. Thanks to the iterative outlier processing method implemented in this algorithm, all outliers are detected. Trajectory errors are shown in Figure 3.27, where it can be observed that, unlike it was shown in Figure 3.23, the L1-Norm shows slightly worse accuracy levels, now that the outliers have been processed.

Table 3.4 - Detected outliers for different number of outliers.

	L2- Norm	L1- Norm	Huber's Function
Total outliers = 233	233	233	233
Total outliers = 413	413	413	413
Total outliers = 630	630	630	630

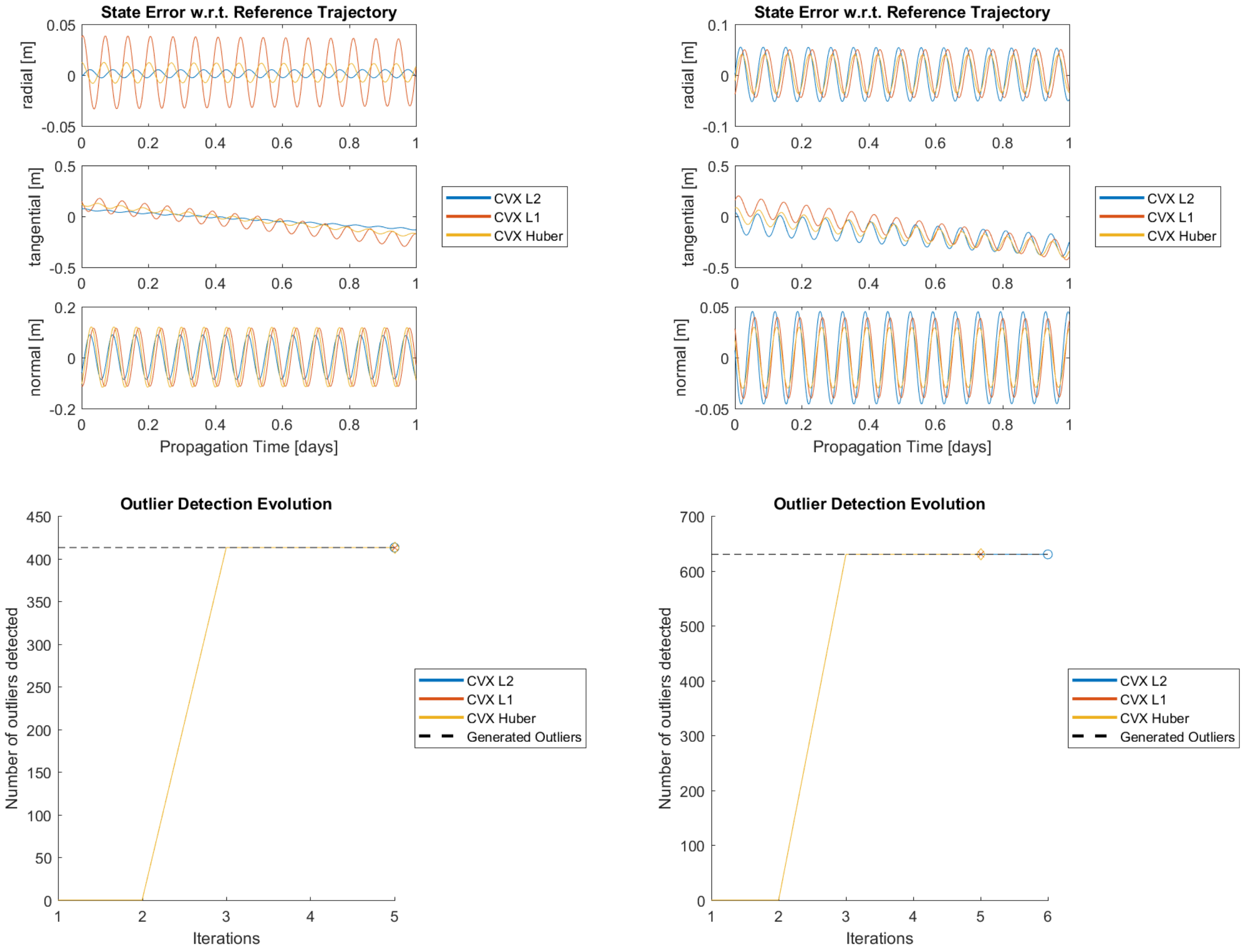


Figure 3.27 - Results for different number of outliers.
Left, 413 outliers; Right, 630 outliers.

3.5.1.2.3 Magnitude of the Outliers

Lastly, the magnitude of the outliers was also altered. The nominal noise includes outliers between $10 - 100$ times larger than the standard deviation of the stochastic noise. For these tests, this maximum factor was augmented by factors 10 and 100, i.e., outliers between $10 - 1.000$ and $10 - 10.000$ respectively. Additionally, two cases where the magnitude of the outliers is barely larger than the stochastic noise are provided, to test whether the outlier detector is able to also handle unclear outliers. These tests will be divided into outliers being $2 - 4$ and $4 - 10$ times the magnitude of the standard deviation of the stochastic noise.

In Table 3.5, it is shown that every outlier is detected for the first three cases, and a number of outliers are missed when their magnitudes are much lower. However, the results for these last cases (see the bottom plots of Figure 3.28) present similar accuracy. This is due to the fact that the non-detected outliers are small enough to be considered stochastic noise, even if the detector fails to locate them. Also, a slightly better performance is observed when the L1-Norm is used (even better for HPF) w.r.t. the L2-Norm case, where three (and one) additional outliers are detected. This explains why in the bottom-right plot of Figure 3.28, the L2-Norm state error gets closer to the less accurate L1-Norm state error. The outlier detection evolution is shown in Figure 3.29.

Table 3.5 - Detected outliers for different outlier magnitudes.

	L2- Norm	L1- Norm	Huber's Function	Total Number
Outlier Factor: $10 - 100$	233	233	233	233
Outlier Factor: $10 - 1.000$	233	233	233	
Outlier Factor: $10 - 10.000$	233	233	233	
Outlier Factor: $2 - 4$	6	6	6	
Outlier Factor: $4 - 10$	191	194	192	

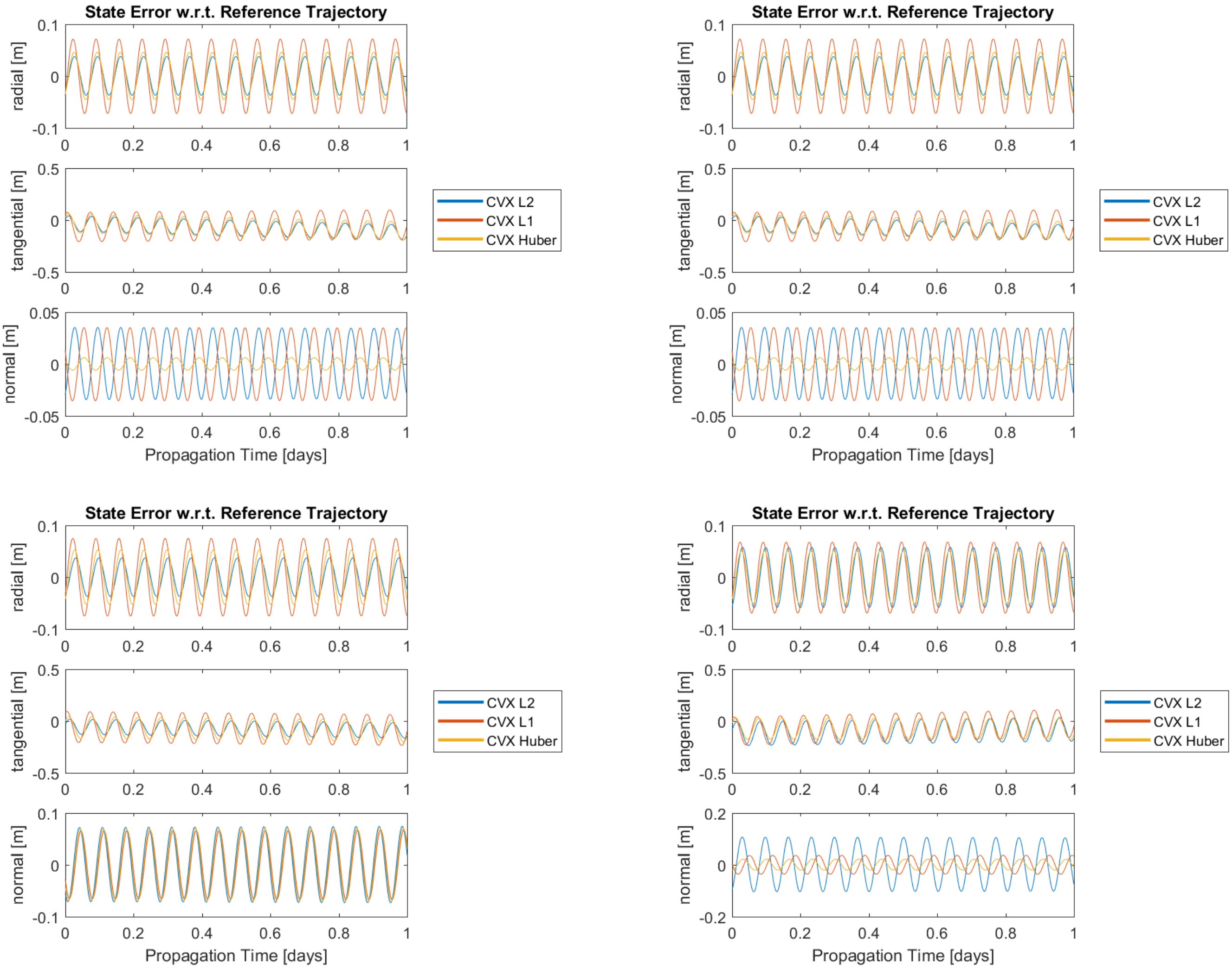


Figure 3.28 - Estimated trajectory w.r.t. the reference trajectory for different outlier magnitudes. Upper-Left, 10-1.000; Upper-Right, 10-10.000; Bottom-Left, 2-4; Bottom-Right, 4-10.

3.5.2 Data Sparsity (Gaps)

Sometimes, and depending on the source of the observations, the measurement data present gaps where no valid information can be extracted. The most frequent reason for that is loss of tracking capabilities, e.g., when using range-angle measurements from only one ground station, most of the orbit of the satellite is not measured.

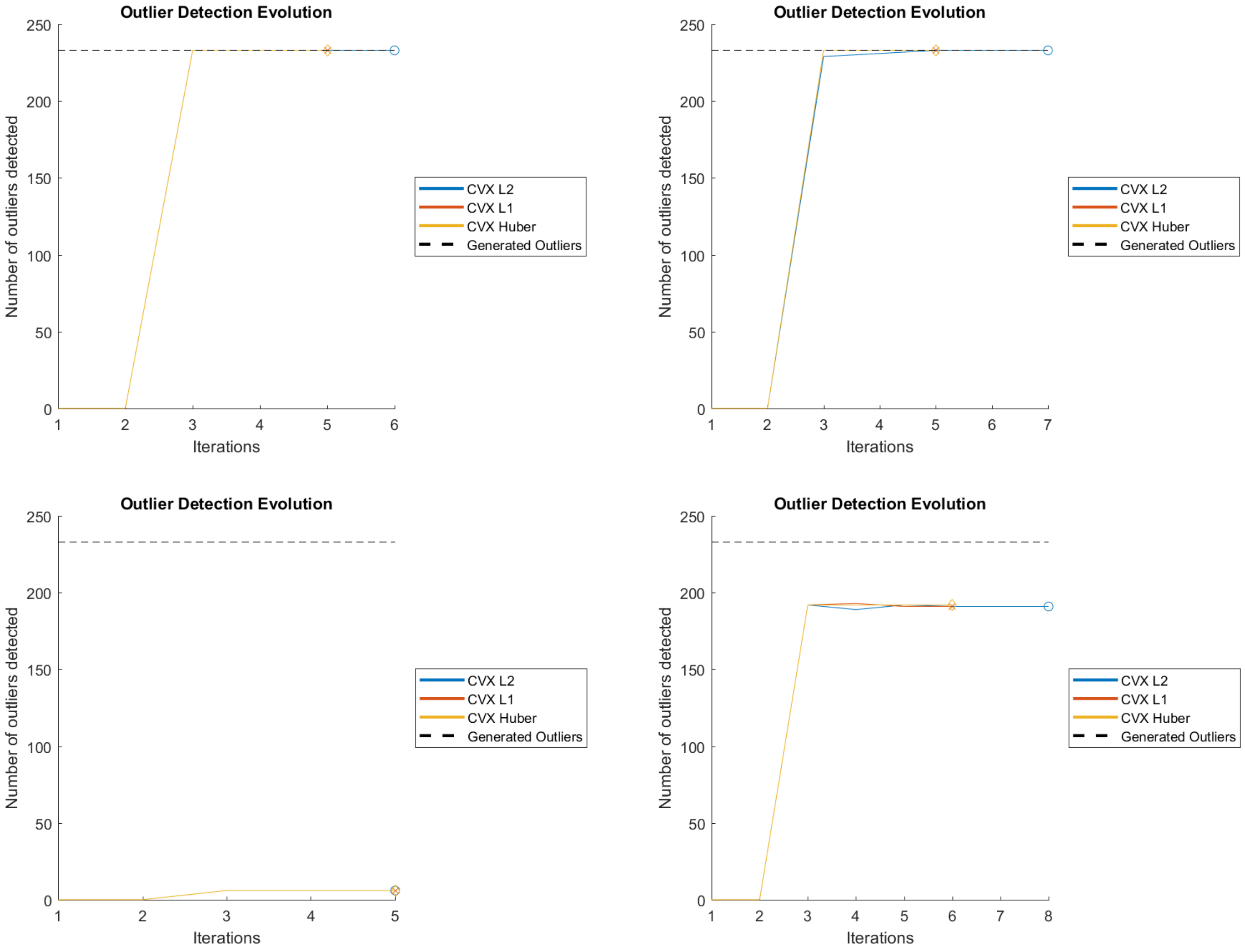


Figure 3.29 - Outlier detection evolution for different outlier magnitudes.
Upper-Left, 10-1.000; Upper-Right, 10-10.000; Bottom-Left, 2-4; Bottom-Right, 4-10.

After running the tests, no visible differences were observed for each of the penalty functions. The most relevant factor was found to be the moment in which these gaps were introduced. Thus, these tests were run using only HPF as penalty function, to serve as a further validation of the capabilities that this penalty function presents. The gaps will be set at the beginning, in between (after around a third of the observations), and at the end of the data set. The duration of the gaps will also vary from 10% to 40% of the observations.

From the results shown in Figure 3.30, it can be concluded that the existence of a gap affects negatively the orbit estimation. Moreover, the epoch in which the gap takes place is clearly relevant. Gaps in the middle of the data will yield smaller errors in the final estimation, while a gap in the beginning or ending result in larger errors in the estimation.

When compared with the gap-free case (upper-left plot), the magnitude of this increment in the error, it is relatively low for every case and does not strongly affect the final results when compared to the full measurement data case.

