POLITECNICO DI MILANO

MASTER THESIS

An analytical 3D shape-based algorithm based on orbit interpolation for multi-revolution low-thrust trajectory optimization

Author:
Jacopo PRINETTO

Supervisor:
Prof. Michelle LAVAGNA

July 11, 2018

Abstract

A novel 3-dimensional shape based algorithm is proposed in order to extend the domain of the analytical solutions to planeto-centric mission scenarios. The main peculiarity of the proposed approach stays in the shape belonging to a non linear interpolation of similar and consecutive orbits; this turns out in a more faithful forecast of the actual transfer shape in real conditions, leading to a better cost-effectiveness final solution which is ranked in terms thrust, power and fuel consumption for any combination of initial and final orbits features. Many classical shape based algorithms [3][42][9], in fact, due to the lack of physical meaning, fail whenever orbits which differ from regular shapes have to be linked, such as highly elliptical or with strong change in planes. More in details, both the distance from the attractor and the declination above the plane identified by the initial and final positions – here defined as reference plane - are parametrized as function of the angular displacement in the reference plane from the initial position by means of non linear interpolation between radii and declinations of the departure and arrival orbits. If more than a single revolution is required the algorithm simply either considers a transfer arc greater than 360 degrees or introduce fictitious intermediate orbits located via a numerical procedure to make the thrust peaks constants in each revolution, increasing the degrees of freedom and the flexibility of the shape. The kinematics and the dynamics are recovered analytically while the thrust and mass profile are obtained simply integrating Tsiolkovsky's equation numerically. The tool allows dealing with multidisciplinary/multiobjectives scenarios [28]: solar panels area needed to supply the requested thrust can be output as part of the optimal solution as well as the maximum thrust. Differently from the available literature [37] [15], the algorithm here proposed includes perturbations and -if consistent with the analyzed scenario – eclipses to tune the thrust accordingly, still preserving a low computational burden for fast low-thrust high accuracy trajectory design. Either free or imposed time of flight scenarios are manageable, giving the chance to solve both rendezvous and topping problems, in interplanetary or planet-centric scenarios. Since the algorithm is mainly analytical the computational cost is very low and therefore it is suitable for the selection and optimization of very complex mission scenarios [28].

Acknowledgements

I would like to thank my advisor Professor M. Lavagna for her support and for the motivation that she gave me during this work; without her help this thesis will never be written. I would like also to extend my thanks to all the Professors that I had the pleasure to meet in these years at Politecnico di Milano that helped me to reach a strong preparation. A special thank is reserved to my family, my girlfriend and my friends on which I could always lean on during my studies and my life: without their constant friendship and love today I would not have reached this important goal.

Contents

List of Figures

List of Tables

List of Abbreviations

TOF	**T**ime **O**f **F**light
MEE	**M**odified **E**quinoctial **E**lements
RF	**R**eference **F**rame(s)
KP	**K**eplerian **P**arameters
ODE	**O**rdinary **D**ifferential **E**quantion(s)
RK	**R**unge **K**utta (integration scheme)
AB	**A**dams **B**ashford (integration scheme)
AM	**A**dams **M**ulton (integration scheme)
GEO	**G**eostationary **E**arth **O**rbit(s)
LEO	**L**ow **E**arth **O**rbit(s)
MEO	**M**edium **E**arth **O**rbit(s)
CPU	**C**entral **P**rocessing **U**nit(s)
GPU	**G**raphics **P**rocessing **U**nit(s)
RAAN	**R**ight **A**scension **A**scending **N**ode

Physical Constants

Speed of Light	$c_0 = 2.997\,924\,58 \times 10^{8}\,\mathrm{m\,s^{-1}}$ (exact)
Mass unit	$MU = 1000$ [kg]
Earth Distance unit	$DU = 6378$ [km]
Lunar Distance unit	$DU = 1738$ [km]
Interplanetary Distance unit	$DU = 149597870.7$ [km]
Earth Time unit	$TU = 806.7856$ [s]
Lunar Time unit	$TU = 1034.8$ [s]
Interplanetary Time unit	$TU = 58.1324$ [days]
Sea level acceleration constant	$g_0 = 9.80665 \left[\frac{m}{s^2}\right]$
Solar flux at earth	$\phi_{Earth} = 1367 \left[\frac{W}{m^2}\right]$
Earth attractor parameter	$\mu = 398600 \left[\frac{km^2}{s^3}\right]$
Sun attractor parameter	$\mu = 132712440018 \left[\frac{km^2}{s^3}\right]$
Lunar attractor parameter	$\mu = 4902.8 \left[\frac{km^2}{s^3}\right]$

List of Symbols

a	Semi-Major axis	$[DU]$
a^r_{pert}	Radial Perturbation	$\left[\frac{DU}{TU^2}\right]$
a^t_{pert}	Transversal Perturbation	$\left[\frac{DU}{TU^2}\right]$
a^z_{pert}	Out of plane Perturbation	$\left[\frac{DU}{TU^2}\right]$
a_s	Semi-Major axis	$[DU]$
A_{sa}	Solar panels surface	$[m^2]$
e	Eccentricity	$[-]$
f	Second MEE	$[-]$
G_{loss}	Gravity loss	$[\%]$
g	Third MEE	$[-]$
h	Fourth MEE	$[-]$
$\hat{\mathbf{h}}_{REF}$	Ref. plane normal	$[-]$
$\hat{\mathbf{h}}_1$	Initial plane normal	$[-]$
$\hat{\mathbf{h}}_2$	Final plane normal	$[-]$
I_s	Specific Impulse	$[s]$
i	Inclination	$[deg]$
k	Fifth MEE	$[-]$
L	Sixth MEE	$[-]$
m	Mass	$[kg]$
N_{rev}	Number of revolution	$[-]$
P_{SS}	Spacecraft Power consumption	$[W]$
p	First MEE	$[DU]$
r	In-plane attractor distance	$[DU]$
s	attractor distance	$[DU]$
T	Thrust	$[FU]$
T_{IN}	In-plane Thrust	$[FU]$
T_{OUT}	Out of plane Thrust	$[FU]$
t	Time	$[TU]$
v_r	Radial velocity	$\left[\frac{DU}{TU}\right]$
v_z	Normal velocity	$\left[\frac{DU}{TU}\right]$
v_θ	transversal velocity	$\left[\frac{DU}{TU}\right]$
x	Non-dimensional anomaly	$[-]$
z	Out of plane displacement	$[DU]$
α	In-plane thrust angle	$[rad]$

β	Out of plane thrust angle	$[rad]$
γ	Flight path angle	$[rad]$
δ	Declination	$[rad]$
η_{tot}	Power production efficiency	$[-]$
θ	Anomaly	$[rad]$
ϕ	Solar aspect angle	$[rad]$
χ	Interpolating function	$[-]$
ψ	Total transfer angle	$[rad]$
ω	angular velocity	$\left[\frac{rad}{TU}\right]$

Dedicated to my Parents

Chapter 1

Introduction

One of the main subject in astrodynamics is the design and optimization of spacecraft trajectories. In simple terms, this process involves the determination of a trajectory that, given some constraints, minimizes one or more cost functions [12]. Among all possible objectives, the most common are certainly the fuel propellant fraction and the time of flight needed to accomplish the mission. Regarding the constraints, the initial and final position and, in some cases, also the velocities and the Time of Flight are typically imposed. In some particular missions other specific constrains and objectives arise as, just to give few examples, the power available, the time spent in some dangerous regions (Van Allen belts), the cost up to the launcher, the scientific drawback (for example the sequence of target visited in an interplanetary trajectory [28]) and so on. Objectives and constrains, as well as the optimization method itself, depend mostly on the propulsive system of the spacecraft and the specific mission. In some cases it is not so clear if something has an objective or constrains nature; moreover, a powerful and commonly used way to manage constrains that are not exactly known in the earlier phases of a project (such as the maximum power available or the maximum thrust level) is to turn a constraint into an objective function using or a penalty method or a multi-objective optimization algorithm. Mathematically speaking, the problem of spacecraft trajectories optimization is quite complex, being composed by seven coupled non linear differential equations [12] (six equation for 3D position and velocity plus one for the mass), that in general does not admit an analytic integral solution. Just to make the problem more complex, the motion of the spacecraft is forced by thrusters and natural perturbations, that at least depend on the epoch and the spacecraft position, velocity, shape and mass. Some discontinuities in the mathematical problem can be introduced by vehicles separations, Gravity assists, instantaneous change in velocity or change in the Reference Frame. Until 1998 all the spacecraft flown equipped with chemical thrusters only: this king of thrusters are able to provide an extremely wide range of thrust, from the smallest with fractions of Newtons to the biggest with thousands or (for launchers only) millions of Newtons. If a propulsive system is able to provide an high thrust to weight ratio the time required to change the velocity of the spacecraft becomes small when compared with the entire trajectory, and therefore it can be approximated with an impulse maneuver without introducing appreciable errors. In this last case the above-mentioned optimal control problem becomes a simpler parametric one: the problem is still complicated by the presence of mixed integer and continuous variables but it can be easier managed with derivative free algorithm[12][32][33]. Unfortunately chemical propulsion has an upper limit on the specific impulse in around $500[s]$ given by the maximum possible ratio between temperature inside the combustion chamber and molecular weight of the expelled gases [22]. For this reasons, in the last years of the last century, electric propulsion becomes of primary interest: on the October 24^{th} 1998 NASA successfully launched Deep Space one [35], an interplanetary mission that used as primary propulsion an electric ion thruster. Electric thrusters give enormous advantages if compared with chemical ones [16][39][17] in terms of Specific impulse, implying a strong reduction of the fuel mass fraction required for the same mission. On the other hand, an enormous amount of power per unit thrust is needed [24](between 15 $[W/mN]$ and 40 $[W/mN]$). Flight proven thrusters are nowadays able to give in around $0.25[N]$ of thrust, while next future implementations, such as Sitael H 20K [40], will be able to have thrust in around one Newton, with a specific impulse of more than $3000[s]$, one order of magnitude higher than bests in-space chemical thrusters. In the lasts decades the continuous development of electric thrusters makes possible interplanetary missions that before were well beyond technical feasibility, such as Dawn [30], Hayabusa [29] and BepiColombo [19]. On the other hand, Electric Propulsion becomes of primary importance in Earth-Centered mission scenarios, especially for Geostationary station keeping. In 2003 an ESA telecommunications satellite named Artemis [20], due to a launcher failure, was injected in an orbit lower that GEO and reaches its target only thanks to its own propulsion system. In the last decades fully electric platforms have been developed by different industries and it is expected to have an increase of interest both for commercial and for scientific satellites.

1.1 Low thrust trajectory optimization

Low thrust trajectories optimization is a complex task: the system of equations governing the dynamics is shown in cylindrical coordinates [42] in equation 1.1.

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

It is composed by six coupled non-linear first order differential equations describing the position and velocity of the spacecraft plus one additional differential equation (Tsiolkovsky equation) that governs the mass variation of the spacecraft. The only possibility to have exact analytical solution is when purely keplerian motion with no thrust and no perturbations is considered. In presence of low thrust control it becomes an optimal control problem. It is necessary to find a control law ($T(t)$, $\beta(t)$, $\alpha(t)$) that, respecting some constrains, is able to lead the spacecraft to the final position and velocity from the initial ones, minimizing one or more cost functions. Typically this can be managed with a lot of different methodologies[7], that can be grouped in three macro-categories:

- **Direct Methods:** A Direct method [13][21] involves the conversion of the continuous optimal control problem into an high-but-finite dimension parametric optimal one. This is typically possible via a discretization of the state and the time (referring to equation 1.1: $r, \theta, z, v_r, \omega, v_z, m, t$) in a finite number of nodes. In the sub-domains between two consecutive nodes the states are usually approximated with a continuous simple function (for example a third order polynomial in the Hermite-Simpson method; higher order are also possibles [25]) that must satisfy equations of motion on the nodes. The equations of motion are threated as constrain, and therefore the algorithm forces the solution to respect them also on intermediate points [21] [38]. The solution is obtained using Nonlinear programming techniques in order to manage the size of the problem. These algorithms are quite robust and allow to easily include technical constrains and discontinuities[28], but unfortunately their computational cost is extremely high, since the typical size of the problem is in the order of hundreds or thousands of unknowns [28] [41] [8]. The algorithm is faster if a near-optimal solution, that is able also the respect the constrains, is imposed as guess. Due to the high computational cost of the algorithm, direct methods are not suitable for fast optimization in large search domain, moreover the classical non-linear solvers are not able to manage integer variables, that become of primary importance in these scenarios (for example the sequence of Gravity assists in an interplanetary trajectory).

- **Indirect Methods:** In indirect methods the continuous optimal control is turned into a finite and reduced dimension parametric optimal problem using the Calculus of Variation or the Pontryagin's principle. Due to the time dependence of the problem also the Lagrange multipliers are time dependent and possess their own dynamics, doubling the size of the problem. The problem is converted into a Two Points Boundary Values Problem, having states ($x_i(t)$) and co-states ($\lambda_i(t)$) as unknowns, that can be solved with a lower computational cost with respect to direct methods. On the other hand, these methods are less robust and require that the guess solution is sufficiently close to the optimal one [12].

- **Shape-based Methods:** Using Shape-based algorithms the continuous optimal control problem is reverted: a shape, with proper degrees of freedom, is

> assumed for the trajectory and the control law is recovered. Typically some assumption [42][9], such as the tangential direction of the thrust, are assumed in order to get faster analytical solutions. These methods, working on a subspace of the problem, are able to give only a suboptimal solution and are extremely fast if compared with the others [3] [28] [42]. They are well suited for fast search of sub-optimal solutions in wide search domains using Heuristics algorithms, with the possibility to use multiple/multidisciplinary objectives [28]. These solutions can be used or as initial guests for direct or indirect optimizations or in the earlier phases of the design of a space mission [28].

These lasts methods are the objective of this work and will be deeply discussed in the following paragraph.

1.2 State of the art and Motivation of the research

The history of analytical solution for trajectories with thrusting spacecraft started in 1953 with the work done by Tsien [26]. In this first work, which has been reworked also by Battin [6], analytical approximated planar solutions are derived in case of or radial or circumferential thrust. Other interesting analytical solutions have been found but they all suffer from some simplifications. Most of shape based algorithms give the advantage to solve, at least for planar motion, the exact problem. In the lasts two decades a lot of work have been done in order to make shape-based algorithm able to solve increasingly complex problems. The first shapes developed, perfectly summarized in the work by Petropoulos and Sims [3], was able only to solve simple planar problems without the possibility to impose exact boundary conditions on the positions and velocities and with no TOF constrain. These algorithms were only used to have a quickly estimation of the low thrust trajectory cost and to generate guess trajectories. An important improvement in flexibility of the trajectory and in precision of the solution was proposed by De Pascale and Vasile [37]; the possibility to impose exactly BC gives also the chance to include Gravity assist maneuvers [4] in the trajectory. Conway and Wall developed a new simple but effective shape for both 2D problems [9] and approximated 3D problems [42] with small displacement from plane. Other improved shape-based algorithms were found by Novak and Vasile in [15], in which a new coupling between analytic solution and LQR controller is presented. Furier series have also been used in order to generate more effective shapes able to yield solutions nearer to the optimal one [18] All the above-mentioned shapes, and some other variations proposed by other authors, give very god results in interplanetary trajectory design: the flexibility of the methods and the reduced computational costs allow to design and optimize complex mission scenarios, with mixed integer and continuous variables and with multidisciplinary objective [28]. Unfortunately, all the above mentioned algorithms fail or give bad results whenever applied in earth-centered scenarios for the following reasons:

- **Number of revolution.** Being the planeto-centric dynamics more constrained with respect to the interplanetary one, the number of revolutions required is higher: typically hundreds or thousands of revolutions are required. On one single revolution the osculating elements are almost unchanged, and therefore it is necessary to develop a shape that is able to replicate this behavior. Algorithms that impose in advance a purely geometrical shape [9][42][3] are not able to closely follow this dynamic if the orbit is not circular or changes of plane are presents.

- **Eclipses.**Nowadays electric propulsion is powered by solar energy only. Even if the batteries can provide energy to the spacecraft during the eclipse for 'normal' operations, the power required by an electric thruster is extremely high and therefore the solution to thrust in shadow complicates the architecture of the propulsive system [43][34]. Moreover the usage of batteries in shadowed regions implies the necessity to have bigger solar arrays in order to recharge them in sunlight [43]. In order to solve this issues the algorithm shall be able also to manage discontinuities.

1.3 Structure of the report

The working principle of the algorithm is to turn a complicated multi-revolution problem in an elevated number of simpler one revolution trajectories. In order to do that an appropriated number (determined by the algorithm itself) of intermediate orbits are introduced and correctly placed. The report follows the two-level structure of the developed algorithm:

- **Chapter 2** is devoted to the presentation of the one revolution shape-based algorithm. After a rigorous parametrization of the three dimensional equations of motion both the TOF free and constrained architectures are presented. Each block involved is discussed in details and the most important equations developed within this work are derived. The numerical techniques adopted are discussed.

- **Chapter 3** is dedicated to the numerical validation of the algorithm developed in Chapter 2. A sensitivity analysis is performed in order to compute the gravity loss for some common maneuvers in different situations and to test the effects of the TOF constrain on the solution. An error analysis is conducted to assess the convergence rate of the algorithm, as well as the required computational time.

- **Chapter 4** is dedicated to the usage of the shape-based algorithm developed in chapter 2 in multi-revolution mission scenarios. Both the planeto-centric and interplanetary cases are explained and discussed.

- **Chapter 5** contains some interesting examples for both earth-centered and interplanetary cases.
- **Chapter 6** is devoted to the conclusions and to the explanation of some future improvements.

Nowadays optimization of planetocentric low-thrust trajectories is performed only using or indirect methods [36][23] or direct methods [8] without tanking into account eclipses effects. The aim of this research is to develop a shape-based algorithm sufficiently fast and flexible to be able to deal with Earth-centered problem. Since the number of revolutions is elevate consecutive revolutions are very closer a shape able to follow this dynamics even in presence of plane changes or elliptical orbits will be developed.

Chapter 2

Shape-based algorithms

In this chapter the developed shape-based algorithm will be analyzed in-depth and derived in both fixed and free TOF formulations starting from the equation of motion of a thrusting spacecraft via a nonlinear interpolation between departure and target orbits. The working principle of the algorithm is to compute the attractor distance and the declination above the reference plane of the trajectory (respectively s and δ) using a non-linear interpolation between the corresponding functions on the departure and target orbits. From these the kinematics and the dynamics are recovered using an analytical approach. Following the requirements analysis in section 2.1 the equations of motion for a thrusting spacecraft will be parametrized is section 2.2. The general architectures of the algorithms are presented in section 2.3 and the main blocks are discussed as well as the interpolating functions. In section 2.4, section 2.5 and section 2.6 the most complex blocks that are common for the two algorithms are derived and commented.

2.1 Requirements

The aim of this work is to adapt the well-known shape based-approaches [4] [37] [9] to a planeto-centric environment, with thousands of revolutions, eclipses and perturbations that can be of the same order of magnitude of the available thrust. The effect is a transfer trajectory with hundreds or thousands of revolutions and discontinuities, in which each arc is similar to the nearby. For these reasons the following requirements have been identified as fundamentals:

1. The algorithm shall link two different states (Modified Equinoctial Elements [10] or Keplerian Parameters) with the possibility also to impose the time of flight.
2. The algorithm shall work with any couple of orbits that are physically linkable with zero radial thrust [42] (every set of orbits for which equation 2.15 is positive along the whole path) without show singularities, including polar and retrograde orbits.
3. The thrust acceleration shall be exactly recovered via an analytical procedure.

4. The thrust and mass profiles shall be recovered via numerical integration of Tsiolkovsky equation [14].

5. The required thrust shall tend to zero as the distance between initial and final orbits decreases, independently from eccentricity and departure or arrival anomaly:
$$\lim_{\Delta KP \to 0} \max\left(\frac{|T|}{m}\right) = 0 \qquad \forall \quad e, i, \theta_1, \theta_2 \tag{2.1}$$

The last requirement is necessary in order to have feasible solutions in Earth-centered mission scenarios: if it is not fulfilled it is not possible to arbitrary reduce the maximum thrust required simply adding more intermediates orbits. The decision to build a shape based on nonlinear interpolation between consecutive orbits belong to this requirement, indeed most of shapes in literature show thrust profile that violates it if eccentric orbits are considered. Another important advantages using such type of algorithms is that, as will be shown later, the boundary conditions are automatically imposed by the interpolating function itself without the necessity to solve equations at each evaluation.

