POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Multi-revolution Low Thrust Trajectory Optimization for Exploration of the Martian Moons

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Elia Luigi Altieri**

Student ID: 969309
Advisor: Prof. Michèle Roberta Lavagna
Co-advisors: Jacopo Prinetto
Academic Year: 2022-23

We are all in the gutter, but some of us are looking at the stars

---

Oscar Wilde

Abstract

Despite the increasing interest gained by the Martian moons, no exploration mission towards Deimos or Phobos has been successfully carried out so far, and only few conceptual studies have been proposed.
This work aims to provide an analysis alternative to those present in the literature [1][2][3], by investigating the possibility to exploit small-size spacecraft [4] and low-thrust propulsion [5] to explore the Martian system with a low-cost mission.
The employment of small-size spacecraft implies a major criticality that arises from the extremely low thrust-over-weight ratio of these platforms, which translates into very large times of flight. In a planetocentric scenario the orbital transfers result therefore in a huge number of revolutions around the body of the main attractor, leading to heavy optimization problems with an unsustainable computational load.
To overcome this problem, an optimization tool based on a direct method is proposed, featuring Control Parametrization [6] and Orbital Averaging [7] to keep the computational effort as low as possible. The optimization algorithm is developed in two versions to solve either the minimum time of flight and the minimum propellant problem.
The algorithm is validated on a GTO-GEO transfer scenario against literature's results [8], and shows good convergence properties and performances. The tool is then applied to the Mars exploration scenario. Particularly, the exploration framework proposed by Canales et al. [2] is adopted, and its analysis is further extended. Two transfers are optimized: the first one connects the spacecraft release orbit to a Deimos resonant orbit, while the second one links the resonant orbit with Phobos's science orbit.
Both the minimum time of flight and the minimum propellant problem are solved and a preliminary mission analysis is proposed.

Keywords: Mission Analysis, Trajectory Optimization, Low Thrust

Abstract in lingua italiana

Nonostante il crescente interesse verso le lune di Marte, nessuna missione esplorativa verso Deimos o Phobos è mai stata condotta con successo, e solo pochi studi concettuali sono stati proposti fino a questo momento. Il presente lavoro ha l'intenzione di fornire un'analisi alternativa a quelle presenti in letteratura [1][2][3], investigando la possibilità di sfruttare satelliti di piccola taglia [4] e una propulsione a bassa spinta [5] per esplorare il sistema Marziano con una missione a basso costo.
L'impiego di satelliti di piccola taglia implica un'importante criticità che insorge a causa del rapporto spinta-su-peso estremamente basso di queste piattaforme, che si traduce in tempi di volo molto prolungati. In uno scenario planetocentrico i trasferimenti orbitali sono pertanto caratterizzati da un elevato numero di rivoluzioni intorno al pianeta, producendo pesanti problemi di ottimizzazione con un carico computazionale non sostenibile. Per ovviare a questa problematica si propone un tool di ottimizzazione basato su un metodo diretto, caratterizzato da una strategia di Parametrizzazione dei Controlli [6] e di Orbital Averaging [7] per poter mantenere il carico computazionale il più basso possibile. L'algoritmo di ottimizzazione è sviluppato in due versioni per risolvere sia il problema del minimo tempo di volo che quello del minimo consumo di propellente.
L'algoritmo è validato nello scenario terrestre di un trasferimento GTO-GEO, confrontandolo con i risultati presenti in letteratura [8], e mostra ottime prestazioni in convergenza. Il tool è poi applicato allo scenario di esplorazione Marziana. In particolare, il framework proposto da Canales et al. [2] è adottato, e la sua analisi è ulteriormente estesa ed approfondita. Due trasferimenti sono ottimizzati: il primo connette l'orbita di rilascio del satellite con un' orbita risonante rispetto a Deimos, mentre il secondo collega l'orbita risonante con l'orbita operativa in prossimità di Phobos. Sia il problema di minimo tempo di volo che quello di minimo consumo di propellente sono risolti ed è proposta un'analisi di missione preliminare.

Parole chiave: Analisi di Missione, Ottimizzazione di Traiettorie, Bassa Spinta

Contents

1 | Introduction

Since the very beginning of the space exploration era one of the major concerns for mission designers is to design spacecraft trajectories in such a way to reduce the cost of the mission and achieve its objectives in the most efficient way. For example, one may want to reduce the propellant consumption to have a lighter spacecraft, which in turn implies a higher payload mass fraction, or a lower launch cost. Another objective may be the minimization of the time of flight to reduce the degradation of the spacecraft due to exposure to the harsh space environment.

Over the years several techniques have been developed to assist the Mission Analyst in finding the optimal trajectory for a specific mission. Formally the problem consists in finding a trajectory and a sequence of controls that minimize an objective function (typically the propellant consumption or the time of flight) while satisfying some constraints (for example departure and arrival conditions) [5]. The choice of the objective and of the constraints, as well as the choice of the optimization algorithm, strongly depends on the nature of the problem. Let's consider for instance a spacecraft equipped with a chemical thruster. Such an engine has been widely employed since the very beginning of the space exploration era. It features a very high thrust over weight ratio T/W but suffers from poor Specific Impulse. The firings of this engine provide a large amount of thrust in a relatively short period, compared to the time scale of the full trajectory.

For this reason the action of a chemical engine can be considered an impulsive event, leading to a parameter optimization problem [9].

As technology evolved a new class of engines destined to change the way of conceiving space missions made its appearance in the last years of the Century: the Low Thrust Engines. This classification encompasses a variety of technologies, from ion engines to solar sails. This class of devices features a much lower thrust magnitude with respect to its chemical counterpart, but a higher Specific Impulse, thus allowing for extremely reduced propellant consumption, and enabling a new class of missions.

The revolutionary possibilities of this technology have only been realized in recent decades, with the stunning success of missions such as Deep Space 1, Dawn, and Hayabusa 1 and 2 [10][11][12]. When dealing with low-thrust propulsion the thrust is continuously ap-

plied along the trajectory to slowly accelerate the spacecraft. Therefore, thrust cannot be approximated as an impulsive event anymore, and the spacecraft dynamics has to be represented by a set of seven differential equations like those reported in Equation 1.1 (expressed in cylindrical coordinates). The first six coupled non-linear differential equations describe the motion of the spacecraft in terms of its position and velocity, while the additional differential equation (Tsiolkovski equation) describes the evolution of the spacecraft's mass, which has to be included in the state vector.

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \omega \\ \dot{z} = v_z \\ \dot{v}_r = -\frac{\mu}{\sqrt{(r^2+z^2)^3}} r + r\omega^2 + \frac{T cos(\beta) sin(\alpha)}{m} \\ \dot{\omega} = \frac{1}{r}\left(\frac{T cos(\beta) cos(\alpha)}{m} - 2\dot{r}\omega\right) \\ \dot{v}_z = -\frac{\mu}{\sqrt{(r^2+z^2)^3}} z + \frac{T sin(\beta)}{m} \\ \dot{m} = -\frac{T}{I_{sp} g_0} \end{cases} \tag{1.1}$$

The employment of these engines requires new trajectory design and optimization techniques, posing challenging optimal control problems.

Thanks to the possibility of exploiting such a light and efficient engine, over the last decade the space sector has been revolutionized by the appearance of small-size spacecraft, like CubeSat, MicroSat, and NanoSat. These platforms played a fundamental role in opening access to space thanks to their reliability, efficiency, and low cost.

So far these devices have found wide application, especially in low Earth orbit (Earth Observation, IoT, Technology Demonstration). More recently CubeSats have also been employed in interplanetary missions [4], as the Lunar CubeStas developed in the framework of the Artemis I program (Lunar IceCube [13], Lunar Flashlight [14], LunarH-Map [15]), or the Italian CubeSat LICIACube [16] operated in the context of the DART mission [17]. The efficiency and the low cost of these platforms make them very interesting candidates for future interplanetary exploration missions but at the cost of an increased complexity in the mission design. Indeed, the intrinsic limitation in CubeSats's performances in terms of power and propulsion, makes it necessary to rethink the way a mission is developed, accounting for these limitations.

Particularly, one of the most critical aspects to be considered is the trajectory design.

In the case of exploration CubeSats in fact, the already complex trajectory design process presents further constraints due to limited control authority. Furthermore, an interplane-

tary CubeSat mission, hardly can rely upon a dedicated launch. Indeed, all the aforementioned interplanetary CubeSats exploited larger missions to reach their target, through a Piggy-Back launch service.
As a consequence, CubeSats trajectory design shall be flexible enough to cope with the driver requirements of larger missions and accommodate a range of release conditions. Another critical aspect to take into consideration in the trajectory design process, is the extremely low T/W ratio of CubeSats. This value significantly affects the time of flight, complicating the overall trajectory design and optimization, as will be explained later.

1.1. State of the Art and Motivation of the Research

Although the natural satellites of Mars, Phobos and Deimos, are starting to be considered potentially key destinations in support of the Martian exploration program, their exploration has been quite limited so far, mainly carried out through flybys of orbiters approaching Mars. The first close-up imaging of Phobos and Deimos was conducted by the Viking mission [18]. Direct exploration of Phobos was attempted by Phobos 1 (1988), Phobos 2 (1988–1989), and Phobos-Grunt (2011), but they failed except for limited data acquisition by Phobos 2 in the vicinity of the moon [19].
Currently, the only approved mission targeting the Martian moons is the MMX mission (Martian Moons eXploration), developed by JAXA and planned to be launched in 2024 [20]. MMX is a sample return mission from Phobos, whose trajectory preliminary design [21] implies a sequence of three impulsive maneuvers to acquire the orbit of the moon.
Only few conceptual studies instead have been carried out in the past to assess the possibility of employing low-thrust propulsion in the Martian system.
For example, Englander [1] used Edelbaum's method [22] to obtain an approximated solution of the spiraling transfer from Mars's SOI to the orbital altitudes of Phobos and Deimos. Canales [2] instead interfaced a Q-law guidance scheme with a direct collocation algorithm to generate low-thrust transfer trajectories between the Martian moons.
The present study aims to advance the current state of the art by investigating the possibility of exploring the Martian systems through small-size spacecraft and very low-thrust propulsion. Particularly, a trajectory optimization tool is proposed featuring good flexibility to accommodate a wide range of initial conditions (the release conditions at the Martian system), and capable of solving either the minimum time of flight and the minimum propellant consumption problem.
As previously discussed, the major criticality of the problem resides in the extremely low T/W. In fact, when dealing with CubeSats, the typical T/W is in the order of 10^{-4}-10^{-5}, implying large transfer times, and in turn, an extremely high number of revolutions around the main attractor (in the order of hundreds or even thousands).
As a consequence, standard optimization algorithms become unpractical due to unaffordable computational load.
Great effort was spent in the past in order to find alternative approaches to solve such a complex problem. In the early 1960's T. N. Edelbaum proposed analytical solutions to optimize continuos-thrust transfers between inclined circular orbits [22] providing an excellent preliminary design tool. Colasurdo and Canalino then extended Edelbaum's analysis to account for Earth shadowing [23].
Kechichian also developed an analytical method for obtaining coplanar orbit-raising ma-

neuvers in the presence of Earth shadow where eccentricity is constrained to remain zero [24]. One of the most popular tools in low-thrust trajectory optimization then is SEPSPOT [7]. It is a program for computing optimal Earth-orbit transfers through Solar Electric Propulsion and it relies on an Indirect Method based on the "Calculus of Variations" which consists in solving the TPBVP through a shooting method.
However, it is sensible to initial guesses for costate variables, and often a converged solution can only be achieved by using known solutions for the initial guess.
Conversely, Conway and Scheel [25] used a direct transcription method for obtaining optimal many-revolution, planar continuos-thrust Earth orbit transfers.
They used a Parallel Shooting method by discretizing the state and control's time histories, resulting in a large-size NLP problem. Betts [26] used instead a direct collocation method to solve a 578-revolution transfer problem bringing forth an optimization with 416123 variables requiring a tremendous computational effort.
Most recently hybrid methods have been developed to combine the flexibility of direct approaches with the computational efficiency of indirect ones.
For example, Ilgen [27] parametrized the costate histories by piecewise linear functions, and solved the optimal transfer problem through NLP. Gao [28] [29] instead parametrized the control law in the form of the true optimal control, converting the optimal transfer problem into a parameter optimization one and solving it through NLP.
Similarly, Jimenez-Lluva [8] proposed a hybrid approach where the analytical nature of the problem formulation was leveraged to reduce the number of optimization variables. A differential evolution algorithm was employed leading to a global optimization process with a practical computational effort.
In the present investigation the work of Kluever and Oleson [6] was taken as a starting point. They obtained the optimal control law time history for a minimum-time Earth-orbit transfer by optimizing the blend of three different extremal feedback control laws.
This idea was extended in the present study to solve also the minimum propellant consumption problem, and applied to the specific scenario of the Martian moons exploration.

1.2. Low Thrust Trajectory Optimization

The trajectory optimization process consists in finding the sequence of controls, in terms of thrust magnitude and direction ($T(t)$, $\alpha(t)$, $\beta(t)$), that allows to minimize the objective function while satisfying the constraints. From a mathematical point of view the problem is referred to as Optimal Control Problem [30].

1.2.1. The Optimal Control Problem

An Optimal Control Problem (OCP) concerns the determination of a set of inputs $u(t)$ to a system that minimizes (or maximizes) a desired performance index J. Accordingly with the general formulation [30], such index is expressed as follows:

$$J = \Phi[x(t_f), u(t_f), t_f] + \int_{t_i}^{t_f} L[x(t), u(t), t]dt \tag{1.2}$$

where the first term represents the terminal cost, which is evaluated at the final time t_f, while the second term is the path cost and is provided by the integral of the Lagrangian $L[x(t), u(t), t]$ from initial to final time. The minimization of J is subjected to the system dynamics

$$\dot{x} = f[x(t), u(t), t] \tag{1.3}$$

and to the boundary constraints

$$\psi[x(t), u(t), t] = 0 \tag{1.4}$$

Different approaches exist for handling this problem [31]. The main distinction is between Indirect Methods and Direct Methods.

1.2.2. Indirect Methods

Indirect Methods analytically construct the necessary and sufficient conditions for optimality by applying the Euler-Lagrange theorem to the continuous OCP, to transform it into a TPBVP. Then they discretize these conditions and solve them numerically through shooting or collocation techniques.
The major advantage of such methods is that they are in general more accurate than direct approaches, and guarantee more reliable error estimates. Both these benefits come from the analytical derivation of the optimality conditions. On the other hand, the region

of convergence of these methods is smaller if compared to the one of Direct Methods, meaning that a more precise initial guess will be required to initialize the method [32]. Moreover, the setup of the problem is further complicated by the need to initialize the adjoint variables, which have not a clear physical interpretation. For these reasons Indirect Methods are not flexible, since, in case of changes in the original OCP (like the addition of new variables or constraints), Euler Lagrange equations shall be applied again to generate a new TPBVP. Finally, to obtain an accurate solution for an Indirect Method, necessary and sufficient conditions must be derived analytically, which can be challenging [32]

1.2.3. Direct Methods

Direct Methods discretize the OCP itself, typically converting the original problem into a Nonlinear Programming Problem (NLP). This process, known as *transcription*, does not require extensive analytical derivation, thus it is more straightforward to implement a Direct Method with respect to an Indirect one. Among the advantages of this class of methods, there is flexibility, enabling the adaptability to changes in the original problem. Another strength of Direct Methods is their robustness: they are more likely to converge even with a poor initial guess. Despite all these advantages Direct Methods suffer from some drawbacks with respect to Indirect Methods: first of all the solution offered by such methods is often a local optimum close to the initial guess and does not offer information about potential improvements. Conversely, Indirect Methods provide helpful information regarding potential improvements through the costate values. Another drawback is the large size of the NLP problem generated by the transcription approach, which implies an extensive computational effort.
Nevertheless, as computer technology advanced and computational capabilities increased over the years, large size problems became less prohibitive, and therefore Direct Methods gained attention thanks to their advantages with respect to Indirect Methods [32][33].

1.2.4. Global Optimization

Both Direct and Indirect Methods provide only locals optimal solutions in the neighborhood of the initial guess. For this reason a distinct set of optimization techniques have been proposed, capable of finding the global optimal solution. These approaches are based on heuristic algorithms and can obtain solutions even when initialized with a random initial guess. Naturally this huge advantage comes with a drawback: heuristic algorithms can handle only few variables, otherwise the process becomes computationally prohibitive. Despite this drawback, evolutionary algorithms are successfully employed for trajectory design. Furthermore, there are examples of these approaches coupled with

Direct or Indirect Methods, to provide the strengths of both strategies [34] [35].

To summarize: Indirect Methods tend to produce more accurate solutions than Direct Methods, but they are more difficult to construct and solve. A typical approach is to first compute an approximation of the solution through a Direct Method, and then use this to initialize an Indirect Method [9].

1.3. Transcription Methods

Direct Methods rely on a process called *transcription* that consists in discretizing the original OCP to pose it as a NLP problem. Naturally, the transcription process implies a loss of accuracy as the continuous problem is represented through a finite set of variables. Therefore the transcription method should be tailored to reduce as much as possible the approximation errors. In astrodynamics applications a trajectory is typically discretized with respect to time through a mesh of nodes delimiting a set of segments.

$$t_i < t_2 < ... < t_{N-1} < t_f \tag{1.5}$$

States and controls are discretized over the mesh, generating a set of optimization variables for the NLP problem

$$y = [\mathbf{x}_1, \mathbf{u}_1, ..., \mathbf{x}_N, \mathbf{u}_N]^T \tag{1.6}$$

Typically the discretization is carried out through shooting or collocation schemes which provide the objective and the constraints for the NLP problem.

1.3.1. Shooting Method

The Shooting Method is one of the most known approaches to solve a Boundary Value Problem, and consists in converting it into an Initial Value Problem (IVP).
Given a general second-order TPBVP:

$$\begin{aligned} &\ddot{x} = f(t, x, \dot{x}), \qquad t_i < t < t_f \\ &x(t_i) = \alpha \\ &x(t_f) = \beta \end{aligned} \tag{1.7}$$

the Shooting Method consists in solving the associated IVP

$$\begin{aligned} &\ddot{x} = f(t, x, \dot{x}), \qquad t_i < t < t_f \\ &x(t_i) = \alpha \\ &\dot{x}(t_i) = v \end{aligned} \tag{1.8}$$

where v must be chosen such that the solution satisfies the final boundary condition $x(t_f) = \beta$. In other words, this approach requires selecting the proper "initial slope" so that the solution will "hit the target", which is the condition $x(t) = \beta$ at time t_f [36].
In the case of nonlinear dynamics, iterative methods are required in order to look for the correct value of v, like Newton's Method or Secant's Method. From here comes the name "Shooting Method", because the initial slope is searched by iteratively integrating the dynamics (shooting), evaluating the final error with respect to the desired final condition, and adjusting the initial slope on the basis of the committed error.
In the case of a low thrust trajectory optimization the optimization variable to find will not be the initial slope (that is the trajectory initial derivatives), but rather the control law. A variant of the Shooting Method exists, which is called Multiple Shooting.
The working principle is exactly the same, but the dynamics is not integrated from the initial time up to the final time with a unique integration process. The trajectory is indeed subdivided into segments and each of them is integrated individually.

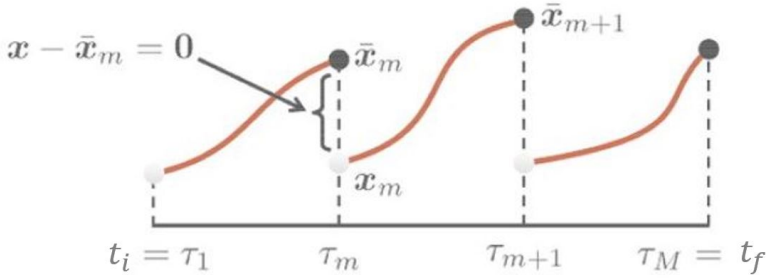

Figure 1.1: Multiple Shooting Scheme

Therefore, with Multiple Shooting some additional constraints will be present in the optimization problem, that is the continuity constraints between the trajectory segments, often referred to as defect constraints.
A major consequence of this approach is that the computational cost of the optimization increases because of the additional constraints and optimization variables.

On the other hand, this scheme allows to improve the overall convergence robustness because in this way the integration error propagation is prevented [37].