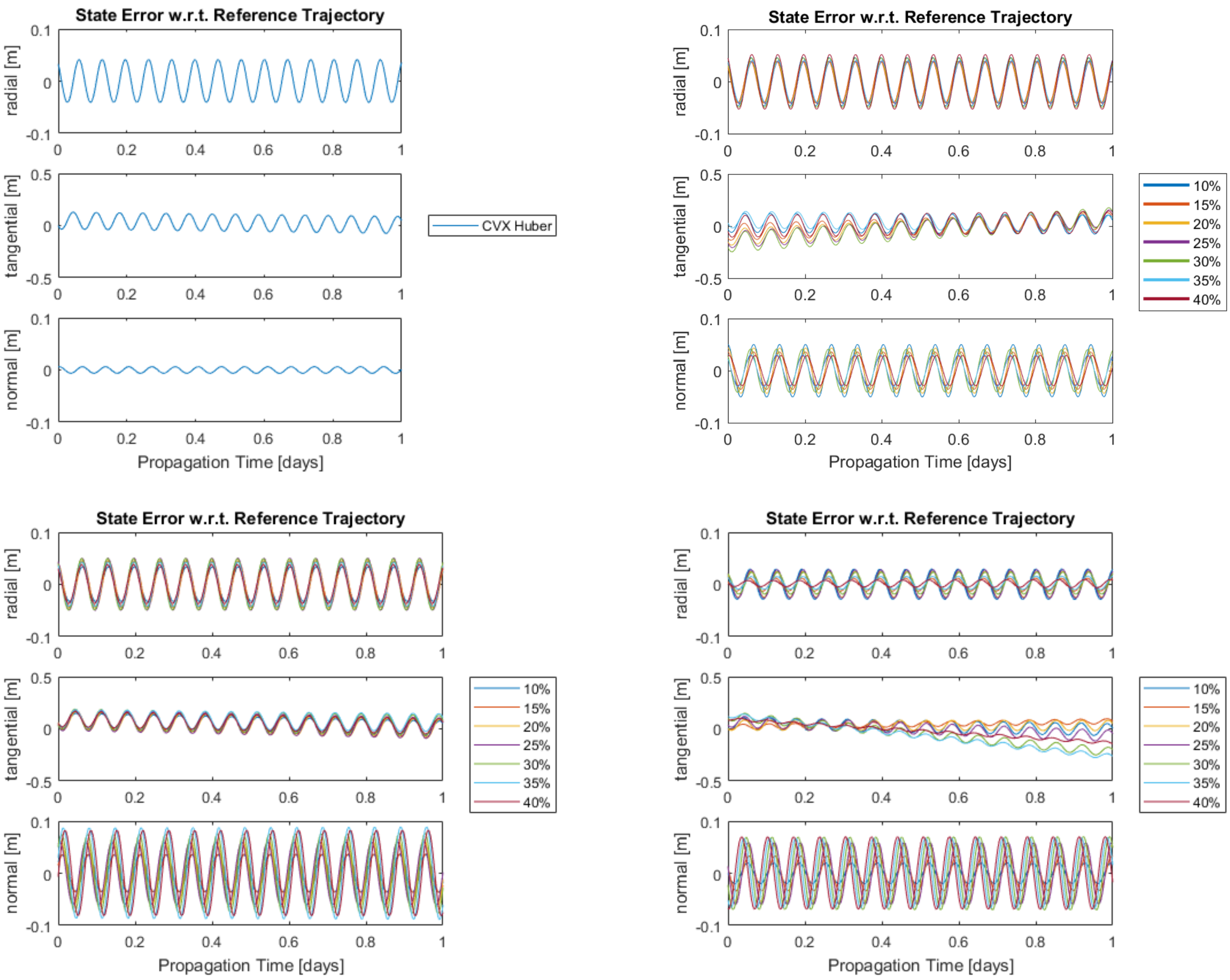


Figure 3.30 – Estimated trajectory w.r.t. the reference trajectory for different gap configurations. Upper-left, no gap; Upper-right, early gap; Bottom-left, mid-gap; Bottom-right, late gap.

A more interesting case will be contemplated in Section 4.4, where angle measurements will be integrated in the algorithm, giving place to, not only gaps, but measurements that define the same partial arc of the trajectory.

3.5.3 Random vs Normal Noise

The way in which the low amplitude noise (stochastic error) is explained in Subsection 2.7.1.1. For modelling purposes, and as explained in Section 2.7.1, the nominal noise layer is generated using a random number generator based on the Normal distribution $\sim N(0, \sigma)$, where σ is the standard deviation, nominally set to 1 m and 1 mm/s for position and velocity measurements, respectively.

On the other hand, the noise could also be generated using a random number generator that creates uniformly distributed numbers within a selected interval, which would be$\pm\sigma\, m$ and $\pm\sigma\, mm/s$.

To further investigate the differences between both, a comparison was generated using nominal and augmented values for σ. It is also relevant to remark that, even if the seed for the random generators remains constant, the small variation in the code, needed to generate a different kind of stochastic error, resulted in a different distribution of the errors in time, as well as in a new number and location of outliers.

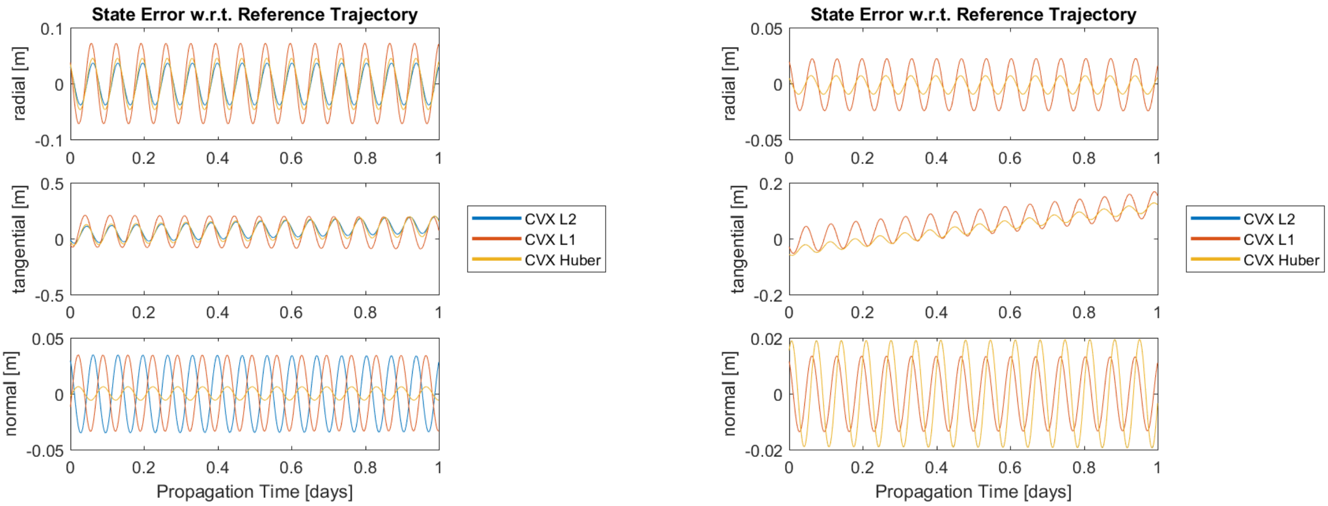


Figure 3.31 - Estimated trajectory w.r.t. the reference trajectory for different stochastic distributions. Left, $N(0,1)$; Right, rand $[-1,1]$.

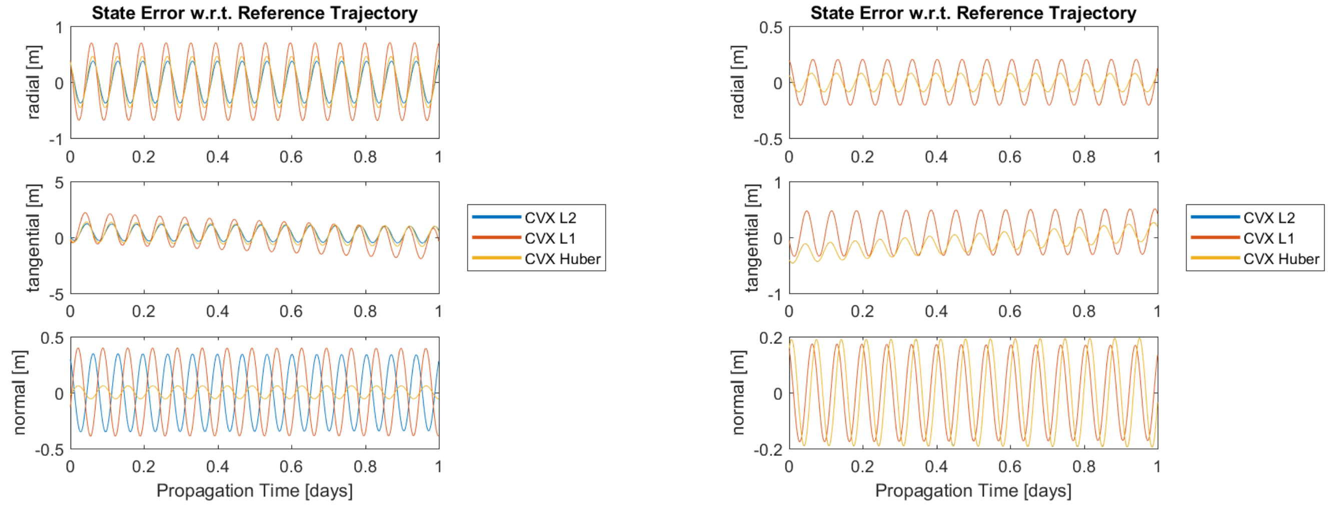


Figure 3.32 - Final solution for the outlier-free measurements w.r.t. the true trajectory for different stochastic error generators. Left, $N(0,10)$; Right, rand $[-10,10]$.

Results are shown in Figure 3.31 for $\sigma = 1$ and in Figure 3.32 for $\sigma = 10$. Both plots reflect similar behaviours only scaled by the value that σ takes for each case. It is clear that the accuracy of the solution for the Gaussian distribution is (approximately around three times) worse. This is due to the normal distribution not being limited to its standard deviation value, σ, which only means that approximately 68% of the values will be within this range. Logically, larger errors yield lower accuracies.

From a practical perspective, the differences in accuracy are only related to the final magnitude of the generated noise layer. However, what can be observed in these plots is the fact that the L2-Norm deals better than the L1-Norm with the normally distributed errors, and that HPF does even better than the L2-Norm. This goes in the same line of reasoning that explains why the L1-Norm is able to deal better with outliers, i.e., distributions with large tails will be better managed by the L1- Norm, which is not so affected by higher cost function values. Therefore, for a uniformly distributed stochastic error, the L1- Norm offers better results than the LSQ method (even when the outliers are successfully processed), and the other way around when the distribution has thinner tails.

3.6 Huber's M Parameter

As introduced in Section 2.5.3, HPF uses a threshold parameter, M, to define the limits where the functions changes its behaviour from L2-Norm to L1-Norm. Depending on where this threshold is established, the function acquires different properties, more similar to one or another penalty function. In this section, an investigation is carried out, trying to establish how varying this parameter affects the results of the POD algorithm, and whether there is some sort of proportional relation between M and how close HPF gets to the L2- or the L1-Norm.

For this sake, the mean Euclidean distance between the reference trajectory and the estimated trajectory was selected as the factor in charge of representing the results, because it represents well the general results and allows for a single-value-based evaluation. As it will be shown in Figure 3.33, it was observed that a variation of M does not reflect a linear evolution of HPF to each of the two norm functions. Actually, its variation follows more of a logarithmic proportionality.

This incremental method was applied for a range of $M = (0, 1]$. To represent clearly the transition, eighteen values were logarithmically distributed in this interval. Additionally, the results for the L1- and L2-Norms were included and represented in Figure 3.33; resulting in a total of twenty case studies. A colour gradient was used to express the proximity of the parameter to each function.

The results showed that the behaviour of the algorithm resembles the LSQ method already when $M \cong 10^{-2}$. Also interesting is the fact that in the first cases, where HPF is more influenced by the L1-Norm, some instability appears in the evolution of the mean Euclidean distance. This was found to be mainly related to the random generation of the noise and the statistical nature of the POD method. Nevertheless, a trend can still be observed that represents a rather abrupt transition between both norms. This transition point was checked to be the moment in which M equals the standard deviation of the stochastic error introduced, once accounted for the scaling factor used.

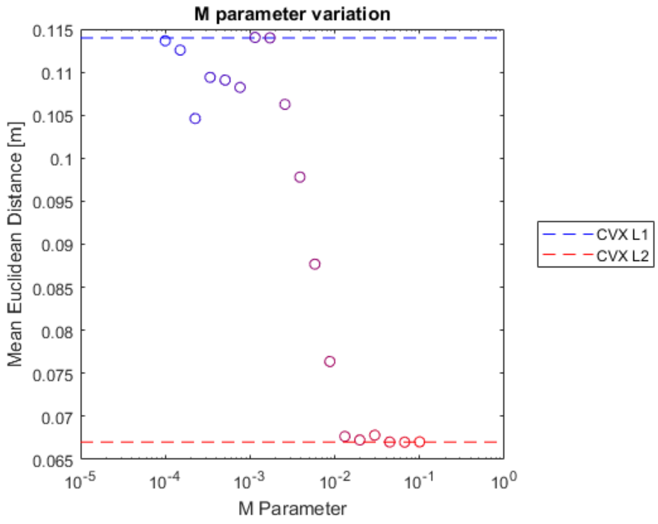


Figure 3.33 - Final estimation of the trajectory w.r.t. the reference trajectory for M = (0, 1].

This agrees with what was introduced previously in Section 2.5.3 and serves as selecting criteria for an adaptive M algorithm that will be tested in 3.6.1, based on the iterative calculation and assignment of the standard deviation of the observation residuals to M.

It could be expected that the optimal values for HPF cases studied here showed a progression from the L1-Norm optimal value towards the L2-Norm value. However, this is not the case, because the definitions of the functions are not numerically equal, but only conceptually, i.e., the behaviour is reflected, but the magnitudes of the optimal values do not approach each of the norms when M is closer to its limits. They do, nonetheless, decrease hugely when M is shifted towards zero.

3.6.1 Adaptive M Parameter

As it was already introduced in Section 2.5.3, the M parameter represents the threshold in which HPF changes its behaviour (see Equation (2.56)). A threshold set to M means that values larger than M times the standard deviation of the data will be treated following the L1-Norm, while the smaller ones will follow the L2-Norm.

This concept was graphically proved before, and can be also very easily explained from a mathematical point of view. From (2.18), an expression for the misfit to be minimised can be written as

$$\rho = Hx - y \tag{3.13}$$

whose components were introduced in Section 2.3. As convergence is approached by the batch algorithm, the term x (accounting for the increment in the parameters) tends to 0, and the first term of the expression will cancel.

$$x \sim 0 \rightarrow \rho \cong y$$

If the outliers were processed correctly from the observations, the standard deviation of y (representing the residual between the measurements and the estimated observations) will be the standard deviation of the Gaussian noise of the observations.

$$M = \sigma(\rho) \cong \sigma(y) \tag{3.14}$$

This concept, using Equation (3.14) after each iteration of the batch algorithm, will allow for the computation of an adaptive M that will, a priori, help HPF vary its thresholds to fit the best the measured data.

A comparison of both methods is shown in Figure 3.34. It can be observed that the accuracy remains practically the same but the convergence of the POD method takes place at a slower pace. In this set of tests, the initial M value for the fixed HPF is now set to the estimation of the noise, i.e., the expected standard deviation of the stochastic noise coming from the measuring technique (see Section 2.7.1).

This fact, of course, causes the fixed M algorithm to converge faster because the value of M needs not to be estimated throughout iterations, but is already known. Therefore, a secondary set of tests was carried out, where M was over and underestimated by a factor 10, and different stochastic noise magnitudes were implemented for each case. Also, the condition $M \geq 1$ was imposed to the algorithm after some testing, benefiting the convergence capacities of the POD[8].

[8] The value that HPF receives, nonetheless, is subject to the scaling factor used, so the scaled value of M may be smaller than 1.

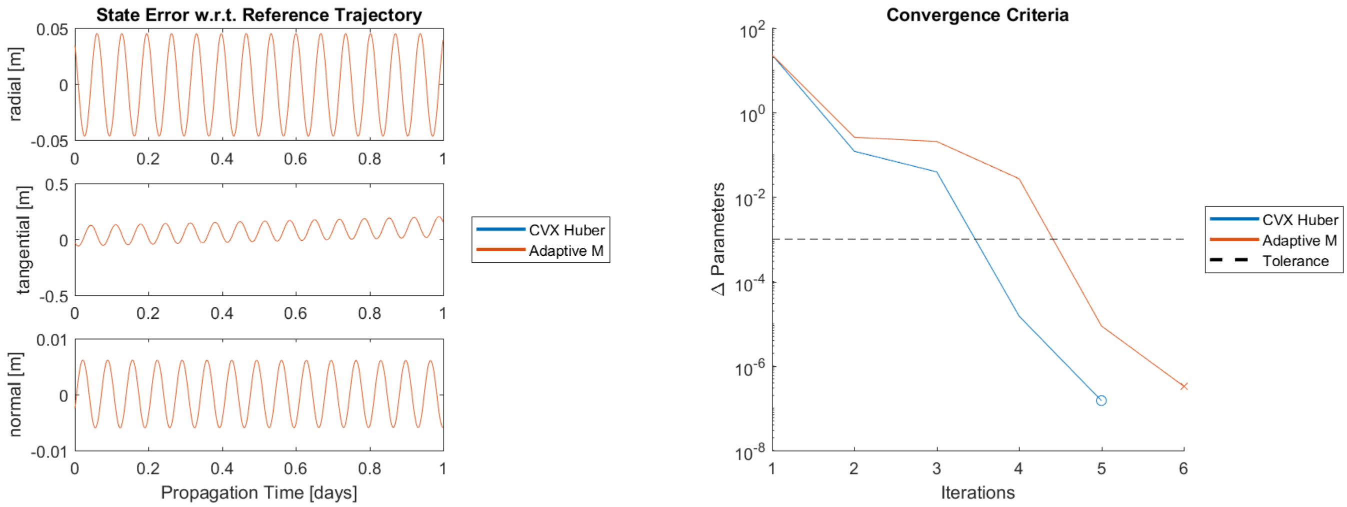


Figure 3.34 – Adaptive M algorithm comparison.

