Politecnico di Torino

Department of Mechanical and Aerospace Engineering
Master's Degree in Space Engineering

Implementation of a Sequential Quadratic Programming Algorithm for GEO Satellites Relocation

Manoeuvres Optimisation

Candidate:
Roberto De Santi

Supervisor:
Prof. Lorenzo Casalino

Co-Supervisor:
GMV | Innovating Solutions
Ing. Diego Carlos Pallares López

A.A. 2023-2024

Acknowledgement

I infinitely and deeply thank my *Mom* and *Dad* for everything they have done for me over these years, helping and supporting me in every possible way. A special thanks to my *brother Stefano*, for his constant presence in the most important moments and his wise guidance.

I would like to thank *Prof. Lorenzo Casalino* for his great availability and his precious advices, which have allowed me to write and improve this document.

A heartfelt thank you to *Diego Carlos Pallares López* for his support and invaluable guidance, without which I would not have been able to accomplish this work.

Lastly, let me emphasize my gratitude to my *Mom* for the trust she has always placed in me, without which I am certain I would not have become an engineer.

Contents

List of Figures

List of Tables

Introduction

The aim of this thesis is to describe the physical and mathematical aspects of optimising tangential manoeuvres for *longitude relocation* operations, which are intended to reposition a constellation of communication satellites in *Geostationary Orbit* (GEO). The optimisation process is conducted by harnessing the capabilities of the *NLPQLP* solver, which consists of a *Fortran* subroutine.

Space-based technologies play an increasingly pivotal role in our interconnected world, as the need for global surveillance and continuous observation of extensive areas of the Earth becomes more and more imperative. In this context, geostationary satellites are confirming their role as a crucial element for space-based services. Their characteristic orbit allows these spacecraft to remain consistently positioned above a fixed point on the Earth's surface, providing a stable platform for a variety of applications. Longitude relocation is a frequent operational activity required for GEO satellites, to decrease or increase their altitude and start drifting towards a new longitude in order to offer new services in a different Earth location. Even though this process is well-known to satellite operators, its optimisation is not always straightforward due to the multiple constraints that must be satisfied, particularly when considering the low-thrust performances of the electric thrusters widely used in recent missions. The optimisation of the manoeuvres required to carry out the relocation of GEO satellites holds paramount importance in order to limit the operational and management costs of the mission. This thesis investigates a novel approach for GEO relocation optimisation by implementing a *sequential quadratic programming* (SQP) algorithm for constrained nonlinear optimisation problems. The primary focus will be directed towards the study of continuous manoeuvres, thus considering the use of electric thrusters. This enhanced approach incorporates a quasi-Newton method to solve a set of variables, aiming to achieve the desired orbit targets and obtain the best feasible manoeuvre plans. From the operational standpoint, configurable and user-defined constraints are introduced. The solver's flexibility enables the customization of constraints for the different phases of insertion and extraction, executed both through electric or chemical propulsion. Specifically, this versatility of the method has been corroborated in the work conducted, with its broadening to impulsive manoeuvres, resulting in excellent results.

Firstly, a brief overview of geostationary satellites and their main characteristics will be provided. Secondly, key concepts related to orbital mechanics and space propulsion will be reviewed to better understand the analyzed problem. Then, a general mathematical description of the SQP optimisation algorithm will be given. Finally, the methodology applied for the study of the problem will be presented, followed by the exposition of the principal results obtained.

Chapter 1

Geostationary Satellites

"The more I read, the more I acquire, the more certain I am that I know nothing"
— Voltaire

Since this work is related to the relocation operation of geostationary satellites, it is useful to begin with a general introduction about these spacecraft. This first chapter will present the main characteristics of GEO satellites, introducing the features of their peculiar orbit without, however, delving into the technical details that define it. Additionally, the principal applications of geostationary spacecraft will be illustrated here, along with a brief overview of the aspects concerning their implementation.

A geosynchronous satellite is a spacecraft with an orbital period equal to the Earth's rotation period: 23h 56' 4". A special case of a geosynchronous satellite is the geostationary one, which is characterized by a circular geosynchronous orbit directly above the Earth's equator. Such synchronization uniquely determines the orbit height; thus, all GEO satellites share the same altitude. This particular characteristic will be discussed more thoroughly in Chapter 2, where it will also be demonstrated from a technical point of view.

The concept of a geostationary orbit was popularized in the 1940s as a way to revolutionize telecommunications, while the first satellite to be placed in this kind of orbit was launched in 1963. Since this inaugural launch, the number of GEO spacecraft has steadily increased, reaching more than one hundred in 1980 and two hundred in 1990. Nowadays, they represent approximately 12% of satellites in orbit, a significant percentage considering the high cost required to reach the GEO orbit, due to its significant altitude of nearly 42,000 km.

1.1 Overview

Geostationary satellites, during their motion, follow the direction of Earth's rotation —eastward— and have the unique property of remaining permanently fixed in exactly the same position in the sky as viewed from an observer on Earth. The primary benefit of this orbital configuration lies in the continuous communication link between the ground station and the spacecraft. Additionally, for Earth observation purposes, it enables the consistent monitoring of a specific geographical area. Since the GEO orbit is characterized by a very high altitude, resulting in a very large visibility cone, a single satellite can cover an area close to 30% of the Earth's surface. In the specific, with a constellation of just 3 satellites, it is generally possible to achieve global coverage of the Earth, except for polar regions. It is easier to understand this characteristic looking at Fig. 1.1, which provides a schematic representation. It is worth highlighting that a perfectly geostationary orbit is a mathematical abstraction achievable only by a spacecraft orbiting around a perfectly symmetric Earth, with no additional forces exerted on it apart from the gravitational attraction from Earth. However, a spacecraft is actually subject to various forces that influence and modify its orbit. These forces, known as *non-Keplerian forces* or *perturbations* in celestial mechanics, arise from different factors, such as the gravitational influence of the Moon and the Sun or the non-sphericity of the Earth. Further specific details of these perturbations are provided in Chapter 2.

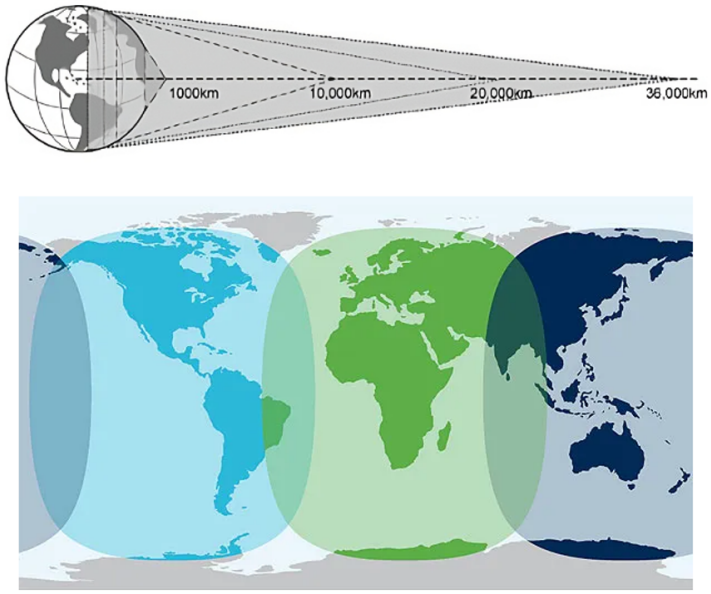

Figure 1.1: Satellite altitude versus visibility cone (above) and coverage of three GEO satellites (below).

1.1.1 Eclipses

Geostationary satellites are typically equipped with solar panels, which generate the electrical power needed to operate the mission payload and onboard electronics. Furthermore, in recent GEO missions, there has been an increasing use of electric propulsion systems, where the electrical energy provided by the solar cells is used to generate acceleration power. It goes without saying that appropriate and periodic exposure to sunlight during the day is crucial for the correct functioning of the satellite. Therefore, the eclipse period represents challenging phases during which the solar panels are unable to produce energy, leaving the satellite reliant solely on the energy previously stored in its onboard batteries. Eclipses are generally a nuisance to the mission, as they cause fluctuations in onboard temperatures and the loss of Sun reference direction. Furthermore, manoeuvres are often prohibited in this time interval and, depending on the spacecraft's design, various on board devices must be switched off during the eclipse.

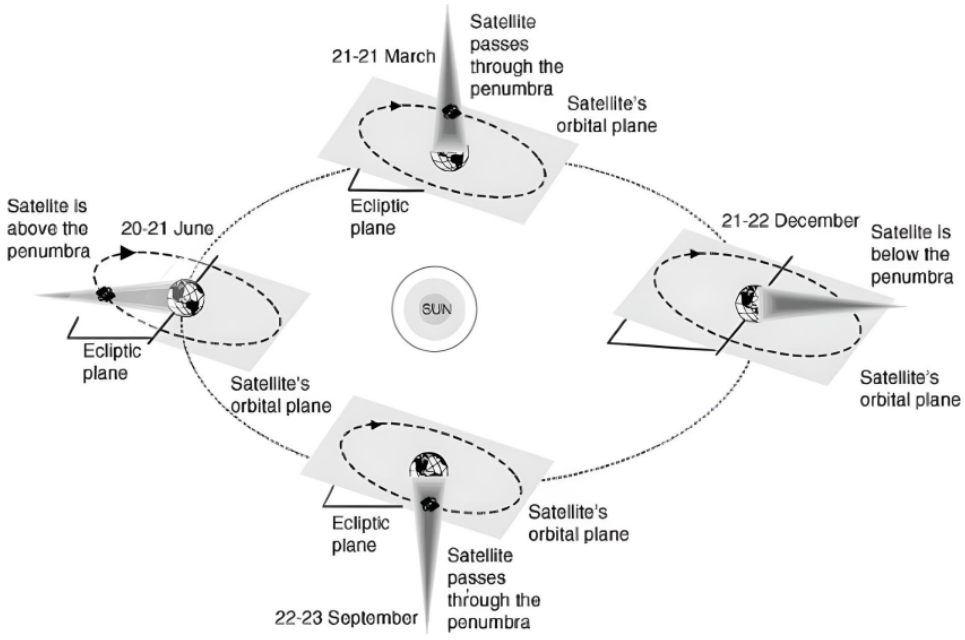

Figure 1.2: Schematic overview of the eclipses of the Sun by the Earth experienced by a satellite during the year

1.1.1.1 Eclipse by Earth

Near the winter and summer solstices—occurring on December 21th and June 21th respectively—a GEO satellite, due to its orbit, is never in shadow or penumbra, remaining continuously illuminated by sunlight. Conversely, near the spring and autumn equinoxes—occurring on March 21th and September 21th respectively—it enters in the the eclipse season. During these periods, the Sun moves through the equatorial plane, which is reasonably close to the orbital plane. Consequently, the spacecraft passes through the Earth's shadow once per day, becoming aligned with the Earth

and the Sun. This phenomenon is known as eclipse of the Sun by the Earth, and each eclipse season lasts for about seven weeks — specifically for 46 days, from February 26 to April 13 and from August 31 to October 16—. Before and after entering the shadow, there exists a singular transition through the penumbra, the duration of which varies depending on the season. This penumbra region occurs due to the nature of the light source, the Sun. As can be inferred from Fig. 1.3, the time interval during which the satellite is in shadow reaches a maximum of 71.5 minutes on the equinox day and decreases in the days before and after it. When the inclination is not zero, both the timing of the eclipse seasons and their overall duration are altered, although the length of the longest eclipse remains constant.

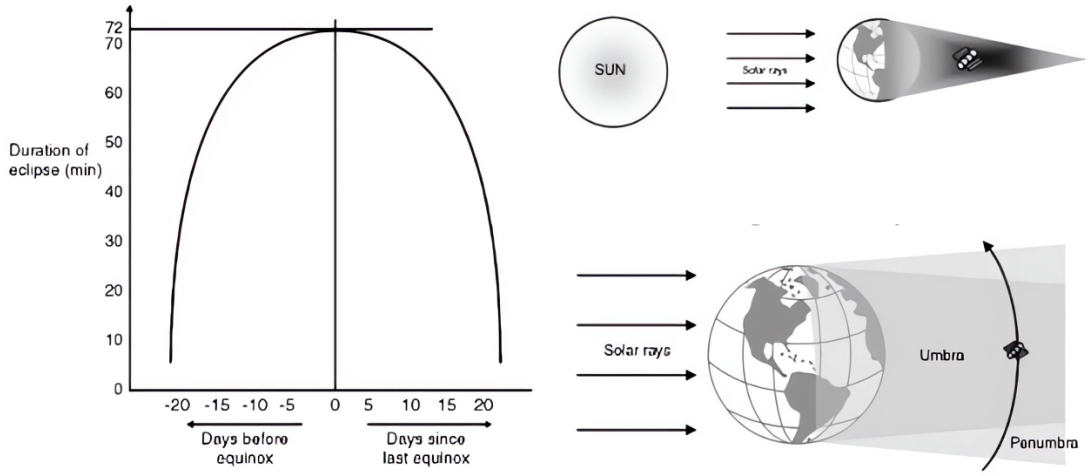

Figure 1.3: Duration of the eclipse in relation to the days around the equinox (left); umbra and penumbra region (right) [1]

1.1.1.2 Eclipse by Moon

The eclipses of the Sun by the Moon, as experienced by a spacecraft, follow similar principles to those perceived by an observer on Earth. However, there are some distinctions arising from the spacecraft's higher velocity in comparison to a ground observer. Consequently, a geostationary satellite encounters solar eclipses more frequently and for longer durations than those witnessed from the Earth's surface. Specifically, the nature and duration of these eclipses are strongly dependent on the Moon's location with respect to the satellite's position along its orbit. One can consider the mean value of the orbital period of the Moon to be 27.3 days, and its flight velocity to be approximately 1 km/s. The velocity of a geostationary spacecraft, which is approximately 3 km/s, allows it to surpass the Moon's shadow when both are moving in the same direction. This scenario arises when the spacecraft and the Moon are positioned on the same side of the Earth. Conversely, a quarter of a day earlier or later, the satellite moves perpendicular to the Moon's shadow, enabling the shadow

to catch up with the spacecraft. As a result, the relative motion between the satellite and the Moon's shadow may lead to the spacecraft experiencing extended periods of eclipse or encountering two eclipses with only a short interval between them outside the shadow. This phenomenon is clearly not possible for a ground-based observer due to the lower rotation velocity of the Earth, with its highest value being 0.46 km/s at the equator. A visual representation of the aforementioned concept is depicted in Fig. 1.4. In general, eclipses caused by the Moon exhibit a distribution characterized by a seemingly irregular pattern, with a variable number of occurrences per year. It is evident that calculating these phenomena results in a significant challenge for astronomers, given the high complexity involved. The description provided applies to both the umbra and penumbra scenarios. Specifically, surrounding the umbra region, there is the wider cone of the penumbra, with a cross-sectional radius measuring approximately 3500 km.

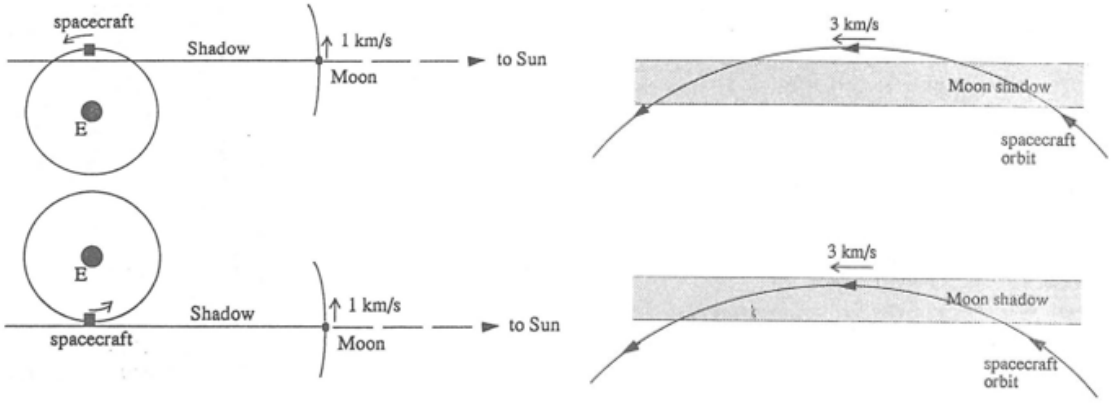

Figure 1.4: Two closely spaced eclipses (top) or one long eclipse (bottom), which may occur when the spacecraft flight direction is approximately aligned with the Moon's shadow [2]

1.1.2 Applications

As previously mentioned, an object in a GEO orbit appears motionless to ground observers, remaining fixed over one spot above the equator. Hence, this characteristic proves favorable for certain technical applications and is therefore exploited for various purposes. Moreover, other advantages arise from the peculiar features of this orbit, as it allows for high temporal resolution data and simplifies the tracking of the satellite by its earth stations.

Geostationary satellites have had a significant impact on global communications, television broadcasting, and weather forecasting, while also finding applications for a number of important defense and intelligence objectives. Nowadays, the geostationary orbit is primarily utilized for communications missions, enabling ground stations to employ fixed-direction antennas. This simplification eliminates the need for Earth-based satellite antennas to rotate for tracking, as they can be permanently pointed at the spacecraft's position in the sky, thereby reducing operational costs.

Other applications include Earth observation missions; for example, weather satellites are often placed in this orbit for real-time monitoring and data collection. Additionally, the GEO orbit is utilized for navigation satellites to provide a known calibration point and enhance GPS accuracy, as well as in a few scientific missions. However, there are also some disadvantages associated with this specific orbit. Due to its high altitude, radio signals take a significant amount of time to reach and return from the satellite, resulting in a small but noticeable signal delay. Furthermore, as mentioned earlier in this paragraph, geostationary satellites do not provide complete geographical coverage, leaving regions at higher latitudes uncovered. Consequently, ground stations at latitudes higher than 60° may have difficulty reliably receiving signals.

Further information about some of the main applications just discussed, and their specific benefits and complications derived from employing a geostationary orbit, are summarized as follows:

- **Communication**

  Geostationary communication satellites are particularly useful due to their high visibility from a large area of the Earth's surface, extending 81° away in latitude and 77° in longitude. Since they appear stationary in the sky, as mentioned previously, movable antennas are not necessary. This allows for the implementation of small and less expensive stationary antennas on Earth. However, the major drawback is the significant latency for signal transmission. This delay poses challenges for latency-sensitive applications such as voice communication, making geostationary satellites primarily suitable for unidirectional entertainment and applications where low-latency alternatives are unavailable. It is worth highlighting that as the observer's latitude increases, communication becomes more difficult due to various factors such as atmospheric refraction, Earth's thermal emission, and line-of-sight obstructions.

- **Observation**

  In this type of mission, GEO satellites are typically employed to capture images in the visual and infrared spectrum of Earth's surface, atmosphere, and oceans. The main objectives involve weather observation, oceanography, and atmospheric tracking. Specifically, geostationary meteorological satellites are utilized for tracking volcanic ash, estimating vegetation coverage, and measuring temperatures of clouds, land, and oceans. They also provide crucial information for creating meteorological prediction models. Nevertheless, the high altitude of their orbit compromises the achievable resolution. Due to their wide field of view and low resolution, they are primarily implemented for short-term and real-time forecasting.

- **Navigation**

  Geostationary satellites can be used to enhance GNSS systems by providing an additional reference signal, consequently leading to a significant improvement in position accuracy. However, it should be noted that for navigation purposes, MEO orbits are typically preferred over GEO orbits due to their better accuracy and lower latency. Moreover, the geostationary orbit does not offer particularly significant advantages for this kind of missions.

1.2 Implementation

Geostationary satellites are launched eastward into a prograde orbit that matches the rotation rate of the Earth at the equator, and they are positioned in a specific slot above a designated point on the Earth's surface. Various actions are necessary to reach the final orbit and ensure the satellite's correct operability. Additionally, several interventions and manoeuvres are required to maintain the satellite's trajectory, make any necessary changes, and ultimately dispose of it at the end of its operational life. However, for a concise and straightforward overview of GEO spacecraft implementation, four main phases can be identified:

- *Launch*
  As previously mentioned, the satellite is launched eastward. Specifically, the launcher releases the spacecraft at an altitude of about 200 km above the Earth, placing it into a geostationary transfer orbit (GTO). The GTO is an elliptical orbit characterized by a low-altitude perigee and an apogee near the geostationary height. The onboard propulsion system is then utilized to raise the orbit's perigee, circularize it, and ultimately reach the GEO orbit. During this process, the *apogee motor* (AFM) boosts the flight velocity and alters the spacecraft direction, thereby adjusting the transfer orbit's inclination from approximately 7° to nearly 0°. Launching a geostationary satellite from a low-latitude site minimizes the amount of inclination change required in this manoeuvre, thus reducing the associated costs. Additionally, launching close to the equator allows for greater exploitation of the tangential velocity caused by Earth's rotation, which is maximized at the equator. However, the selection of the launch site is subject to additional requirements common to all space launches, which arise from other issues such as safety considerations.

- *Station acquisition*
  This phase begins after the AFM firing. During station acquisition, a sequence of smaller orbit manoeuvres is executed over a span of up to one month to shift the spacecraft to the intended longitude and adjust the orbit's eccentricity and inclination. Specifically, these manoeuvres serve to rectify any discrepancies arising from the AFM burn and residual errors stemming from the launcher injection.

- *Operational phase*
  After an initial period, known as the commissioning phase, during which tests and checks are conducted on the satellite to ensure all systems function correctly, the satellite can begin its operational life. The routine operations of the mission typically endure for several years until one of several conditions is met: depletion of onboard fuel, degradation of electric power generators, occurrence of a significant onboard error, or obsolescence of the mission.

- *Disposal*
  At the conclusion of the mission, aging GEO spacecraft are commonly transitioned into a circular orbit situated a few hundred kilometers above the geostationary altitude. This measure is undertaken to reduce the risk of potential collisions. Following this phase, ground control over the satellite is terminated, allowing it to drift indefinitely in what is colloquially known as a *graveyard orbit*.

1.2.1 Manoeuvre Types

Geostationary spacecraft are outfitted with thrusters capable of being activated either through ground command or an automatic onboard system to adjust the orbit as per specific requirements. These requirements typically align broadly with the phases observed earlier:

- *Station acquisition*: manoeuvres aimed at positioning the spacecraft into the desired geostationary orbit at the mission's outset.
- *Station keeping*: Regular manoeuvres performed throughout the mission's lifespan to counteract external perturbations that may alter the orbit from its geostationary position.
- *Station shifts*: manoeuvres executed to alter the satellite's longitude, if needed by the mission's objectives.
- *Re-orbiting*: Actions taken at the mission's conclusion to transition the spacecraft out of the geostationary orbit, achieved by adjusting its altitude up or down by several hundred kilometers from the nominal geostationary radius.

1.2.2 Legal Aspects

As mentioned at the beginning of the chapter, the geostationary orbit belt is defined by the distance from the Earth, which must be near the geostationary radius, and the latitude relative to the Earth's equator, which should approximately be zero. It goes without saying that this orbit represents a very limited region of space. The ring-shaped region outlined has only one free dimension to allocate to different spacecraft, namely the longitude[1] relative to the Earth.

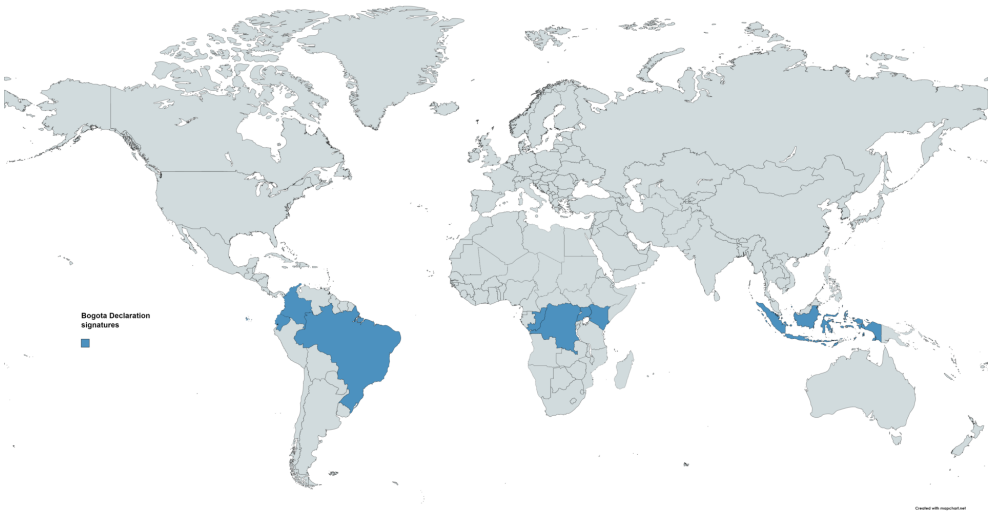

Figure 1.5: The eight signatories to the 1976 Bogota Declaration

[1]The longitude of the projection of the spacecraft on the Earth's surface

The allocation of longitude positions is primarily performed to avoid signal interference between neighboring spacecraft that utilize the same radio frequency. In light of the above, there are a limited number of slots available in this orbit, and thus only a limited number of satellites can be operated in it.

Nowadays, the allocation of frequencies for terrestrial radio communications is coordinated by the International Telecommunication Union (ITU). The ITU's allocation mechanism is also responsible for resolving any disputes regarding access to the GEO orbital slots.

As a matter of fact, the exploitation of the geostationary orbit has also led, over time, to conflicts between different countries. Many nations, even those without access to space technology, requested longitude positions for potential future use out of fear of losing access to this crucial resource. Additionally, a group of eight equatorial nations claimed sovereignty over the geostationary orbit above their territories. These requests culminated in the 1976 Bogota Declaration, but the claims did not gain international recognition and received no positive response from space-faring states. These events clearly demonstrate the importance of the geostationary orbit, which, due to its unique characteristics, can be likened to a natural resource essential for various human activities.

Chapter 2

Orbital Mechanics

"Eppur si move!"
— Galileo Galilei

Before delving into the specifics of the work conducted, it proves beneficial to provide a brief overview of foundational notions and physical aspects related to orbital mechanics. Therefore, this Chapter introduces essential concepts to enhance the understanding of the analyzed problem, these fundamentals will be deemed assimilated through the exposition of the thesis.

Firstly, the two-body problem will be presented, starting from Kepler's laws and Newton's universal law of gravitation, and leading to its solution and the equations that characterize it. Then, the main methods of parameterizing an orbit will be discussed. Finally, the typical physical and technical characteristics of the geostationary orbit will be examined.

2.1 Two-Body Problem

The initial stage in investigating any issue related to celestial mechanics is usually represented by the *Two-Body Problem* (2BP). As the name suggests, it involves studying the motion of two celestial bodies, considering their mutual gravitational influence as the sole interaction between them. Within the simplified framework of the Two-Body Problem, the first fundamental contribution to the mechanics of spaceflight is represented by *Kepler's Three Laws*. These laws, formulated purely empirically by the physicist Johannes Kepler between 1609-1620, assert that:

1. *The orbit of a planet is elliptical and lies in a plane, with the Sun occupying one of the foci of this ellipse.*

2. *The radius vector connecting the planet and the Sun sweeps out equal areas in equal times.*

3. *The square of the period of revolution of a planet is proportional to the cube of the distance from the Sun.*

Sinoptically, the first law thus consisted of proposing a heliocentric model in which orbits are not circular but elliptical. Meanwhile, the second law stems from the conservation of orbital angular momentum.

The second pillar of orbital mechanics is instead represented by Newton's contribution. In 1689, in the *'Philosophiae Naturalis Principia Mathematica'*, Newton analytically demonstrated the three aforementioned laws and formulated the *Law of Universal Gravitation*. This law states that in the Universe, two material bodies exert mutually an attractive and central force directly proportional to the product of their masses and inversely proportional to the square of their distance:

$$\boldsymbol{F} = -G\frac{mM}{r^2}\frac{\boldsymbol{r}}{r} \tag{2.1}$$

where $G = 6.67 \cdot 10^{-11}\,\frac{m^3}{kgs^2}$ is the *gravitational constant*, while r is the is the radius vector equal to the distance between mass M and m. It is customary, for simplicity in the case of bodies orbiting around Earth, such as satellites, to substitute the gravitational parameter $\mu = GM$ within the equation. In the case of the Earth: $\mu = 3.986 \cdot 10^5\,\frac{km^3}{s^2}$. The same can be applied to rotating bodies around other celestial objects by substituting the appropriate gravitational parameter.

Kepler and Newton's work represent the foundations of orbital mechanics and find application in the study of the orbit of any celestial body. Although enunciated with reference to the motion of celestial bodies such as planets and stars, they clearly hold validity for the study of spacecraft motion as well.

As previously mentioned, in 2BP, the motion of two bodies is studied under the influence of their mutual gravitational interaction as the only force acting on them. An additional simplification to this problem is the so-called *Restricted Two-Body Problem* (R2BP), where one of the two bodies is considered to have a significantly smaller mass than the other. The assumptions underlying this model are:

- Distribution of the mass *homogeneous* and *spherical*
- *Punctiform masses* concentrated in the centres of the bodies
- $m \ll M$
- Only gravitational forces (the influence of third bodies is neglected)

In the R2BP, the secondary body, in this case the spacecraft, is assumed to experience the exclusive gravitational attraction of a primary body. Thanks to the simplifications made, it is possible to derive analytically the equation that governs its motion.

Starting from equation (2.1) and under the assumptions of the previously presented Restricted Two-Body Problem, can be obtained:

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^2}\frac{\boldsymbol{r}}{r} \tag{2.2}$$

This formula describes the motion of the secondary body relative to the primary body and represents a second-order vector equation.

2.1.1 Constants of Motion

In the R2BP, there exist important physical properties arising from specific laws of conservation in dynamic systems. Consequently, there are quantities that remain constant over time during the evolution of the system. In the case of satellites, two fundamental constants of motion are defined, which remain unchanged during the orbit of the body:

- *Angular Momentum*

$$\boldsymbol{h} = \boldsymbol{r} \wedge \boldsymbol{v} = \boldsymbol{r} \wedge \dot{\boldsymbol{r}} \tag{2.3}$$

  In the context of the R2BP, the trajectory is planar, meaning it lies within a plane. Since r and v define the orbital plane, h must be always perpendicular to it.

- *Specific Mechanic Energy*

$$E_g = \frac{v^2}{2} - \frac{\mu}{r} \tag{2.4}$$

  Where the first term represents kinetic energy, while the second term represents the potential energy associated with the Earth's gravitational field.

2.1.2 Trajectory Equation

The equation of the trajectory in the R2BP describes the locus of points occupied by the secondary body during its orbit. By solving the equation of motion (2.2) and through several steps not reported here for simplicity, one can obtain it:

$$r = \frac{h^2/\mu}{1 + B/\mu \cos \nu} \tag{2.5}$$

Where ν is the true anomaly which defines the position of the satellite in its orbit. And it has been assumed that B is a constant phase-shifted by ν from r. Through considerations of a geometric nature, equation (2.5) can be written as follows:

$$r = \frac{p}{1 + e \cos \nu} \tag{2.6}$$

Where p is the *semilatus rectum* of the orbit and e the eccentricity, two important parameters that define the shape of the trajectory. The formula (2.6) describes the potential orbits of a satellite, which are defined by the intersection between a plane and a cone, also known as *conic sections*. The ellipse is, in fact, just one of the possible trajectories that a body can have. Specifically, a satellite can assume a circular, elliptical, parabolic, or hyperbolic orbit, as shown in Fig. 2.1. The solution to the R2BP, also known as the *Kepler Problem*, is an unperturbed orbit referred to as *Kepler Orbit*.

In the case of an elliptical orbit, two important quantities can be defined: r_p and r_a, which are the radii of the periapsis and apoapsis, respectively.

These two points represent the extremes of the ellipse along the major axis and can be obtained from equation (2.6):

$$\begin{cases} r_p = \dfrac{p}{1 + e\cos 0} = \dfrac{p}{1+e} & \nu = 0 \\ r_a = \dfrac{p}{1 + e\cos \pi} = \dfrac{p}{1-e} & \nu = \pi \end{cases} \tag{2.7}$$

Through geometric considerations, one can then write:

$$p = a\,(1 - e^2) \tag{2.8}$$

Using equations (2.7) and (2.8):

$$\begin{cases} r_p = a\,(1-e) \\ r_a = a\,(1+e) \end{cases} \tag{2.9}$$

from which:

$$e = \frac{r_a - r_p}{r_a + r_p} \tag{2.10}$$

By comparing equations (2.5) and (2.6), it can finally be stated for completeness that:

$$\begin{cases} p = \dfrac{h^2}{\mu} \\ e = \dfrac{B}{\mu} \end{cases} \tag{2.11}$$

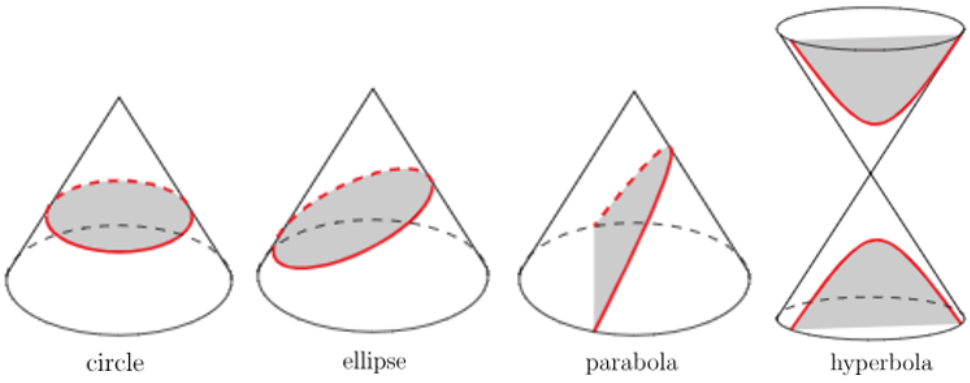

Figure 2.1: Conic sections

2.2 Orbital Elements

A generic Keplerian orbit can be parameterized in various ways; in particular, given an inertial frame of reference and an arbitrary epoch (a specified point in time), it is generally necessary to use a set of six elements. In the context of the Restricted Two-Body Problem, indeed, the orbit and position of a body can be uniquely defined by employing six different parameters, which corresponds to the number of degrees of freedom of the problem.

2.2.1 Keplerian Elements

In orbital mechanics, the *Classical Orbital Elements* or *Keplerian Elements* are a group of six elements commonly used to identify a specific Keplerian orbit. These parameters are:

- *Eccentricity* ($\boldsymbol{e}$)

  It defines the shape of the orbit. In the Tab. 2.1, various values of eccentricity are summarized in relation to the type of trajectory of the body, which, as seen earlier, is represented by one of the conic sections.

- *Semi-major Axis* ($\boldsymbol{a}$)

  It defines the dimension of the orbit and in the specific the semi-major axis is connected to its energy. Indeed, starting from the equation of mechanical energy, through geometric considerations, it can be shown that:

$$E = -\frac{\mu}{2a} \tag{2.12}$$

  Tab. 2.1 shows the different values of a and the corresponding energy E for various types of trajectories.

Orbit	$\boldsymbol{E}$	$\boldsymbol{a}$	$\boldsymbol{e}$
Circle	< 0	> 0	$e = 0$
Ellipse	< 0	> 0	$0 < e < 1$
Parabola	0	∞	$e = 1$
Hyperbola	> 0	< 0	$e > 1$

Table 2.1: Types of orbit as conic sections

- *Inclination* ($\boldsymbol{i}$)

  It is one of the two parameters necessary to describe the orientation of the plane of the orbit. It is the angle between the plane of the orbit and the reference plane (the Earth's equator or the ecliptic).

- *Right Ascension of Ascending Node* ($\boldsymbol{\Omega}$)

  The RAAN is the other parameter necessary to define the orientation and the plane of the orbit. It represents the angle between the *Vernal Point*[1] (γ) and the *ascending node*[2] of the orbit. The RAAN is measured in the equatorial plane; in the case where the reference plane is the ecliptic, it is referred to as *Longitude of the Ascending Node*.

- *Argument of Periapsis* ($\boldsymbol{\omega}$)

  It provides the position of the orbit's periapsis, which is the closest point of the trajectory to the principal body. More precisely ω is the angular distance between the periapsis and the ascending node measured on the orbital plane in the direction of motion.

- *True Anomaly* ($\boldsymbol{\nu}$)

  It identifies the angular position of the secondary body at a given moment along its orbit. It is measured on the orbital plane starting from the periapsis.

While the provided definitions implicitly refer to a celestial body orbiting the Earth, they can be adapted, with appropriate modifications, to describe any orbits around an arbitrary primary body. In Fig. 2.2, the just listed parameters are clearly depicted.