1.3.2. Collocation Method

The idea at the basis of the collocation technique is to approximate the continuous state $x(t)$ and control $u(t)$ through piecewise polynomials. Polynomials are used because they can be represented through a small set of coefficients and because the computation of their integrals and derivatives is easy [9].
In contrast to Shooting Methods, where the dynamics was explicitly integrated, in Collocation Method it is implicitly satisfied by converting the differential equations of motion into a set of algebraic constraints (defect constraints) and by forcing the polynomials to satisfy these constraints at the collocation nodes of the mesh. The basic derivation for a cubic polynomial is here reported [33].
Firstly, the trajectory is divided into a series of small arcs, here referred to as *phases*.
Let's consider the generic k-th phase extending from time t_k to time t_{k+1} and with duration $t_{k+1} - t_k = h$.

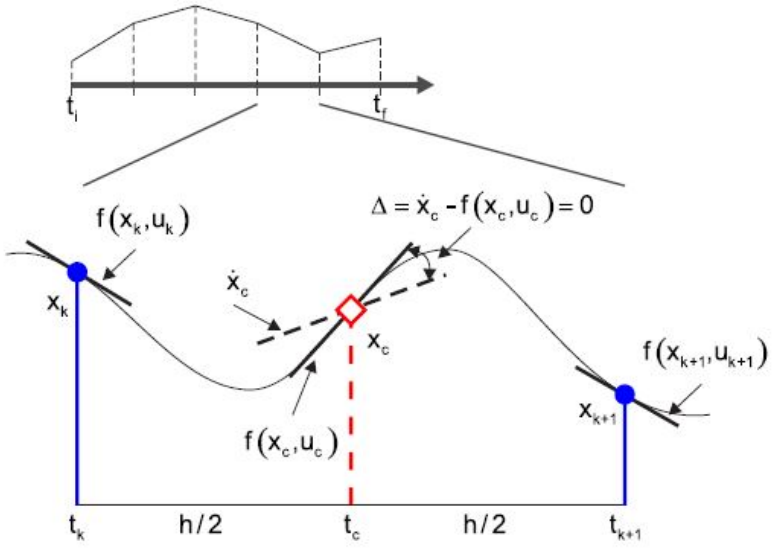

Figure 1.2: Hermite-Simpson Collocation Scheme

The states x_k, x_{k+1} and the controls u_k, u_{k+1} at the boundaries of the phase are provided by the optimization variables, while the corresponding state derivatives $\dot{x}_k, \dot{x}_{k+1}$ come from the evaluation of the system dynamics: $f(x_k, u_k)$,$f(x_{k+1}, u_{k+1})$. The state function $x(t)$ inside the phase instead is approximated with a cubic polynomial:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{1.9}$$

whose derivative is

$$\dot{x}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{1.10}$$

where a_0, a_1, a_2, a_3 are the polynomial coefficients. Their values can be retrieved by requiring the polynomial to satisfy the dynamics at the phase boundaries, that is forcing the polynomial and its derivative evaluated at times t_k and t_{k+1}, to be equal to the states x_k and x_{k+1} and their derivatives $\dot{x}_k$ and $\dot{x}_{k+1}$. Assuming $t_k = 0$ and $t_{k+1} = h$ this condition translates into the following system:

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{1.11}$$

Reversing Equation 1.11 the polynomial coefficients can be retrieved:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{h^2} & -\frac{2}{h} & \frac{3}{h^2} & -\frac{1}{h} \\ \frac{2}{h^3} & \frac{1}{h^2} & -\frac{2}{h^3} & -\frac{1}{h^2} \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} \tag{1.12}$$

At this point the polynomial is fully defined and it can be evaluated at the *collocation point*, which is the inner point located in the middle of the phase, at time $t_c = \frac{h}{2}$.

$$\begin{aligned} x_c &= x\left(\frac{h}{2}\right) = \frac{1}{2}(x_k + x_{k+1}) + \frac{h}{8}\left[f(x_k, u_k) - f(x_{k+1}, u_{k+1})\right] \\ \dot{x}_c &= \dot{x}\left(\frac{h}{2}\right) = -\frac{3}{2h}(x_k - x_{k+1}) - \frac{1}{4}[f(x_k, u_k) + f(x_{k+1}, u_{k+1})] \end{aligned} \tag{1.13}$$

The dynamics is then enforced at the collocation point by requiring that the derivative of the polynomial in that point is equal to the system dynamics evaluated with the state x_c, provided by the polynomial at time t_c. From this condition the defect constraint is derived:

$$\Delta = \dot{x}_c - f(x_c, u_c) \tag{1.14}$$

The derivation reported so far relies on a collocation scheme with a single collocation node in the middle of the phase, which is called Hermite-Simpson scheme. However, higher order schemes do exist with multiple collocation points within each phase, like the Gauss-Lobatto scheme. These high-order methods allow for higher accuracy at the price of an increased computational effort.

Compared to Collocation Methods, Shooting Methods tend to create small, dense nonlinear programs with fewer decision variables. In general, Shooting Methods are best suited for applications where the control trajectory is simple but the dynamics must be computed accurately, while Collocation Schemes are mostly employed when dynamics and controls require similar accuracy but the control structure is complex and not known a priori [9].

1.4. Thesis Outline

The report of the study tackles in depth the developed algorithm, exploring its functionalities and its advantages and drawbacks. Then the results of its application to the Martian exploration scenario are presented providing a preliminary mission analysis. The thesis is organized as follows:

- **Introduction**: In the first chapter the study is introduced by explaining the motivations at the basis of the research and providing the mathematical context in which the algorithm has been developed.

- **Dynamics Models**: The models utilized are presented, including the reference frames and the coordinate systems employed in the work. The formulation of the dynamics is then introduced together with the Orbital Averaging technique and the orbital perturbations models considered.

- **Methodology**: The algorithm logic is presented and detailed in its twofold versions: the one for the minimum time of flight problem, and that for the minimum propellant consumption.

- **Validation and Testing**: The algorithm is validated against literature scenarios and the performance comparison is presented. The algorithm is then stressed including an attitude constraint.

- **Mars Exploration Scenario**: The algorithm is run on the scenario of interest. Either the minimum time of flight and the minimum propellant consumption problems are solved, and a preliminary mission analysis is proposed.

- **Conclusions and Future Developments**: The results of the analysis are presented, highlighting the criticalities and proposing further improvements to the algorithm.

2 | Dynamics Models

2.1. Reference Frames

In the present work two reference frames were used to describe the orbital dynamics: an Inertial Reference Frame centered on the main attractor, and a Local Reference Frame centered on the spacecraft.

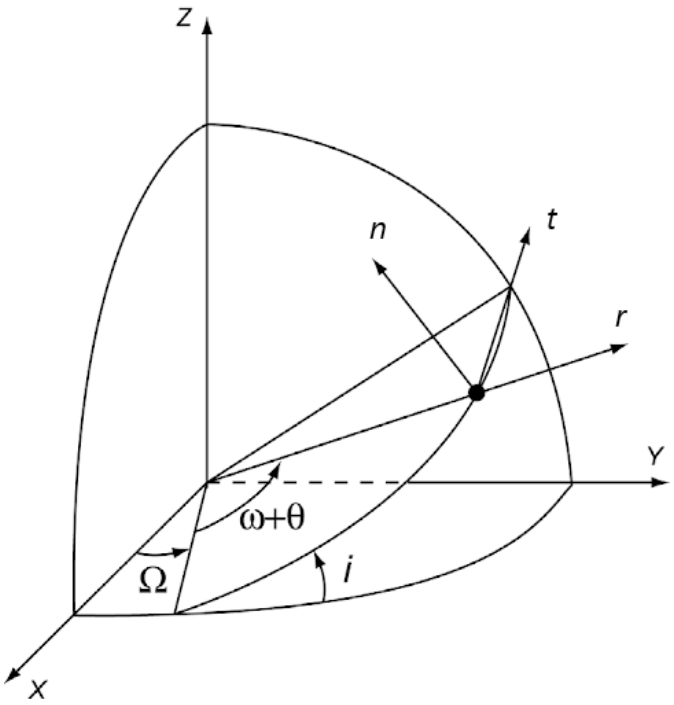

Figure 2.1: Local and Inertial Reference Frames

For the former a J2000 reference frame is adopted with the x-axis on the equatorial plane, pointing towards the direction of the vernal equinox at noon of the 1^{st} January 2000, the z-axis pointing toward the celestial North Pole, and the y-axis taken on the equatorial plane to complete the frame. This Inertial Frame was used to describe the spacecraft's trajectory and the Sun ephemerides.

The Local Reference Frame is instead a co-rotating frame whose axes are identified by the radial, tangential, and normal directions, as defined below.

$$\begin{aligned}\hat{r} =& \frac{\mathbf{r}}{||\mathbf{r}||} \\ \hat{t} =& \frac{(\mathbf{r}\times\mathbf{v})\times\mathbf{r}}{||\mathbf{r}\times\mathbf{v}||||\mathbf{r}||} \\ \hat{n} =& \frac{\mathbf{r}\times\mathbf{v}}{||\mathbf{r}\times\mathbf{v}||}\end{aligned} \tag{2.1}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the spacecraft position and velocity vector, respectively. The Local Reference Frame was used to represent the perturbing accelerations as well as the thrust steering angles.

2.2. Coordinate Systems

In a Trajectory Optimization problem the most simple state representation is that of Cartesian coordinates. Nevertheless, this choice is also the most disadvantageous because the position and velocity variables will be rapidly changing along the trajectory, therefore a huge amount of discrete points would be required to catch these fast variations [5].
For this reason the best way to represent the dynamics in a Trajectory Optimization problem is through a set of coordinates featuring a very slow variation, like the Orbital Elements. In the present work the state is represented by means of the Modified Equinoctial Elements (MEE) [38], instead of the classical Keplerian elements.
Indeed, although MEE do not have a clear physical interpretation, they are not affected by the presence of singularities, which instead constitute a limiting factor of Keplerian Elements when dealing with equatorial and circular orbits.
The MEE can be easily retrieved from Keplerian elements through the following conversions:

$$\begin{aligned}p =& a(1-e^2) \\ f =& e\cos(\omega+\Omega) \\ g =& e\sin(\omega+\Omega) \\ h =& \tan(i/2)\cos/\Omega \\ k =& \tan(i/2)\sin/\Omega \\ L =& \Omega+\omega+\theta\end{aligned} \tag{2.2}$$

As stated above, MEE feature very poor physical meaning, therefore, throughout the present investigation, even Keplerian Elements were adopted in the representation of the results to have physical insights into the problem's dynamics.
Keplerian Elements were retrieved from MEE through the following conversions:

$$\begin{aligned} a =& \frac{p}{1-f^2-g^2} \\ e =& \sqrt{f^2+g^2} \\ i =& 2\tan^{-1}\left(\sqrt{h^2+k^2}\right) \\ \omega =& \tan^{-1}(g/f) - \tan^{-1}(k/h) \\ \Omega =& \tan^{-1}(k,h) \\ \theta =& L - (\Omega+\omega) \end{aligned} \tag{2.3}$$

For the purpose of trajectory graphical representation the following conversions [26] were used to extract the Cartesian state from MEE:

$$\mathbf{r} = \begin{bmatrix} \frac{r}{s^2}\left(\cos L + \alpha^2 \cos L + 2hk \sin L\right) \\ \frac{r}{s^2}\left(\sin L + \alpha^2 \sin L + 2hk \cos L\right) \\ \frac{2r}{s^2}\left(h \sin L - k \cos L\right) \end{bmatrix} \tag{2.4}$$

$$\mathbf{v} = \begin{bmatrix} -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(\sin L + \alpha^2 \sin L - 2hk \cos L + g - 2fhk + \alpha^2 g\right) \\ -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(-\cos L + \alpha^2 \cos L + 2hk \sin L - f + 2ghk + \alpha^2 f\right) \\ \frac{2}{s^2}\sqrt{\frac{\mu}{p}}\left(h \cos L + k \sin L + fh + gk\right) \end{bmatrix} \tag{2.5}$$

$$\begin{aligned} \alpha^2 =& h^2 - k^2 \\ s^2 =& 1 + h^2 + k^2 \\ r =& \frac{p}{w} \\ w =& 1 + f\cos L + g \sin L \end{aligned} \tag{2.6}$$

2.3. Equations of Motion

In general the spacecraft flight dynamics is expressed by the following set of first-order differential equations

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_{pert} \tag{2.7}$$

where μ is the gravitational parameter of the attracting body around which the spacecraft is orbiting, $\mathbf{r}$ is the spacecraft position vector, and the term $\mathbf{a}_{pert}$ includes all the orbital perturbations, like the effect of a non-spherical gravitational field, the attraction of a third body other than the main attractor, and the thrust acceleration.
In general no analytical solution can be found for equation 2.7 and numerical methods are required. A different approach is the so called Method of Variation of Orbital Elements. This method converts the equation of motion 2.7 into another set of equations that describe the variation of each Keplerian osculating element in terms of a perturbing potential. These equations are referred to as Lagrange's Planetary Equations.
Starting from this set, the perturbing accelerations can be included to obtain the Gauss form of Lagrange's Planetary Equations [39].
As previously mentioned the Keplerian Elements result in singularities under specific conditions, and therefore their use is unattractive. For this reason in the present work a formulation of Lagrange's Planetary Equation in terms of MEE is employed [38]:

$$\begin{aligned}
\dot{p} &= \frac{dp}{dt} = \frac{2p}{w}\sqrt{\frac{p}{\mu}}a_t \\
\dot{f} &= \frac{df}{dt} = \sqrt{\frac{p}{\mu}}\left[a_r \sin L + \left[(w+1)\cos L + f\right]\frac{a_t}{w} - (h\sin L - k\cos L)\frac{g a_n}{w}\right] \\
\dot{g} &= \frac{dg}{dt} = \sqrt{\frac{p}{\mu}}\left[-a_r \cos L + \left[(w+1)\sin L + g\right]\frac{a_t}{w} + (h\sin L - k\cos L)\frac{g a_n}{w}\right] \\
\dot{h} &= \frac{dh}{dt} = \sqrt{\frac{p}{\mu}}\frac{s^2 a_n}{2w}\cos L \\
\dot{k} &= \frac{dk}{dt} = \sqrt{\frac{p}{\mu}}\frac{s^2 a_n}{2w}\sin L \\
\dot{L} &= \frac{dL}{dt} = \sqrt{\mu p}\left(\frac{w}{p}\right)^2 + \frac{1}{w}\sqrt{\frac{p}{\mu}}\left(h\sin L - k\cos L\right)a_n
\end{aligned} \tag{2.8}$$

In equation 2.8 the terms a_r, a_t, a_n represent the perturbing accelerations in the Local Reference Frame described in section 2.1.
To complete the set of equation of motion the mass equation must be added as well:

$$\dot{m} = \frac{dm}{dt} = -\frac{T}{I_{sp} g_0} \tag{2.9}$$

For implementation purposes the dynamics equations 2.8 were assembled in matrix format:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \mathbf{M}\mathbf{a}_{pert} + \mathbf{b} \tag{2.10}$$

where

$$\mathbf{M} = \begin{bmatrix} 0 & \frac{2p}{w}\sqrt{p\mu} & 0 \\ \sqrt{\frac{p}{\mu}}\sin L & \sqrt{\frac{p}{\mu}}\frac{1}{w}\left[(w+1)\cos L + f\right] & \sqrt{\frac{p}{\mu}}\frac{g}{w}\left[h\sin L - k\cos L\right] \\ -\sqrt{\frac{p}{\mu}}\cos L & \sqrt{\frac{p}{\mu}}\frac{1}{w}\left[(w+1)\sin L + g\right] & \sqrt{\frac{p}{\mu}}\frac{f}{w}\left[h\sin L - k\cos L\right] \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\cos L}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\sin L}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\left[h\sin L - k\cos L\right] \end{bmatrix} \tag{2.11}$$

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p}\left(\frac{w}{p}\right)^2 \end{bmatrix}^T \tag{2.12}$$

$$\mathbf{a}_{pert} = \begin{bmatrix} a_r & a_t & a_n \end{bmatrix}^T \tag{2.13}$$

Note that the perturbation term $\mathbf{a}_{pert}$ in equation 2.10 includes either the thrust acceleration and the orbital perturbation accelerations.

2.4. Orbital Averaging

The dynamics equations reported in Eq. 2.8 have to be integrated numerically to obtain the trajectory evolution of the spacecraft.
Due to the nature of the problem studied in this investigation, which implies a very large time of flight for the transfer, a classical integration scheme applied directly to this dynamics formulation would result in an unsustainable computational load.
To overcome this criticality an Orbital Averaging technique was adopted, which allows for taking large integration steps (in the order of days) to reduce the computational effort without a significant loss of accuracy.

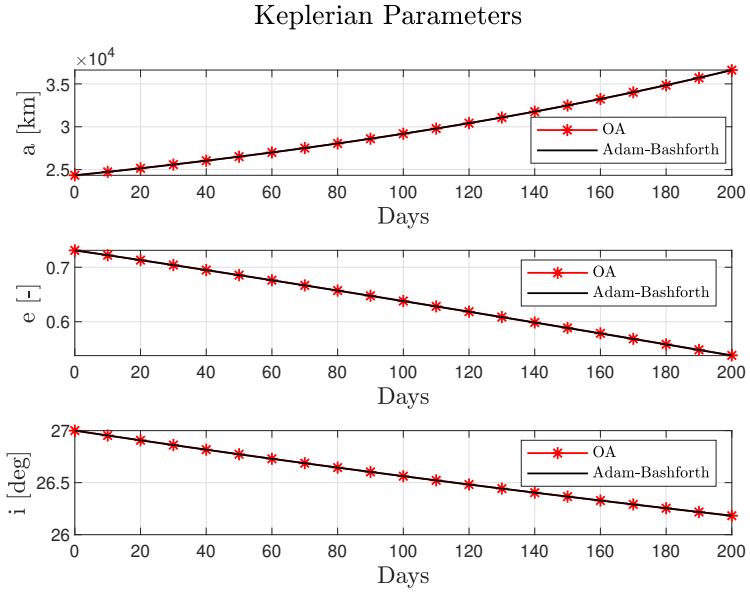

Figure 2.2: Orbital Averaging comparison with Adam-Bushforth integrator

The idea at the base of the Orbital Averaging technique is to evaluate, at each integration step, the mean variations of the orbital elements along one single revolution, and use them to propagate the dynamics.
The advantages of Orbital Averaging come at the cost of losing the control on the fast-varying element, the true longitude L, whose information is deleted by the averaging process. Formally the average variation of the orbital elements is computed as

$$\frac{d\bar{\mathbf{x}}}{dt} \approx \frac{1}{P}\int_0^{2\pi} \dot{\mathbf{x}}\frac{dt}{dL}dL \tag{2.14}$$

where P is the orbital period and the term dL/dt can be approximated by equation 2.15,

under the assumption that the thrust acceleration is much smaller than the gravitational acceleration [28].

$$\frac{dL}{dt} \approx \sqrt{\mu p}\left(\frac{1+f\cos L+g\sin L}{p}\right)^2 \tag{2.15}$$

The time rates of the mean equinoctial elements ($\bar{\mathbf{x}} = [\bar{p}\ \bar{f}\ \bar{g}\ \bar{h}\ \bar{k}]^T$) are obtained expanding equation 2.14 [28]:

$$\begin{aligned}\frac{d\bar{x}_l}{dt} = \frac{1}{2\pi}(1-\bar{f}^2-\bar{g}^2)^{3/2}.\Bigg[\int_{L_1}^{L_2}\frac{f_l^T(\bar{\mathbf{x}},L,\boldsymbol{u})}{(1+\bar{f}\cos L+\bar{g}\sin L)^2}dL + \\ \int_0^{2\pi}\frac{f_l^p(\bar{\mathbf{x}},L,\boldsymbol{a}_{pert})}{(1+\bar{f}\cos L+\bar{g}\sin L)^2}dL\Bigg]\end{aligned} \tag{2.16}$$

where $\bar{\mathbf{x}} = [\bar{x}_1\ \bar{x}_2\ \bar{x}_3\ \bar{x}_4\ \bar{x}_5] = [\bar{p}\ \bar{f}\ \bar{g}\ \bar{h}\ \bar{k}]^T$ and $l = 1, 2, ..., 5$.
Similarly, the averaged mass flow rate is computed as

$$\frac{d\bar{m}}{dt} = \frac{1}{2\pi}(1-\bar{f}^2-\bar{g}^2)^{3/2}\int_{L_1}^{L_2}\frac{\dot{m}}{(1+\bar{f}\cos L+\bar{g}\sin L)^2}dL \tag{2.17}$$

The terms $f_l^T(\bar{\mathbf{x}}, L, \boldsymbol{u})$ and $f_l^p(\bar{\mathbf{x}}, L, \boldsymbol{a}_{pert})$ represent the dynamics of the osculating orbital elements due to thrust and perturbations respectively, computed through equations 2.10 removing the sixth row of the matrix, since the true longitude's dynamics is not considered when performing Orbital Averaging, as explained previously.
The integral limits L_1 and L_2 are the starting and ending angles (measured in true longitude) of the thrust arc along the orbital revolution.
This formulation allows to take into account planet shadowing just by selecting L_1 and L_2 as the angles of exit and entry in the eclipse condition respectively. Note that the evaluation of mean state derivatives requires the a priori knowledge of the control law $\mathbf{u}$, that is provided by the control parametrization strategy explained in section 3.2. The previous integrals were computed numerically using a Gauss Legendre quadrature technique.
In the present work the Orbital Averaging technique described above was applied to evaluate the averaged dynamics within a classical single-step Runge-Kutta 4 (RK4) integration scheme.