In Figure 3.35, the results for the case where $\sigma = 10^3\ m$ are shown. The rest of the stochastic noise magnitudes tested, yield exact results only scaled by a factor σ, so they have not been included.

As represented in Figure 3.34, adaptive HPF takes always one extra iteration to find convergence, except in the case where $\sigma = 10^3\ m$ and $M = 10\sigma$, where the adaptive algorithm outperforms the fixed one largely.

In terms of accuracy levels, the adaptive algorithm performs better when the noise is underestimated, and both stay levelled when overestimated, in the overall error (component-wise, there are differences).

Summing up, there is no clearly superior method and the selection of one or another depends entirely on the confidence on how certain the noise conditions are. The drawback for selecting an adaptive method will be the extra iteration that this method usually needs.

This method will be included for some of the next tests, so that more insight can be gained on how this affects the results of the POD.

3.7 Conclusions

In this chapter, the performances and capabilities of the developed algorithm have been tested for computational aspects.

The validation processes for the implementation of the manoeuvres, the STM, and the SM have yielded excellent results. Even if some interesting remarks were raised, the maturity of the tools is perfectly validated and its implementation in the POD algorithm is safe.

The use of scaling factors to accommodate the problem to the tolerances and thresholds of the solver is vital here. For that sake, a scaling factor was defined that allows for a more consistent problem initialisation and, thus, better performances of the overall POD algorithm. Nevertheless, a more robust formulation of the problem might be of use for future investigations.

Error estimation has proven the implementation of the LSQ in this algorithm to be correct. However, for the other two penalty functions, no other (mature) error estimator has been found. For this reason, error estimators will not be considered as main reference for the results obtained in further tests.

It has been shown that the LSQ method reports the best accuracies when the data is outlier-free, and L1-Norm and HPF offer better performances when outliers are present. This constitutes one of the basic results of this research, and confirms L1-Norm and HPF penalty functions to clearly outperform traditional methods when the outliers are not (perfectly) processed. Additionally, HPF has proven itself to be both accurate and robust when needed to, at the price of a less efficient computational effort, offering the capability to be tuned to the required case making use of the M parameter as well.

The outlier detection function, which is implemented every loop in the POD algorithm, has been proved to offer satisfactory results and can be, thus, validated as well.

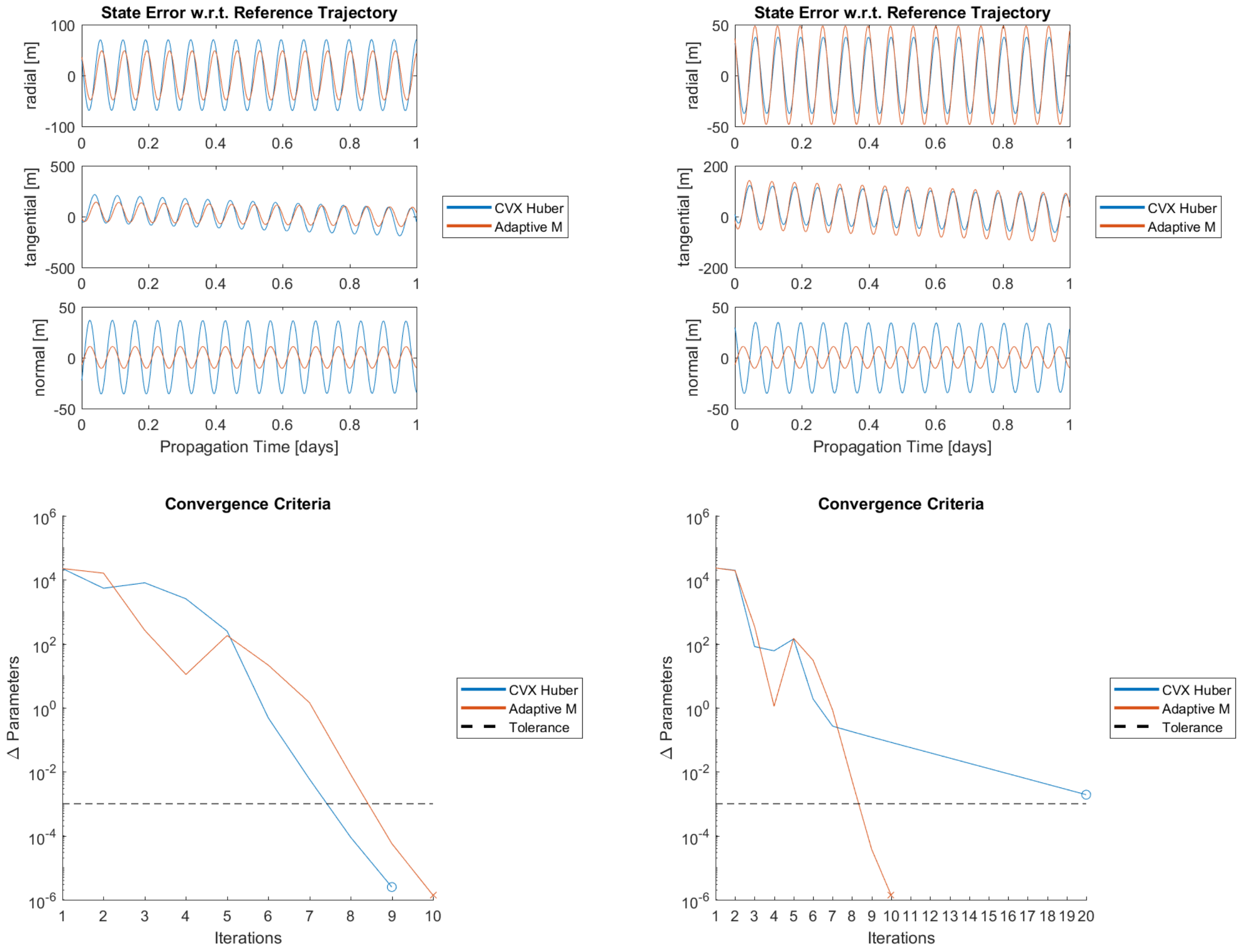


Figure 3.35 – Errors and convergence rates for under- (left) and overestimated (right) M parameter for $\sigma = 10^3\ m$.

4 THE POD CHALLENGE

In this chapter, a different testing approach takes place. The tests included here will be more focused on investigating how do the different penalty functions face real-world situations that may arise for a general POD tool, such as, different scenarios, parameter estimation, modelling errors, or use of AE measurements.

The normal HPF will stablish $M = \sigma$, i.e., 1 for the nominal noise (as opposed to the previously assumed value of 1,35, see Section 2.5.3) while the adaptive version will estimate its value each iteration. This will offer a better comparison between both versions of HPF, since $\sigma = 1$ means a perfect estimation of the stochastic noise magnitude. The LSQ and L1-Norm will remain the same. The scaling parameters defined in Section 3.3 will be applied throughout this chapter.

4.1 ORBITAL CONDITIONS

In some cases, different scenarios offer observations that make the POD process much more difficult. The scenario that has been used up to this point is described in Section 2.8, and many results have been already shown in previous chapters. Here, tests will be run to show how the different penalty functions behave under different scenario settings, namely, Medium Earth Orbit (MEO), GEO, and high-eccentricity orbits, such as the classical GEO Transfer Orbit (GTO); described in Table 4.1. The noise will remain the same for every case, even if it may be unrealistic.

Table 4.1 – Scenario conditions for different study-cases.

	$\boldsymbol{a}\ (\boldsymbol{km})$	$\boldsymbol{e}\ (-)$	$\boldsymbol{i}\ (°)$	$\boldsymbol{\Omega}\ (°)$	$\boldsymbol{\omega}\ (°)$	$\boldsymbol{\theta}\ (°)$	***Reference***
MEO	29599.8	0.0	56.0	197.632	0.0	345.153	Galileo
GEO	42164.1	0.0	0.0	0.0	0.0	83.0	MTG-I
GTO	24396.1	0.728	6.0	0.0	178.0	0.0	Ariane V

After the tests are run, the results are gathered in Figure 4.1, where similar orders of magnitude for the errors are observed. More in detail, the GEO scenario seems to be the most favourable condition for orbit determination, even if the observations correspond only to one orbital revolution. This might be related to the slower dynamics involved in such a trajectory, where no atmospheric drag and much weaker irregular gravity field apply. The GTO scenario presents a more abrupt behaviour near the pericentre of the orbit, as expected. These peaks are, however, the orbital sections where the higher accuracy is reached. This is due to the changing dynamics of this particular trajectory, in which smaller errors near the pericentre will yield larger errors near the apocentre.

Regarding the penalty functions, the L1-Norm presents the lowest accuracy levels, not surprisingly, whereas the L2-Norm and HPF fit the reference trajectory closer. However, the differences are relatively small and it is possible, thus, to conclude that the orbit in which the satellite flies should not affect the selection of the POD method to be used.

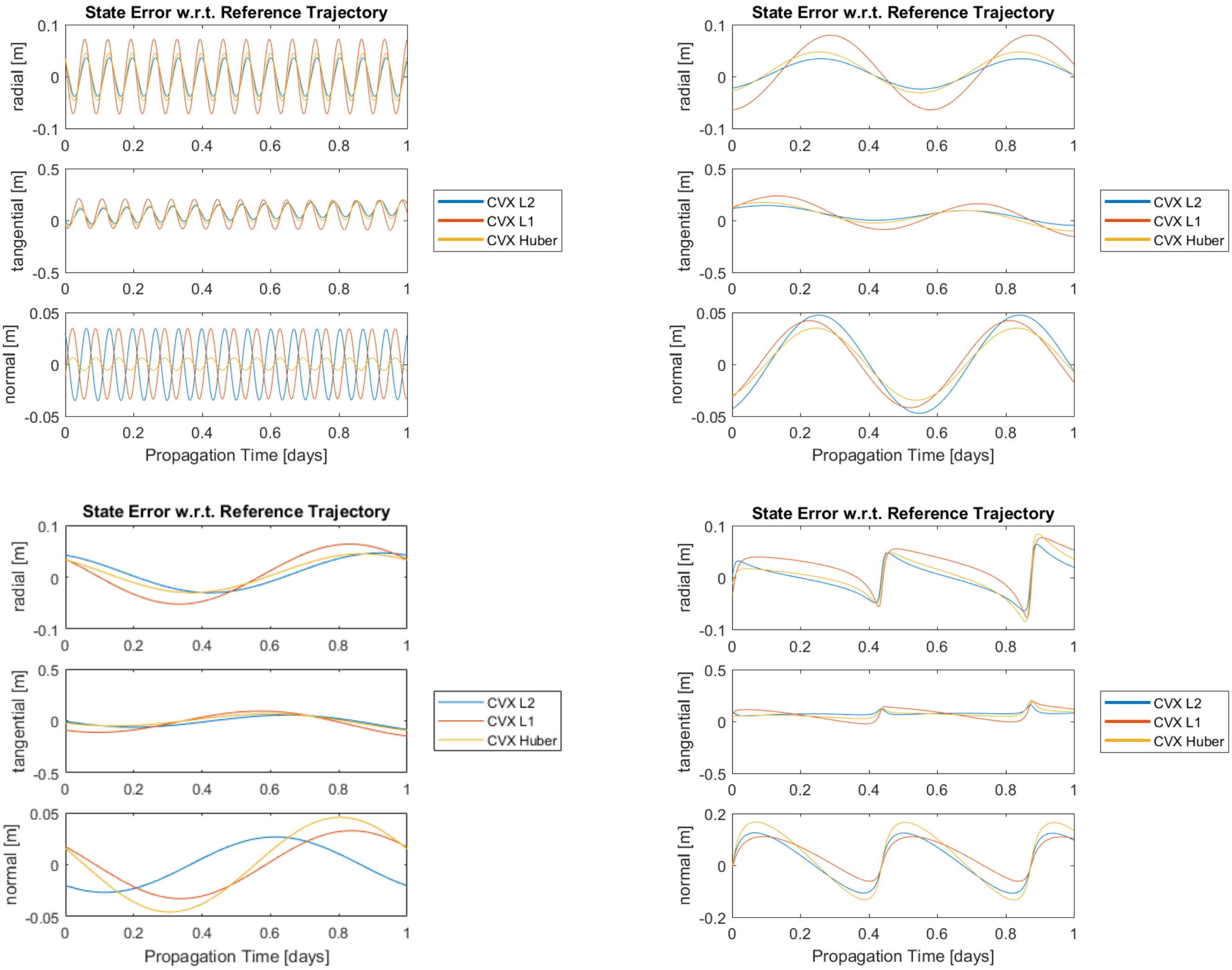


Figure 4.1 - Estimated trajectory w.r.t. the reference trajectory for different scenarios. Upper-left, LEO; Upper-right, MEO; Bottom-left, GEO; Bottom-right, GTO.

4.2 Estimation of Parameters

Throughout this report, the objective of the POD algorithm has been to obtain the state vector that best fits a trajectory to the cloud of measured data. However, in reality, the estimation is usually not only focused on the state but also on other parameters of interest.

These parameters are usually coefficients used to define the mechanical or structural properties of the satellite. In this section, three parameters will be used for this sake: the drag coefficient of the satellite, c_D, the SRP coefficient, c_R, and the manoeuvre commands.

The reason why these parameters should be estimated as well is that they are defined theoretically but are affected by different empirical and unpredictable sources such as unexpected variations in the atmospheric density, imperfections in the construction of the thrusters, errors in the estimated attitude of the spacecraft, etc. Ultimately, they compensate for the unmeasurable imperfections that the modelling of the satellite or the force models might have.

It is important to keep in mind, that the main target of the estimation of these parameters is not to improve the definition of the design characteristics of the satellite (even though this is also interesting), but to accommodate these imperfections to be able to fit a more precise orbit estimation that could serve to obtain more accurate trajectory estimations. The estimation of these parameters can be independently activated, in case the user wants to leave some of them fixed while others are updated.

Table 4.2 – Manoeuvre plan.

Hill Frame	$\boldsymbol{magnitude}[\boldsymbol{m/s^2}]$	*Start [Julian Date (JD)]*	*Duration [s]*
$\boldsymbol{M_1}$	$[10^{-3}\ 10^{-3}\ 10^{-3}]$	$JD_0 + 0.25$	150
$\boldsymbol{M_2}$	$-[10^{-3}\ 10^{-3}\ 10^{-3}]$	$JD_0 + 0.75$	200

To estimate the manoeuvre commands, it is necessary to define a manoeuvre plan first. The plan that will be implemented in these tests will consist on two manoeuvres, applied in the RTN frame, as shown in Table 4.2.

Table 4.3 – Initial parameters and increments applied.

	$\boldsymbol{C_d}$	$\boldsymbol{C_r}$	$\boldsymbol{M_1}[\boldsymbol{m/s^2}]$	$\boldsymbol{M_2}[\boldsymbol{m/s^2}]$
Reference	2.2	1.2	$[10^{-3}\ 10^{-3}\ 10^{-3}]$	$-[10^{-3}\ 10^{-3}\ 10^{-3}]$
Initial error in the parameters ($\boldsymbol{\Delta_1}$)	+0.4	+0.2	$+[10^{-4}\ 10^{-4}\ 10^{-4}]$	$+[10^{-4}\ 10^{-4}\ 10^{-4}]$

The magnitude of the acceleration was chosen under the assumption of chemical propulsion technology, defined by an $I_{sp} = 220\ s$; and a satellite's mass of $2000\ kg$. Manoeuvres in these environments usually deliver $1 - 5\ N$. If, for instance, $2\ N$ is selected as the thrust level, the acceleration of the manoeuvre would be $10^{-3} m/s^2$ of acceleration (based on information from (Airbus Safran Launchers, 2017)). The error for this acceleration was estimated to be a 10% of the acceleration itself.