2.2 Parametrization of low thrust equations

The aims of this section is to formalize the parametrization of the 3D equation motion of a thrusting spacecraft in cylindrical coordinates. Differently from the work done by [42] it makes no assumption on small displacement out the plane. The equations of motion presented in the first chapter can be rewritten as shown in equation 2.2 [11]

$$\begin{cases} \ddot{r} - r\dot{\theta}^2 = -\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha \\ r\ddot{\theta} + 2\dot{r}\dot{\theta} = \frac{T_{IN}}{m}cos\alpha \\ \ddot{z} = -\frac{\mu}{s^3}z + \frac{T_{OUT}}{m} \end{cases} \tag{2.2}$$

The idea is to parametrize every quantities involved in equation 2.2 as function of the non-dimensional anomaly x introduced in equation 2.3 and then, under some typical assumptions [9], recover the thrust, mass and time profiles. Physically $'x(t)'$ is the angle between the initial position vector and the projection of the spacecraft on the reference plane normalized with the total transfer angle ψ, as can be seen in figure 2.1.

$$x(t) = \frac{\theta(t)}{\psi} \qquad with \quad x \in \mathbb{R} \quad \wedge \quad 0 \leq x \leq 1 \tag{2.3}$$

It is supposed to have also an analytical parametrization of the in-plane radius $'r'$ and the out-of-plane displacement $'z'$, as shown in equation 2.4. In general the parametrization of these quantities is not unique: the one selected in this work will

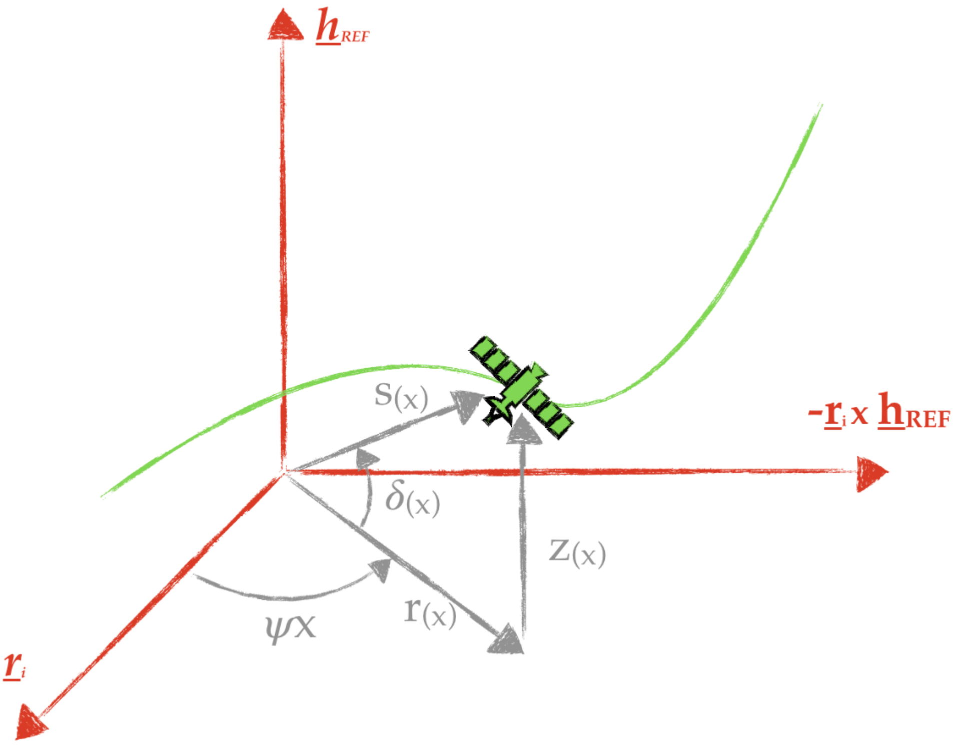


FIGURE 2.1: Spacecraft position in cylindrical and spherical coordinates

be shown later in section 2.5

$$\begin{cases} r = r(x) \\ z = z(x) \end{cases} \tag{2.4}$$

The first and second time variation of the in-plane angular displacement can be easily computed as shown in equation 2.5 and equation 2.6

$$\dot{\theta} = \frac{d\theta}{dt} = \psi \dot{x} \tag{2.5}$$

$$\ddot{\theta} = \frac{d^2\theta}{dt^2} = \psi \ddot{x} \tag{2.6}$$

Since both the in-plane radius and the out of plane displacement are considered as function of the in-plane non-dimensional anomaly in order to take their first and second time derivative is necessary to apply the rules for composite function derivation, as shown respectively in equation 2.7, equation 2.8, equation 2.9 and equation 2.10

$$\dot{r} = \frac{dr}{dt} = \frac{dr}{dx}\frac{dx}{dt} = r'\dot{x} \tag{2.7}$$

$$\ddot{r} = \frac{d^2r}{dt^2} = \frac{d}{dt}\left(r'\dot{x}\right) = r''\dot{x}^2 + r'\ddot{x} \tag{2.8}$$

$$\dot{z} = \frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = z'\dot{x} \tag{2.9}$$

$$\ddot{z} = \frac{d^2z}{dt^2} = \frac{d}{dt}\left(z'\dot{x}\right) = z''\dot{x}^2 + z'\ddot{x} \tag{2.10}$$

These quantities can be substituted into equation 2.2 giving equation 2.11

$$\begin{cases} r''\dot{x}^2 + r'\ddot{x} - r\psi^2\dot{x}^2 = -\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha \\ 2\psi r'\dot{x}^2 + r\psi\ddot{x} = \frac{T_{IN}}{m}cos\alpha \\ z''\dot{x}^2 + z'\ddot{x} = -\frac{\mu}{s^3}z + \frac{T_{OUT}}{m} \end{cases} \tag{2.11}$$

from the first two equations of system shown in equation 2.11, that describe the in-plane motion, is possible to extract the second time derivate of the non-dimensional anomaly, as shown in equation 2.12

$$\begin{cases} \ddot{x} = \frac{1}{r'}\left[-\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha - r''\dot{x}^2 + r\psi^2\dot{x}^2\right] \\ \ddot{x} = \frac{1}{r\psi}\left[-2\psi r'\dot{x}^2 + \frac{T_{IN}}{m}cos\alpha\right] \end{cases} \tag{2.12}$$

In order to analytically compute the time derivative of $'x'$ is necessary to remove the dependency from the thrust per unit mass in equation 2.12 : this is possible if and only if the in-plane thrust is supposed to be only tangential to the trajectory [9], indeed in this case the thrust angle is exactly equal to the flight path angle that can be easily computed as shown in equation 2.13

$$\tan\alpha = \tan\gamma = \frac{v_r}{v_\theta} = \frac{\dot{r}}{r\dot{\theta}} = \frac{r'}{r\psi} \tag{2.13}$$

Putting together equation 2.13 and equation 2.12 is possible to derive equation 2.14

$$-\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\gamma - r''\dot{x}^2 + r\psi^2\dot{x}^2 = \frac{T_{IN}}{m}sin\gamma - 2\frac{r'^2}{r}\dot{x}^2 \tag{2.14}$$

now it is possible to extract the square of the time variation of x as shown in equation 2.15

$$\dot{x}^2 = \frac{\mu r}{s^3\left(r\psi^2 - r'' + 2\frac{r'^2}{r}\right)} = \frac{Nu}{De} \tag{2.15}$$

taking the time derivate of equation 2.15 is possible to derive also the second time variation of the non-dimensional anomaly, as shown in equation 2.16 and equation 2.17

$$\ddot{x} = \frac{1}{2}\left(\frac{Nu' - \dot{x}^2De'}{De}\right) \tag{2.16}$$

$$\begin{cases} Nu' = \mu r' \\ De' = 3\frac{s'}{s}De + s^3\left(r'\psi^2 - r''' + \frac{2rr'r'' - r'^3}{r^2}\right) \end{cases} \tag{2.17}$$

Since now every time depended quantities is known it is possible to recover from equation 2.11 the thrust per unit mass as shown in equation 2.18 and equation 2.19.

$$\frac{T_{IN}}{m} = \frac{1}{\cos\gamma}\left(2\psi r' \dot{x}^2 + r\psi\ddot{x}\right) \tag{2.18}$$

$$\frac{T_{OUT}}{m} = z''\dot{x}^2 + z'\ddot{x} + \frac{\mu}{s^3}z \tag{2.19}$$

In order to recover the mass profile it is necessary to numerically integrate the Tsiolkovsky equation [22] shown in 2.20, in which the thrust per unit mass magnitude can be computed with equation 2.21

$$\frac{dm}{dt} = -\frac{\left|\frac{T}{m}\right| m}{I_S g_0} \tag{2.20}$$

$$\left|\frac{T}{m}\right| = \sqrt[2]{\left(\frac{T_{IN}}{m}\right)^2 + \left(\frac{T_{OUT}}{m}\right)^2} \tag{2.21}$$

A numerical integration is also needed to compute the time vector from the variation of the non-dimensional anomaly [9] , as shown in equation 2.22

$$t = \int_0^t \mathrm{d}\tau = \int_0^t \frac{1}{\dot{x}}\,\mathrm{d}x \tag{2.22}$$

It is important to underline that, differently from the parametrization proposed by Wall [42], the one here presented is more general, being suitable for any shape, and removes the assumptions of low displacements from reference plane. The drawback is an increased complexity of the model, as will be shown later in this chapter.

2.3 General Architecture

In this section the general architecture of the algorithm is presented both for the free and prescribed TOF problems. The main blocks are identified and discussed as well as requirements of the interpolation functions. It is important to remark that, in the Matlab implementation, every quantity is a vector of size n in which the $i-th$ element corresponds to the $i-th$ value of the non-dimensional anomaly x; The non-dimensional anomaly is a linearly spaced vector between zero and one with size equal to n and it is passed to the algorithm as a parameter. Being the mass and time profiles computed via a numerical integration (see equation 2.20 and equation 2.22) the higher is n the lower is the numerical error: a more detailed error analysis is proposed in Chapter 3 . The thrust profile belongs to the mass and, therefore, it is affected by the same error, while the other quantities are exact for any value of n since they are computed with a fully analytical approach.

2.3.1 Free Time of Flight

In this section the solution of the problem with no constraints on TOF will be discussed. Moreover the requirements on the interpolating function $\chi(x)$ will be defined and discussed: one of the possible choices will be proposed.

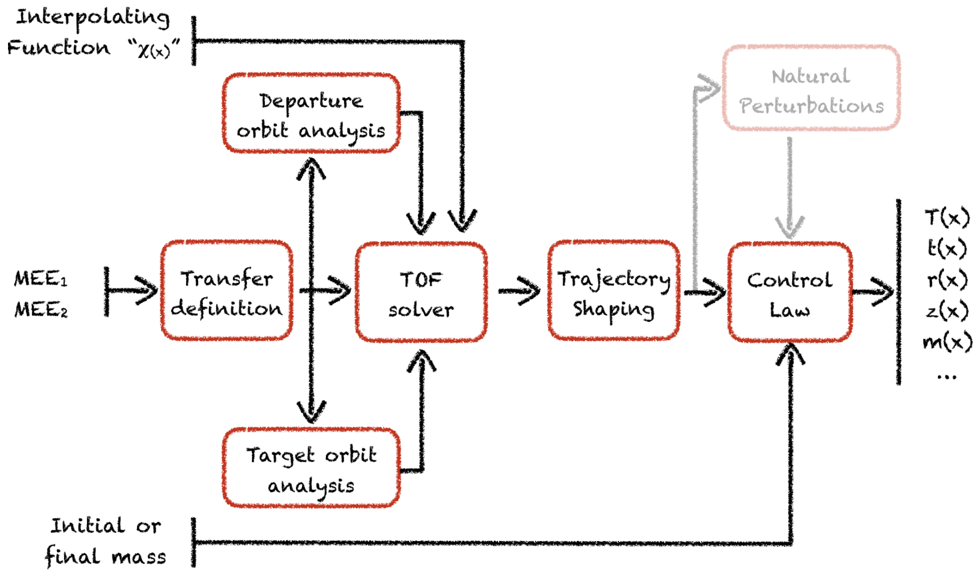


FIGURE 2.2: TOF free algorithm architecture

In figure 2.2 the architecture of the algorithm is presented: it consists of five main interconnected blocks plus one secondary block in which natural perturbations are implemented. It can be manually switched on if perturbation effects are desired.

- **Transfer definition:** This block takes as inputs the departure and arrival states. Its aims are to compute the reference frame and the transfer angle as shown is section 2.4. For fastness purposes it compute also the sine and cosine of the angles ψx and $\psi(1-x)$ using the angle transformation formulas. These outputs are the bases of transfers and are therefore passed to the ′ Departure/Target′ blocks and to the ′Trajectory shaping′ block.

- **Departure/Target orbit analysis:** These blocks take as input the data coming from the Transfer definition one. They compute some important geometrical quantities that will be used for the interpolation of the shape. The output includes all the quantities computed in section 2.6 and they are passed directly to the ′Transfer shaping′ block.

- **Trajectory shaping:** This block takes as input the geometries of departure and target orbits computed in the previous blocks and an interpolating function. It computes the exact geometry of the transfer using the non-linear interpolation between departure and target orbits as explained in section 2.5. The output includes the whole geometry and is directly passed to the ′Control law′ block and to the ′Natural perturbation′ one.

- **Natural perturbations:** This block takes as inputs the trajectory and computes the acceleration due to the natural perturbations (J2 effect). The output, consisting of the acceleration, is sent to the 'Control Law' Block. This block is not fundamental, and can be manually switched on if needed.

- **Control Law:** This block takes as input the geometry of the transfer and the natural perturbation acceleration. The control law, as well as the time vector are computed via a mixed numerical-analytical procedure. In order to compute the time vector, equation 2.22 is integrated using an high order multi-step predictor-corrector scheme (Adams-Bashford-3 Adams-Multon-4 [1]), while for the mass profile the same integration scheme is used to integrate Tsiolkovsky equation[14] (equation 2.20). These integrations can be performed both forward and backward: in this way it is possible to solve both problems with imposed dry mass or imposed wet mass. The thrust indeed is computed as the element-wise product of the mass and thrust acceleration vectors (eventually corrected with natural perturbations). Since this block presents two ODE to be solved, it is one of the most time consuming. Anyway it is important to underline that typically an ODE solver spends most of the time to evaluate the function to be integrated but, as can be seen in equation 2.22 and equation 2.20 these information have been yet computed during previous blocks, and are therefore available. Moreover, the selection of an accurate integration scheme is necessary to reduce the number of computation nodes, keeping the numerical errors low.

Boundary conditions

The boundary conditions to be imposed are simply the initial and final state. They are imposed directly on the interpolating function as requirements, indeed it is easy to show that the same interpolating function can be used for every trajectory. As first requirement the interpolating function $\chi(x)$ shall be continuous with its derivatives till the third order in the domain $[0;1]$. The boundary conditions are expressed in cylindrical coordinates as follows:

- **Position:** Looking at equation 2.33, equation 2.36 and equation 2.37, it is easy to derive that initial and final conditions on positions (in plane radius $r(x)$ and out of plane displacement $z(x)$) are automatically satisfied if the interpolating function satisfies equation 2.23

$$\begin{cases} \chi(0) = 0 \\ \chi(1) = 1 \end{cases} \tag{2.23}$$

- **Velocity:** From the definition of the radial velocity ($\dot{r}$) in equation 2.7 and the out of plane velocity ($\dot{z}$) in equation 2.9 it is clear that $r', z', \dot{x}$ must match the corresponding quantities of the initial state for $x = 0$ and final state for $x = 1$.

If the previous conditions are verified the boundary conditions on transversal velocity are automatically satisfied, being this velocity defined as $v_r = r\psi\dot{x}$. The requirements on r' and z' at the initial and final point can be directly derived from equation 2.34 and equation 2.35 and are summarized in equation 2.24

$$\begin{cases} \chi'(0) = 0 \\ \chi'(1) = 0 \end{cases} \tag{2.24}$$

For the initial and final conditions on $\dot{x}$, it is necessary to look at equation 2.15 in which appears also the second derivatives of r, therefore it is necessary to impose boundary conditions also on r''. From equation 2.34 is possible to derive the conditions on χ shown in equation 2.25

$$\begin{cases} \chi''(0) = 0 \\ \chi''(1) = 0 \end{cases} \tag{2.25}$$

In the examples presented in this work the seventh order polynomial function shown in equation 2.26 was selected.

$$\chi(x) = -20x^7 + 70x^6 - 84x^5 + 35x^4 \tag{2.26}$$

It is easy to show that this function satisfies the above-mentioned boundary conditions. Moreover, it is easy to prove that this interpolation function leads to solutions with null initial and final thrust: typical undesired peaks at the beginning and the end of the transfer arc (see the thrust profile obtained in [42] and [9]) are avoided.

2.3.2 Constrained Time of Flight

In this section the solution of the problem with constrained TOF will be discussed. Moreover, requirements on the interpolating functions $\chi(x)$ will be defined and some possibilities will be explored. In figure 2.3 the architecture of the algorithm is presented: it consists of six main blocks interconnected plus one secondary block in which natural perturbations can be implemented.

- **Transfer definition:**This block takes as inputs the departure and arrival states. Its aims is to compute the reference frame and the transfer angle as shown is section 2.4. For fastness purposes, it computes also the sine and cosine of the angles ψx and $\psi(1-x)$ using the angle transformation formulas. These outputs are the bases of the transfers and are therefore passed to the ' Departure/Target' block and to the 'TOF solver' block.

- **Departure/Target orbit analysis:** These blocks take as input the data coming from the Transfer definition one. They compute some important geometrical quantities that will be used for the interpolation of the shape. The output includes all the quantities computed in in section 2.6 and it is passed directly to the 'TOF solver' block.

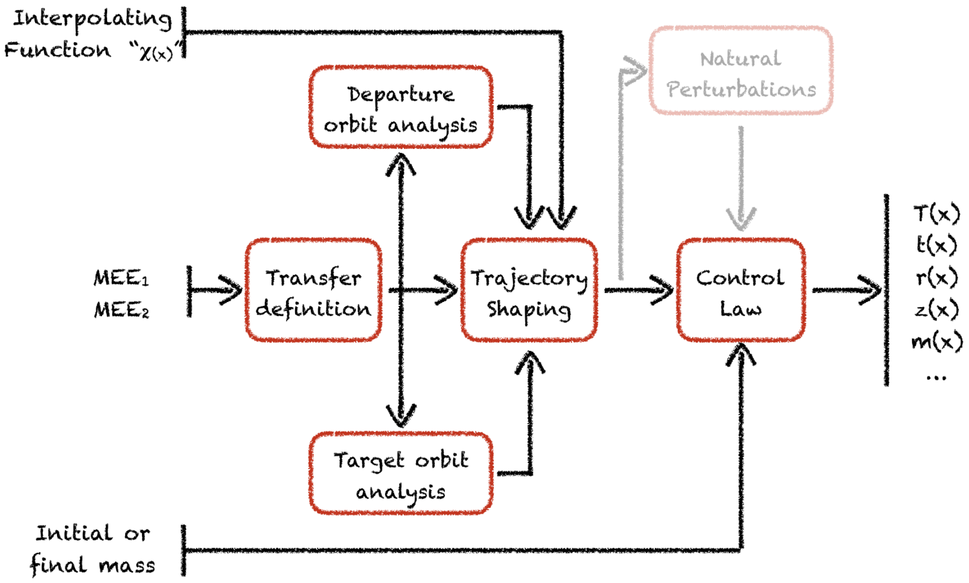


FIGURE 2.3: constrained TOF algorithm architecture

- **TOF solver:** The aim of this block is to impose a prescribed time of flight to the shape. In order to do that, the shape itself shall contain a further degrees of freedom not fixed with the boundary conditions mentioned in the previous section. This degrees of freedom is included in the interpolating function: an eight order polynomial that, after the imposition of the boundary condition above-mentioned, reads as in equation 2.27

$$\chi(x,a) = a_s x^8 - (20+4a_s)x^7 + (70+6a_s)x^6 - (84+4a_s)x^5 + (35+a_s)x^4 \tag{2.27}$$

In order to force the time of flight to be equal to the desired one, it is necessary to numerically solve equation 2.28 in the unknown a

$$TOF - \int_0^t \frac{1}{\dot{x}(a)}\,\mathrm{d}x = 0 \tag{2.28}$$

this can be easily done using a newton algorithm initialized with $a = 0$. It is easy to prove that for this value the shape in equation 2.27 is exactly equal to the one adopted for TOF free problem (see equation 2.26), this guarantees a feasible guess for the solver that is also close to the solution, making the convergence faster. The integral presents in equation 2.28 shall be solved numerically at each step of the Newton solver [1]. For this purpose a Cavalieri-Simpson integration scheme [1]is adopted in order to guarantee an acceptable compromise between numerical error reduction and fast computation. The function $\dot{x}(a)$ can be computed at each step using equation 2.15. The output of this block is the interpolating function, that is passed to the trajectory shaping block.

- **Trajectory shaping:** This block takes as input the geometries of departure and target orbits computed in the previous blocks and the interpolating function. It computes the exact geometry of the transfer using the non-linear interpolation between departure and target orbits as explained in section 2.5. The output includes the whole geometry and is directly passed to the 'Control law' block and to the 'Natural perturbation' one.
- **Natural perturbations:** This block takes as inputs the trajectory and computes the acceleration due to the natural perturbations (J2 effect). The output, consisting of the acceleration, is sent to the 'Control Law' Block. This block is not fundamental, and can be manually switched on if needed.
- **Control Law:** This block takes as input the geometry of the transfer and the natural perturbation acceleration. The control law, as well as the time vector are computed via a mixed numerical-analytical procedure. The block is identical to the one described in the previous section, and all considerations proposed are still valid for this case.

2.4 Reference frame and transfer angle definition

The selection of the Reference Frame for each transfer arc is fundamental in order to have easier boundary conditions for the shape. The selected RF can be seen in Figure 2.4 and it contains both the initial and final position vectors. The first step to define the RF is to select the reference plane: two different cases can be identified. If the initial and final position vectors are not aligned, the normal of this plane is defined as shown in equation 2.29. It can be seen that the selection of the reference frame is slightly different from the one suggested in [9] in order to include also polar and retrograde orbits.

$$\begin{cases} \hat{\mathbf{h}}_{REF} = \frac{\mathbf{r}_i \wedge \mathbf{r}_f}{|\mathbf{r}_i \wedge \mathbf{r}_f|} \qquad if \qquad \mathbf{r}_i \wedge \mathbf{r}_f \cdot \mathbf{h}_i > 0 \\ \hat{\mathbf{h}}_{REF} = -\frac{\mathbf{r}_i \wedge \mathbf{r}_f}{|\mathbf{r}_i \wedge \mathbf{r}_f|} \qquad if \qquad \mathbf{r}_i \wedge \mathbf{r}_f \cdot \mathbf{h}_i < 0 \end{cases} \tag{2.29}$$

If the initial and final position vector are aligned, the previous definition makes no sense since the magnitude of their cross product is zero. In this case the solution is not unique and, in theory, it doesn't affect the solution while in real implementation numerical errors make some choices inconvenient. A good solution was found through the selection of a reference plane which normal is the normalized mean of the initial and final one, as shown in equation 2.30

$$\hat{\mathbf{h}}_{REF} = \frac{\mathbf{h}_i + \mathbf{h}_f}{|\mathbf{h}_i + \mathbf{h}_f|} \tag{2.30}$$

The first axis of the reference frame is coincident with the initial position vector while the third is coincident with the reference plane normal $\hat{\mathbf{h}}_{REF}$. The second axis completes the right-handed orthonormal set. The developed shape gives bad results if the transfer angle is lower than π; moreover, if this value is near zero, it can give complex results in equation 2.15. Since one of the requirements is to have no singularities, a further revolution is considered if the transfer angle is lower than π in order to have ψ always in the domain $[\pi;3\pi]$ as shown in equation 2.31.

$$\begin{cases} \psi = 2\pi + \arccos\left(\hat{\mathbf{r}}_i \cdot \hat{\mathbf{r}}_f\right) & if \quad \mathbf{r}_i \wedge \mathbf{r}_f \cdot \hat{\mathbf{h}}_{REF} > 0 \\ \psi = 2\pi - \arccos\left(\hat{\mathbf{r}}_i \cdot \hat{\mathbf{r}}_f\right) & if \quad \mathbf{r}_i \wedge \mathbf{r}_f \cdot \hat{\mathbf{h}}_{REF} < 0 \end{cases} \tag{2.31}$$

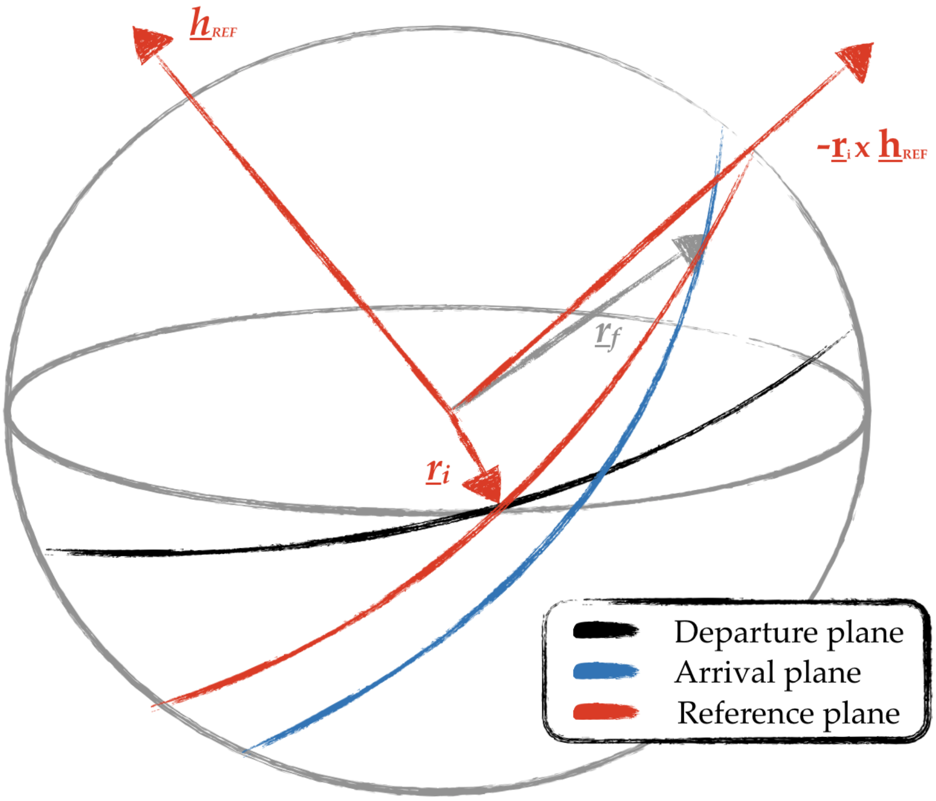


FIGURE 2.4: RF definition

It is important to remark that selecting this reference frame the declination of the initial and final positions are zero producing easier boundary conditions (see section 2.3.1) that can be imposed directly to the interpolating function. This can be done outside the algorithm and requires only one evaluation for all transfers.