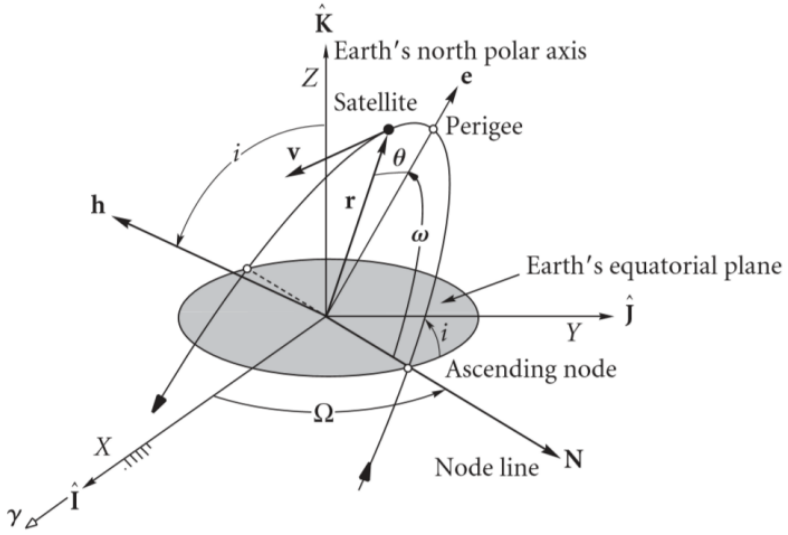

Figure 2.2: Keplerian Orbital Elements [3]

[1]One of the two equinoctial points where the celestial equator intersects the ecliptic, specifically, it is defined in correspondence with the vernal equinox.

[2]The point where a satellite crosses from the southern hemisphere to the northern one. It belongs to the line of nodes, defined by the intersection between the plane of the orbit and the reference plane.

2.2.2 Alternative parameters

In some cases, classical orbital elements are not suitable for defining certain specific orbits. Therefore, the following alternative parameters are introduced:

- *Longitude of Periapsis*: $\boldsymbol{\pi} = \Omega + \omega$

  Which is used for equatorial orbits ($i = 0$), where there is no line of nodes, and Ω is therefore undefined.

- *Argument of Latitude*: $\boldsymbol{u} = \omega + \nu$

  Which is employed for circular orbits ($e = 0$) where ω is undefined, as the periapsis cannot be identified.

- *True Longitude*: $\boldsymbol{l} = \Omega + \omega + \nu$

  It is used in the specific case of orbits with zero eccentricity and inclination ($i = e = 0$), for which it is not possible to define ω and Ω. This applies, for example, to a geostationary orbit.

These parameter are obviously useful for orbits with inclinations or eccentricities close to zero, though not entirely null. It should be emphasized that it is possible to refer to the same parameters listed above using different names depending on the context.

2.2.3 Orbital State Vectors

Given a reference epoch, an alternative method to uniquely determine the trajectory of an orbiting body involves the use of the *Orbital State Vectors*, defined by the vectors of position and velocity expressed in a Cartesian coordinate system:

$$\boldsymbol{x}(t) = [\boldsymbol{r}(t), \boldsymbol{v}(t)]^T = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$$

The state vectors can be formulated in various other ways, not limited to the traditional position-velocity vectors. Other possible representations include the *Two-Line Element Set* (TLE) and the *Vector Covariance Matrix* (VCM), whose description is beyond the scope of this work. It is worth noting the explicit time dependence of the state vectors, which implies the variability of these parameters along the trajectory. While in the R2BP, the classical orbital parameters, with the exception of the true anomaly, remain constant, making them more convenient for representing a Keplerian orbit. Through appropriate transformations, it is possible to obtain Keplerian orbital elements from the state vectors and vice versa.

2.3 Orbital Perturbations

The equation of motion, as defined in equation (2.2), is obviously applicable only under the simplifications of the R2BP. However, in reality, the satellite is subjected to several forces—non-Keplerian forces—that alter its trajectory. The presence of these perturbations renders the expression in equation (2.2) invalid as presented, necessitating appropriate modifications. Therefore, by expressing the

effect of the external forces in the form of perturbation accelerations experienced by the satellite, one can correct the equation of motion as:

$$\ddot{\boldsymbol{r}}' = \ddot{\boldsymbol{r}} + \boldsymbol{a_p} \tag{2.13}$$

where $\ddot{r}$ represents the acceleration described by equation (2.2) under the assumption of the R2BP. Meanwhile, the term a_p denotes the sum of perturbing accelerations acting on the spacecraft, resulting from all forces except the spherically symmetric Earth gravity and the forces generated by the onboard thrusters. It should be noted that the acceleration produced by the propulsion system during manoeuvres constitutes an active orbit control. Therefore, it cannot be considered a perturbation and should be added separately to the motion equation through an appropriate individual term.

The primary non-Keplerian forces to be considered for a geostationary spacecraft are:

- Gravitational attraction arising from the non-spherical component of the Earth's gravitational field.
- Gravitational attraction of the Sun and the Moon treated as point masses.
- Solar radiation pressure.

To these, the effect of non-uniform motion of the coordinate system can also be added. There are also other less relevant disturbances, such as the effect of tides.

The perturbations induced by non-Keplerian forces cause a real orbit to experience variations in its elements over time. These changes are not considered in a Keplerian orbit, which serves as an idealized, mathematical representation of the real orbit at a specific moment.

Therefore, it is customary to provide orbital elements even for perturbed orbits. A common approach is to use the *osculating orbital elements*. These elements are computed using the actual spacecraft position and velocity obtained from the perturbed motion at a given moment, known as the epoch.

Another viable option, which proves advantageous in scenarios involving perturbed orbits, is to employ the arithmetic mean of orbital elements averaged over a specific time interval, for example an entire sidereal day. These elements are known as *mean elements*.

In the following subsections, the effects of the main external perturbations are briefly described from a more technical perspective. While in Fig. 2.3, the primary perturbation effects as a function of the orbit's altitude are illustrated. The graph confirms that the most significant perturbative accelerations acting on a GEO satellite are those caused by the gravitational attraction of the Sun and the Moon, as well as the solar radiation pressure.

2.3.1 Non-spherical Earth Potential

The acceleration due to Earth's gravitational field, as expressed in equation (2.2), assumes the Earth as a uniform-density perfect sphere with all its mass concentrated at its center. However, in reality, Earth is not perfectly spherical, nor is it characterized by homogeneous mass distribution. Therefore, the gravitational attraction cannot be adequately represented as originating from a point mass or a

sphere when describing the motion of spacecraft orbiting around Earth. This results in inevitable perturbations compared to the motion outlined by equation (2.2). Hence, it is necessary to rewrite the potential function associated with Earth's gravitational field in a more realistic and accurate manner:

$$U = \frac{\mu}{r} + \mu \sum_{l=2}^{L} \sum_{m=0}^{l} \frac{R_{\oplus}^{l}}{r^{l+1}} P_{lm}(\sin\theta)(C_{lm}\cos m\lambda + S_{lm}\sin m\lambda) \tag{2.14}$$

where R_{Earth} denotes the Earth's radius, λ and θ represent the satellite's longitude and latitude, respectively. While, P_{lm} are the Legendre polynomial functions of degree l and order m. Moreover, μ/r represents the contribution from the symmetric part of the spherical Earth's gravitational field—corresponding to $l = m = 0$—which determines equation (2.2). The other terms allow us to define the real Earth's potential function, obtained through the series expansion in harmonics. In the context of geostationary orbits, an expansion up to order $L = 8$ can generally be considered adequately precise. The perturbative accelerations resulting from the Earth's non-symmetric attractive field, whose equations are omitted here for brevity, are calculated under considering that the spacecraft's gravitational acceleration is represented by the gradient of a potential function U:

$$\ddot{\boldsymbol{r}} = \nabla U \tag{2.15}$$

The gradient of the term μ/r yields the acceleration obtained in the R2BP, which clearly does not represent a perturbation.

2.3.2 Sun and Moon influence

To describe the perturbations caused by the gravity of the Sun and the Moon, one can utilize the equation describing the gravitational attraction of a *third body* on a satellite orbiting Earth. Given a generic third body, a spacecraft experiences the following acceleration a_k:

$$\boldsymbol{a_k} = \frac{\mu_k}{\boldsymbol{r_k}^3}\left(\frac{3}{\boldsymbol{r_k}^2}(\boldsymbol{r_k}\cdot\boldsymbol{r})\boldsymbol{r_k} - \boldsymbol{r}\right) \tag{2.16}$$

with μ_k the gravitational parameter of the third body and r_k the vector denoting its position respect the center of the coordinate system. Replacing the values related to the Sun and the Moon in the equation just seen, one can obtain their respective perturbative accelerations.

The influence of a third body is considered in the *Three-Body Problem* (3BP), in which the motion of a less massive third body, such as a spacecraft, is influenced concurrently by two primary bodies, for example the Earth and the Moon. The *Restricted Three-Body Problem* (R3BP) simplifies this scenario by considering the spacecraft to have negligible mass. Additionally, the *Circular Restricted Three-Body Problem* (CR3BP) extends this assumption to include circular orbits.

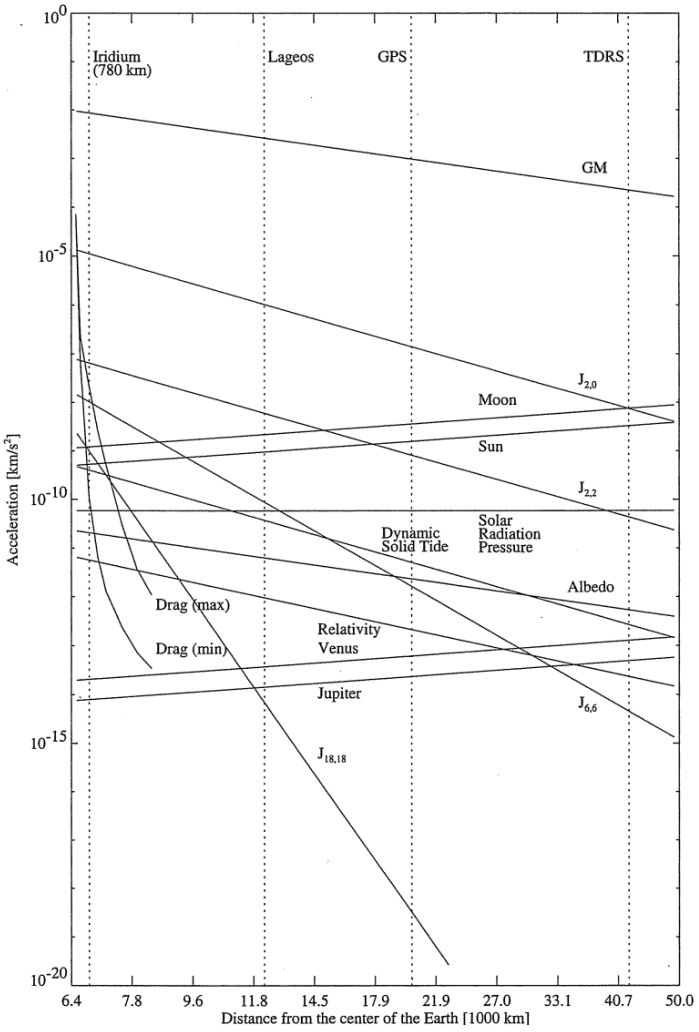

Figure 2.3: Magnitude scale of different perturbations affecting a satellite's orbit [4]

2.3.3 Solar radiation pressure

This perturbation is associated with photons composing the solar flux. Upon impacting the satellite, these particles exchange energy, thereby altering its momentum. The magnitude of the force exerted by solar radiation pressure on the satellite can be expressed as:

$$\boldsymbol{F_{sr}} = \boldsymbol{p_{sr}}(1+\varepsilon)S \tag{2.17}$$

where S is the satellite cross section and ϵ is the reflectivity coefficient of the spacecraft ranging in surface $0 < \varepsilon < 1$; with $P = 4.56 \cdot 10^{-6} N/m^2$ the pressure generated by the solar radiation, in the vicinity of Earth, on an orthogonal surface.

Given the satellite mass m, it is then possible to estimate the perturbative acceleration a_{sr}, derived from the solar radiation:

$$\boldsymbol{a_{sr}} = \frac{\boldsymbol{p_{sr}}(1+\varepsilon)S}{m} \tag{2.18}$$

2.4 Geostationary Orbit

Since this thesis focuses on the study of satellites in geostationary orbit, it is useful to describe the main physical characteristics of such orbits. To accomplish this, it is necessary to define two crucial parameters of a satellite: the orbital period T and the circular velocity v_c.
The orbital period is the time required for a satellite to complete a single orbit around another celestial body. For an elliptical orbit, using the conservation of angular momentum and through geometric considerations, it is demonstrated that:

$$T = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{2.19}$$

For a circular orbit, such as the ideal geostationary orbit, it is possible to easily derive the tangential velocity of the satellite by equating the gravitational acceleration a_c, described by equation (2.2), to the centripetal acceleration $\ddot{r}$:

$$\begin{cases} a_c = -r\dot{\nu}^2 \\ \ddot{r} = -\dfrac{\mu}{r^2} \end{cases} \longrightarrow \quad r^2\dot{\nu}^2 = \frac{\mu}{r} \tag{2.20}$$

Where r represents the radius of the circular orbit and $\dot{\nu}$ corresponds, in this specific case, to the angular velocity. Substituting the expression for the tangential or circular velocity:

$$v_c = v_t = r\dot{\nu} \quad \longrightarrow \quad v_c = \sqrt{\frac{\mu}{r}} \tag{2.21}$$

2.4.1 Orbital Characteristics

The geostationary orbit is a circular, equatorial orbit with a period equal to one *sidereal day*[3]. Therefore, it represents a particular case of a geosynchronous orbit, characterized by both zero inclination and eccentricity. Referring to $\omega_\oplus$ as the period of Earth's rotation and as ω the period of the geostationary orbit:

$$\begin{cases} \omega_\oplus = \dfrac{2\pi}{86164,1} \simeq 7.292 \cdot 10^{-5}\ rad/sec \\ \omega = \omega_\oplus \end{cases} \tag{2.22}$$

[3]Time taken by the Earth to complete one full rotation around its axis. Shorter by 4 minutes than the solar day: the duration for Earth to return to the same position relative to the Sun

As can be easily seen from equation (2.19), fixing the period of an orbit automatically implies establishing its semi-major axis. Applying the circular velocity definition from equation (2.21) allows us to ascertain the required semi-major axis, or radius, for a geostationary satellite to complete its orbit within one sidereal day:

$$v = \sqrt{\frac{\mu_\oplus}{r}} = \omega_\oplus r \tag{2.23}$$

$$\frac{\mu_\oplus}{r_{gs}} = \omega_\oplus^2 r_{gs}^2 \quad \longrightarrow \quad r_{gs} = \left(\frac{\mu_\oplus}{\omega_\oplus^2}\right)^{1/3} \simeq 42164 \; km \tag{2.24}$$

However, depending on the context, different values can be used to indicate the altitude of a geostationary orbit. Specifically, the average distance of the GEO spacecraft from the center of the Earth is approximately 42164.5 km. Nonetheless, the average value of the geostationary semi-major axis is often cited as 42165.8 km. These two averages differ because the latter is calculated based solely on the Earth's attraction in R2BP, while the former takes into account the complete Earth potential, as well as the influence from the Sun and the Moon.

The table below summarizes the main parameters related to the ideal GEO orbit.

Geostationary Orbit	
Orbital Period	$T = 86164\,\text{sec}$
Semi-major Axis	$a = 42164\,\text{km}$
Altitude	$h = 35786\,\text{km}$
Inclination	$i = 0°$
Eccentricity	$e = 0$
True Anomaly	$\nu = \omega_\oplus t$

Table 2.2: Main characteristics of the GEO orbit

2.4.2 Mean Longitude Drift Rate

The mean longitude drift rate, or simply drift rate, holds significant importance and finds extensive application in the context of geostationary orbits, especially in the planning of relocation manoeuvres. If the semi-major axis of the ideal geostationary orbit is denoted by A and the deviation from this value by δa, the drift rate can be expressed as follows:

$$D = -1.5\frac{\delta a}{A} \tag{2.25}$$

D is dimensionless, but for practical purposes, it is often converted to degrees per day by multiplying by 361 deg/day. The drift rate depends solely on the variation in semi-major axis and is frequently used in place of a since it can be more relevant for describing the properties of a geostationary orbit. In fact, for a perturbed orbit, D is valuable for quantifying the relative error in

the semi-major axis and, simultaneously, the accumulated error over time in terms of longitude. In practical terms, a $D = 0.1$ deg/day indicates that an error of 0.1 deg in longitude is accumulated each day by the satellite. Then from the drift value, the corresponding error in a, in this case 7.8 km, can be derived.

During relocation operations, the drift rate is utilized to determine the semi-major axis of the *extraction orbit* (defined in subsection 2.4.4) and to compute the change in longitude over time. In particular selecting the target value for drift rate, the extraction orbit a is defined, and the resulting change in longitude over time while remaining in that orbit is communicated. A higher orbit than the GEO ($\delta a > 0$) corresponds to a longer orbital period, causing the satellite's longitude to lag behind the Earth's rotation, resulting in a mean westward longitudinal drift ($D < 0$). Conversely, a lower orbit ($\delta a < 0$) induces an eastward drift ($D > 0$). This topic is explained in further detail in subsection 2.4.4.

2.4.3 Parameterization

It is evident that the geostationary orbit possesses unique characteristics; therefore, its parameterization typically involves the use of slightly different parameters compared to classical orbital elements. Since the ideal orbit of a GEO satellite lies on the equatorial plane, it is convenient to project certain vector quantities onto it.

2.4.3.1 Inclination Vector

The first step is to define the vector the *orbital pole* or the *three-dimensional inclination vector*:

$$\boldsymbol{I} = \begin{bmatrix} \sin i \ \sin \Omega \\ -\sin i \ \cos \Omega \\ \cos i \end{bmatrix} \tag{2.26}$$

Which is unit vector orthogonal to the orbital plane, positive with respect to the motion of the spacecraft along its orbit, and therefore parallel to the angular momentum vector. Its projection on the x-y-plane allows for the delineation of the *two-dimensional inclination vector*:

$$\boldsymbol{i} = (i_x, i_y) = (\sin i \ \sin \Omega, -\sin i \ \cos \Omega) \simeq (i \ \sin \Omega, -i \ \cos \Omega) \tag{2.27}$$

Where because of small inclinations one can approximate $\sin i \simeq i$. The inclination vector is used in the case of perturbed orbits, where, in contrast to ideal ones, the inclination is close to zero but not exactly zero. This vector has a magnitude equal to i and points in the direction of $\Omega = -90°$.

2.4.3.2 Eccentricity Vector

The two-dimensional eccentricity vector, which can be visualized in the x-y-plane, has a magnitude of e and points from the center of the coordinate system towards the orbit perigee:

$$\boldsymbol{e} = (e_x, e_y) = (e \cos(\Omega + \omega), e \sin(\Omega + \omega)) \tag{2.28}$$

This vector is also employed to denote real geostationary orbits, which are characterized by low eccentricity values.

2.4.3.3 Mean Longitude

To describe the position of a GEO spacecraft in the Earth's rotating system, it is useful to utilize a specific parameter known as the *mean longitude*. It indicates the average angular position, relative to the Greenwich Meridian, of the satellite along its orbit, which remains constant for an ideal GEO spacecraft. This parameter, due to its distinctive characteristics, proves particularly convenient and is written as follows:

$$\lambda = l - G \tag{2.29}$$

Where G is the Greenwich sidereal angle and l the True Longitude:

$$\begin{cases} G = G_0 + \Psi(t - t_0) & \text{with} \quad \Psi = \omega_\oplus \\ l = \Omega + \omega + \nu \end{cases} \tag{2.30}$$

In detail, G represents the angle that specifies the position of the zero meridian relative to the vernal point, and it can be computed as a function of UTC time utilizing the Earth's uniform angular velocity, denoted as Ψ above. In equation (2.30), G_0 denotes the value of G at a chosen epoch t_0.

The mean longitude is conventionally deemed positive for east longitudes and negative for west longitudes, typically adjusted to fall within the interval (-180°, +180°). In Fig. 2.4, the angles just mentioned are represented more clearly. In the image, the x-axis points in the direction of the vernal point, while the angle s is equivalent to the true longitude.

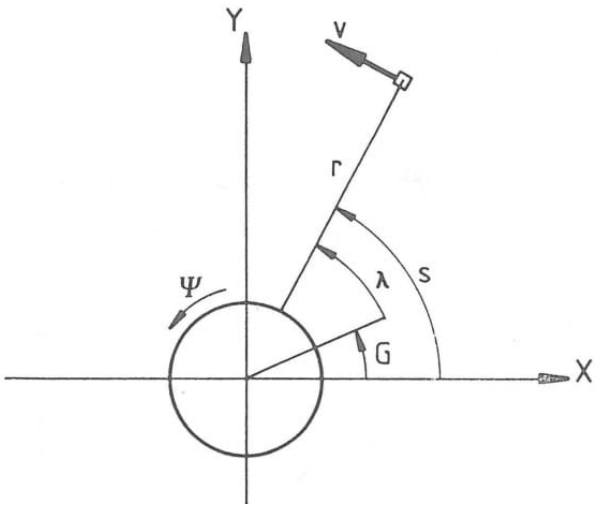

Figure 2.4: Seen from north: Sidereal angles of Greenwich (G), spacecraft (s) and the satellite's longitude (λ) [2]

Starting from equation (2.29), it can be demonstrated, through some steps and approximations not reported here for brevity, that:

$$\begin{aligned} \lambda = l - G &= \Omega + \omega + \nu - G_0 - \Psi(t - t_0) = \\ &= \Omega + \omega - G_0 + \Psi(t_0 - t_p) - 1.5(\delta a/A)\Psi(t - t_p) + 2e \sin \Psi(t - t_p) \end{aligned} \tag{2.31}$$

The operations necessary to achieve this formulation involve solving the differential equation known as *Kepler's equation*, which describes the time derivative of the true anomaly, $d\nu/dt$. The solution to this equation allows to obtain an expression for the true anomaly ν as a function of time, which can be inserted into equation (2.31). Kepler's equation is solved by incorporating the trajectory equation depicted in equation (2.6) and linearizing it for a small eccentricity. Hence, the characteristics of a GEO orbit are exploited, particularly the almost negligible eccentricity ($e \approx 0$), and the orbital period equal to one sidereal day ($\omega = \Psi$).

In equation (2.31), t_p represents the moment of perigee passage used as a constant of integration for calculating the linear solution of Kepler's equation, while t_0 indicates a designated epoch. The equation (2.31) can then be modified by considering the term:

$$\Psi(t - t_p) \approx l - \Omega - \omega \tag{2.32}$$

which allows to obtain the following formula for the satellite longitude:

$$\lambda = \lambda_0 + D(l - l_0) + 2e_x \sin l - 2e_y \cos l \tag{2.33}$$

Where λ_0 is a new constant, called *mean longitude at epoch*, which is defined by:

$$\begin{aligned} \lambda_0 &= \Omega + \omega - G_0 + (1 + D)\Psi(t_0 - t_p) = \\ &= (1 + D)l_0 - G_0 - D(\Omega + \omega) \end{aligned} \tag{2.34}$$

At this point, all the elements needed to represent an approximately geostationary orbit are available:

$$(\lambda_0, D, e_x, e_y, i_x, i_y) = (\lambda_0, D, \boldsymbol{e}, \boldsymbol{i}) \tag{2.35}$$

The information regarding the semi-major axis of the orbit is provided by D, while $_0$ plays a similar role to the true longitude l. This set of parameters, called *synchronous elements*, is used for real geostationary orbit instead of the classical elements defined previously. Various alternative definitions exist in the literature for similar sets of elements, sometimes also known as *equinoctial elements*.

2.4.4 Longitude Relocation

Longitude relocation is a common activity necessary for GEO satellites, involving the shift of their operational longitude to enable the provision of new services in different locations on Earth. As discussed in Chapter 1, the peculiar feature of a GEO satellite is to remain stationary above a specific area of the Earth, maintaining a constant longitude λ relative to the Greenwich Meridian. This ability is made possible by the precise set of orbital parameters defining the geostationary orbit. Specifically, the particular height established for the orbit ensures synchronization between the spacecraft and Earth's rotation, resulting in matching of their angular velocities.

This phenomenon can be further understood through equation (2.25), which indicates that the drift rate of a GEO orbit approaches zero, albeit not precisely due to unavoidable external perturbations. This mathematical insight elucidates why the longitude of GEO satellites remains consistent

over time. In light of the above, it becomes evident that modifying the operational λ of the spacecraft necessitates adjusting its orbital altitude. In fact, altering the semi-major axis of the orbit, the previously mentioned synchronization between the satellite and the Earth is disrupted, causing the satellite to adopt a different angular velocity than the Earth. Consequently, the satellite can deviate from its original longitude, as its new angular velocity enables it to shift relative to the Earth, thereby altering its angular position over time.

To elaborate further, the new orbit attained after adjusting the semi-major axis, referred to as the *drift* or *extraction* orbit, is characterized by a distinct value of D, which in this case will be no more zero. As the satellite completes multiple revolutions in this orbit, it accrues a certain phase shift concerning the initial longitude, proportionate to the time spent in orbit, leveraging the new drift rate. The approximate change in longitude attained during this period can be estimated through the following formula:

$$\Delta\lambda = D \cdot t_{drift} \tag{2.36}$$

Where t_{drift} represents the time spent in the drift orbit. Ultimately, the relocation process entails multiple orbital manoeuvres aimed at either increasing or decreasing the orbit's altitude. Modifying the semi-major axis it is achieved a new orbital period either shorter or longer than a sidereal day, which induces an apparent 'drift' Eastward or Westward, respectively, towards a new longitude. In this operation three main phases can be identified:

- *Extraction phase*: In this phase, the satellite is gradually manoeuvred out of the current geostationary orbit. The spacecraft's velocity and trajectory are adjusted to transition to a drift orbit, characterized by an altitude different from that of the GEO orbit.
- *Drift phase*: During this phase, the satellite simply orbits for a certain period of time, gradually drifting to a different longitude. No additional manoeuvres are generally executed during this period.
- *Insertion phase*: After a sufficient time period, the spacecraft is finally brought back into a geostationary orbit. Precise manoeuvres are executed at the appropriate moment to achieve the desired operational longitude by the end of this phase.

Once the drift orbit is selected and the desired change in longitude is determined, equation (2.36) is employed to obtain an initial estimate of the time required for the drift phase. However, it should be noted that even during manoeuvre periods, namely during the extraction and insertion manoeuvres, a slight variation in longitude occurs, which although minimal, is not negligible. It is convenient noting that to move the satellite Eastward relative to Earth, thus gaining longitude, it is necessary to reduce its orbital altitude. This results in achieving a positive D, as depicted in equation (2.25), given that a negative δa is applied. Conversely, increasing the satellite's semi-major axis, denoted by a positive δa, determines a negative D, enabling the satellite's shift in the Westward direction, decreasing its operational longitude over time.

Chapter 3

Space Propulsion

"Newton's Third Law. The only way humans have figured out how to move forward is to leave something behind"
— Interstellar

To provide a clearer introduction to the conducted work, it is advantageous to describe some fundamental concepts of space propulsion. Therefore, this chapter will discuss the fundamental principles of space propulsion, introducing some key quantities related to this subject. Next, an insight into electrical thrusters—those considered for the analyzed problem—will be provided.

As seen in the previous Chapter, the trajectory of a generic satellite can be uniquely defined by its state vector, i.e., by its position and velocity at a given time. It goes without saying that to modify the orbit of a spacecraft, it is necessary to change its velocity. Therefore, propulsion can be defined as the ability to generate a force capable of altering the velocity of a satellite, allowing, in this way, the modification or maintenance of its trajectory. An *orbital manoeuvre* is the use of propulsion systems to change the orbit of a spacecraft.

3.1 Space propulsion fundamentals

3.1.1 Overview

Space propulsion can be divided into two main categories, depending on its purpose:

- *Auxiliary Propulsion*: it has the intent to contrast external perturbations to maintain the desired orbit (station-keeping manoeuvres).
- *Primary Propulsion*: it aims to change a given orbit

All types of thrusters are grounded in *Newton's third law of motion*, regardless of their specific mode of operation:

"If two bodies exert forces on each other, these forces have the same magnitude but opposite directions"

Thus, the only way for a satellite to generate thrust in space is to carry onboard something to exchange momentum with: the propellant. A conventional thruster provides energy to the propellant mass, which accelerates and, once ejected, creates thrust (T). The effect of this force is to produce a change of velocity, ΔV. The specific energy source used to generate the acceleration of the propellant allows the classification of propulsion systems into three distinct categories:

- *Chemical Propulsion*: It utilizes chemical reactions of the propellant or between two substances—an oxidizer and a fuel—to generate high-energy gases, which are expelled to produce thrust.
- *Electric Propulsion*: It exploits electric power or electromagnetic fields to generate thrust.
- *Nuclear Propulsion*: It utilizes the energy released from nuclear reactions to accelerate the propellant. In particular, an onboard nuclear reactor can power electric engines or provide thermal energy to the propellant.

As mentioned in Chapter 2, if a satellite is only subjected to the gravitational attraction exerted by the primary body, and no perturbative forces are present, its trajectory is a conic section. The effect of a manoeuvre thrust on an orbit can be calculated by numerically integrating the differential equation (2.2) of spacecraft motion, including the thrust acceleration on the right-hand side. A formulation for the thrust can be easily obtained by applying *Newton's second law of motion*:

$$\boldsymbol{T} = m\frac{d\,\boldsymbol{V}}{dt} \tag{3.1}$$

Where T represents the *thrust vector*, and the hypothesis of constant mass has been applied. This assumption is particularly appropriate in the case of station-keeping manoeuvres with electric propulsion, where the mass of burned fuel is negligible. It is self-evident that the change in velocity resulting from the application of thrust can be estimated by integrating the preceding equation. If the duration of the thrust is very short relative to the orbital period, one can consider the thrust to be impulsive and calculate its effect semi-analytically by adding a velocity increment, or ΔV:

$$\Delta\,\boldsymbol{V} = \int_{t_0}^{t_f} \frac{\boldsymbol{T}}{m}\,dt \simeq \frac{\boldsymbol{T}}{m}\Delta t \tag{3.2}$$

The theoretical model of an *impulsive manoeuvre* assumes that the manoeuvre takes place via a single burn at one point in space. Thus, the variation of the trajectory occurs through instantaneous finite changes in the spacecraft's velocity, due to the application of infinite thrust in an infinitesimal instant. In the physical world, truly instantaneous changes in velocity are clearly not feasible, as it is impossible to apply an infinite force during an infinitely short time. Nevertheless, the impulsive model, although less accurate, is a very simple and convenient way to study the effect of a manoeuvre on an orbit. It is important to highlight that this simplification is acceptable only for chemical thrusters, given their mode of operation through short impulses. In fact, electric propulsion systems, in light of their operational features, currently face limitations in force and acceleration generation. This limitation imposes their use for long-term manoeuvres, in which the trajectory changes gradually, and the thrust is applied continuously. Hence, the manoeuvres described in this work, considering that the spacecraft of interest are equipped with electric thrusters, will be categorized as *continuous manoeuvres*: low thrust applied over a long period of time.

3.1.2 Significant Parameters

As described above, the fundamental principle of propulsion is the action-reaction law, and this characteristic unites all thrusters. Consequently, it is possible to illustrate the general functioning of a generic propulsion system and introduce some important parameters through an apposite example. It is considered a body on which no external force is applied, in motion with a velocity v at time t. At the time $t + dt$, the body expels an infinitesimal mass of propellant m_p at a velocity $c - v$, where c represents the velocity of the propellant relative to it. The body reaches a velocity $v + dv$ after ejection. Therefore, the spacecraft increases its velocity due to the loss of a part of its mass. Assuming a closed system, one can consider the global momentum as constant during the process. Hence, by imposing the conservation of the total momentum of the system:

$$mv = (m - dm_p)(v + dv) - dm_p(c - v) \tag{3.3}$$

Carrying out the mathematical simplifications and neglecting the infinitesimal terms of higher order, it is possible to write:

$$m\,dv = dm_p\,c \tag{3.4}$$

Since the propellant is ejected continuously rather than discretely, it is necessary to modify the equation just presented. First, the *propellant flow* is defined:

$$\dot{m}_p = \frac{dm_p}{dt} \tag{3.5}$$

Then, incorporating this term into equation (3.4), one can obtain:

$$m\frac{dv}{dt} = \dot{m}_p\,c \tag{3.6}$$

This equation can be used to write an alternative formulation for the *Thrust* (T). In fact, by considering equation (3.1), the thrust is given by the equation:

$$T = m\frac{dv}{dt} = \dot{m}_p\,c \tag{3.7}$$

with which one can express the *thrust power* (P_T):

$$P_T = \frac{1}{2}\dot{m}_p\,c^2 = \frac{1}{2}T\,c \tag{3.8}$$

which is the power required to accelerate the propellant to the velocity that generates the thrust T. It is worth highlighting that c represents an important parameter called the *effective discharge velocity*, generally used to describe the performances of a thruster. As a matter of fact, the higher c is, the greater the thrust generated for the same propellant flow. Additionally, it is important to specify that the effective discharge velocity does not properly represent the exit velocity of the propellant.

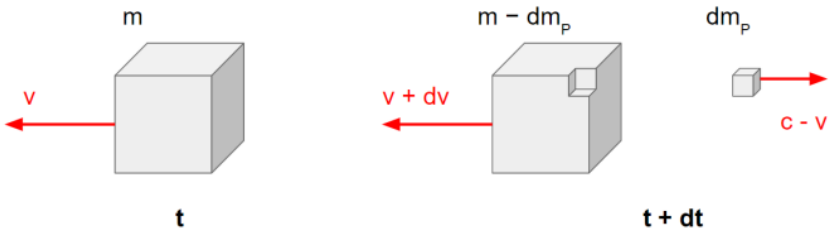

Figure 3.1: Momentum conservation scheme [3]

In fact, the correct formulation of thrust should take into account a term called *static thrust*, that adds up to the *dynamic* one:

$$T = \dot{m}_p \, u_e + A_e(p_e - p_0) \tag{3.9}$$

where u_e is the exit velocity of the propellant, A_e represents the exit section of the nozzle, p_e is the pressure of the propellant in the exit section, and p_0 is the ambient pressure (near zero in space). The physical reason behind this additional term is that the propellant outside the nozzle, ejected at the pressure p_e, exerts a force on the propellant that is still inside the thruster and still belongs to it. The resulting force generated is proportional to the difference between the exit pressure and the ambient pressure. Hence, to avoid specifying the division between static and dynamic thrust, the effective discharge velocity is defined. Furthermore, to be precise, the value of c can be obtained only once the thrust is known, and it depends solely on the characteristics of the employed propulsor and propellant:

$$c = \frac{T}{\dot{m}_p} \tag{3.10}$$

Nevertheless, it must be said that for electric space propulsion, the static thrust is usually very low; therefore, the effective discharge velocity tends to be equal to the exit velocity: $c \simeq u_e$.
It is now useful to introduce some additional entities generally employed to quantify and evaluate the effect of the thrust and the performances of the propulsor. The first one is *total impulse*:

$$I_t = \int_{t_0}^{t_f} T \, dt \tag{3.11}$$

it measures the effectiveness of the propulsion system and its capacity to generate thrust. The second parameter is the *propellant mass*, which is simply the integral of the propellant flow:

$$m_p = \int_{t_0}^{t_f} \dot{m}_p \, dt \tag{3.12}$$

it is used to estimate the propulsor consumption and represents the expense required for creating thrust. With the total impulse, it is finally possible to define the *specific impulse*:

$$I_s = \frac{I_t}{m_p g_0} \tag{3.13}$$

where g_0 is the gravity acceleration on the Earth's surface. The specific impulse is a measure of the efficiency with which the propulsion system utilizes the propellant to generate thrust. Specifically, if m_p and T are considered constant, substituting equations (3.10), (3.11), (3.12) into equation (3.13), one can obtain:

$$I_s = \frac{c}{g_0} = \frac{T}{\dot{m}_p\, g_0} \tag{3.14}$$

as seen in this expression, c and I_s represent the same quantity, differing only by a multiplicative constant. Both of these parameters are interchangeably used to evaluate the efficiency of a thruster. Obviously, it is crucial to have a propulsion system with a high specific impulse for the optimal success of a mission. In fact, for a given propellant mass, a higher specific impulse (I_s) allows the generation of a higher thrust for the same operating time or the production of the same thrust but over a longer interval. The importance of a high specific impulse is perhaps more evident when considering the *Tsiolkovsky equation*. This expression, also referred to as the *ideal rocket equation*, is one of the most important equations in space propulsion. It connects the variation of velocity, ΔV, obtained in the manoeuvre, with the cost required for its generation by the thruster in terms of propellant consumption. The variation of velocity is computed in the ideal case, with no external forces acting, and represents the propulsive cost demanded by the propulsion system. The derivation of this equation falls beyond the scope of this work; thus, for brevity, it is directly reported:

$$\Delta V = c\ ln\frac{m_0}{m_f} = I_s g_0\ ln\frac{m_0}{m_f} \tag{3.15}$$

where m_0 is the initial mass of the spacecraft, and m_f represents the final mass, i.e., without the propellant consumed. It is evident that a high I_s is associated with a minor fuel consumption. In particular, if the specific impulse is too low, manoeuvring becomes practically infeasible for a satellite, undermining the ability to execute any corrective and control action over it.
By appropriately reversing equation (3.15), it is possible to estimate the final mass achieved at the end of the manoeuvre, as expressed by:

$$m_f = m_0\, e^{-\frac{\Delta V}{c}} \tag{3.16}$$

Furthermore, by utilizing this expression, one can compute the propellant mass burned during the manoeuvre phase as:

$$m_p = m_0 - m_f = m_0(1 - e^{-\frac{\Delta V}{c}}) \tag{3.17}$$

Where it has been assumed that the difference between the initial and final mass of the spacecraft is entirely composed of the propellant consumed, which is a theoretically valid hypothesis.