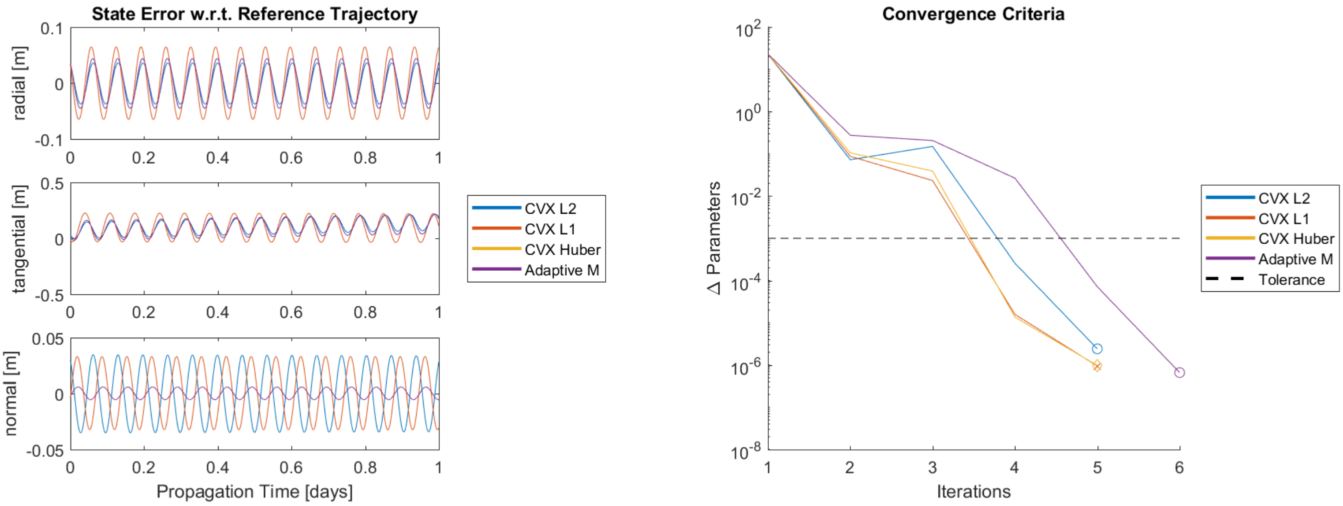


Figure 4.2 – Results for the POD solution when a manoeuvre plan is included.

The theoretical basis to implement this estimation is explained in Chapter 7 of (Montenbrück & Gill, 2000) and briefly introduced in this report in Subsection 2.3.4.1, where a validation process for the SM was also provided. With that information, tests will be run where the reference trajectory is obtained using the following altered parameters.

First of all, manoeuvres will be introduced to the reference trajectory to assess how well the POD batch method can handle this kind of perturbations. Results are shown in Figure 4.2, where the manoeuvre plan described in the first row of Table 4.3 is applied. It is clear that the accuracy remains within the same levels of the nominal case (see Figure 3.25 for comparison).

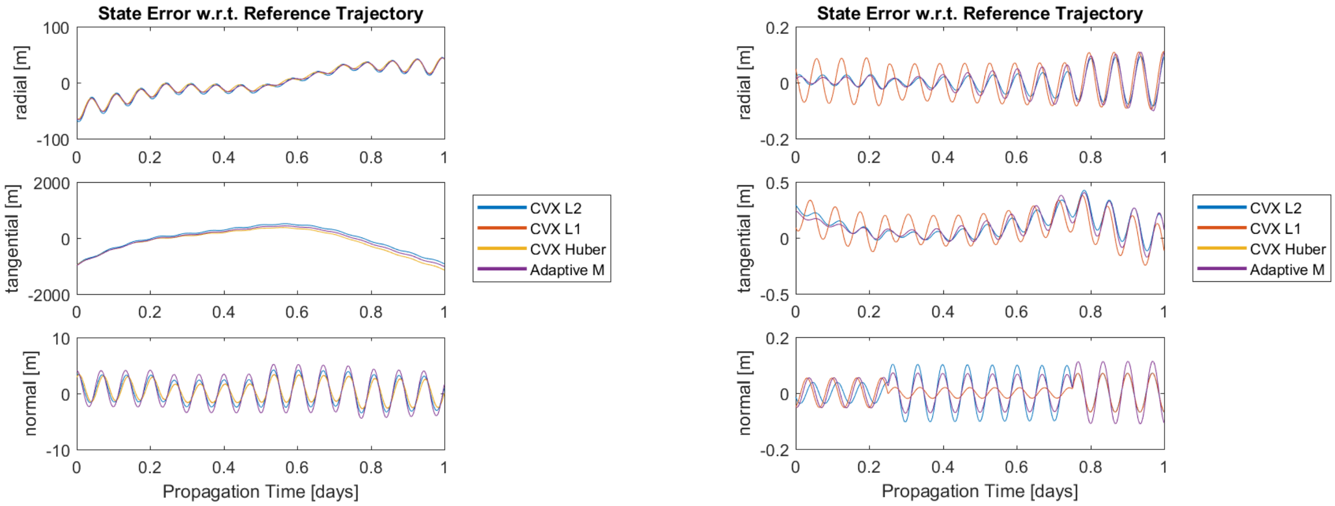


Figure 4.3 - Estimated trajectory w.r.t. the reference trajectory for perturbed parameters (Δ_1). Left, without SM; Right, with SM.

If the perturbations are introduced and the usual POD process is executed, i.e., without the SM, results shown in the left-hand side of Figure 4.3 are obtained. It can be observed that the way in which the POD algorithm compensates for the variations of the parameters is by changing the initial semi-major axis. This makes perfect sense, because the reference trajectory will undergo larger drag and SRP accelerations (because of the applied increments), leading to a faster orbital decay. To compensate this decay (and its subsequent drift in the tangential direction), the POD algorithm places the initial state at a lower altitude that the reference trajectory. The SM is expected to fix this error.

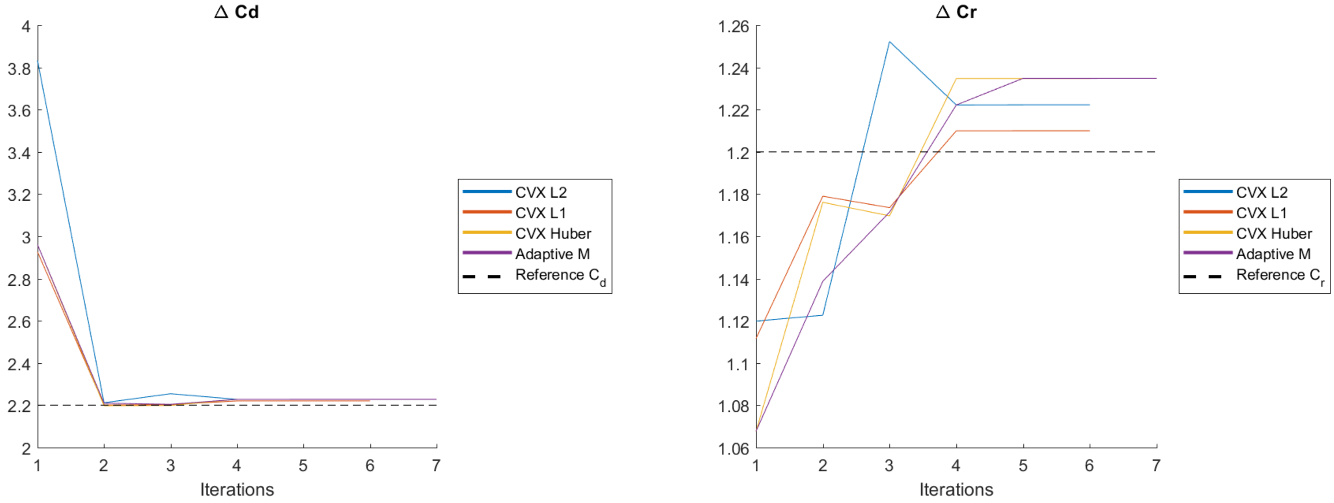


Figure 4.4 – Evolution of the estimated C_D and C_R parameters for Δ_1.

When a POD process including SM is run to estimate these coefficients, the results shown in the right-hand side of Figure 4.3 are obtained. There is clearly a huge improvement in the accuracy for each penalty function. It is also worth remarking that the L2-Norm and both HPF's are much closer to each other than to the L1-Norm.

Also insightful, the evolution of the estimated coefficients is depicted in Figure 4.4. There, the estimation of the manoeuvre commands was not included because the results evolve in a similar manner, with relative errors well below 1% (as for the other coefficients shown in Figure 4.4).

The parameters are successfully estimated by all penalty functions, with slight differences in the final value obtained. The convergence rates are represented in Figure 4.5, where the usual behaviour can be observed.

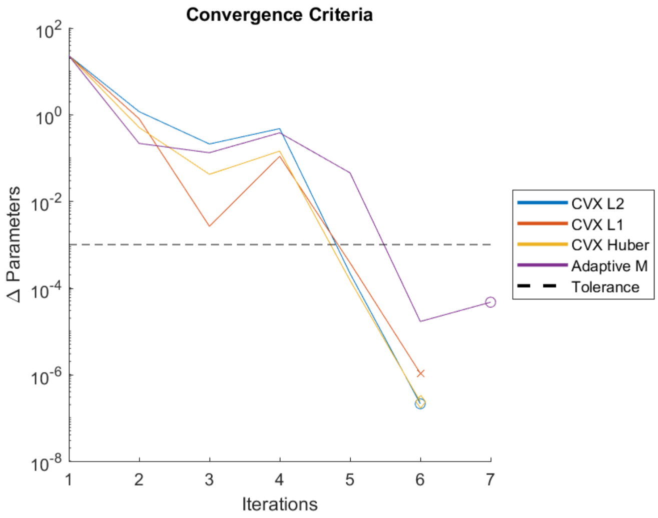


Figure 4.5 – Convergence criteria for Δ_1.

According to the results shown, the estimation of the parameters is clearly validated. However, a tweak was needed for the scaling factor designed in Section 3.3.1. When the SM was included, the CVX toolbox *failed* to find a solution for HPF minimisation. To fix that, the scaling factor defined by Pf (see Equation (3.10)) had to be readjusted from 10 to 1 for HPF, as it was explained in Section 3.3.2.

4.3 Errors in the Models

One of the main reasons why POD methods are so important is because the existing dynamic models are not accurate enough to predict the orbit of a satellite in the real world. Thus, observations help improving the quality of the trajectory estimation.

This situation can also be simulated in a controlled environment. If the reference trajectory is computed using highly accurate models, but the POD is run setting its POP to low accuracy, there will be a gap of quality between both dynamic models that could simulate the gap that would exist between real observations and a sophisticated dynamic model.

This intended reduction in the accuracy of the POD models can be applied mainly to the irregular gravity model, third body accelerations, SRP, or atmospheric drag (see Section 2.2.2).

4.3.1 Irregular Gravity

According to the validation document of the POP used in this POD method (Peñarroya, 2017), the differences between different degree and order levels affect quite substantially the output of the POP, yielding up to kilometre errors after five days of propagation for a LEO, for instance. In this section, the reference trajectory will be generated using order and degree 20 for the EGM2008 (Pavlis, Holmes, Kenyon, & Factor, 2012). On top of that trajectory, the nominal noise layer will be implemented as usual, giving place to the observations.

On the other hand, the POD process will be run setting the degree and order to 10, and the final estimation of the trajectory will be assessed.

When the difference between both configurations is lowered (from 20 to 10), the results shown in Figure 4.6 are obtained. There, two plots are represented: on the left-hand side, the orbital error due to the difference in the EGM coefficients is represented (assuming equal initial state and conditions); on the right-hand side, the estimated trajectory coming from the POD method is shown.

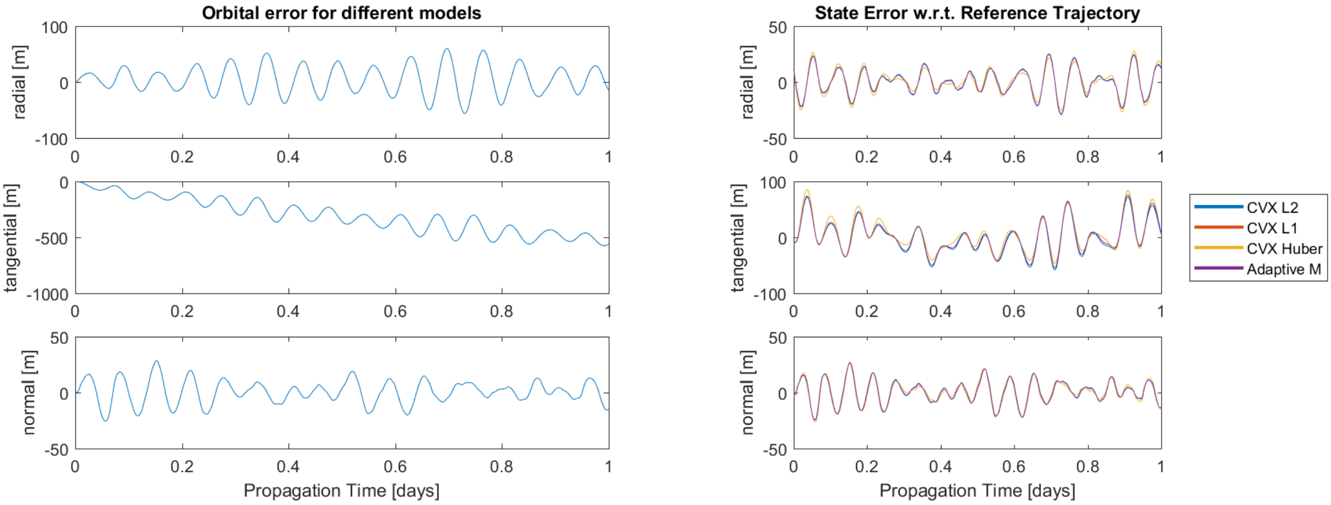


Figure 4.6 – Results for reduced accuracy model, orbital error (left) and POD error (right).

The largest orbital error appears in the tangential component, where a drift represents the lower order EGM propagation being left behind by the higher order one. When the POD batch method is applied, the drift is compensated and the error is drastically reduced, proving the excellent performance of the algorithm against EGM errors, mainly reducing the drift.

4.3.2 Atmospheric Model

As it was introduced in Section 2.2.2, the atmospheric models are responsible for most of the propagation errors in LEO orbits. Even the most accurate models, as the NRLMSISE-00 model implemented in the POP used in this study, are unable to accurately describe the behaviour of the atmosphere. Sudden changes of temperature, for instance, may have a relevant effect on the drag force the satellite undergoes and, thus, affect severely the predictions yielded by the models.

In order to simulate that, a varying c_D will be defined and implemented. This variation will mimic the unexpected variations existing on the real atmosphere basing its definition on a sinusoidal signal, with magnitude 0.5, and centred on the reference value for c_D. Two frequencies will be tested: a frequency of 1 cycle per day, and a frequency of 1 cycle per orbit. These frequencies were selected to represent the two most realistic events: a variation having to do with the orbital position of the satellite around the Earth (and its position w.r.t. the Sun), and a variation due to the rotation of the Earth (and the atmosphere around it). These variations are shown in Figure 4.7.

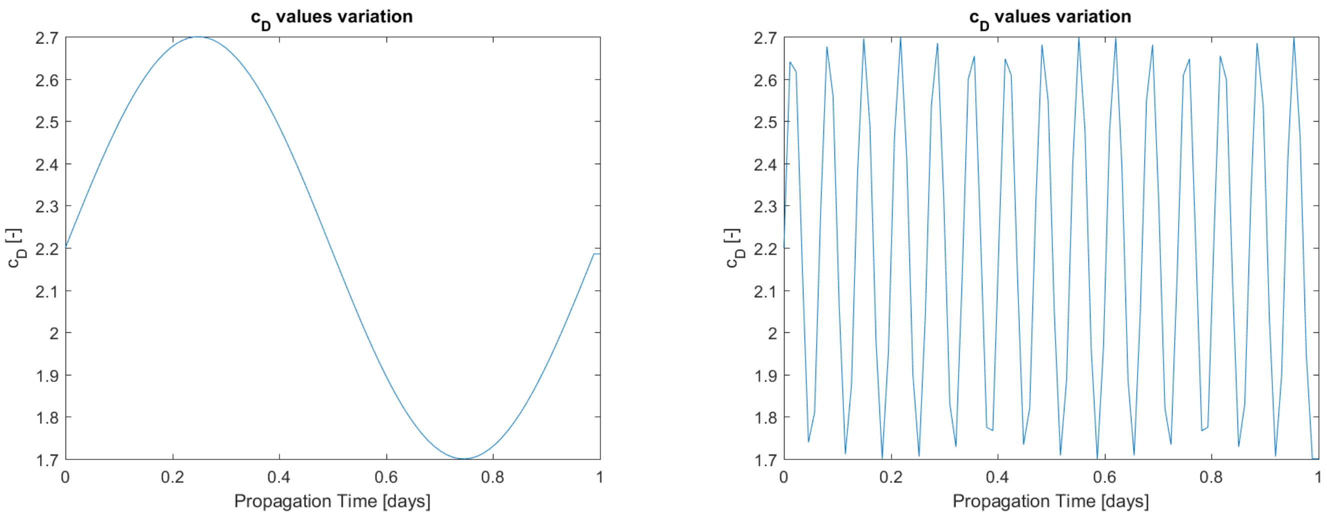


Figure 4.7 – Variations on the c_D value: f=1 osc./day (left) and f=1 osc./orbit (right).

When implemented in the original scenario, they generate an orbital error w.r.t. the non-perturbed c_D trajectories, shown in the figure below. It is easily noticeable that a drift in the tangential component exists. According to the plots, the trajectory corresponding to one oscillation per day gets ahead of the nominal trajectory, something that might result counterintuitive considering that the drag is larger (in mean). However, this is explained by the drift existing in the radial component. As the satellite is decelerated by the drag force, it loses altitude, resulting in the mentioned tangential drift.