2.5. Orbital Perturbations

As shown in equation 2.16 the spacecraft averaged dynamics is evaluated by considering the thrust acceleration and the orbital perturbations as two separate contributions. The former acts only along the thrusting arcs while the latter acts on the entire orbit.
Both these accelerations are expressed in the Local Frame.

2.5.1. Thrust

The thrust affects the mass dynamics through equation 2.9 and produces a spacecraft acceleration whose components are defined by the pitch and yaw thrust steering angles.

$$\mathbf{a}_T = a_T[\sin\alpha\cos\beta, \cos\alpha\cos\beta, \sin\beta]^T \tag{2.18}$$

The pitch angle α is defined on the orbital plane, measured clockwise from the tangential direction. The yaw angle β is instead the out-of-plane angle.

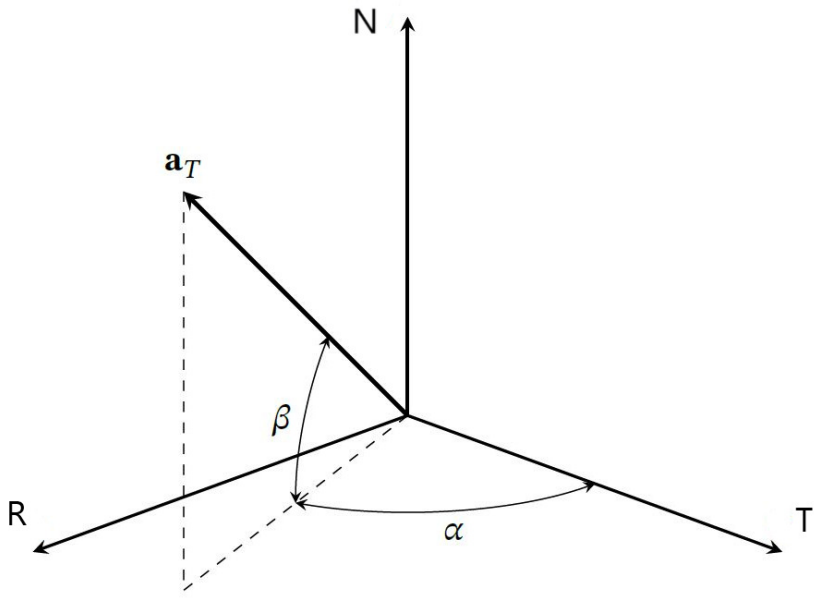

Figure 2.3: Pitch and Yaw angles in Local Reference Frame

2.5.2. Eclipse

When dealing with Low-Thrust trajectories, one of the most common types of propulsion is Solar Electric Propulsion (SEP). This kind of propulsion implies the use of an engine driven by electrical current, and for this reason, strongly relying on the Solar Arrays.
This hardware configuration has major implications for the trajectory design, as the thrust amount will depend on the illumination of the Solar Arrays.
Therefore, the eclipse of the spacecraft is not a negligible condition, as the thrust drops to zero when the spacecraft enters the planet's shadow.
As discussed in section 2.4, Orbital Averaging allows to account for this condition by imposing the angular position of eclipse entry and exit as limits of the averaging integrals. Such angles were retrieved in this work by using the following total eclipse model [40]:

$$\begin{aligned} |\rho_p - \rho_s| &> \eta \\ \rho_p &> \rho_s \end{aligned} \tag{2.19}$$

where η is the angle from the center of the planet to the center of the Sun, as seen by the spacecraft, and ρ_p and ρ_s are the angular radius of the planet and of the Sun respectively.

$$\begin{aligned} \rho_s &= \sin^{-1}(R_s/D_s) \\ \rho_p &= \sin^{-1}(R_p/D_p) \\ \eta &= \cos^{-1}(\hat{\mathbf{D}}_s \cdot \hat{\mathbf{D}}_p) \end{aligned} \tag{2.20}$$

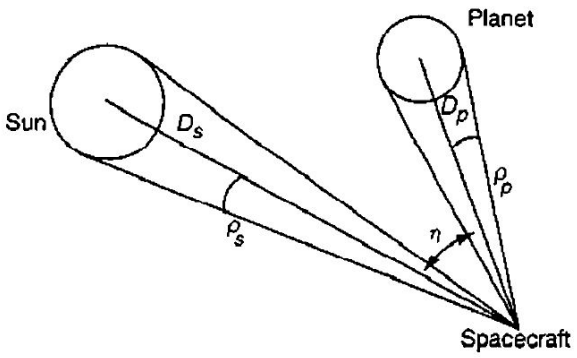

Figure 2.4: Eclipse Model

For the evaluation of the eclipse conditions Sun ephemerides were retrieved from SPICE [41] at 1^{st} January 2000.

2.5.3. J2 Perturbation

In order to represent a realistic dynamics, planet's non-spherical gravity distribution was considered in the present study.
Particularly, the effect of planet's oblateness (J_2 effect) is accounted for by introducing the following perturbing accelerations expressed in the Local Reference Frame [42]:

$$\begin{aligned} a_r &= -\frac{3\mu J_2 R_p^2}{2r^4}\left[1 - 12\frac{(h\sin L - k\cos L)^2}{s^4}\right] \\ a_t &= -\frac{12\mu J_2 R_p^2}{r^4}\cdot\frac{(h\sin L - k\cos L)(h\cos L + k\sin L)}{s^4} \\ a_n &= -\frac{6\mu J_2 R_p^2}{r^4}\cdot\frac{(h\sin L - k\cos L)(1 - h^2 - k^2)}{s^4} \end{aligned} \tag{2.21}$$

where J_2 is a planetary coefficient that identifies the second zonal harmonic of the gravitational potential, and R_p is the radius of the planet.
The impact of this perturbation is especially significant in the case of elliptical and inclined orbits, on which a drift is induced on both the Right Ascension of Ascending Node (RAAN) and the Argument of Pericentre.

3 | Methodology

As mentioned in previous sections the intrinsic criticality of the investigated problem lies in the huge amount of transfer revolutions. Hence, classical Direct Transcription methods would lead to large NLP problems, with hundreds or even thousands of free variables and constraints.

On the other hand, Indirect Methods would allow to reduce significantly the computational effort, but they would not guarantee any flexibility, because in case of changes in the original OCP (due for example to a variation in the initial or final transfer conditions), necessary and sufficient conditions must be derived analytically from the beginning.

Since flexibility is an essential requirement for the present work, a direct approach was employed: the OCP defined in section 1.2 was replaced with a Nonlinear Programming problem, which will be described in the following sections.

To reduce the number of optimization variables and to alleviate the computational load, a control parameterization strategy was adopted.

While classical Direct Transcription methods imply a discretization of both states and controls along the trajectory, the idea at the basis of the developed algorithm is to only parametrize the control profile $\mathbf{u}(t)$ with as few parameters as possible, which become the optimization variables of the NLP problem.

3.1. Nonlinear Programming

The general Nonlinear Programming Problem [30] consists in finding the n-vector $\mathbf{x}^T = (x_1, .., x_n)$ of the optimization variables that minimize a scalar objective function

$$F(\mathbf{x}) \tag{3.1}$$

subject to the constraints

$$\mathbf{c}_L \leq \mathbf{c}(\mathbf{x}) \leq \mathbf{c}_U \tag{3.2}$$

and to the simple bounds

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \tag{3.3}$$

A classical way to solve this problem is through gradient-based methods.
The Lagrangian of this problem [37] is assembled through the Lagrange Multipliers $\boldsymbol{\lambda}$:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = F(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x}) \tag{3.4}$$

The necessary conditions for a point $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ to be an optimum require solving the system

$$\begin{aligned} \nabla_x L(x, \boldsymbol{\lambda}) &= \mathbf{g}(\mathbf{x}) - \mathbf{G}^T(\mathbf{x})\boldsymbol{\lambda} = 0 \\ \nabla_\lambda L(x, \boldsymbol{\lambda}) &= -\mathbf{c}(\mathbf{x}) = 0 \end{aligned} \tag{3.5}$$

where $\mathbf{g}(\mathbf{x}) = \nabla_x F(\mathbf{x})$ and $\mathbf{G}$ are the gradient of the objective function and the Jacobian of the equality constraint vector, respectively. The system 3.5 can be solved through the multi-variable version of the Newton method starting from an initial guess $(\mathbf{x}, \boldsymbol{\lambda})$. The iterative correction $(\Delta\mathbf{x}, \Delta\boldsymbol{\lambda})$ for the solution is constructed by solving the linear system

$$\begin{bmatrix} \mathbf{H}_L & -\mathbf{G}^T \\ \mathbf{G} & 0 \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} -\mathbf{g} \\ -\mathbf{c} \end{Bmatrix} \tag{3.6}$$

also referred to as Karush-Kuhn-Tucker (KKT) condition. In this work the Matlab solver `fmincon` with the "interior-point" algorithm was employed to solve the NLP problem.

3.2. Control Parametrization

To parametrize the control and obtain the optimization variables, the strategy proposed by Kluever and Oleson [6] was adopted in this work. With this approach the thrust magnitude is assumed to be constant and only its direction is optimized.
Three different extremal feedback control laws are derived analytically such that at each revolution they maximize the time rate of change of the first three orbital elements: a, e, i. Each of these analytical control laws, if applied, would drive the corresponding orbital element to the target state in the minimum time.
By properly blending these control laws it is possible to retrieve the optimal steering program that allows to acquire the target orbit with a minimum time transfer.
The extremal feedback control laws are derived analytically starting from the governing equation of motion of the corresponding orbital elements [6]:

$$\dot{a} = (2a^2 v/\mu) a_T \cos\nu \tag{3.7}$$

$$\dot{e} = (a_T/v)[2(e + \cos\theta)\cos\nu + (r/a)\sin\theta\sin\nu] \tag{3.8}$$

$$\dot{i} = \frac{r}{\sqrt{\mu a(1-e^2)}} \cos(\omega+\theta) a_T \sin\beta \tag{3.9}$$

where v is the orbital velocity, a_T the thrust acceleration magnitude, θ is the true anomaly, β is the out-of-plane (yaw) steering angle, and ν is the angle measured from the velocity vector to the projection of the thrust vector onto the orbit plane.
The optimal steering angles that maximize the orbital element's rate are found by setting the following partial derivatives equal to zero:

$$\frac{\partial \dot{a}}{\partial \nu} = -\frac{2a^2 v}{\mu} a_T \sin\nu = 0 \tag{3.10}$$

$$\frac{\partial \dot{e}}{\partial \nu} = \frac{a_T}{v}\left[-2(e+\cos\theta)\sin\nu + \frac{r}{a}\sin\theta\cos\nu\right] = 0 \tag{3.11}$$

$$\frac{\partial \dot{i}}{\partial \beta} = \frac{r}{\sqrt{\mu a(1-e^2)}} \cos(\omega+\theta) a_T \cos\beta = 0 \tag{3.12}$$

From equation 3.10 it is clear that the extremal steering law for maximum $\dot{a}$ is

$$\nu_a = 0 \tag{3.13}$$

that means that the thrust vector is aligned with the velocity vector, as to increase the orbit energy. From equation 3.11 it comes out that the extremal steering law for $\dot{e}$ is

$$\tan \nu_e = \frac{r \sin \theta}{2a(e + \cos \theta)} \tag{3.14}$$

The second partial derivative $\partial^2 \dot{e}/\partial \nu^2$ must be computed as well to determine if equation 3.14 maximizes or minimizes $\dot{e}$.

$$\frac{\partial \dot{e}^2}{\partial \nu^2} = \frac{a_T}{v} \left[-2(e + \cos \theta) \cos \nu - \frac{r}{a} \sin \theta \sin \nu \right] \tag{3.15}$$

From equation 3.14 it is possible to retrieve the expressions of the sine and cosine of the optimal angle ν_e:

$$\sin \nu_e = \frac{r \sin \theta}{[r^2 \sin^2 \theta + 4a^2(e + \cos \theta)^2]^{\frac{1}{2}}} \tag{3.16}$$

$$\cos \nu_e = \frac{2a(e + \cos \theta)}{[r^2 \sin^2 \theta + 4a^2(e + \cos \theta)^2]^{\frac{1}{2}}} \tag{3.17}$$

Substituting equations 3.16 and 3.17 in equation 3.15 it is possible to demonstrate that $\partial^2 \dot{e}/\partial \nu^2$ is a negative quantity at every point of the orbit, resulting in a feedback law 3.14 that always maximizes $\dot{e}$.
Finally, the feedback steering law for the out-of-plane steering angle is derived from equation 3.12 and results in a $\pm \pi/2$ yaw steering angle.
Also in this case it is necessary to evaluate the second partial derivative of equation 3.9:

$$\frac{\partial^2 \dot{i}}{\partial \beta^2} = \frac{-r}{\sqrt{\mu a(1 - e^2)}} \cos(\omega + \theta) a_T \sin \beta \tag{3.18}$$

which, by requiring its negativity, provides the following feedback law for maximum $\dot{i}$

$$\beta_i = \begin{cases} +\pi/2, & if \ \cos(\omega+\theta) > 0 \\ -\pi/2, & if \ \cos(\omega+\theta) < 0 \end{cases} \tag{3.19}$$

Since any out-of-plane steering is essentially wasted when the angle $\omega+\theta$ is close to $\pm\pi/2$ the following steering law is considered instead:

$$\beta_i = (\pi/2)\cos(\omega+\theta) \tag{3.20}$$

Note that the angle ν does not coincide with the pitch angle, as it is only the angle between the velocity vector and the in-plane thrust projection. To retrieve the pitch angle, ν must be summed with the flight path angle γ, measured from the local horizon to the velocity vector. To summarize, the three analytical laws for the steering angles that ensure the maximum variation of the orbital elements within a revolution, are the following:

	Pitch Angle	**Yaw Angle**
Semi-major axis	$\alpha_a = 0 + \gamma$	$\beta_a = 0$
Eccentricity	$\alpha_e = \arctan\left(\frac{r\sin\theta}{2a(e+\cos\theta)}\right) + \gamma$	$\beta_e = 0$
Inclination	$\alpha_i =$ any	$\beta_i = (\pi/2)\cos(\omega+\theta)$

Table 3.1: Extremal Feedback Control Laws

The overall optimal steering program is obtained by the optimal blend of the two in-plane feedback steering laws (3.13, 3.14) and the out-of-plane steering law (3.20).
Let's consider the in-plane thrust direction: the thrust direction $\mathbf{q}_a$ that maximize $\dot{a}$ and the direction $\mathbf{q}_e$ that maximize $\dot{e}$ are defined in equations 3.21.

$$\begin{aligned} \mathbf{q}_a &= [\sin\alpha_a, \cos\alpha_a, 0]^T \\ \mathbf{q}_e &= [\sin\alpha_e, \cos\alpha_e, 0]^T \end{aligned} \tag{3.21}$$

To assemble the blended in-plane thrust direction $\mathbf{q}$ (expressed in Local Frame), and retrieve the overall optimal pitch steering angle, the following formulation was used:

$$\mathbf{q} = \frac{G_a\mathbf{q}_a + G_e\mathbf{q}_e}{||G_a\mathbf{q}_a + G_e\mathbf{q}_e||} = [\sin\alpha, \cos\alpha, 0]^T \tag{3.22}$$

where G_a and G_e are gains used to prioritize the relative influence of the two in-plane steering laws. Similarly, the yaw steering angle is weighted through the parameter G_i that scales the amplitude of the yaw steering profile.

$$\beta = G_i(\pi/2)\cos(\omega + \theta) \tag{3.23}$$

Equations 3.22 and 3.23 provide the parametrization of the thrust direction $\mathbf{u}$ by means of the three weighting gains G_a, G_e, and G_i. Therefore, the optimal steering program

$$\mathbf{u}^*(t) = [\sin\alpha^*\cos\beta^*, \cos\alpha^*\cos\beta^*, \sin\beta^*]^T \tag{3.24}$$

is obtained by computing the optimal set of gains $G_a^*(t)$, $G_e^*(t)$, and $G_i^*(t)$ that minimize the objective function while satisfying the trajectory boundary conditions and the constraints of the problem. In figure 3.1 and 3.2 the in-plane and out-of-plane steering programs are depicted for different values of the control gains G_e and G_i for a single revolution on a slightly elliptical orbit.

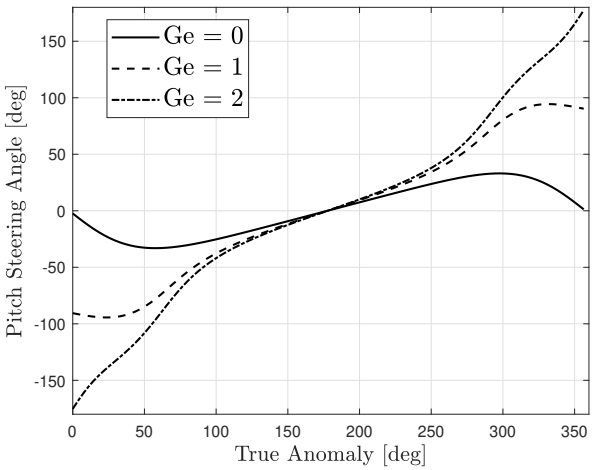

Figure 3.1: Pitch Angle

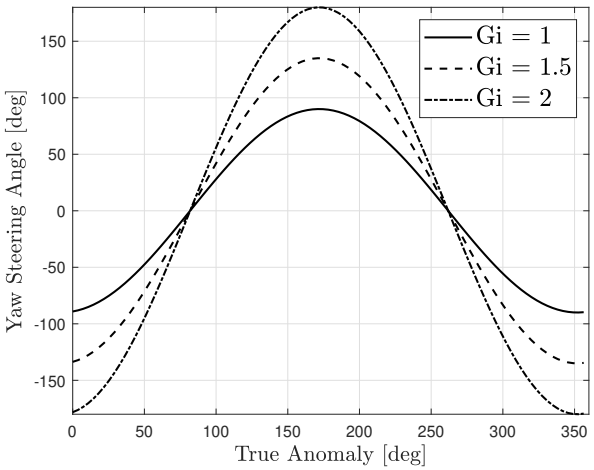

Figure 3.2: Yaw Angle

From Figure 3.1 and 3.2 it is possible to appreciate how the steering angles increase as the control gains give more importance to the variation of eccentricity and inclination with respect to the rate of change of semi-major axis (G_a is kept equal to 1).
In the next pages the thrust vectors corresponding to the steering programs of Figure 3.1 and 3.2 are represented along the orbit.