3.2 Chemical propulsion

Chemical propulsion systems utilized in GEO satellites generally involve exploiting a controlled chemical reaction to energize the propellant, thereby creating high-temperature and high-pressure gases to be expanded and thus accelerated through the spacecraft's nozzle, generating thrust. In fact, following the Newtonian principle of action and reaction, the effect of expelling a mass at high velocity is to propel the satellite forward. The produced thrusts are particularly elevated compared to electric propulsion, resulting in significant accelerations and thus high ΔV. The operating mode entails, as previously mentioned, impulse manoeuvres, given that large velocity variations are generated in very short periods of time.

Common propellants include *hypergolic* combinations such as *hydrazine* (N_2H_4) and *nitrogen tetroxide* —NTO—(N_2O_4), which ignite spontaneously upon simple contact and offer reliable and controllable thrust. Moreover, the exothermic decomposition reaction of hydrazine can be exploited to obtain monopropellant thrusters, achieving similar performance compared to bipropellant chemical propulsion systems. Thrusters fueled by hydrazine are typically designed to generate forces ranging from 0.5 to 20 N, with a specific impulse between 220-300 s.

The main advantages of chemical propulsion are those linked to the high thrust-to-weight ratio, proven reliability, and flexibility. However, there are also some general drawbacks; in many cases, the use of chemical propellant imposes strict mass and volume constraints on satellite design. In particular, propellant tanks and associated hardware often limit payload capacity and mission objectives. Furthermore, the finite propellant reserves require satellite operators to carefully manage fuel consumption to maximize mission duration. Last but not least, safety concerns must also be considered, as hypergolic propellants are generally toxic; therefore, their handling and disposal pose challenges, necessitating stringent safety protocols and disposal procedures.

3.3 Electric Propulsion

Electrical propulsors have become extensively employed in recent missions owing to their high efficiency. This type of propulsion system is particularly well-suited for long-term manoeuvres characterized by low variations in velocity, such as station-keeping operations or, as investigated in this thesis, relocation manoeuvres. As mentioned in the previous paragraph, electric thrusters exploit electrical power, usually generated using solar arrays, to accelerate a propellant gas. It is possible to distinguish different categories of electric propulsors, depending on how this energy is employed to create thrust:

- *Electrothermal propulsion*: uses electrical power, either through resistance or an electric arc, to heat a propellant. The heated propellant is then accelerated and expanded in a nozzle, where the thermal energy is converted into kinetic energy, generating thrust.

- *Electrostatic propulsion*: uses electrical power to ionize the propellant and accelerate it by exploiting an electric field. In the specific, the ionized gas contains both ions and electrons, but only the former are accelerated within the thruster and then neutralized at its exit.

- *Electromagnetic propulsion*: uses electric and magnetic fields to accelerate the propellant, creating thrust. In the specific, acceleration is produced by the electromagnetic forces due to the interaction of internal and/or external magnetic fields with the electrical current in the propellant flow.

It is worth highlighting that some electrostatic thrusters also utilize magnetic fields, although the acceleration is only generated by the electric field. In contrast, electromagnetic acceleration is created by both electric and magnetic forces. Besides the classification by physical phenomenon, there is another possible categorization based on the characteristic power of the thruster:

- *Microthrusters*: used, in general, for precision attitude control.
- *1 kW*: used for station keeping or orbit injection —or de-orbit— of small satellites.
- *5-10 kW*: used for GEO insertion or de-orbit of large satellites.
- *100+ kW*: are under development concepts that could be implemented in human exploration missions.

3.3.1 Electrostatic Propulsion

Electrostatic thrusters accelerate the propellant through electrostatic forces, generated by exploiting electrical fields. In this kind of propulsion system, the presence of magnetic fields is not directly related to the propellant acceleration. In particular, to correctly operate an electrostatic thruster, three different processes are required:

- *Ionization*

  Electrostatic thrusters operate with ionized propellants, as electrical forces necessitate the presence of charged particles to produce acceleration. In the ionization process, some atoms of the propellant are separated from one of their electrons, thus creating an ion —and the electron itself—. For this operation, energy is clearly required; this *first ionization energy*, ε_i, is different for each atom on the periodic table. Thus, the ionization reaction can be summarized as:

  $$A + \varepsilon_i \rightleftharpoons A^+ + e^-$$

  where A and $A+$ are respectively the propellant's atom and ion, and e^- is the free electron. Clearly, the process is not perfect since not all of the propellant's atoms are ionized, and eventually, some atoms can lose more than one electron, creating, for example, ions with a double charge. The second step to generate thrust is to separate ions from electrons. Applying an electric field to both electrons and ions will cause each to accelerate but in opposite directions, resulting in null thrust.

This is due to the fact that cations and electrons have the same charge in magnitude but opposite signs. Considering the physics behind this phenomenon:

$$\boldsymbol{F} = q\boldsymbol{E}$$

where $\boldsymbol{F}$ is the electrostatic force acting on the considered particle, $\boldsymbol{E}$ is the electric field, and q is the elementary charge. Then, electrons and ions are subjected to equal but opposite forces, which mutually neutralize each other. In the Fig. 3.2, this phenomenology is represented in a schematic way.

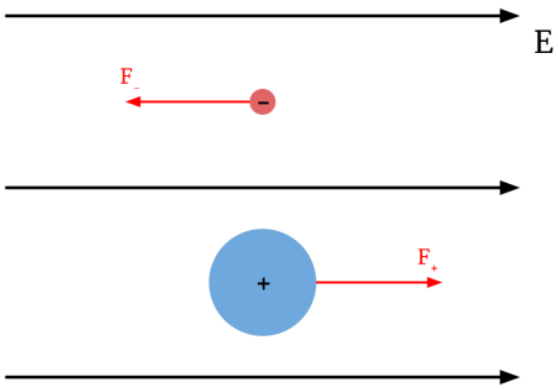

Figure 3.2: Electric forces acting on ions and electrons [3]

- *Acceleration*

  The acceleration of ions is enabled by an electric field, which, being a conservative field, is connected to an electric potential. Thus, the ions experience the application of an electric potential difference, referred to as the net accelerating potential (V_N). Similar to a mass subjected to a gravitational field, when ions move from a higher potential to a lower one, their velocity increases, leading to acceleration. Mathematically, this implies that electrical potential energy is converted into kinetic energy. One can consider that inside the thruster, ions possess only electrical potential energy —the propellant is globally static, with only chaotic thermal agitation velocity, which is globally null—. Assuming this energy is completely converted into kinetic energy at the exit section, it is possible to impose the conservation of total energy:

$$qV_N = \frac{1}{2} m_+ u_+^2 \tag{3.18}$$

  where q and V_N have already been defined, m_+ represents the atomic mass of the ion —comparable to the mass of the propellant atom itself— and u_+ denotes the exit velocity. The expression describing the effective discharge velocity can be readily obtained.

As stated in paragraph 3.1.2, it can be assumed to be equal to u_+. Therefore:

$$u_+ = \sqrt{\frac{2qV_N}{m_+}} \tag{3.19}$$

- *Neutralization*

  The final process required for an electrostatic thruster is to neutralize the accelerated beam of ions. It is crucial to maintain the overall neutrality of the propulsion system by balancing the positive flow of charges with an equal current of electrons. Otherwise, if ions were just accelerated, the thruster itself would accumulate a negative charge internally. To prevent this undesired situation, the ejected ion beam passes through a sort of electron cloud created at a certain distance from the thruster's exit. The current of electrons, referred to as the *neutralizing current*, is not accelerated since it does not have to contribute to thrust production. Each ion acquires an electron and becomes neutral; this process leads to a globally neutral gas —as the propellant was before ionization—.

3.4 Orbital Manoeuvres

As stated at the beginning of this chapter, an orbital manoeuvre involves utilizing satellite's propulsion system to alter its trajectory. In particular, manoeuvre thrusts enable the modification of an orbit by changing one or more orbital parameters. In fact, the thrusters, generating thrust, produce an acceleration that influences the spacecraft motion equation seen in Chapter 2.

The acceleration is represented by a vector comprising three components, which depend on the directions in which the vector itself is projected. Specifically, acceleration is typically split in one of two ways: with respect to the radius vector connecting the Earth and the satellite r, or with respect to the spacecraft velocity vector V. Consequently, the following two sets of three mutually perpendicular components can be respectively identified:

- *Projection in the r direction*: $(\boldsymbol{a_T, a_R, a_W})$

  When the acceleration is resolved in the radial direction, a_R represents the component of acceleration parallel to r, while a_T is the tangential component orthogonal to the radius vector r. Finally, a_W is the out-of-plane component, perpendicular to the plane defined by a_R and a_T.

- *Projection in the V direction*: $(\boldsymbol{a_V, a_N, a_W})$

  The acceleration is split with reference to the velocity vector direction. Specifically, a_V represents the component of acceleration aligned with V, a_N is the component normal to V. Similarly, as in the previous projection, a_W is the out-of-plane component and is perpendicular to the plane defined by a_V and a_N.

However, for a circular orbit like the geostationary one, the two projections are equivalent since the velocity V is purely tangential, making its direction orthogonal to r. For this reason, particular attention will not be paid to this distinction in this document.

3.4.1 Gauss-Planetary Equations

The orbital parameter evolution due to the acceleration generated by the manoeuvre thrusts can be evaluated by referring to the *Gauss-planetary equations*. These allow for understanding the effects of the thrusters on the Keplerian orbit of a satellite. The equations, described in further detail in Blanco and McCuskey [5], define the connection between the time derivatives of orbital elements and their corresponding cause, which is a generic acceleration different from the one caused by the Earth's gravitational field. In this specific context, the acceleration obviously refers to that produced by the propulsors which deviate the satellite from the Earth's gravitational action.

To generalize the discussion, reference is made to a generic orbit. The acceleration imparted by the thrusters is decomposed with respect to the velocity vector V. Following the convention used by Gauss in his own treatment, a_V is considered positive in the direction of increasing longitude, a_N is positive in the direction of increasing radial distance, and a_W is directed positively toward the north pole. Consequently, Gauss' equations can be formulated as follows:

$$\frac{da}{dt} = \frac{2}{n\sqrt{1-e^2}}\left[a_N e \sin\nu + \frac{a(1-e^2)}{r}a_V\right] \tag{3.20}$$

$$\frac{de}{dt} = \frac{\sqrt{1-e^2}}{na}\left[a_N \sin\nu + \left(\frac{e+\cos\nu}{1+e\cos\nu} + \cos\nu\right)a_V\right] \tag{3.21}$$

$$\frac{d\omega}{dt} = \frac{\sqrt{1-e^2}}{nae}\left[-a_N\cos\nu + \left(1+\frac{r}{a(1-e^2)}\right)a_V\sin\nu\right] - \frac{d\Omega}{dt}\cos i \tag{3.22}$$

$$\frac{di}{dt} = \frac{1}{na\sqrt{1-e^2}}\frac{r}{a}\cos(\omega+\nu)a_W \tag{3.23}$$

$$\frac{d\Omega}{dt} = \frac{1}{na\sqrt{1-e^2}}\frac{r}{a}\frac{\sin(\omega+\nu)}{\sin i}a_W \tag{3.24}$$

$$\frac{dM}{dt} = n + \frac{1}{na}\left[\frac{2r}{a} - \frac{(1-e^2)}{e}\cos\nu\right]a_N - \frac{(1-e^2)}{nae}\left[1+\frac{r}{a(1-e^2)}\right]a_V\sin\nu \tag{3.25}$$

In which the same notation seen in Chapter 2 is used to represent the classical orbital elements. While M denotes the *mean anomaly* of the satellite, a new quantity is defined as:

$$M = n(t-t_p) \tag{3.26}$$

The mean anomaly defines the angular distance of an orbiting body, such as a spacecraft, from the pericenter at an arbitrary time t. Specifically, in the formula just presented, t_p represents the time at which the body is at the pericenter, and n is the *mean angular motion* of the satellite, calculated as:

$$n = \frac{2\pi}{T} = \sqrt{\frac{\mu}{a^3}} \tag{3.27}$$

The Gauss equations prove convenient as they enable us to understand the influences of the three components of acceleration separately. It is evident that the only component contributing

to increasing a, namely the size of the orbit, is a_V, assuming that $e \ll 1$—this is particularly true when considering a GEO orbit. Hence, if the manoeuvre objective is to vary the orbit's altitude, it is advisable to thrust only in the tangential direction. Since the longitude relocation operation involves changing the semi-major axis of a GEO orbit, as explained in Chapter 2, the manoeuvres analyzed in this thesis will primarily consist of tangential acceleration.

To change the shape of the orbit, represented by the eccentricity e, both a_V and a_N are effective, with varying significance depending on where the thrust is applied along the orbit. Consequently, by performing tangential manoeuvres to vary the semi-major axis during the relocation operation, an additional effect is the modification of eccentricity.

The orbit plane, and therefore the argument of the ascending node Ω and the inclination i, are modified solely with thrust out of the plane, providing an acceleration a_W. The position where the thrust is applied redistributes the effect on i and Ω differently: at the nodes, there is only a change in inclination; at the anti-nodes, only $\Delta\Omega$ is obtained; otherwise, the thrust effect is divided by changing both parameters.

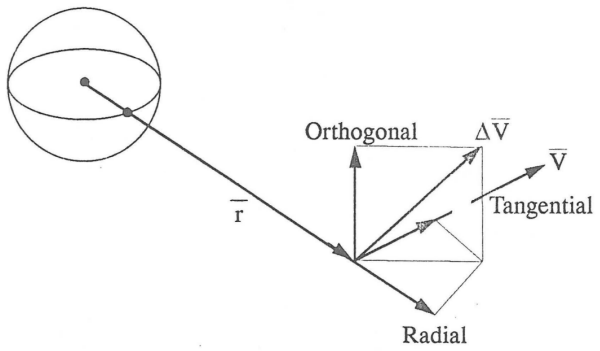

Figure 3.3: ΔV produced by a manoeuvre thrust projected with respect to the radius vector r [2]

3.4.2 Manoeuvres Thrusts for GEO Satellites

Thruster burns used for controlling geostationary orbits are typically either perpendicular to the orbital plane or tangential to the orbit; while radial burns are rarely employed. For a thruster, it is generally required that the force vector must pass through the spacecraft's center of mass to avoid generating any torque on it. However, another solution to avoid the creation of this torque consist in firing two thrusters simultaneously. Further requirements involve ensuring that antennas, solar panels, or other protruding equipment are not damaged by the exhaust plumes of the thrusters. In some cases, compromises are necessary, resulting in a component of force in an undesired direction, which inevitably decreases manoeuvre efficiency.

As mentioned, the focus of this work will be directed on the investigation of tangential manoeuvres, which are commonly referred to as *East-West thrusts*. As seen in the Gauss equations, a thrust in the tangential, i.e. east-west direction, alters both the longitude drift rate —and thus the semi-major axis— and the orbit's eccentricity. Given a three-axis spacecraft, it is possible to provide

it with a boost in its flight direction through a propulsor positioned on its westward surface. This is termed an eastward thrust, with the corresponding imparted ΔV considered positive. Conversely, a westward thrust consists of an acceleration in the opposite direction, resulting in a negative ΔV.

In the case of impulsive manoeuvres, a single tangential thrust acts as a short impulse, which instantaneously alters the spacecraft's flight velocity but not the position of the burn itself. Hence, the ΔV changes the orbit altitude everywhere except at the point where the firing has been executed, where the new orbit maintains its original altitude. Specifically, an eastward thrust raises the orbit height, but the point of the burn maintains its altitude, becoming the perigee of the new orbit. Conversely, a westward thrust, with negative ΔV, would lower the orbit, positioning the final orbit's apogee at the thrust point.

As observed in the previous subsection, a radial thrust only affects the eccentricity of the orbit, but it turns out to be less effective than a tangential thrust of the same magnitude. Therefore, using radial thrust to alter orbit eccentricity is not cost-effective. Radial and east-west thrusts are typically referred to as *in-plane manoeuvres*.

Finally, a manoeuvre producing a force perpendicular to the orbital plane represents an out-of-plane thrust, also known as a *North-South thrust*. These manoeuvres are used to alter the orientation of the orbit's plane, correcting both the inclination and the ascending node.

In many situations, a combination of manoeuvres in the east-west direction, called a *multiple in-plane thrust sequence*, is executed when it is required to alter the drift rate of the orbit controlling the eccentricity vector or without affecting it. It is evident that the relocation operation is composed of this kind of sequence, in order to properly modify the semi-axis of the orbit along with controlling the eccentricity. A sequence of two subsequent tangential manoeuvres is also performed to carry out a transition from one circular orbit to another with either greater or lesser altitude, an operation commonly known as a Hohmann transfer.

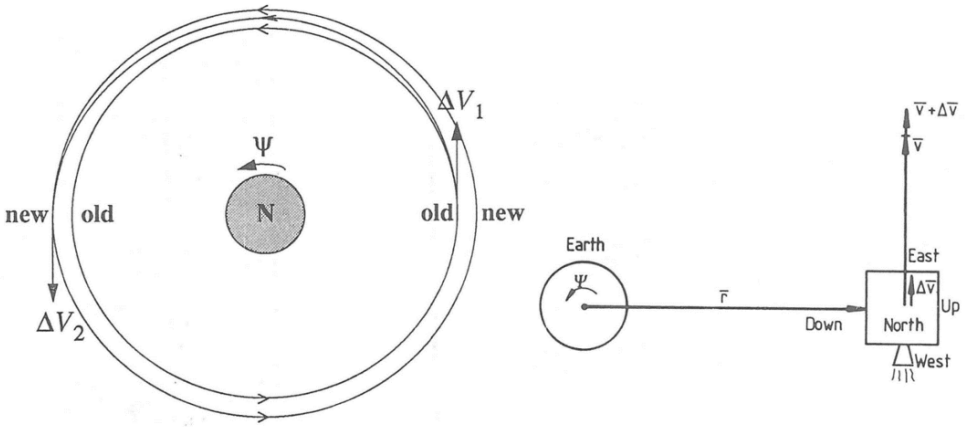

Figure 3.4: Multiple East-West Thrust Manoeuvres to change orbit: trajectory overview (left) and spacecraft view (right) as seen from the North [2]

Chapter 4

SQP algorithm for constrained nonlinear optimisation

"There's a way to do it better - find it."
— T. Edison

In this chapter, the global characteristics and mathematical aspects of the algorithm employed for optimising the manoeuvres addressed in this thesis are described. Firstly, an outline of the SQP method will be provided, illustrating the main characteristics of its algorithm while conveying the principal mathematical concepts that underlie it. Then, the general specifics of its implementation in the Fortran code NLPQLP will be presented.

optimisation is a branch of applied mathematics that studies theories and methods for finding extrema—maximum and minimum points—of a mathematical function within a specified domain. In general, an optimisation problem aims to maximize or minimize one or more specific quantities, expressed in the form of a suitable *objective function*. The problem can be subject to the satisfaction of a system of equality and inequality functions, referred to as *constraints*. In this case, the set of variables that constitutes the solution resolves the optimised problem while satisfying a series of specific conditions.

In the context of planning station shift manoeuvres, the optimisation usually focuses on minimizing the durations of these manoeuvres, resulting in a lower quantity of propellant used. Indeed, the duration of a manoeuvre is directly linked to propellant consumption; therefore, minimizing it limits consumption, leading to a reduction in the costs of the mission.

The methods typically employed to solve nonlinear optimisation problems are based on iterative algorithms. Thus, the solution is progressively obtained through a series of iterations, each aiming to improve upon the solution provided by the previous one. This process continues until an acceptable result is reached, convergence towards the optimal solution is achieved, or the maximum number of preset iterations is reached. It is worth highlighting that the starting point for these methods is anchored in an initial guess of the solution, often obtained analytically. This is used to initiate the computations in the first iteration of the process. Clearly, providing a precise estimation of the

solution proves advantageous for promoting the proper convergence of the method, especially for methods characterized by slow convergence and low robustness. In general, the more accurate the initial approximation, the faster and more probable the convergence of the algorithm.

4.1 Sequential Quadratic Programming

When some of the constraints or the objective function are nonlinear, it becomes necessary to implement a *nonlinear programming* process to solve the optimisation. *Sequential Quadratic Programming* (SQP) methods belong to the most powerful nonlinear programming algorithms. Specifically, SQP iterative algorithm is widely adopted to solve constrained nonlinear optimisation problems for various applications in science, engineering, industry and management.

4.1.1 Generalities

Methods for solving nonlinear programming problems can generally be classified into two main categories. The first class encompasses approaches that involve transforming the constrained problem into either a non-constrained problem or a series of unconstrained problems. Algorithms falling under this category utilize sequential or exact penalty functions, as well as sequential or exact augmented Lagrangian functions. The second class comprises methods that exploit the transformation of the constrained problem into a *sequence* of quadratic programming problems. The algorithm employed in this work belongs to this latter class. Within this category, a further subdivision can be made into two classifications:

- Sequential inequality quadratic programming (SIQP): in this case the sub-problems include both equality and inequality constraints.
- Sequential equality quadratic programming (SEQP): the sub-problems involve only equality constraints.

Sequential quadratic programming algorithm can be considered as part of quasi-Newton algorithms; however, if the problem is unconstrained, it reduces to Newton's method, also known as the *Newton–Raphson* method. The Newton's method allows finding the roots of a function f and consequently its extrema —i.e. the zeroes of its derivative f'. In particular, it is based on the idea of approximating the objective function and its higher-order derivatives around the current solution using a Taylor series. A detailed description of its algorithm is not relevant to the scope of this work, but it proves useful to present at least the expression implemented by the Newton-Raphson in its simplest uni-dimensional form:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{4.1}$$

where f is a real-valued function, the roots of which are being searched, f' is its derivative, and x_k is the solution at the k^{th} iteration. Obviously, the implementation for the search of the extrema is analogous, with the due differences.

The SQP algorithm shares many similarities with Newton's method and is indeed regarded as an alternative to it. Both are used to solve nonlinear problems, but they differ in their approaches, as one can observe from the overview carried out in the following paragraph.

4.1.2 SQP Algorithm

As mentioned before, SQP is implemented to solve smooth nonlinear constrained optimisation problems. Therefore, the general optimisation problem to minimize an objective function f under nonlinear equality and inequality constraints is considered. This can be written as:

$$\begin{aligned} & \min f(\boldsymbol{x}) \\ \boldsymbol{x} \in \mathbb{R}^n : \quad & g_j(\boldsymbol{x}) = 0, \quad j = 1, \ldots, m_e \\ & g_j(\boldsymbol{x}) \geqslant 0, \quad j = m_e + 1, \ldots, m \end{aligned} \tag{4.2}$$

where x is an n-dimensional parameter vector, which can be called solution vector. It is assumed that all problem functions $f(x)$ and $g_j(x)$, $\forall j = 1, ..., m$, are continuously differentiable on the whole $\mathbb{R}^n$. To be precise, the formulation of the optimisation problem addressed in this thesis presents some differences from the one just illustrated, since no equality constraints are applied. Instead, all constraints are expressed through inequality functions, therefore, specifying the presence of upper and lower bounds x_U and x_L, the problem becomes:

$$\begin{aligned} & \min f(\boldsymbol{x}) \\ \boldsymbol{x} \in \mathbb{R}^n : \quad & g_j(\boldsymbol{x}) \geqslant 0, \quad j = 1, \ldots, m \\ & \boldsymbol{x_L} \leqslant \boldsymbol{x} \leqslant \boldsymbol{x_U} \end{aligned} \tag{4.3}$$

However, for the purpose of the description presented in this section, it is preferable to refer to the more general case depicted in equation (4.2).

The fundamental concept of the algorithm revolves around formulating and solving a quadratic programming sub-problem in each iteration. This sub-problem is derived by linearizing the constraints and approximating the Lagrangian function quadratically:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}) - \sum_{j=1}^{m} u_j g_j(\boldsymbol{x}) \tag{4.4}$$

where $x \in \mathbb{R}^n$ is the primal variable and $u = (u_1, ..., u_m)^T \in \mathbb{R}^m$ the multiplier vector. The major reason for using a quadratic sub-problem, i.e., problem with a quadratic objective function and linear constraints is that such problems are relatively easy to solve and yet in their objective function can reflect the non-linearities of the original problem.
It is worth to present immediately here the *first-order necessary conditions* that characterized any particular local solution of the problem defined in equation (4.2).

In the specific if x^* is a local optimum of the optimisation problem and if the constraints regularity condition is fulfilled at x^*, then exist and are unique the multipliers u_j^* that together with x^* satisfy the following conditions:

$$\begin{aligned}
&1. \quad \nabla f(\boldsymbol{x}^*) - \sum_{j=1}^{m} u_j^* \nabla g_j(\boldsymbol{x}^*) = 0 && \text{Stationarity} \\
&2. \quad \forall j \in \{1, \ldots, m_e\} : \ g_j(\boldsymbol{x}^*) = 0 && \text{Primal Feasibility} \\
&3. \quad \forall j \in \{j = l, \ldots, m\} : \ g_j(\boldsymbol{x}^*) \geqslant 0 && \text{Primal Feasibility} \\
&4. \quad \forall j \in \{j = l, \ldots, m\} : \ u_j^* \geqslant 0 && \text{Dual Feasibility} \\
&5. \quad \forall j \in \{j = l, \ldots, m\} : \ u_j^* g_j(\boldsymbol{x}^*) = 0 && \text{Complementary slackness}
\end{aligned} \tag{4.5}$$

whit $l = m_e + 1$, used to simplify the notation. The first equation establishes the gradient of the Lagrangian function associated with the problem to be zero for the local solution defined by x^* and u_j^*. The second and third conditions determine that in correspondence of x^* the inequality and equality constraints must be satisfied, representing the feasibility condition of the solution. The fourth condition imposes the non-negativity of the multipliers associated with the inequality constraints. Finally, the last equation states that the multiplier of an inactive inequality constraint[1] must be zero.
These conditions ensure a solution of a nonlinear programming to be optimal and take the name of *Karush–Kuhn–Tucker conditions* (KKT), while u_j^* are known as KKT multipliers.

Regarding the sequential quadratic programming algorithm the first step, that represents also the major concern in SQP methods, involves the choice of appropriate quadratic sub-problems. A reasonable approach is the linearization of the actual constraints about the current approximation x_k. Thus the general form of the quadratic sub-problem is:

$$\begin{aligned}
\min_{d} \quad & (\boldsymbol{r_k})^T \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^T B_k \boldsymbol{d} \\
\text{subject to} \quad & \nabla h(\boldsymbol{x_k})^T \boldsymbol{d} + h(\boldsymbol{x_k}) \geqslant 0 \\
& \nabla g(\boldsymbol{x_k})^T \boldsymbol{d} + g(\boldsymbol{x_k}) = 0
\end{aligned} \tag{4.6}$$

being $d = x - x_k$. The vector r_k and the symmetric matrix B_k remain to be chosen appropriately. Anyhow the most appropriate selection for the objective function typically involves the local quadratic approximation to f at x_k. That is, B_k is taken as the Hessian and r_k as the gradient of f at x_k.

The next step instead consist in the use of the method of *Lagrange multipliers* to incorporate both the objective and the constraint functions in $\mathcal{L}(x, u)$, the Lagrangian function. In particular, as already said, the SQP method utilizes a quadratic model of the Lagrangian function as the objective function, in this way it is possible to take non-linearities in the constraints into account while maintaining the linearity of the constraints in the sub-problem. The optimisation is then

[1]The set of active constraints are the inequality constraints whose condition is satisfied with equality in the current solution x_k: $g_j(x_k) = 0$. Conversely if its condition is not satisfied with equality the constraint is defined inactive.

reformulated by applying the Lagrange multipliers to the constraints, and the purpose of the problem becomes to find the minimum of $\mathcal{L}(x, u)$, instead of $f(x)$. It is necessary to clarify the reformulated problem is indeed equivalent to the previous one, since when the constraints and the conditions expressed in equation (4.5) are satisfied minimizing $\mathcal{L}(x, u)$ or $f(x)$ is the same operation. This strategy is widely adopted to simplify the problem of searching for the extrema of a function subject to equation constraints.
It should be noted the application of the KKT conditions allows to generalize the method of Lagrange multipliers to the case of problems with inequality constraints. As a matter of fact, such method would be suitable in presence only of equality constraints.
At this point it may be convenient to highlight the distinction between equality and inequality constraints within the Lagrange function through a temporary change in notation. Specifically, for better clarity in the description, the constraints and their corresponding Lagrange multipliers will be respectively represented as follows: $h(x)$ and λ for inequalities, and $g(x)$ and σ for equalities. Moreover, for the time being, it is assumed that there are only two constraint functions present, to simplify the treatment: one for equality and one for inequality. Hence, the Lagrangian function is rewritten as:

$$\mathcal{L}(\boldsymbol{x}, \lambda, \sigma) = f(\boldsymbol{x}) - \lambda h(\boldsymbol{x}) - \sigma g(\boldsymbol{x}) \tag{4.7}$$

Given a current iterate $(x_k, \lambda_k, \sigma_k)$, the quadratic Taylor series approximation in x for the Lagrangian is:

$$\mathcal{L}(\boldsymbol{x_k}, \lambda_k, \sigma_k) + \nabla\mathcal{L}(\boldsymbol{x_k}, \lambda_k, \sigma_k)^T\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T\nabla^2\mathcal{L}(\boldsymbol{x_k}, \lambda_k, \sigma_k)\boldsymbol{d} \tag{4.8}$$

A strong motivation for using this function as the objective function in the quadratic sub-problem is that it generally ensures the algorithm to have good local convergence properties.
Hence, the algorithm defines an appropriate search direction d_k at an iterate $(x_k, \lambda_k, \sigma_k)$, as a solution to the quadratic programming sub-problem, that in light of the above can be expressed as:

$$\begin{aligned} \min_{d} \quad & f(\boldsymbol{x_k}) + \nabla f(\boldsymbol{x_k})^T\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T\nabla^2_{xx}\mathcal{L}(\boldsymbol{x_k}, \lambda_k, \sigma_k)\boldsymbol{d} \\ & \nabla h(\boldsymbol{x_k})^T\boldsymbol{d} + h(\boldsymbol{x_k}) \geqslant 0 \\ & \nabla g(\boldsymbol{x_k})^T\boldsymbol{d} + g(\boldsymbol{x_k}) = 0 \end{aligned} \tag{4.9}$$

This formulation is obtained using the Taylor expansion approximation around the current solution x. In the specific through further simple steps, not reported here for brevity, it is immediate to derive from equation (4.8) the following expression for the Lagrangian:

$$\mathcal{L}(\boldsymbol{x}, \lambda, \sigma) = f(\boldsymbol{x_k}) + \nabla f(\boldsymbol{x_k})^T(\boldsymbol{x} - \boldsymbol{x_k}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x_k})^T\nabla^2_{xx}\mathcal{L}\,(\boldsymbol{x} - \boldsymbol{x_k}) + ... \tag{4.10}$$

Then, substituting $d = x - x_k$ one can obtain the objective function expression just seen in (4.9). This formulation for the quadratic sub-problem is analogous to the more general case presented in equation (4.6). In fact, it can be noticed that the term $f(x_k)$ present in objective function may be

left out for the minimization problem, since it is constant under the $\min_d$ operator.

Therefore in the SQP method at a current iterate x_k, the step to the next iterate is obtained through information generated by solving the quadratic sub-problem. This is assumed to reflect in some way the local properties of the original problem.
It is significant to highlight that generally SQP methods do not compute the Hessian matrix, but utilize instead an appropriate approximation of it. As a matter of fact, a convenient approach involves exploring alternatives to the actual Hessian of the Lagrangian, for example, as just mentioned, approximating matrices that enable the quadratic sub-problem to be solved at any x_k, and facilitate the analysis of global convergence of the resulting algorithm. This is the principle difference between the Newton-Raphson method and the quasi-Newton algorithms, like the SQP, since the former necessarily need to compute the Hessian matrix.
In fact, if one applied the standard Newton's method to minimize the Lagrangian, described by equation (4.8), searching for the solution $\nabla\mathcal{L}(x,u)=0$, it would be necessary to iterate the following equation:

$$\begin{bmatrix} \boldsymbol{x_{k+1}} \\ \lambda_{k+1} \\ \sigma_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_k} \\ \lambda_k \\ \sigma_k \end{bmatrix} - \begin{bmatrix} \nabla^2_{xx}\mathcal{L} & \nabla h & \nabla g \\ \nabla h^T & 0 & 0 \\ \nabla g^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \nabla f + \lambda_k \nabla h + \sigma_k \nabla g \\ h \\ g \end{bmatrix} \tag{4.11}$$

where x_k, λ_k, σ_k represent the solution and the Lagrange multipliers at the k^{th} iteration, while ∇^2_{xx} denotes the Hessian matrix. Note that functions and gradients are computed at x_k, λ_k, σ_k. Moreover, the first matrix after the minus represents $\nabla^2\mathcal{L}$, while the second one is $\nabla\mathcal{L}$. This expression is analogous to the one seen in equation (4.1), but adapted for a multidimensional minimization problem.
It is evident that one of the drawbacks of the Newton-Raphson is the computational cost required to calculate the inverse of the Hessian at each iteration, which can be very high. Furthermore, this matrix can be singular, thus non-invertible, therefore in several cases the algorithm terminates and is not possible to proceed further calculating directly the next iterate. This is clearly understandable looking at equation (4.7).
Conversely the quasi-Newton methods approximate directly the inverse of the Hessian matrix and update it at each iteration, avoiding the need to compute and then invert the matrix at every step. In the specific, the various strategies employed to perform this estimation distinguish the different quasi-Newton methods. These algorithms are particularly advantageous when is impractical to compute the inverse of the Hessian or the operation is prohibitively expensive, such as for large-scale problems.
In detail, the SQP method implemented in this work employs the BFGS formula to approximate the Hessian matrix of the Lagrangian function, as it will be addressed in more detail in the next paragraph.

Hence, modifying equation (4.9) and returning to the notation used in (4.2), it is finally possible to write the sub-problem in the form that follows:

$$\begin{aligned} & \min \ \frac{1}{2}\boldsymbol{d}^T C_k \boldsymbol{d} + \nabla f(\boldsymbol{x_k})^T \boldsymbol{d} \\ \boldsymbol{d} \in \mathbb{R}^n : \quad & \nabla g_j(\boldsymbol{x_k})^T \boldsymbol{d} + g_j(\boldsymbol{x_k}) = 0, \quad j = 1, \ldots, m_e \\ & \nabla g_j(\boldsymbol{x_k})^T \boldsymbol{d} + g_j(\boldsymbol{x_k}) \geqslant 0, \quad j = m_e + 1, \ldots, m \end{aligned} \tag{4.12}$$

where $x_k \in \mathbb{R}^n$ represents the iterates, an approximation of the solution, $v_k \in \mathbb{R}^m$, an approximation of the multipliers, and $C_k \in \mathbb{R}^m$, an approximation of the Hessian of the Lagrangian function. This represents the reference quadratic sub-problem for the optimiser used in this thesis — with $m_e = 0$ — and solved to optimise the manoeuvres of interest. It is worth underline that in the presented treatment, inequality functions have always been expressed in the form $h(x) \geqslant 0$ —which is the formulation used for all constraints in this thesis—. In any case, formally speaking, it is always possible to transform any constraint $h(x) \leqslant 0$ into $h'(x) \geqslant 0$, where $h'(x) = -h(x)$.
Let d_k be the optimal solution and u_k the corresponding vector of Lagrange multipliers of this quadratic sub-problem. A new iterate is obtained by:

$$\begin{pmatrix} \boldsymbol{x_{k+1}} \\ \boldsymbol{v_{k+1}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x_k} \\ \boldsymbol{v_k} \end{pmatrix} + \alpha_k \begin{pmatrix} \boldsymbol{d_k} \\ \boldsymbol{u_k} - \boldsymbol{v_k} \end{pmatrix} \tag{4.13}$$

with $\alpha_k \in (0, 1]$ a suitable step-length parameter. For completeness, it is specified that the upper and lower bound, mentioned at the beginning of this Chapter, are expressed by:

$$\boldsymbol{x_L} - \boldsymbol{x_k} \leqslant \boldsymbol{d} \leqslant \boldsymbol{x_U} - \boldsymbol{x_k}$$

for the sub-problem. The v_k vector needs to be updated if the algorithm employs a line search approach, as this may depend on the approximation v_k of the optimal Lagrange multipliers.
In conclusion a SQP method models the nonlinear programming problem (4.2) at a given approximate solution, x_k, by a quadratic programming sub-problem (4.12). This is obtained linearizing the nonlinear constraints of (4.2) and minimizing a quadratic approximation of the Lagrangian function. The solution to this sub-problem is then used to construct a better approximation x_{k+1} and this process is iterated to create a sequence of approximations that it is hoped will converge to a solution x^*.