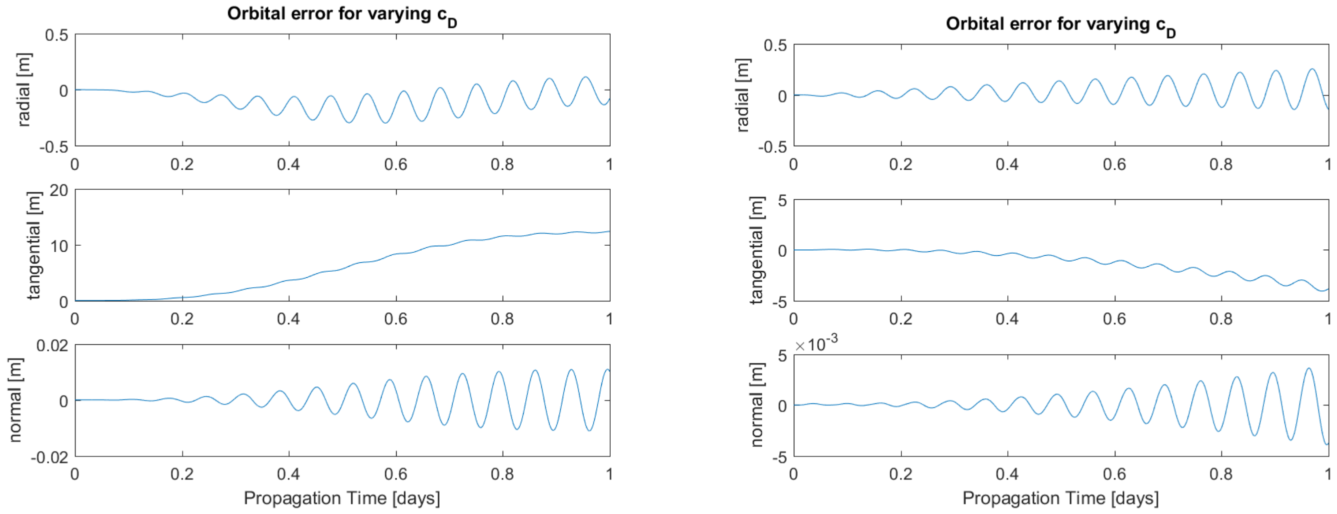


Figure 4.8 – Results for orbital error for varying c_D, f=1 osc./day (left) and f=1 osc./orbit (right).

As always, the nominal noise layer is introduced on top of the trajectory obtained from the variation of the c_D value, i.e., the reference trajectory for this test; and the POD process is run to estimate a solution. Results, shown in Figure 4.9, show a significant reduction (although no complete) of the error for the daily oscillation case, while the orbital oscillation case sees no reduction.

It also interesting to mention that, for the case where the frequency of the oscillation was higher (of 1 cycle per orbit, right-hand plot of Figure 4.8), the orbital error generated was more stable. This is mainly due to two reasons: the negative and positive oscillation cancelling each other, and bringing the mean c_D closer to the reference value; and the cyclic character of the variation causing a resonance in the errors that is easier to absorb by adjusting the initial state.

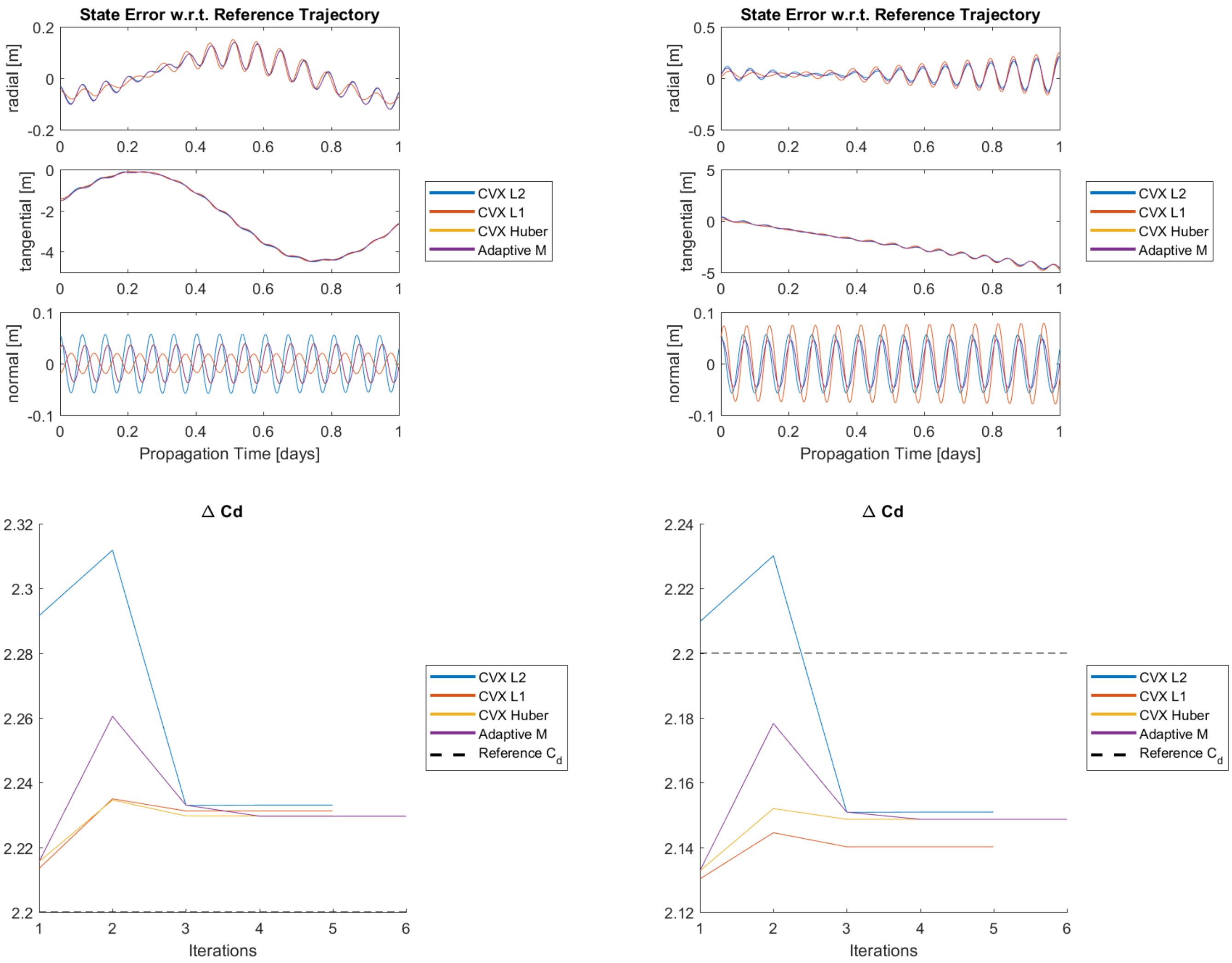


Figure 4.9 – Results for POD error for varying c_D, f=1 osc./day (left) and f=1 osc./orbit (right).

When comparing the different penalty functions used, HPF and its adaptive variant (overlapped in the plots) show the best accuracy, but not by much. This improvement in accuracy is shadowed by the longer computational time and larger number of iterations (in the case of the adaptive HPF) needed. Overall, the results are not as accurate as it has been seen in other tests. This can be due to the use of a constant estimated value for the ballistic coefficient; something that could maybe be improved by proposing a polynomial expression for the evolution of this coefficient. This idea, however, has not been tested in this thesis.

Something that might be counterintuitive is the fact that the errors represented in the upper half of Figure 4.9 are not clearly revolving around 0. This, however, is only an artefact of the frame selection (RTN), and when the results are plotted in ITRF, the expected behaviour appears (see Figure 4.10).

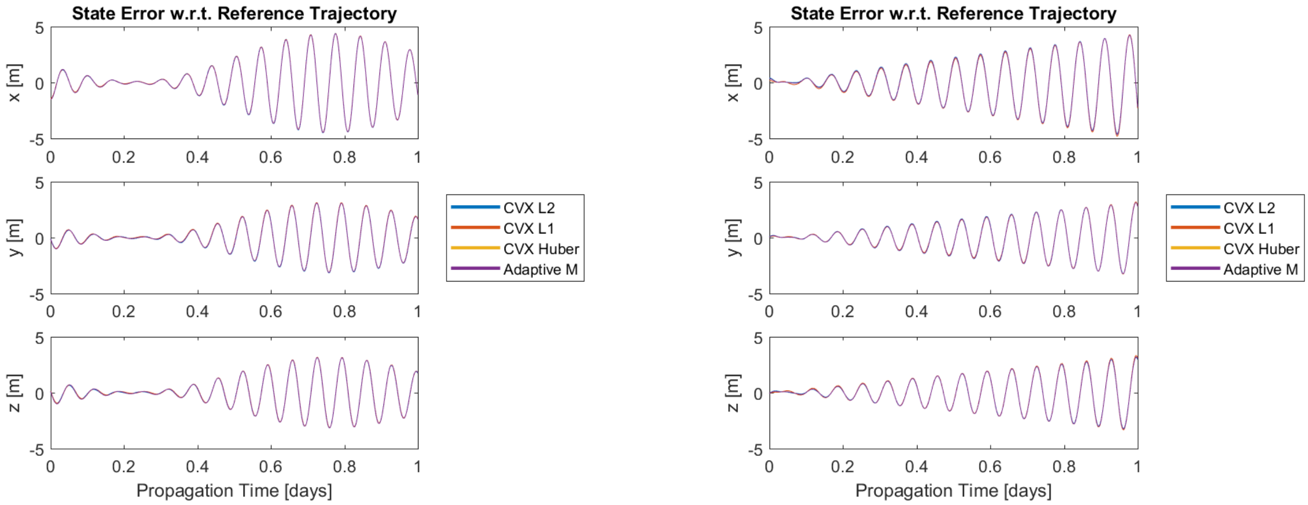


Figure 4.10 – ITRF representation of the results, f=1 osc./day (left) and f=1 osc./orbit (right).

4.4 Azimuth-Elevation Measurements

As it was introduced in Section 2.3.1, azimuth and elevation (AE) observations are used very often for space applications. Conceptually, they differ from PVT observations given by GPS tracking in the sparsity of the data, which is totally dependent on whether the satellite is visible from a certain tracking station and not so much on the epoch (differently from the tests performed in Section 3.5.2).

For these tests, a few modifications were necessary:

- The observations are given as AE, and transformed when needed (see (Montenbrück & Gill, 2000));
- A ground station was defined as observation point, of which the coordinates are shown in Table 4.4.
- Observations whose observability from the defined ground station was null or yielding an elevation value lower than 10° were rejected.
- A new model matrix needed to be defined regarding the partial derivatives of the observation magnitudes (azimuth and elevation) w.r.t. the state components (position and velocity), as introduced in Section 2.3.3 and explained in Section 7.4 from (Montenbrück & Gill, 2000).
- An initial guess needs to be assumed for the first iteration. This guess is set to be $10\sigma\ m$ apart from the true state.
- The observation frequency has been assumed to be $0.1\ \mathrm{Hz}$, i.e., a measurement every 10 seconds.

Once these changes are applied, some tests were run. Now, the visibility between the station and the satellite is vital for the observations, so tests were run using different scenarios. The adaptive HPF has not been implemented here because initial tests showed that its implementation was not mature enough to handle this kind of observations. Perfect atmospheric corrections have been assumed to ease the testing.

Table 4.4 – Ground Station Coordinates.

$\boldsymbol{\lambda}$ [°]	$\boldsymbol{\phi}$ [°]	**Altitude** [$\boldsymbol{m}$]
30	56	100

Same as before, the LEO and MEO scenarios defined in Table 4.1 are used. It is worth remarking that GEO scenarios are not applicable for AE measurements, since the observations are practically constant. The observations obtained using these scenarios are described in Figure 4.11, where the jumps in the azimuth observations are due to the data being ranged from $0 - 360°$.

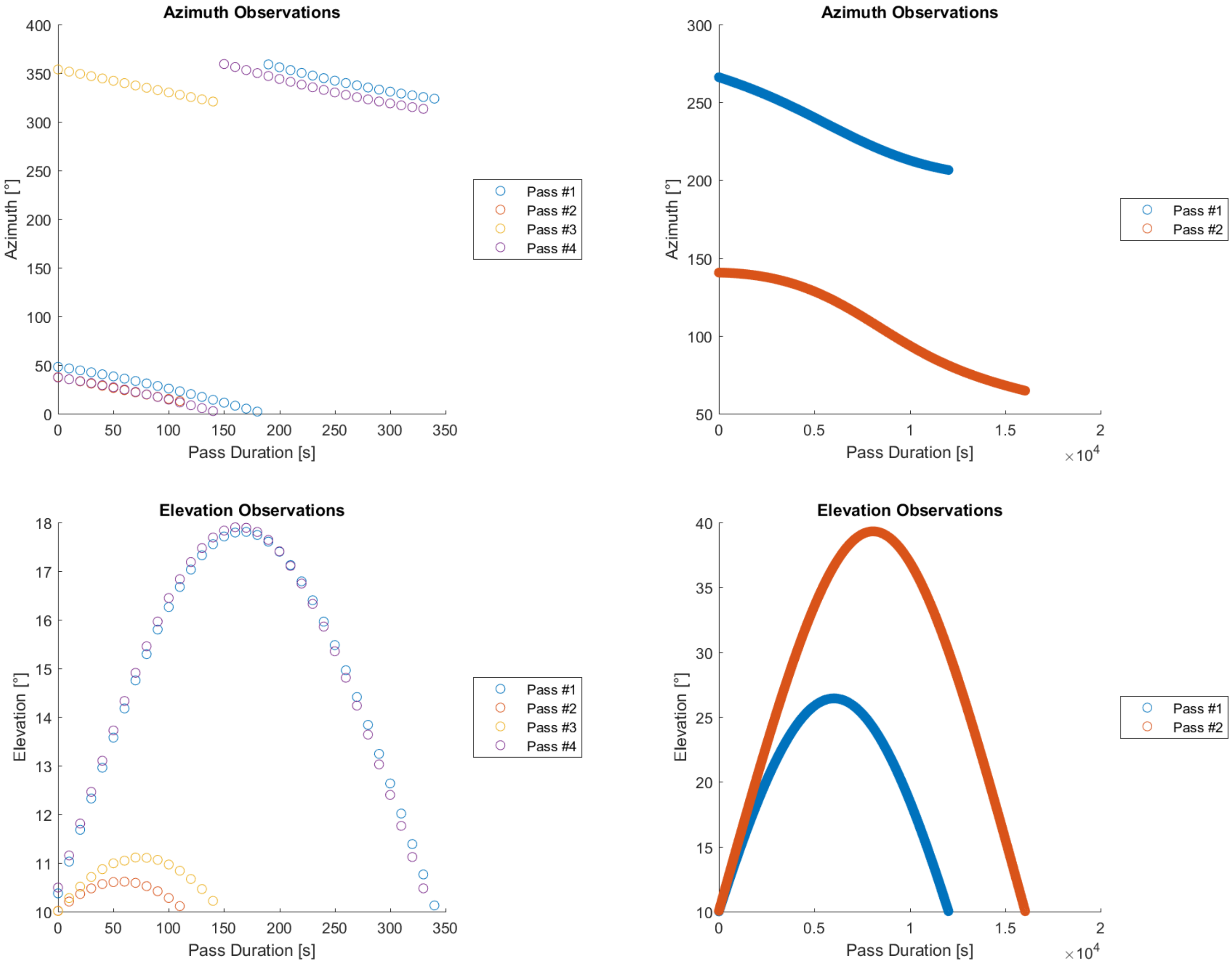


Figure 4.11 – Information about the observations. Left, LEO; Right, MEO.

As shown in Figure 4.12, the L1-Norm offers the best results in the LEO scenario, followed by HPF and the LSQ method. The LSQ, for instance, seems to find a more eccentric solution, which is mainly due to the orbital arcs being relatively short at each pass. Because of this, and due to the higher sensitivity of the LSQ method to outlying measurements, the L1-Norm is able to find a more accurate trajectory.

In the MEO scenario, the number of observations is clearly superior (due to the longer orbital period and favourable location of the ground station), and the results do not differ that significantly, although the L1-Norm seems to improve the results from the other penalty functions.