2.5 Trajectory computation

In section 2.2 it was supposed to have an analitycal parametrization for the in-plane radius and out-of-plane displacement, as shown in equation 2.32

$$\begin{cases} r = r(x) \\ z = z(x) \end{cases} \tag{2.32}$$

The aim of this section is to define this parametrization and to derive all the geometrical quantities needed for the other blocks.

Differently from most of the literature [9] [42] [3] in which meaningless functions are used for the parametrization, here these quantities are selected in order to make the solution closer to the real dynamics: the idea is that if the thrust is low, the transfer path will be similar to the initial orbit at the beginning. The more the spacecraft approach the target, the closer will be its trajectory to the final orbit.

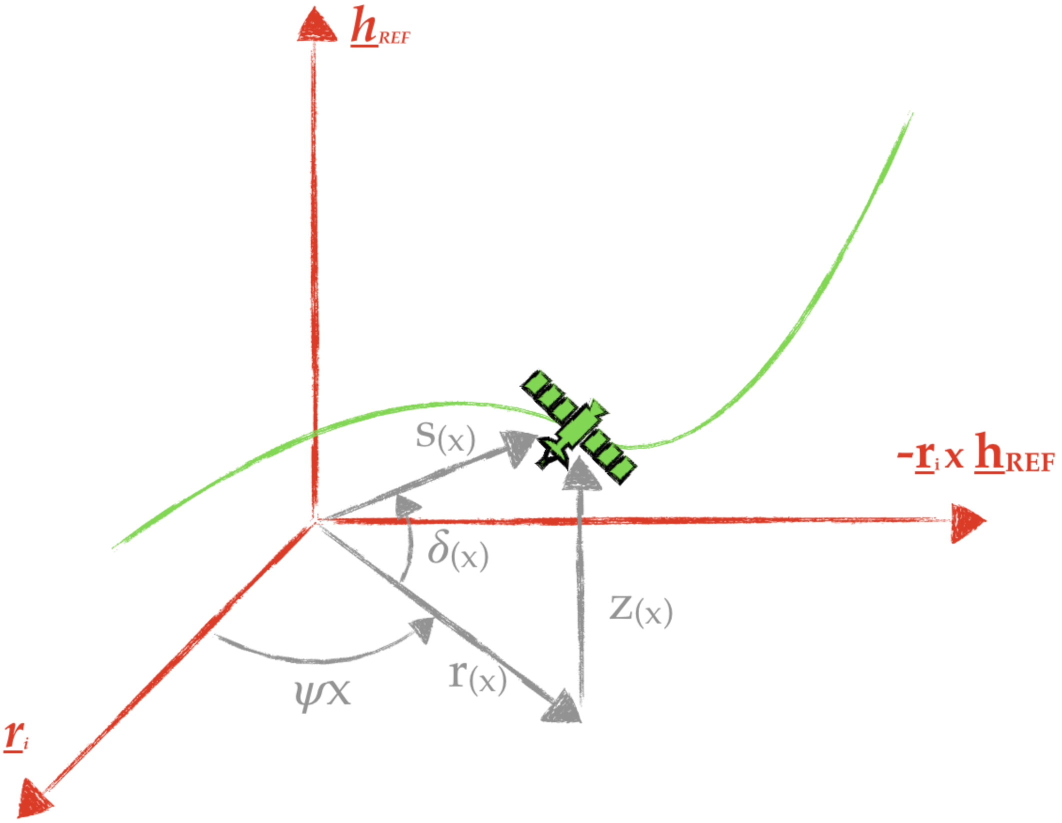


FIGURE 2.5: Spacecraft position in cylindrical and spherical coordinates

In order to traduce this idea into equations, r and z are expressed as function of the declination over the referencer plane and the attractor distance, as shown in figure 2.5, using equation 2.33: indeed, as will be shown in this section, it is quite easy to find a meaningful analytical description of the shape in spherical coordinates [15].

$$\begin{cases} r = s\cos\delta \\ z = s\sin\delta \end{cases} \tag{2.33}$$

In section 2.2 appear also some derivatives of these quantities: they can be computed with equation 2.34 and equation 2.35

$$\begin{cases} r' = -s\delta' \sin\delta + s' \cos\delta \\ r'' = -\left(2s'\delta' + s\delta''\right) \sin\delta + \left(s'' - s\delta'^2\right) \cos\delta \\ r''' = -\left[3\left(s''\delta' + s'\delta''\right) + s\left(\delta''' - \delta'^3\right)\right] \sin\delta + \left[s''' - 3\delta'\left(s'\delta' + s\delta''\right)\right] \cos\delta \end{cases} \tag{2.34}$$

$$\begin{cases} z' = s\delta' \cos\delta + s' \sin\delta \\ z'' = \left(2s'\delta' + s\delta''\right) \cos\delta + \left(s'' - s\delta'^2\right) \sin\delta \end{cases} \tag{2.35}$$

The declination δ and the attractor distance s are then parametrized using a non-linear interpolation between departure and arrival orbits, as shown in equation 2.36 and equation 2.37

$$\delta_{(x)} = \left(\delta_{2(x)} - \delta_{1(x)}\right) \chi_{(x)} + \delta_{1(x)} = \Delta\delta_{(x)}\chi_{(x)} + \delta_{1(x)} \tag{2.36}$$

$$s_{(x)} = \left(s_{2(x)} - s_{1(x)}\right) \chi_{(x)} + s_{1(x)} = \Delta s_{(x)}\chi_{(x)} + s_{1(x)} \tag{2.37}$$

from which is possible to compute the derivatives with respect to x needed to compute the position variation, as shown in equation 2.38 for the declination and equation 2.39 for the attractor distance

$$\begin{cases} \delta' = \Delta\delta'\chi + \chi'\Delta\delta + \delta'_1 \\ \delta'' = \Delta\delta''\chi + \chi''\Delta\delta + 2\Delta\delta'\chi' + \delta''_1 \\ \delta''' = \Delta\delta'''\chi + \chi'''\Delta\delta + 3\Delta\delta''\chi' + 3\Delta\delta'\chi'' + \delta'''_1 \end{cases} \tag{2.38}$$

$$\begin{cases} s' = \Delta s'\chi + \chi'\Delta s + s'_1 \\ s'' = \Delta s''\chi + \chi''\Delta s + 2\Delta s'\chi' + s''_1 \\ s''' = \Delta s'''\chi + \chi'''\Delta s + 3\Delta s''\chi' + 3\Delta s'\chi'' + s'''_1 \end{cases} \tag{2.39}$$

As can be seen from the previous equations, this block needs as input also declinations and attractor distances of the initial and final orbits.

2.6 Initial and final orbit analysis

In this section attractor distances along the departure and arrival orbits as well as their declinations over the reference plane are derived as function of the non-dimensional variable x. These quantities are necessary to compute the shape of the trajectory (see equation 2.36 and equation 2.37) as well as their derivatives till the order shown in equation 2.38 and equation 2.39.

2.6.1 Declinations over the reference plane

The declinations over the reference plane of the departure and arrival orbits can be computed as function of x considering the projection of these orbits on the celestial sphere, as shown in figure 2.6.

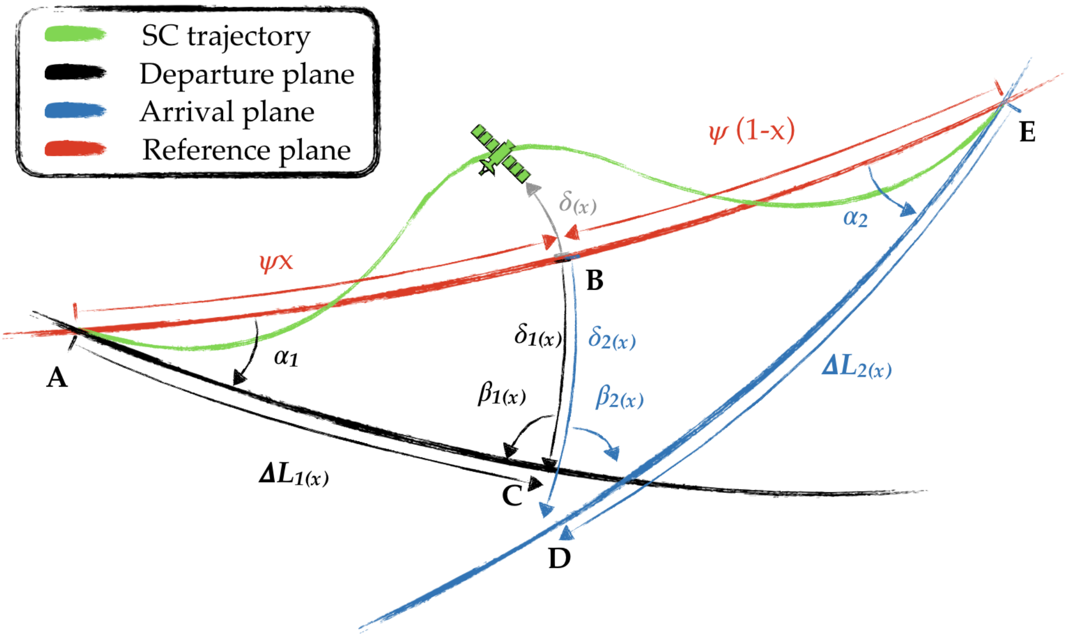


FIGURE 2.6: angular motion

Departure orbit

Let's start considering the spherical triangle ABC shown in figure 2.6. The cosine of the angle α_1 is equal to the angle between the normals of the initial and reference planes and can be computed with equation 2.40

$$\cos\alpha_1 = \hat{\mathbf{h}}_1 \cdot \hat{\mathbf{h}}_{REF} \tag{2.40}$$

For the sine of the same angle it is necessary to determine if A is the ascending or descending node with equation 2.42 and then compute it with equation 2.41 [14]. In this way the declination will have the correct sign with no necessity to split the two cases, reducing as results the weight of the algorithm.

$$\sin\alpha_1 = \xi_1\sqrt{1-\cos{\alpha_1}^2} \tag{2.41}$$

$$\begin{cases} \xi_1 = 1 \qquad if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} > 0 \\ \xi_1 = -1 \qquad if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} < 0 \end{cases} \tag{2.42}$$

The declination δ_1 can be computed as function of the non-dimensional anomaly x applying the sine theorem at the spherical triangle ABC, keeping into account that the angle in $\hat{B}$ is equal to $\frac{\pi}{2}$ due to the definition of declination [6].

$$\sin\delta_1 = \sin\alpha_1 \frac{\sin(\psi x)}{\sin\beta_1} \tag{2.43}$$

Since in equation 2.34 and equation 2.35 appear also the first, second and third derivatives of this quantity. it is necessary to derive equation 2.43 three time, as shown in equation 2.44, equation 2.45 and 2.46.

$$\delta_1' = \frac{\psi \sin\alpha_1 \cos\left(\psi x\right) - \beta_1' \cos\beta_1 \sin\delta_1}{\cos\delta_1 \sin\beta_1} \tag{2.44}$$

$$\begin{split}\delta_1'' = \frac{-\psi^2 \sin\alpha_1 \sin\left(\psi x\right) + \sin\beta_1 \sin\delta_1 \left(\delta_1'^2 + \beta_1'^2\right)}{\cos\delta_1 \sin\beta_1} + \\ - \frac{2\delta_1'\beta_1' \cos\delta_1 \cos\beta_1 + \beta_1'' \sin\delta_1 \cos\beta_1}{\cos\delta_1 \sin\beta_1}\end{split} \tag{2.45}$$

$$\begin{split}\delta_1''' = \frac{-\psi^3 \sin\alpha_1 \cos\left(\psi x\right) + \sin\beta_1 \sin\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right)}{\cos\delta_1 \sin\beta_1} + \\ + \frac{-\cos\beta_1 \cos\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right) + \cos\beta_1 \sin\delta_1 \left(3\delta_1'^2\beta_1' + \beta_1'^3 - \beta_1'''\right)}{\cos\delta_1 \sin\beta_1} + \\ + \frac{\sin\beta_1 \cos\delta_1 \left(3\beta_1'^2\delta_1' + \delta_1'^3\right)}{\cos\delta_1 \sin\beta_1}\end{split} \tag{2.46}$$

As can be seen in the previous equations it is necessary to compute also the angle β_1 and its derivatives. It can be done applying the cosine theorem to the spherical triangle ABC , as shown in equation 2.47

$$\beta_1 = \arccos\left(\sin\alpha_1 \cos(\psi x)\right) \tag{2.47}$$

The last equation can be derived three times giving as results equation 2.48, equation 2.49 and equation 2.50.

$$\beta_1' = \psi \sin\alpha_1 \frac{\sin\left(\psi x\right)}{\sin\beta_1} \tag{2.48}$$

$$\beta_1'' = \frac{\psi^2 \sin\alpha_1 \cos\left(\psi x\right) - \cos\beta_1 \beta_1'^2}{\sin\beta_1} \tag{2.49}$$

$$\beta_1''' = \frac{-\psi^3 \sin\alpha_1 \sin\left(\psi x\right) - 3\beta_1'\beta_1'' \cos\beta_1 + \beta_1'^3 \sin\beta_1}{\sin\beta_1} \tag{2.50}$$

Since we will need also the angle ΔL_1(see figure 2.6) for the determination of the distance from attractor, the sine and cosine spherical theorem can be applied to the spherical triangle ABC as show in equation 2.51

$$\begin{cases} \sin\left(\Delta L_1\right) = \frac{1}{\sin\alpha_1} \sin\delta_1 \\ \cos\left(\Delta L_1\right) = \cos\left(\psi x\right) \cos\delta_1 \end{cases} \tag{2.51}$$

The previous equation can be derived in order to compute the first, second and third derivatives of the angle ΔL_1 as shown in equation 2.52, equation 2.53 and equation 2.54.

$$\Delta L_1' = \frac{\delta_1'}{\sin\alpha_1 \cos(\psi x)} \tag{2.52}$$

$$\Delta L_1'' = \frac{\delta_1'' + \psi \sin\alpha_1 \sin(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \tag{2.53}$$

$$\Delta L_1''' = \frac{\delta_1''' + 2\psi \sin\alpha_1 \sin(\psi x) \Delta L_1'' + \psi^2 \sin\alpha_1 \cos(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \tag{2.54}$$

These quantities will be fundamental for the attractor distance determination, as will be shown later.

Target orbit

Considering now the triangle BDE in figure 2.6 it is possible to compute the cosine of the angle α_2 with equation 2.55.

$$\cos\alpha_2 = \hat{\mathbf{h}}_2 \cdot \hat{\mathbf{h}}_{REF} \tag{2.55}$$

Also in this case, in order to correctly compute the sine of the angle α_2, it is necessary to determine if the node E is ascending or descending: It can be done using equation 2.57. The sine of the angle α_2 can be computed as shown in equation 2.56;

$$\sin\alpha_2 = \xi_2 \sqrt{1 - \cos\alpha_2{}^2} \tag{2.56}$$

$$\begin{cases} \xi_2 = 1 & if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} < 0 \\ \xi_2 = -1 & if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} > 0 \end{cases} \tag{2.57}$$

The declination δ_2 can be computed as function of the non-dimensional anomaly x applying the sine theorem at the spherical triangle EBD keeping into account that the angle $\hat{B}$ is equal to $\frac{\pi}{2}$ for construction.

$$\sin\delta_2 = \sin\alpha_2 \frac{\sin(\psi(1-x))}{\sin\beta_2} \tag{2.58}$$

As for the departure orbit analysis also in this case it is necessary to compute the first, second and third derivatives of the declination δ_2 as shown in equation 2.59, equation 2.60 and equation 2.61.

$$\delta_2' = \frac{-\psi \sin\alpha_2 \cos(\psi(1-x)) - \beta_2' \cos\beta_2 \sin\delta_2}{\cos\delta_2 \sin\beta_2} \tag{2.59}$$

$$\delta_2'' = \frac{-\psi^2 \sin\alpha_2 \sin\left(\psi(1-x)\right) + \sin\beta_2 \sin\delta_2 \left(\delta_2'^2 + \beta_2'^2\right)}{\cos\delta_2 \sin\beta_2} + \\ - \frac{2\delta_2'\beta_2' \cos\delta_2 \cos\beta_2 + \beta_2'' \sin\delta_2 \cos\beta_2}{\cos\delta_2 \sin\beta_2} \tag{2.60}$$

$$\delta_2''' = \frac{\psi^3 \sin\alpha_2 \cos\left(\psi(1-x) + \sin\beta_2 \sin\delta_2 \left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right)\right.}{\cos\delta_2 \sin\beta_2} + \\ + \frac{-\cos\beta_2 \cos\delta_2 \left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right) + \cos\beta_2 \sin\delta_2 \left(3\delta_2'^2\beta_2' + \beta_2'^3 - \beta_2'''\right)}{\cos\delta_2 \sin\beta_2} + \\ + \frac{\sin\beta_2 \cos\delta_2 \left(3\beta_2'^2\delta_2' + \delta_2'^3\right)}{\cos\delta_2 \sin\beta_2} \tag{2.61}$$

The angle β_2 and its derivatives present in the previous equations can be computed by means of the cosine theorem applied at the spherical triangle EBD, as shown in equation 2.62. The derivation of this equation leads to equation 2.63, equation 2.64 and equation 2.65

$$\beta_2 = \arccos\left(\sin\alpha_2 \cos(\psi(1-x))\right) \tag{2.62}$$

The last equation can be derived three times giving as results equation 2.63, equation 2.64 and equation 2.65.

$$\beta_2' = -\psi \sin\alpha_2 \frac{\sin\left(\psi(1-x)\right)}{\sin\beta_2} \tag{2.63}$$

$$\beta_2'' = \frac{-\psi^2 \sin\alpha_2 \cos\left(\psi(1-x)\right) - \cos\beta_2 \beta_2'^2}{\sin\beta_2} \tag{2.64}$$

$$\beta_2''' = \frac{\psi^3 \sin\alpha_2 \sin\left(\psi(1-x)\right) - 3\beta_2'\beta_2'' \cos\beta_2 + \beta_2'^3 \sin\beta_2}{\sin\beta_2} \tag{2.65}$$

Since we will need also the angle ΔL_2(see figure 2.6) for the determination of the distance from attractor the sine and cosine spherical theorem can be applied to the spherical triangle EBD as show in equation 2.66

$$\begin{cases} \sin\left(\Delta L_2\right) = \frac{1}{\sin\alpha_2} \sin\delta_2 \\ \cos\left(\Delta L_2\right) = \cos\left(\psi(1-x)\right) \cos\delta_2 \end{cases} \tag{2.66}$$

The previous equation can be derived in order to compute the first, second and third derivatives of the angle ΔL_2 as shown in equation 2.67, equation 2.68 and equation 2.69. These quantities will be fundamental for the attractor distance determination, as will be shown later.

$$\Delta L_2' = \frac{\delta_2'}{\sin\alpha_2 \cos\left(\psi(1-x)\right)} \tag{2.67}$$

$$\Delta L_2'' = \frac{\delta_2'' - \psi \sin\alpha_2 \sin(\psi(1-x))\Delta L_2'}{\sin\alpha_2 \cos(\psi(1-x))} \tag{2.68}$$

$$\Delta L_2''' = \frac{\delta_2''' + 2\psi \sin\alpha_2 \sin(\psi(1-x))\Delta L_2'' + \psi^2 \sin\alpha_2 \cos(\psi(1-x))\Delta L_2'}{\sin\alpha_2 \cos(\psi(1-x))} \tag{2.69}$$

2.6.2 Attractor distances

In order to compute the shape of the trajectory, it is necessary to compute the distance from the attractor as function of x for both departure and arrival orbits. The selection of Modified Equinoctial Elements[10] makes the analytical computation of the desired quantities easier, therefore it is adopted in this phase. The first step is to determine the longitude (intended as the sixth Modified Equinoctial Element) along the departure orbit as function of x, as shown in equation 2.70 (see also figure 2.6) together with its variation, in equation 2.71. L_1 is the initial longitude of the satellite on the departure orbit.

$$l_1(x) = L_1 + \Delta L_1(x) \tag{2.70}$$

$$l_1'(x) = \Delta L_1'(x) \tag{2.71}$$

Now it is easy, using the angle transformation formulas, to compute the sine and the cosine of the longitude as function of x with equation 2.72.

$$\begin{cases} \sin l_1(x) = \sin(L_1)\cos(\Delta L_1(x)) + \cos(L_1)\sin(\Delta L_1(x)) \\ \cos l_1(x) = \cos(L_1)\cos(\Delta L_1(x)) - \sin(L_1)\sin(\Delta L_1(x)) \end{cases} \tag{2.72}$$

In a similar way, the same quantities are computed also for the target orbit, as shown in equation 2.73, equation 2.74 and equation 2.75. L_2 is the final longitude of the satellite on the arrival orbit.

$$l_2(x) = L_2 - \Delta L_2(x) \tag{2.73}$$

$$l_2'(x) = -\Delta L_2'(x) \tag{2.74}$$

$$\begin{cases} \sin l_2(x) = \sin(L_2)\cos(\Delta L_2(x)) - \cos(L_2)\sin(\Delta L_2(x)) \\ \cos l_2(x) = \cos(L_2)\cos(\Delta L_2(x)) + \sin(L_2)\sin(\Delta L_2(x)) \end{cases} \tag{2.75}$$

From now, the formulation is identical for the departure and arrival orbit and therefore it will be shown with the generic index i. The attractor distance can be computed using equation 2.76, in which q_i is defined as shown in equation 2.77

$$s_i(x) = \frac{p_i}{q_i(x)} \tag{2.76}$$

$$q_i(x) = 1 + f_i \cos l_i(x) + g_i \sin l_i(x) \tag{2.77}$$

As shown in section 2.2, the first, second and third derivatives of s are needed to solve the problem. These quantities are computed respectively in equation 2.78, equation 2.79 and equation 2.80

$$s_i(x)' = -\frac{p_i q_i'}{q_i^2} \tag{2.78}$$

$$s_i(x)'' = 2\frac{p_i q_i'^2}{q_i^3} - \frac{p_i q_i''}{q_i^2} \tag{2.79}$$

$$s_i(x)''' = -6\frac{p_i q_i'^3}{q_i^4} + 6\frac{p_i q_i' q_i''}{q_i^3} - \frac{p_i q_i'''}{q_i^2} \tag{2.80}$$

The derivatives of q that appear in the lasts equation can be computed using equation 2.81, equation 2.82 and equation 2.83.

$$q_i(x)' = (-f_i \sin l_i + g_i \cos l_i)\,\Delta L_i' \tag{2.81}$$

$$q_i(x)'' = (1 - q_i)\Delta L_i'^2 + q_i' \frac{\Delta L_i''}{\Delta L_i'^2} \tag{2.82}$$

$$q_i(x)''' = -q_i' \Delta L_i'^2 + 3(1 - q_i)\Delta L_i' \Delta L_i'' + q_i' \frac{\Delta L_i'''}{\Delta L_i'} \tag{2.83}$$

Despite the high number of equations to be evaluated, the algorithm is very fast. Indeed, the presence of repeated terms allows a strong reduction of the numbers of required FLOPS, by computing them only one time. Moreover, the vector structure of each variable allows an efficient vectorization of the algorithm: an higher number of evaluations are possible with only one set of instructions.