4.1.3 Properties Overview

SQP algorithms typically exhibit traits similar to Newton-like methods. They converge rapidly when the iterates are near the solution, but they may display erratic behavior, especially when the iterates are distant from a solution. However, incorporating constraints significantly increases the complexity of both analyzing and implementing these algorithms. Furthermore, the effectiveness of SQP methods hinges on the availability of rapid and precise algorithms for solving quadratic programs. Fortunately, these are often relatively easy to solve due to the availability of good procedures for their solution.

It is worth noting that nonlinear optimisation problems can have multiple local solutions, with the global solution being the local solution corresponding to the least value of f. SQP methods ensure convergence only to a local solution of a nonlinear problem. Therefore, they must be distinguished from algorithms aimed at finding global solutions, which operate on different principles.
Another important question is connected to the local and global convergence properties of the algorithm and its relative asymptotic rate of convergence. Assuming that the quadratic sub-problem can be solved, the sequence generated by the algorithm can then converge to a solution. In the specific, *local convergence* results proceed from the assumptions that the initial x-iterate is close to a solution x^* and that the initial C_k approximates the Hessian matrix of the problem appropriately. On the other hand, *global convergence* consists of the capacity of the algorithm to converge to some local solution starting from any arbitrary remote point. In general, SQP methods enjoy good theoretical local convergence under appropriate conditions. To ensure global convergence, a method for measuring progress toward a solution is necessary. The algorithm is then equipped with a merit function, the reduction of which indicates this progress and establishes that an acceptable step has been taken. Moreover, a procedure for adjusting the step length parameter α_k is required to guarantee the reduction of the merit function at each step.
Regarding the rate of convergence, it is possible to identify three main measures: let $\{x^k\}$ be a sequence converging to x^*, then this is said to converge *linearly* if there exist a positive constant $\xi > 1$ such that:

$$\| x^{k+1} - x^* \| \leqslant \xi \| x^k - x^* \|$$

for all k sufficiently large. The convergence is *superlinear* if there exists a sequence of positive constant $\xi \to 0$ such that the previous condition is satisfied. Finally, the sequence converges *quadratically* if there exists a positive constant ξ such that:

$$\| x^{k+1} - x^* \| \leqslant \xi \| x^k - x^* \|^2$$

In the expressions just presented, $\| \cdot \|$ denotes the 2-norm of a vector. An SQP method can achieve quadratic and superlinear convergence in many cases under suitable conditions.
The algorithm also proves to be robust to variations in the initial conditions and the parameters of the problem.

4.2 NLPQLP Optmiser

NLPQLP is a Fortran implementation of a SQP algorithm, characterised by a distributed and non-monotone line search. The code is used in academic and commercial institutions to solve smooth nonlinear programming problems of the form illustrated in equation (4.2). Furthermore, it represents the solver employed to carry out the optimisation process of the relocation manoeuvres investigated in this work.
With regard to the features of the SQP method employed by the solver as well as the assumptions required, one can clearly apply the considerations expressed in the previous section.

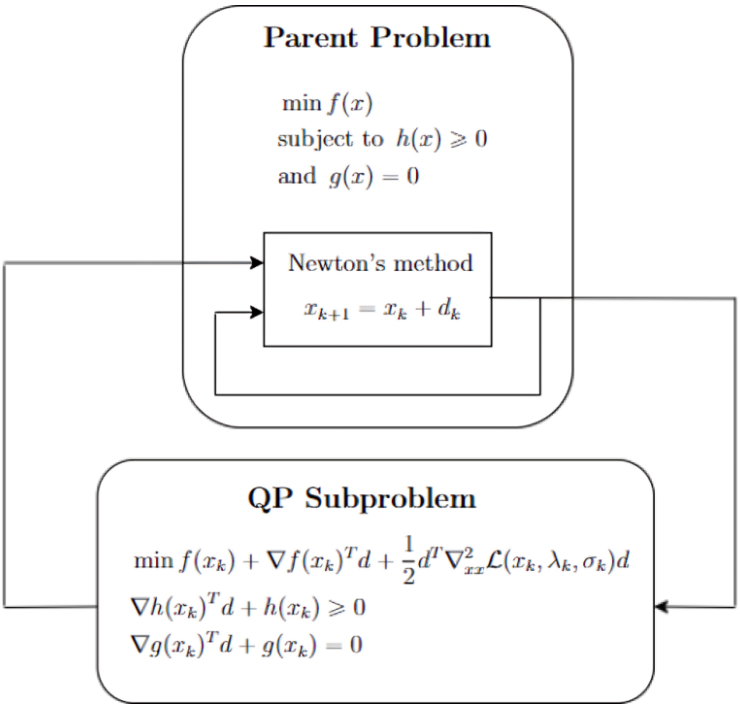

Figure 4.1: Schematic overview of the basic SQP algorithm

The optimisation performed by NLPQLP is solved under the following further assumptions, which define its domain of application:

- The problem is not too large (the problem size depends on hardware facilities)
- Functions and gradients can be evaluated with sufficiently high precision
- The problem is smooth and well-scaled

The NLPQLP code can handle both equality and inequality constraints; the latter are all expressed in the form $g(x) \geqslant 0$. It is worth noting that the functions f and g_j, $j = 1, ..., m$, need to be defined only on set $E = \{x \in \mathbb{R}^n : x_L \leqslant x \leqslant x_U\}$, since the iterates computed by the algorithm will never violate the lower and upper bounds.

4.2.1 Numerical Method

As said earlier NLPQLP, being based on a SQP method, generates a sequence of quadratic programming sub-problem, which are solved successively by a external Fortran function, known as QL. In case of computational errors as for example caused by inaccurate function or gradient evaluations, the non-monotone line search is then activated.

The algorithm build an approximation of the inverse of the Hessian matrix, updating at each iteration this quasi-Newton matrix, C_k, using the BFGS formula:

$$C_{k+1} = C_k + \frac{\boldsymbol{q}_k \boldsymbol{q}_k^T}{\boldsymbol{p}_k^T \boldsymbol{q}_k} - \frac{C_k \boldsymbol{p}_k \boldsymbol{p}_k^T C_k}{\boldsymbol{p}_k^T C_k \boldsymbol{p}_k} \tag{4.14}$$

in which:

$$\boldsymbol{q_{k+1}} = \nabla_x \mathcal{L}(\boldsymbol{x_{k+1}}, \boldsymbol{u_k}) - \nabla_x \mathcal{L}(\boldsymbol{x_k}, \boldsymbol{u_k})$$
$$\boldsymbol{p}_{k+1} = \boldsymbol{x_{k+1}} - \boldsymbol{x_k}$$

Special precautions, as proposed by Powell [6], through modifications to the standard BFGS formula, ensure that $p_k^T q_k > 0$, thereby preserving the positive definiteness of all matrices C_k throughout the iterations, assuming C_0 is positive definite. In certain instances —such as when convergence becomes exceedingly slow— additional measures like a scaling factor and restart procedure can be applied.
Although the modified BFGS formula ensures that the matrix C_k is positive definite, it is possible that equation (4.12) is not solvable due to inconsistent constraints. Therefore one can not always implement the quadratic sub-problem (4.12) as it stands. More specifically, the feasible region of (4.12) can be empty although the original problem (4.2) is solvable. The second drawback is the recalculation of gradients of all constraints at each iteration, although some of them might be inactive at an optimal solution, thus locally redundant. One possible remedy, to avoid both disadvantages, is to introduce an additional variable $\delta \in \mathbb{R}^n$, leading to a modified quadratic programming problem. In broad terms, the operation entails adding a term $1/2\rho_k\delta^2$ to the objective function and reformulating the constraints. In which ρ_k represents an additional penalty parameter designed to reduce the influence of δ on the solution. A more detailed and comprehensive description can be found in [7].
The step-length parameter α_k seen in equation (4.13) is required to enforce global convergence of the SQP method. In this way it can be reached a point satisfying the necessary Karush-Kuhn-Tucker optimality conditions when starting from arbitrary initial values. These are typically a user-provided $x_0 \in \mathbb{R}^n$ and $v_0 = 0$, while the initial estimation of the Hessian is the identity matrix, $C_0 = I$.
Moreover, α_k should satisfy at least a sufficient decrease condition of a merit function $\phi_r(\alpha)$ given by:

$$\phi_r(\alpha) = \psi_r \left(\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{v} \end{pmatrix} + \alpha \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{u} - \boldsymbol{v} \end{pmatrix} \right) \tag{4.15}$$

with a suitable penalty function $\psi_r(x, v)$. Possible choices for $\psi_r(x, v)$ are the L_1-exact penalty function:

$$\psi_r(\boldsymbol{x}, \boldsymbol{v}) = f(\boldsymbol{x}) + \sum_{j=1}^{m_e} \boldsymbol{r_j} |g_j(\boldsymbol{x})| + \sum_{j=m_e+1}^{m} \boldsymbol{r_j} |\min(0, g_j(\boldsymbol{x}))| \tag{4.16}$$

Or the augmented Lagrangian function:

$$\psi_r(\boldsymbol{x}, \boldsymbol{v}) = f(\boldsymbol{x}) - \sum_{j \in J} (\boldsymbol{v_j} g_j(\boldsymbol{x}) - \frac{1}{2} \boldsymbol{r_j} g_j(\boldsymbol{x})^2) - \frac{1}{2} \sum_{j \in K} \boldsymbol{v_j}^2 / \boldsymbol{r_j} \tag{4.17}$$

The NLPQLP version utilized for this work, implements the latter. With $J = \{1, ..., m_e\} \cup \{j : m_e < j \leqslant m,\ g_j(x) \leqslant v_j / r_j\}$ and $K = \{1, ..., m\} \backslash J$.
In the specific, r_j, $j = 1, ..., m$, is a vector of penalty parameters, governing the degree of constraint violation and penalizes the objective function as soon as an iterate leaves the feasible domain. It is updated according to a suitable rule to ensure a descent direction of d_k with respect to the chosen merit function. In more detail, it is chosen to get:

$$\phi_r'(0) = \nabla \psi_{r_k}(\boldsymbol{x_k}, \boldsymbol{v_k})^T \begin{pmatrix} \boldsymbol{d_k} \\ \boldsymbol{u_k} - \boldsymbol{v_k} \end{pmatrix} < 0 \tag{4.18}$$

The implementation of a line search algorithm is a critical issue when employing a nonlinear programming method. As a matter of fact, it greatly influences the overall efficiency of the resulting code. On one hand, a line search is crucial for stabilizing the algorithm; on the other hand, it is undesirable to consume excessive computational resources with numerous function calls. Additionally, the behavior of the merit function can become irregular in constrained optimisation scenarios. Even the implementation can be complex, if linear constraints and bounds of the variables are to be satisfied during the line search.
Usually, the step-length parameter α_k is chosen to satisfy the Armijo condition:

$$\phi_r(\sigma \beta^i) \leqslant \phi_r(0) + \sigma \beta^i \mu \phi_r'(0) \tag{4.19}$$

with constants μ, β, σ that are from the ranges $0 < \mu < 0.5$, $0 < \beta < 1$ and $0 < \sigma \leqslant 1$. The starting point is $i = 0$, and then it is increased until the equation (4.19) is satisfied for the first time, at a generic i_k. Hence, the correct step-length is obtained as: $\alpha_k = \sigma \beta^{i_k}$.
Usually, the test parameter μ ensuring a sufficient descent property that meets the Armijo condition is quite small. Meanwhile, the selection of the reduction parameter β should be adjusted to the current slope of the merit function. If β is too small, the line search terminates rapidly, but this frequently leads to excessively small step sizes, thereby increasing the number of outer iterations required. Conversely, selecting a larger value of β approaching one necessitates a greater number of function calls during the line search. It is then necessary a compromise, which is obtained by first applying a polynomial interpolation, generally a quadratic one, and use (4.19) only as a stopping criterion. It is immediate to get the minimizer of the quadratic interpolation by:

$$\bar{\alpha}_i = \frac{0.5\, \alpha_i^2\, \phi_r'(0)}{\alpha_i \phi_r'(0) - \phi_r(\alpha_i) + \phi_r(0)} \tag{4.20}$$

since α_i is the actual iterate of the line search procedure and $\phi_r(0)$, $\phi_r'(0)$ and $\phi_r(\alpha_i)$ are given. Then the maximum between $\bar{\alpha}_i$ and the Armijo parameter is taken as a new iterate. This operation is necessary to avoid irregular values, since $\bar{\alpha}_i$ can be outside of the feasible domain. If a sufficient decrease is not achieved after a preset number of iterations, it is required to terminate the algorithm.

If the number of iterations becomes too high, the line search is repeated using a different stopping criterion. Instead of fulfilling equation (4.19), the process continues as soon as it is satisfied:

$$\phi_{r_k}(\alpha_k) \leqslant \max_{k-p(k)\leqslant j\leqslant k} \phi_{r_j}(0) + \alpha_k \, \mu \phi'_{r_k}(0) \tag{4.21}$$

where p_k is a predetermined parameter: $p(k) = \min\{k, p\}$, with p a given tolerance. Therefore, an increase of the merit function value is permitted in a certain error situation. To implement the non-monotone line search, clearly a set of merit function values from previous iterations is demanded. Anyhow, monotone line searches are performed as long as they terminate successfully, while only in case of error, a non-monotone approach is employed.
The use of non-monotone line search is in general advantageous to determine the step size or step-length along the search direction of a SQP method. Indeed, it allows to improve performance and convergence properties of the algorithm. Traditional monotone line search methods strictly enforce a decrease in the objective function at each iteration. Conversely, non-monotone line search methods, as just mentioned, are more flexible, permitting occasional increases in the objective function by considering a history of function values over multiple iterations.

4.2.1.1 QL Subroutine

The computational challenges associated with solving the quadratic sub-problems are significant, and their solutions directly impact the overall performance of the SQP method. The technical details of their resolution are outside the scope of and will not dealt with here. However it is appropriate to briefly mention at least the main features of the *QL* subroutine, as it is part of the NLPQLP code. The Fortran subroutine QL is used to solve strictly convex quadratic programming problems, implementing the primal-dual method of Goldfarb and Idnani, whose specifics can be found in [8]. Initially, a Cholesky decomposition of C_k is computed by an upper triangular matrix R such that $C_k = R^T R$. This operation is made possible by the positive definite property of C_k, ensured by the modified BFGS formula, as already stated. Successively, violated constraints are added to an active set. At each step, the minimizer of the objective function subject to this new set is computed. The algorithm terminates when an optimal solution, that satisfies all linear constraints and bounds, is obtained.

4.2.2 Program Features

NLPQLP code is structured in form of a Fortran subroutine, characterized by several inputs that must be provided by the user. Its sole output is an integer, indicating the status of the optimisation process. This communicates whether the operation was successful or provides information about the cause which led to its termination. Nonlinear problem functions and analytical gradients are provided within the calling program through reverse communication. In general, the main data passed to the subroutine include:

- Objective function value and number of solution variables
- Upper and lower bounds of the sub-problem

- Constraint values and number of constraints
- Gradient of the constraint and objective function values
- Additional data, such as final accuracy, upper bound for the number of function calls during the line search, minimum step-length value, maximum number of outer iterations —one iteration corresponds to one formulation and solution of (4.12)—, etc.

As mentioned earlier, if the optimality conditions cannot be met within a user-defined tolerance, the code will report the encountered error. The most common errors are the following:

- The algorithm terminates when it reaches the maximum number of iterations specified by the user.
- The line search algorithm stopped because the user-provided maximum number of sub-iterations was exceeded.
- The search direction d_k, is close to zero, but the current iterate is still infeasible. It suggests issues with poorly scaled problem functions.

4.2.3 Performance Overview

The updated version of NLPQLP was tested on a dataset comprising 306 problems, employing a stopping tolerance of 10^{-7}. As stated in [9] it achieved a success rate in problem resolution exceeding 90%. The numerical results obtained indicated stability and robustness of the algorithm for this set of standard test problems. The code was also extensively tested on problems with up to 100 variables. In general, the analysis performed demonstrated the efficiency and reliability of the algorithm and its capability to handle problems with numerous constraints. The integration of the non-monotone line search represents a notable enhancement in performance compared to the monotone approach. The non-monotone procedure allows for efficient problem solving even in scenarios characterized by highly noisy function values and numerical differentiation challenges. Moreover, internal restarts further bolster stability by addressing severe errors encountered during the computation of the search direction, attributable to inaccurate derivatives. Scaled restarts assume critical importance, particularly when the convergence of the SQP method slows considerably. This typically occurs in instances involving poorly scaled variables or functions, inaccurate derivatives, or imprecise solutions of (4.12). In such circumstances, deviations in the search direction adversely affect the update process (4.14), leading to progressively inaccurate quasi-Newton matrices C_k.

Further information about the performance of the code can be found in [9] and [7]. In particular, the latter provides a comparison of NLPQLP with other nonlinear programming codes, highlighting that SQP codes result to be the most efficient, followed by the generalized reduced gradient, multiplier, and penalty methods.

Chapter 5

Methodology

"Success is not final, failure is not fatal: It is the courage to continue that counts."
— W. Churchill

In this chapter, the methodology used to formulate the optimisation problem is outlined. Specifically, the approach employed to mathematically express the constraints associated with the physical problem of relocation is introduced. Additionally, the processes for computing both the constraints and objective function values, along with their corresponding gradient function values — which are essential for the NLPQLP code to execute the optimisation — are illustrated. Finally, a comprehensive overview of the key concepts related to the implementation of the NLPQLP code is provided.

The optimisation problem involves finding the best feasible manoeuvre plan for the relocation process, considering the single extraction and insertion phases individually. In formulating the problem, the main focus will be directed towards the case of continuous manoeuvres, which have been found to be the most time-demanding to study. The discussion is further extended to the study of impulsive manoeuvres, highlighting the main differences in the case. In the following paragraphs, the problem and methodology are defined, with reference, for simplicity, to the relocation of a single satellite. Nonetheless, such considerations can be extended in a similar manner to address constellations composed of multiple satellites. The relocation operation theoretically consists of purely tangential manoeuvres in the East-West direction. Therefore, North-South manoeuvres, characterized by out-of-plane thrusts, will not be taken into account. More specifically, North-South and East-West manoeuvres will be considered completely decoupled for the purposes of the discussion.

5.1 Generalities

The vectorial differential equations describing the motion of the satellite are:

$$\frac{d\boldsymbol{r}}{dt} = \boldsymbol{V} \tag{5.1}$$

$$\frac{d\boldsymbol{V}}{dt} = -\frac{\mu_\oplus}{r^2}\frac{\boldsymbol{r}}{r} + \boldsymbol{a_p} + \frac{\boldsymbol{T}}{m} \tag{5.2}$$

Here, r represents the position vector of the satellite relative to Earth, V denotes the velocity vector of the spacecraft, T signifies the thrust vector, and $\mu_\oplus$ stands for the Earth's gravitational parameter. Finally, a_p indicates accelerations due to external perturbations affecting the satellite orbit. Specifically, perturbative accelerations arising from the following effects are considered:

- Non-spherical Earth's gravitational attraction
- Moon gravity
- Sun gravity
- Solar radiation pressure

The orbit is propagated by integrating the differential motion equations of the satellite, considering the manoeuvres performed and the perturbations listed above. This operation is highly complex and exceeds the scope of this thesis; therefore, it will not be explored here. Propagating the spacecraft trajectory is crucial for estimating the orbital elements after executing manoeuvres. A dedicated library provided by GMV has been used for propagation, employing an *8th Order Runge-Kutta Integration method* to solve the integration. For the estimation of non-spherical Earth gravity, an expansion of the Legendre polynomial to the 8th order has been utilized.

Similarly, GMV's source codes have been used for estimating and calculating the number and epochs of eclipses. The procedures for calculating umbra and penumbra events are not relevant to understanding the conducted work. Furthermore, they are characterized by a high level of complexity; hence, they will not be discussed.

One can adopt a spherical coordinate system to describe the position and velocity vectors of the satellite in the inertial frame, based on the equatorial plane of the central body, i.e., the Earth. The position vector of the spacecraft is given by:

$$\boldsymbol{r} = \begin{bmatrix} r \\ \lambda \\ \theta \end{bmatrix} \tag{5.3}$$

where r is the distance from the Earth, while λ and θ are the satellite longitude and latitude respectively. The spacecraft's velocity vector consists of the following components:

$$\boldsymbol{V} = \begin{bmatrix} V_r \\ V_\theta \\ V_\varphi \end{bmatrix} \tag{5.4}$$

where V_r, V_θ and V_φ are respectively the radial, tangential and normal components of the velocity. The acceleration vector expressed with respect to the velocity vector is:

$$\boldsymbol{a} = \begin{bmatrix} a_N \\ a_V \\ a_W \end{bmatrix} \tag{5.5}$$

It is worth underlining that for the problem formulation, the maneuver duration has been expressed in days, while the ΔV of the impulsive maneuvers in km/day.

5.2 Problem Formulation

It is assumed that the mass of the reference satellite remains constant throughout the entire relocation process. This simplifies the problem without compromising the validity and accuracy of the results. The assumption is particularly appropriate given the nature of the relocation manoeuvres, which require very low thrust, delta-V, and propellant consumption. Consequently, the mass of propellant consumed — representing the mass variation during the manoeuvres — can be considered negligible compared to the total weight of the satellite.
In accordance with the features of the thrusters equipped on the satellite, the thrust generated by the propulsion system is constant during each manoeuvre and is the same across all of them.
As discussed in Chapter 3, acceleration is defined as the ratio of thrust to mass. Therefore, based on the information just presented, it turns out to be purely tangential and constant. Thus, the acceleration vector can be expressed as:

$$\boldsymbol{a} = \begin{bmatrix} 0 \\ a_V \\ 0 \end{bmatrix} \tag{5.6}$$

The tangential component a_V — that as stated in Chapter 3 is aligned with the velocity direction— is a constant and is the only non-null component. The out-of-plane component of the acceleration, a_W, is equal to zero, as North-South manoeuvres are not executed. Also the radial component a_R is null, since the main objective is to change the semi-major axis and in case of continuous thrust eccentricity can be easily control through the tangential burns.
The solution variables of the optimisation are represented by the duration of the manoeuvres and their *midpoints*. The latter, calculated as the middle epoch between the start and end times of a manoeuvre, provide temporal information regarding the placement of manoeuvres within the planning process. Let n be the number of manoeuvres required for the extraction or insertion phase, then the solution vector can be expressed as:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{2n} \end{bmatrix} \tag{5.7}$$

where the first n values represent the duration of each manoeuvre, while the successive n denote their respective midpoints. With the acceleration fixed, these variables uniquely determine the set of manoeuvres necessary to change the satellite orbit, defining their temporal sequence and magnitude. Another fundamental piece of information to identify a manoeuvre is its ΔV, but this is completely dependent on the duration, due to the constant acceleration.

In fact, the ΔV of the i^{th} manoeuvre can be written as:

$$\Delta V_i = a_V \cdot x_i \quad \forall i = 1, ..., n \tag{5.8}$$

from which it is evident that duration and ΔV convey the same information. In any case, when dealing with continuous thrust, the most reasonable and practical choice is generally to refer to duration rather than delta-V. The conversion between them is then straightforward, as just seen.

5.2.1 Constraints

The constraints characterizing the optimisation arise from orbital considerations and operational requirements, additionally, manoeuvres must be scheduled to avoid eclipse periods, introducing further limitations. From a mathematical standpoint, each constraint is represented by a constraint function $g_j(x)$, expressed as an inequality $g(x) \geqslant 0$. As illustrated in the previous chapter, given an approximate solution x_k, the NLPQLP subroutine requires as input the results of the constraint functions at the current iterate $g_j(x_k)$ to carry out the optimisation. This implies the need to store such values in one vector, thus given an iterate x_k:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} g_1(\boldsymbol{x_k}) \\ g_2(\boldsymbol{x_k}) \\ \vdots \\ g_m(\boldsymbol{x_k}) \end{bmatrix} \tag{5.9}$$

in which m is the total number of constraints of the problem, y is referred to as the vector of constraints and its generic component y_j represent the value of the constraint function g_j subject to the solution vector x.
It is advisable for expositional clarity to underline the distinction between orbital, operational, and eclipse-related constraints, decomposing the constraint vector y into three additional vectors: u, v and w. This approach allows for a more structured and organized description of these elements in the upcoming subsections. Furthermore, it is consistent with the same strategy followed for the Fortran implementation of the optimisation problem. Therefore:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix} \tag{5.10}$$

The components of the vector u, v and w are the values of the functions $g(x)$ corresponding to the orbital, operational, and eclipse separation constraints, respectively.

5.2.1.1 Orbital Constraints

The orbital constraints stem from the necessity of achieving the correct final orbit with appropriate precision after the manoeuvre phase. This is accomplished by controlling the following mean equinoctial elements:

1. *Longitude* ($\boldsymbol{\lambda}$)
2. *x component of Eccentricity the Vector* ($\boldsymbol{e_x}$)
3. *y component of Eccentricity the Vector* ($\boldsymbol{e_y}$)
4. *Drift* ($\boldsymbol{D}$)

Inclination control is not necessary, as East-West manoeuvres do not modify it. In any case, this parameter will clearly experience variations during the process due to the external perturbations. Nevertheless, these changes will be corrected by specific North-South manoeuvres that does not take part in the relocation process. The required control actions on the above mentioned parameters are expressed mathematically through orbital constraint functions $g_j(x)$, $j = 1, ..., m_a$. However deriving the equation of these functions is not straightforward, as it will be explained later. Therefore, only the process for computing the components of the orbital vector u will be described here, presenting the final formulas applied to obtain the mathematical values stored in it:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m_a} \end{bmatrix} \tag{5.11}$$

with m_a representing the number of orbital constraints.
For each orbital parameter, a target value is provided by the user, along with the respective tolerance required to achieve it. The orbital constraints impose that for the generic parameter, the difference between the obtained actual value and the target must be less than or equal to the maximum admissible error established for it, which corresponds to its tolerance. Mathematically, $g_j(x)$ is represented by the sum of the tolerance and the difference between the desired value for the element and its final value at the end of the default time window for manoeuvres:

$$g_j(\boldsymbol{x}) = \pm(actual - target) + toll \geqslant 0$$

The difference is considered with both positive and negative signs to account for the absolute value of the attained error, which can be either positive or negative. Consequently, two constraints are defined for each orbital element: if the error exceeds the tolerance, one constraint will always be satisfied, but the second one will not. In the equation of $g(x)$, only the final value achieved by the orbital parameter is a function of x, as it depends on the executed manoeuvres. The first step to compute this value is to utilize the solution variables of x to define the orbital manoeuvres for extraction or insertion.

Specifically, given the generic i^{th} manoeuvre, its start time is calculated as:

$$t_{\text{start}_i} = x_{n+i} - 0.5 \cdot x_i \tag{5.12}$$

while its end time is:

$$t_{\text{end}_i} = x_{n+i} + 0.5 \cdot x_i \tag{5.13}$$

With these pieces of information, it is then possible to propagate the satellite orbit considering the presence of manoeuvres. Finally, starting from the state vector, all the equinoctial elements are computed. It is evident that the dependence of these values on x is highly nonlinear, intricate, and complex, making it extremely challenging to obtain an equation for $g(x)$, which is why it cannot be reported.
For the insertion phase all the equinoctial elements illustrated at the beginning of the paragraph are controlled, thus the orbital constraint vector u becomes:

$$u_{1,2} = f_\lambda \left(\pm\Delta\lambda + toll_\lambda\right) \tag{5.14}$$

$$u_{3,4} = f_e \left(\pm\Delta e_x + toll_e\right) \tag{5.15}$$

$$u_{5,6} = f_e \left(\pm\Delta e_y + toll_e\right) \tag{5.16}$$

$$u_{7,8} = f_D \left(\pm\Delta D + toll_D\right) \tag{5.17}$$

where $\Delta*$ denotes the difference between the target and achieved values, $toll_*$ represents the tolerance, and f_* is a multiplicative factor. This factor, as will be explained in paragraph 5.4, is employed if necessary to enhance the final accuracy of the results.
For the extraction phase, the control of the longitude is not present; in this case, components $u_{7,8}$ replace $u_{1,2}$:

$$u_{1,2} = f_D \left(\pm\Delta D + toll_D\right) \tag{5.18}$$

the expression (5.14) is no longer required, while the components $u_{3,4}$ and $u_{5,6}$ remain unchanged. Consequently, the size of the orbital vector u is $m_a = 8$ for insertion and $m_a = 6$ for extraction. The distinctions in constraints come from inherent differences between the two manoeuvre phases, each with its own unique objectives. Specifically, in extraction, it is crucial to achieve the drift orbit with high accuracy in terms of semi-major axis while controlling eccentricity. Needless to say, respecting the target drift in this phase is paramount, as the main aim is to leverage the altitude differences —precisely expressed by the drift— between the final and GEO orbits to change the satellite operative longitude. During insertion, semi-major axis —still represented by the drift— and eccentricity are managed to attain the correct geostationary altitude. However, it is essential to introduce the constraint on the final target longitude in this phase, as it constitutes the ultimate goal of the relocation activity. Achieving the target longitude with precision at the end of the process is fundamental to avoid costly subsequent corrective actions.
It is worth noting that the inclusion of tolerance in formulating orbital constraints simplifies their fulfillment and speeds up the method convergence. Mathematically, as the algorithm requires $g(x) \geqslant 0$, the addition of tolerance, which is a positive value, facilitates the satisfaction of this inequality. In essence, achieving the target value with a certain margin of error is evidently easier than achieving it precisely, thereby allowing for a broader range of admissible values as outcomes of $g(x)$.

5.2.1.2 Operational Constraints

The operational constraints are typically related to both the characteristics of the thrusters and the needs of satellite operators. Taking into account the low-thrust performances of electric thrusters, as well as their associated platform restrictions and limitations, along with practical operational requirements, the following set of constraints can be delineated:

1. *Minimum separation between manoeuvres*: the interval time between the end of one manoeuvre and the start of the subsequent one must not be less than the user-defined minimum value.
2. *Minimum manoeuvre duration*: the duration of each manoeuvre must be greater than the user-provided minimum duration.
3. *Start times within the analysis interval*: the start times of each manoeuvre must fall within the analysis interval preset for the manoeuvre operation
4. *End times within the analysis interval*: the end times of each manoeuvre must fall within the analysis interval preset for the manoeuvre operation.
5. *Consecutive manoeuvres*: manoeuvres must occur in consecutive temporal order.

This is applied to the optimisation problem and is representative of the basic constraints generally required for a satellite equipped with electric thrusters. However, this set can be modified if further or different restrictions are requested by satellite operators. In fact, operational constraints are typically more susceptible to changes than orbital ones, which tend to be less specific since they depend only on the manoeuvre phase — extraction or insertion .
Given the solution vector x, and denoting the i^{th} manoeuvre with the subscript i, the inequality functions used for express mathematically the aforementioned constraints are:

1. *Minimum separation between manoeuvres*:

$$g_j(\boldsymbol{x}) = (x_{n+i+1} - 0.5 \cdot x_{i+1}) - (x_{n+i} + 0.5 \cdot x_i) - c_1 \geqslant 0 \tag{5.19}$$

   with $i = 1, ..., n-1$, $j = 1, ..., n-1$ and c_1 representing the minimum separation value.

2. *Minimum manoeuvre duration*:

$$g_j(\boldsymbol{x}) = x_i - c_2 \geqslant 0 \tag{5.20}$$

   with $i = 1, ..., n$, $j = n-1, ..., 2n-1$ and c_2 denoting the minimum required duration.

3. *Start times within the analysis interval*:

$$g_j(\boldsymbol{x}) = (x_{n+i} - 0.5 \cdot x_i) - c_3 \geqslant 0 \tag{5.21}$$

   with $i = 1, ..., n$ and $j = 2n-1, ..., 3n-1$, while c_3 is start epoch of the analysis interval

4. *End times within the analysis interval*:

$$g_j(\boldsymbol{x}) = c_4 - (x_{n+i} + 0.5 \cdot x_i) \geqslant 0 \tag{5.22}$$

with $i = 1, ..., n$, $j = 3n - 1, ..., 4n - 1$ and c_4 being the end epoch of the analysis interval

5. *Consecutive manoeuvres*:

$$\begin{cases} g_j(\boldsymbol{x}) = (x_{n+i+1} + 0.5 \cdot x_{i+1}) - (x_{n+i} - 0.5 \cdot x_i) \geqslant 0 \\ g_{j+1}(\boldsymbol{x}) = (x_{n+i+1} + 0.5 \cdot x_{i+1}) - (x_{n+i} + 0.5 \cdot x_i) \geqslant 0 \end{cases} \tag{5.23}$$

with $i = 1, ..., n - 1$ and , $j = 4n - 1, ..., 5n - 2$

The values of these functions, subject to the current iterate x_k, are inserted into the operational constraints vector v:

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{m_b} \end{bmatrix} \tag{5.24}$$

the size of which will consequently be $m_b = 6n - 3$, representing the number of operational constraints. The user defines the analysis interval by specifying the start and end epochs of the temporal window designated for the execution of the manoeuvres. The parameters c_1, c_2, c_3 and c_4 are the same for each j, yet there exists the option to initialize distinct values of c_1 and c_2 based on the specific manoeuvre or pair of manoeuvres under consideration. It is important to note that the requirement concerning the consecutiveness of manoeuvres implies two constraint values for each j. Its associated equations ensure that the end time of a manoeuvre must occur temporally after the start and end times of the previous one. In the formulations reported above, the start and end times for manoeuvres are computed as depicted in equations (5.12) and (5.13), respectively.
It is worth to present the approach used to get the mathematical values stored in v on Fortran, which has required to handle separately the computation of the first $n - 1$ elements —that are the constraints on the minimum separation— from the others. This can be illustrated in a simple way by partitioning the operational constraints vector into two further vectors v' and v'':

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v'} \\ \boldsymbol{v''} \end{bmatrix} \tag{5.25}$$

Where v' is an $(n - 1) \times 1$ vector, representing the restrictions on the the minimum manoeuvre separation, while the vector v'' of $5n - 2$ components, denotes all the remaining constraints. The calculation of the elements of v' follows the formulation seen in equation (5.19), thus that expression is simply applied for each i^{th} value:

$$v'(i) = (t_{\text{start}_{i+1}} - t_{\text{end}_i}) - c_1 \qquad \text{with} \quad i = 1, ..., n - 1$$

with t_{start_i} and t_{end_i} the start and end time of i^{th} manoeuvre. More interesting to depict, is the formulation on Fortran for v''. This consists in the construction of a $(5n-2) \times 2n$ matrix and a $(5n-2) \times 1$ vector, utilized to rearrange the system of expressions expressed by (5.37), (5.21), (5.22) and (5.23) into matrix form as:

$$\boldsymbol{v''} = A \cdot \boldsymbol{x} - \boldsymbol{b} \tag{5.26}$$

In more detail, b represents the constant terms present in such equations:

$$\boldsymbol{b} = \begin{bmatrix} c_2 \cdot \mathbf{1_n} \\ c_3 \cdot \mathbf{1_n} \\ -c_4 \cdot \mathbf{1_n} \\ \mathbf{0_p} \end{bmatrix} \tag{5.27}$$

with $p = 2n - 2$ and 1_n denoting the vector of size n consisting solely of ones. The matrix A, in order to enhance clarity, can be viewed as the union of the two matrix A' and A'':

$$A = \begin{bmatrix} A' \\ A'' \end{bmatrix}$$

with A' having dimension $3n \times 2n$ and representing the coefficients of the solution variables in (5.37), (5.21) and (5.22):

$$A' = \begin{bmatrix} \mathrm{I}_n & 0_n \\ -0.5 \cdot \mathrm{I}_n & \mathrm{I}_n \\ -0.5 \cdot \mathrm{I}_n & -\mathrm{I}_n \end{bmatrix} \tag{5.28}$$

where I_n and 0_n are respectively the identity and zero matrix of dimension n. While A'' has size $(2n-2) \times 2n$ and incorporates the coefficients of x present in (5.23) and (5.24). The matrix is partially reported in the following figure.

As can be easily verified, by applying (5.26), the respective expressions for $g(x)$ shown previously are obtained. The choice to employ the matrix representation of equation (5.26) to handle the calculations of the constraint functions subject to x, makes the process more streamlined and systematic. Furthermore, it facilitates the incorporation of any modifications or additions to the constraint formulation.