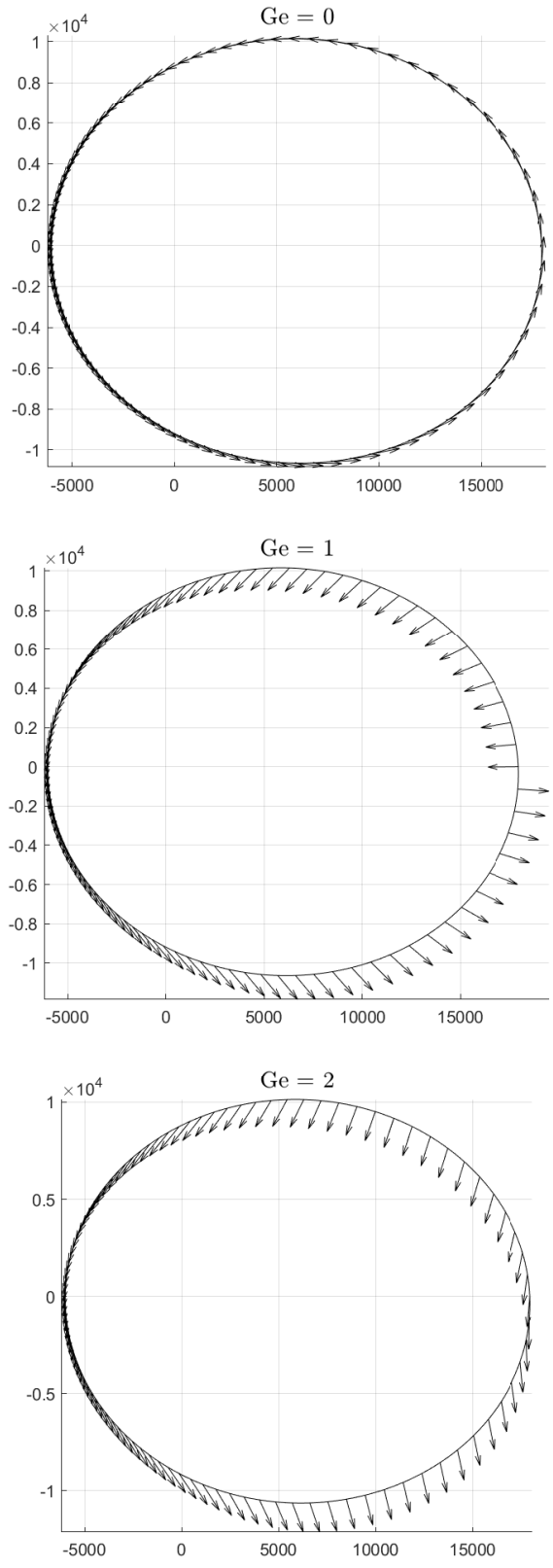

Figure 3.3: Thrust vector evolution for different eccentricity gains

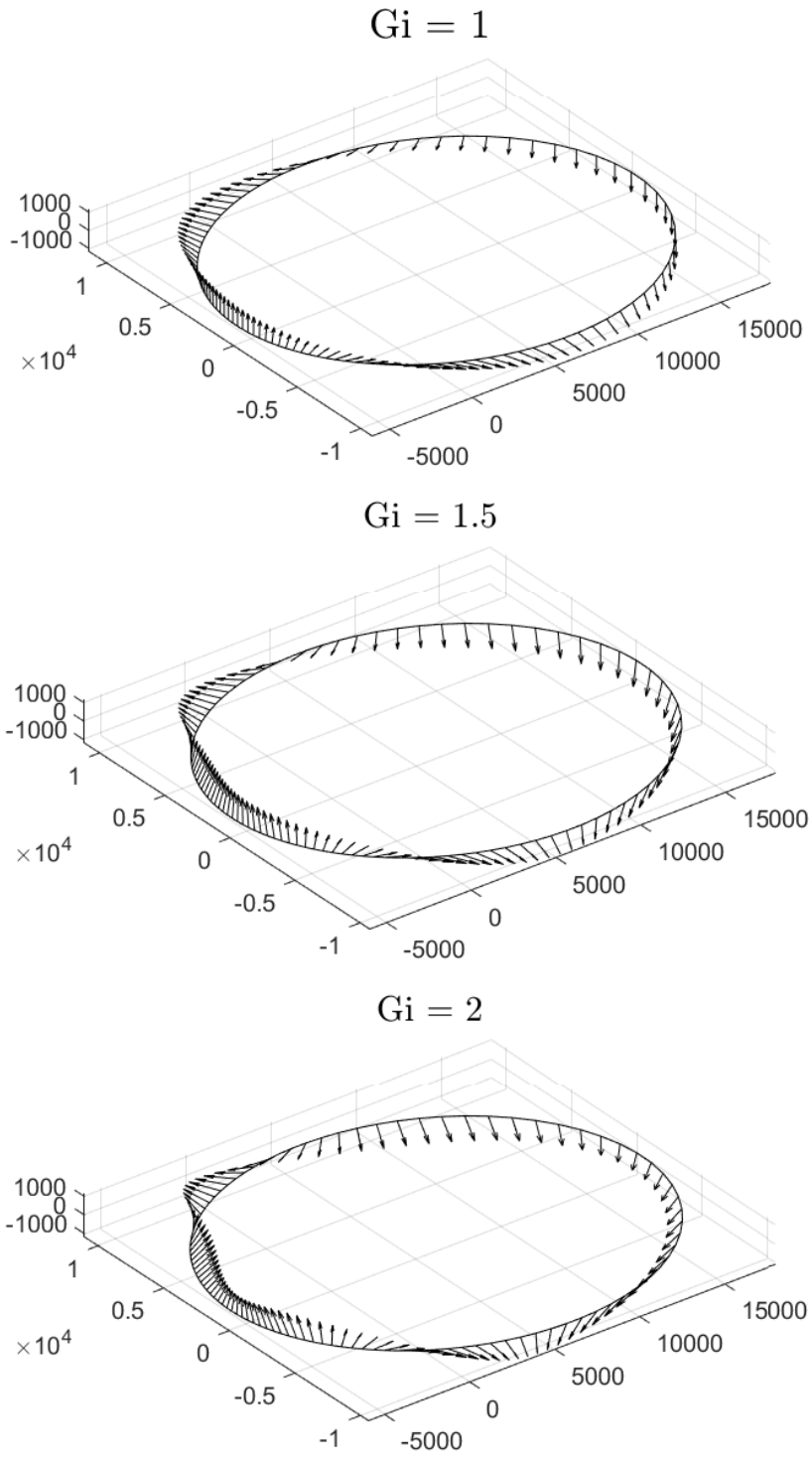

Figure 3.4: Thrust vector evolution for different inclination gains

The time histories of the gains are obtained by linear interpolation through a discrete set of n nodes, equally spaced in time.
Hence, the nodal values of the gains, together with the time of flight, become the optimization variables of the NLP problem. Note that only the relative scale of these gains is important, therefore $G_a(t)$ is set to unity for all time, to reduce the size of the NLP problem. The resulting vector of optimization variables is the following:

$$\mathbf{x}_{opt} = [G_{e_1}, G_{e_2}, ...G_{e_n}, G_{i_1}, G_{i_2}, ...G_{i_n}, TOF] \tag{3.25}$$

3.3. Shooting Method

The main constraint for the NLP problem at hand is the one on the arrival conditions, in terms of semi-major axis, eccentricity, and inclination.
Indeed, the orbital parameters of the spacecraft at the end of the transfer must coincide with those of the target.
Note that only the first three Keplerian elements (a, e, i) at final time are taken as constraints for the optimization, while the RAAN (Ω), the Argument of Pericentre (ω), and the true anomaly (θ) are considered free parameters.
The reason behind this choice is that the present investigation focuses on targeting almost circular and equatorial orbits (the Martian moons orbits), therefore, the control on Ω and ω is not required.
For what concerns the true anomaly θ, the Orbital Averaging technique do not allow any control on it, as mentioned in section 2.4.
Since the investigated problem deals with Martian moons randez-vous, in principle the control of θ would be strictly required.
However, a possible strategy to arrive at the moons could be to first acquire their semi-major axis, eccentricity, and inclination, and afterward perform a phasing maneuver to randez-vous with the moons.
At each optimization step of the NLP problem a Shooting Method [36] is employed: the trajectory is explicitly integrated from the initial conditions up to the final time (defined by the TOF) with the control law previously derived, and the difference between the final state and the target state is evaluated, as well as the objective function.

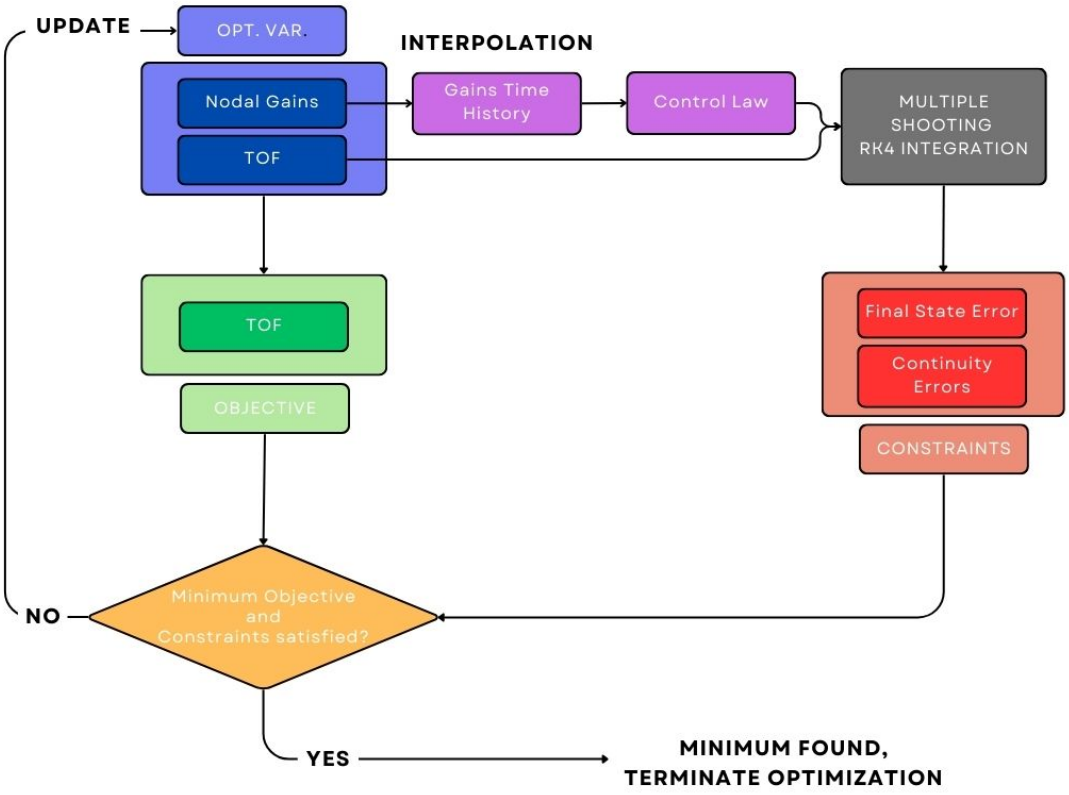

Figure 3.5: Optimization Logic

As mentioned in section 2.4 the integration scheme adopted in this work is the single-step RK4, whose k-th iteration [25] for the computation of the averaged state $\bar{\mathbf{x}}_k$ is reported in Equation 3.26. The terms $\mathbf{u}_{k-1}$ and $\mathbf{u}_k$ represent the control profile at the boundaries of the integration step (of length h), while $\mathbf{w}$ is the control profile at its midpoint. The control profiles are the control time histories along one revolution (from $L = 0$ to $L = 2\pi$), computed with the control gains at the integration time nodes: $G(t_{k-1})$ and $G(t_k)$.

$$\begin{aligned}
\bar{\mathbf{x}}_k^{(1)} &= \bar{\mathbf{x}}_{k-1} + \frac{1}{2}h\mathbf{f}(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \\
\bar{\mathbf{x}}_k^{(2)} &= \bar{\mathbf{x}}_{k-1} + \frac{1}{2}h\mathbf{f}(\bar{\mathbf{x}}_k^{(1)}, \mathbf{w}_k) \\
\bar{\mathbf{x}}_k^{(3)} &= \bar{\mathbf{x}}_{k-1} + h\mathbf{f}(\bar{\mathbf{x}}_k^{(2)}, \mathbf{w}_k) \\
\bar{\mathbf{x}}_k &= \bar{\mathbf{x}}_{k-1} + \frac{1}{6}h\left[\mathbf{f}(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) + 2\mathbf{f}(\bar{\mathbf{x}}_k^{(1)}, \mathbf{w}_k) + 2\mathbf{f}(\bar{\mathbf{x}}_k^{(2)}, \mathbf{w}_k) + \mathbf{f}(\bar{\mathbf{x}}_k^{3}, \mathbf{u}_k)\right]
\end{aligned} \tag{3.26}$$

This integration scheme was coupled with the Orbital Averaging technique to reduce the computational effort: the states used in the integration are averaged and each time that the dynamics $\mathbf{f}$ is evaluated in the RK4 scheme, the formulation expressed in equations 2.16 and 2.17 is used to compute the state derivatives.

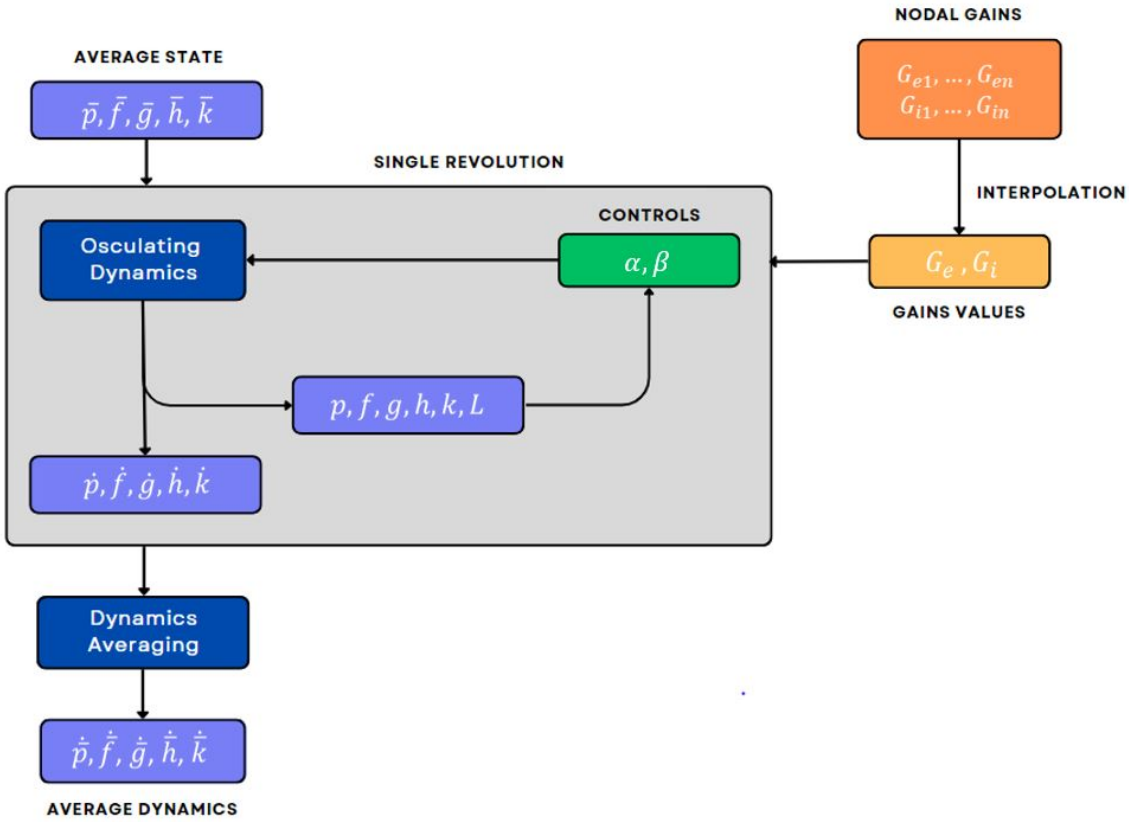

Figure 3.6: Integration Logic

The average state $\bar{x}_k = [\bar{p}, \bar{f}, \bar{g}, \bar{h}, \bar{k}]$, is used to compute the osculating state derivatives $(\dot{p}, \dot{f}, \dot{g}, \dot{h}, \dot{k})$ along one single revolution. To do this the osculating dynamics 2.10 is evaluated for discrete values of true longitude L along the orbit, keeping fixed the average elements. This is what occurs at the numerator of the integral in Equation 2.16.
In the evaluation of the osculating dynamics the controls are computed through the feedback extremal laws derived in the previous section (from the knowledge of current orbital state) and are fed into the osculating dynamics itself.
This feedback mechanism for the control law enables the a priori knowledge of the control, which was required in the formulation of the Orbital Averaging technique described in section 2.4. Then, the derivatives of the osculating orbital elements are averaged (through the integral action) to obtain the average dynamics.
To improve the algorithm's robustness in convergence, a Multiple-Shooting [37] scheme has been implemented as well. With this approach the trajectory is subdivided into segments and the integration is carried out along each segment separately.
The continuity between the segments is then imposed as a further constraint.
The Multiple-Shooting scheme allows to improve the convergence performances of the

algorithm at the cost of an increased computational effort, as explained in section 1.3. Indeed, additional optimization variables must be included, that is, the states at the boundaries of the segments, resulting in the following vector of optimization variables.

$$\mathbf{x}_{opt} = [G_{e_1}, G_{e_2}, ...G_{e_n}, G_{i_1}, G_{i_2}, ...G_{i_n}, \bar{\mathbf{x}}_{ms}(t_1), \bar{\mathbf{x}}_{ms}(t_2), ..., \bar{\mathbf{x}}_{ms}(t_M), TOF] \tag{3.27}$$

It is worth to remark the fact that, in order to alleviate the solver effort, the optimization variables were scaled in such a way to have an optimization vector composed of elements with comparable dimensions. This is particularly important in the multiple shooting version of the algorithm, where the states' position coordinates are of some order larger than the other variables.
Note also that all the optimizations presented in this work were performed starting from a guess obtained by manually tuning the gains as to obtain a guess trajectory satisfying the final conditions of the transfer. In this way the optimization was initiated with a feasible guess resulting in a faster and more robust process.

3.4. Minimum Propellant Problem

The algorithm described so far is a slight reformulation of that proposed by Kluever and Oleson [6]. However, the two authors limited its applicability to the minimum time of flight problem, by defining the objective function of the NLP problem as the TOF itself, to be minimized. Besides the reformulation of the algorithm, this work aims to further extend its applicability to the minimum propellent problem.
In order to do that, a new version of the algorithm was developed, which includes a burning-coasting structure for the control.
It has been observed empirically [29] that the most efficient way to perform a maneuver is to apply the thrust at the absides of the orbit, leaving a coasting phase in between the thrusting arcs.
This concept was leveraged to produce a bang-bang burn structure for the control, to be optimized in terms of the length of the thrusting arcs. The periapsis- and apoapsis-centered burn structure proposed by Gao and Li [29] has been employed in the present work, and coupled with the control laws previously discussed.

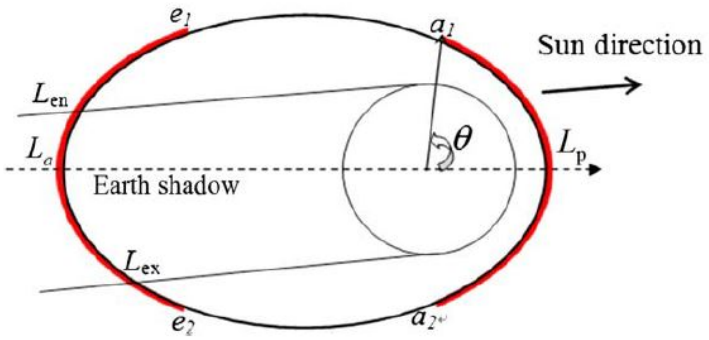

Figure 3.7: Burning-Coasting structure

The angular bounds which delimit the thrusting arcs are defined as

$$\begin{aligned} a_1 &= L_p + w_p\pi \\ a_2 &= L_p - w_p\pi \\ e_1 &= L_a + w_a\pi \\ e_2 &= L_a - w_a\pi \end{aligned} \tag{3.28}$$

where the periapsis L_p and the apoapsis L_a angles in true longitude are

$$\begin{aligned} L_p &= \omega + \Omega \\ L_a &= L_p + \pi \end{aligned} \tag{3.29}$$

The parameters w_a and w_p introduced in the equations 3.28, are weights bounded from 0 to 1 which define the length of the burning arcs.
If $w_a + w_p = 1$ there are no active coasting arcs (out of shadow), and therefore, the following constraint must be included:

$$w_p + w_a \leq 1 \tag{3.30}$$

In this extended version of the algorithm the weighting parameters w_a and w_p are included in the pool of the optimization variables, leading to a higher computational load.
In a similar fashion to what was done for the control parameters G_a and G_e, also in this case the time histories of $w_a(t)$ and $w_p(t)$ were obtained by linear interpolation through a discrete set of nodes, equally spaced in time.
At each optimization step the algorithm integrates the trajectory by solving the averaging integrals with the limits given by the angular bounds of the thrusting arcs, which in turn are defined by w_a and w_p. Furthermore, if the shadowing condition is included, the burn arcs may be discontinuous. In this case the time rate of each mean equinoctial element due to thrust should be computed using piecewise integration. Referring to the scheme of Figure 3.7:

$$\begin{aligned} \frac{d\bar{\mathbf{x}}_l}{dt} = \frac{1}{2\pi}(1-\bar{f}^2-\bar{g}^2)^{3/2}\cdot \Bigg[& \int_{L_{ex}}^{e_2} \frac{f_l^T(\bar{\mathbf{x}}, L, \boldsymbol{u})}{(1+\bar{f}\cos L + \bar{g}\sin L)^2} dL + \\ & \int_{a_2}^{a_1} \frac{f_l^T(\bar{\mathbf{x}}, L, \boldsymbol{u})}{(1+\bar{f}\cos L + \bar{g}\sin L)^2} dL + \\ & \int_{e_1}^{L_{en}} \frac{f_l^T(\bar{\mathbf{x}}, L, \boldsymbol{u})}{(1+\bar{f}\cos L + \bar{g}\sin L)^2} dL + \\ & \int_{0}^{2\pi} \frac{f_l^p(\bar{\mathbf{x}}, L, \boldsymbol{a}_{pert})}{(1+\bar{f}\cos L + \bar{g}\sin L)^2} dL \Bigg] \end{aligned} \tag{3.31}$$

To do this, the sequential order of the angular positions $L_{ex}, L_{en}, a_1, a_2, e_1, e_2$ should be determined in advance. For this reason a dedicated function responsible to identify the sequence of these angular positions was included in the algorithm.
To solve the minimum propellant problem the cost function to be extremized by the optimization can be either the propellant mass (to be minimized) or the final mass of the spacecraft (to be maximized).