Chapter 3

Numerical validation and performance analysis

In this chapter some simple examples are proposed in order to crosscheck the resulting trajectory and to test the performance of the algorithm, both for imposed or free TOF. All examples are performed with Matlab R2017b on a laptop with a sixth generation Intel i7 processor working at 2.6 GHz. For the examples of this chapter no parallel computation is performed, since one of their aims is to test the speed of the algorithm. The crosschecking of the trajectory was performed integrating the equation of motion 3.1 using the control law computed with the shape based approach, eventually corrected with perturbation effects. The integration scheme selected is a simple but effective explicit Runge-Kutta of the fourth order (RK4) [1][14]. The system of ordinary differential equations is integrated over the anomaly instead of the time since the first is for construction a linearly spaced vector.

$$\begin{cases} \frac{dr}{d\theta} = \frac{v_r}{\omega} \\ \frac{dz}{d\theta} = \frac{v_z}{\omega} \\ \frac{dv_r}{d\theta} = \frac{1}{\omega}\left(-\frac{\mu}{\sqrt{(r^2+z^2)^3}}r + r\omega^2 + \frac{T\cos(\beta)\sin(\alpha)}{m}\right) \\ \frac{d\omega}{d\theta} = \frac{1}{r\omega}\left(\frac{T\cos(\beta)\cos(\alpha)}{m} - 2\dot{r}\omega\right) \\ \frac{dv_z}{d\theta} = \frac{1}{\omega}\left(-\frac{\mu}{\sqrt{(r^2+z^2)^3}}z + \frac{T\sin(\beta)}{m}\frac{1}{\omega}\right) \\ \frac{dt}{d\theta} = \frac{1}{\omega} \\ \frac{dm}{d\theta} = -\frac{T}{I_s g_0}\frac{1}{\omega}; \end{cases} \tag{3.1}$$

All the analytical derivatives shown in chapter 2 have been checked and compared with numerical derivatives using an high order finite difference scheme (6^{th} order central finite difference) The specification of the spacecraft selected for these example are listened in table 3.1.

TABLE 3.1: Spacecraft specifications

Spacecraft feature	Value
Specific Impulse [s]	3800
Dry mass[kg]	1000

3.1 TOF free algorithm

In this section the TOF free algorithm is tested. As shown in chapter 2, the algorithm solves the full 3D equation of motion with no approximations, therefore it is quite general and can work for a wide range maneuvers. Anyway the typical application of TOF free algorithms is the multi-revolution topping in planeto-centric scenario: as will be explained in detail in chapter 4 those complex missions are thrated as consecutive simpler single revolution trajectories. Therefore, as first step, it is necessary to investigate the performance of the algorithm on simpler single revoluton missions. The objective of this section are manly two:

- **Verification:** the first purpose is to prove the formal correctness of the developed algorithm with numerical results regardless the efficency and the effectiveness of the output trajectory. In order to do that an unrealistic single revolution transfer between different orbits on different planes is implemented.

- **Performance analysis:** the second purpose is to test the performance of the algorithm in terms of computational time, convergence rate and costs for different single revolution trajectories. A sensitivity analysis is also performed in order to show the effectiveness of the developed shape algorithm for different realistic Geocentric trajectories.

3.1.1 Verification

In order to shown and verify the algorithm, an unrealistic mission scenarios is supposed around the Moon. The objective is to move a spacecraft between the two different positions on different orbits. It is important to underline that here no optimization was performed: initial and final anomaly have been selected randomly and the Time of Flight was left free. The decision to implement a technical unfeasible trajectory belongs both to the necessity to make plots and graphs more readable and to stress the algorithm. As can be seen from the initial and final Keplerian Elements in table 3.2, the transfer include a strong change of plane combined with an high energy increase. The eccentricity and the argument of pericenter are also changed.

TABLE 3.2: Departure and Arrival states

Keplerian Element	Initial State	Final State
Semi-major axis [DU]	4.5	7
Eccentricity [-]	0.2	0.1
Inclination [deg]	5	15
RAAN [deg]	0	10
Argument of pericenter [deg]	20	5
True Anomaly [deg]	10	350

The resulting trajectory is shown in figure 3.1 together with its numerical verification. The errors on the final position and velocity are null, since they are imposed a priori, while the relative error on the Time of Flight and the fuel mass using 100 computational nodes are respectively $2.3 \cdot 10^{-7}$ and $4.2 \cdot 10^{-5}$: a good approximation for almost all scenarios. If the output trajectories are used only for initial mission design or as guests for direct optimization processes they will be also more precise than necessary: a reduction of the number of nodes can be considered.

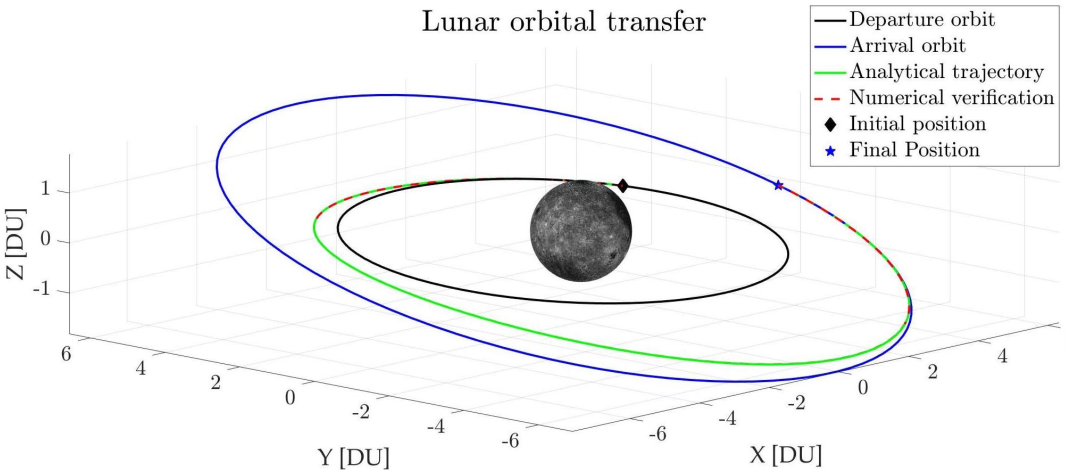


FIGURE 3.1: Spacecraft trajectory

The output control law and mass profile are reported in figure 3.2 as function of the angular displacement θ from the initial position vector.

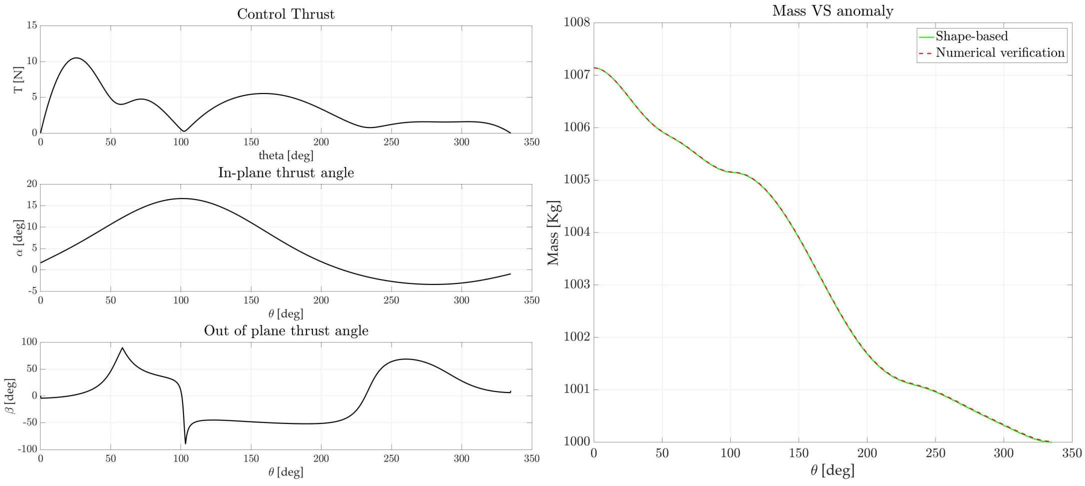


FIGURE 3.2: a) Control law b) Mass profile

In figure 3.1 and figure 3.2 it is possible to see also the perfect matching of the numerical crosscheck: the error between this solution and the analytical one tends to zero if the number of nodes increases. The strong change of plane is traduced in an elevate out of plane thrust angle for almost the whole trajectory. The required thrust is in around $10N$, well beyond technical capabilities [17]. Even if the trajectory is not feasible, it explains quite well the working principle of the algorithm.

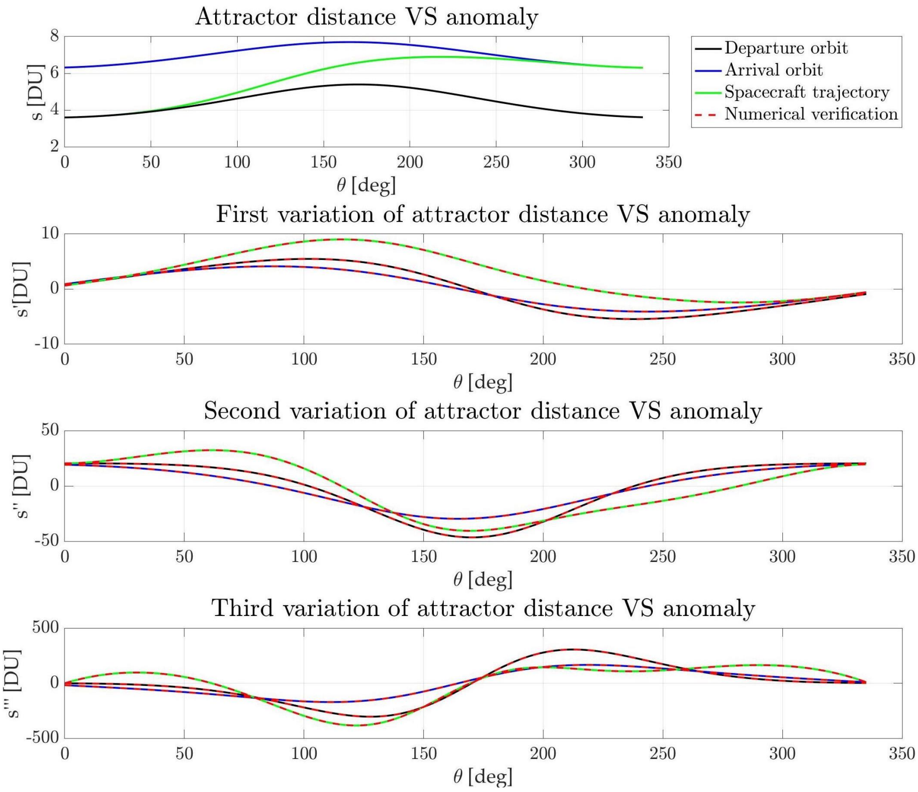


FIGURE 3.3: Attractor distance

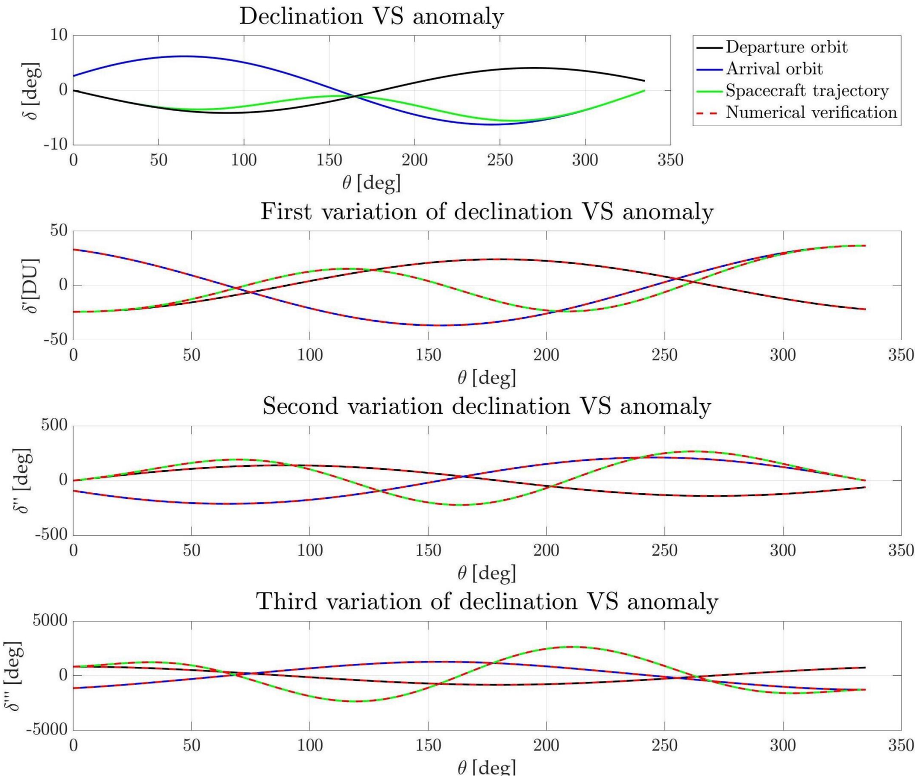


FIGURE 3.4: Declinations

In figure 3.3 and figure 3.4 the quantities from which all the kinematic and dynamics are recovered (see equation 2.37 and 2.36) are shown together their derivatives: also for these quantities the numerical verification matches exactly the analytical one.

It is easy to recognize that s and δ have been obtained via a non-linear interpolation of the corresponding quantities of the departure and arrival orbits. Moreover, being the derivative a linear operator also the derivatives of s and δ are non-linear combinations of the corresponding functions. Due to the requirements of the interpolating function identified in section 2.3 the trajectory matches exactly the geometry of the departure orbit at the starting point and of the arrival one at the ending point. For shake of completeness the osculating elements are reported in figure 3.5: the link between the spacecraft trajectory and the initial and final orbit is not so intuitive as happens in spherical or cylindrical coordinates.

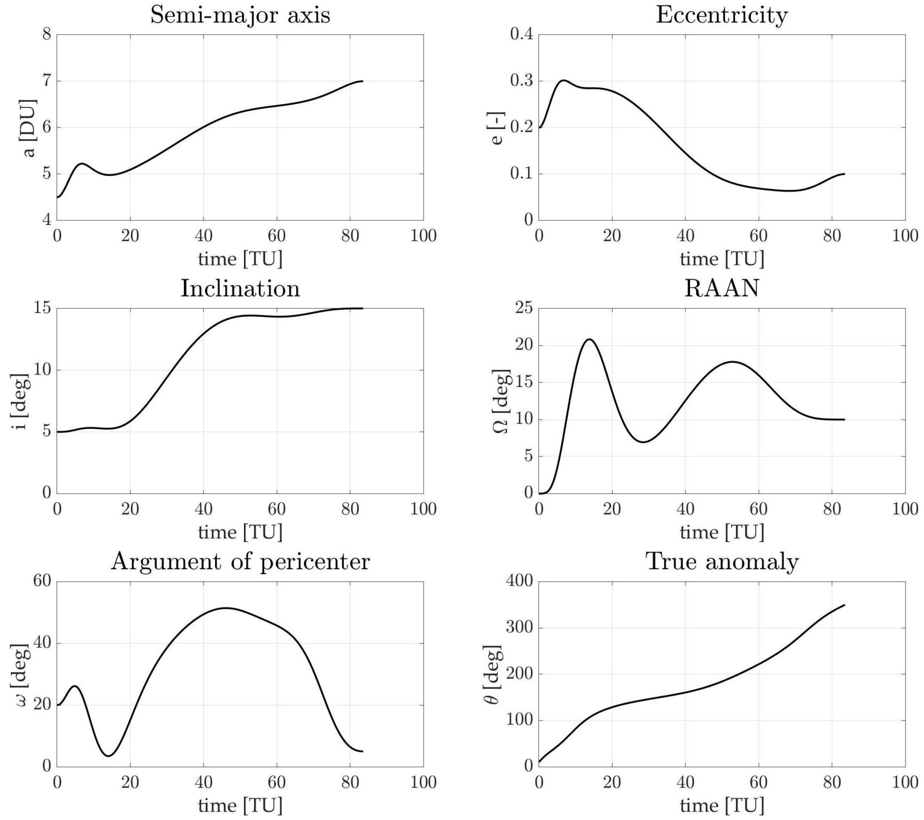


FIGURE 3.5: Osculating element

3.1.2 Performance analysis

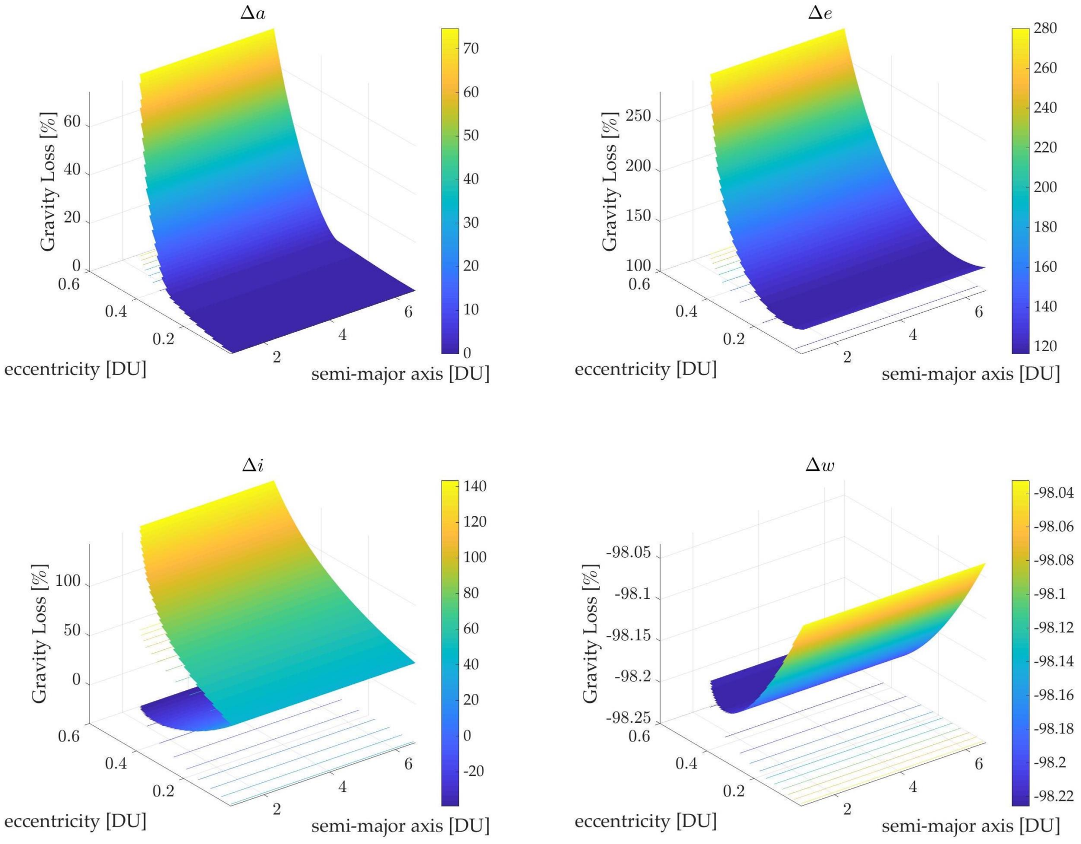
The performance analysis is carried out in two different steps: firstly, a sensitivity analysis is performed in order to tests the effectiveness of the algorithm if compared with impulsive maneuvers in a wide range of working conditions for Earth-centered scenarios. The second step consists in an analysis of computational effectiveness and convergence rate in different conditions.

Sensitivity cost analysis

One of the most important parameter of merit of a low thrust trajectory is the gravity loss factor: it shows how a specific maneuver is more expensive in therm of ΔV when compared to the corresponding impulsive one, as shown in equation 3.2.

$$G_{loss} = 100\left(\frac{\Delta V_{low}}{\Delta V_{impulsive}} - 1\right) \tag{3.2}$$

In figure 3.6 the gravity losses are reported for different maneuvers in different conditions of eccentricity and semi-major axis, while in figure 3.7 is reported the corresponding required thrust. All the low thrust maneuvers are performed in one revolution, from pericenter to pericenter.


Δa

Δe

Δi

Δw

FIGURE 3.6: Gravity Losses

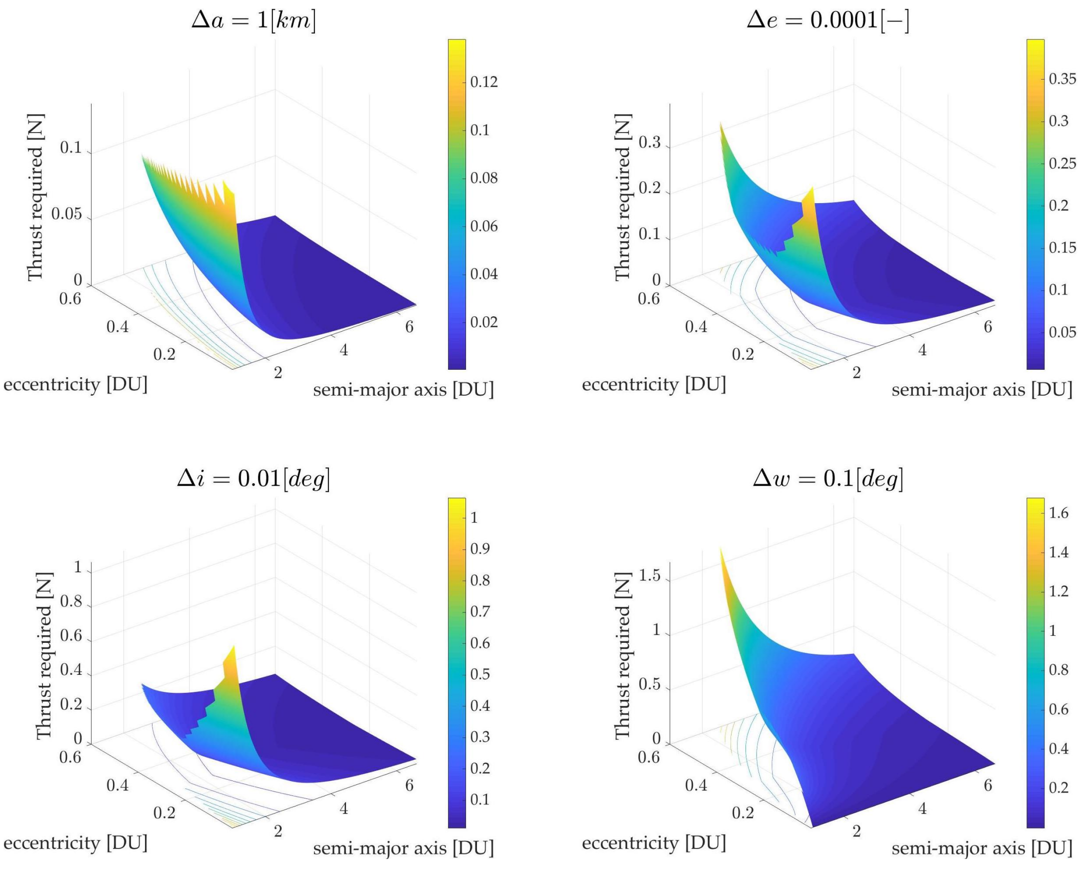


FIGURE 3.7: Thrust required

The domain of the sensitivity analysis covers all the Geocentric orbits with semi-major axis ranging from $1[DU]$ to $6.6108[DU]$ (GEO orbit) and eccentricity ranging from 0 to 0.6: it means that almost every Geocentric orbit of commercial interest is included in this analysis. Looking at figure 3.6, it is possible to see that the effectiveness of every maneuver is almost independent from the semi-major axis while it is more affected by the eccentricity, especially for large values (more than 0.3). Regarding the thrust required, as expected from theory, it decreases while the semi-major axis increases. More in detailed it is important to underline that:

- **Semi-major axis variation** $(\mathbf{\Delta a})$: The low thrust trajectory is here compared with the optimal elliptic Hohman transfer. It is interesting to show that the cost of low thrust trajectory converges to the Hohman one if the eccentricity goes to zero and the gravity losses remain below 10% for eccentricity values lower than 0.3. The required thrust strongly decreases as the semi-major axis grows, as expected from theory, for any values of the eccentricity.
- **Eccentricity variation** $(\mathbf{\Delta e})$: Also in this case the low thrust trajectory cost is compared with the optimal elliptic Hohman transfer. In this case the limitation of the developed algorithm and, more in general, of every low thrust optimization method that requires the continuity of the thrust is underlined.