5.2.1.3 Separation from Eclipses

As mentioned in Chapter 1, eclipses represent challenging periods during which the satellite typically cannot execute manoeuvres, or its thrust capabilities are limited. Regarding the optimisation problem, it is imposed to not perform manoeuvres during eclipses and the surrounding periods. This restriction entails the introduction of specific constraints to avoid the overlap between manoeuvres and the umbra and penumbra phases.

$$A'' = \begin{bmatrix} \overbrace{0.5 & 0.5 & 0 & \dots & 0}^{n} & \overbrace{-1 & 1 & 0 & \dots & 0}^{n} \\ -0.5 & 0.5 & 0 & \dots & 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0.5 & 0.5 & \dots & 0 & 0 & -1 & 1 & \dots & 0 \\ 0 & -0.5 & 0.5 & \dots & 0 & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -0.5 & 0.5 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}$$

Figure 5.1: Matrix A'' representing the coefficients of x in consecutiveness of maneuvers constraint expressions

The complete equation for the functions $g(x)$ associated with these constraints is not directly provided, since it is clearer to present only the individual steps followed to calculate their values. Furthermore, the equations for these functions are quite complex and not easily expressed in mathematical terms. Additionally, they would require intricate notation to concisely be represented in an unique formula. The constraint values are stored, as mentioned at the beginning of this section, in the vector w:

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{m_c} \end{bmatrix} \tag{5.29}$$

with $m_c = n$ representing the number of constraints needed to enforce the separation between manoeuvres and eclipses, which precisely matches the number of manoeuvres.
Once the analysis interval is defined, the first step involves computing the eclipses caused by the Earth and the Moon, determining the number and duration of these events. Specifically, a list is generated containing the start and end times of each eclipse that occurs during the manoeuvre window. Moreover, users have the option to apply a margin to these times, either advancing or delaying the eclipse dates respectively, thereby extending its actual duration. This margin can be different for the start and end epochs of the eclipse.
Subsequently, for each i^{th} manoeuvre, the duration of the overlap between it and each of the eclipses in the list is estimated. This operation is repeated for all manoeuvres, and the results are inserted into a dedicated vector w'. Referring to the i^{th} manoeuvre, n_e components of w' corresponding to the given manoeuvre are obtained, with n_e representing the number of eclipses. The size of this vector is consequently $n_e \cdot n$, and its components are:

$$\boldsymbol{w'_{j+(i-1)\cdot n_e}} = \max(e_{\text{start}_j}, x_{n+i} - 0.5 \cdot x_i) - \min(e_{\text{end}_j}, x_{n+i} + 0.5 \cdot x_i) \tag{5.30}$$

with $j = 1, ..., n_e$, $i = 1, ..., n$ and where e_{start_j} and e_{end_j} are the start and end epochs of the j^{th} eclipse. Given a specific manoeuvre, it can be easily verified that if there is an overlap between

it and a generic eclipse, a negative value representing the duration of the overlap is computed and stored in w'. Conversely, in the case of no overlap, a positive value is obtained, equal to the time separation between the manoeuvre and the eclipse.
At the end, for each manoeuvre, only the minimum value of its corresponding n_e components of w' is selected and stored in w. In practice, considering the i^{th} manoeuvre, the most negative value, indicating the maximum duration overlap, is inserted into the constraint vector:

$$w_i = \min(w'_j) \qquad \text{with} \quad j = 1 + (i-1) \cdot n_e, ..., i \cdot n_e \tag{5.31}$$

with $i = 1 \div n$. It is important to emphasize that each component of w is linked to a specific manoeuvre and if one of them does not overlap with any eclipses in the list, then the time separation between it and the nearest eclipse is recorded. Since this value is positive, as inferred from (5.30), it indicates that the respective manoeuvre satisfies the eclipse separation constraint.
However, a geostationary satellite generally encounters only one eclipse by Earth per day, being each umbra periods separated by about 24 hours. Consequently, since the duration of a single manoeuvre is typically less than a day, this usually overlaps with only one eclipse. Therefore, the maximum overlap value for a specific manoeuvre stored in w actually represents its unique overlap.
Finally, it is worth noting that e_{start_j} and e_{end_j} are dependent on the satellite's orbit, which in turn relies on the manoeuvres executed, and thus on the solution variables x. Hence, to be precise, the eclipse start and end epochs can be expressed as a function of x, although this operation would be particularly complex.

5.2.2 Objective Function

The objective function of the problem, is represented by the summation of manoeuvre durations:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i \tag{5.32}$$

The choice of such a function aims to minimize propellant consumption of the mission, and thus to reduce operational costs. As a matter of fact, minimizing the sum of manoeuvre durations, while maintaining the same propellant flow, results in a decrease in the total mass of propellant burned, m_p :

$$m_p = \dot{m}_p \, t$$

where t represents the duration of each manoeuvre. If m_p is reduced, this results in a lower satellite mass at launch for the same operational lifespan. Alternatively, it allows for a longer operational life or to enable a larger number of manoeuvres executable for the same amount of onboard propellant. It is noteworthy that optimising either the duration or the delta-V yields equivalent results for continuous manoeuvres, as their acceleration is assumed constant.

5.2.3 Gradient Functions

As seen in Chapter 4, to formulate the quadratic sub-problem, it is necessary to calculate the gradients of the objective and constraint functions. Therefore, finite differences, in the specific central differences, are employed to approximate the derivatives of these functions. Thus, considering the objective function and $i = 1, ..., 2n$, its derivatives are obtained as:

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} \approx \frac{f(x_1, \ldots, x_i + h, \ldots, x_{2n}) - f(x_1, \ldots, x_i - h, \ldots, x_{2n})}{2h} \tag{5.33}$$

While, considering $j = 1, ..., m$, the derivatives of the constraint functions become:

$$\frac{\partial g_j(\boldsymbol{x})}{\partial x_i} \approx \frac{g_j(x_1, \ldots, x_i + h, \ldots, x_{2n}) - g_j(x_1, \ldots, x_i - h, \ldots, x_{2n})}{2h} \tag{5.34}$$

In this way it is possible to estimate all derivatives and utilized them to get the gradients $\nabla f(x)$ and $\nabla g(x)$. Hence, in reference to the objective function, its gradient is represented by the following vector:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_{2n}} \end{bmatrix} \tag{5.35}$$

While the gradient of the constraint function is constituted by the matrix:

$$\nabla g = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_{2n}} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_{2n}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_{2n}} \end{bmatrix} \tag{5.36}$$

It is noted that the arguments of the functions have been omitted in (5.35) and (5.36) to streamline notation.
The NLPQLP subroutine, requires the values of these gradient functions subject to x, to perform the optimisation. Therefore, in line with the above-mentioned, given the current iterate x_k the parameter h is used to increase and decrease its solution variables. Equations (5.33) and (5.34) approximate properly the derivatives of a function only whether the parameter h is close to zero or in any case sufficiently small. For this reason the value chosen for the computation is:

$$h = 10^{-6}$$

In more detail the forward and backward solution vector values of x^k are computed, by respectively adding and removing h from a single element x_i^k of x^k. Subsequently, the associated cost function and constraints values are calculated and the operation is repeated for all the solution variables contained in x, varying i from 1 to $2n$.
Equations (5.39) and (5.33) are applied, once determined the solution vectors of the i^{th} forward and backward steps, to get the objective functions values. Finally using (5.33) the components of $\nabla f(x)$ are estimated.
In a similar manner forward and backward solution vector values, are used to compute the constraint function values by following the methodology illustrated in the previous subsections. Hence the vector y is obtained for the forward and backward steps. Denoting these by y^{fw} and y^{bw} respectively, it is applied the following expression to get the (i, j) element of $\nabla g(x)$:

$$\frac{y_j^{fw} - y_j^{bw}}{2h} \qquad \text{with} \quad j = 1, ..., m$$

In this manner the i^{th} column of ∇g is initialised and repeating the process for $i = 1, ..., n$ the complete matrix is obtained.

5.3 Formulation for Impulsive Manoeuvres

The optimisation problem has been initially formulated and studied only concerning continuous manoeuvres and electric propulsion. This work strategy was based on the fact that continuous manoeuvres, due to their nature, typically prove to be more complex to investigate. Indeed, the prolonged duration of these manoeuvres, which may extend beyond a single orbit, along with the electric propulsion system's limitations, make their constrained optimisation difficult. Additionally, as they involve an high number of constraints, the correct implementation of the SQP algorithm becomes more challenging.
However, once valid results were obtained, the implementation of the method was further validated by extending it to the case of chemical propulsion, and thus to impulsive manoeuvres. The problem formulation is very similar to the one just described, but there are clearly some crucial differences. Since these manoeuvres, as the name suggests, consist of short impulsive accelerations that produce an instantaneous velocity variation—indicated as ΔV—discussing durations loses meaning.
For this reason, it is arbitrarily assumed that each manoeuvre is characterized by a constant duration of 1 second, which is approximately $1.157 \cdot 10^{-5}$ days. This value is small enough to consider the manoeuvres as impulsive, while also allowing for simplification of the study. In fact, using a unitary value for the durations results in the velocity variations and acceleration having the same numerical value. Referring to the i^{th} manoeuvre, this characteristic is immediately understandable from the following expression:

$$\Delta V_i = a_i \cdot t_i = a_i \cdot 1 \qquad \text{with} \quad i = 1, ..., n$$

where a_i, ΔV_i, and t_i respectively denote the acceleration, velocity variation, and duration of the i^{th} manoeuvre. It is evident that a and ΔV convey the same information, as they both define

the intensity of the manoeuvre. Thus, these two parameters can be arbitrarily used to identify the manoeuvres, but since ΔV proves to be a more representative measure of a propulsor's effect, it is chosen as the primary parameter for the problem formulation.
Consequently, the velocity variations generated by the manoeuvres and their midpoints become the solution variables of the optimisation in the impulsive case. The first n components of the solution vector x are now represented by ΔV and no longer by the manoeuvre durations.
Given that the durations remain constant during the optimisation, these new solution variables still uniquely define the set of manoeuvres.

5.3.1 Constraints

The orbital, operational, and eclipse constraints and their formulation are basically the same as in the continuous case. The only difference lies in the operational constraint related to the minimum duration of the manoeuvres, which obviously becomes irrelevant. In its place, an analogous constraint concerning the minimum ΔV is defined. Its equation becomes:

$$g_j(\boldsymbol{x}) = a_i \cdot x_i^{aux} - c_5 \geqslant 0 \tag{5.37}$$

where $i = 1, ..., n$, $j = n-1, ..., 2n-1$, a_i represents the acceleration of the i^{th} manoeuvre in kilometers per day, and c_5 denotes the minimum ΔV required by the user for each manoeuvre. Therefore, the matrix A and the vector b discussed in subsection 5.2.1.2 can be appropriately modified:

$$\boldsymbol{b} = \begin{bmatrix} c_5 \cdot \mathbf{1_n} \\ c_3 \cdot \mathbf{1_n} \\ -c_4 \cdot \mathbf{1_n} \\ \mathbf{0_p} \end{bmatrix}$$

$$A' = \begin{bmatrix} D_n & 0_n \\ -0.5 \cdot \mathrm{I}_n & \mathrm{I}_n \\ -0.5 \cdot \mathrm{I}_n & -\mathrm{I}_n \end{bmatrix}$$

where D_n is a diagonal matrix of dimension $n \times n$, presenting the manoeuvre acceleration values a_i on its diagonal.
The other equations for the constraints illustrated in the previous paragraph can still be considered valid, provided that the vector x used for calculations is changed. Indeed, the solution vector can no longer be utilized; instead, a vector containing the duration of the manoeuvres —besides their midpoint — must be employed:

$$\boldsymbol{x^{aux}} = \begin{bmatrix} f \cdot \mathbf{1_n} \\ x_{n+1} \\ \vdots \\ x_{2n} \end{bmatrix} \tag{5.38}$$

in which the first n components precisely represent the durations in days. In particular, 1_n is a vector of size n consisting of only ones, and $f = 1/86400 = 1.157407 \cdot 10^{-5}$ is the factor representing the number of days in a second.
In this manner, by using the auxiliary vector x^{aux} instead of x, all the formulations and expressions from section 5.2.1 can still be employed for the impulsive case. The start and end times of the manoeuvres are still computed using equations (5.12) and (5.13) with x^{aux}. These pieces of information, along with the new solution variables, are utilized to determine the manoeuvres and propagate the satellite orbit when computing the orbital and eclipse separation constraints.
Although chemical thrusters do not rely on the power provided by solar panels to generate thrust, it is still good practice to avoid performing manoeuvres during these periods. For this reason, the constraint for eclipse separation is maintained, and as mentioned earlier, it remains analogous to the continuous case.

5.3.2 Gradient and Objective Functions

The method applied for computing gradient values is the same as introduced in section 5.2.3. The objective function of the optimisation problem for impulsive manoeuvres is instead characterized by the summation of the absolute values of ΔV:

$$f(\boldsymbol{x}) = f_F \cdot \sum_{i=1}^{n} |x_i| \tag{5.39}$$

where $f_F = 0.001$ is a factor applied to facilitate the convergence of the algorithm. Unlike durations, velocity variations can be positive or negative—corresponding to manoeuvres towards the east or west direction, respectively. Therefore, the use of absolute values is essential to correctly minimize the delta-V.
However, the purpose of optimisation, depicted by the cost function, remains unchanged: to reduce operational costs by minimizing the consumed propellant mass. As seen in Chapter 3, minimizing the ΔV required to perform the manoeuvre results in a lower propellant mass consumed.

5.3.3 Radial Cross-Coupling

Considering the high-thrust performance of chemical thrusters and the short duration of their manoeuvres, meeting constraints such as eclipse separation becomes less complex. However, the impulsive nature of the burn also presents some drawbacks, such as the inability to consider the ΔV produced by the manoeuvres as purely tangential. In reality, the thrust applied to the satellite by the propulsion system is not perfectly tangential; components in other directions are always present.

In practical terms, the error effect of thrust application in the radial direction in terms of ΔV is usually considered for relocation operations carried out through impulsive manoeuvres. This entity is referred to as *radial cross-coupling* and represents the maximum admissible ΔV in the radial direction. Specifically, it is defined as the ratio, expressed as a percentage, between the delta-V applied in the radial direction and the applied tangential delta-V. The user can determine this

parameter, and there is the option to define different values for it when considering manoeuvres towards the east or west direction.

For continuous manoeuvres, there is no need to consider the radial cross-coupling since the thrusts are much smaller, resulting in significantly lower errors. Additionally, the overall effect is diminished because the long duration of the manoeuvre allows errors to compensate over the orbit. Conversely, for impulsive manoeuvres, the high intensity of the thrust produced at a single point in the orbit results in a significant error that must be taken into account.

5.4 Implementation

In order to implement the optimisation problem described in the previous paragraphs using Fortran, a comprehensive review of several Fortran modules from GMV's corporate source code was necessary. Specifically, since the starting point for the relocation optimisation process relies on the analytical solution, the modules responsible for generating it were studied. This analytical solution serves as the initial guess for the optimiser, and its derivation follows a different strategy depending on whether continuous or impulsive manoeuvres are conducted. In both cases, equations from [2] are applied to obtain an analytical estimation of the total ΔV required to achieve the target orbit. These approximated orbital mechanics equations focusing on the geostationary orbit are not included here, as they were not analyzed or delved into detail for the purposes of this thesis.
In the impulsive case, the total ΔV is then subdivided into three manoeuvres separated by 12 hours, following a series of steps that cannot be disclosed due to confidentiality concerns.
For continuous manoeuvres, the analytical solution is computed by strategically distributing the total Delta-V across the minimum number of full orbits, resulting in a set of manoeuvres equally spaced temporally.
In both cases, the procedure involves an iterative process that cyclically seeks to obtain a solution satisfying orbital and operational constraints.
The Fortran implementation of the SQP algorithm for the optimisation problem of relocation manoeuvres carried out by the NLPQLP subroutine required the development of four modules:

- *Module* 1: which initializes all the arguments required for constrained optimisation. It contains subroutines to initialize the input values necessary for calling the NLPQLP subroutine.
- *Module* 2: it calculates the values of orbital, operational, and eclipse separation constraints, as well as the cost function value. It also computes the corresponding gradient values for these functions.
- *Module* 3: it calls the subroutines from the previous two modules to carry out the optimisation process. It implements the NLPQLP solver, and based on the result provided by it, recalculates the constraints and cost function or the gradient functions, or both, for the next execution. The process continues until convergence or the maximum number of executions of optimiser is reached.
- *Module* 4: which handles and generates manoeuvres. It contains general subroutines for checking for overlap between manoeuvres in a given list, separating manoeuvres if there is overlap

between two or more of them, reordering manoeuvres chronologically based on the start date of the manoeuvre, etc.

The first module is responsible for assigning values to vectors and matrices, and allocating and deallocating local memory. The second module implements the equations formulated in the previous paragraph. It includes several corporate subroutines essential for propagating the satellite orbit, computing the eclipse list, and performing other operations. The third module constitutes the proper implementation of the NLPQLP subroutine, following the guidelines outlined in [9]. Essentially, the module recursively calls the NLPQLP optimiser, updating the constraint, gradient or cost function values for the next execution. Its functioning is schematically summarized in Algorithm 1. The last module is essential for creating the variables containing the information required to represent continuous and impulsive manoeuvres in Fortran. The need for functions to detect overlapping manoeuvres and separate them arises from the nature of the NLPQLP subroutine. This code represents a mathematical abstraction of a real physical problem, therefore it could potentially generate infeasible solutions from the physical standpoint. In the specific during optimisation, overlapping manoeuvres may be generated. Although these consist in a solution that does not fulfill the constraints and is ultimately discarded at the end of the process, they are utilized during the current execution to compute orbital constraints. Hence, without separating these manoeuvres during this operation, inevitable crashes in the corporate code responsible for orbital propagation would occur. Consequently, the subroutines of the fourth module generate an additional manoeuvre characterized by an acceleration equal to the sum of the overlapping manoeuvres within that segment, in place of each overlapping region.

There is the option to optimise only the single extraction or insertion manoeuvre, or alternatively, the full relocation process consisting of extraction and insertion phases sequentially performed. Although the theory behind the computation of the full relocation or the single extraction-insertion manoeuvres is the same, two different subroutines are used for practical reasons. In particular, in the full relocation case, the subroutine must verify that the user-provided extraction target drift is sufficiently high to perform the relocation within the preset temporal window, computing a new value if necessary.

The module responsible for optimisation can be invoked either at the first iteration or at the last iteration of the algorithm that generates the analytical solution. It is evident that in the latter case, starting from a better solution that may already satisfy the constraints results in a higher probability of convergence for the optimiser.

It is worth noting that in the adopted implementation of the SQP algorithm, there is no provision for changing the number of solution variables in the optimisation operation. Therefore, during this process, the size of the solution vector x remains unchanged and equal to that of the initial guess provided by the analytical method. This means that in order to find the best feasible manoeuvre plan, the optimiser only has the possibility to vary the temporal arrangement and duration of burns, while the number of manoeuvres considered to perform the relocation operation will not be modified.

Algorithm 1 Implementation of the NLPQLP subroutine for nonlinear constrained optimisation (*Module* 3)

Require: initial guess list of manoeuvres and information about the relocation operation
Require: orbital targets, orbital and operational constraints data
Require: user-defined maximum number of executions and outer iterations of the optimiser
Require: indication about the relocation phase, whether it is extraction or insertion
Require: indication about the type of manoeuvres, whether they are impulsive or continuous

```
Compute solution variables related to the initial guess manoeuvres
Compute orbital, operational, and eclipse constraint values subject to the initial iterate
Compute the objective function value subject to the initial iterate
Compute constraint and objective functions gradient values subject to the initial iterate
Set lower and upper bounds of the nonlinear optimisation problem
Use information obtained through the previous steps to initialize input vectors and matrices to pass to the NLPQLP subroutine
loop
    Call the NLPQLP subroutine
    Store the current iterate in the solution vector
    if optimisation failed and new objective function and constraint values are required then
        Re-compute objective function and constraint values subject to the variable values of the current iterate
    else if optimisation failed and new gradient values are required then
        Re-compute gradient values subject to the variable values of the current iterate
    else if optimisation did not converge then
        Re-compute objective function, constraint, and gradient values subject to the variable values of the current iterate
        Update the number of executions of the optimiser: i = i + 1
        if i exceeds the preset maximum number of executions of the optimiser then
            Exit
        end if
    else if optimisation converged then
        Exit loop
    end if
end loop
Generate the solution by creating a new list of manoeuvres using the information stored in the final iterate, which represents the optimal solution variables
return The new list of optimised manoeuvres and information about the convergence status
```

Chapter 6

Results

"Yesterday is history, tomorrow is a mystery, but today is a gift.
That is why it is called the present"
— Anonymous

In this Chapter, the main results obtained from the implementation of the NLPQLP code for the constrained optimisation process of relocation manoeuvres will be illustrated. The nature of the problem and its implementation, characterized by numerous degrees of freedom, enable the simulation of a wide range of scenarios. More specifically, a practically infinite number of results can be obtained, as it is possible to test different case studies by simply changing the orbital target, operational requirements, epochs defining the manoeuvre window, or the starting orbit. Additionally, the great flexibility of the optimiser implementation must be taken into account. Indeed, the NLPQLP subroutine can be called at the first or last iteration of the analytical method, allowing for the opportunity to follow two different approaches for the analysis.
Therefore, in the next paragraphs, only some of the possible attainable results are presented, which can be considered the most representative for the investigated problem.

Before entering the particulars of the solutions, it proves convenient to specify some further aspects of the methodology followed.

The starting orbit for the relocation and extraction phases is the geostationary orbit, which is also the arrival orbit for the insertion phase. Anyhow, as already mentioned in this thesis, the GEO orbit is an ideal representation of a real orbit. In the specific, it is a theoretical abstraction used to define an orbit with particular characteristics, which is, however, achieved only momentarily in reality. For this reason, the orbital elements describing the reference GEO orbit in this Chapter will not be exactly the same as those seen in Tab. 2.1, but they will present a certain degree of error due to the presence of perturbations and real effects that slightly alter them.

The spacecraft equipped with electric thrusters considered for the study of continuous manoeuvres represent actually existing satellites, and their respective data will be used to compute the corresponding solutions reported in the next sections. However, for privacy reasons, this information cannot be made public. Nonetheless, for the sake of consistency, one can refer to the general characteristics illustrated in Tab. 6.1, which approximate the real satellite data. These will be utilized to make practical considerations about the propellant mass savings achieved through optimisation.

Nominal Thrust (T)	Specific Impulse (I_s)	Propellant Flow ($\dot{m}_p$)	Satellite Mass (m_0)
0.4 N	1500 s	0.05 g/s	4000 kg

Table 6.1: Data for electrically propelled satellites

The analysis of impulsive manoeuvres, as mentioned in the previous Chapter, has been conducted solely to certify the validity of the method and its implementation, and to explore the possibility of extending its application to satellites employing chemical propulsion systems. Therefore, it has not been possible to refer to currently operational spacecraft, and as stated in Chapter 5, these manoeuvres are assumed to be simple impulsive burns lasting one second. Nevertheless, for basic propulsion-related evaluations, it is supposed that the spacecraft mass remains consistent with the electric propulsion scenario and that hydrazine is used as fuel. Thus, the general characteristics for hydrazine thrusters provided in [2] can be utilized, which are shown in Tab. 6.2.
For chemical propulsors, the computation of the propellant flow rate becomes more complex, as it would require the use of pressure and temperature data and the application of an analytical model to estimate it realistically at each instant of time. Hence, for simplicity this entity is not evaluated, in addition it is easier to refer to the Tsiolkovsky equation for estimating the propellant consumption in case of impulsive manoeuvres.

Nominal Thrust (T)	Specific Impulse (I_s)	Satellite Mass (m_0)
20 N	300 s	4000 kg

Table 6.2: Data for chemically propelled satellites

The solutions to be presented have been attained by leveraging GMV's corporate platform, ***FocusSuite®***, which was employed to conduct all the necessary simulations. The GMV's relocation manoeuvre planning library for geostationary satellites, containing modules for various flight mechanics operations such as orbit propagation, is written in Fortran 90. This also applies to the modules responsible for executing the optimisation process and computing the analytical solution used as an initial guess by the optimiser, as already mentioned. Consequently, the corporate platform generates results by executing the binary file produced by compiling the Fortran 90 source code on the *Linux* operating system through the *GNU* Fortran compiler (*gfortran*).

The orbital targets, along with their relative tolerances, and the operational constraints, are defined within the same platform. Specifically, a uniform tolerance value has been employed across all simulations—both for continuous and impulsive manoeuvres—to indicate the accuracy required to achieve orbital targets. These values are presented in the subsequent table.

D (deg/day)	λ (deg)	e
$0.1 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$	$0.1 \cdot 10^{-3}$

Table 6.3: Maximum admissible errors for the orbital targets

Additionally, to streamline the study, identical operational requirement data for continuous manoeuvres have been applied throughout all corresponding analyses. Details regarding these parameters are outlined in the table below.

Continuous case options	
Maximum Duration:	65000 s
Minimum Duration:	720 s
Minimum Separation:	4200 s
Minimum Separation to Eclipses:	300 s

Table 6.4: Operational requirements for continuous manoeuvres

Conversely, given their different nature, distinct operational data have been utilized for impulsive manoeuvres. Nevertheless, these requirements remain consistent across all simulations involving such manoeuvres. The corresponding numerical values are provided in the following table.

Impulsive case options	
Maximum ΔV:	6 m/s
Minimum ΔV:	0.001 m/s
Minimum Separation:	3600 s
Minimum Separation to Eclipses:	300 s

Table 6.5: Operational requirements for impulsive manoeuvres

The values for the operative requirements and orbital target tolerances outlined above have been selected arbitrarily, while still choosing realistic values that roughly reflect parameters commonly used for manoeuvres planning in relocation operations. Some of these variables depend on satellite requirements, additionally they clearly vary based on the specifics of the case and on the considered mission scenario. Hence, it becomes evident that identifying specific values universally applicable is not feasible.

Finally, the settings options used for the optimisation process need to be defined. These are depicted in Tab. 6.6, which reports the values employed for the conducted analyses. The following notation is used:

- l - number of available processors, i.e., number of simultaneous function evaluations
- acc - final accuracy for the optimisation
- $acc_{_{QP}}$ - tolerance for the QP solver
- $stpmin$ - minimum step-length value
- n_{fun}^{max} - maximum number of function calls during the line search
- n_{iter}^{max} - maximum number of outer iterations for the NLPQLP code
- n_{exc}^{max} - maximum number of executions (NLPQLP routine calls) during the optimisation
- n_{analyt}^{max} - maximum number of iterations for the analytical method

l	acc	$acc_{_{QP}}$	$stpmin$	n_{fun}^{max}	n_{iter}^{max}	n_{exc}^{max}	n_{analyt}^{max}
1	10^{-5}	10^{-9}	10^{-10}	50	30	8	50

Table 6.6: Optimisation process configuration specifics

It is worth noting that one iteration corresponds to one formulation and solution of the quadratic programming sub-problem, while one execution indicates one call of the NLPQLP subroutine itself. More specifically, the iterations are directly performed within one call of the NLPQLP subroutine internally.

The presentation of the solutions in the following paragraphs outlines and summarizes, in chronological order, the strategy employed for testing the optimiser and simulating the diverse scenarios under analysis. This entails starting with extraction for continuous manoeuvres, followed by the insertion and full relocation phases, all within the realm of electric propulsion. Subsequently, the validation process extends the investigation to encompass these three phases in the context of chemical propulsion, i.e., impulsive manoeuvres.

6.1 Extraction with Continuous Manoeuvres

The optimisation of extraction manoeuvres for electric propelled satellites has been the first chronological step for the implementation of the NLPQLP code. Consequently, the extraction has been the first phase to be simulated and tested on the corporate platform ***FocusSuite***, to verify the Fortran formulation of the physical problem and the correctness of the optimisation process.
The application of the optimiser for these manoeuvres served as the foundation for its entire implementation and subsequent adjustments were mainly limited to adapt it for the single insertion and full relocation cases. Notably, the extraction phase demanded the most time and effort, as it necessitated a comprehensive integration of the SQP algorithm and the development of all the associated Fortran modules illustrated in the previous Chapter.

Given the absence of orbital constraints related to longitude control during this phase, the optimiser's convergence is faster and easier to achieve compared to both single insertion and full relocation scenarios. Additionally, this operation proves to be simpler and more straightforward to simulate. In contrast, full relocation is a more intricate and complex process—comprising extraction and insertion carried out sequentially—which inevitably demands a longer simulation time. As for insertion, as one might infer, it is necessary to simulate an extraction phase beforehand, thereby prolonging the analysis and reducing immediacy in its simulation.
In light of the above, due to the ease of testing extraction, a higher number of case studies for this operation will be illustrated here, allowing also to present general concepts about relocation manoeuvres and highlighting specific characteristics of the optimisation process. For sake of brevity, the concepts explained in this Section will not be reiterated in the subsequent paragraphs.

In further detail, in this Paragraph, the optimal manoeuvre plan solutions for three defined extractions will be reported. The relevant information regarding the initial orbital state and the analysis interval is summarized in Tab. 6.7, while the selected targets are delineated in Tab. 6.8. It is important to underline that these targets are required to be reached on the date representing the end of the manoeuvre window.

	Operation 1	Operation 2	Operation 3
Initial longitude - λ_0 (deg)	148.0	45.6	148.0
Initial semi-major axis - a_0 (km)	42164.7	42164.8	42164.7
Initial drift - D_0 (deg/day)	0.024	0.0024	0.024
Initial eccentricity - e_0	0.001	0.0001	0.001
Simulation start time	2024/06/02-00:00	2024/09/14-00:00	2024/06/02-00:00
Simulation end time	2024/06/06-00:00	2024/09/19-00:00	2024/06/08-00:00

Table 6.7: Analysed extraction operations, initial state

	Operation 1	Operation 2	Operation 3
Longitude target - λ_t (deg)	-	-	-
Drift target - D_t (deg/day)	3.0	-3.0	5.0
Eccentricity target - e_t	0.0	0.0	0.0
Manoeuvre window start time	2024/06/03-00:00	2024/09/16-00:00	2024/06/03-00:00
Manoeuvre window end time	2024/06/05-00:00	2024/09/18-00:00	2024/06/07-00:00
Operation Type	Extraction	Extraction	Extraction

Table 6.8: Analysed extraction operations, set targets

Before illustrating the results for these three precise operations, it is convenient to present the orbital solution attained using the optimiser for a generic extraction to a drift orbit characterized by $D = 3$ deg/day and zero eccentricity. This general example allows for a better practical understanding of concepts related to the relocation process, specifically the exploitation of the drift orbit, which have been described purely theoretically in previous chapters. However, for this introductory case study, the specific details regarding the optimisation process itself will not be addressed.
Firstly, looking at Fig. 6.1-6.2 it is possible to appreciate the general evolution of the mean orbital elements, which may be considered representative for every generic extraction operation. More specifically, in Fig. 6.1 one can observe the variation in the orbit drift longitude, achieved through the planned manoeuvres. The time execution of the latter can be identified in the approximately linear segments of the curve. Conversely, the mean drift remains practically constant in the periods where no manoeuvres are performed. Thus the plot shows how the the target value is reached: through several burns, in this case four, resulting in a gradual and step-wise change of D.
A drift orbit characterized by a positive drift rate, means that the satellite has a higher orbital velocity than the Earth's rotation, thereby allowing for increasing its longitude over time. Clearly, such drift orbit is characterised by a lower semi-major axis than the GEO one, thus to achieve it is necessary to reduce a, as evidenced by Fig. 6.2. As already stated in Chapter 2 a lower altitude orbit results in a shorter orbital period and therefore a higher orbital velocity, thus in a positive drift rate.
For this reason the semi-major axis evolution, depicted in this graphic follows a pattern analogous but antithetical to the drift, which also confirms that the two elements convey the same orbital information.
Regard the mean longitude this is simply the integral over time of the drift, thus, when the latter is nearly constant, the longitude increases linearly with a slope equal to the drift rate. This characteristic is perfectly visible in Fig. 6.1, represented by the linear curve in the final stretch of the graphic. Therefore, once the drift orbit, i.e. the drift target, is reached, the longitude tends to increase linearly, allowing to change the geostationary satellite position with respect the Earth. Hence, as discussed in Chapter 2, the satellite's revolutions in the extraction orbit, which consists in the drift phase, are leveraged to modify the longitude. Then, at the correct moment the spacecraft will return to the geostationary orbit via insertion operation, but in a new operational longitude compared to the beginning of the process. It is important to highlight that the mean longitude is the unique orbital elements not directly changed by the manoeuvres. Conversely, as just explained

its variations are obtained indirectly by exploiting the drift orbit characteristics.

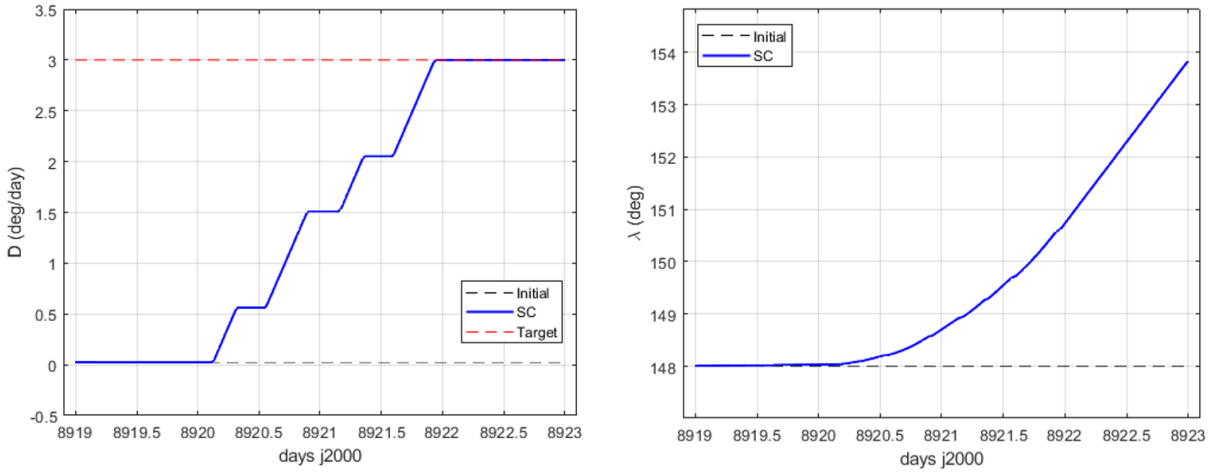

Figure 6.1: Mean drift longitude (left) and mean longitude (right) evolution during extraction

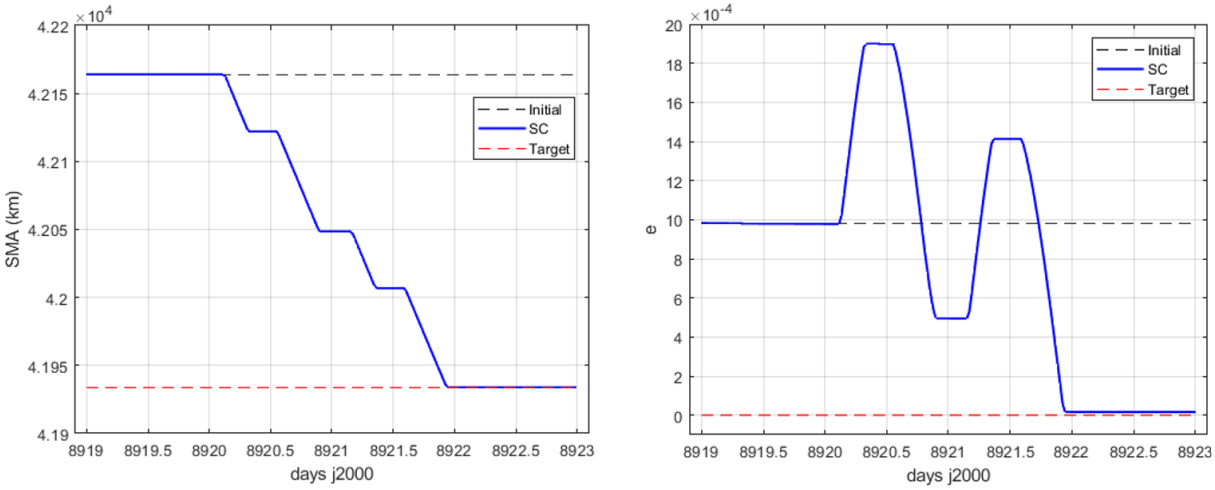

Figure 6.2: Mean semi-major axis (left) and mean eccentricity (right) evolution during extraction

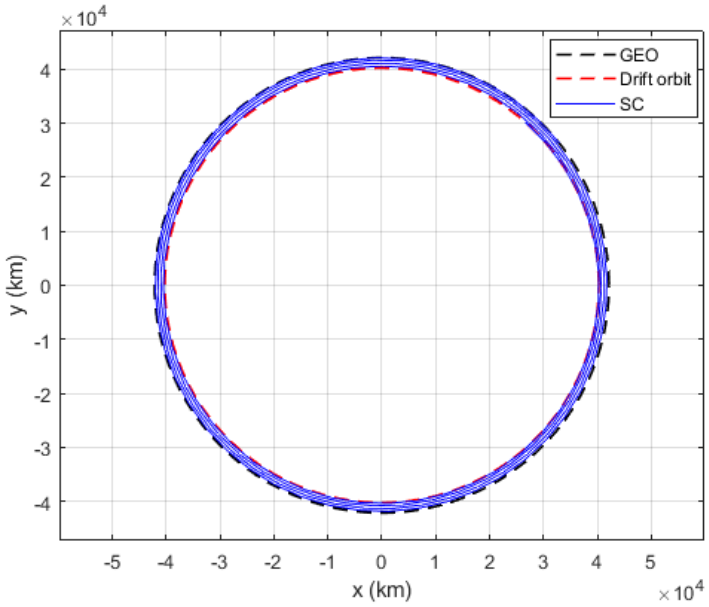

Figure 6.3: Satellite's trajectory during extraction

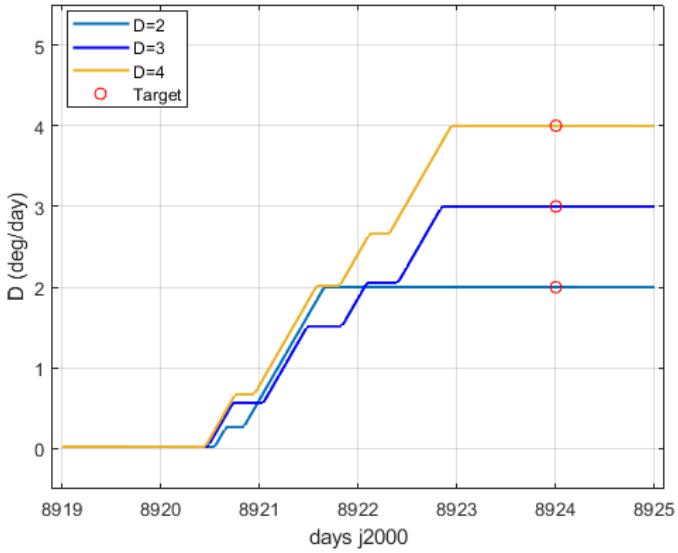

Figure 6.4: Mean semi-major axis evolution during extraction for different drift targets

The mean eccentricity variation is represented in Fig. 6.2, this element remains approximately null during the entire extraction process. However, it is interesting to note how manoeuvres change it: generally an even number of burns are performed, so that the variation caused by one manoeuvre is counterbalanced by that of the next one. The entity and nature of the eccentricity modifications generated by the tangential manoeuvres depend on in which spatial point during the orbit the latter are executed. In any case, it can be stated that generally an higher number of manoeuvres results in easier capacity in properly controlling this parameter. The Fig. 6.3, although may not be very informative, provides an idea of the general trajectory followed by the satellite to reach the drift orbit. The graphic does not accurately reflect the correct proportions; in fact, the difference between the starting geostationary orbit and the extraction arrival orbit has been exaggerated for exposition clarity.
However, it is possible to appreciate the modality with which the target orbit is reached, after several prolonged orbits. In conclusion in Fig. 6.4 different extraction operations characterised by distinct drift targets values are represented. In can be inferred that the curve pattern are similar, and obviously, the higher the target drift to achieved, the more costly is the operation, resulting in a greater number of manoeuvres or longer duration. This holds true whether the semi-major axis is raised or lowered; the only difference is the sign of the thrust imparted by the thrusters, respectively in East or West direction.