At each optimization step the evaluation of this quantity would require an additional integration of the whole trajectory, leading to a significant growth of the computational load. In order to avoid this criticality, a different cost function was defined as the sum of the parameters w_a and w_p at the trajectory nodes (which are optimization variables):

$$J = \sum_{i=1}^{n} w_{a_i} + \sum_{i=1}^{n} w_{p_i} \tag{3.32}$$

where n is the number of nodes at which the thrust weights are defined for the interpolation. Such a cost function clearly does not provide directly an information on the consumed propellant mass, but it is strictly connected with it.
Indeed, the amount of propellant consumed depends on the length of the burning arcs. Therefore, the minimization of the overall length of such arcs (through the minimization of the thrust weights sum), implies automatically the minimization of the consumed propellant mass.

3.5. Attitude Constraint

In order to further stress the algorithm and investigate its capabilities, another strong constraint was included. So far the described algorithm treated the spacecraft as a point-mass with only three degrees of freedom. Although this is a reasonable assumption for a preliminary design, it constitutes a limiting factor for more accurate evaluations.
Indeed, when dealing with SEP, spacecraft attitude cannot be neglected, as it determines the solar light incidence on the Solar Arrays and in turn, the amount of thrust produced. Considering the spacecraft as a point-mass and assuming constant maximum thrust, is equivalent to assuming that the Solar Arrays are always oriented optimally towards the Sun. This hypothesis could be partially reasonable if the spacecraft is equipped with orientable Solar Arrays, but it doesn't hold when considering body-fixed Solar Panels.
Since the main purpose of the present study is to investigate the applicability of such trajectory design to small-size satellites, which typically are equipped with fixed panels, a new version of the algorithm was implemented to test its behavior under a variable thrust depending on the spacecraft illumination conditions.
A shape for the spacecraft was first selected, which is that of the interplanetary CubeSat LICIACube [16] represented in Figure 3.8.

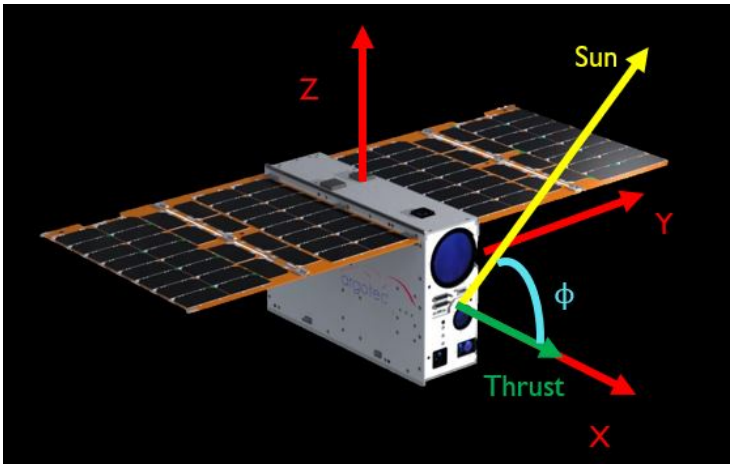

Figure 3.8: LICIACube body-fixed reference frame

As a first approximation, the thrust was scaled through a simple cosine law of the Sun incidence angle ϕ.

$$T = T_{max} \cos(\pi/2 - \phi) \tag{3.33}$$

The spacecraft was assumed to have no constraints on the rotation around the roll axis

(x axis), which means that Solar Panels were assumed to be always facing the Sun. Under this assumption the incidence angle ϕ can be computed as the angle between the Sun direction vector and the thrust vector.

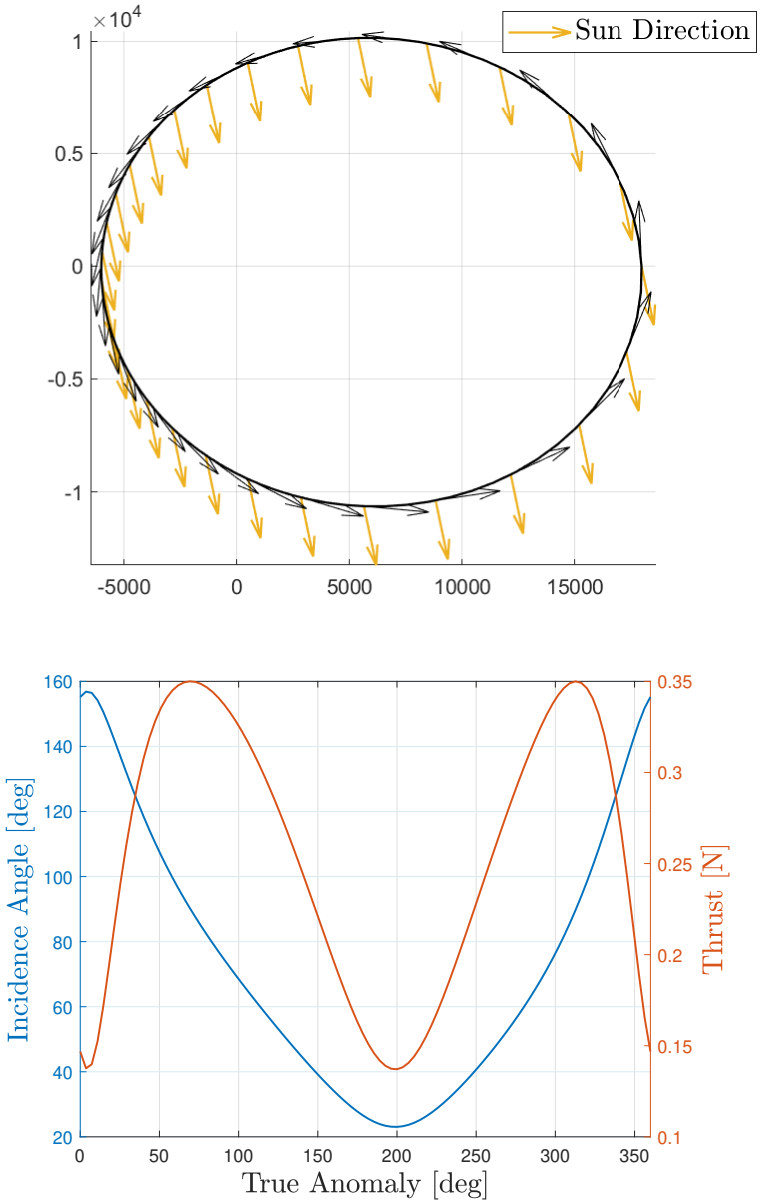

Figure 3.9: Thrust oscillation due to variable Sun incidence angle

From Figure 3.9 it is possible to see how the sunlight incidence angle oscillates along the orbit. This produces an oscillation in the amount of thrust generated as well, with two peaks of maximum thrust in correspondence of those orbit locations where the Sun direction vector is almost perpendicular to the thrust vector (and therefore to the Solar Panels).

4 | Validation and Testing

In order to validate the algorithm effectively, a test-bench scenario was selected, and the algorithm was run on it to compare the results against the literature.
The validation scenario is that of a GTO-GEO transfer, as this test case was deeply investigated in the past. Particularly, the work of Jimenez-Lluva and Root [8] has been taken as reference, since the method proposed by the authors relies on a heuristic algorithm, leading to a global optimization. The spacecraft properties and the initial and final orbital parameters are reported in Table 4.2 and 4.1.

	a [km]	**e [-]**	**i [deg]**
Initial GTO	24505.9	0.725	7.0
Target GEO	42165.0	0.1e-12	0.1e-12

Table 4.1: Orbital Parameters

Thrust	**Initial Mass**	**Specific Impulse**	**T/W**
350 mN	2000 kg	2000 s	1.75e-4

Table 4.2: Spacecraft Parameters

4.1. Algorithm Validation

The algorithm was first validated assuming a spherical gravitational field and then the perturbing J2 effect due to Earth oblateness was included in the dynamical model.
Note that the work taken as reference doesn't include the Eclipse condition, whereas the here presented algorithm provides the possibility to include such a feature in the optimization, as described in section 2.4. The algorithm was therefore also stressed by running the test scenario under J2 perturbation and Eclipse condition.

The optimization results are reported in Table 4.3 and 4.4, together with the reference values.

Minimum TOF Problem					
Scenario	TOF [days]	m_p [kg]	a_f [km]	$e_f[-]$	i_f [deg]
Calculated Unperturbed	139.19	214.53	42164.99	1.6e-9	1.3e-11
Reference Unperturbed	137.45	211.91	42164.65	5.53e-4	7.41e-5
Calculated J2	139.32	214.74	42164.99	1.1e-4	1.4e-4
Reference J2	137.71	212.32	42164.68	5.28e-4	2.59e-4
Calculated J2 and Eclipse	142.35	215.47	42164.95	6.7e-6	1.7e-6

Table 4.3: Minimum TOF results comparison with Jimenez-LLuva and Root results [8]

Minimum Propellent Problem					
Scenario	TOF [days]	m_p [kg]	a_f [km]	$e_f[-]$	i_f [deg]
Calculated Unperturbed	150	195.65	42164.99	1.1e-12	3.77e-9
Reference Unperturbed	150	194.75*	N/A	N/A	N/A
Calculated J2	150	200.83	42165.00	4.89e-7	2.44e-4
Reference J2	N/A	N/A	N/A	N/A	N/A
Calculated J2 and Eclipse	150	206.24	42166.30	8.35e-4	1.94e-4

Table 4.4: Minimum Propellant results comparison with Jimenez-LLuva and Root results [8]

Even if the found minimums are local, the algorithm presents good behavior, providing a solution quite close to the global minimum of the reference.
Unfortunately the work taken as a reference only provide the results of the perturbed scenario (for the minimum propellant problem) for an algorithm version featuring longitude-targeting. For sake of consistency such a result was not used to carry out the validation. However, since the inclination of both the initial and final orbits is quite low, the perturbing effect due to Earth oblateness does not affect significantly the spacecraft motion. Indeed, it is possible to see in Table 4.3 that the results of the unperturbed and perturbed scenarios are quite similar. Also note that in the minimum propellant problem, the value of the propellant mass marked with the asterisk was not directly provided by the authors, but rather derived from their graphs. Let's now have a close look at the scenario including both the J2 perturbation effect and the eclipse condition.

4.1.1. Minimum Time of Flight Problem

In the next images representing the transfer, the color gradient expresses the time evolution of the orbit.

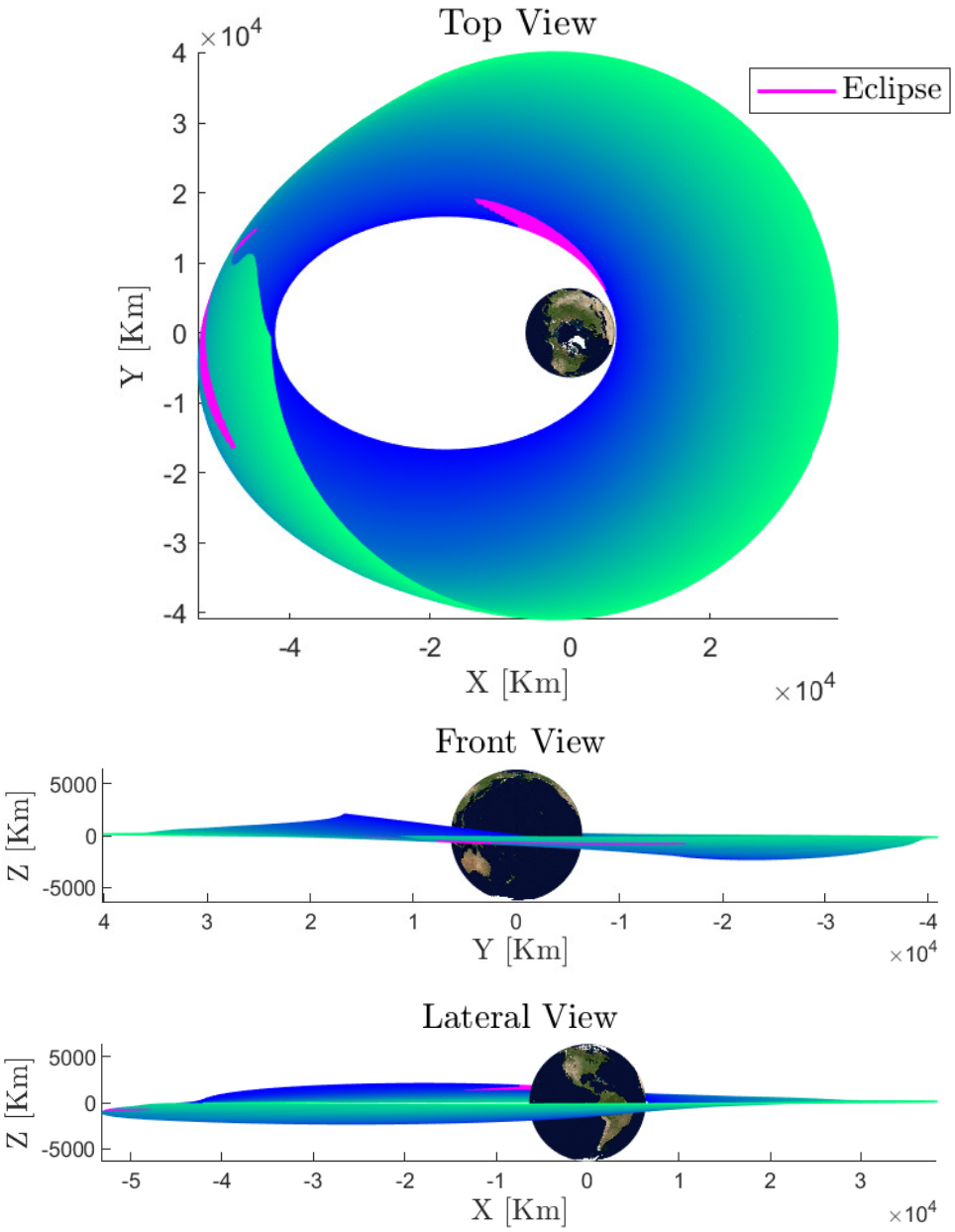

Figure 4.1: Minimum Time of Flight GTO-GEO Transfer

It is interesting to analyze the optimal steering program for this transfer. In Figure 4.2 the in-plane and out-of-plane thrust angles profile is reported and the color gradient allows us to have insights into the evolution of these profiles along the transfer, from the first to the last revolution.

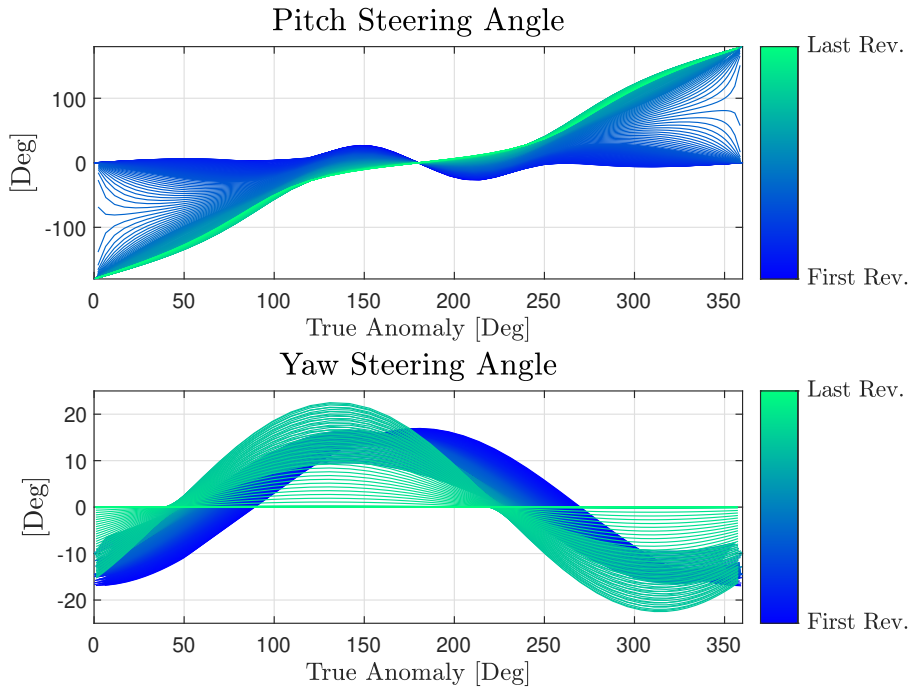

Figure 4.2: Pitch and Yaw angles profile

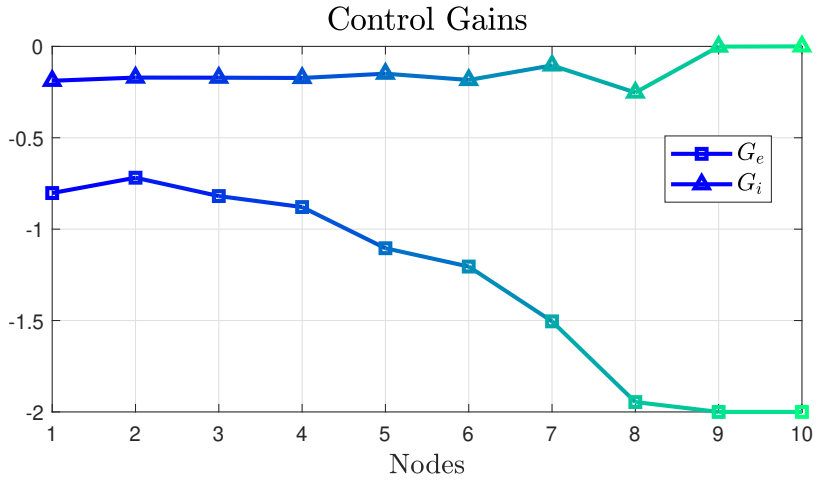

Figure 4.3: Control Gains time history

From Figure 4.2 it is possible to appreciate how the algorithm looks for an efficient control law: during the early stages of the transfer, when the spacecraft is close to the attractor, the optimal control law features low amplitude steering angles, as to keep the thrust aligned with the velocity vector as much as possible, in order to increase orbital energy and semi-major axis as fast as possible.
Then, in the second phase of the transfer, the amplitude of the steering angles grows, as to correct for eccentricity and inclination when the spacecraft is far from Earth and

experiences a lower gravitational attraction. From the plot of the yaw steering angle it is possible to observe a slight drift of the angle profile towards the lower values of the true anomaly. The reason for this behavior lies in the structure of the optimal control law for the out-of-plane angle (Eq. 3.23). In particular the responsible for this drift is the argument of pericentre which appears as argument of the cosine law. The J2 perturbation has the effect of inducing a precession motion in the argument of pericentre which results in the drift highlighted in Figure 4.2.
The control law is driven by the gains reported in Figure 4.3. Note that the control gains assume negative values because the acquisition of the final GEO orbit requires a reduction of eccentricity and inclination. From the time history of G_i it is clear that the rate of change of the inclination is almost constant along the transfer, except for the last revolutions, when the target inclination has been reached and therefore no out-of-plane component of the thrust is required.

4.1.2. Minimum Propellant Problem

For what concerns the minimization of the propellant consumed, the transfer was solved by assuming a fixed time of flight of 150 days, as it was done in the reference [8].
Therefore, in this case the time of flight was not anymore included in the optimization variables.