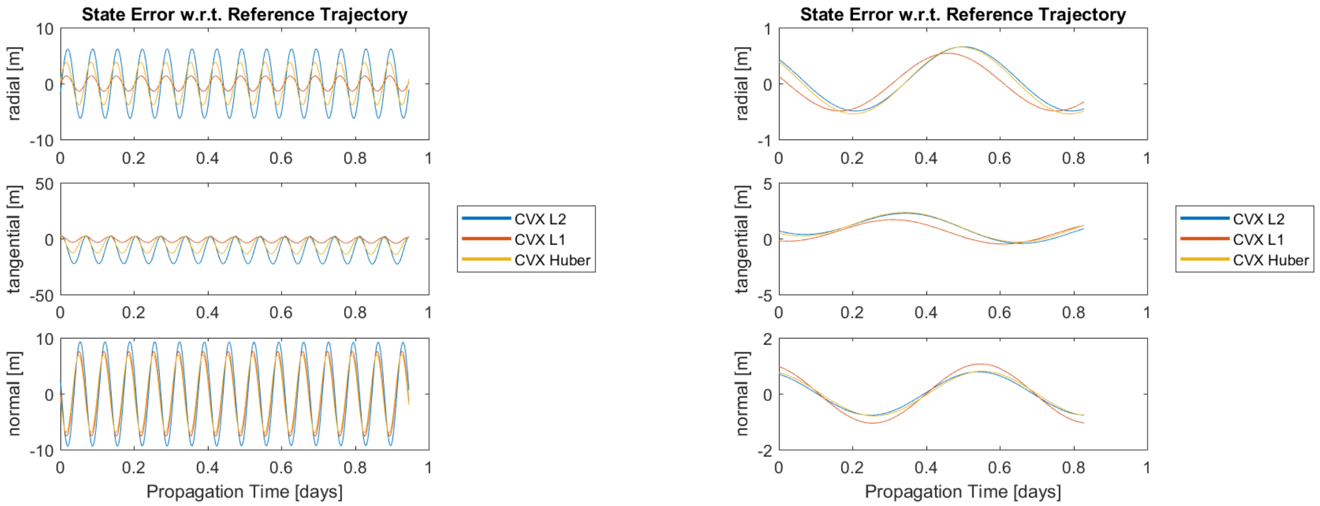


Figure 4.12 - Estimated trajectory w.r.t. the reference trajectory for different scenarios. Left, LEO; Right, MEO.

It is safe to conclude that, when the data is restricted to a small portion of the orbital arc, the L1-Norm offers better solutions.

Specifically, for the LEO case (the most restrictive due to the lesser number of observations), after the observations with an elevation of less than $10°$ were rejected, a total of 15 outlying measurements remained in the observation data. All of them were detected in each penalty function implementation.

Based on the results obtained so far, it seems logical that, if not every outlier is detected, the L1-Norm will offer better results. To try that, a similar test to what was shown in Subsection 3.5.1.2.3 was run, where the magnitude of the outliers is ranged from $2 - 10\sigma$ (as opposed to the nominal configuration with $10 - 100\sigma$). Results are shown in Figure 4.13.

For the LEO scenario, the number of detected outliers was still the same for all the penalty functions, although it saw a reduction, due to the blurrier frontier between the stochastic noise and the outlier's magnitude. This led to clearly larger state errors, when compared to what was observed in Figure 4.12.

The number of outliers detected for the MEO scenario was also the same for each penalty function. However, it was practically unaffected w.r.t. the previously obtained results. This is due to the larger number of observations making the outliers not so predominant in the optimisation process, and the magnitude of the latter being much more restrained. Interesting enough, the L1-Norm needs more iterations to converge.

More sparse observations, thus, would represent a scenario where more robust methods would be required to improve the accuracy of the results. HPF ends up between the two norms, as happened in previous tests.

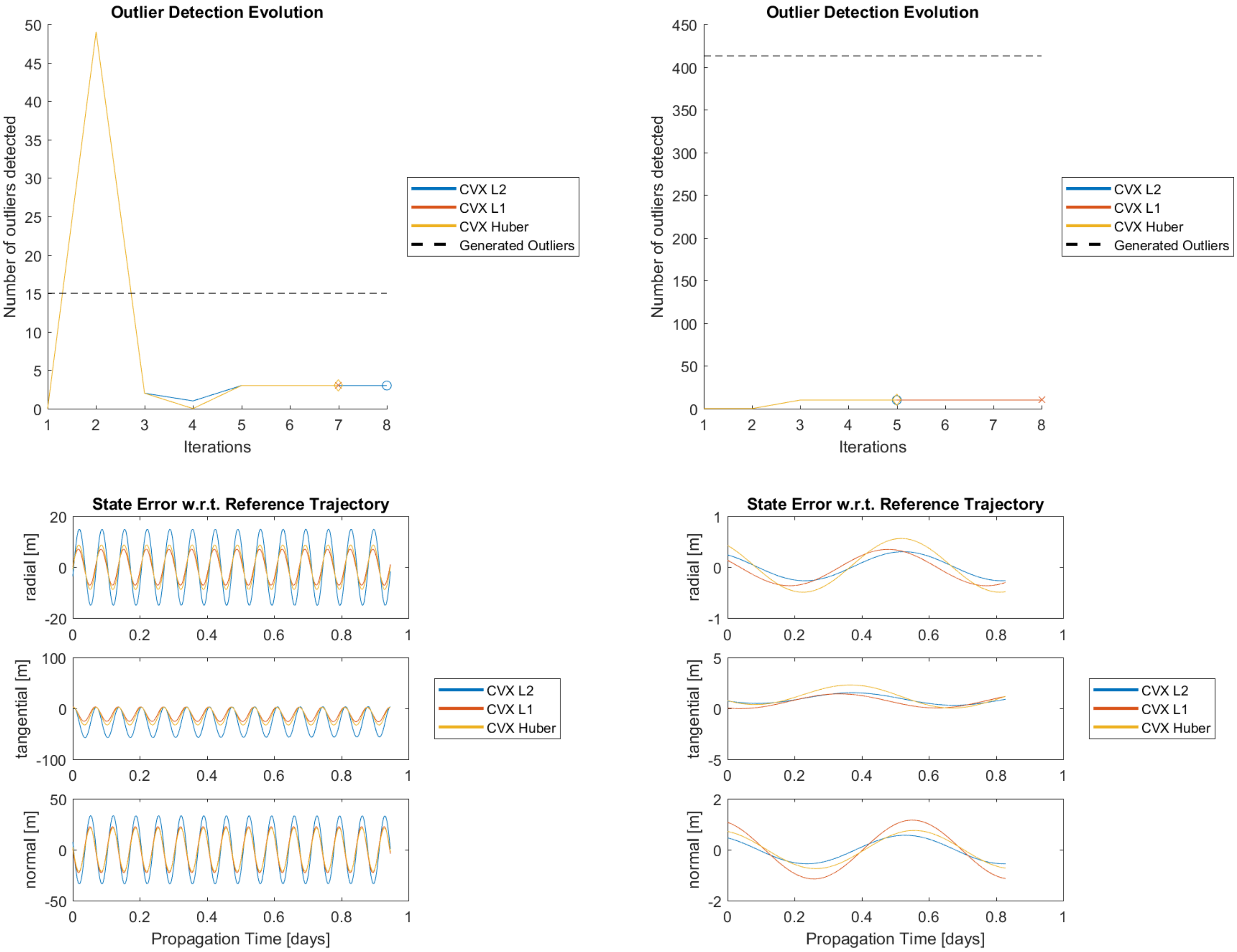


Figure 4.13 - Results for outlier magnitudes from $2-10\sigma$.
Left, LEO; Right, MEO.

4.5 Conclusions

Based on the additional tests carried out in this chapter, the adaptive M HPF looks promising and the results obtained with this method are highly satisfactory. However, further developments and a dedicated study should be made to properly assess its applicability and, perhaps, figure out a better way to predict its value.

The orbital conditions do not really affect the POD process and, given the needed observations and parameters, any orbit can be accurately predicted using any of the penalty functions studied here.

When it comes to parameter estimation, all the penalty functions behave in a similar manner and, although it is true that differences exist, the results are too close to each other to proclaim a particular cost function to be better.

On the other hand, in the tests regarding errors in the modelling (an exemplification of the gap existing between reality and a certain simulation environment), HPF, both adaptive and normal formulations, shows slightly more accurate results in the drag model. The differences are, again, not conclusive, but visible.

The most relevant test of this chapter comes with AE observations. This kind of observations is particularly interesting due to the orbital sparsity they present and its limited availability. When the number of observations is restricted and not uniformly distributed, finding the best trajectory estimation becomes more difficult for the algorithm. This leads to a not so accurate outlier detection and, thus, L1-Norm and HPF offering better performance.

5 Real Study-Case: GOCE

At this point, the different functionalities of the POD batch methods have been extensively tested and evaluated. Many comparisons have been made and conclusions about the different characteristics of the penalty functions used have been extracted.

However, all these case studies have taken place under a controlled and simulated environment, where every aspect is known and results can be biased by this fact. Thus, in this section, real data will be used as observations to serve as a final test for the POD algorithm developed during this thesis, and as a final comparison between the performances of the different penalty functions when real observations are used.

The observations have been provided by the TU Delft and belong to the Gravity field and steady-state Ocean Circulation Explorer (GOCE) satellite. This mission was launched on the 17th of March 2009 by ESA (European Space Agency) and it was the first Earth explorer mission in orbit. Its main objective was to unravel one of the most fundamental forces on Earth: the gravity field (ESA, 2017).

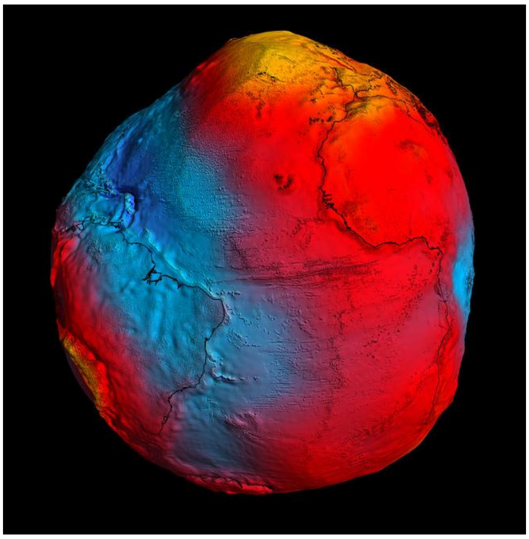


Figure 5.1 - The GOCE geoid (ESA, 2017).

As the objective was to measure the gravity field in a highly detailed manner, a small electrical ion thruster was used to compensate for the drag forces the satellite underwent. The results obtained are shown in Figure 5.1, obtained after the mission ended in October 2013, when the satellite ran out of fuel.

The format in which the data is given is sp3, a standard format for orbital positions and sometimes velocities (more detailed information can be read in (Hilla, 2016)). For these tests, two different trajectories were provided: SPPLEO and Reduced-Dynamics Orbit Determination (RDOD). The former is a coarse-point positioning estimation of the position and clock of the LEO receiver using pseudo-range observations. Once this solution is obtained, a PosFit solution is estimated as the results of fitting a LSQ method and eliminate outlying measurements. Finally, a RDOD is obtained starting from the PosFit solution and including the precise GPS carrier-phase measurements. All the observations are given in ECEF. More information about this process and its nomenclature can be found in (DLR, 2017).

Always through this document, the obtained results have been compared to a reference trajectory that served as ground truth. In previous chapters, this was very simple, since the observations were artificially generated from that trajectory. Here, however, there is no "true" knowledge of the actual trajectory of the satellite, thus, the RDOD solution will be used as reference trajectory, while the SPPLEO will be the input, i.e., the measurements; for the POD batch method. Figure 5.2 shows the difference between both solutions.

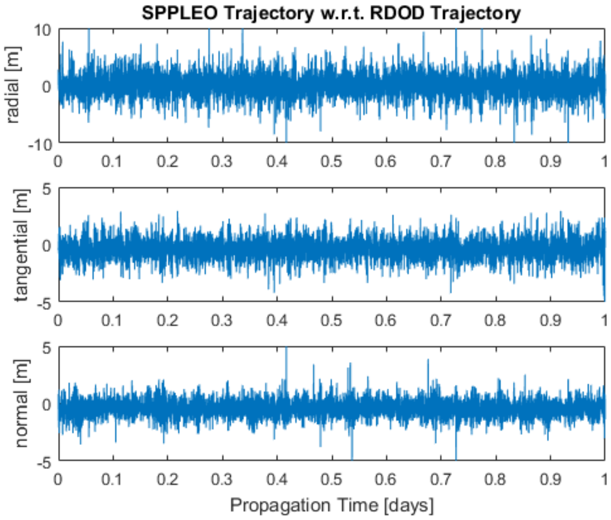


Figure 5.2 – Errors in the measurements (SPPLEO) w.r.t. the reference trajectory (RDOD).

In (Bock, et al., 2007), it is explained which models were used in the POP for the orbit determination of GOCE. Table 5.1 shows a comparison of the simulation environment used there and the one configured for the results obtained in this test.

Table 5.1 – Orbit determination environment comparison (Bock, et al., 2007).

	(Bock, et al., 2007)	**MyProp**
Gravitational Forces	EIGEN-5S (150 x 150)	EGM2008 (100 x 100)
Solid Earth, pole, and ocean tides	Yes	No
Luni-Solar planetary gravity	DE405	DE405
Drag	No	Yes
SRP	No	No
Estimation Technique	Batch LSQ	LSQ, L1-Norm, HPF

The coefficients used for the gravitational model, though lower order and degree are used, are high enough not to represent any problems in terms of accuracy. The 3rd body accelerations and the SRP are equally configured. Solid Earth, pole, and ocean tides have not been included in *MyProp*, which could lead to dynamical errors in the final results.

The drag force was deactivated in the propagations run by (Bock, et al., 2007), accounting for the drag compensation that the ion thruster gives. However, in this test, the drag acceleration has been left active, in order to make use of the SM to estimate the drag coefficient. If the estimation results in a value close to 0, the parameter estimation will be successfully validated, even for real data.

The estimation of the parameters will be restricted to the drag coefficient estimation, i.e., the rest of the parameters will be blocked at their initial value. Results are shown in Figure 5.3.

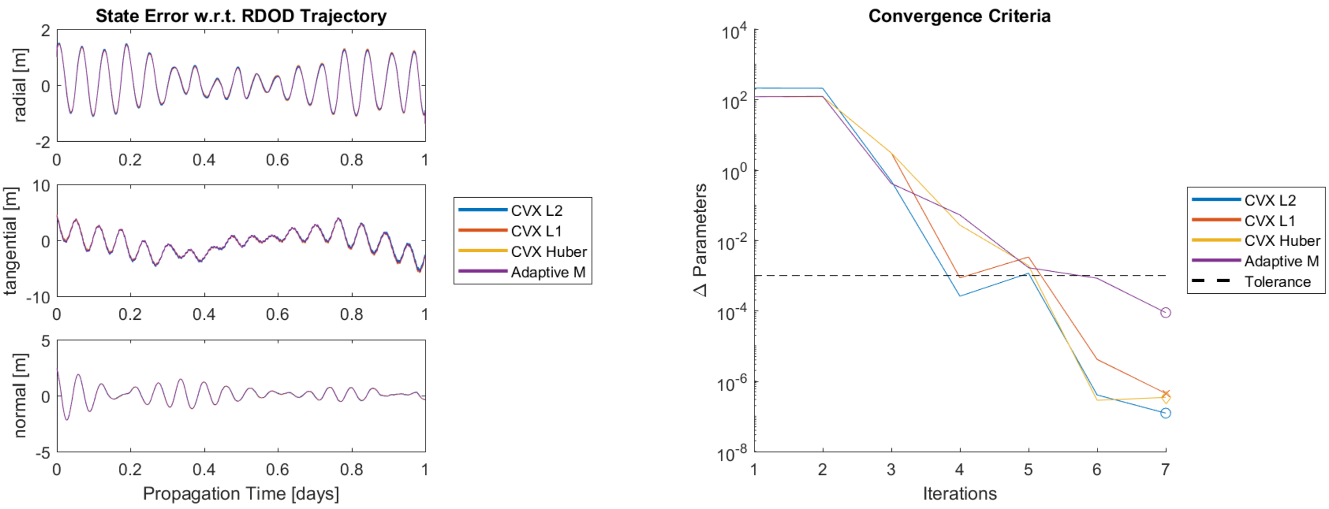


Figure 5.3 – Results for real data observations.

The estimation of the drag coefficient converges to a value of 0, which agrees exactly with the design of the mission, as shown in Figure 5.4.

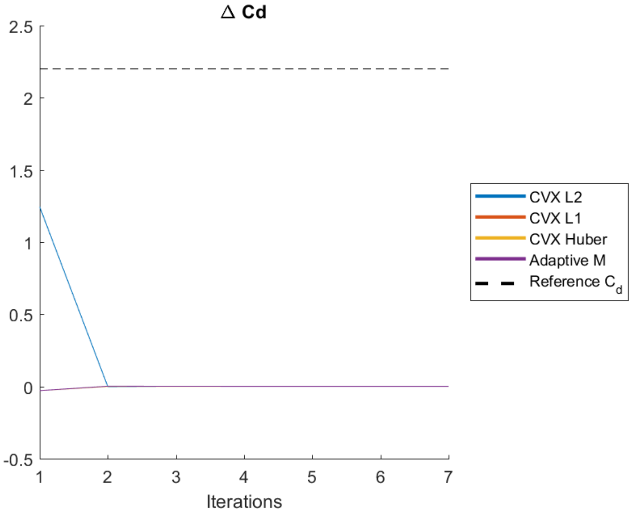


Figure 5.4 – Estimation of the c_D coefficient.