6.1.1 Operation 1

This case study is particularly interesting because the analytical method is unable to find a solution that meets the imposed constraints within 50 iterations, while the optimiser manages to achieve it.

The Tab. 6.9 displays information regarding the manoeuvre lists of the solutions obtained at the end of both the analytical and numerical methods. It is worth noting that the analytical solution shown in it serves as the initial guess for the NLPQLP solver, as reiterated several times in this document. The table allows for an immediate comparison between the two solutions generated by their respective methods, highlighting particularly that the total durations of the manoeuvres undergo minimal variations between the two cases. This characteristic is generally valid when the optimisation process is executed in the final iteration of the analytical method. In fact, in other results that will be presented later in this Chapter, the same behavior can be noted, confirming this observation.

The analytical method thus succeeds to find a solution that is sufficiently good in terms of the overall duration of manoeuvres but fails to fulfill all the imposed constraints. In the specific, the constraint on the target eccentricity of the arrival orbit is not satisfied at the end of the extraction. As can be seen in Tab. 6.10, the error in reaching the corresponding target value exceeds the imposed tolerance of 10^{-4}.

The limitation of the analytical solution lies in attempting to control eccentricity by always executing two pairs of manoeuvres, each pair consisting of two identical manoeuvres, as is evident looking at Tab. 6.9. In contrast, the NLPQLP algorithm explores a broader range of possible solutions during optimisation, identifying a manoeuvre plan capable of satisfying all constraints. Specifically, the major flexibility of the algorithm allows for finding a a solution that better controls eccentricity, thus respecting the corresponding constraint.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/03-03:34:37	2024/06/03-08:23:26	West	17329	-1.522
2024/06/03-13:41:40	2024/06/03-22:12:26	West	30647	-2.691
2024/06/04-03:30:41	2024/06/04-08:19:30	West	17329	-1.522
2024/06/04-13:37:44	2024/06/04-22:08:30	West	30647	-2.691
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/03-03:40:51	2024/06/03-08:32:04	West	17473	-1.534
2024/06/03-13:29:45	2024/06/03-21:59:12	West	30567	-2.684
2024/06/04-03:37:29	2024/06/04-08:27:10	West	17381	-1.526
2024/06/04-13:26:14	2024/06/04-21:55:01	West	30527	-2.681

Table 6.9: Operation 1, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.338 \cdot 10^{-7}$	N/A	$0.224 \cdot 10^{-3}$
Numerical	$0.881 \cdot 10^{-4}$	N/A	$0.165 \cdot 10^{-4}$

Table 6.10: Operation 1, orbital targets errors at the end of the manoeuvre period

Observing the start and end times of the manoeuvres, it is possible to appreciate how the optimiser manages to temporally rearrange them, refining the solution provided by the analytical method. As mentioned earlier, the total duration of the manoeuvres does not experience a significant reduction, but a decrease of approximately 4 seconds is still achieved. Clearly, given the constrained nature of the problem and its physical characteristics, as well as the limited two-day time window for manoeuvres, it leads to the feasible solution space becoming highly restricted. In addition, considering the good quality of the initial solution, it is easy to understand why variations in terms of manoeuvres midpoints are on the order of tens of minutes and not higher.

It goes without saying that reducing the total duration of manoeuvres results in a proportional decrease in the sum of their ΔV. As already observed, the delta-V is merely the product of acceleration by duration, and since the former is assumed to be constant, variations in duration directly and proportionally impact ΔV.

In Fig. 6.5-6.6-6.7-6.8, the evolution of the orbital elements during the extraction operation within the analyzed interval is depicted, aligning with overall trends observed in the illustrative case seen previously. Eccentricity is displayed in a polar geometry graph, which offers a comprehensive view by simultaneously presenting the variation of both x and y components of the vector e. The effect of each manoeuvre on eccentricity is evident and all the burns are precisely timed to achieve the target eccentricity by the end of the extraction process. Changes in other orbital elements are represented using both mean and osculating elements; notably, the former provide a clearer perspective on the orbit's evolution, while the latter indicate instantaneous parameter values. All the general orbit considerations stated at the beginning of the Paragraph, remain still valid.

These trends and Tab. 6.10, which summarizes the final errors encountered in reaching the orbital targets, confirm the validity of the solution obtained at the end of the optimisation. Although the reduction in manoeuvre times is very modest, the most interesting result is the NLPQLP code's ability to find a satisfactory solution able to meet the numerous imposed constraints.

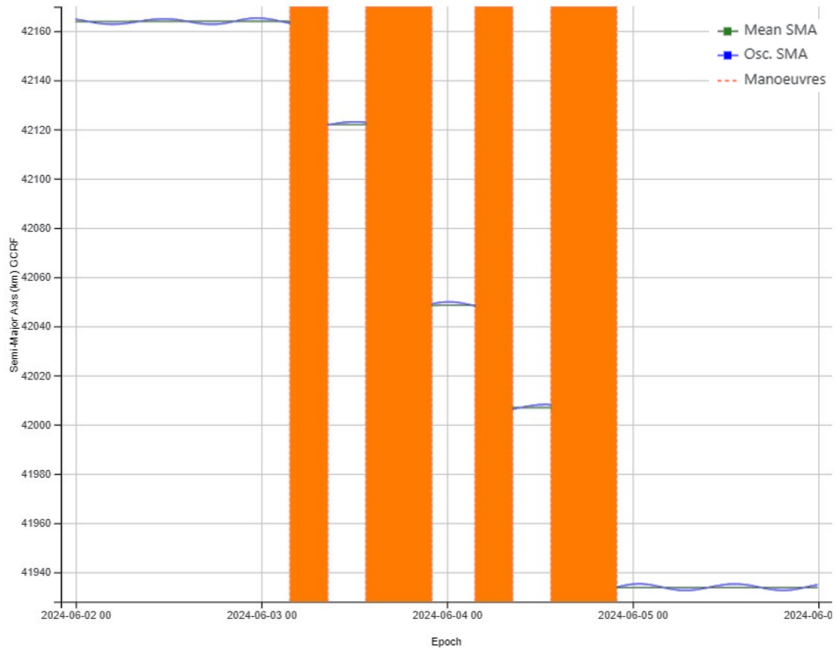

Figure 6.5: Operation 1, semi-major axis evolution

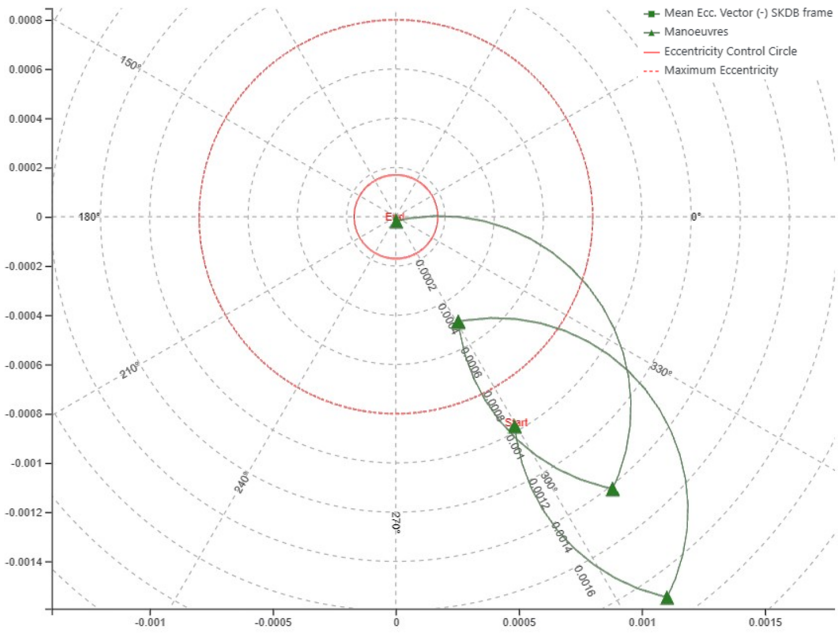

Figure 6.6: Operation 1, mean eccentricity vector evolution

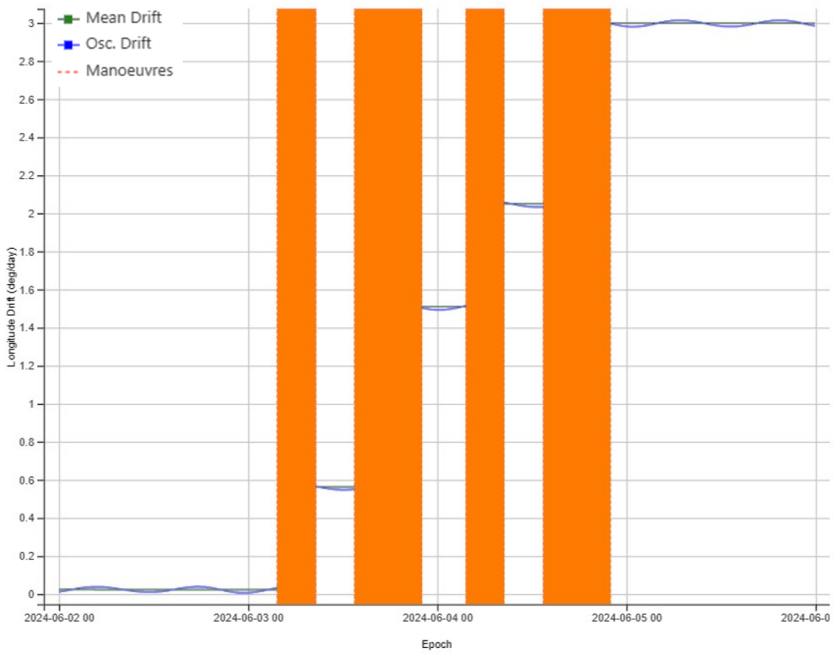

Figure 6.7: Operation 1, drift evolution

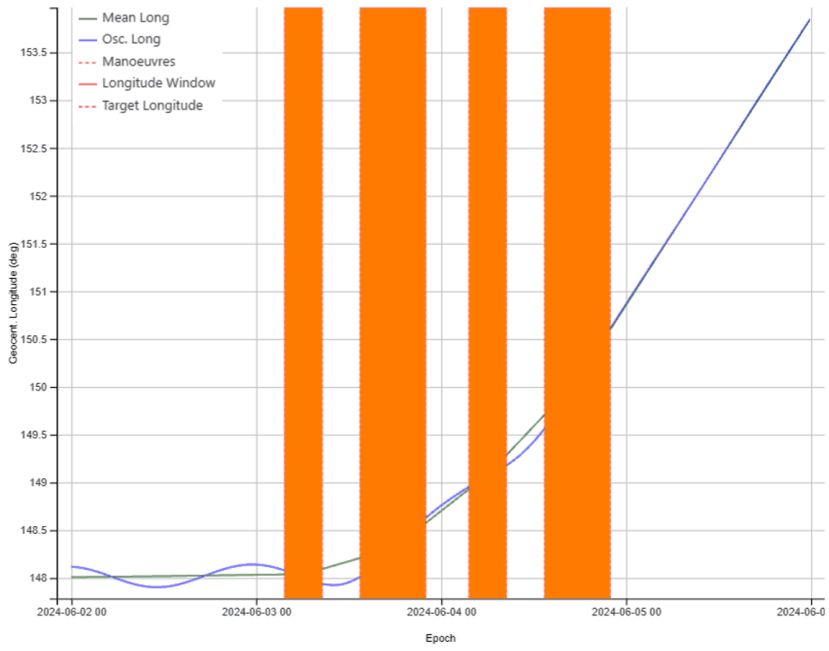

Figure 6.8: Operation 1, longitude evolution

6.1.2 Operation 2

The extraction discussed in this subsection differs from the previous one in terms of the sign of the target drift, i.e., the direction of manoeuvres performed —in the East direction—, and the presence of the constraints related to eclipses. As a matter of fact, Operation 1 was executed in June, during which no eclipses were accounted for. In contrast, for Operation 2, the manoeuvring window was specifically selected in September to coincide with an eclipse period. In more detail, during the analysis interval, the ***FocusSuite*** flight dynamics software computed five Sun eclipses by Earth.

Furthermore, it has been decided to present the results for this particular operation because, in this instance, both the analytical method and the optimiser successfully found a solution that satisfies the constraints. This allows for conducting two analyses: initially calling the optimiser at the last iteration of the analytical method (Analysis 1), and then at the first iteration of the same (Analysis 2).

6.1.2.1 Analysis 1

In this simulation, the analytical method gets a solution in three iterations, this solution is then passed to the NLPQLP code, which performs two internal iterations to optimise it. The manoeuvre solutions obtained from both methods are presented in Tab. 6.11. The evolution of the orbital parameters differs from those of Operation 1 mainly in terms of the sign of the drift, resulting in an increase of the semi-major axis through the manoeuvres.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/16-01:29:10	2024/09/16-08:32:02	East	25372	2.228
2024/09/16-13:43:37	2024/09/16-20:13:39	East	23402	2.055
2024/09/17-01:25:14	2024/09/17-08:28:06	East	25372	2.228
2024/09/17-13:39:41	2024/09/17-20:09:44	East	23402	2.055
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/16-01:29:19	2024/09/16-08:32:15	East	25376	2.228
2024/09/16-13:43:41	2024/09/16-20:13:42	East	23401	2.055
2024/09/17-01:25:28	2024/09/17-08:28:19	East	25372	2.228
2024/09/17-13:39:44	2024/09/17-20:09:41	East	23397	2.055

Table 6.11: Operation 2, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.345 \cdot 10^{-5}$	N/A	$0.429 \cdot 10^{-4}$
Numerical	$-0.999 \cdot 10^{-4}$	N/A	$0.447 \cdot 10^{-4}$

Table 6.12: Operation 2, analysis 1: orbital targets errors at the end of the manoeuvre period

It can be observed by looking at Tab. 6.11 that the two lists of manoeuvres generated by the analytical method and the optimiser are nearly identical. There is a slight difference in the start and end times of the manoeuvres, and thus in the midpoints, but these discrepancies are very minimal, on the order of a few seconds. Furthermore, as seen in the case analyzed in 6.1.1, calling the optimiser at the end of the analytical method results in a reduction of only a few seconds in the overall duration of the manoeuvres through the optimisation. The presence of such a pronounced similarity between the analytical and numerical solutions could be explained by the existence of the constraints related to the eclipses. These, combined with the orbital and operational constraints, further narrow down the range of feasible solutions.
Moreover, given that the initial solution used as initial guess is very good —as was also the case in Operation 1—, this is only slightly modified by the NLPQLP code.

Note that the eccentricity of the starting orbit for Operation 2 is an order of magnitude lower than that of Operation 1, making it apparently easier for the analytical method to find a solution that correctly controls the eccentricity. It is interesting to note that increasing the value of the initial eccentricity to 0.001 results in more iterations—precisely 39—for the analytical method to reach a solution, thus making the resolution process more difficult. This result also attests to the strong dependency of the problem solution on the characteristics of the starting orbit, an aspect that will be further investigated in the simulations of Operation 3.

6.1.2.2 Analysis 2

For this analysis, the manoeuvre list produced at the first iteration of the analytical method is directly used to initiate the optimisation process, which achieves a proper solution after three iterations. It must be noted that with this approach, the algorithm will receive a rough approximation of the solution that may not fulfill several of the applied constraints. The main results of the analysis are summarized in Tab. 6.13 and 6.14.

In contrast to the cases seen so far, as expected, there is a significant reduction in the overall duration of manoeuvres of approximately 50 minutes. This substantial decrease is clearly explained by the quality of the solution used as the initial guess, which is much worse compared to that of Analysis 1. In this situation, the optimiser must modify and significantly improve the manoeuvre plan. In all cases tested, calling the optimiser at the first iteration of the analytical method resulted significant reductions of the manoeuvres duration, which is logically understandable. It must be highlighted that the two different optimisation strategies followed in Analysis 1 and 2 lead to two distinct final solutions for the same extraction operation. Specifically, in the case of Analysis 2, the sum of the durations of the solution manoeuvres is imperceptibly greater —by about 0.2 seconds—.

This is linked to the fact that, as mentioned in Chapter 4, the SQP algorithm, due to its intrinsic characteristics, only ensures convergence towards local solutions and not global ones of a nonlinear problem. Ultimately, this demonstrates how the method employed depends on the initial guess provided: thus, by varying the starting point for the optimisation, different results are generally achieved, even when considering the same case study.

For brevity, the evolution of the orbital parameters resulting from the optimal solution manoeuvres is reported only for the numerical result of Analysis 1. This decision is made due to the lack of significant differences in trends between the two analyses, as they pertain to the same extraction operation.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/16-01:15:05	2024/09/16-08:24:34	East	25769	2.263
2024/09/16-13:23:44	2024/09/16-20:11:59	East	24495	2.151
2024/09/17-01:11:09	2024/09/17-08:20:38	East	25769	2.263
2024/09/17-13:19:48	2024/09/17-20:08:03	East	24495	2.151
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/16-01:21:36	2024/09/16-08:18:44	East	25027	2.198
2024/09/16-13:30:01	2024/09/16-20:05:55	East	23754	2.086
2024/09/17-01:17:48	2024/09/17-08:14:46	East	25018	2.197
2024/09/17-13:26:03	2024/09/17-20:01:50	East	23746	2.085

Table 6.13: Operation 2, analysis 2: manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.916 \cdot 10^{-1}$	N/A	$0.722 \cdot 10^{-4}$
Numerical	$-0.999 \cdot 10^{-4}$	N/A	$0.703 \cdot 10^{-4}$

Table 6.14: Operation 2, analysis 2: orbital targets errors at the end of the manoeuvre period

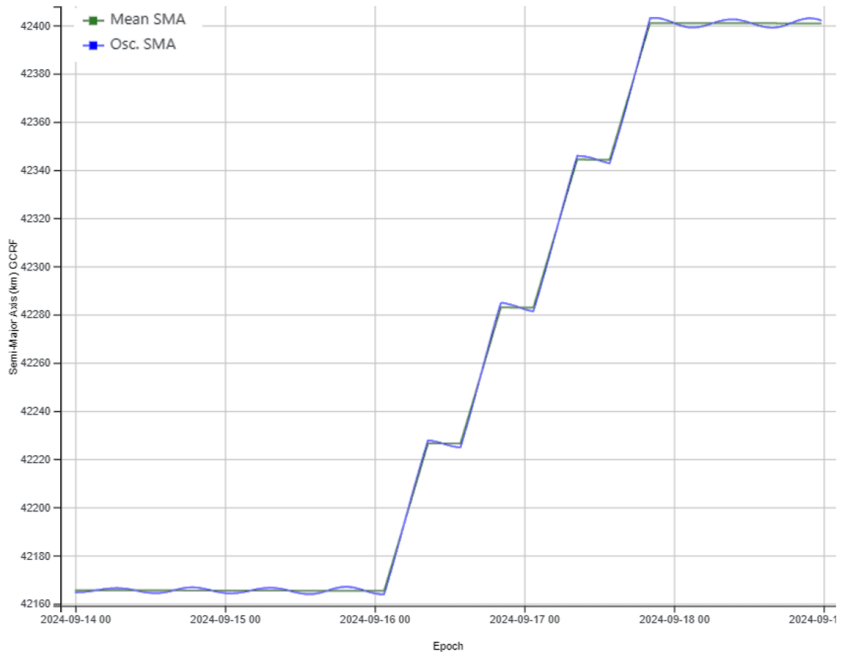

Figure 6.9: Operation 2, Analysis 1: semi-major axis evolution

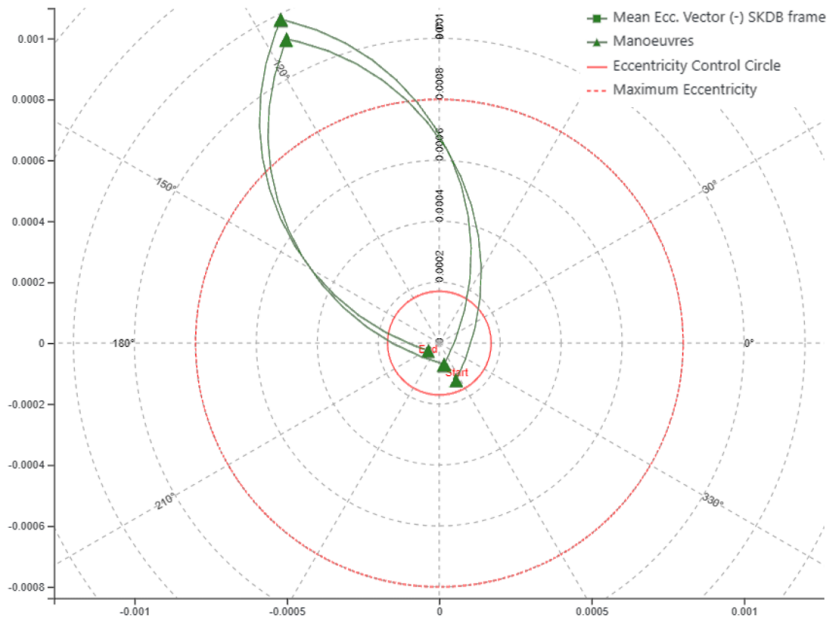

Figure 6.10: Operation 2, Analysis 1: mean eccentricity vector evolution

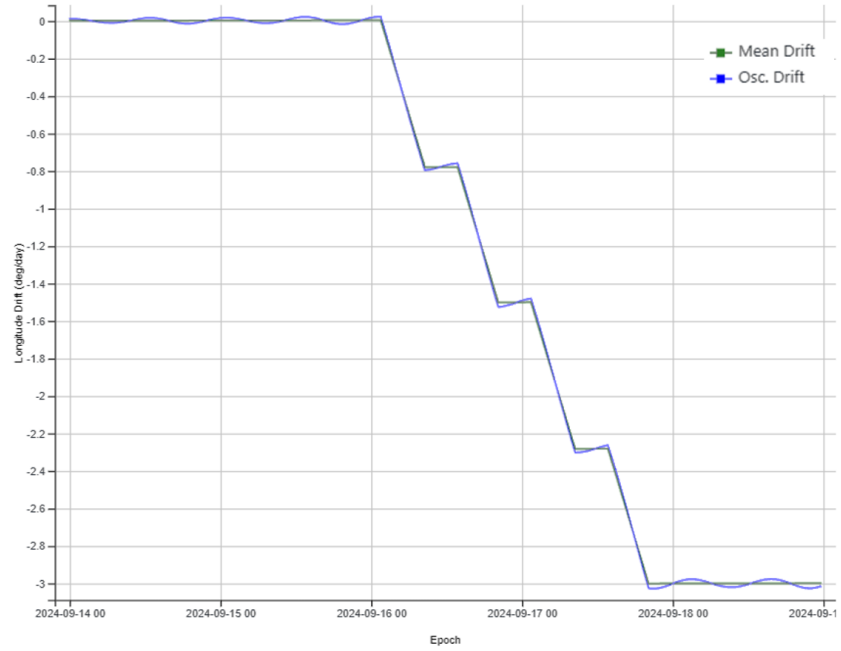

Figure 6.11: Operation 2, Analysis 1: drift evolution

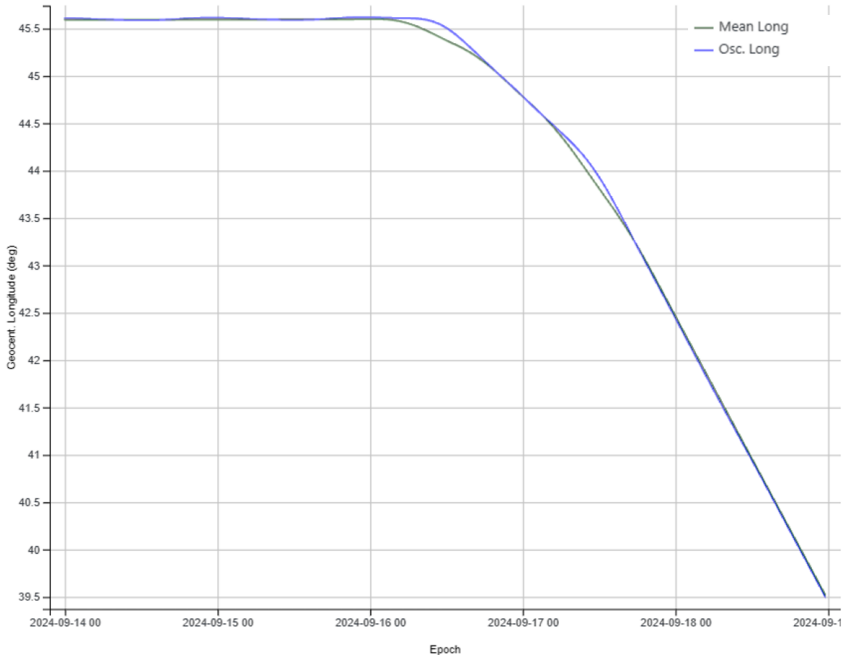

Figure 6.12: Operation 2, Analysis 1: longitude evolution

6.1.3 Operation 3

Similar to Operation 1, this investigated case study witnesses convergence of the optimiser but not of the analytical method. Likewise, the solution reached by the analytical method after the maximum number of iterations allowed is not acceptable in terms of eccentricity. However, as discussed in subsection 6.1.1, through the optimisation process executed at the end of the traditional method, it is possible to obtain a list of manoeuvres that solve the constrained problem and thus also meet the eccentricity target.

In any case, it is useful to present this simulation, as the selected value for the drift target is notably high. This allows for the verification of the optimiser's correct functioning, even in situations where a high number of manoeuvres is required to execute the extraction.
In more detail, this analysis demonstrates the capability of the NLPQLP algorithm to manage problems characterized by numerous solution variables. Specifically, in the case under consideration, it adeptly handles 18 solution variables, achieving a solution in just two iterations.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/03-02:14:46	2024/06/03-05:43:55	West	12549	-1.102
2024/06/03-09:08:02	2024/06/03-14:48:02	West	20399	-1.791
2024/06/03-17:06:44	2024/06/03-22:46:43	West	20399	-1.791
2024/06/04-02:10:51	2024/06/04-05:39:59	West	12549	-1.102
2024/06/04-09:04:06	2024/06/04-14:44:06	West	20399	-1.791
2024/06/04-17:02:48	2024/06/04-22:42:47	West	20399	-1.791
2024/06/05-02:06:55	2024/06/05-05:36:03	West	12549	-1.102
2024/06/05-09:00:11	2024/06/05-14:40:10	West	20399	-1.791
2024/06/05-16:58:52	2024/06/05-22:38:51	West	20399	-1.791
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/03-01:08:04	2024/06/03-05:51:36	West	17012	-1.494
2024/06/03-08:27:27	2024/06/03-15:02:37	West	23710	-2.082
2024/06/03-16:56:03	2024/06/03-23:09:39	West	22415	-1.968
2024/06/04-01:36:15	2024/06/04-05:10:37	West	12862	-1.130
2024/06/04-09:10:18	2024/06/04-14:45:28	West	20110	-1.766
2024/06/04-17:15:52	2024/06/04-22:34:39	West	19127	-1.680
2024/06/05-01:57:15	2024/06/05-04:44:36	West	10041	-0.882
2024/06/05-09:38:55	2024/06/05-14:33:53	West	17698	-1.554
2024/06/05-17:26:03	2024/06/05-22:10:22	West	17060	-1.498

Table 6.15: Operation 3, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.120 \cdot 10^{-6}$	N/A	$0.261 \cdot 10^{-3}$
Numerical	$0.952 \cdot 10^{-4}$	N/A	$0.454 \cdot 10^{-4}$

Table 6.16: Operation 3, orbital targets error summary

In the graphs and tables provided, it is possible to appreciate the increase in the number of burns compared to the two operations previously illustrated. It is obvious that the higher the absolute value of the target drift, the more expensive the extraction operation becomes in terms of propulsion. Consequently, due to the restriction on the maximum duration of manoeuvres, it is generally true that increasing the drift target typically results in a rise in the number of solution manoeuvres. It is important to clarify that the choice of a 5 deg/day drift was made for illustrative purposes. Such a high value is rarely used in relocation operations, being closer in module to the values used for de-orbiting operations. This case study serves to demonstrate additional insights into the impact of the starting orbit's eccentricity on the solution process. For clarity, it is designated as Operation 3b, an extraction that replicates Operation 3 but begins from a geostationary orbit with $e = 0.0001$. Despite the reduction in initial eccentricity, contrary to what was observed in 6.1.2.1, the analytical method still fails to control e. This underscores the complexity of the analyzed physical problem, making it challenging to draw general conclusions. As expected, altering the initial orbit leads to a different solution for the analytical method, affecting the optimiser's convergence toward a distinct solution compared to Operation 3.

Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/03-00:06:17	2024/06/03-06:54:42	West	24505	-2.152
2024/06/03-08:14:15	2024/06/03-15:06:18	West	24723	-2.171
2024/06/03-16:50:59	2024/06/03-22:29:38	West	20319	-1.784
2024/06/04-00:47:42	2024/06/04-05:46:25	West	17923	-1.574
2024/06/04-09:24:30	2024/06/04-14:34:58	West	18628	-1.636
2024/06/04-17:26:09	2024/06/04-21:30:38	West	14670	-1.289
2024/06/05-01:28:05	2024/06/05-05:11:19	West	13394	-1.176
2024/06/05-10:10:39	2024/06/05-14:14:40	West	14642	-1.286
2024/06/05-17:46:12	2024/06/05-20:53:29	West	11237	-0.987

Table 6.17: Operation 3b, manoeuvres data

D (deg/day)	λ (deg)	e
$0.992 \cdot 10^{-4}$	N/A	$0.393 \cdot 10^{-4}$

Table 6.18: Operation 3b, orbital targets error summary

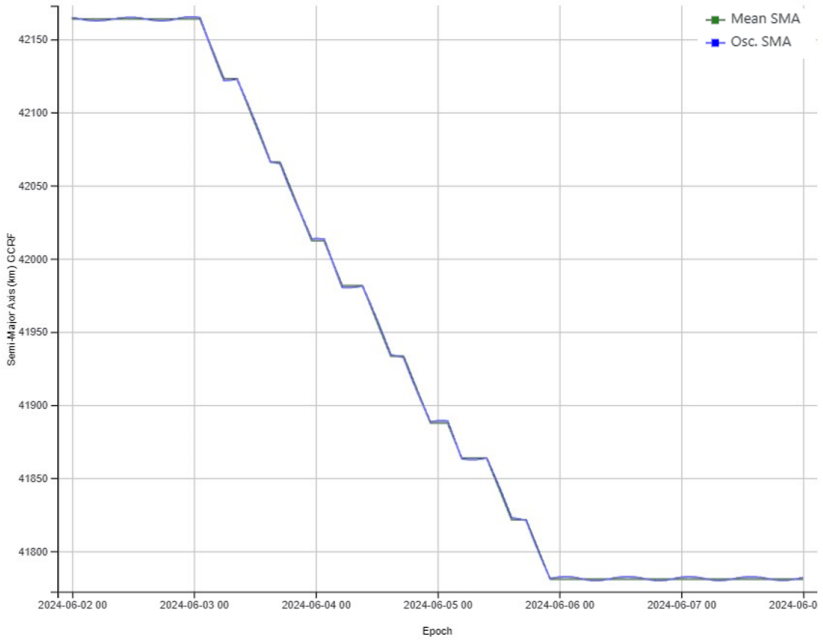

Figure 6.13: Operation 3, semi-major axis evolution

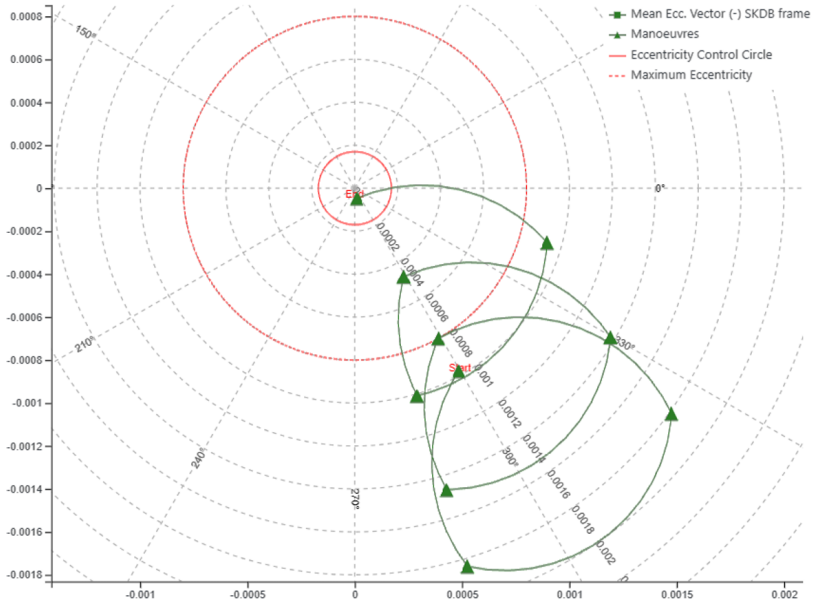

Figure 6.14: Operation 3, mean eccentricity vector evolution

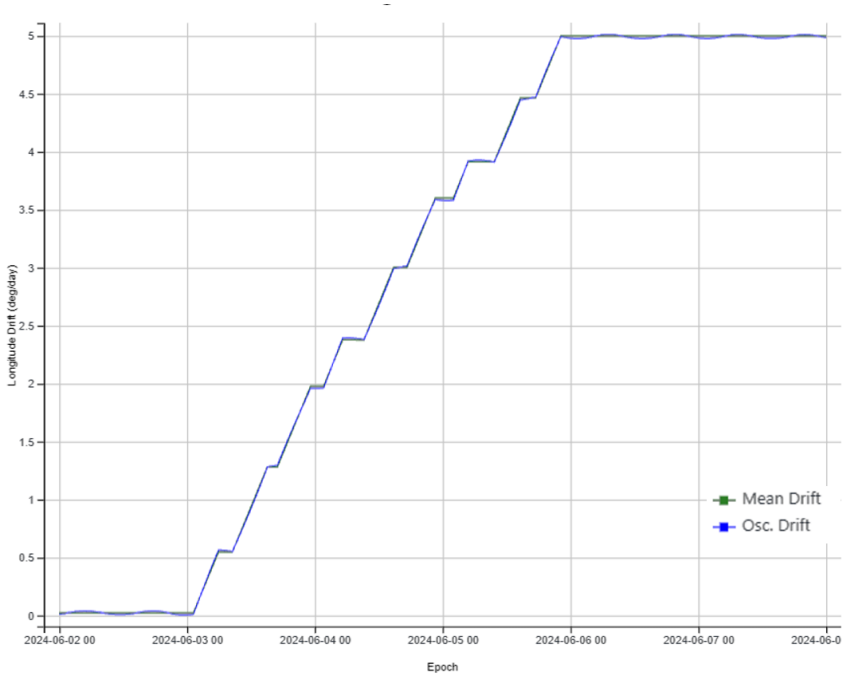

Figure 6.15: Operation 3, drift evolution

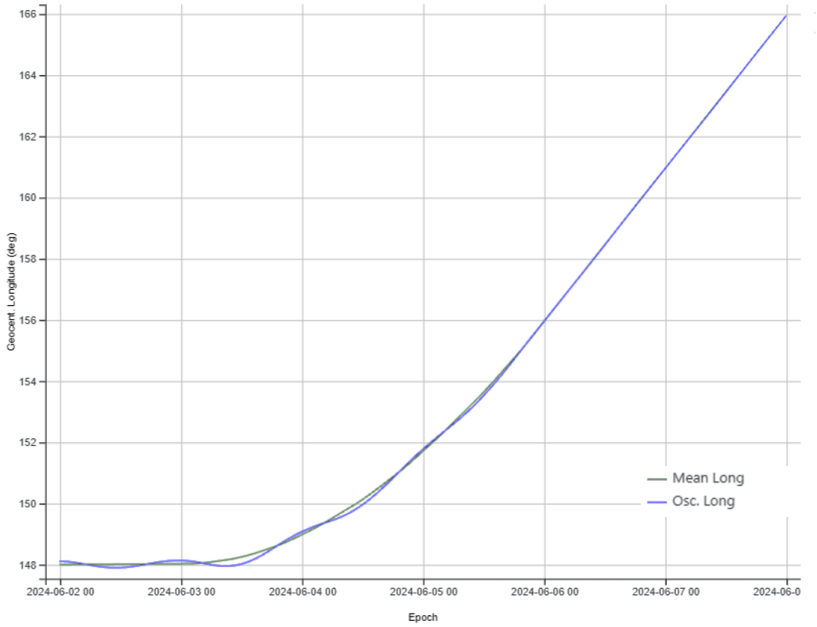

Figure 6.16: Operation 3, longitude evolution

6.2 Insertion with Continuous Manoeuvres

The insertion phase represents somewhat of a dual counterpart to extraction, as during this operation, the satellite, after being removed from the reference geostationary orbit, is reintroduced into it. From a physical standpoint, insertion can be seen like performing extraction in reverse. Therefore, many of the general considerations made in Paragraph 6.1 regarding manoeuvres and the evolution of orbital parameters still apply. Clearly, in this case, the drift target will be close to zero, as the destination orbit is the GEO one.