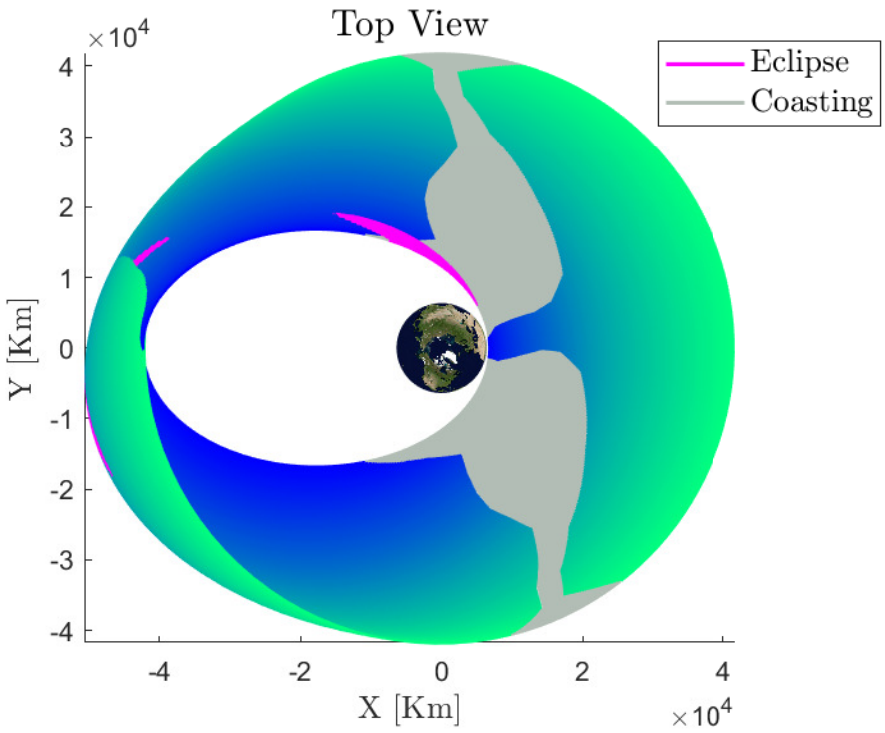

Figure 4.4: Minimum Propellant GTO-GEO Transfer

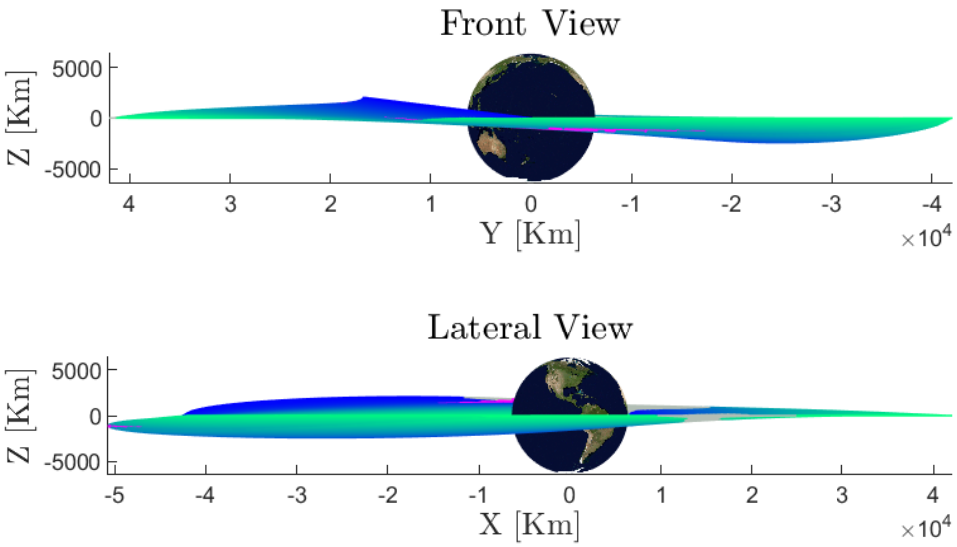

Figure 4.5: Minimum Propellant GTO-GEO Transfer

In the minimum propellant problem the control law presents a quite similar behavior to that of the minimum time of flight problem.
Naturally, in this case the transfer time will be larger, leading to a higher number of revolutions.

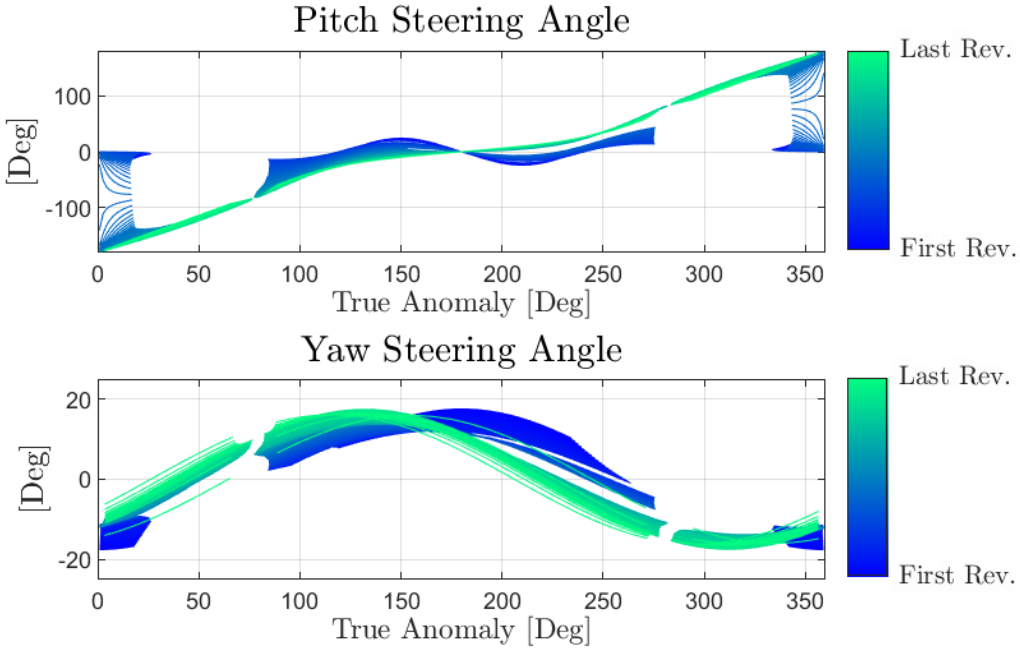

Figure 4.6: Pitch and Yaw angles profile

In Figure 4.7 the time history of the weighting parameters w_a and w_p is depicted.
It is possible to appreciate how the algorithm looks for a bang-bang control law that first tends to increase the pericentre altitude by prioritizing the thrust at the apocentre.
Then, in the second phase of the transfer, w_p increases, as the control law tries to circularize the orbit by broadening the thrusting arcs at the orbit pericentre.

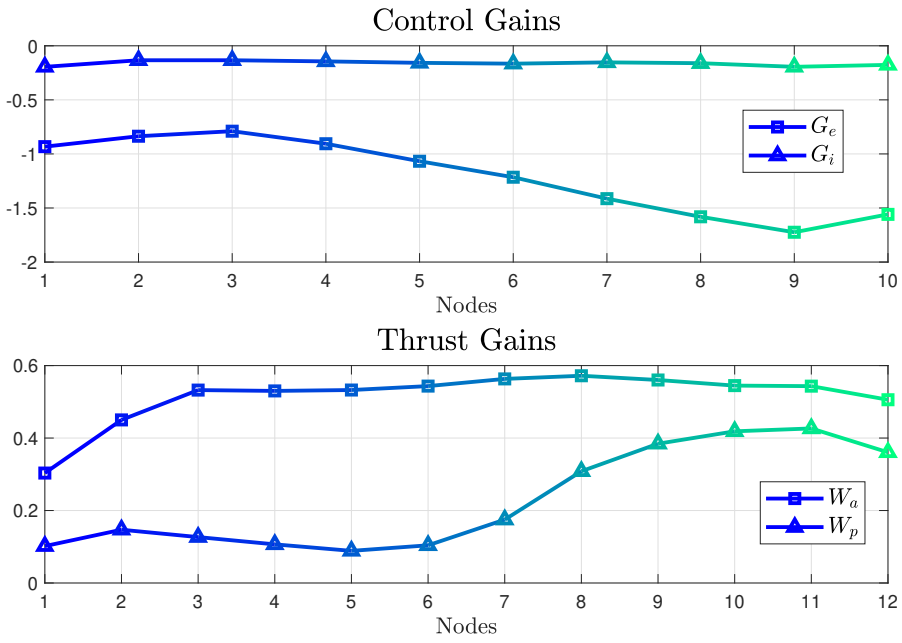

Figure 4.7: Control Gains and Thrust Gains time history

4.2. Attitude Constraint Results

The modified algorithm including the attitude constraint discussed in section 3.5 was only tested on the minimum time of flight scenario, and the results are shown in Table 4.5.

	Constrained Attitude	**Free Attitude**
Time of Flight	183.93 days	142.35 days
Propellant Mass	220.57 kg	215.47 kg

Table 4.5: Constrained vs Free attitude comparison

In the case of the constrained attitude, the time of flight increases significantly by almost the 30% with respect to the free attitude case.

Note that the results of the exploration scenario, illustrated in the following section of the report, were obtained by running the optimization algorithm without including this constraint on the attitude, because of convergence issues.

4.3. Algorithm Performances

For both the minimum time of flight and the minimum propellant consumption problem the multiple shooting scheme explained in section 3.3 was employed, as to improve the convergence properties of the algorithm. Particularly, the trajectory was discretized in 5 segments, each of which was integrated individually through 35 Runge-Kutta 4 integration steps. As explained previously the continuity between the multiple shooting segments was forced by imposing continuity constraints. In Table 4.6 and 4.7 the defect relative errors between the states at segment boundaries are reported.
The algorithm shows very good convergence properties in terms of constraints satisfaction. Only the semimajor-axis error is slightly higher with respect to the others, but still it is totally acceptable. Furthermore, although the heavy optimization problem, the overall computational load is not impractical and the algorithm allows to carry out optimizations in a relatively limited time. The algorithm's performances are reported in Table 4.8 and 4.9 in terms of CPU time, number of iterations, and number of function evaluations.
All the simulations presented in this work were done with a Intel(R) Core(TM) i7-8750H processor @2.20 GHz running on a single core, while the language used is MATLAB 2021b.

Minimum TOF Problem				
Relative Error	$\lvert\varepsilon_a\rvert$	$\lvert\varepsilon_e\rvert$	$\lvert\varepsilon_i\rvert$	$\lvert\varepsilon_m\rvert$
Node 2	1e-4	7e-10	6e-10	2e-8
Node 3	8e-5	5e-10	2e-10	1e-8
Node 4	8e-5	3e-9	5e-10	1e-8
Node 5	2e-6	8e-10	2e-10	1e-8

Table 4.6: Continuity violation at segments boundaries

Minimum Propellant Problem				
Relative Error	$\lvert\varepsilon_a\rvert$	$\lvert\varepsilon_e\rvert$	$\lvert\varepsilon_i\rvert$	$\lvert\varepsilon_m\rvert$
Node 2	2e-5	1e-10	4e-11	5e-9
Node 3	3e-7	2e-11	3e-12	4e-9
Node 4	6e-5	1e-9	5e-10	4e-9
Node 5	2e-5	7e-10	3e-10	3e-9

Table 4.7: Continuity violation at segments boundaries

Minimum TOF Problem			
Scenario	CPU Time	Iterations	F-Count
Calculated Unperturbed	210 s	235	5963
Calculated J2	2766 s	566	14799
Calculated J2 and Eclipse	1302 s	66	3647

Table 4.8: Algorithm performance on Minimum TOF problem

Minimum Propellent Problem			
Scenario	CPU Time	Iterations	F-Count
Calculated Unperturbed	5941 s	539	44297
Calculated J2	4634 s	361	11456
Calculated J2 and Eclipse	810 s	25	2136

Table 4.9: Algorithm performance on Minimum Propellant problem

5 | Mars Exploration Scenario

As mentioned in the introduction, the main goal of the present work is to investigate the possibility of exploiting low-thrust trajectory design to propose a solution for exploring the Martian system with small-size spacecraft.
Given the small masses of the two Martian moons, the design of a low-thrust transfer for capture into a science orbit in the vicinity of either Deimos or Phobos is challenging. Indeed, in contrast to other multi-moon planetary systems like that of Jupiter or Saturn, the gravitational influence of the Martian moons is so small that they do not significantly influence the path of a spacecraft.
An important consequence of this "weak" gravitational environment is that it is not possible to exploit invariant manifold trajectories emanating from the orbits near the moons to perform low energy transfers within the Martian system.
To overcome this problem Canales, Gupta, et al. [2] proposed an alternative exploration framework leveraging Mars-Deimos resonant orbits. Their work was taken as a starting point for the present investigation and further extended.

5.1. Science Orbits

The authors propose a mission scenario that leverages Mars-Deimos resonant orbits for exploring both moons with reduced propellant cost.
The idea is to exploit a Mars-Deimos resonant orbit which allows for multiple Deimos flybys and, at the same time, is designed in such a way to intersect invariant manifolds arriving into Phobos capture orbit, which could be either a Lyapunov or Halo orbit around the Mars-Phobos L2 libration point. Note that while the authors computed the resonant orbits including the gravitational attraction of the Martian moons, in the present investigation their gravitational influence was neglected.
This approximation can be considered quite reasonable, due to the poor mass of Phobos and Deimos.

Among the resonant orbits proposed by Canles, the one selected in this study is the 5:4 Deimos resonant orbit which is depicted in Figure 5.1.
This kind of orbit is such that the spacecraft travels 5 revolutions around Mars in the same time that Deimos takes to travel 4 revolutions.
According to the analysis of Canales, this orbit offers good flyby conditions for observing Deimos, and at the same time allows to access Phobos L2 orbits with a limited propellant expense.

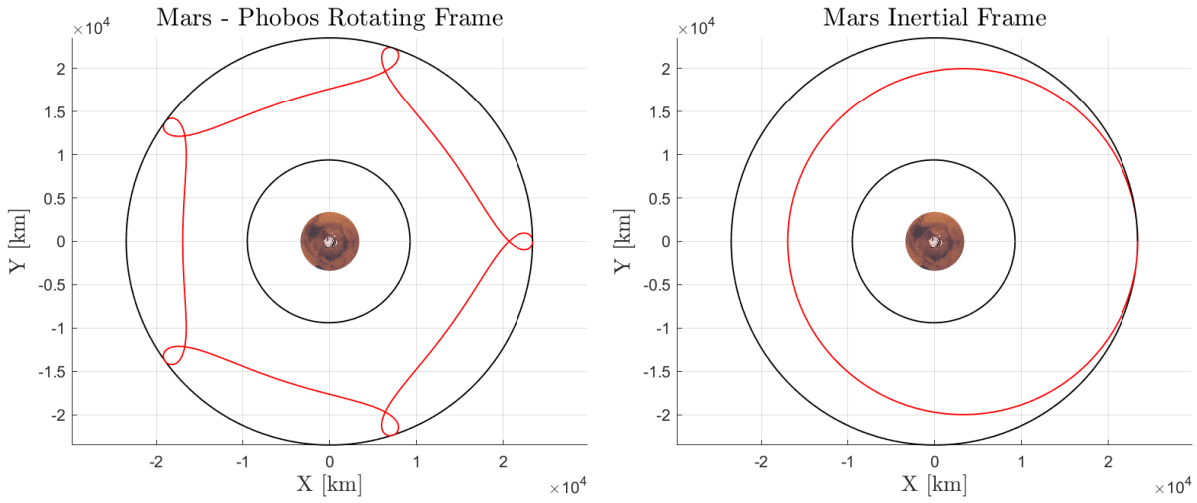

Figure 5.1: Mars-Deimos 5:4 resonant orbit

The authors then investigated the transfer between this resonant orbit and the Phobos science orbit, either with impulsive and continuous thrust.
In the latter case the OCP was solved through an algorithm combining the Q-law (to retrieve the guess) with a Direct Collocation scheme (to produce the optimal transfer). The choice of the Direct Collocation scheme was justified by the relatively high thrust-over-weight ratio assumed, which was in the order of 10^{-4}, leading to a time of flight of approximately 25 days.
However, for the purpose of this study, such a T/W ratio is too high, and hardly can be achieved with a CubeSat. A more realistic T/W for a small-size spacecraft is in the order of 10^{-5}, leading to much more large times of flight. To handle such long transfers, the control parametrization algorithm introduced in Chapter 3 was employed.
The mission scenario here proposed is composed of two transfers: the first one (Transfer A) consists in the acquisition of the Deimos resonant orbit starting from a possible release orbit around Mars. The acquisition of the resonant orbit was not investigated by Canales, even if it constitutes a critical element of the overall mission scenario, as will be explained later. The second transfer analyzed (Transfer B) is instead the one linking the Deimos resonant orbit with Phobos's science orbit.

5.2. Release Orbit

As mentioned in Chapter 1 an interplanetary CubeSat hardly can reach Mars autonomously. More likely it would need to rely upon a larger mission, and to be released once at the Martian system. For this reason an important aspect to take into consideration in the overall trajectory design, is the acquisition of Deimos resonant orbit starting from any release conditions, which could be either a release at the Mars Sphere of Influence, or on a bounded orbit around the planet.
Typically, spacecraft arriving at Mars perform a Mars Orbit Insertion maneuver (MOI): a high thrust burning which results in the acquisition of a High Elliptical Orbit (HEO). In this work a HEO was considered as the releasing orbit for the CubeSat and taken as the initial condition for the exploration of the Martian system.

Mission	Period	Semi-Major Axis	Eccentricity	Perigee Altitude
Mars Express	48 hr	31877 km	0.8838	314 km
Mars Odyssey	19 hr	17185 km	0.7857	292 km
TGO	98 hr	51512 km	0.9283	300 km
MRO	35 hr	25824 km	0.8522	426 km
MAVEN	35 hr	25824 km	0.8539	382 km
MGS	45 hr	30535 km	0.8808	250 km

Table 5.1: High Elliptical Orbit parameters of past Mars missions

The Keplerian parameters for this orbit were selected on the basis of past Mars missions and are reported in Table 5.2.

Semi-Major Axis	Eccentricity	Inclination	RAAN	Arg. of Pericentre
28229 km	0.8675	15 deg	270 deg	270 deg

Table 5.2: Release Orbit selected parameters

Note that the HEO orbits of the past missions reported in Table 5.1 are characterized by a high inclination, because the target operational orbits were polar.

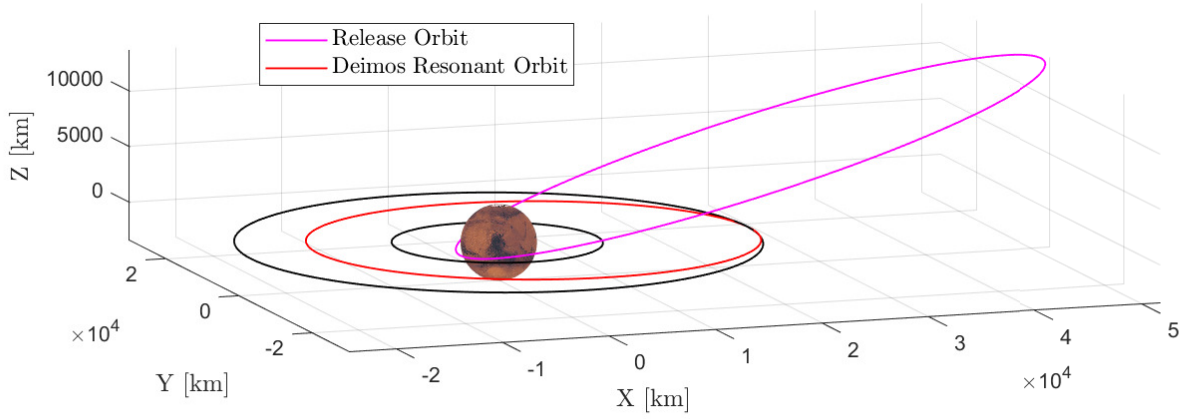

Figure 5.2: Release HEO orbit

However, in this work a milder inclination (15 degrees) has been selected to limit the plane change, as it is an extremely expensive maneuver. The spacecraft parameters assumed in this work are reported in Table 5.3.

Thrust	Initial Mass	Specific Impulse	T/W
1.1 mN	15 Kg	2150 s	7.33e-5

Table 5.3: Spacecraft Parameters

5.3. Transfer A

For Transfer A only the minimum time of flight problem was solved, because the minimization of the propellant consumption would imply a transfer duration that is out of practical interest. The optimal time of flight and propellant consumption that were found by the algorithm for Transfer A are reported in Table 5.4.