This serves as an additional validation of the SM implemented in the POD method. Comparatively, L1-Norm and HPF seem to find the solution for this value already with high accuracy in the first iteration.

However, the errors obtained (shown in Figure 5.3), are still in the order of the errors contained in the measurements (see Figure 5.2). The stochastic component is removed, but a lower frequency error is clearly present. This error is due to the differences in the models implemented in the propagators used for this thesis and in TU Delft, and not to the POD method itself. Trying to fit a trajectory for a relatively long timespan becomes more difficult the more the models differ from reality. Further improvements in the models of the POP, such as solid tides, for example; would be needed to improve the results.

When the data arcs are reduced to $12, 8, 4$, and 2 hours, respectively; the following results are obtained:

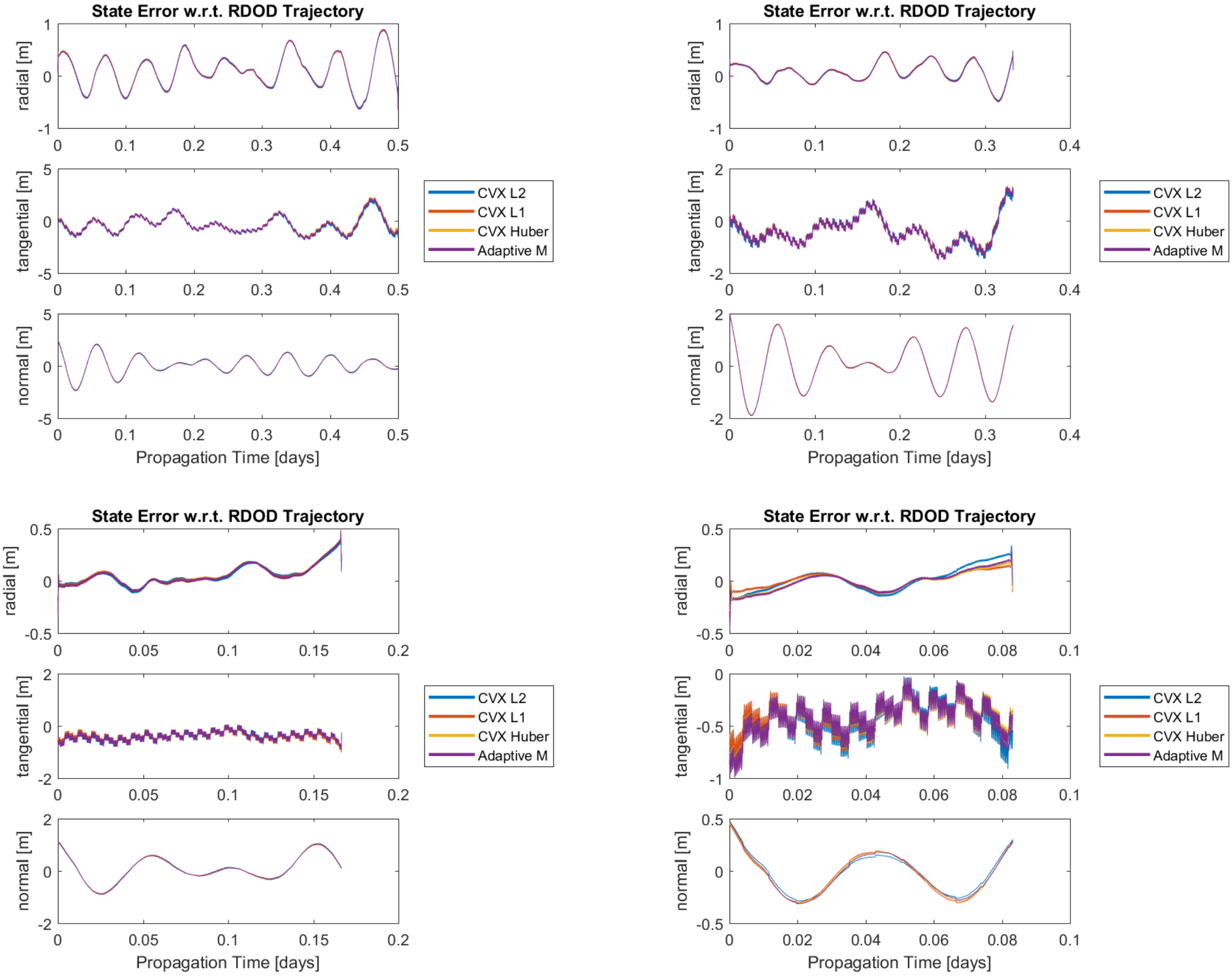


Figure 5.5 – Results for real data arcs of 12h (upper-left), 8h (upper-right), 4h (bottom-left), and 2h (bottom-right).

Not surprisingly, when the data arcs are shorter, the magnitude of the errors is significantly reduced. This is due to the differences in the models spreading wider as time passes by. When the temporal dimension is restricted, these models have no time to differentiate themselves significantly, and a closer estimation can be obtained with the POP implemented in the developed algorithm.

Interestingly enough, when the timespan is reduced, the tangential component starts to show small offsets every few seconds (already present before, but not easily visible). In the beginning the possibility of the jumps being due to the manoeuvres was considered but the magnitude of the accelerations generated by the thruster were too small to cause this effect. A second consideration included the update in the empirical acceleration by the feedback loop on-board.

Again, the update frequency and the magnitudes (shown in (Bock, et al., 2007)) did not agree with the observed behaviour. When the plots are generated using the ITRF, Figure 5.6 is obtained. There, the 2-hour arc is plotted, because it is easier to distinguish the jumps.

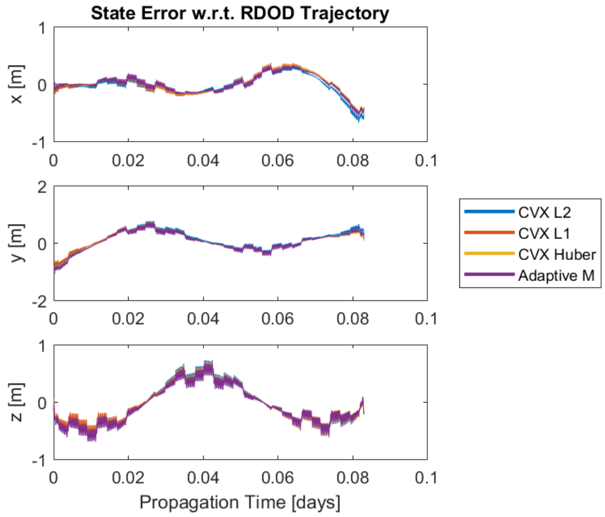


Figure 5.6 - ITRF representation of the results.

The erratic behaviour is still there when the inertial reference frame is utilised, discarding, thus, the frame transformation to RTN as the cause of the error.

After further investigation, the root of the problem was found to be related to the transformation from ECEF to ITRF that needs to be made in this test. As mentioned before, the observations are given in ECEF but, due to the design of the POP used in this algorithm, a transformation to ITRF was needed. This transformation was coded following the recommendation gathered in (IERS, 2016), but is unrelated to the libraries used within *MyProp*, which are taken from the NOVAS (Naval Observatory Vector Astrometry Software) libraries (NOVAS, 2011). This hypothesis has not been properly confirmed but early investigations point to the problem lying in the numerical resolution used for the epoch. NOVAS' libraries usually require the user to give the epoch as two separate inputs: the integer part of the JD and the decimal part. This was not implemented in the used transformation and the order of magnitude of the errors derived from there are in accordance with the values observed in the tangential component.

Summing up, the final solution is not as accurate as the RDOD solution offered by TU Delft. However, this was not expected from the developed algorithm, which was designed to serve as a shell to compare the different penalty functions studied along this report. From that point of view, the performances of the penalty functions used remain quite similar and no clear preference can be given.

6 Conclusions and Recommendations

This research has served as a comparison between different penalty functions and its application to different orbit determination study-cases. Emphasis has been placed on how these penalty functions could be utilised for this sake, and on how well these different methods perform when applied to different situations.

As a result of this main goal, a precise orbit determination (and propagation) tool has been developed, validated, and tested, yielding excellent results, yet where room for improvement exists. This represents a huge deal of work, considering that all these tools have been developed from scratch, and the level of readiness that they have reached after the tests performed in the timeframe of this thesis.

6.1 Conclusions

In this chapter, the research questions, as previously posed in Chapter 1, will be revisited and proper answers, based on the experience gathered during the thesis, will be provided, complemented with the different results obtained throughout this research. Note that the third sub-question has been slightly modified from its initial formulation.

1. **What are the main drawbacks of the currently used penalty functions in POD?**

The literature review showed that, currently, most of the POD methods implement the LSQ method to perform an orbit determination. This method has been used for several decades already and has always yielded excellent results, when complemented with the proper tools, e.g., for outlier detection.

In Section 3.5.1, it was clearly observed that, when the observations present outlying measurements, the LSQ method suffers hugely from these outliers and fails to accomplish certain levels of accuracy. This is not related to the algorithm used, but to the definition of the penalty function itself, that weighs the errors according to the L2-Norm of the residuals, tilting, thus, towards values that are clearly apart from the trend of the data.

While it is true that, nowadays, POD methods and algorithms are paired with powerful outlier detection functions, certain applications might present a restriction in this sense. The computational effort of running a outlier detection function usually increases the number of iterations used to reach convergence and, while a more accurate solution is found afterwards, there will be a trade-off depending on the application.

It was also observed that, when the data set features very sparse observations, the LSQ method also struggles finding an accurate solution for the estimated trajectory.

2. **How can convex cost functions be applied to POD algorithms in an efficient and robust way?**

Modern optimisation techniques are becoming more and more popular, and CO is definitely among the most prolific optimisation topics these days. Thanks to the tool developed by (Grant & Boyd, 2017), applying these methods to different problems becomes much easier.

Due to the nature of the penalty functions involved, which are all convex, the implementation was rather simple from the MATLAB environment used for this research (although a reformulation was needed for the L1-Norm). The CVX toolbox was implemented in the POD algorithm in substitution of the usual LSQ formulation. Its interface is extremely user-friendly and easy to understand and implement.

From a computational point of view, the fact that the toolbox is designed as a general problem optimiser, makes the computation last longer than with the LSQ solution. But only up to 7% for the same function definition. When other penalty functions are applied, the computational time also rises due to the optimisation process, but also to the definition of the function itself. For instance, the definition of a function such as HPF is much more delicate to implement due to its fragmented nature.

On the other hand, the versatility offered by the tool clearly outweighs the rather small time differences, especially for implementation in a research environment, where tests needed to be run repeatedly and with different configurations. The results obtained in this report could be used as a first order conclusion, based on the performances instead of on the computational effort, and, once a method is chosen, a dedicated optimisation process can be designed, enhancing the computational performances of a generalised optimisation toolbox as CVX.

Regarding its final application, some adjustments were needed to improve the minimisation process and reach convergence satisfactorily, especially for the L1-Norm and HPF. After some empirical tests, a relatively simple solution was found so that CVX was able to properly process the different penalty functions.

3. **How do the L1-Norm and the Huber penalty function perform in comparison to the current state of the art**?

After the application of the CO techniques to POD methods had been determined as feasible, a comparison between LSQ, L1-Norm, and HPF took place in the last three chapters of this report. The core of the research and the most relevant outcome came from this question.

It was clearly observed that, when no outlier detection is present, the L1-Norm and HPF offer much better results than the traditional LSQ method, i.e., can be considered very robust functions in this aspect. This is also true when outliers are processed but the noise presents characteristics that make it difficult to process, such as outliers whose magnitudes do not differ that much from the stochastic noise.

When the outliers are correctly processed, LSQ methods are slightly favoured and offer higher efficiency and accuracy. However, many tests resulted in HPF yielding the best results or, at least, at a similar level of performance compared to the LSQ methods. This is a great result, since HPF was also paired with the best penalty function when the outliers were not processed, i.e., the L1-Norm. Being able to use a penalty function that performs extremely well under very contrasting conditions is something to be taken into consideration when designing POD algorithms.

The second drawback of the LSQ method was also that sparse data sets are more difficult to process by the algorithm. Again, this was shown to be solved by robust functions, which were able to improve the results and estimate a more accurate trajectory when AE observations were used. Especially in the LEO scenario, where the observations where more scattered.

Regarding the rest of the tests carried out, the LSQ and HPF usually offer the best results, while the L1-Norm presents lower accuracy (given that outliers were processed). With that in mind, the recommendation would be very dependent on the application needed.

It is clear, and all these years of application to the space industry prove, that the LSQ offers high quality results when it comes to efficiency and accuracy, if the tool in use is adequately complemented with outlier processing tools and the quality of the measurements is also fine.

HPF has offered excellent results throughout all the different tests carried in this report and, its adaptive form (although not finely tuned) is also able to obtain extremely good results both when the observations are processed and when not.

The L1-Norm, on the other hand, has proven itself to be a great option when used with data containing outliers, where its robustness shines. However, when accuracy is a concern and processing tools are available (which is practically always), this penalty function is not the option to take. For this reason, there is no real advantage on using this penalty function over HPF. The computational efficiency is slightly higher for the L1-Norm, but the better results offered by HPF are clearly more significant in this trade-off.

Be that as it may, the selection of a penalty function depends entirely on the target of its application. Processed and rich data should be paired with LSQ-like methods. Sparse and noisy observations need a more robust penalty function that could estimate the trajectory accurately enough not to be too bothered by the presence of outliers. If the computational effort is not a very restrictive constraint, as in on-ground applications, HPF will offer the versatility needed to obtain excellent results independently of the quality of the observations. If further developments and validations of this penalty function are conducted in the future, and its readiness level is accordingly risen, it could be a more robust replacement for traditional orbit determination methods even in a professional environment.

6.2 Recommendations

During this research, several topics arose as points where further investigation could be carried out to obtain interesting results. In this section, some of the main points of interest are gathered to propose a possible continuation to this topic.

The scaling parameters defined in Section 3.3 were computed making use of an empirical process. This resulted in the desired convergence capabilities and the final results were obtained without major hindrances. However, after the first computation of these scaling parameters, in Section 4.2, convergence problems reappeared giving place to a slight reformulation of these parameters. A more reliable estimation of the scaling factors to be used in these problems (or a different technique to improve the convergence capabilities of the CVX toolbox) could add robustness to the algorithm.

In Section 3.4, it was explained how the L2-Norm was the only penalty function that presented a properly defined error estimator (the formal error). An attempt of error estimation for the L1-Norm was considered, but the proposed approach was not mature enough. From that point on, all the results were compared to the reference trajectory to assess the quality of the POD solution. It would have been, however, extremely interesting to be able to also compare the obtained results to the corresponding estimation of the error of each penalty function. Thus, the design of error estimators for the L1-Norm and/or HPF could give raise to new considerations to be taken into account when assessing which penalty function should be used for a certain application.

Systematic errors other than gravity model and atmospheric model errors were not introduced in the generated noise in this research. This kind of errors could be very interesting to observe and test, since they would add a behaviour that is neither predicted by the models nor by the parameters. A priori, and if the outlier processing function does not remove these observations subject to systematic errors by considering them outliers, they should represent another case where more robust penalty functions outperform traditional methods.

When the tests concerning AE observations (see Section 4.4) took place, a single ground station was considered. This was mainly due to the fact that what was intended was to represent the sparsity that usually characterises this kind of data. Nevertheless, this is not usually the case in reality, where more than one ground facility is available. A priori, the results after adding more measurements should tilt the balance towards the LSQ side (as opposed as the L1-Norm), but to truly prove this, some tests could be run adding one or more ground stations.

As was explained in Chapter 2, the POP implemented in the algorithm developed in this research included many different force models and perturbations (even eclipses were considered). However, there are still many upgrades to be made. Something that would have been interesting for the comparison made in this report using GOCE data is the inclusion of solid Earth, polar, and ocean tides. This could be implemented without much effort in the current POP, but the time-frame did not allow for this, as it was not one of the main tasks of the thesis.

The estimation of a ballistic coefficient as a constant value might be, according to the results observed in Section 4.3.2, an important limiting factor. Thus, it would be really interesting to estimate this parameter with a polynomial to check whether the final estimation sees any improvement or not.