Indeed the change of eccentricity is extremely ineffective with respect to the impulsive domain: more that the double of ΔV is required. This undesired behavior can be better understood looking at some in-literature available trajectories[21][2] optimized with direct methods in which the solution shows a typical bang-bang structure with the spacecraft thrusting only near the apsis. Any solution, in which the continuity of the thrust is imposed in advance, will be surely sub-optimal and, the higher is the eccentricity of the orbit the lower will be the effectiveness.

- **Change of plane** $(\mathbf{\Delta i})$: A direct and general comparison between low thrust and impulsive trajectories is not so easy as in the previous cases since the solution will be strongly affected by the specific plane of the orbit and the direction of the apsis line. In this work the best and the worst cases have been explored: any possible solution will be between them. The optimal solution for the impulsive case is when the maneuver takes place at the apocenter (upper surface is figure 3.6), while the worst cases is when the change of plane is performed at the pericenter (lower surface is figure 3.6). In figure 3.7 it is possible to see that the thrust required for the low thrust maneuver is affected by both the semi-major axis and the eccentricity. While the effect of the semi-major axis is the same as in the previous maneuvers, the higher is the eccentricity the lower is the thrust required, indeed the most demanding change in plane (see figure 3.7) takes place near the apocenter, that is farther from the attractor in high elliptical orbits than in low elliptical one.

- **In-plane apsis line rotation** $(\mathbf{\Delta}\ \omega)$): The rotation of the apsis line is extremely convenient if compared with the impulsive case, since it takes only a few percentage of the impulsive one. The thrust peak is strongly affected by the eccentricity and the semi-major axis of the orbit: the higher is the eccentricity, the higher is the thrust required.

From the previous considerations, it is clear that the algorithm gives results very close to the optimal one if low elliptical transfer with high energy changes and modest change in plane are considered. Mission scenarios such as deployment of multiple satellites in LEO or multi-objective GEO topping problem could benefit from this algorithm.

Convergence rate and CPU time

One of the biggest advantages of this algorithm, and in general of all shape-base algorithms, is the fastness. In figure 3.8 it is shown the CPU time as function of the number of computational nodes. For typical application, as will be further investigate in this chapter, between 15 thousands and 20 thousands revolution can be evaluated each second. The aforementioned graph shows a quasi bilinear trend: the discontinuity between 250 and 300 computational nodes belongs to the way the algorithm is implemented. As mentioned in chapter 2 the MATLAB implementation benefits to the vectorization of the algorithm: a single set of instruction works for all the nodes. This is true till the saturation of the memory on the processor itself, that probably happens in the above-mentioned range.

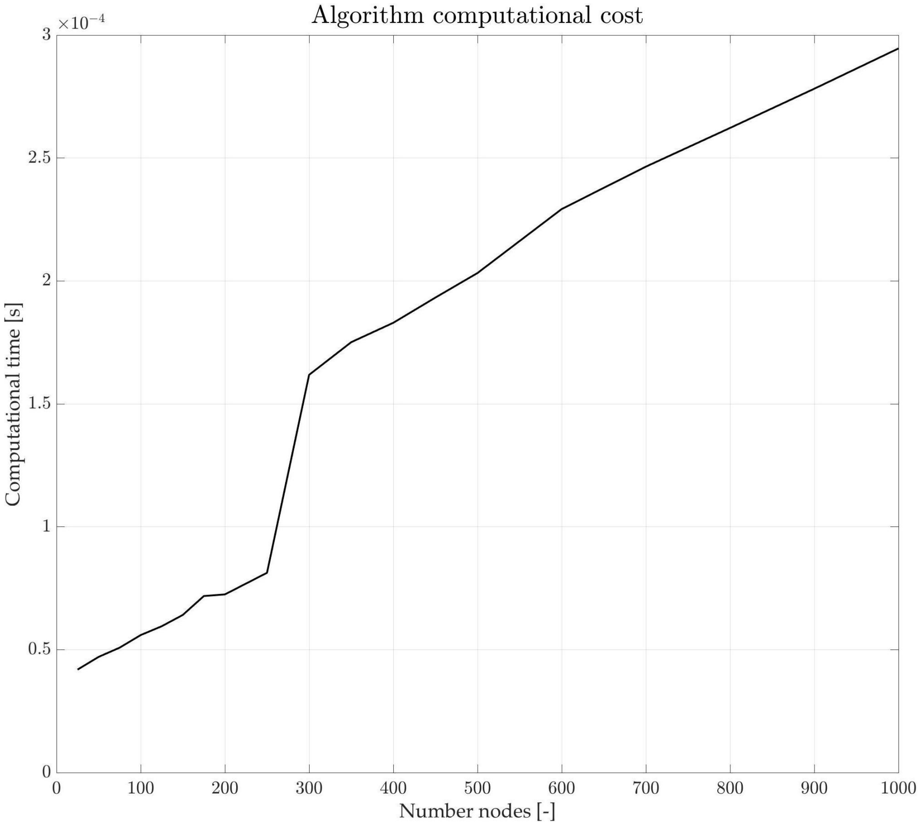


FIGURE 3.8: Computational Time

The error on final position and velocity is null independently from the number of computational nodes used, being these quantities computed with a full analytical approach. On the other hand, fuel mass and time of flight belong to numerical integration, and therefore a convergence analysis is needed in order to quickly estimate the error and the number of computational nodes required.

In figure 3.9 it is shown the trend of the ten-based logarithm of the relative errors of the fuel mass and time as function of the computational nodes and the orbital eccentricity. It is important to underline that the resultant relative errors are almost uncorrelated with the maximum thrust required if the thrust is low enough to allow a smooth variation of the osculating elements, as happens in realistic situations. For most common trajectories, with thrust equal or lower than one Newton and eccentricities not so high, between 50 and 100 computational nodes are sufficient to guarantee a relative error on the fuel mass around 10^{-5} and of 10^{-6} on the time of flight. Moreover, for fast and rough approximations only few tens of points are sufficient to contain the relative error on both quantities below 10^{-2}.

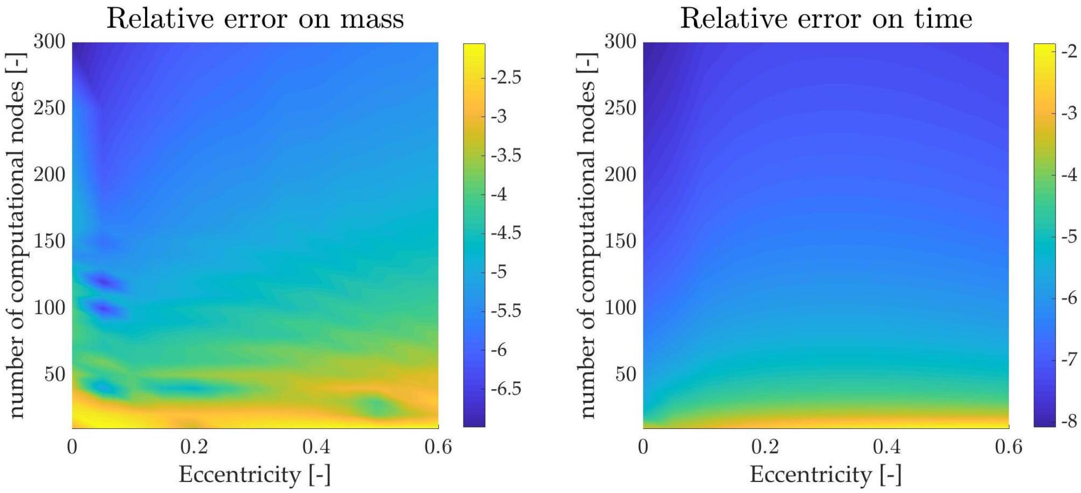


FIGURE 3.9: Relative error: logarithmic scale based on 10

The algorithm was also tested in non-realistic scenarios and, provided that the trajectory is feasible, it is always possible to reach a certain relative error increasing the number of computational nodes.

3.2 TOF imposed algorithm

The imposition of time of flight for the one revolution trajectory makes no sense in earth-centered scenarios since the dynamics is much more constrained with respect to interplanetary trajectories. The multi-revolution rendezvous in low thrust environment is an extremely complex task, and cannot be performed considering directly the time of flight on each arc as degrees of freedom, indeed thousands degrees of freedom should be used. Moreover the resulting domain of possible Time of flights for one revolution in planeto-centric problems is extremely tight and therefore it is almost useless to take it as degree of freedom. For these reasons, the performance analysis in this section will be performed with interplanetary trajectories. In this section, after a numerical verification of the algorithm with the same example presented in the previous section, a performance analysis will takes place in order firstly to investigate the sensitivity of the results with respect to the imposed TOF in different conditions and then to show its computational cost.

3.2.1 Numerical Verification

In order to test the algorithm the same Lunar example of the previous section is considered. The difference is that the time of flight is now imposed to the shape. It is considered as a degrees of freedom in a quasi newton optimization algorithm (MATLAB $fmincon$ function) having as objective the minimization of the fuel consumption. The resulting trajectory is shown in figure 3.10, while the control law and the mass profile are shown in figure 3.11. Also in this example 100 nodes have been used for the computation of the trajectory. The relative error on the fuel mass and on the time of flight are respectively $2 \cdot 10^{-4}$ and $4.3 \cdot 10^{-4}$. The growth of the error between free and imposed time of flight is due to the necessity to numerically solve the time equation (equation 2.28) as explained in chapter 2: the newton solver is initialized with $a_s = 0$ (this makes the interpolation function equals to the one used for the TOF free algorithm, guaranteeing the existence of the initial guess) and takes only five steps to converge with a residual lower to 10^{-6}. The above-mentioned error arises to the integral present in equation 2.28 that has to be numerically solved at each step using a Cavalieri-Simpson quadrature scheme. The adoption of an higher order integration scheme was tested, but the benefits in term of error reduction are not sufficient to justify the increase in CPU time.

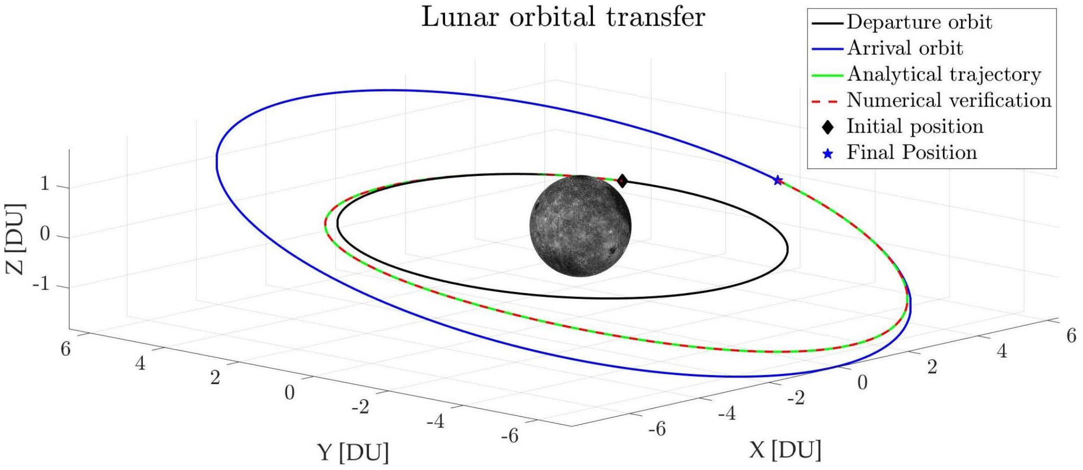


FIGURE 3.10: Spacecraft trajectory

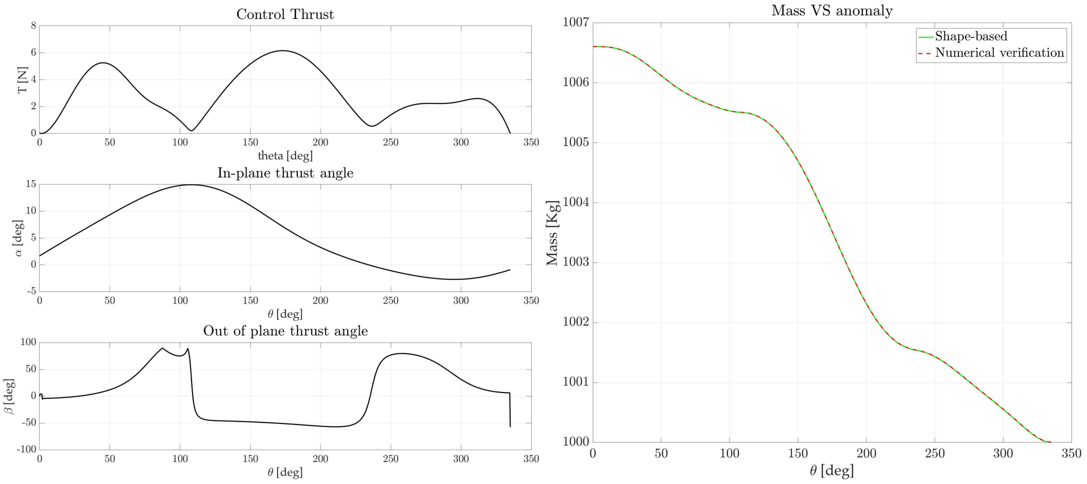


FIGURE 3.11: a) Control law b) Mass profile

As can be seen in table 3.3, the increased flexibility of the shape (see the shapes specifications in chapter 2 for further details) results in a trajectory that better fits the dynamics. The fuel mass required, as well as the maximum thrust needed and the time of flight are lower if compared with the solution obtained with free time of flight.

TABLE 3.3: Comparison between algorithms

Algorithm	TOF free	TOF imposed
Fuel Mass [kg]	7.15	6.60
Maximum thrust [N]	10.48	6.61
Time of Flight [hours]	24.02	23.05

This flexibility is well suited for multi-objective optimization of interplanetary trajectories.

3.2.2 Performances analysis

In this section a performance analysis is executed in order to test the sensitivity of the solution with respect to the time of flight, the convergence rate and the computational time required by the algorithm. Regarding the sensitivity analysis it is important to underline that, being the TOF free algorithm a particular solution of the TOF imposed one, the trends shown in the previous section are still valid and they can be seen as a suboptimal cases for this algorithm.

Sensitivity cost analysis

The sensitivity analysis is performed with a simple single revolution (it means a transfer angle between π and 3π, as explained in details in Chapter 2) Earth-Mars rendezvous, with null initial and final relative velocity and using the same spacecraft specification adopted in the previous examples, that are summarized in table 3.1. The analytical ephemerids adopted are the ones suggested by the NASA-JPL Solar System Dynamics team, available at [31]. The main mission parameter are summarized in table 3.4: the domain of the TOF for the trajectory is extremely wide for a single revolution transfer, while the departure date is kept fixed since the aim of this example is only to test the sensitivity and the convergence basin of the algorithm and not to find a global optimal solution.

TABLE 3.4: Mission parameters

Parameter	Value
Departure Date	11^{th} May 2022
TOF [days]	from 490 to 820

As can be seen from figure 3.12, the convergence radius of the algorithm is wide and the fuel mass and thrust optimal solutions are quite close. For both it is present a plateau in which the distance from optimal solution is low: as will be shown later, this fact allows an high flexibility when more complex scenarios are taken into account.

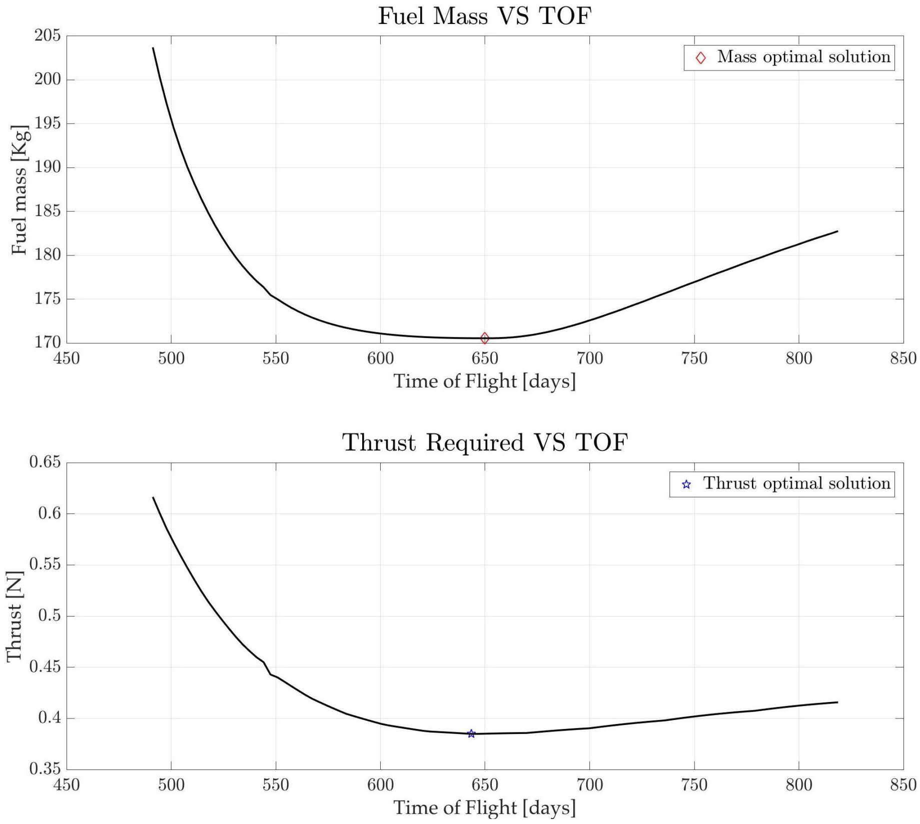


FIGURE 3.12: Sensitivity with respect to the TOF

Convergence rate and CPU time

As expected, the TOF imposed algorithm is slower if compared with the TOF free algorithm due to the necessity to numerically solve equation 2.28. In figure 3.13 is possible to see the computational cost of the algorithm: more than 8.5 thousands of revolutions can be computed each second if 100 computational nodes are adopted, a number that, as mentioned before, is capable to give relative errors on the mass and time of flight in the order of magnitude of 10^{-4}. Also in this case it is present a step in the computational time around 250 nodes: the reason is the same that of the TOF free algorithm, indeed the required memory is not affected by the additional steps required to solve equation 2.28.

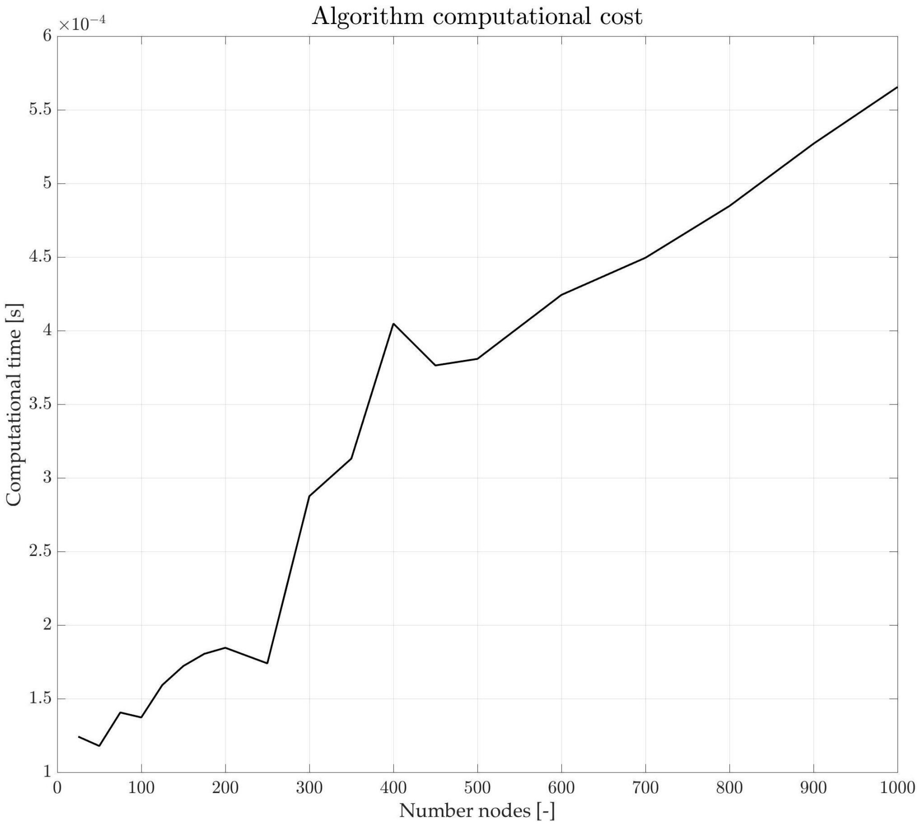


FIGURE 3.13: CPU time

Chapter 4

Use of the shape-based algorithms in multi-revolution scenarios

The algorithms presented in chapter 2 can link two different states in a single revolution (namely with a transfer angle between π and 3π, as explained in 2), anyway most of the trajectories, both in interplanetary and geocentric scenarios, involve more than one revolution. Typically in interplanetary trajectories few revolutions are sufficient to reach any possible target [9][28][30], while in planetary environments hundreds or thousands are needed. Moreover in interplanetary trajectories the time of flight is a fundamental constrain, since no phasing maneuver can be done due to the high orbital period. In planeto-centric environment the time of flight is not a constrain but an objective, indeed it could be in the order of months while phasing maneuvers could be easily performed in hours or days. For these reasons the two cases are threated in different sections: in the first section the more complicated planeto-centric algorithm is explained, while in the second two possible solution of the interplanetary problem, with different costs and benefits, are explored.