Time of Flight	Propellant Mass
155.36 days	0.6983 kg

Table 5.4: Transfer A minimum Time of Flight results

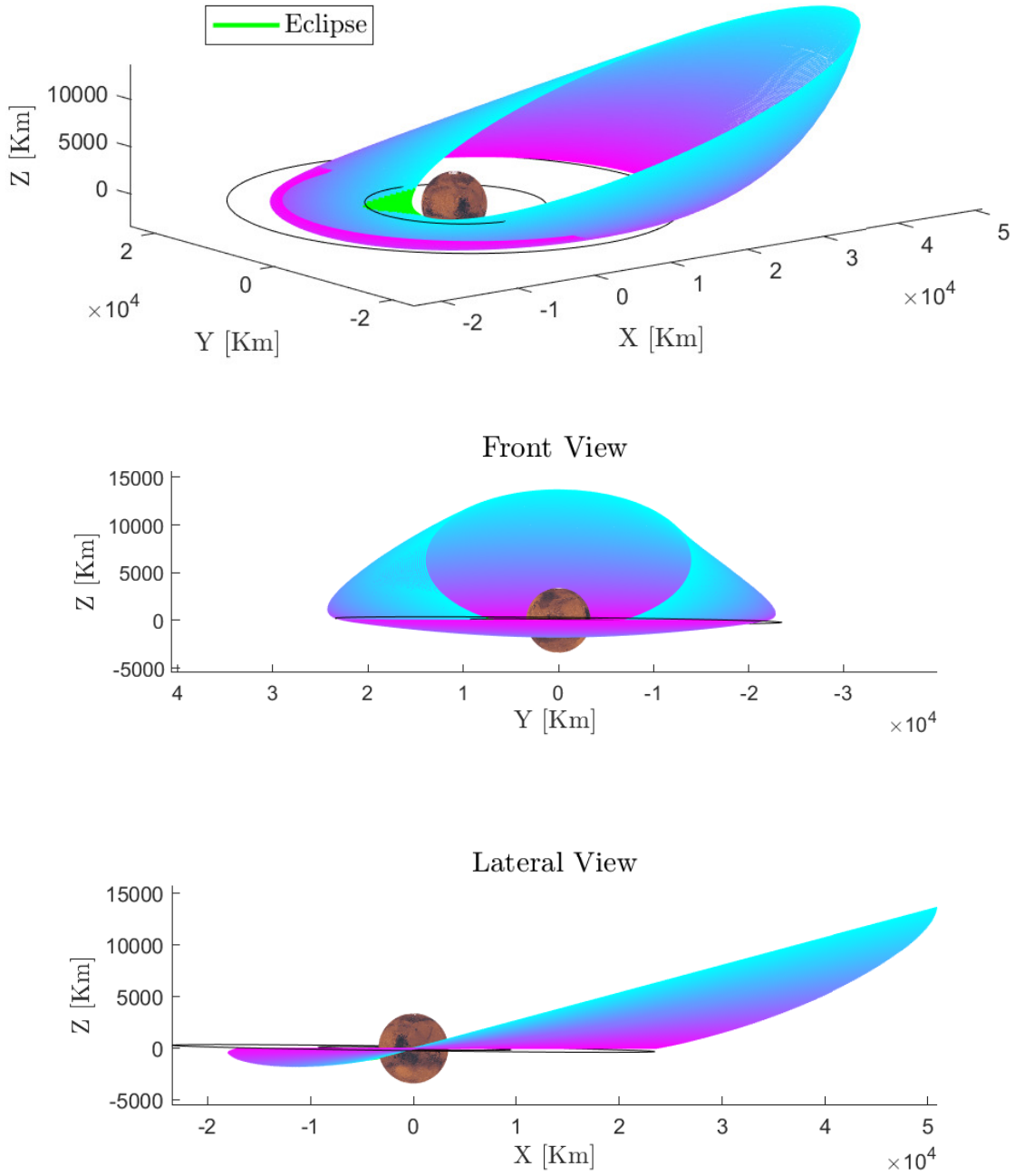

Figure 5.3: Transfer A minimum Time of Flight trajectory

Figure 5.3 depicts the orbit evolution along Transfer A. It is possible to observe that also in this case the change of plane and the orbit circularization are mainly performed in the first part of the transfer, when the spacecraft is far from the planet.

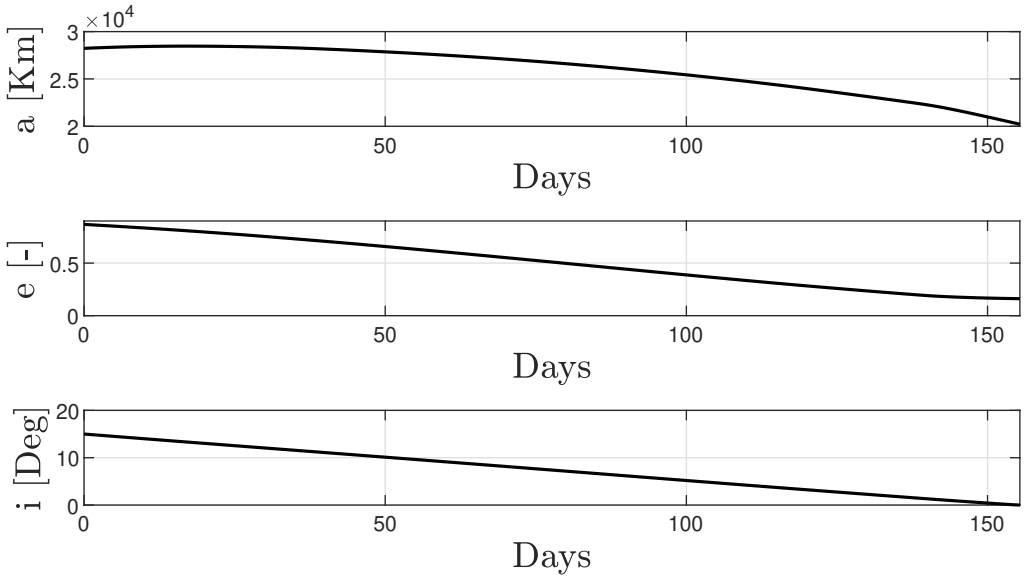

Figure 5.4: Keplerian elements evolution

This behavior is obtained through the gains time history reported in Figure 5.6 that produce the optimal control law depicted in Figure 5.5 which features large pitch and yaw steering angles during the first revolutions, while their amplitude decreases in the last stages of the transfer.

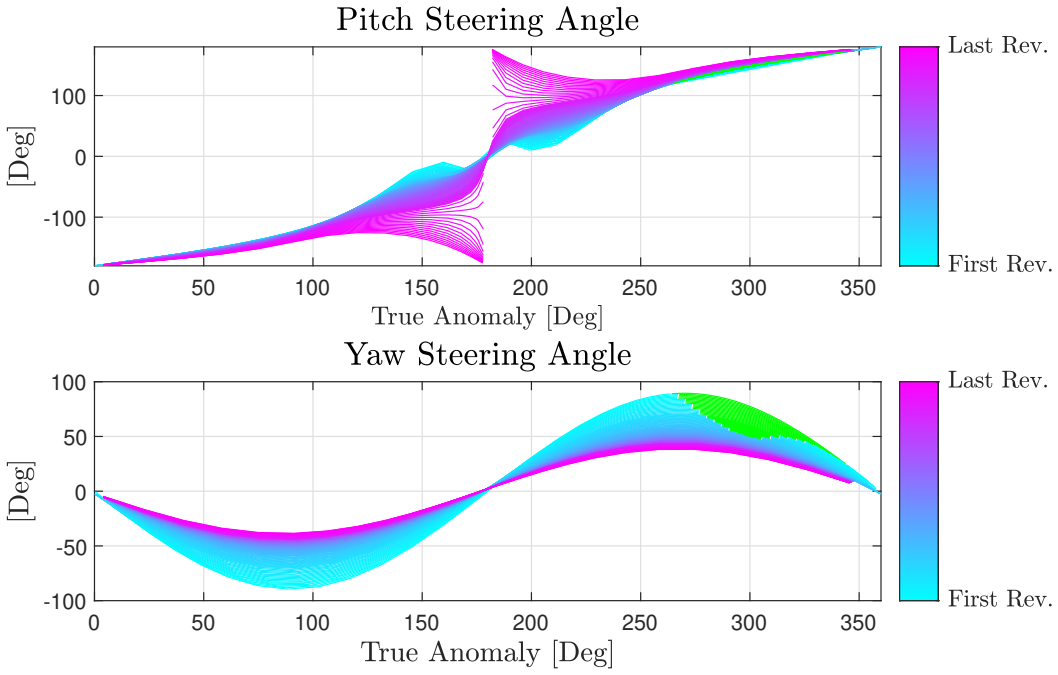

Figure 5.5: Pitch and Yaw angles profile

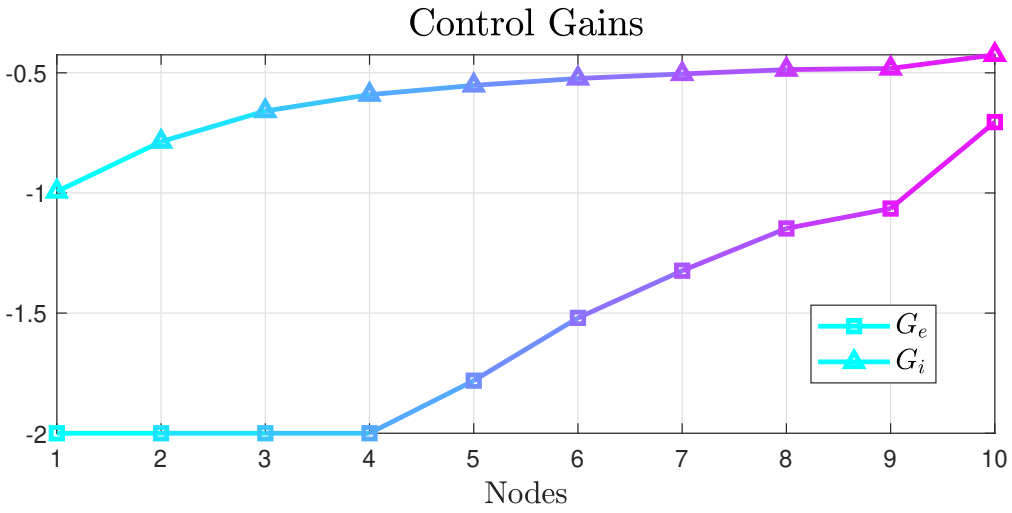

Figure 5.6: Control Gains time history

The reason for this control profile is that plane change and eccentricity variations are more expensive than semi-major axis variation. Therefore the algorithm tends to perform the inclination and the eccentricity corrections far from the planet, where the gravitational attraction is lower.
The reported results present an intrinsic criticality: the pitch angle profile features a discontinuity at the apocentre of the last revolutions of the transfer. In Figure 5.5 this discontinuity appears clearly at the true anomaly of 180 degrees, where the pitch steering angle shows a sharp variation. The criticality associated with this behavior concerns the attitude angles rate of change. The reported control law indeed does not take into account the physical limitation of the attitude control system of the spacecraft.
Note also that in all the scenarios discussed in this section the semi-major axis control gain G_a was set to -1 instead of 1 (as was done in the validation scenario) because these transfers require a lowering of the semi-major axis.

5.4. Transfer B

The second transfer is intended to connect the Deimos resonant orbit with the Phobos L2 science orbit. In this study the complex three-body dynamics in the proximity of Phobos was not analyzed, and Phobos orbit itself was taken as the target condition for the transfer. Although this is an approximation, it is not so far from reality.
Indeed, due to the poor gravitational attraction of Phobos, the orbital parameters of the stable Mars-Phobos L2 Manifold are almost identical to those of Phobos's orbit.
For this transfer either the minimum time of flight and minimum propellant problem were solved, taking as initial mass the residual mass after the first transfer (14.302 kg).

5.4.1. Minimum Time of Flight

The results for the minimization of the time of flight are reported in Table 5.5.

Time of Flight	Propellant Mass
122.35 days	0.5228 kg

Table 5.5: Transfer B minimum Time of Flight results

During this spiral-down transfer the thrust vector has to be pointed in the direction opposite to that of the velocity vector, as to decrease the orbital energy and reduce the semi-major axis. Moreover, some eccentricity adjustments are required to acquire the orbit of Phobos, which is almost perfectly circular. Once again the algorithm tends to provide a control law acting on the eccentricity when the spacecraft is far from the attractor (in the early stages of the transfer) and then tends to align the thrust direction to the velocity direction.

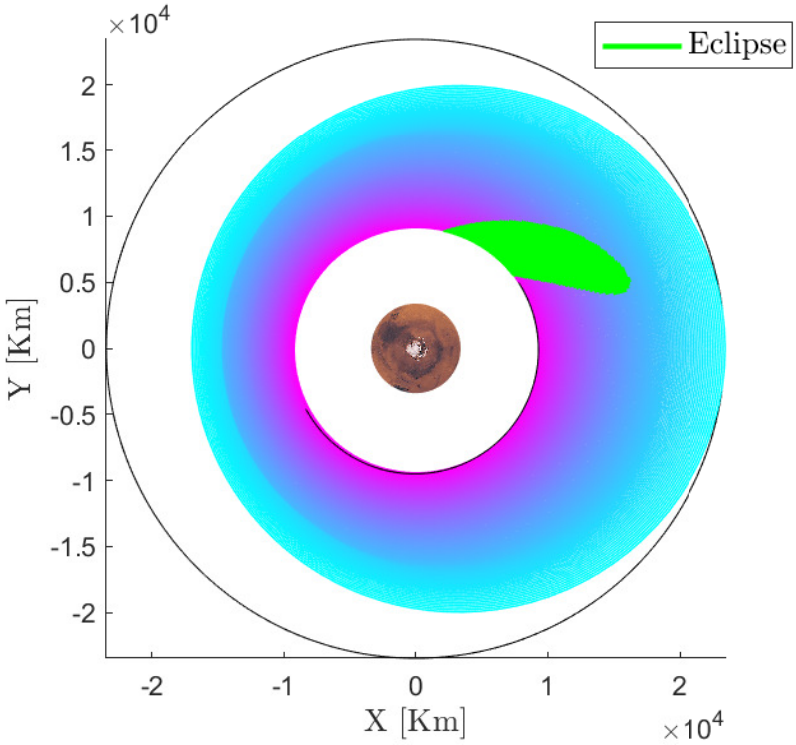

Figure 5.7: Transfer B minimum Time of Flight trajectory

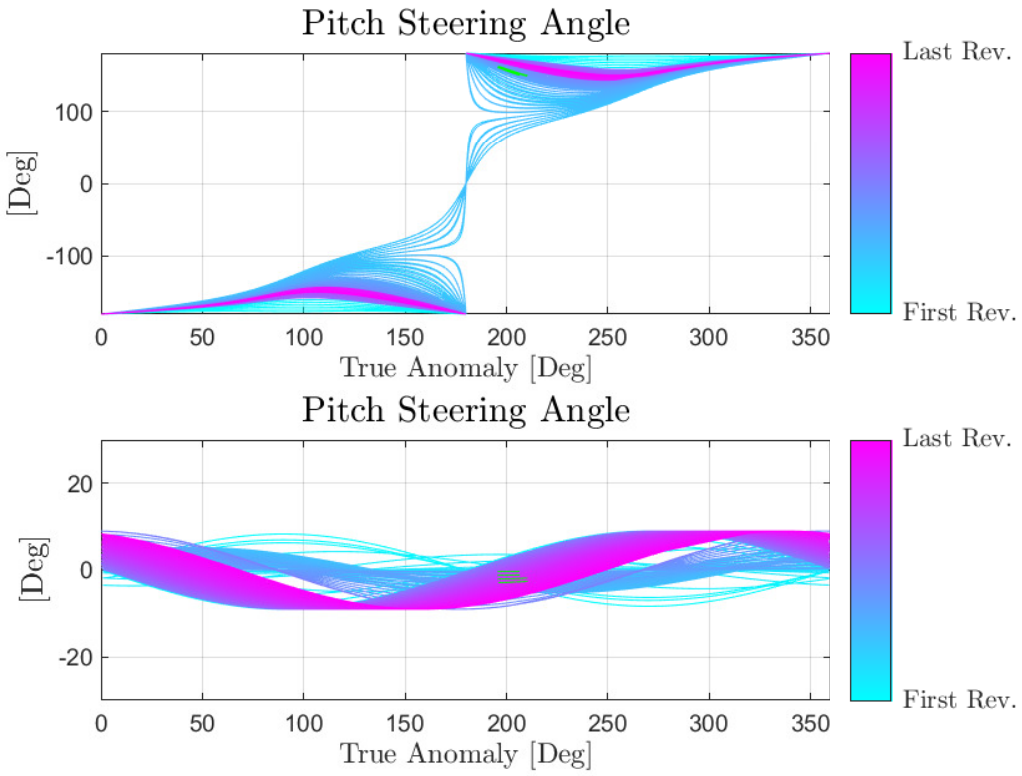

Figure 5.8: Pitch and Yaw angles profile

The control logic is clearly visible in the steering angles time history (Figure 5.8).
Note that during the last revolutions the pitch angle is almost completely set to 180 degrees (or equivalently -180 degrees), which corresponds to the direction opposite to that of the spacecraft motion. Also, a little control on the inclination is present because Phobos's orbit is not exactly equatorial.

5.4.2. Minimum Propellant Consumption

Regarding the minimization of the propellant consumption for the second transfer, the time of flight was not left free. The reason for this choice is that the algorithm would look for a minimum propellant mass solution regardless of the time of flight, leading to solutions that could be not practical due to extremely large transfer times.
A possible approach could be to include the time of flight in the optimization variable vector, and to bound it within some defined limits. However, a different way was followed in this work: the time of flight was fixed, and therefore not included in the set of optimization variables of the problem. The problem was solved for different values of the time of flight in order to assess how the minimum propellant mass evolves with it.

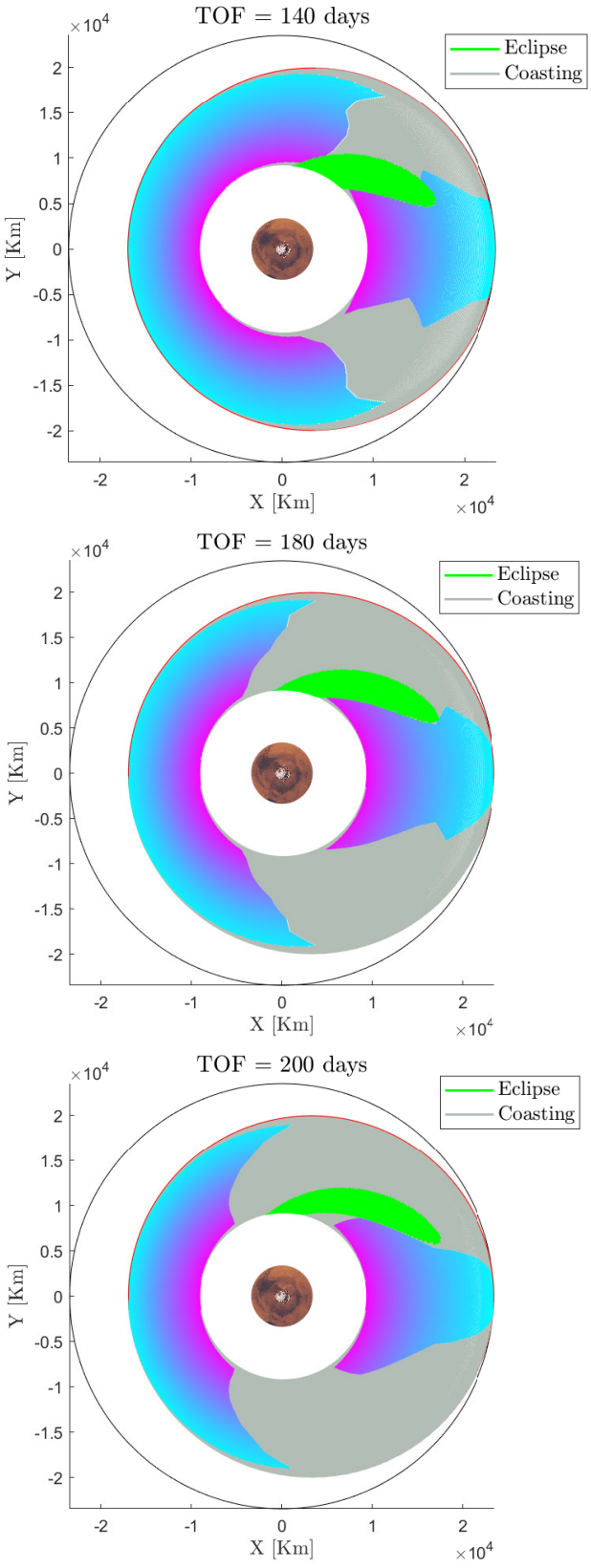

Figure 5.9: Transfer B minimum propellant trajectories

In Figure 5.9 the transfers for three different times of flight (140, 180, and 200 days) are depicted. Clearly, the maneuver is mainly performed at the pericentre: there the burning arcs are longer in order to produce a lowering of the apoapsis altitude and the circularization of the orbit. Similarly, the burning arcs at the apocentre are responsible for the lowering of the pericentre. As the time of flight increases, this pattern is maintained but an overall shortening of the burning arcs is experienced. In particular, this effect becomes more and more significant at pericentre as the spacecraft gets closer to the planet. Correspondingly, the burning arcs at the apocentre present an opposite behavior, broadening during the last revolutions of the transfer. This trend reveals the tendency of the control to first adjust the apocentre altitude during the early stages of the transfer, while correcting for the pericentre altitude in the final part.
Increasing the transfer time allows for a lower overall thrust amount, but only up to a certain point. In fact, increasing the time of flight over a certain threshold doesn't lead to an improvement in the propellant consumption. Not only, a slight increment in the consumption is experienced. The reason for this could be that, even if the burning arcs get shorter, the number of revolutions (and therefore the number of burning arcs) becomes so high that in the end the overall thrust amount start to increase.
The problem was therefore solved for a wide range of times of flight in order to identify the trend of propellant mass versus time of flight.