The main distinguishing feature for the insertion lies in the addition of the constraint on controlling longitude. As already mentioned, this orbital parameter is not directly modified by manoeuvres, so it is necessary to time the insertion manoeuvres accurately to reach the GEO orbit with the required longitude after the drift phase.

The optimisation of the individual insertion does not provide insights beyond those discussed in the previous paragraph. Additionally, in order to simulate this phase, as previously mentioned, it is necessary to first execute the simulation of an extraction, making it cumbersome to conduct numerous analyses. For these reasons, optimisation of the individual insertion has been tested only a few times on the ***FocusSuite*** platform, solely to verify the correct formulation of the longitude constraints, before investigating the full relocation. However, for consistency, the result of an insertion operation is still presented in this Paragraph, allowing to demonstrate the correct functioning of the optimiser even when employed for this single phase.

In more detail, an extraction to a drift orbit characterized by $D = -3$ deg/day, executed between June 16, 2024, and June 17, 2024, is computed using the NLPQLP code after the analytical method. The orbit reached after the extraction manoeuvres represents the starting point for the investigated insertion operation. The key details about this initial orbit are outlined in Tab. 6.19. The orbital targets are summarized in Tab. 6.20, from which it can be inferred that a longitude variation of about 12 degrees is required. The definition of the manoeuvre interval must take into account also the time demanded for the drift period, in which the satellite effectively changes its longitude.

	Operation 4
Initial longitude - λ_0 (deg)	132.28
Initial semi-major axis - a_0 (km)	42401.6
Initial drift - D_0 (deg/day)	-3.0
Initial eccentricity - e_0	0.0
Simulation start time	2024/06/18-00:00
Simulation end time	2024/06/28-00:00

Table 6.19: Analysed insertion operation, initial state

	Operation 4
Longitude target - λ_t (deg)	120.0
Drift target - D_t (deg/day)	0.0001
Eccentricity target - e_t	0.0
Manoeuvre window start time	2024/06/19-00:00
Manoeuvre window end time	2024/06/27-00:00
Operation Type	Insertion

Table 6.20: Analysed insertion operation, set targets

The optimisation process is carried out at the end of the analytical method, which finds a correct solution in 9 iterations. The main results about the manoeuvre lists obtained through the two methods are depicted in Tab. 6.21. The trends of the orbital parameters in the reported graphs and the information contained in Tab. 6.22 demonstrate that the solution achieved from the optimisation is globally correct and complies with all the imposed orbital constraints.
The results in terms of reduction in manoeuvre duration and variation of manoeuvre midpoints align with those observed for the extraction case. Specifically, as already seen in the analyses presented in Paragraph 6.1, optimisation only marginally improves the solution produced by the traditional method.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/21-05:20:04	2024/06/21-12:18:17	West	25092	-2.204
2024/06/21-17:28:55	2024/06/22-00:05:30	West	23795	-2.090
2024/06/22-05:16:09	2024/06/22-12:14:21	West	25092	-2.204
2024/06/22-17:24:59	2024/06/23-00:01:34	West	23795	-2.090
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/21-05:20:03	2024/06/21-12:18:18	West	25095	-2.204
2024/06/21-17:28:46	2024/06/22-00:05:22	West	23797	-2.090
2024/06/22-05:16:08	2024/06/22-12:14:17	West	25089	-2.203
2024/06/22-17:24:51	2024/06/23-00:01:21	West	23790	-2.089

Table 6.21: Operation 4, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.577 \cdot 10^{-7}$	$0.186 \cdot 10^{-4}$	$0.676 \cdot 10^{-4}$
Numerical	$0.999 \cdot 10^{-4}$	$0.994 \cdot 10^{-4}$	$0.662 \cdot 10^{-4}$

Table 6.22: Operation 4, orbital targets errors at the end of the manoeuvre period

This scenario has been included for completeness however, instead of separately optimising the extraction and insertion phases, it is more logical to employ the analytical method to immediately compute the solution for the entire relocation operation and then directly optimise the generated manoeuvres.

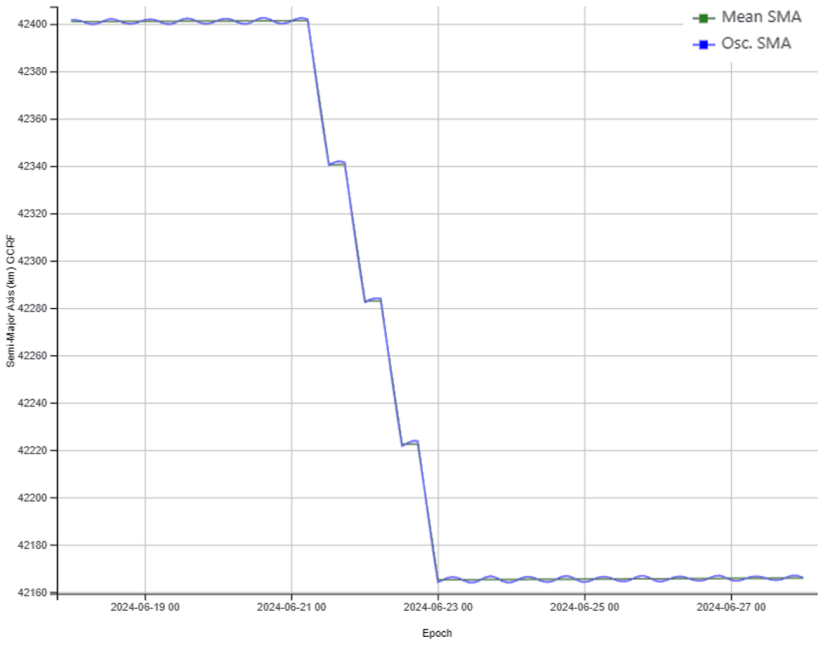

Figure 6.17: Operation 4, semi-major axis evolution

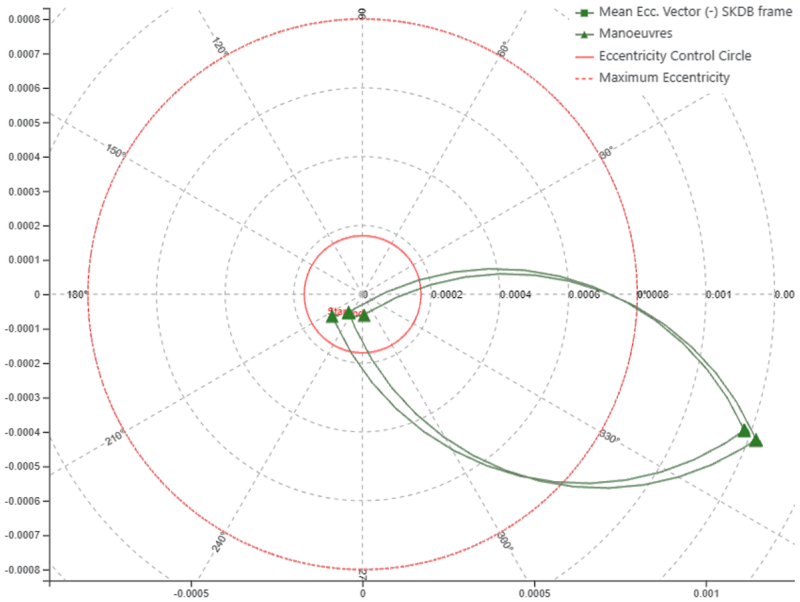

Figure 6.18: Operation 4, mean eccentricity vector evolution

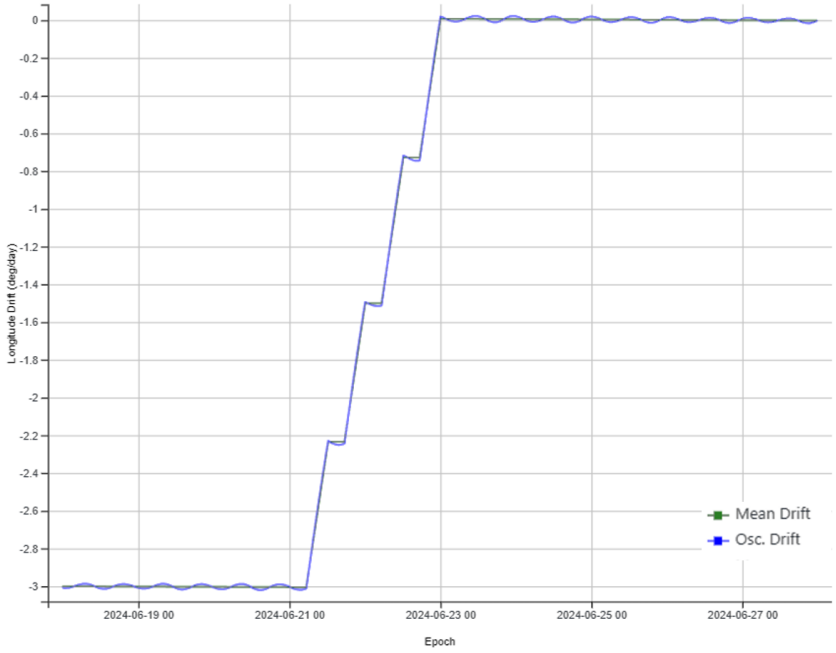

Figure 6.19: Operation 4, drift evolution

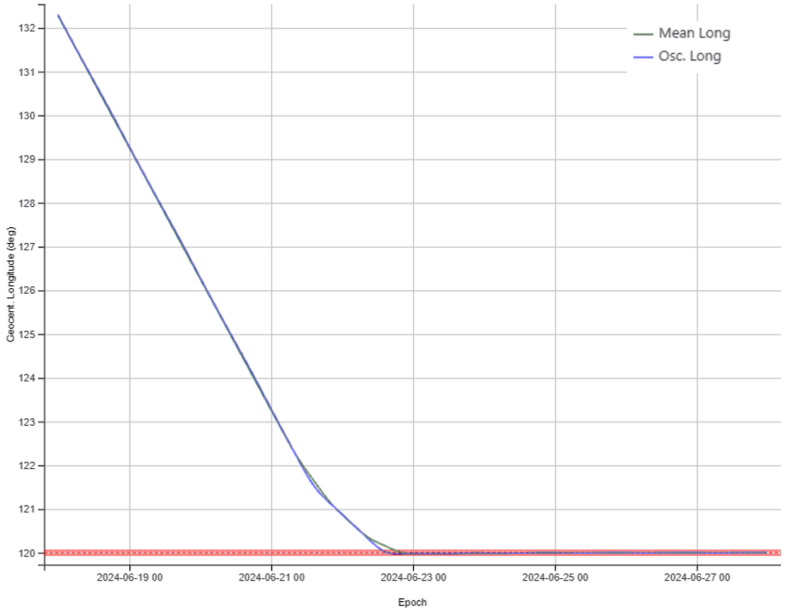

Figure 6.20: Operation 4, longitude evolution

6.3 Full Relocation with Continuous Manoeuvres

Given that the primary objective of this work was to optimise East-West manoeuvres for relocation operations, it is significant to present the results achieved from implementing the NLPQLP algorithm for the optimisation of the entire relocation process.
Many of the concepts and considerations highlighted in the preceding paragraphs for the single insertion and extraction phases can naturally extend to the full relocation scenario, as it involves both extraction and insertion performed sequentially. Specifically, the aspects observed regarding manoeuvres executed in the two individual phases remain still valid when considering the entire relocation. Furthermore, the results emphasized in the previous case studies concerning the optimisation process and its implementation and characteristics are applicable to the analyses presented here. Implementing the NLPQLP subroutine has necessitated minor adjustments in the code compared to the individual phases, as the analytical method's approach to computing a solution for them differs slightly from the algorithm used to calculate the full relocation.
It is important to note that the examples provided for the optimisation of extraction manoeuvres are representative instances of the general optimiser implementation, offering a comprehensive understanding of almost all its key features. Hence, the considerations outlined in Paragraph 6.1 will not be reiterated here, and only the peculiarities of the specific case studies will be emphasized.

Two distinct operations have been simulated, and the main information about them is summarized in Tab. 6.23. Each operation requires a longitudinal shift of approximately 30 degrees, which is accomplished in both of them by utilizing a drift orbit characterized by $D = -3$ deg/day.
The manoeuvre windows vary between the two cases: the first one takes place during an eclipse period, specifically in September, while the other operation occurs in December when no eclipses take place. In this manner two distinct situations are analysed, with Operation 5 characterized by more restrictions due to the presence of constraints related to the separation from eclipses.

	Operation 5	Operation 6
Initial longitude - λ_0 (deg)	48.55	328.72
Initial semi-major axis - a_0 (km)	42164.7	42164.7
Initial drift - D_0 (deg/day)	0.016	0.023
Initial eccentricity - e_0	0.0001	0.0001
Simulation start time	2024/09/11-00:00:00	2024/12/01-00:00:00
Simulation end time	2024/09/26-00:00:00	2024/12/16-00:00:00
Manoeuvre window start time	2024/09/12-00:00	2024/12/02-00:00:00
Manoeuvre window end time	2024/09/25-00:00	2024/12/15-00:00:00
Operation Type	Full Relocation	Full Relocation

Table 6.23: Analysed relocation operations, initial state and general overview

Note that the start and end times for the manoeuvre window outlined in Tab. 6.23 apply to the entire relocation process, encompassing both extraction and insertion phases. Finally, the orbital targets for both extraction and insertion must be defined, these are shown in Tab. 6.23 and 6.24.

	Operation 5, 6
Longitude target - λ_t (deg)	-
Drift target - D_t (deg/day)	-3
Eccentricity target - e_t	0.0

Table 6.24: Analysed relocation operations, extraction orbit targets

	Operation 5	Operation 6
Longitude target - λ_t (deg)	18.0	300.0
Drift target - D_t (deg/day)	0.0001	0.0001
Eccentricity target - e_t	0.0	0.0

Table 6.25: Analysed relocation operations, insertion orbit targets

6.3.1 Operation 5

The case study is notable because the analytical method fails to find a solution, particularly struggling with respecting the constraint related to eccentricity during the insertion while avoiding the overlap between manoeuvres and eclipses. In particular reached the maximum number of iterations, one of the manoeuvre violates the eclipse constraints, overlapping with an umbra period. Additionally, the insertion eccentricity target is not achieved with the required accuracy. Utilizing the optimiser after the traditional method it is possible to correct this initial and not acceptable solution, generating a list of manoeuvres that satisfy all constraints and thus solve the nonlinear constrained problem. The particulars of results just commented are presented in Tab. 6.26 and 6.27-6.28.

As evidenced in previous case studies, applying optimisation at the conclusion of the analytical method only yields marginal improvements in manoeuvre solutions, reducing their duration by a few seconds (approximately 10 s in this specific case). However, the most notable outcome of this simulation lies in the optimiser's ability to address highly complex problems. In fact, this operation was specifically chosen as it represents one of the most challenging scenarios that can be simulated on the corporate platform (for the formulated problem in Fortran).

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/12-01:20:15	2024/09/12-08:08:23	East	24488	2.150
2024/09/12-13:18:17	2024/09/12-20:06:25	East	24488	2.150
2024/09/13-01:16:19	2024/09/13-08:04:27	East	24488	2.150
2024/09/13-13:14:21	2024/09/13-20:02:29	East	24488	2.150
2024/09/22-03:15:56	2024/09/22-09:21:27	West	21930	-1.926
2024/09/22-14:36:34	2024/09/22-21:56:53	West	26419	-2.320
2024/09/23-03:12:01	2024/09/23-09:17:31	West	21930	-1.926
2024/09/23-14:32:38	2024/09/23-21:52:57	West	26419	-2.320
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/09/12-01:10:24	2024/09/12-08:07:18	East	25014	2.197
2024/09/12-13:27:15	2024/09/12-20:06:29	East	23954	2.104
2024/09/13-01:06:53	2024/09/13-08:03:48	East	25014	2.197
2024/09/13-13:23:21	2024/09/13-20:02:46	East	23966	2.105
2024/09/22-04:12:34	2024/09/22-08:39:41	West	16027	-1.407
2024/09/22-16:16:19	2024/09/22-21:55:12	West	20333	-1.786
2024/09/23-02:20:09	2024/09/23-10:28:19	West	29290	-2.572
2024/09/23-13:21:54	2024/09/23-21:59:17	West	31043	-2.726

Table 6.26: Operation 5, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.105 \cdot 10^{-6}$	N/A	N/A
Numerical	$-0.956 \cdot 10^{-4}$	N/A	N/A

Table 6.27: Operation 5, orbital targets errors at the end of the extraction manoeuvres

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.122 \cdot 10^{-4}$	-0.371	$0.125 \cdot 10^{-3}$
Numerical	$0.100 \cdot 10^{-3}$	$-0.986 \cdot 10^{-4}$	$0.944 \cdot 10^{-4}$

Table 6.28: Operation 5, orbital targets errors at the end of the insertion manoeuvres

Given the presence of numerous operational and orbital constraints affecting both insertion and extraction phases, alongside additional limitations imposed by eclipses, finding a solution becomes exceptionally challenging. Moreover, the low performance of electric thrusters necessitates prolonged continuous manoeuvres, further complicating the problem. Nevertheless, the successful resolution and optimisation without encountering errors by the code demonstrate its effectiveness in addressing even the most arduous problems characterized by a significantly restricted field of feasible solutions.

Similar situations, where the optimiser successfully obtains a solution while the analytical method fails to converge, have been encountered in various tests involving full relocation operations during eclipse periods. This observation underscores the intricacy of the problem and emphasizes the analytical method's difficulty in addressing such scenarios. Finally, this case study further validates the NLPQLP algorithm's capability to effectively manage scenarios with multiple constraints.

The subsequent graphs illustrate the evolution of orbital parameters throughout the optimal relocation process. The graphs exhibit a symmetrical trend in the drift and semi-major axis, where is possible the identification of the distinct phases that composed the process. Specifically, the initial segment depicts the extraction phase, during which the desired drift target is attained by increasing the orbit altitude. Subsequently, the drift phase is exploited, during which satellites stationed in the new orbit undergo a reduction in longitude, as evidenced in the central portion of Fig. 6.24. This figure showcases the evolution of longitude, ultimately reaching the target value specified by requirements and then maintaining a constant value, a characteristic inherent to the geostationary orbit, as discussed in Chapter 1.

The concluding segments of Fig. 6.21 and 6.23 depict the insertion phase, wherein the semi-major axis, along with the drift, is adjusted to values typical of the geostationary orbit. Figure 6.22 illustrates the evolution of the eccentricity vector, demonstrating how this parameter is influenced by the various manoeuvres executed. The reported trend attests to the accuracy of the solution achieved by the optimiser, adhering to the orbital targets within the predetermined manoeuvre window.

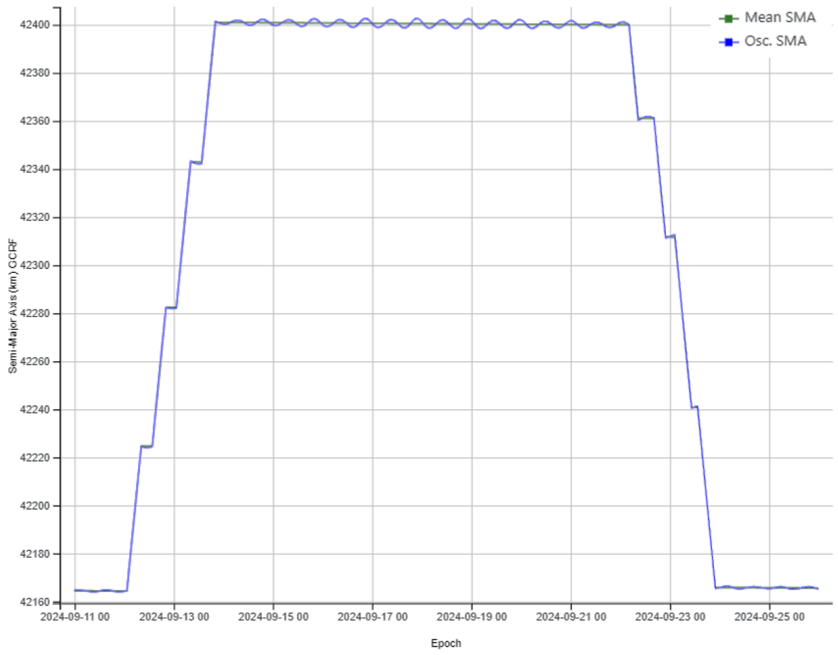

Figure 6.21: Operation 5, semi-major axis evolution

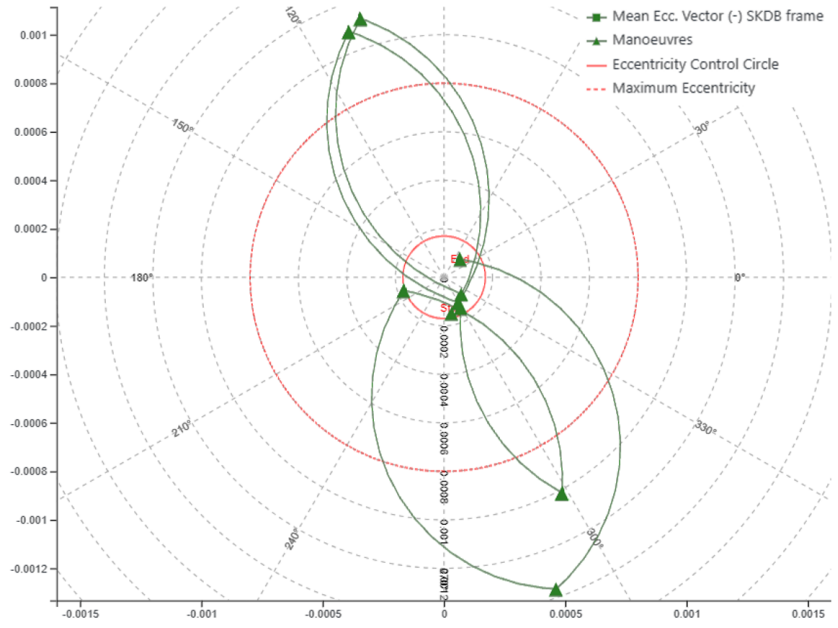

Figure 6.22: Operation 5, mean eccentricity vector evolution

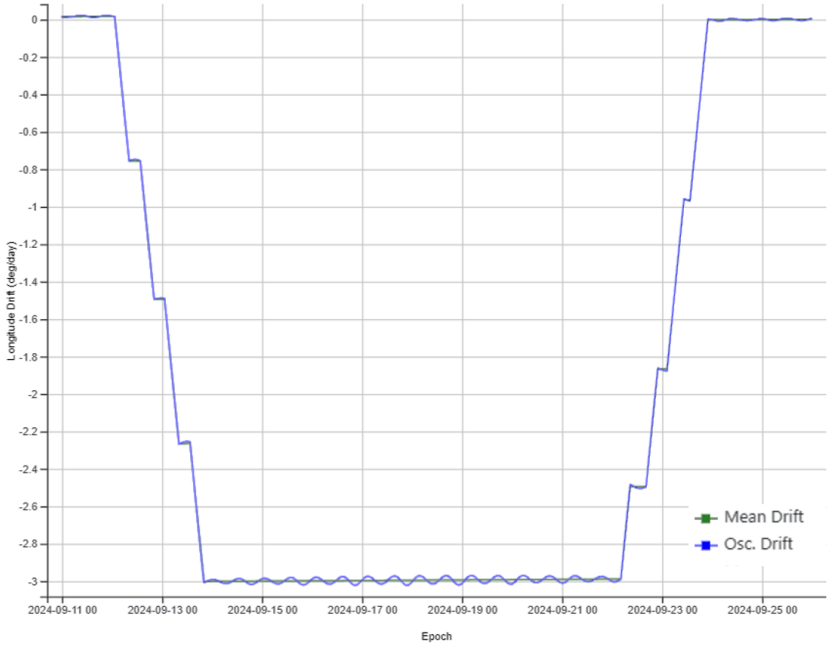

Figure 6.23: Operation 5, drift evolution

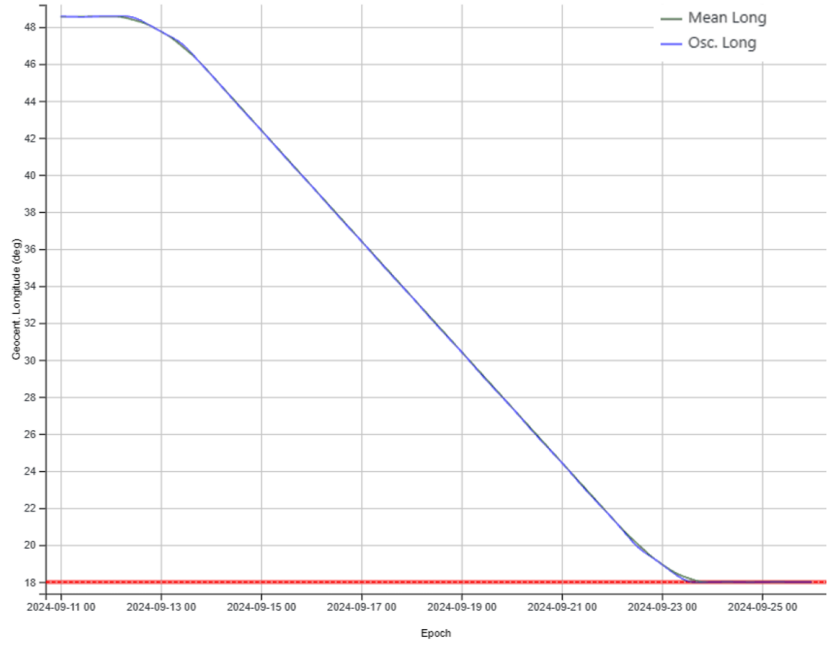

Figure 6.24: Operation 5, longitude evolution

6.3.2 Operation 6

This analysis is provided for completeness to demonstrate the functioning of the optimiser when invoked during the initial iteration of the analytical algorithm for relocation scenarios. In this particular case, as observed in other simulations presented, the eccentricity constraint is not met — in the specific for the insertion manoeuvres of the relocation— after reaching the preset maximum number of iterations of the traditional method. Therefore, the optimiser is utilized to compute a solution, and, as said, it is directly called during the first iteration. The solution achieved by the optimiser, along with the initial guess provided by the analytical method, are presented in Tab. 6.29.

Examining Tab. 6.30, which displays the errors in achieving the orbital targets by the aforementioned solutions, it is evident that the manoeuvre generated by the analytical method represents a very poor approximation of the correct solution. Conversely, the NLPQLP code employs 3 and 2 iterations, respectively, to optimise the extraction and insertion manoeuvres respecting all the constraints. As expected, the reduction in manoeuvre duration is consistent.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/12/02-02:36:01	2024/12/02-09:22:01	East	24359	2.139
2024/12/02-14:34:03	2024/12/02-21:20:03	East	24359	2.139
2024/12/03-02:32:05	2024/12/03-09:18:05	East	24359	2.139
2024/12/03-14:30:07	2024/12/03-21:16:07	East	24359	2.139
2024/12/11-23:30:22	2024/12/12-06:15:36	West	24314	-2.135
2024/12/12-11:15:60	2024/12/12-18:26:02	West	25803	-2.266
2024/12/12-23:26:26	2024/12/13-06:11:40	West	24314	-2.135
2024/12/13-11:12:04	2024/12/13-18:22:06	West	25803	-2.266
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/12/02-02:25:00	2024/12/02-09:20:53	East	24953	2.191
2024/12/02-14:43:00	2024/12/02-21:22:36	East	23976	2.106
2024/12/03-02:21:26	2024/12/03-09:17:22	East	24956	2.192
2024/12/03-14:38:31	2024/12/03-21:18:01	East	23970	2.105
2024/12/11-20:04:08	2024/12/12-05:57:14	West	35586	3.125
2024/12/12-08:28:25	2024/12/12-18:09:32	West	34867	3.062
2024/12/12-22:54:28	2024/12/13-02:47:59	West	14012	1.230
2024/12/13-11:28:16	2024/12/13-15:08:34	West	13218	1.161

Table 6.29: Operation 6, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.918 \cdot 10^{-1}$	N/A	N/A
Numerical	$-0.949 \cdot 10^{-4}$	N/A	N/A

Table 6.30: Operation 6, orbital targets errors at the end of the extraction manoeuvres

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.918 \cdot 10^{-1}$	0.576	$0.218 \cdot 10^{-3}$
Numerical	$0.100 \cdot 10^{-3}$	$0.997 \cdot 10^{-4}$	$0.448 \cdot 10^{-4}$

Table 6.31: Operation 6, orbital targets errors at the end of the insertion manoeuvres

For the sake of brevity, the evolutions of the orbital parameters determined by the manoeuvre solutions generated by the optimiser are not reported. The trends are globally similar to the ones presented for Operation 5, considering also that the extraction orbit is practically the same. Therefore, they do not offer insights for further significant considerations and for the analysis results, the tables mentioned above are sufficient.

6.4 Impulsive Manoeuvres

The primary focus of this thesis was on satellites equipped with electric thrusters, thus on the optimisation of continuous manoeuvres. However, upon completing the application of the NLPQLP algorithm for continuous manoeuvres, it was decided to extend its implementation to include impulsive manoeuvres. This extension aimed to validate the work conducted and explore the possibility of using the optimiser for a broader range of problems. In particular, this strategy enabled the utilization of the SQP method for optimising the relocation operations of satellites equipped with chemical propulsors. The necessary modifications have been explained in Chapter 5, primarily involving problem formulation, particularly the definition of the solution vector. The simulations conducted yielded satisfactory results, confirming the functionality of the optimisation process for impulsive manoeuvres. The overall functioning of the optimiser and the general aspects of the results generated by it are similar to those presented for the continuous case. However, it is important to reiterate that the analysis of impulsive manoeuvre cases was not the primary objective of this work. Therefore, their study and testing were not conducted in detail, resulting in only a few simulations being carried out. The most representative results obtained are reported here solely for completeness. Specifically, three operations were considered: single extraction, single insertion, and full relocation. This choice was made for consistency to demonstrate the functionality of the optimiser for the three possible processes related to satellite relocation that can be tested on the company's platform. The summarized results are briefly provided in the following paragraphs.

It is noteworthy that the presented solutions for the insertion phase, whether carried out individually or as part of the relocation process, do not consist solely of manoeuvres in a single direction (east or west), as one might expect. Instead, manoeuvres in both east and west directions can be observed, generated by the specific algorithm constituting the analytical solution. Upon receiving the list of generated manoeuvres, the optimiser maintains not only the number of manoeuvres but also the direction of their acceleration. Therefore, the list of optimised manoeuvres will align with the direction of the analytical solution. This is a clear limitation of the implemented optimiser that restricts its ability to explore a wider range of solutions. The obtained results indicate that, as expected given the nature of impulsive manoeuvres characterized by brief durations, compliance with constraints is much easier. It is evident that the very limited duration of impulsive manoeuvres facilitates the fulfillment of several of the constraints defined in this work, with separation from eclipses being just one illustrative example. Consequently, the traditional method readily finds a solution in many cases, and this solution is often optimal, obviating the need for iterations of the NLPQLP algorithm.

In the forthcoming graphs illustrating the evolution of orbital parameters for the distinct operations, it can be observed that there are differences compared to those obtained through continuous manoeuvres. Although the global trends are similar, the discrete nature of the variations induced by impulsive manoeuvres is notable. This characteristic is particularly evident in the graphs concerning mean eccentricity and in the figures related to semi-major axis and drift. Here, changes occur instantaneously, resulting in a distinctive and pronounced "stepped" pattern. This stands in contrast to the continuous case, where long-duration manoeuvres led to a gradual variation of parameters, resulting in inclined segments in the graphs of mean drift and semi-major axis, rather than distinctly vertical ones.

6.4.1 Extraction: Operation 7

In this study case, both the analytical method and the optimiser found a solution. The analysis was initially conducted by optimising the manoeuvre list obtained at the end of the analytical method. However, since the same solution was returned by the NLPQLP code after just one iteration, it was decided to repeat the simulation by performing the optimisation at the first iteration. The main information about the operation is summarized in Tab. 6.32 and 6.33, while the results are illustrated in Tab. 6.34 and 6.35. Specifically, the solution from the analytical method, reached after 4 iterations, and the solution generated by the optimiser when called at the first iteration are reported. Therefore, for brevity, the initial guess provided to the optimiser is not included here.