4.1 Use of the shape based algorithm in planeto-centric mission scenarios

In this section it is explained how the shape-based algorithm presented in chapter 2 can be successfully used to solve complex multi-revolution scenarios. The necessity of a dedicated section for this task arises to the fact that in planeto-centric environment hundreds or thousands of revolutions are needed to move the satellite from an orbit to another. The classical way [9][42] to deal with multi-revolution problems is to consider an 'augmented' transfer angle, as shown in equation 4.13. Unfortunately, this solution works quite well for a very low number of revolutions (typically less than three or four), as will be shown in chapter 5.

$$\psi = \psi + 2\pi N_{rev} \tag{4.1}$$

Moreover, this strategy implies the continuity of the thrust over the whole trajectory, making impossible to switch off thrusters during eclipses. For these reasons this solution will not be adopted for planeto-centric trajectories: in the following sections two different approaches with different requirements and peculiarities are suggested and analyzed. As underlined in chapter 3, the adoption of the imposed TOF algorithm on multi revolutions problems is not straightforward and it can't be directly applied if the number of revolutions exceeds some units. On the other hand, in chapter 2 it was proved that using the free TOF algorithm it is only possible to impose or the initial position and epoch, or the final one. It means that if it is needed to move a spacecraft between different orbits without the necessity to constrain both the exact position and time on the departure and final orbits, the TOF free algorithm can works. Instead, if it is necessary to link exactly two different time dependent positions without the possibility to perform a further phasing maneuver, a different strategy is needed. It is important to underline that, being the TOF free algorithm able to impose exactly the epoch (and the mass) or at the beginning or at the end, most of the practical problems, such as orbit injection and rising, can be solved with the first strategy.

4.1.1 Multi-revolution trajectories with free TOF

This algorithm allows to link two different states with an high number of revolutions imposing exact boundary conditions on the position and forcing or the initial or the final date and mass. The working principle is to generate a family of intermediate orbits placed in such a way as to make the thrust peaks constant, and in a sufficient number to not exceed the maximum available thrust. The full multi-revolution trajectory is then threated as series of easier one revolution trajectories that are solved with the TOF free algorithm presented in chapter 2. The departure and arrival anomalies are selected in order to thrust only in sunlight: in the k^{th} arc the starting point coincides with the exit from the eclipse on the k^{th} intermediate orbit, while the target point is the entrance of the eclipse on the $k^{th} + 1$. During the eclipses phases the motion is keplerian if the perturbation effects are switched-off, otherwise it is perturbed only by the natural forces. If an intermediate orbit doesn't experience any phase in shadow, one fictitious eclipse of null duration is introduced in order to maintain the same structure of the algorithm.

Algorithm description

Before introducing the architecture scheme it is important to formalize the problem: the initial and final states are imposed as Modified Equinoctial Elements (MEE), as shown in equation 4.2.

$$\begin{cases} MEE_i = [p_i, f_i, g_i, h_i, k_i, L_i] \\ MEE_f = [p_f, f_f, g_f, h_f, k_f, L_f] \end{cases} \tag{4.2}$$

Since the algorithm can work both forward and backward in time, two different but similar formulations are available: for sake of brevity only the forward algorithm is deeply analyzed. The steps involved are the following:

1. **initialization of the problem:** at the first step of the algorithm the spacecraft is at the initial state, while the desired is the final one, therefore equation 4.3 holds.

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_f \\ m_k(0) = M_{initial} \\ t_k(0) = 0 \end{cases} \tag{4.3}$$

   with $k = 1$ since the first trajectory has to be designed.

2. **Positioning the k-th intermediate orbit:** the spacecraft is in the position described by MEE_1 with mass $m_k(0)$ and the time after departure in $t_k(0)$. The objective is to place the k^{th} intermediate orbit in such a way that the maximum thrust required is equal to the maximum available one. Accordingly to equation 4.4 the position of the intermediate orbit depends on the value of the parameter η_k: the higher is η_k the higher will be the gap between the current orbit and the intermediate one and so the higher will be the thrust required.

$$\begin{cases} p_k = (p_2 - p_1)\,\eta_k + p_1 \\ f_k = (f_2 - f_1)\,\eta_k + f_1 \\ g_k = (g_2 - g_1)\,\eta_k + g_1 \\ h_k = (h_2 - h_1)\,\eta_k + h_1 \\ k_k = (k_2 - k_1)\,\eta_k + k_1 \end{cases} \tag{4.4}$$

   The exact position on the intermediate orbit, described by the sixth parameter L_k, is depending on the other five Modified Equinoctial Elements computed in equation 4.4 since, as said before, it must coincide with the entrance on the eclipse, no matter if it is real or fictitious. The model adopted for the eclipse is the standard cylindrical one. It is important to underline that, being the shadowed region function of the time due to the Earth motion around the sun, it has to be computed at each k^{th} step. The desired value of η_k is found solving numerically equation 4.5 in which the term $max\,(T(\eta_k))$ is the maximum thrust required during the k^{th} trajectory that can be computed using the TOF free algorithm presented in chapter 2.

$$max\,(T(\eta_k)) - T_{available} = 0 \tag{4.5}$$

The numerical solution of equation 4.5 in general is not straightforward since there are points in which its continuity is not guaranteed, as well as the existence of the solution itself. The adopted solution is an hybrid Newton-Bisection algorithm developed ad-hoc. As first step, the algorithm tries to solve the equation with Newton method, then if it fails, the algorithm tries again with the Bisection method. If also in this case the algorithm fails in reaching a predefined tolerance in a certain number of iterations, the equation is transformed into an inequality, as shown in equation 4.6.

$$max\left(T(\eta_k)\right) - T_{available} < 0 \tag{4.6}$$

The last inequality can always be solved since the developed shape-based algorithm fulfill the requirement in equation 4.7

$$\lim_{\Delta MEE \to 0} \max\left(\frac{|T|}{m}\right) = \lim_{\eta_k \to 0} \max\left(\frac{|T|}{m}\right) = 0 \tag{4.7}$$

If the value of η_k found with previous equations is equal or greater than one it means that the thrust available on-board is sufficient to reach the final position MEE_2, as can be seen from equation 4.4: in this case η_k is automatically switched to one and the k^{th} trajectory is computed again; the algorithm stops. Otherwise if η_k is between 0 and 1 it is necessary to prepare the states for the next step using equation 4.8: the new starting position is the exit from the eclipse of the k^{th} intermediate orbit, the initial mass of the spacecraft on the trajectory $k+1$ is exactly equal to the final mass of the trajectory k since during the eclipse no fuel is consumed. The time after departure at the beginning of the $k+1$ trajectory is equal to the arrival time of the trajectory k plus the time spent in shadow ($\Delta t_{eclipse}$)

$$\begin{cases} MEE_1 = MEE_k \\ MEE_2 = MEE_f \\ m_{k+1}(0) = m_k(1) \\ t_{k+1}(0) = t_k(1) + \Delta t_{eclipse} \end{cases} \tag{4.8}$$

The value of k is then increased and the algorithm goes back to the beginning of point 2. The cycle stops when η_k is equal or greater to one.

3. **Trajectory analysis**: For fastness purposes, the previous block gives as outputs only the initial/final mass and the time of flight. If more information are needed once the intermediate orbits are placed this block is activated and compute all the kinematics and dynamics quantities of the trajectory. The computational time required by the algorithm is in around 50% higher if this block takes place, therefore in the optimization of complex mission scenarios it is better to run it only on a limited number of solutions.

The backward version of the algorithm involves the same steps, with some important differences:

- Now the initial state is the arrival one, and the desired is the departure one, therefore equation 4.3 is substituted with equation 4.9.

$$\begin{cases} MEE_1 = MEE_f \\ MEE_2 = MEE_i \\ m_k(1) = M_{final} \\ t_k(1) = 0 \end{cases} \tag{4.9}$$

- The mass imposed in equation 4.9 is the final one instead of the initial one and the variable $t_k(x)$ now means 'time before the arrive' instead of 'time after departure' ant it is a negative quantity.

- The k^{th} intermediate orbit is integrated itself backward in time, therefore the update of the states becomes the one shown in equation 4.10 instead of the one in equation 4.8.

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_k \\ m_{k+1}(1) = m_k(0) \\ t_{k+1}(1) = t_k(0) - \Delta t_{eclipse} \end{cases} \tag{4.10}$$

  It means that the final mass to be imposed to the $k^{th} + 1$ single revolution trajectory is the initial one computed for the k^{th} revolution, as happens for the time.

The effect of having both the forward and backward direction for the trajectory design allows to keep into account more scenarios. The architecture of the algorithm is presented in figure 4.1 for completeness.

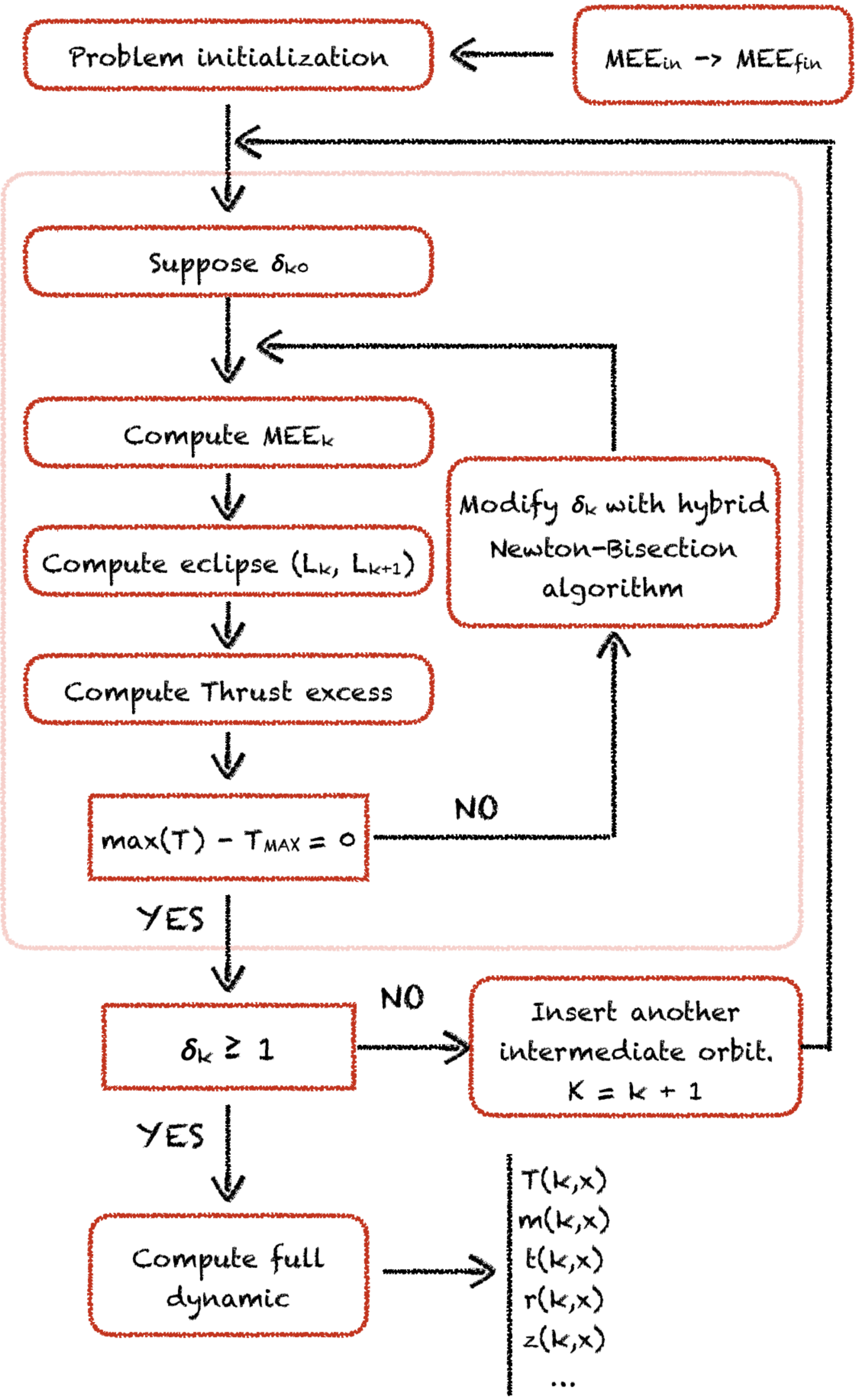


FIGURE 4.1: Multi-revolution algorithm architecture

Example of multi-revolution trajectory

In order to better understand the working principles and the steps involved in the multi-revolution algorithm proposed, a very simple planar example is here reported: The mission consists in the design of a trajectory able to link the equatorial and circular orbits indicated in table 4.1; the departure epoch is fixed at the 20 July 2020.

TABLE 4.1: Departure and Arrival states

Keplerian Element	Initial State	Final State
Semi-major axis [DU]	1.7	4
True Anomaly [deg]	0	free

The spacecraft has an initial wet mass of 800 kg, a maximum available thrust of 40 N and a specific impulse of 3800 s. The output trajectory is quite explicative and it is reported in figure 4.2.

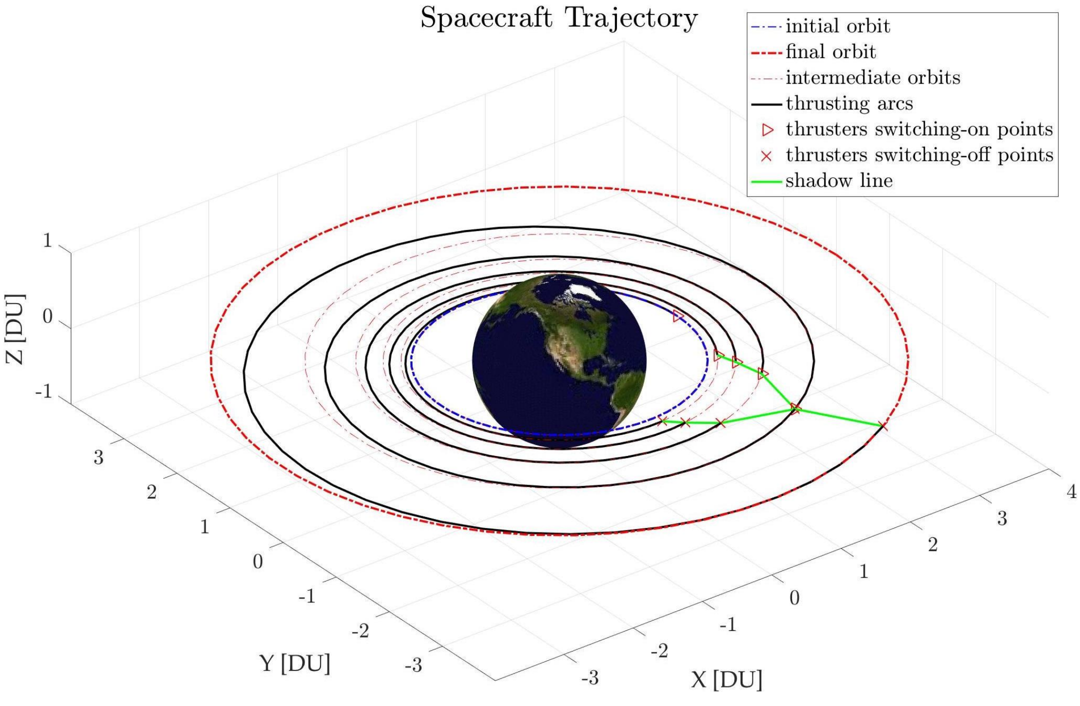


FIGURE 4.2: Spacecraft Trajectory

Since the initial mass is imposed the direction of the algorithm will be forward: the problem is solved from the initial orbit, finding firstly the position of the first intermediate orbit and then going on in the loop described in the previous section.

As can be seen from figure 4.3, the thrust peaks are all aligned at 40 N with the only exception of the last one, indeed, as mentioned in the theoretical description, in general the last revolution examined doesn't saturate the available thrust. The different angular length of the trusted arcs is obviously due to the presence of the eclipse that is different for each intermediate orbit. Also in the mass profile in figure 4.3 it is possible to see the effects of the eclipse phases in which no fuel in consumed. As

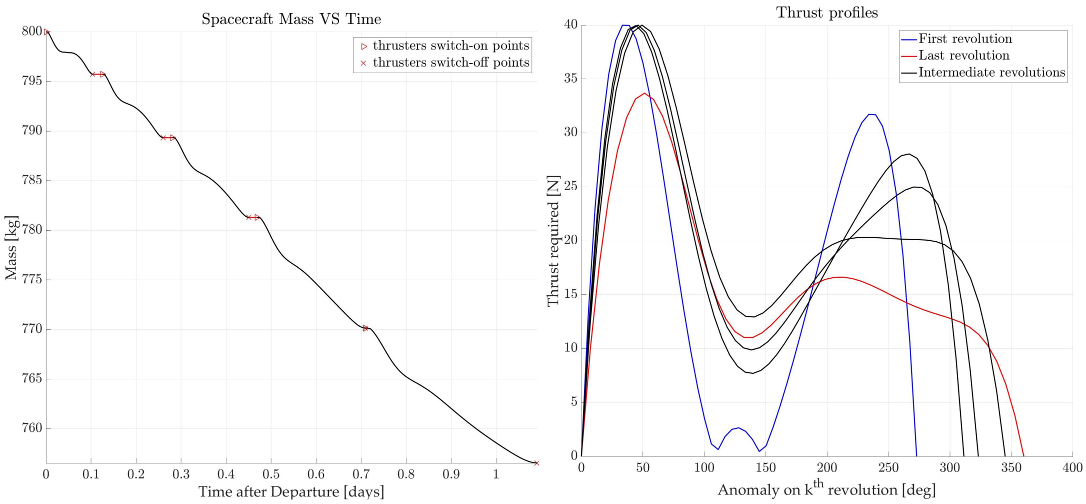


FIGURE 4.3: a) Thrust required b) Mass profile

introduced before when the intermediated orbits don't intercept the Earth shadow region a fictitious eclipse of zero length is introduced. It can be appreciated on both figure 4.2 and 4.3 in the last intermediate revolution, in which the switch-on and the switch-off positions are coincident.

4.2 Use of the shape based-algorithm in interplanetary mission scenarios

Interplanetary trajectories design and optimization introduces some other issues and constrains/objectives that is necessary to identify and formalize. Regardless from the technology adapter the thrust is linearly related with the power available: typically a specific power between 15 $\frac{W}{mN}$ and 40 $\frac{W}{mN}$ is needed. By definition in interplanetary trajectories the variation of the distance from sun is not negligible and, in most of the cases, it can affect the available power of around one order of magnitude if solar arrays are used as primary power sources due to the dependence of the solar flux to the inverse of the square of the distance [43].

Solar panels are affected by aging effects that reduces the amount of available power during time. This aging effect can be estimated between 2% and 4% [34] depending on the technology adopter for the solar panels. Interplanetary missions can lasts decades [28][30], therefore the amount of available power is also function of the total time of flight. These effects can be merged together in a unique constrain that, for shake of fastness in the convergence of the optimizer and flexibility in the mission design, is threated as an objective. The physical quantity that synthesize all the above-mentioned issues is obviously the solar panel surface needed to accomplish the mission. It can be performed evaluating equation 4.11 [43] in each computational nodes and considering the maximum value.

$$A_{SA}(x) = \frac{kT + P_{ss}}{\eta_{tot}\cos\phi\,(1-\beta)^{t}\,\frac{\phi_{Earth}}{s^2}} \tag{4.11}$$

In equation 4.11 k is the power per unit thrust, P_{ss} is the power consumption of the rest of the spacecraft, η is the efficiency of the power production/conversion, $cos(\phi)$ is the cosine of the sun angle, β is the aging factor of the solar panels and ϕ_{Earth} is the solar flux at 1 $[AU]$. The sun-angle can be supposed a-priori or computed point by point from the control law, if the geometry of the spacecraft is known. Since the optimal solution is the same for every positive multiple of the objective function, equation 4.11 can be simplified in order to generate an objective function that is less dependent from the specific parameter of the spacecraft giving the objective function in equation 4.12.

$$obj = MAX\left[\frac{T(x)s(x)^2}{(1-\beta)^{t(x)}}\right] \tag{4.12}$$

This objective function,that due to the presence of the aging effect tries to contain also the Time of Flight, can be used [28] inside a multi-objective multidisciplinary genetic algorithm together with the fuel mass. It is possible also to impose a constrain on the maximum thrust or introduce it as a third objective function: in this last case the Pareto front allows to select directly the solution that better fits the requirements and the constrains of the mission. In the following paragraph two different approaches are reported.

Augmented transfer angle

The peculiarities of interplanetary trajectories identified at the beginning of this chapter allow the possibility to use a simple 'Augmented Transfer Angle' strategy to take into account multiple revolutions, as indicated in equation 4.13 and impose the necessity to use the imposed TOF algorithm in order to guarantee the rendezvous conditions.

$$\psi = \psi + 2\pi N_{rev} \tag{4.13}$$

This strategy is extremely fast since reduces as much as possible the degrees of freedom and requires only one evaluation of the shape-based algorithm for each element of the population in the global optimization algorithm, on the other hand the flexibility of the shape is limited to the time of flight.

If one or two revolution are required this strategy is convenient and valid, otherwise is better to move to the 'intermediate orbit placement'.

Intermediate orbit parametric placement

This solution consists in the adoption of N intermediate orbits, in which N is one of the degrees of freedom of the global optimization algorithm, placed using a parameter ϵ that is itself part of the optimization vector. This solution is a good compromise between flexibility and fastness, indeed, theoretically speaking, one can use directly all the parameters of the intermediate orbits as degrees of freedom, but the size of the problem grows rapidly, vanishing all the benefits of the shape based approach. The position of the n^{th} intermediate orbit is described by equation 4.14, in which it is easy to prove that, being $\epsilon > 0$ all the intermediate orbits $(1 < n < N + 1)$ are placed between the departure and arrival one.

$$MEE_n = \left(MEE_f - MEE_i\right) \frac{1}{e^{\epsilon} - 1} \left(e^{\epsilon \frac{n}{N+2}}\right) + MEE_i \tag{4.14}$$

The exponential distribution was selected among different possibilities because of its capability to give good results also with a quite elevate value of N.

Chapter 5

Test Cases

In this chapter some mission scenarios are proposed and solved for both interplanetary and geocentric trajectories. The missions have been selected and designed with the double intent of stress as much as possible the developed models and to be of engineering interest. All the examples have been performed using MATLAB 2017b on a laptop equipped with a sixth generation quad-core Intel processor working at 2.6 GHz.