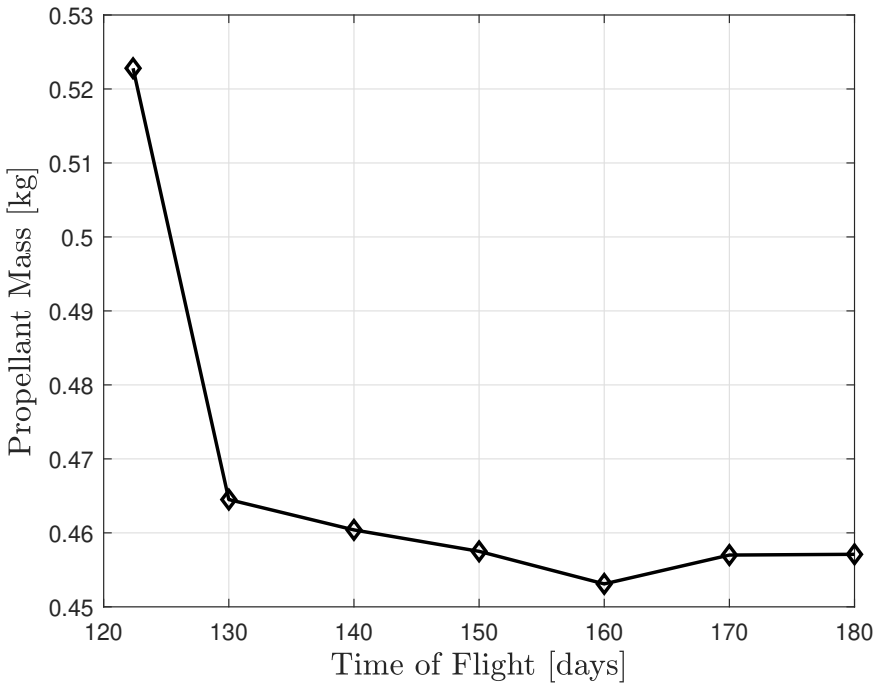

Figure 5.10: Optimal Propellant-Time of Flight solutions

From Figure 5.10 it is evident how an improvement in the propellant mass is achievable only up to a time of flight of 160 days. For longer transfers, the graph presents a plateau on higher propellant mass values for the aforementioned reasons.
The analysis carried out reveals an overall improvement in the performances with respect to the minimum time of flight solution (first marker in the graph) consisting in a reduction of about 13.3% in the consumed propellant, with an extension of the 31.1% in the time of flight (about 40 days). However, the majority of the improvement is achieved by extending the transfer duration by only 10 days (6.2%). With a 130 days flight indeed, a propellant save of 11.2% is reached.

Time of Flight	Propellant Mass
122.35 days	0.522 kg
130 days	0.464 kg
140 days	0.460 kg
150 days	0.457 kg
160 days	0.453 kg
170 days	0.457 kg
180 days	0.457 kg

Table 5.6: Optimal transfers results

5.5. Feasibility Analysis and Performances

For both Transfer A and B, the multiple-shooting scheme explained in section 3.3 was employed. The trajectory was discretized in 10 segments for Transfer A and 4 segments for Transfer B. Such a discretization allows to accurately integrate the dynamics with a reasonable number of optimization variables, in order to achieve a practical computational load. Obviously, as the transfer duration increases, an higher number of segments is required to integrate the dynamics, thus, the minimum propellant consumption problem results in the most expensive, in terms of computational load.
For both transfers the algorithm showed excellent convergence robustness, that is, the violation of the defect (continuity) constraints is well below the selected tolerance.
The relative errors are reported in Table 5.7 and 5.8.

Minimum TOF Problem - Transfer A				
Relative Error	$\|\varepsilon_a\|$	$\|\varepsilon_e\|$	$\|\varepsilon_i\|$	$\|\varepsilon_m\|$
Node 2	7e-9	5e-14	6e-15	4e-12
Node 3	3e-9	3e-14	5e-15	4e-12
Node 4	6e-10	9e-15	5e-15	4e-12
Node 5	9e-11	2e-15	5e-15	4e-12
Node 6	1e-10	4e-15	5e-15	4e-12
Node 7	4e-10	2e-14	5e-15	4e-12
Node 8	5e-10	3e-14	5e-15	4e-12
Node 9	2e-9	1e-13	1e-15	4e-12
Node 10	8e-10	9e-14	7e-14	4e-12

Table 5.7: Continuity violation at segments boundaries

Minimum TOF Problem - Transfer B				
Relative Error	$\|\varepsilon_a\|$	$\|\varepsilon_e\|$	$\|\varepsilon_i\|$	$\|\varepsilon_m\|$
Node 2	5e-5	2e-9	1e-9	3e-9
Node 3	3e-5	2e-8	5e-9	4e-9
Node 4	6e-5	8e-8	9e-8	5e-9

Table 5.8: Continuity violation at segments boundaries

Furthermore, the algorithm showed excellent convergence properties also in terms of errors on the final conditions.

Minimum TOF Problem			
Transfer	$\mathcal{E}_a$ [km]	$\mathcal{E}_e$ [-]	$\mathcal{E}_i$ [deg]
A	3e-9	5e-8	1e-5
B	0.0548	3e-6	0.05

Table 5.9: Minimum Time of Flight final absolute errors

Minimum Propellant Problem			
Transfer	$\mathcal{E}_a$ [km]	$\mathcal{E}_e$ [-]	$\mathcal{E}_i$ [deg]
B (140 days)	1e-11	2e-16	2e-11
B (150 days)	1e-11	2e-16	1e-17
B (160 days)	9e-12	2e-16	1e-14

Table 5.10: Minimum Propellant final absolute errors

From the absolute errors reported in Table 5.9 and 5.10 emerges a very good robustness of the algorithm in converging toward the target orbit with remarkable precision.
There is only one case in which the final error is slightly higher: Transfer B in the minimum time of flight problem.
However, even in this case the error achieved is very low (0.02% for semi-major axis, 6.8% for eccentricity, and 3.9% for inclination) and totally acceptable. The computational performances of the exploration scenarios are reported as well, in terms of CPU time, number of iterations, and number of function evaluations.

Minimum TOF Problem			
Scenario	CPU Time	Iterations	F-Count
Transfer A	5212 s	255	21865
Transfer B	1099 s	98	4854

Table 5.11: Algorithm performance on Minimum TOF problem

Minimum Propellant Problem			
Scenario	CPU Time	Iterations	F-Count
Transfer B - 140 days	956 s	41	3475
Transfer B - 150 days	2355 s	91	7648
Transfer B - 160 days	2263	76	6173

Table 5.12: Algorithm performance on Minimum Propellant problem

6 | Conclusions and Future Developments

The goal of the present study was to propose a preliminary mission analysis for the exploration of the Martian moons. Although the increasing interest in these celestial bodies, no dedicated mission has been successfully carried out so far. The idea at the basis of the investigation is to evaluate the possibility of exploiting small-size spacecraft in the Martian system, in order to make the exploration of the moons affordable and achievable. The employment of small-size spacecraft in such an exploration scenario presents two main criticalities. The first one concerns the extremely low T/W ratio of small-size spacecraft. Although the small mass of these platforms, the T/W is far lower than that of larger spacecraft due to limited onboard power resources. This characteristic has a major consequence: the produced acceleration is so low that an extremely large time of flight is necessary to perform a transfer. From one side this fact represent a problem because of the extended exposure of the spacecraft to the space environment.
On the other hand, such a large transfer time, in a planetary scenario, requires a demanding effort in the trajectory design and optimization, due to the remarkable number of revolutions around the body of the main attractor.
The second criticality arising when dealing with small-size spacecraft is that these devices hardly can reach their target by themselves. More likely they have to be carried there by some other larger mission. This adds a further degree of complexity in the design of the overall mission, due to driver requirements from the main spacecraft.
In light of these arguments, the trajectory design for the investigated scenario results challenging and cannot rely on standard approaches. Therefore, a trajectory optimization tool was developed, intended to comply with two main requirements derived from the previous considerations:

- **High Flexibility**: The trajectory optimization algorithm shall ensure good flexibility in terms of variable initial (release) conditions, in order to allow the adaptability of the mission analysis to any main mission.

- **Limited Computational Cost**: Although the high number of orbital revolutions around the main attractor, the algorithm shall be able to converge to a feasible solution with a relatively limited computational load, in order to obtain a tool that is fastly usable to perform many optimizations.

The flexibility requirement was met by employing a Direct Optimization strategy, which allows for adaptability to changes in the original problem and ensures high robustness in convergence. Concerning the second requirement, the computational effort was kept low by means of two design choices for the optimization algorithm. The first one regards the selection of a control parametrization scheme, which allows for including only few variables in the optimization vector. The second choice concerns the employment of the Orbital Averaging technique to ease the integration of the trajectory.
Naturally, these advantages don't come for free. The choice of using a Direct Method implies a sub-optimality of the found solution, as typically it is a local minimum.
Furthermore, Orbital Averaging features a significant drawback which is the loss of control on the fast oscillating state variable: the true anomaly. This could seem a major problem in the case of Transfer B (from Deimos resonant orbit to Phobos's science orbit) as the spacecraft is targeting a randez-vous with Mars's innermost moon.
However, a possible solution to this problem could be to perform a phasing maneuver once Phobos's orbit is acquired. Concerning the algorithm core, the approach proposed by Kluever and Oleson [6] to solve minimum time of flight problems was taken as a starting point, and was further extended. An eclipse model was included to account for Mars shadowing, and a multiple-shooting version of the algorithm was implemented to improve its robustness in convergence. But the major step forward was the reformulation of the algorithm to solve also the minimum propellant problem. To do that, the burning-coasting logic proposed by Gao [29] was adopted.
The result is an algorithm that intrinsically looks for a low time of flight solution (because of the analytical control law which maximizes orbital element variations) while looking for a minimum propellant consumption burning-coasting law. It is worth remarking that no considerations were done about the engine duty cycle. Further analysis of the specific exploration scenario should take into consideration also the technological feasibility of the burning-coasting program, in relation to the switching frequency of the engine.
Another critical aspect to point out about the presented results is that no attitude constraints were included in the optimization process.
This cannot be considered a valid assumption in a more detailed analysis, because there is a region in the control time history in which the required control effort, in terms of angular rates, is outside the control capabilities range of a spacecraft.

Another further development could be the inclusion of a high-fidelity dynamical model. Indeed, in the study only a Two-Body model was adopted with the inclusion of the J2 perturbing effect. To have a more realistic representation of the Martian system's gravitational environment, also the attraction of the two moons should be included, even if their contribution is very little and can be neglected in a first approximation.
The inclusion of Phobos's gravitational field, in particular, would allow to model the stable Mars-Phobos L2 Manifold in the Circular Restricted Three Body Problem dynamics (CRTBP), enabling a more complete analysis of the capture into the Phobos science orbit. Moreover, the considered release orbit parameters were chosen by looking at past Martian missions, but no considerations about the interplanetary leg were done.
Therefore, the high elliptical orbit was chosen regardless of any incoming asymptote. Despite the criticalities and the possible improvements aforementioned, the preliminary analysis presented in this work constitutes a solid starting point for a more detailed investigation.

Bibliography

[1] Englander, Jacob, et al. Mars, phobos, and deimos sample return enabled by arrm alternative trade study spacecraft. *AIAA/AAS Astrodynamics Specialist Conference*, 2014.

[2] Canales, David, et al. A transfer trajectory framework for the exploration of phobos and deimos leveraging resonant orbits. *Acta Astronautica*, 2022.

[3] Derz, Uwe, Andreas Ohndorf, and Bernd Bischof. Mission analysis of robotic low thrust missions to the martian moons deimos and phobos. *62nd International Astronautical Congress*, 2011.

[4] Cappelletti, Chantal, Simone Battistini, and Benjamin Malphrus. *Cubesat Handbook From Mission Design to Operations*. Academic Press, 2020.

[5] Conway, Bruce A. *Spacecraft Trajectory Optimization*. Cambridge University Press, 2010.

[6] Kluever, Craig A., and Steven R. Oleson. Direct approach for computing near-optimal low-thrust earth orbit transfers. *Journal of Spacecraft and Rockets*, 1998.

[7] Sackett, Lester L., Harvey L. Malchow, and T. N. Delbaum. Solar electric geocentric transfer with attitude constraints: Analysis. *NASA CR-134927*, 1975.

[8] Jimenez-Lluva, David, and Bart Root. Hybrid optimization of low-thrust many-revolution trajectories with coasting arcs and longitude targeting for propellant minimization. *Acta Astronautica*, 2020.

[9] Kelly, Matthew. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 2017.

[10] Rayman, Marc D., et al. Results from the deep space 1 technology validation mission. *Acta Astronautica*, 2000.

[11] Russell, Christopher, and Carol Raymond. *The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres*. Springer Science Business Media, 2012.

[12] Kawaguchi, Jun'ichiro, Akira Fujiwara, and Tono Uesugi. Hayabusa—its technology and science accomplishment summary and hayabusa-2. *Acta Astronautica*, 2008.

[13] Lunar ice cube. https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=L-ICECUBE.

[14] Lunar flashlight. https://www.jpl.nasa.gov/missions/lunar-flashlight.

[15] Lunarh-map. https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=LUNAH-MAP.

[16] Liciacube. https://www.asi.it/en/planets-stars-universe/solar-system-and-beyond.

[17] Dart. https://dart.jhuapl.edu/.

[18] Tolson, R. H., et al. Viking first encounter of phobos: preliminary results. *Science*, 1978.

[19] Duxbury, Thomas C. Spacecraft exploration of phobos and deimos. *Planetary and Space Science*, 2013.

[20] Martian moons exploration. https://www.mmx.jaxa.jp/en/.

[21] Ogawa, Naoko, et al. Orbit design for the martian moons exploration mission. *Transactions of the Japan society for Aeronautical and Space Sciences*, 2019.

[22] Edelbaum, Theodore N. Propulsion requirements for controllable satellites. *American Rocket Society Journal*, 1961.

[23] Colasurdo, Guido, and Lorenzo Casalino. Optimal low-thrust maneuvers in presence of earth shadow. *AIAA/AAS Astrodynamics Specialist Conference*, 2004.

[24] Kechichian, Jean Albert. Low-thrust eccentricity-constrained orbit raising. *Journal of Spacecraft and Rockets*, 1998.

[25] Scheel, Wayne A., and Bruce A. Conway. Optimization of very-low thrust, many-revolution spacecraft trajectories. *Journal of Guidance, Control, and Dynamics*, 1994.

[26] Betts, John T. Very low-thrust trajectory optimization using a direct sqp method. *Journal of Computational and Applied Mathematics*, 2000.

[27] Oleson Kluever. Hybrid method for computing optimal low thrust otv trajectories. *Advances in Astronautical Sciences*, 1994.

[28] Yang, Gao. Direct optimization of low-thrust many-revolution earth-orbit transfers. *Chinese Journal of Aeronautics*, 2009.

[29] Gao, Yang, and Xinfeng Li. Optimization of low-thrust many-revolution transfers and lyapunov-based guidance. *Acta Astronautica*, 2009.

[30] Betts, John T. *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial and Applied Mathematics, 2010.

[31] Morante, David, Manuel Sanjurjo Rivo, and Manuel Soler. A survey on low-thrust trajectory optimization approaches. *Aerospace*, 2021.

[32] Pritchett, Robert E. Strategies for low-thrust transfer design based on direct collocation techniques. *Diss. Purdue University Graduate School*, 2020.

[33] Topputo, Francesco, and Chen Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. *Abstract and Applied Analysis*, 2014.

[34] Williams, Steven N., and Victoria Coverstone-Carroll. Mars missions using solar electric propulsion. *Journal of Spacecraft and Rockets*, 2000.

[35] Englander, Jacob A., Bruce A. Conway, and Trevor Williams. Optimal autonomous mission planning via evolutionary algorithms. *21st AAS/AIAA Space Flight Mechanics Meeting*, 2011.

[36] Jim Lambers. Lecture notes in numerical analysis ii, 2009-2010.

[37] Betts, John T. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 1998.

[38] Walker, Michael JH, B. Ireland, and Joyce Owens. A set modified equinoctial orbit elements. *Celestial Mechanics*, 1985.

[39] Wakker, Karel F. *Fundamental of Astrodynamics*. TU Delft Repository, 2015.

[40] Wertz, James Richard. *Mission geometry; orbit and constellation design and management*. 2001.

[41] Nasa spice toolkit. https://naif.jpl.nasa.gov/naif/toolkit.html.

[42] Kechichian, Jean Albert. The treatment of the earth oblateness effect in trajectory optimization in equinoctial coordinates. *Acta Astronautica*, 1997.

List of Figures

List of Tables

List of Symbols

Variable	Description	SI unit
a	Semimajor-axis	km
a_n	Normal Perturbation	m/s^2
$\mathbf{a}_{pert}$	Perturbing Acceleration	m/s^2
a_r	Radial Perturbation	m/s^2
$\boldsymbol{a_T}$	Thrust Acceleration	m/s^2
a_t	Tangential Perturbation	m/s^2
e	Eccentricity	-
f	Second MEE	-
g	Third MEE	-
g_0	Gravitational Acceleration	m/s^2
G_a	Control Parameter	-
G_e	Control Parameter	-
G_i	Control Parameter	-
h	Fourth MEE	-
i	Inclination	rad
I_{sp}	Specific Impulse	s
J_2	Zonal Harmonics Coefficient	$km^5 s^{-2}$
k	Fifth MEE	-
L	True Longitude	rad
m	Mass	kg
p	First MEE	-
$\mathbf{q}$	In-Plane Thrust Direction	-
$\boldsymbol{r}$	Spacecraft Position	km

Variable	Description	SI unit
$\mathbf{T}$	Thrust	N
$\mathbf{u}$	Thrust Direction	-
$\mathbf{v}$	Spacecraft Velocity	km/s
w_a	Apocentre Thrust Weight	-
w_p	Pericentre Thrust Weight	-
α	Pitch Steering Angle	rad
β	Yaw Steering Angle	rad
γ	Flight Path Angle	rad
θ	True Anomaly	rad
μ	Gravitational Parameter	$km^3 s^{-2}$
ν	Velocity to (in-plane) Thrust Angle	rad
Ω	Right Ascension of Ascending Node	rad
ω	Argument of Pericentre	rad

List of Acronyms

CRTBP	Circular Restricted Three Body Problem.
GEO	Geostationary Orbit.
GTO	Geostationary Transfer Orbit.
HEO	Highly Elliptical Orbit.
IVP	Initial Value Problem.
MEE	Mean Equinoctial Elements.
MOI	Mars Orbit Insertion.
NLP	NonLinear Programming.
OCP	Optimal Control Problem.
RAAN	Right Ascension of the Ascending Node.
RK4	Runge-Kutta 4.
SEP	Solar Electric Propulsion.
SOI	Sphere Of Influence.
T/W	Thrust to Weight ratio.
TOF	Time Of Flight.
TPBVP	Two Points Boundary Value Problem.

Acknowledgements

This work is the result of years of dedication, commitment, and sacrifices. But not only. Behind each of these pages there are many people without whom this work wouldn't have been possible, and without whom I wouldn't be the person I am.

The greatest thanks to my family, for their love and patience, for the unconditional support they gave me since the first day, and for teaching me more than any university can do.

Thank you to my friends, who always believed in me, in the good and in the bad times, and who always were there to wait for me when I came home.

Thank you to all the incredible people I've met on the way, and to those with whom I spent days and nights, side by side, in the study room of Building 27, sharing crises, successes, insecurities, and jokes. Because they are the only ones who can really understand the intensity of this journey.

Thank you to my advisor, Professor Michéle Lavagna, for giving me the opportunity to work on a truly inspiring and challenging topic, and to my co-advisor, Jacopo Prinetto, for his availability and for the support he provided to me across all these months.

Finally, thank you to Andy Weir for inspiring my life through his writings.