External accelerations could also be included in the POD process as they are taken by the accelerometers on-board the satellites. Their value could be estimated to compensate for errors that are not accurately defined by the rest of the parameters.

Finally, further studies could be carried concerning the development and validation of a POD method featuring HPF as penalty function. Results from this thesis place this convex function as a not only accurate but also robust engine for any orbit determination process. As mentioned before, if its readiness level is sufficiently risen, applications including this cost function could be developed and applied to professional environments and should, therefore, be investigated.

7 REFERENCES

AGI. (2016). *STK Programming Interface*. (STK) Retrieved 11 01, 2016, from http://help.agi.com/stkdevkit/index.html?page=source%2FcoreLibraries.htm

AGI. (2017, 01 25). *Satellite Tool Kit*. Retrieved from Satellite Tool Kit: https://www.agi.com/products/stk11/

Airbus Safran Launchers. (2017). *CHEMICAL MONOPROPELLANT THRUSTER FAMILY.* Retrieved from http://www.space-propulsion.com/brochures/hydrazine-thrusters/hydrazine-thrusters.pdf

Altamimi, Z., Boucher, C., & Willis, P. (2005). Terrestrial reference frame requirements within GGOS. *Journal of Geodynamics, 40*, 363-374.

Arnol'd, V. I. (1997). *Mathematical Methods of Classical Mechanics (2nd ed.).* Springer Science & Bussiness Media.

Banahan, M., Brady, D., & Doran, M. (1991). *The C Book.* Addison-Wesley.

Bassett, G., & Koenker, R. (1978). Asymptotic Theory of Least Absolute Error Regression. *Journal of the American Statistical Association*, 618-622.

Bhagat, M. R. (2016). *Convex Guidance for ENVISAT Rendez-Vous.* Delft, The Netherlands: TU Delft.

Biedenharn, L. C., & Louck, J. D. (n.d.). *Angular Momentum in Quantum Physics.* Reading: Addison-Wesley.

Bierman, G. J., & Thornton, C. L. (1977). Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study. *Automatica, 13*, 23-35.

Bock, H., Jäggi, A., Svehla, D., Beutler, G., Hugentobler, U., & VIsser, P. (2007). Precise orbit determination for the GOCE satellite using GPS. *Advances in Space Research, 39*, 1.638-1.647.

Boyd, S., & Vandenberghe, L. (2009). *Convex Optimisation* (7th ed.). Cambridge: Cambridge University Press.

Boyer, C. B. (1989). *A History of Mathematics.* New York: Wiley.

Branham, R. L. (1986). Error Estimates with L1 Solutions. *Celestial Mechanics, 39*, 239-247.

Capitaine, N., & Wallace, P. T. (2004). *Improvements in the Precession-Nutation models.* Paris.

Celestrak. (2016). *EOP and Space Weather Data*. Retrieved 11 2, 2016, from https://celestrak.com/SpaceData/

Cerri, L., Berthias, J. P., Bertiger, W. I., Haines, B. J., Lemoine, F. G., Mercier, F., . . . Ziebart, M. (2010). Precision Orbit Determination Standards for the Jason Series of Altimeter Missions. *Marine Geodesy, 33:S1*, 379-418.

Community Coordinated Modeling Center (CCMC) | Goddard Space Flight Center. (2016). *Modelweb*. (NASA) Retrieved 12 14, 2016, from FTP Directory: ftp://hanna.ccmc.gsfc.nasa.gov/pub/modelweb/

Danish Space Research Institute. (2016, 10 22). Retrieved from https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwjxqZaNu47PAhUDCBoKHejUDAwQFghLMAY&url=http%3A%2F%2Fwww.space.aau.dk%2Fcubesat%2Fdocuments%2FMission_Analysis1.pdf&usg=AFQjCNFyqMhvr98DfYIwmoTKqtuZHVv1Zg&sig2=lwxPsmgLS1YFkbmDvpf3

de Bruijn, F. J. (2017). *Guidance Control and Dynamics of a New Generation of Geostationary Satellites.* Delft: TU Delft.

de Pater, I., & Lissauer, J. J. (2007). *Planetary Sciences.* Cambridge: Cambridge University Press.

DeMars, K., Jah, M., Giza, D., & Kelecy, T. (2010). *Orbit Determination Performance Improvements For High Area-To-Mass Ratio Space Object Tracking Using An Adaptive Gaussian Mixtures Estimation Algorithm.* Austin: University of Texas.

Dershowitz, & Reingold. (2007). *Calendrial Calculations.* Cambridge: Cambridge University Press.

DLR. (2017, 09 6). *Raumflugbetrieb und Astronautentraining*. Retrieved from GPS High Precision Orbit Determination Software Tools (GHOST): http://www.dlr.de/rb/desktopdefault.aspx/tabid-10749/10536_read-23371/

Dow, J. M., Neilan, R. E., Weber, R., & Gendt, G. (2007). Galileo and the IGS: Taking advantage of multiple GNSS constellations. *Advances in Space Research, 39*, 1.545-1.551.

ESA. (2017, 07 24). *GOCE*. Retrieved from Introducing GOCE: http://www.esa.int/Our_Activities/Observing_the_Earth/GOCE/Introducing_GOCE

ESOC. (2008). *ESA Bulletin 134.* ESA.

Feulner, G., & Kienert, H. (2014). Climate simulations of Neoproterozoic snowball Earth events: Similar critical carbon dioxide levels for the Sturtian and Marinoan glaciations. *Earth and Planetary Science Letters, 404*, 200-205.

Fu, L. L., Christensen, E. J., Yamarone, C. A., Lefebvre, M., Ménard, Y., Dorrer, M., & Escudier, P. (1994). TOPEX/POSEIDON Mission Overview. *Journal of Geophysical Research, 99*(C12), 24.369-24.381.

Geebelen, K., Wagner, A., Gros, S., Swevers, J., & Diehl, M. (2017). *Moving Horizon Estimation with a Huber Penalry Function for Robust Pose Estimation of Tethered Airplanes.* Leuven: Faculty of Mechanical Engineering.

GMAT. (2016). *User's Guide*. (GMAT) Retrieved 10 25, 2016, from http://gmat.sourceforge.net/doc/nightly/help.html

Grant, M. C., & Boyd, S. P. (2017). *The CVX Users' Guide.* CVX Research, Inc.

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *American Society for Quality, 11*(1), 1-21. Retrieved from http://www.jstor.org/stable/1266761 .

Hilla, S. (2016, 02 21). Retrieved from The Extended Standard Product 3 Orbit Format: https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwjL25bp0pfVAhVCZ1AKHWbTAAEQFggqMAE&url=ftp%3A%2F%2Fftp.igs.org%2Fpub%2Fdata%2Fformat%2Fsp3d.pdf&usg=AFQjCNFYJ_IY8hHSiN_pqOMGkctatJBG_w

Hindi, H. (2004). *A Tutorial on Convex Optimization.* Boston: American Control Conference.

Huang, Z., & Yuan, H. (2007). Improving the ionospheric delay correction of satellite-based augmentation systems in equatorial regions. *Advances in Space Research, 39*, 1.552-1.558.

IAG. (2009). In H. Drewes, *Geodetic Reference Frames: IAG Symposium Munich, Germany, 9-14 October 2006.* Munich: Springer Science & Bussines Media.

IEEE. (2001). *Genetic Algorithms*. Retrieved 02 11, 2016, from http://www.ewh.ieee.org/soc/es/May2001/14/Begin.htm

IERS. (2010). *IERS Conventions 2010.* Frankfurt am Main: IERS.

IERS. (2016). *IERS Bulletins.* Retrieved 10 30, 2016, from https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html

Jacchia, L. G. (1964). Static Diffusion Models of the Upper Atmosphere with Empirical Temperature Profiles. *170*.

Jäggi, A., Hugentobler, U., Bock, H., & Beutler, G. (2007). Precise orbit determination for GRACE using undifferenced or doubly differenced GPS data. *Advances in Space Research, 39*, 1.612-1.619.

Kalra, M., & Singh, S. (2015). *A Review of Metaheuristic scheduling techniques in cloud computing.* Cairo: Egyptian Informatics Journal.

Lemoine, J. M., Bruinsma, S., Loyer, S., Biancale, R., Marty, J. C., Perosanz, F., & Balmino, G. (2007). Temporal gravity field models inferred from GRACE data. *Advances in Space Research, 39*, 1620-1629.

Liu, X. (2013). *Autonomous Trajectory Planning by Convex Optimization.* Ames, Iowa: Iowa State University.

Lowrie, W. (n.d.). *Fundamentals of Geophysics.* Cambridge: Cambridge University Press.

Lucchesi, D. M. (2007). The LAGEOS satellites orbital residuals determination and the way to extract gravitational and non-gravitational unmodeled perturbing effects. *Advances in Space Research, 39*, 1559-1575.

Luthcke, S. B., Zelensky, N. P., Rowlands, D. D., Lemoine, F. G., & Williams, T. A. (2010). The 1-Centimeter Orbit: Jason-1 Precision Orbit Determination Using GPS, SLR, DORIS, and Altimeter Data Special Issue: Jason-1 Calibration/Validation. *Marine Geodesy, 26:3-4*, 399-421.

Marquardt, D. W. (1963). An Algorithm for LSQ Estimation of Non-Linear Parameters. *Journal of the Society for Industrial and Applied Mathematics, 11*, 431-441.

Mathematics Source Library. (2016). *Verner's 8th and 9th Order Embedded Runge-Kutta Method*. Retrieved 10 25, 2016, from http://www.mymathlib.com/diffeq/embedded_runge_kutta/embedded_verner_8_9.html

Montenbrück, O., & Gill, E. (2000). *Satellite Orbits.* Berlin: Springer.

Montenbruck, O., Helleputte, T., Kroes, R., & Gill, E. (2005). Reduced dynamic orbit determination using GPS code and Carrier Measurements. *Science and Technology, 9*, 261-271.

NASA. (2016). *NASA*. Retrieved 10 22, 2016, from Hubbles finds universe is expanding faster than expected: https://www.nasa.gov/feature/goddard/2016/nasa-s-hubble-finds-universe-is-expanding-faster-than-expected

National Geospatial-Intelligence Agency. (2016). *Earth Gravitational Model 2008*. Retrieved 10 22, 2016, from http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/

NorthWest Research Associates, Inc. (2016). *10.7cm Solar Radio Flux*. Retrieved 11 1, 2016, from https://spawx.nwra.com/spawx/f10.html

NOVAS. (2011). *NOVAS Guide.* USNO.

NRL. (2016). *NRLMSISE-00: A New Empirical Model of the Atmosphere*. Retrieved 11 2, 2016, from https://www.nrl.navy.mil/research/nrl-review/2003/atmospheric-science/picone/

OHB SE. (2015). Retrieved 10 22, 2016, from OHB SE "At a Glance": https://www.ohb-system.de/tl_files/system/pdf/ohb_gb_2015_e.pdf

Owen, A. B. (2006). *A Robust Hybrid of Lasso and Ridge Regression.* Stanford: Stanford University. Retrieved from http://statweb.stanford.edu/%7Eowen/reports/hhu.pdf

Pavlis, N. K., Holmes, S. A., Kenyon, S. C., & Factor, J. K. (2012). The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). *Journal of Geophysical Research, 117*.

Pearlman, M. R., Degnan, J. J., & Bosworth, J. (2002). The internation laser ranging service. *Advances in Space Research, 30*, 135-143.

Peñarroya, P. (2017). *Development of a Tool for Precise Orbit Propagation.* LR. TU Delft.

Sergey, K., Sergey, R., & Suriya, T. (2007). GLONASS as a key element of the Russian Positioning Service. *Advances in Space Research, 39*, 1.539-1.544.

Sorenson, H. W. (1970). Least-squares estimation: From Gauss to Kalman. *IEEE Spectrum*, 63-68.

Space Weather Live. (2016). *The Ap-index*. Retrieved 11 2, 2016, from https://www.spaceweatherlive.com/en/help/the-ap-index

Springer, T. A., Gendt, G., & Dow, J. M. (2007). The International GNSS Service (IGS): Perpectives and visions for 2010 and beyond. Darmstadt, Germany.

Sun, W. (2004). Wei Sun. International Astronautical Federation.

Svehla, D., & Rothacher, M. (2003). Kinematic and reduced-dynamic precise orbit determination of low earth orbiters. *Advances in Geosciences, 1*, 47-56.

Tapley, B. D., Ries, J. C., Davis, G. W., Eanes, R. J., Schutz, B. E., Shum, C. K., . . . Zelensky, N. P. (1994). Precision Orbit Determination for TOPEX/POSEIDON. *Journal of Geophysical Research, 99*(C12), 24.383-24.404.

Tapley, B. D., Schutz, B. E., & Born, G. H. (2004). Differenced Altimeter Data. In F. Cynar (Ed.), *Statistical Orbit Determination* (pp. 145-147). Elsevier.

Tapley, B. D., Schutz, B. E., & Born, G. H. (2004). *Statistical Orbit Determination.* San Diego: Elsevier.

Taylor, J. R. (2005). *Classical Mechanics.* University Science Books.

TU Delft. (2016). *TUDAT*. Retrieved 11 2, 2016, from http://tudat.tudelft.nl/projects/tudat/wiki

Visser, H. (2004). *Aircraft Performance Optimization.* Delft: TU Delft.

Visser, P. d. (2017). *Experimental Design & Data Analysis Space Topic.* Delft: TU Delft.

Visser, P. N. (2007). GOCE gradiometer validation by GPS. *Advances in Space Research, 39*, 1630-1637.

Visser, P., & Scharroo, R. (1998). Precise orbit determination and gravity field improvement for the ERS satellites. *Journal of Geophysical Reseach, 103*(C4), 8113-8127.

Wakker, K. F. (2015). *Fundamentals of Astrodynamics.* Delft: TU Delft.

Webb, F., & Zumberge, J. (1995). *An Introduction to GIPSY-OASIS II.* Pasadena: Jet Propulsion Laboratory. doi:JPLM D-11088

Weber, R., Slater, J. A., Fragner, E., & et al. (2005). Precise GLONASS orbit determination within IGS/IGLOS pilot project. *Advances in Space Research, 36*, 369-375.

Wertz, J. R. (2009). *Orbit & Constellation Design & Management.* Hawthorne: Springer NY.

Xinfu, L., Shen, Z., & Lu, P. (2015). Entry Trajectory Optimization by Second-Order Cone Programming. *Journal of Guidance Control and Dynamics*.

Xu, G. (2008). *Orbits.* Berlin: Springer.

Zhang, Q., Moore, P., Hanley, J., & Martin, S. (2007). Auto-BAHN: Software for near real-time GPS orbit and Clock computations. *Advances in Space Research, 39*, 1.531-1.538.

8 Annexes

8.1 Glossary

- **Cost Function**: mathematical function used to obtain the solution of an optimisation problem. See Section 2.5.

- **Estimated Parameters, $\boldsymbol{X}^*(\boldsymbol{t})$**: parameters describing the trajectory generated by running the POP using the outcome of the POD process as initial state, i.e., the product of the POD method. It is modified every iteration using:
$$X^*_{k+1} = X^*_k + x_k$$
where x_k is the x that minimises the cost function.

- **Measurement Errors**: the generated noise to be implemented on top of the generated observations, when needed (it is described in detail in the modelling section). It includes stochastic noise and outliers.

- **Misfit , $\boldsymbol{\rho}$**: the outcome of the linearized observational equation:
$$\rho = Hx - y$$
where y is the difference between the actual and the modelled measurements, i.e., the measurement residual; x is the difference between the current estimation of the parameters and the previous estimation (see *Estimated trajectory*); and
$$\boldsymbol{H} = \begin{pmatrix} H_0 \times \ STM(t0, t0) \\ H_1 \times \ STM(t0, t1) \\ \vdots \\ H_n \times \ STM(t0, tn) \end{pmatrix}$$
is the design matrix that transforms the parameters to be estimated to its corresponding modelled observations. It is composed of H_k, containing the partial derivatives of the measurements w.r.t. the state vector; and $STM(t_0, t)$, the state transition matrix that contains the partial derivatives of the state vector at epoch t_k w.r.t its values at a certain epoch t_0.

- **Noise Layer**: the generated noise to be implemented on top of the reference trajectory. Its nominal definition and parameters are included in Subsection 2.7.1.5.

- **Optimal Value:** value obtained from giving a certain estimation to a penalty function. It is the value that the CVX toolbox minimises in the inner loop (see Section 3.3).

- **Reference Parameters, $\boldsymbol{X}(\boldsymbol{t})$**: parameters describing the trajectory generated by the POP as ground truth. When real data is used, this term refers to the RDOD solution.

- **Robustness**: the capacity of a certain function or algorithm to reach a valid solution under constrained, disadvantageous, or uncertain conditions.

- **Test Configuration**: set of settings imposed for a test, i.e., the penalty function to be used, the magnitude of the noise, etc.