5.1 Electric orbital rising to GEO

Geostationary orbit is of primary interest for commercial satellites, especially for telecommunication purposes. The specific energy of GEO is high, and therefore the injection on this orbit typically requires a big effort, both from a technical and economical points of view. For a commercial satellite there are typically two possibilities to reach this orbit, with different advantages and disadvantages:

- **Geostationary Transfer Orbit:** The launcher provides the energy required to reach an high elliptical transfer orbit (GTO), then the spacecraft shall perform an apogee maneuver with its own propulsion system to make the orbit circular and change the orbital plane. This solution reduces a lot the specific cost of the launch but requires a strong propulsion unit on the spacecraft capable to give a ΔV between $1,5\,km/s$ and $2\,km/s$. Using chemical thrusters this implies that almost one half of the launched mass consists of propellant needed for the orbit injection. It is important to underline that this strong, massive and costly system is completely useless for the rest of the lifetime of the satellite, since station keeping and end of life disposal maneuvers can be performed also with smaller and cheaper thrusters.
- **Direct Injection:** The launcher directly performs the maneuvers required to inject the satellite on or nearby the Geostationary orbit. This solution removes the necessity of a strong propulsive unit, reduces the amount of on-board propellant and so the mass to be launched. Unfortunately, the high specific energy of the GEO orbit precludes this solution for small and cheep launchers, such as VEGA, making necessary the usage of more expensive heavy launchers.

No matter the way a GEO satellite is injected in orbit, its lifetime is affected by the amount of on-board propellant for station keeping and attitude maneuvers. For this reason, in the lasts decades full electric platforms have been designed and successfully launched: the specific impulse is at least one order of magnitude higher if compared with chemical thrusters and so the fuel consumption is much lower. On 2001 an Ariane 5 launcher fails the injection of the ESA satellite Artemis [20], leaving it at an height of 17000 kilometers: the satellite was successfully injected in GEO orbits in 2003 using its station-keeping ion thrusters, preventing the loss of the satellites. The new generations of thrusters under development [40][24] could open the way to commercial solutions for the insertion of satellites in geostationary orbit by means of electric thrusters[17]. The biggest disadvantage of this solution nowadays is the time required [**Ramos**] [2], in the order of some months, that for commercial activities is not welcomed, but the continue advance in the electric propulsion field gives hope for an application in the next future.

In this scenario the possibility to launch a satellite in GEO using electric propulsion is explored. The satellite, with the specifications listed in table 5.1, is firstly placed in a parking orbit by the European VEGA launcher [5], and then with its own propulsion unit reaches GEO.

TABLE 5.1: Spacecraft specifications

Spacecraft feature	Value
Specific Impulse [s]	3800
Dry mass[kg]	800
Maximum Thrust [N]	0.5

The inclination of the parking orbit is fixed at 5.4 $[deg]$, the minimum reachable from the Kourou without a plane change, and the standard parking orbit plane for VEGA [5]. The apocenter and pericenter radii are considered as degree of freedom for the optimization process. Their values can range from 1.03 $[DU]$ to 6.6108 $[DU]$, including therefore any possible intermediate orbit between LEO and GEO. Due to the inclination of the Earth rotation axis, the eclipses encountered by a satellite above LEO orbits are strongly affected by the period of the year: a satellite in GEO goes in Earth shadow only nearby the equinoxes [11]. Since in this example the spacecraft is supposed to be able to thrust only in sunlight, the solution will depend also on the season of the GEO arrival therefore the two opposite cases (arrival at the equinoxes or at the solstices) will be discussed. In order to highlight the importance of having included the eclipses in the model also the solution without them will be reported. Figure 5.1 shows the costs in term of fuel and TOF for any possible combination of the degrees of freedom considered for a GEO insertion during a solstice. As introduced in chapter 3, high elliptical trajectories are penalized by the shape-based methods and appear inconvenient both for time of flight and mass.

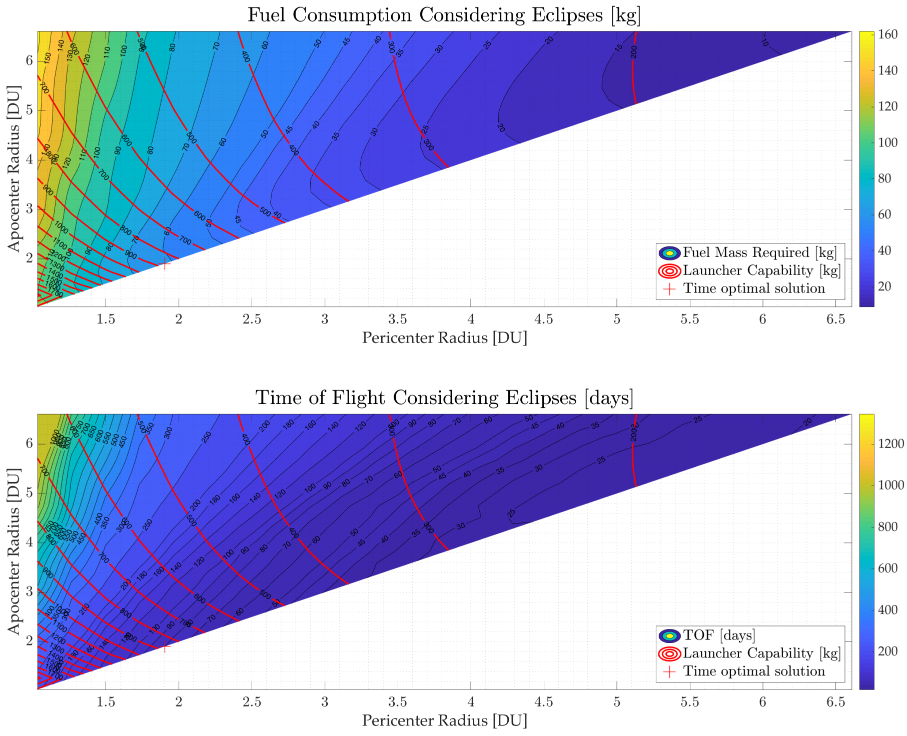


FIGURE 5.1: Solstice opportunities considering eclipses effects

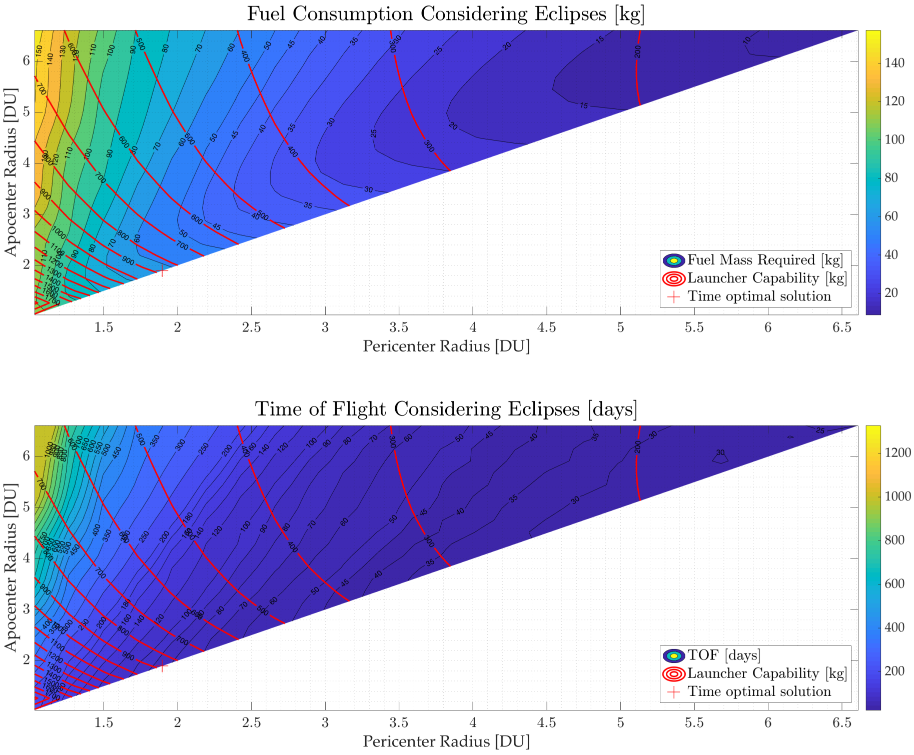
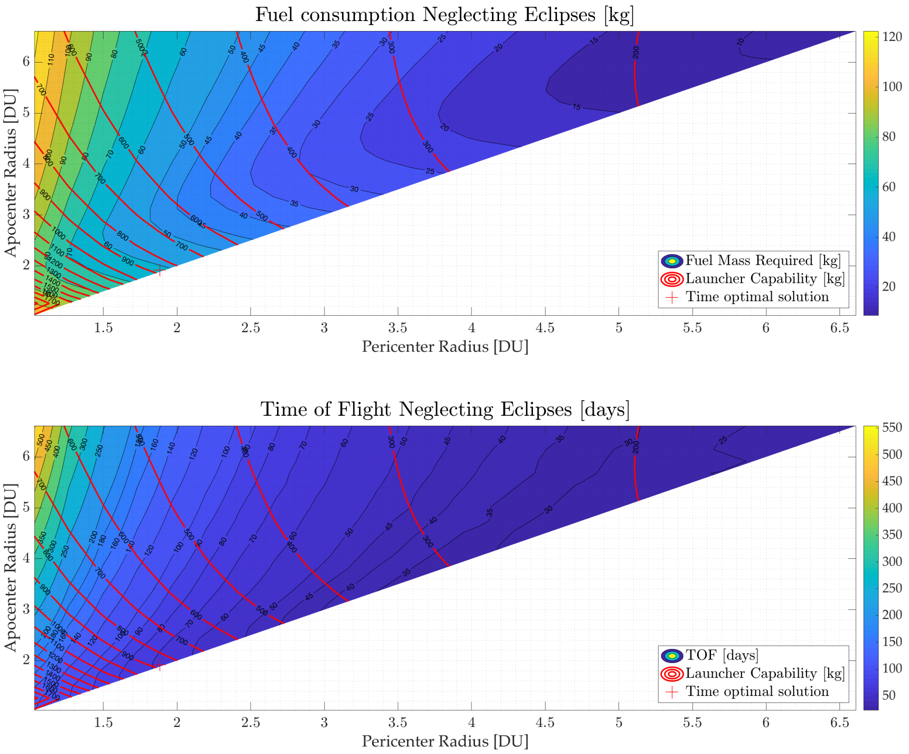
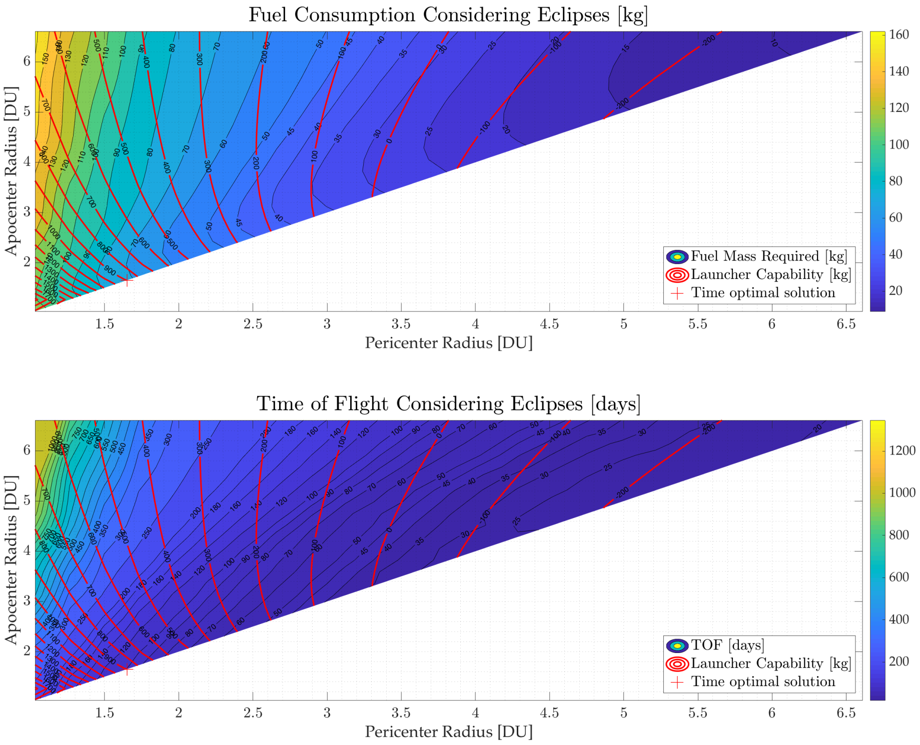


FIGURE 5.2: Equinox opportunities considering eclipses effects


Fuel consumption Neglecting Eclipses [kg]

Time of Flight Neglecting Eclipses [days]

FIGURE 5.3: Opportunities neglecting eclipses effects


Fuel Consumption Considering Eclipses [kg]

Time of Flight Considering Eclipses [days]

FIGURE 5.4: Opportunities including disposal maneuver

Figure 5.2 and figure 5.3 show the same costs for the injection during the equinoxes and without considering the eclipses effects respectively. In order to fulfill the normative on space debris prevention, in figure 5.4 it is also reported the costs for the same scenario including into the constraints also the propellant needed by the last stage of the launcher to perform a disposal maneuver. This disposal maneuver consists in a reduction of the altitude of the perigee at 150 $[km]$ with a single impulsive maneuver. Red lines represent the launchable mass using VEGA launcher: since a complete set of information for the launcher is not available, data are extrapolated applying the Tsiolkovsky equation to the last stage of the launcher (AVUM) from the reference orbit available on the user-manual [5]. The time optimal problem was solved for all the above-mentioned cases using a Nelder-Mead simplex algorithm [27][1] modified with a penalty method in order to force the solution to show an initial mass lower than the launchable one on the same orbit. The decision to adopt a derivative-free algorithm arises from the fact that, being the number of revolution discrete, the time of flight is not continue.

TABLE 5.2: Time optimal constrained solutions

Model	Eclipses		Eclipses and disposal	No eclipses
Arrival epoch	Equinox	Solstice	Equinox	—
Fuel Mass [kg]	62.30	61.72	71.5	61.59
TOF [days]	83.89	79.84	98.47	66.5
Revolutions [-]	280	279	425	214
r_p parking [DU]	1.8994	1.9019	1.6517	1.8816
r_a parking [DU]	1.9030	1.9029	1.6524	1.9207
CPU time [s]	23	31	41	12

From the results listed in table 5.2, it is clear that the time optimal strategy consists of a quasi circular switching orbit with a radius of around 2 $[DU]$. The fuel consumption is similar for all the trajectories while the time of flight increases between 20% and 25% if eclipses are considered. The effect of the injection season is much smaller: this is due to the fact that, even if nearby the equinoxes eclipses are present at any distance from ground, the fraction of time spent in shadow decreases with the radius. The CPU time highlights the capability of the algorithm to find sub-optimal solutions for multi-revolutions discontinuous trajectories very quickly: in these simulations 50 computational nodes have been used for the optimization processes and 100 for the plot of the final trajectory. All optimization processes have been initialized with the reference VEGA parking orbit (200 $[km]$ x 1500 $[km]$ height LEO orbits), the differences in the CPU times between the solutions reflect the different number of revolutions required and the increase in complexity if eclipses are encountered. In literature there is no database with time optimal solutions of GEO rising problems including eclipses in the model, therefore the only crosscheck can be done with the solutions without them. The solution found within this work is aligned with the ones available in literature **[tizio35][Ramos]** . For sake of completeness in figure 5.5 is reported the 3D trajectory for the case with eclipse and disposal maneuver that represents the most complex solution among the ones listed in table 5.1. It is easy to see that the distribution of the density of the intermediate orbits is much higher near the Earth. Regarding the eclipses it is possible to see the distortion of the shadow region caused by the motion of the Earth around the Sun and, as lower effect, the plane changes.

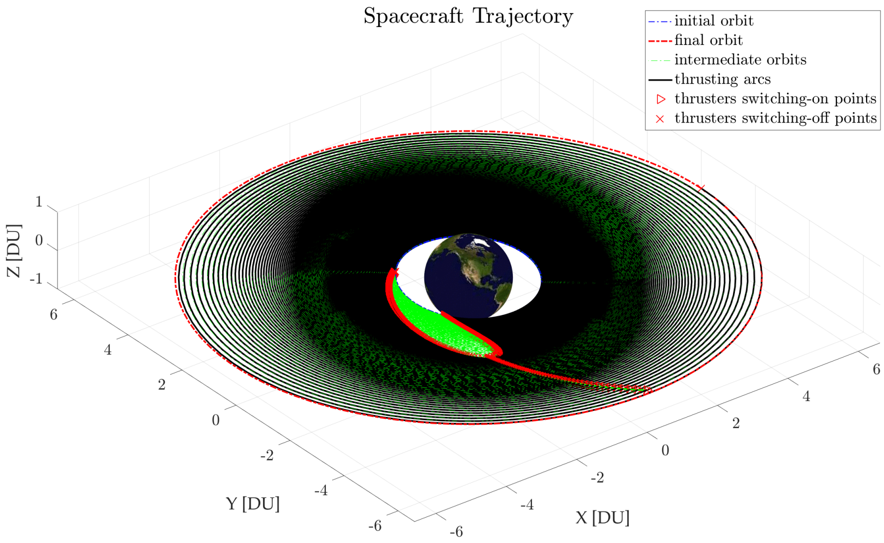


FIGURE 5.5: Time optimal trajectory with eclipses and launcher disposal maneuver

5.2 Earth-Mars rendezvous

The Earth-Mars rendezvous problem is a classical scenario for validation of low thrust trajectories design. The problem is identical to the one solved in [37]. In table 5.3 and table 5.4 are listed the spacecraft parameters and the search domain respectively.

TABLE 5.3: Spacecraft specifications

Spacecraft feature	Value
Specific Impulse [s]	3000
Dry mass[kg]	1000
Maximum Thrust [N]	0.22

The thrust required and the fuel mass fraction over the whole search domain are reported in figure 5.9; white regions are the ones in which or the thrust exceeds 1 $[N]$ or the fuel mass fraction exceeds 0.5 $[-]$. The Thrust optimal and fuel mass fraction optimal solutions are also reported: it is evident that for this algorithm the search domain proposed by Vasile is too small and therefore both the optimal solutions are located at the border of the domain. Anyway, inside the domain it is possible to

TABLE 5.4: Search domain

	Range
Departure Date	from 2028 to 2031
Time of Flight [days]	from 600 to 1000
Number of revolutions [-]	from 0 to 3

recognize two convenient regions in which the thrust and the fuel mass fraction are low even for Time of Flight reduced.

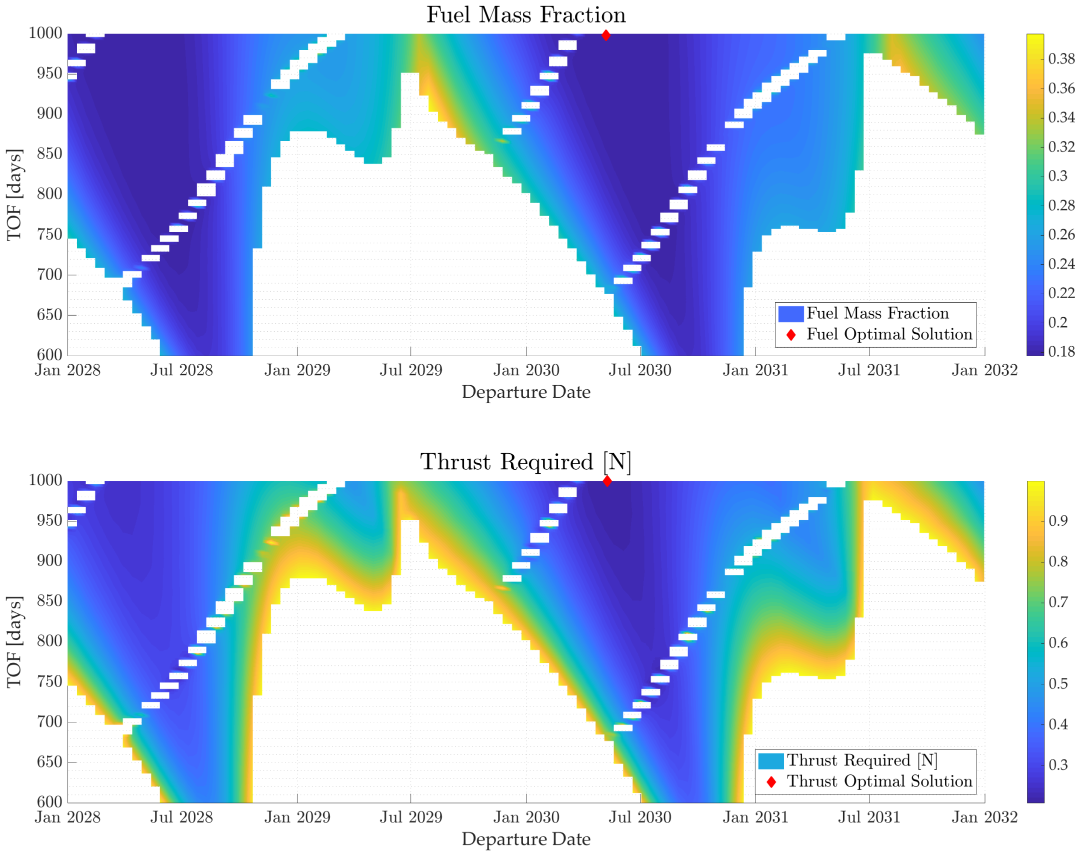


FIGURE 5.6: Thrust and fuel mass fraction required

TABLE 5.5: Optimal solutions

Solutions	fuel optimal	thrust optimal
Departure Date	8 May 2030	3 May 2030
TOF [days]	1000	1000
Thrust required [N]	0.208	0.207
Fuel Mass fraction [kg]	0.1777	0.1778
Revolutions [-]	1	1
CPU time [s]	8.7	4.7

In table 5.5 the thrust and fuel mass fraction optimal solutions are reported: it can be seen that they are almost equivalent and are extremely similar to the ones that can be found in [37], in which the fuel mass fraction ranges between 0.177 and

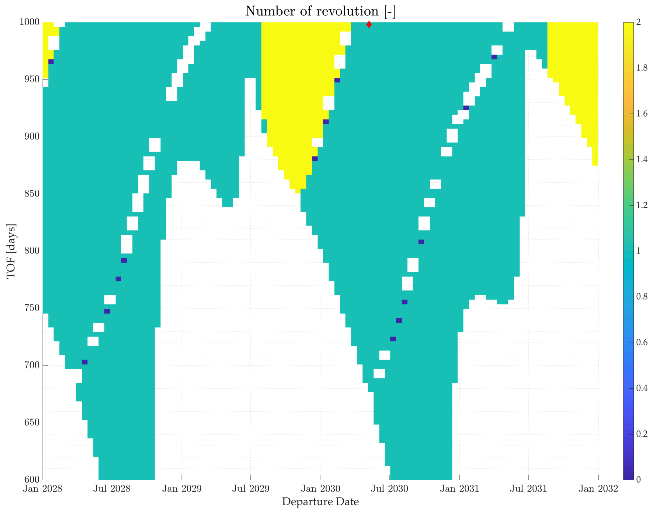


FIGURE 5.7: Number of revolution required

0.188 and the TOF ranges between 717 and 998; In these trajectories 100 computational nodes have been used. This solution was found using the build-in MATLAB Genetic Algorithm function ('ga') with a population of 100 individual and a standard stopping criteria based on the average changes on the cost function (less than 10^{-6}). The developed algorithm doesn't give any advantages in term of optimality of the solution in this scenario and slightly penalize the TOF. The only advantage is that it requires only 5-10 seconds to run instead of the 10 minutes mentioned in [37].