	Operation 7
Initial longitude - λ_0 (deg)	24.9
Initial semi-major axis - a_0 (km)	42164.7
Initial drift - D_0 (deg/day)	0.026
Initial eccentricity - e_0	0.0001
Simulation start time	2024/10/05-00:00
Simulation end time	2024/10/10-00:00

Table 6.32: Analysed extraction operation, initial state

	Operation 7
Longitude target - λ_t (deg)	-
Drift target - D_t (deg/day)	-2.5
Eccentricity target - e_t	0.0
Manoeuvre window start time	2024/10/06-00:00
Manoeuvre window end time	2024/10/09-00:00
Operation Type	Extraction

Table 6.33: Analysed extraction operation, set targets

It is interesting to note that the solution reached by the optimiser differs from that of the analytical method; however, the sum of the delta-V, which represents the objective of the optimisation problem, remains the same. Therefore, the optimiser and the analytical method converge to two different solutions, but both are similarly optimal in terms of total delta-V. This can explain why the optimiser does not need to further improve the solution generated by the analytical method and simply returns it. Consequently, both solutions represent a minimal point of the cost function and thus an optimal solution to the optimisation problem. This result demonstrates that, due to the nature of impulsive manoeuvres, which easily fulfill the constraints, a seemingly wider range of feasible solutions is available for the physical problem. This allows it to yield two different lists of manoeuvres, both minimizing the delta-V.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/10/06-02:30:18	2024/10/06-02:30:19	East	1	3.725
2024/10/06-14:28:20	2024/10/06-14:28:21	East	1	3.479
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/10/06-05:49:20	2024/10/06-05:49:21	East	1	3.725
2024/10/06-17:59:01	2024/10/06-17:59:02	East	1	3.479

Table 6.34: Operation 7, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.231 \cdot 10^{-6}$	N/A	$0.104 \cdot 10^{-6}$
Numerical	$0.999 \cdot 10^{-4}$	N/A	$0.881 \cdot 10^{-4}$

Table 6.35: Operation 7, orbital targets errors at the end of the manoeuvre period

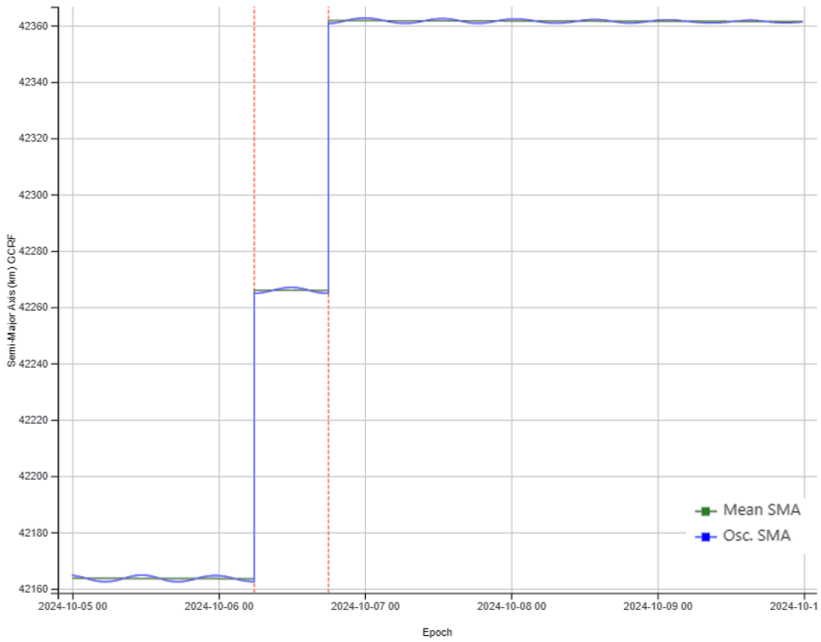

Figure 6.25: Operation 7, semi-major axis evolution

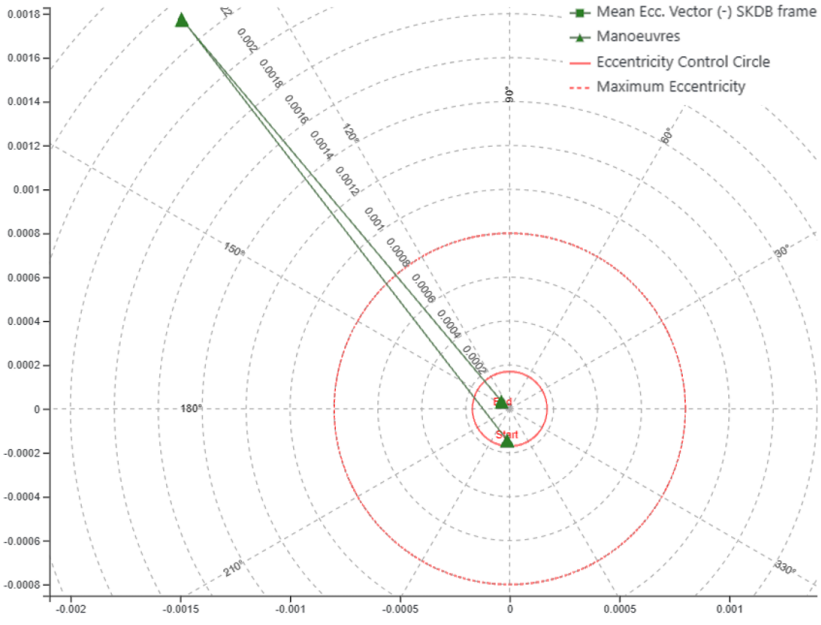

Figure 6.26: Operation 7, mean eccentricity vector evolution

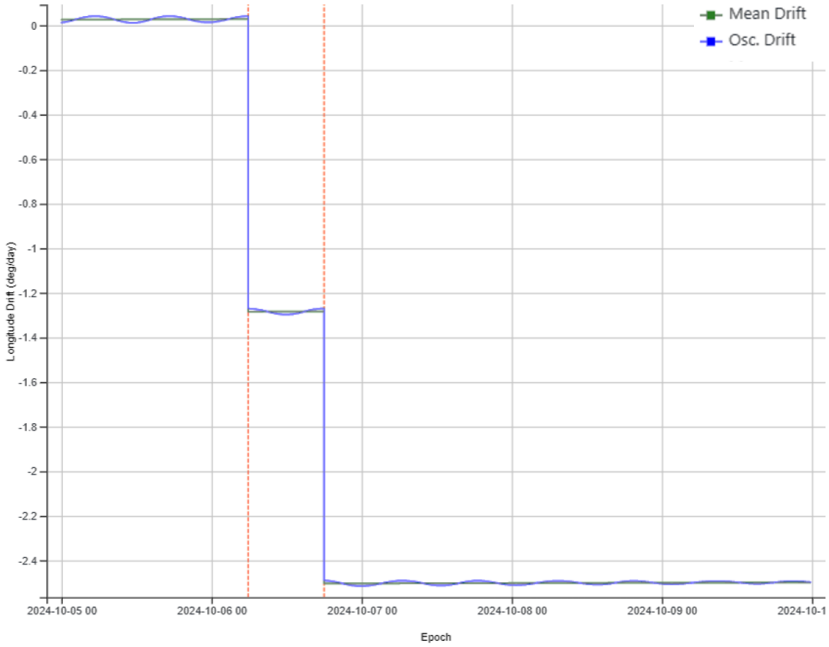

Figure 6.27: Operation 7, drift evolution

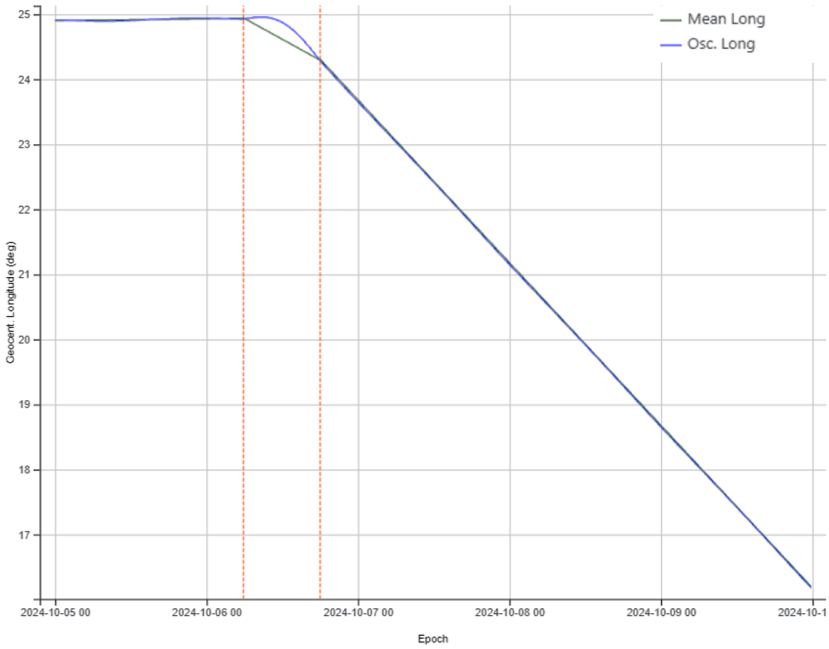

Figure 6.28: Operation 7, longitude evolution

6.4.2 Insertion: Operation 8

The results for this operation do not offer particularly insightful findings for analysis. They are mainly presented to showcase the functionality of the NLPQLP code in optimising the single insertion phase performed via impulsive manoeuvres. The insertion operation is practically equal to Operation 4, with the only difference being that it is computed through impulsive manoeuvres. Therefore, for details regarding the operation features, it is possible to refer to Tab. 6.19 and 6.20. The NLPQLP code is invoked at the end of the analytical algorithm, and both methods converge. A reduction of approximately 0.043 m/s in delta-V is achieved through optimisation. Interestingly, the operation conducted via impulsive manoeuvres proves to be more propulsively expensive than that performed through continuous manoeuvres. This can be attributed to the peculiar method employed by the analytical algorithm, which generates manoeuvres in two different directions.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/22-15:43:24	2024/06/22-15:43:25	West	1	-3.711
2024/06/22-15:43:25	2024/06/22-15:43:26	West	1	-3.711
2024/06/22-15:43:26	2024/06/22-15:43:27	West	1	-3.711
2024/06/23-03:41:27	2024/06/23-03:41:28	West	1	-4.307
2024/06/23-15:39:29	2024/06/23-15:39:30	East	1	3.427
2024/06/23-15:39:30	2024/06/23-15:39:31	East	1	3.427
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/22-16:00:13	2024/06/22-16:00:14	West	1	-3.700
2024/06/22-17:04:44	2024/06/22-17:04:45	West	1	-3.700
2024/06/22-19:45:27	2024/06/22-19:45:28	West	1	-3.700
2024/06/23-03:35:00	2024/06/23-03:35:01	West	1	-4.319
2024/06/23-18:01:07	2024/06/23-18:01:08	East	1	3.416
2024/06/23-19:26:58	2024/06/23-19:26:59	East	1	3.416

Table 6.36: Operation 8, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.725 \cdot 10^{-5}$	$-0.696 \cdot 10^{-4}$	$0.833 \cdot 10^{-4}$
Numerical	$0.999 \cdot 10^{-4}$	$0.276 \cdot 10^{-4}$	$0.881 \cdot 10^{-4}$

Table 6.37: Operation 8, orbital targets errors at the end of the manoeuvre period

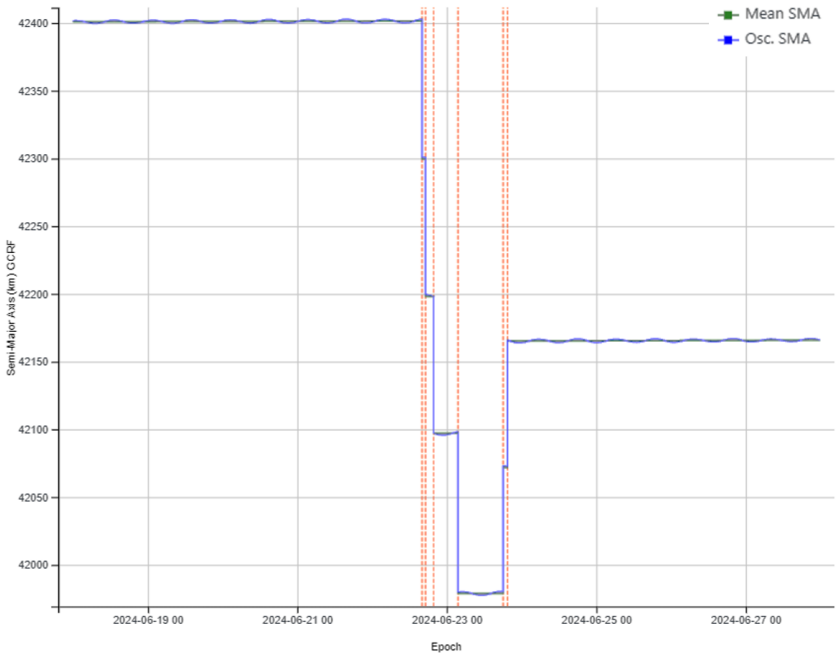

Figure 6.29: Operation 8, semi-major axis evolution

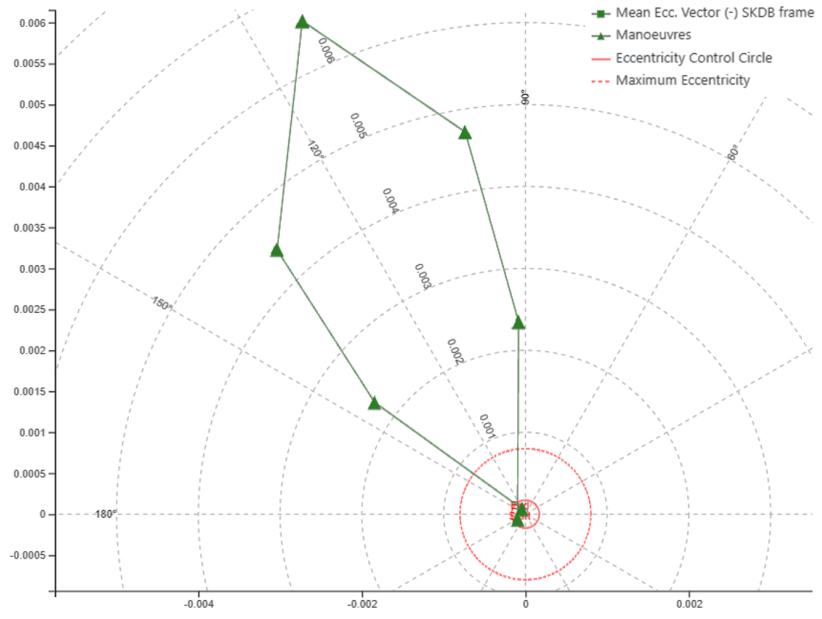

Figure 6.30: Operation 8, mean eccentricity vector evolution

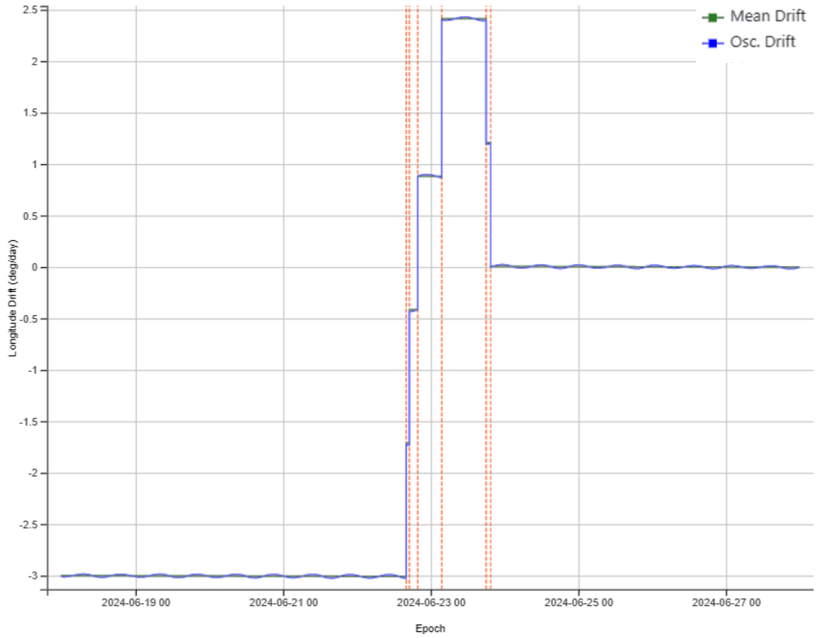

Figure 6.31: Operation 8, drift evolution

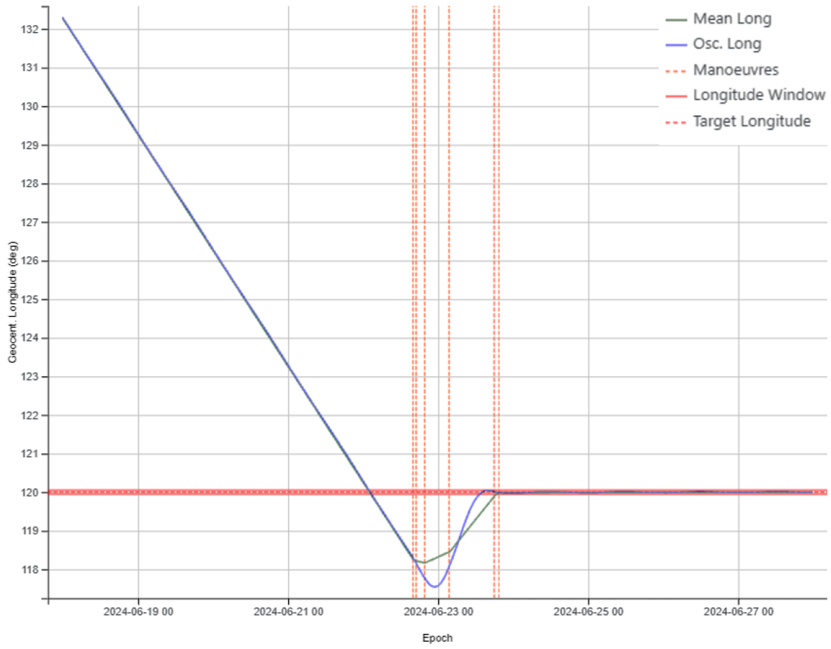

Figure 6.32: Operation 8, longitude evolution

6.4.3 Full Relocation: Operation 9

The simulation is conducted by invoking the NLPQLP code at the conclusion of the analytical method. The details and orbital targets of the full relocation operation analyzed are outlined in Tab. 6.38, 6.39, and 6.40.

	Operation 9
Initial longitude - λ_0 (deg)	139.13
Initial semi-major axis - a_0 (km)	42164.8
Initial drift - D_0 (deg/day)	0.003
Initial eccentricity - e_0	0.0001
Simulation start time	2024/06/10-00:00:00
Simulation end time	2024/06/29-00:00:00
Manoeuvre window start time	2024/06/11-00:00
Manoeuvre window end time	2024/06/28-00:00
Operation Type	Full Relocation

Table 6.38: Analysed relocation operation, initial state and general overview

	Operation 9
Longitude target - λ_t (deg)	-
Drift target - D_t (deg/day)	-3
Eccentricity target - e_t	0.0

Table 6.39: Analysed relocation operation, extraction orbit targets

	Operation 9
Longitude target - λ_t (deg)	118.0
Drift target - D_t (deg/day)	0.0
Eccentricity target - e_t	0.0

Table 6.40: Analysed relocation operation, insertion orbit targets

The most interesting aspect of the obtained results lies in achieving the same solution for the extraction phase between the analytical and numerical methods, while a different one is attained for the insertion phase. Specifically, as can be inferred by examining Tab. 6.41 and 6.42, the manoeuvres constituting the extraction (the first two in the respective lists presented in Tab. 6.41) are identical for both the analytical and optimised solutions. In fact, as observed in Operation 7, the optimiser returns the same solution received as the initial guess. In contrast, for the insertion manoeuvres, the NLPQLP algorithm undergoes 6 iterations to refine the input manoeuvre list and achieves a reduction of 0.04 m/s in delta-V.

Analytical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/12-05:56:21	2024/06/12-05:56:22	East	1.000	5.002
2024/06/12-17:54:23	2024/06/12-17:54:24	East	1.000	3.535
2024/06/18-13:49:48	2024/06/18-13:49:49	East	1.000	1.584
2024/06/19-01:47:50	2024/06/19-01:47:51	West	1.000	-4.173
2024/06/19-13:45:52	2024/06/19-13:45:53	West	1.000	-3.011
2024/06/19-13:45:53	2024/06/19-13:45:54	West	1.000	-3.011
Numerical Solution				
Start date (UTC)	End date (UTC)	Dir.	Duration (s)	ΔV (m/s)
2024/06/12-05:56:21	2024/06/12-05:56:22	East	1.000	5.002
2024/06/12-17:54:23	2024/06/12-17:54:24	East	1.000	3.535
2024/06/18-17:31:29	2024/06/18-17:31:30	East	1.000	1.566
2024/06/19-01:49:37	2024/06/19-01:49:38	West	1.000	-4.122
2024/06/19-13:06:56	2024/06/19-13:06:57	West	1.000	-3.027
2024/06/19-16:13:38	2024/06/19-16:13:39	West	1.000	-3.027

Table 6.41: Operation 9, manoeuvres data

Solution	D (deg/day)	λ (deg)	e
Analytical	$-0.572 \cdot 10^{-10}$	N/A	$0.118 \cdot 10^{-10}$
Numerical	$-0.572 \cdot 10^{-10}$	N/A	$0.118 \cdot 10^{-10}$

Table 6.42: Operation 9, orbital targets errors at the end of the extraction manoeuvres

Solution	D (deg/day)	λ (deg)	e
Analytical	$0.516 \cdot 10^{-5}$	$0.275 \cdot 10^{-4}$	$0.312 \cdot 10^{-4}$
Numerical	$-0.893 \cdot 10^{-5}$	$-0.994 \cdot 10^{-4}$	$0.999 \cdot 10^{-4}$

Table 6.43: Operation 9, orbital targets errors at the end of the insertion manoeuvres

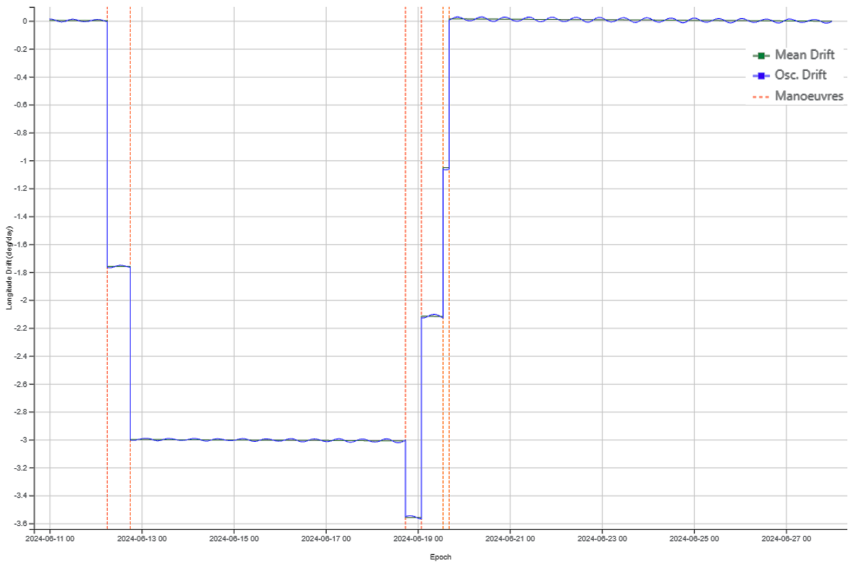

Figure 6.33: Operation 9, drift evolution

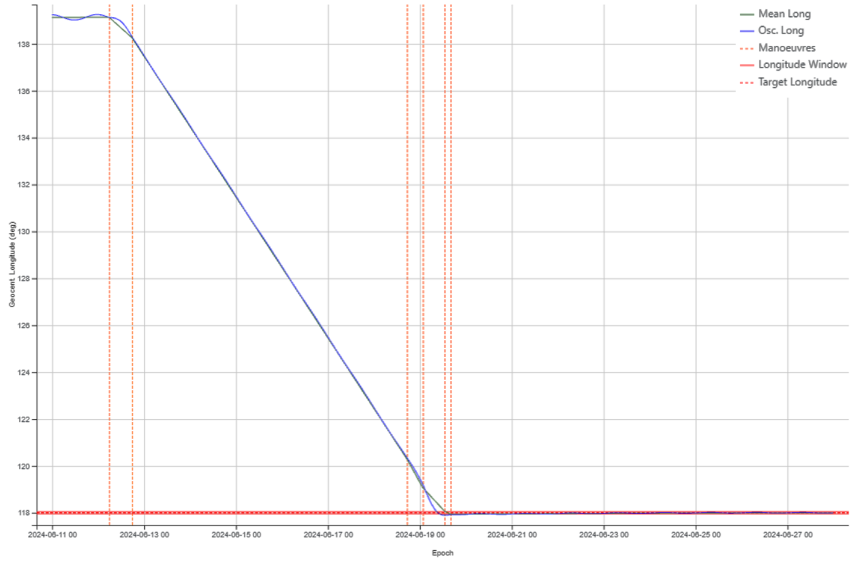

Figure 6.34: Operation 9, longitude evolution

6.5 Final Assessment

It is finally possible to summarize the savings achieved through the optimisation process for the various analyzed operations. Specifically, in Tab. 6.44, the obtained reductions in terms of manoeuvre duration for continuous manoeuvres and total ΔV for impulsive manoeuvres are illustrated. Based on these values, it is also possible to estimate the propellant mass saved due to the manoeuvre optimisation. This allows to observe the results of the optimisation process from a more practical standpoint. In particular, to evaluate the propellant savings for impulsive manoeuvres, the Tsiolkovsky equation has been utilized, while for continuous manoeuvres, the estimation has been made by simply multiplying the propellant flow rate by the manoeuvre duration decrease. The notation used in Tab. 6.44 is summarized in the following list:

- CS_{analyt}: convergence status of the analytical method for the operation under consideration.
- d_s: manoeuvres duration saving due to optimisation. This applies only to operations performed through continuous manoeuvres.
- ΔV_s: saving in terms of ΔV achieved through optimisation. This applies solely to operations carried out through impulsive manoeuvres.
- m_p^s: saving in propellant mass in grams due to optimisation.

Operation	CS_{analyt}	d_s	ΔV_s	m_p^s (g)
Op.1	Negative	4 sec	-	0,2
Op.2	Positive	2 sec	-	0,1
Op.2*	Negative	50 min	-	149,2
Op.3	Negative	6 sec	-	0,3
Op.4	Positive	3 sec	-	0,2
Op.5	Negative	9 sec	-	0,5
Op.6*	Negative	36 min	-	106,6
Op.7*	Positive	-	0,228 m/s	3,0
Op.8	Positive	-	0,043 m/s	0,6
Op.9	Positive	-	0,037 m/s	0,5

Table 6.44: Overview of optimisation results for the investigated operations
(* denotes that the optimiser is invoked at the first iteration of the analytical method)

The reduction in total ΔV for the impulsive case is on the order of tenths of cm/s. Once again, it can be observed that invoking the optimiser at the first iteration of the traditional method results in a significant reduction in the problem's cost function value: tens of minutes for continuous manoeuvres and tens of cm/s for impulsive manoeuvres. Finally, manoeuvre optimisation generally results in savings of several hundred milligrams in terms of propellant mass. While this decrease may not be substantial, it remains a positive outcome, clearly making it advantageous to execute manoeuvres planned through the optimisation process.

Conclusion

The methodology and results concerning the implementation of an SQP method, applied by the Fortran subroutine NLPQLP, for nonlinear constrained optimisation of relocation manoeuvres for geostationary satellites, have been presented and discussed.

The thesis work was conducted at GMV, and the formulation and implementation of the optimisation process were carried out within GMV's Flight Dynamics Software, ***FocusSuite***, specifically in the frame of the corporate Fortran manoeuvre planning library for GEO satellites.

Simulations of various scenarios, crucial for verifying method functionality and obtaining the results depicted in this document, were predominantly executed using GMV's platform ***FocusSuite***.

The achieved results are excellent, demonstrating the successful operation of the optimiser for all manoeuvre phases associated with satellite relocation activity, conducted in different temporal scenarios and considering distinct mission requirements. Notably, the produced code enables the optimisation of single extraction and insertion phases, as well as entire full relocations, considering both continuous and impulsive manoeuvres. The analysis and testing have primarily focused on a real satellite constellation equipped with electric thrusters. However, the optimiser implementation has also been extended to impulsive manoeuvres, enabling its utilization for satellites outfitted with chemical thrusters. This decision further strengthened the implemented optimisation procedure of the NLPQLP subroutine.

The performed tests underscore the significant potential of the SQP algorithm in handling constrained optimisation problems with numerous constraints, even in the presence of numerous solution variables. The adaptability of the NLPQLP code enables adaptation and resolution of a wide range of scenarios, facilitating the straightforward addition of new constraints.

The flexibility of the implementation allows for conducting the optimisation procedure either at the end of the analytical method, which generates the initial guess for the NLPQLP routine, or at its first iteration. In the former case, the optimiser simply refines the manoeuvre list received as input, confirming the analytical method's capability to discover a viable solution near the optimum. In the latter case, the consistency of the improvements after the optimisation is significantly enhanced, as it starts from a less precise and approximate solution. Clearly, the convergence of the method in this situation turns out to be a little slower in some tests, as expected since the initial solution is less accurate.

The optimiser turns out to be dependent, on the initial guess solution; this sensitivity must be considered before the optimisation process. The analyses conducted have also emphasized that the physical problem of GEO satellite relocation demonstrates a notable sensitivity to the initial orbital conditions, resulting in variations in solutions when these are altered.

The most significant and interesting result lies in the possibility of directly using the optimiser as a solver for computing relocation manoeuvre plans fulfilling several constraints, given its capability to efficiently resolve various complex problems highlighted by the various simulations investigated. This is particularly significant, as in many cases, it has been observed that the analytical method manages to identify a manoeuvre list, but it is not always able to satisfy the numerous and stringent constraints that may be imposed by the relocation operation. This challenge is especially pronounced when considering the low-thrust performance of the electric thruster and the presence of eclipses.

The implementation of the optimiser not only enhances and expands upon the functionality of the analytical method but also surpasses its abilities. This is evident in its capacity to manage problems with numerous constraints, finding solutions across diverse situations, and generating feasible manoeuvre plans even under challenging conditions. Additionally, the enhanced solver potentially reduces the necessity for extensive operator input.

Possible future developments and refinements of the present work may include:

- Incorporating additional constraints, such as collinearity constraints between satellites, which, due to lack of time, could not be included. Given the solver's significant potential in solving highly constrained problems, it would be worth exploring the possibility of adding new constraints previously unconsidered for the analytical method, given its lesser ability to find a solution.

- Additionally, more extensive analysis and testing of impulsive manoeuvre cases are needed. The optimiser has been tested only in a few simulations, yielding good results but not verifying its performance in every possible mission scenario.

- Further enhancement of the optimiser's implementation and its utilization as a solver warrants consideration. Specifically, the application of specific changes should be studied to enable dynamic variation in the number and direction of manoeuvres during the optimisation process. This adaptation is likely to enhance solution quality and lead to better outcomes, thereby augmenting the solver capabilities of the optimiser. By allowing it to explore a broader range of solutions during the optimisation problem-solving process, the potential for finding optimal solutions across various scenarios can be significantly improved.

- Adjust the problem formulation to incorporate the option of executing manoeuvres with reduced thrust during eclipse periods, a strategy employed in certain cases for specific satellite constellations.

- Further tests and validation are necessary to ensure the functioning of the optimiser in all possible situations. This comprehensive approach will help verify the robustness and reliability of the optimiser across diverse scenarios, thereby instilling confidence in its performance and effectiveness.

Appendix A: *FocusSuite®*

The ***FocusSuite*** platform, as referenced in [10], stands as GMV's innovative Flight Dynamics solution tailored for satellite operations. It comprises several Flight Dynamics tools and products designed to assist satellite operators in diverse tasks including satellite control, space debris conjunction assessment, and mission analysis. Through this software, GMV endeavors to meet the varied requirements of its wide clientele, striving to provide user-friendly and adaptable products compatible with different platforms. Its primary features encompass:

- Comprehensive mission operations analysis across all mission phases (launch and early orbit phase operations, on-orbit operations, etc.) and orbital regimes (LEO, MEO, GEO, interplanetary missions, etc.)
- Space surveillance and tracking capabilities to facilitate collision avoidance services
- Advanced configurability and intuitive user interfaces
- Elevated reliability and security

It furnishes end-to-end solutions for satellite operations throughout their life-cycle, drawing from a library of proven missions, alongside enabling automation and planning for these operations made easier by the presence of scripts that can be customized for specific needs. The software also offers an open framework aimed at facilitating the development and evolution of GMV's products.

With its open and extensible nature, ***FocusSuite*** ensures easy interoperability with third-party services, scalability, and flexibility. Its architecture is compatible with both public and private clouds, and it interfaces seamlessly with several Control Centers.

Some of the functionalities allowed by the platform's tools include date and state vector conversions between different reference frames and formats, ephemeris retrieval for the Sun and Moon, orbit propagation and determination, and others. Among the various products included in ***FocusSuite***, it is worth mentioning at least ***FocusGEO***, an integrated application that offers complete life cycle support for geostationary satellites, encompassing a variety of missions within its collection.

In summary, the ***FocusSuite*** software is exploited by GMV to provide for all operational needs, covering orbit determination and propagation, manoeuvre planning and optimisation, station-keeping, and more with platform portability. These results are achieved through the utilization of existing software, a database-driven approach, and graphical interactive reporting functions, ensuring mission success and efficiency. Other specified goals of the platform and further details are illustrated on GMV's web page, as indicated in [10].

Bibliography

[1] Thompson R., *Satellites, Geo-stationary orbits and Solar Eclipses*, 2019.

[2] Soop E.M., *Handbook of Geostationary Orbits*, 1994. Vol. 3 of Space Technology Library, Springer Science & Business Media, illustrated edition.

[3] De Santi S., *Missions towards NEAs with Departure from Lagrangian Points L4 and L5*, 2020.

[4] Montenbruck O., Gill E., *Satellites Orbit*, 2000. Springer-Verlag Berlin Heidelberg.

[5] Blanco V.M., McCuskey S.W., *Basic physics of the solar system*, 1961. Addison-Wesley series in the engineering sciences, Addison-Wesley Publishing Company.

[6] Powell M.J.D., *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, 1978. In: Nonlinear Programming 3, O.L. Mangasarian, R.R. Meyer, S.M. Robinson eds., Academic Press.

[7] Schittkowski K., *NLPQL: A Fortran subroutine solving constrained nonlinear programming problems*, 1985/86. Annals of Operations Research, Vol. 5, 485–500.

[8] Goldfarb D., Idnani A., *A numerically stable method for solving strictly convex quadratic programs*, 1983. Mathematical Programming, Vol. 27, 1–33.

[9] Schittkowski K., *NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search*, July 2015. User's Guide, Version 4.2.

[10] GMV, *FocusSuite*, 2024. `https://www.gmv.com/en/products/space/focussuite`.

[11] H. M. Haranas I.I., *Gauss planetary equations in a non-singular gravitational potential*, 2010.

[12] Powell M.J.D., *A fast algorithm for nonlinearly constrained optimization calculations*, 1978. In: Numerical Analysis, G.A. Watson ed., Lecture Notes in Mathematics, Vol. 630, Springer.

[13] Di Pillo G., *Metodi per la Soluzione di Problemi di Programmazione Nonlineare.*

[14] Stoer J., *Foundations of recursive quadratic programming methods for solving nonlinear programs*, 1985. In: Computational Mathematical Programming, K. Schittkowski, ed., NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 15, Springer.

[15] Thomas D., *A Comparison of GEO Satellites Using Chemical and Electric Propulion*, 2016.

[16] Rocket & Space Technology, *Orbital Mechanics*, 2024. `https://www.braeunig.us/space/orbmech.htm`.

[17] Cornell University, *Sequential quadratic programming*, 2024. `https://optimization.cbe.cornell.edu/index.php?title=Sequential_quadratic_programming`.

[18] Softwareentwicklung Schittkowski GmbH, *NLPQLP*, 2024. `https://www.schittkowski.de/numericalsoftware_nlpqlp.php`.

[19] Li L., Zhang J., Zhao S., Qi R., Li Y., *Autonomous onboard estimation of mean orbital elements for geostationary electric-propulsion satellites*, 2019. In: Aerospace Science and Technology 94.

[20] Boggs P.T., Tolle J.W., *Sequential Quadratic Programming*, 1996. Acta Numerica, Vol. 4, 1-51.

[21] Nocedal J., *Large Scale Unconstrained Optimization*, 1996.

[22] Bonnans J.F., Gilbert J.C., Lemaréchal C.,Sagastizábal C.A., *Numerical Optimization Theoretical and Practical Aspects*, 2006.

[23] Battipede M., *Introduction to Ground Tracks*.

[24] GMV, *Deployment of an SBAS system demonstration in Southern Africa*, 2016.

[25] Soler T., Eisemann D.W., *Determination of Look Angles To Geostationary Communication Satellites*, 2019.

[26] Kohn D., *The Teledesic Network: Using Low-Earth-Orbit Satellites to Provide Broadband, Wireless, Real-Time Internet Access Worldwide*, 2016.

[27] Wertz J.R., Larson W.J., *Space mission analysis and design*, 1999. In: Space Technology Library, Springer Dordrecht.

[28] Nocedal J., Wright S.J., *Numerical Optimization*, 2006. Springer.

[29] ESA, *Orbits*, 2024. `https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Orbits`.

[30] Hanson D., Peronto J., Hilderbrand D. , *NOAA's Eyes in the Sky – After Five Decades of Weather Forecasting with Environmental Satellites, What Do Future Satellites Promise for Meteorologists and Society?*, 2015.

[31] Australian Government, *BOM*, 2024.
http://www.bom.gov.au/.

[32] The Planetary Society, *Coverage of a geostationary satellite at Earth*, 2024.
https://www.planetary.org/space-images/coverage-of-a-geostationary.

[33] SpaceRef, *NOAA Satellites, Scientists Monitor Mt. St. Helens for Possible Eruption*, 2024.
https://spaceref.com/press-release/noaa-satellites-scientists-monitor\
-mt-st-helens-for-possible-eruption/.

[34] NASA, *GOCI*, 2019.
https://oceancolor.gsfc.nasa.gov/data/goci/.

[35] Science On a Sphere, *GOES-R: Today's Satellite for Tomorrow's Forecast*, 2024.
https://sos.noaa.gov/catalog/datasets/goes-r-todays-satellite-for-tomorrows-\
forecast/.

[36] FAA, *Satellite Navigation-WAA -How It Works*, 2024.
https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/
techops/navservices/gnss/waas/howitworks.