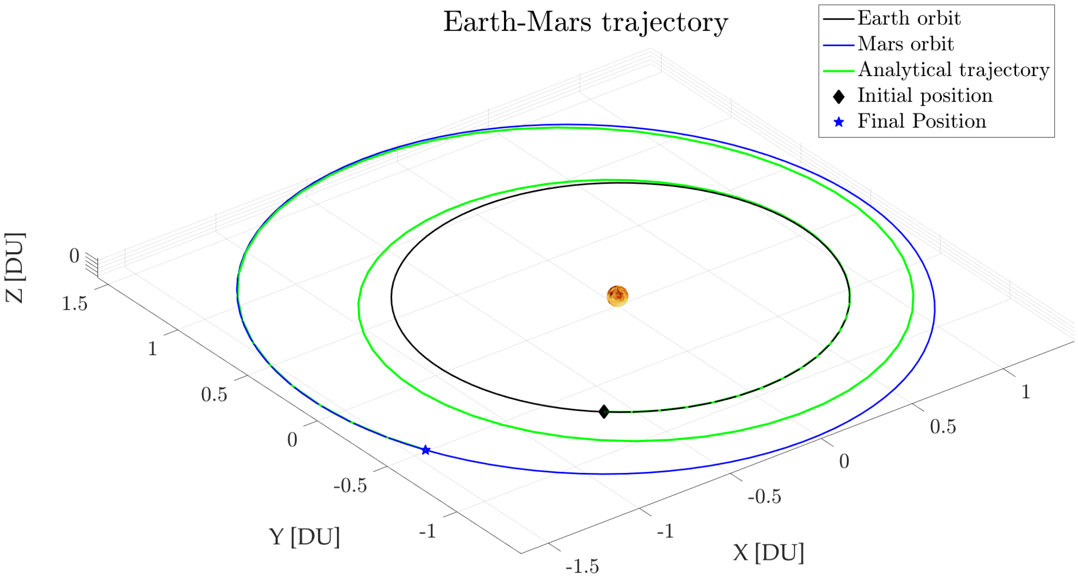


FIGURE 5.8: Earth Mars trajectory

The output optimal trajectory is reported in figure 5.8 while the control law and the fuel mass consumption are reported in figure 5.9.

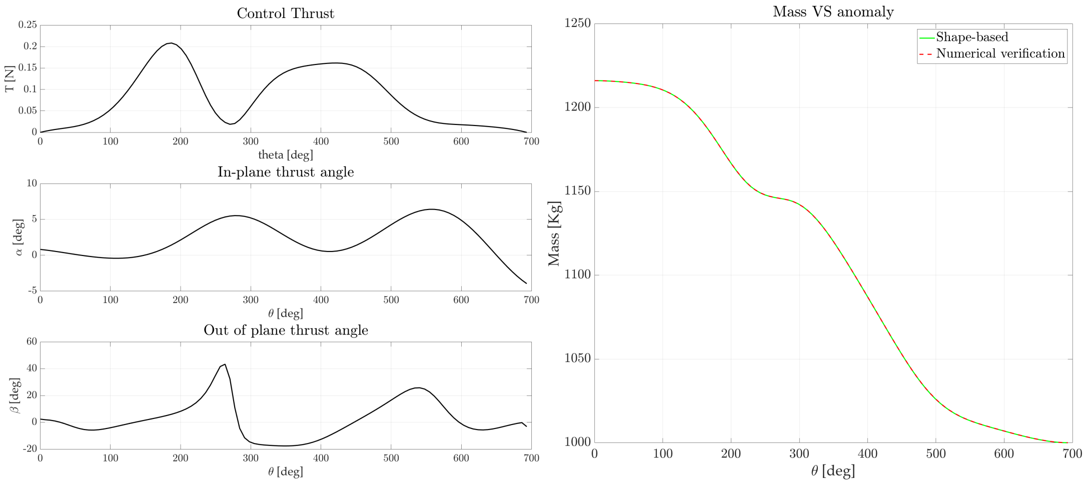


FIGURE 5.9: a) Control law b) Mass profile

5.3 Earth-Nereus Mission

This scenario was selected to underline the ability of the shape based algorithm to find near optimal solution when high elliptical orbits are considered. Nereus is a Near Earth Object with an high elliptical orbit on a plane slightly different with respect to the Earth one; the full set of Keplerian element referred to the 1^{st} January 2000 is reported in table 5.6.

TABLE 5.6: Nereus Keplerian elements

Keplerian Element	Value
Semi-major axis [AU]	1.4896
Eccentricity [-]	0.36026
Inclination [deg]	1.4238
RAAN [deg]	69.127
Argument of pericenter [deg]	112.66
True Anomaly [deg]	314.76

It is important to underline that the pericenter of Nereus orbit is located nearby the Earth's one, therefore a quasi ballistic solution with a non zero escape velocity is expected if or the spacecraft's fuel mass fraction or the thrust required are selected as objectives. In order to try to find the best trajectory an extremely wide search space is considered: the degrees of freedom and their ranges are reported in table 5.7.

TABLE 5.7: Search domain

	Range	Optimum
Departure Date	from 2030 to 2050	9^{th} February 2042
Time of Flight [days]	from 500 to 1500	690.5
Number of revolutions [-]	from 0 to 2	1
v_{inf} departure [km/s]	from 0 to 6	5.93
v_{inf} in plane angle [deg]	from -90 to +90	-8.07
v_{inf} out of plane angle [deg]	from -90 to +90	44.79

Regarding the optimization process, the MATLAB genetic algorithm is adopted with a population of 1000 individuals and the optimal trajectory is obtained after 5 minutes with 100 computational nodes. As can be seen from the output trajectory reported in figure 5.10, the launcher inserts the spacecraft, that has the specification listed in table 5.3, directly in a quasi ballistic orbit, as expected from theory. The only use of the thrusters is the relative approach phase to Nereus, that is reported for sake of completeness in figure 5.12; slightly more than 10 mN of thrust are sufficient. Control law and fuel consumption are reported in figure 5.11: as can be easily derive the fuel mass fraction is only 0.0052. The degrees of freedom associated to the optimal solution are listed in table 5.7.

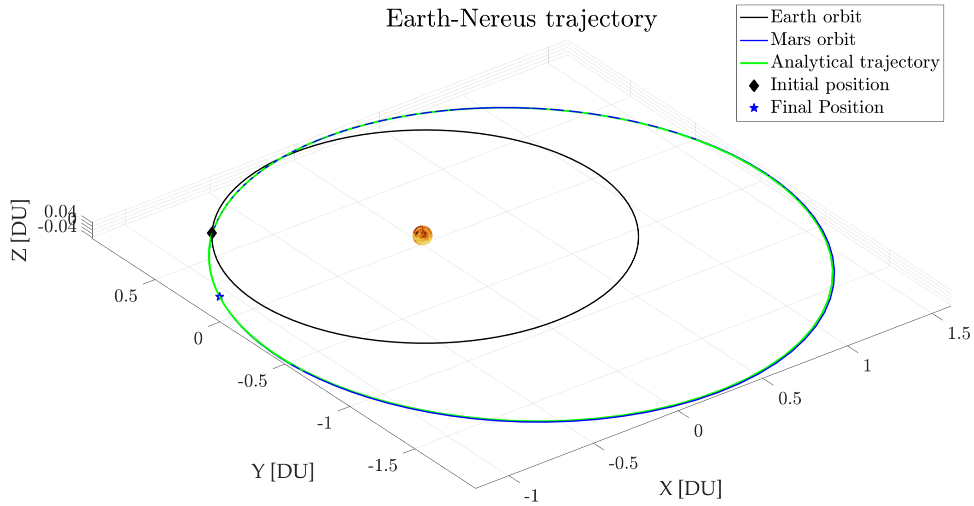


FIGURE 5.10: Earth Nereus trajectory

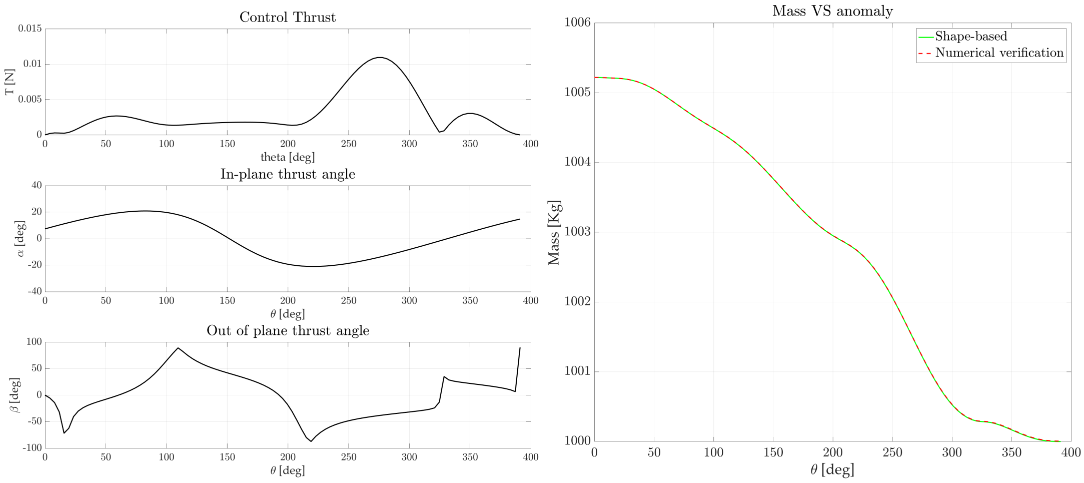


FIGURE 5.11: a) Control law b) Mass profile

This example shows that the developed algorithm is capable to manage also high elliptical orbits in interplanetary trajectories: this is possible only because the peculiarity of the shape is to be a non-linear interpolation between arrival and departure orbits.

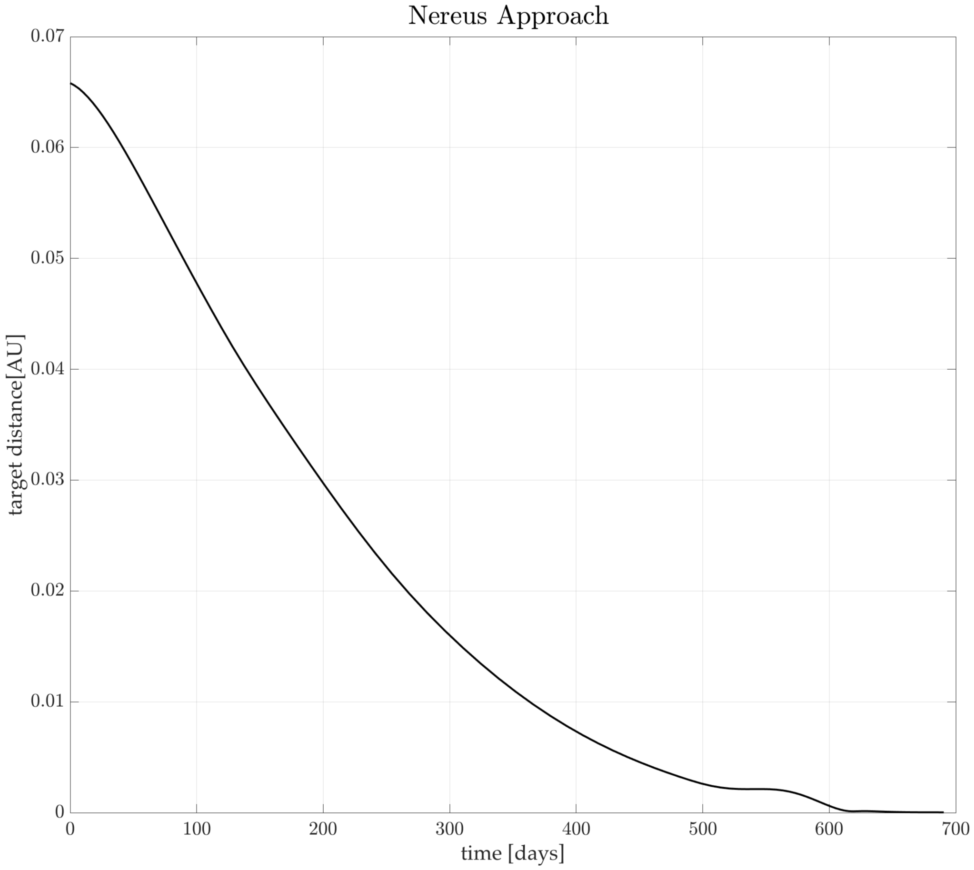


FIGURE 5.12: Spacecraft-Nereus distance

5.4 Bepi-Colombo comparison

In order to try to prove the capabilities of the algorithm in real situation it was also tested in comparison with an arc of the exact flight plan of the BepiColombo ESA spacecraft. Since the flight plan was developed keeping into account the effect of interplanetary perturbations, the exact ephemerids and not only continuous thrust, this algorithm shows not so god results: it takes 30 kg of propellant for an arc that will takes only around 5 kg in the real plan. The thrust peak is also higher with respect to the Bepi-Colombo one. As better explained in chapter 6 one possibility to reduce this discrepancy could be to consider directly the perturbations on the shapes of the trajectories and including ballistic arcs. Anyway shape-based algorithms are not thought to outputs optimal flight plans, so the above-mentioned discrepancy will always be present, even if reduced.

Chapter 6

Conclusions and future works

The goal to develop a shape-based algorithm able to deal with Earth-centered scenarios is achieved and, especially when utilized in multi-revolution problems, the developed methodology shows some great advantages. The first is the extremely low computational cost that makes the algorithm suitable for the optimization of complex mission scenarios. This computational speed belongs to the decision to include directly the boundary conditions in the interpolating functions, avoiding the necessity to compute them at each step (as happens in some classical shape-based algorithms [9][42]) and to the attention posed on the selection of light numerical techniques able at the same time to contain the numerical error. The MATLAB implementation required a big effort since the usage of build-in function was reduced to the minimum: almost every function, from the most complex, such as 'fminsearch' (that implements Nelder-Mead simplex algorithm) to the simpler, such as dot and cross products, have been rewritten in order to be as light as possible. The vectorial structure of the single revolution algorithm is quite effective if less than 200 nodes are adopted, and guarantees a further reduction on the computational cost. In order to further improve the speed of the algorithm some strategies have been identified:

- **C porting:** the reduced number of build-in functions adopted makes easy a future porting in C language, that is more efficient if compared with MATLAB [1]. The only part of the algorithm that will be developed from scratch or taken from literature is the Heuristic algorithm.

- **GPU computing:** most of Heuristic algorithms evaluate independently at each step a wide number of individuals. If the Heuristic algorithm is parallelized on a GPU the advantages are enormous, being common GPUs made by hundreds of cores.

The eclipse are included in the model with a simple but effective strategy, that shows also a good robustness even if an elevate number of discontinuities is introduced: this is possible mainly thanks to the particular combined Newton-Bisection algorithm discussed in chapter 4. The proposed interpolating functions show an acceptable degree of flexibility, anyway a further analysis can be performed in order to explore more performing shapes, maybe introducing two different interpolating

functions for the planar and out of plane motions. In the developed algorithm Natural Perturbation can be implemented only a-posteriori in order to correct the control law: another possible future improvement is the implementation of the perturbations directly on the intermediate orbits, in this way the resulting interpolated kinematics will be closer to the real dynamic. In order to do that without increase to much the computational weight the perturbed intermediates orbits will have to be determined with an analytical (or semi-analytical) approach.
Another interesting possible expansion of the algorithm is the inclusion of the time of flight constrain on the multi-revolution planeto-centric trajectories. This could be obtained using the TOF free multi-revolution algorithm for the whole trajectory excluded the lasts revolutions, in which the TOF is constrained in order to reach a moving target.

Regarding the interplanetary trajectories the developed algorithm shows an improvement on the solution optimality with respect to the in-literature available [3][42] only if high elliptical orbits are considered, otherwise the developed algorithm shows results with a similar optimal fuel mass fraction[37] but with a larger Time of Flight, making it inconvenient. Also in the interplanetary case case the computational speed is elevate and allows to threat complex problems. The biggest disadvantages in interplanetary trajectory is underlined with the BepiColombo comparison: if the solution is compared with an exact flight plan costs are different. It is evident the lack of a detailed dynamical model in interplanetary environment: in this case a direct optimization algorithm is obviously preferable. Anyway the developed shape based algorithm can be successfully used for the generation of a guest trajectory for direct optimization and can be also used in the firsts phases of a mission design.

Bibliography

[1] “Calcolo Scientifico”. In: ed. by P. Gervasio A. Quarteroni F. Saleri. Springer, 2012.

[2] David B. Spencer Albert L. Herman. “Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods”. In: *Journal of Guidance, Control, and Dynamics* 25.1 (2002), pp. 40, 47.

[3] Jon A. Sims Anastassios E. Petropoulos. “A Review of Some Exact Solutions to the Planar Equations of Motion of a Thrusting Spacecraft”. In: *In Proceedings of the 2nd International Symposium on Low Thrust Trajectories*. Toulouse, France, 18 June 2002.

[4] James M.Longuski Anastassios E.Petropoulos. “Shape-Based Algorithm for Automated Design of Low-Thrust. Gravity-Assist Trajectories”. In: *Journal of Spacecraft and Rockets* 41.5 (2004), pp. 787, 796.

[5] “VEGA user’s Manual”. In: ed. by Arianespace. Arianespace, 2014.

[6] “An Introduction to the Mathematics and Methods of Astrodynamics”. In: ed. by R. H. Battin. Reston: AIAA Education Series, 1999.

[7] John T. Betts. “Survey of Numerical Methods for Trajectory Optimization”. In: *Journal of Guidance, Control, and Dynamics* 21.2 (1998), pp. 193, 207.

[8] John T. Betts. “Very low-thrust trajectory optimization using a direct SQP method”. In: *Journal of Computational and Applied Mathematics* 120 (2000), pp. 27, 40.

[9] Bruce A.Conway Bradley J.Wall. “Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design”. In: *Journal of Guidance, Control, and Dynamics* 32.1 (2009), pp. 95, 101. DOI: 10.2514/1.36848.

[10] P. J. Cefola. “Equinoctial Orbital Elements - Application To Artificial Satellite Orbits”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Palo Alto, California: AIAA, 1972.

[11] “Orbital Mechanics”. In: ed. by V. A. Chobotov. Reston: AIAA Education Series.

[12] Bruce A. Conway. “Spacecraft Trajectory Optimization”. In: Cambridge University Press, 2010. Chap. 3.

[13] Steven R. Oleson Craig A. Kluever. “Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers”. In: *Journal of Spacecraft and Rockets* 35.6 (1998), pp. 509, 515.

[14] "Orbital Mechanics for Engineering Students". In: ed. by Howard Curtis. Oxford: Elsevier, 2005.

[15] M. Vasile D. M. Novak. "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories". In: *Journal of Guidance, Control, and Dynamics* 34.1 (2011), pp. 128, 147.

[16] "Fundamentals of Electric Propulsion: Ion and Hall Thrusters". In: ed. by Ira Katz Dan M. Goebel. Jet Propulsion Laboratory California Institute of Technology, 2008.

[17] John W. Dankanich and Gordon R. Woodcock. "Electric Propulsion Performance from Geo-transfer to Geosynchronous Orbits". In: *30th International Electric Propulsion Conference*. Florence, Italy: IEPC, 2007.

[18] Ella Atkins Ehsan Taheri Ilya Kolmanovsky. "Shaping low-thrust trajectories with thrust-handling feature". In: *Advance in Space Research* (2017), pp. 879, 890.

[19] "http://sci.esa.int/bepicolombo/". In: ed. by ESA. ESA, May 2018.

[20] "https://earth.esa.int/web/eoportal/satellite-missions/". In: ed. by ESA. ESA, May 2018.

[21] C.Zhang F.Topputo. "Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications". In: *Abstract and Applied Analysis* 2014 (2014), p. 15.

[22] "Rocket Propulsion Elements". In: ed. by O.Biblaz George P. Sutton. Hoboken: Wiley, 2010. Chap. 17.

[23] Kathryn F. Graham and Anil V. Rao. "Minimum-Time Trajectory Optimization of Multiple Revolution Low-Thrust Earth-Orbit Transfers". In: *Journal of Spacecraft and Rockets* 52.3 (2015), pp. 711, 727.

[24] R.Kukies J.Kuhmann J.-P. Porst Marcel Berger H. J. Leiter Ch.Altmann and Michael Rath. "Evolution of the AIRBUS DS GmbH Radio Frequency Ion Thruster Family". In: *34th International Electric Propulsion Conference*. Hyogo-Kobe, Japan: IEPC, 2015.

[25] A. L. Herman and B. A. Conway. "Direct Optimization Using Collocation Based on High Order Gauss-Lobatto Quadrature Rules". In: *Journal of Guidance, Control, and Dynamics* 19.3 (1996), pp. 592, 599.

[26] Tsien H.S. "Take-Off from Satellite Orbit". In: *Journal of the American Rocket Society* 23.4 (1953), pp. 233–236.

[27] "Numerical Optimization". In: ed. by Stephen J. Wright J. Nocedal. New York: Springer, 2006.

[28] M. Lavagna J. Prinetto. "Main Belt Active Asteroids Samples Collection And Return Mission Design". In: *Italian Association of Aeronautics and Astronautics XXIV International Conference*. Palermo-Enna, Italy: AIDAA, 18-22 September 2017.

[29] "http://global.jaxa.jp/projects/sat". In: ed. by JAXA. JAXA, 2018.

[30] "https://dawn.jpl.nasa.gov". In: ed. by JPL. NASA, Oct. 2017.

[31] "https://ssd.jpl.nasa.gov". In: ed. by JPL. NASA, May 2018.

[32] Bellei G. Lavagna M. "MOPSO Technique Assessment to Cope with First Guess Generation for Interplanetary Trajectories Differently Controlled". In: *58th International Astronautical Congress*. Hyderabad, India: IAC, 2007.

[33] M. Lavagna M. Scotti. "Global optimization of multiple gravity assist trajectories: development of STA Interplanetary Module v3.0". In: *Master of Science thesis*. Milan, Italy: Politecnico di Milano, 2013.

[34] "Space Vehicle Design". In: ed. by James R. French Michael D. Griffin. Reston: AIAA education Series, 2004.

[35] "https://www.jpl.nasa.gov/missions/deep-space-1-ds1/". In: ed. by NASA. NASA, May 2018.

[36] S. Ceccherini N.R. Ramos F. Topputo. "Indirect Optimization of Electric Propulsion Orbit Raising to GEO with Homotopy". In: *Master of Science thesis*. Milan, Italy: Politecnico di Milano, 2017.

[37] M. Vasile P. De Pascale. "Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories". In: *Journal of Spacecraft and Rockets* 43.5 (2006), pp. 1065, 1076.

[38] B. Conway P. Enright. "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming". In: *Journal of Guidance, Control, and Dynamics* 15.4 (1992), pp. 994, 1002.

[39] E. Y. Choueiri R. G. Jahn. "Electric Propulsion". In: *Encyclopedia of Physical Science and Technology* 5 (2002), pp. 125, 141.

[40] "http://www.sitael.com". In: ed. by sitael. sitael, Oct. 2017.

[41] John T. Betts. "Optimal interplanetary orbit transfers by direct transcription". In: 42 (July 1994), pp. 247–268.

[42] Bradley J. Wall. "Shape-Based Approximation Method for Low-Thrust Trajectory Optimization". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Honolulu, Hawaii: AIAA, 2008.

[43] "Space Mission Analysis and Design". In: ed. by James R. Wertz Wiley J. Larson. London: Kluver Academy Publisher, 1999, p. 407 427.