POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Syzygy design strategy for the multiple gravity assist problem

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Giovanni Mapelli**

Student ID: 928371
Advisor: Prof. Camilla Colombo
Co-advisors: Alessandro Masat
Academic Year: 2022-23

Giovanni Mapelli: *Syzygy design strategy for the multiple gravity assist problem*

When referring to this work, full biographical detail must be given:
Giovanni Mapelli, "Syzygy design strategy for the multiple gravity assist problem", 2023, Politecnico di Milano, Faculty of Industrial Engineering, Department of Aerospace Science and Technologies, Master in Space Engineering, Supervisor: Camilla Colombo, Co-supervisor: Alessandro Masat.

Abstract

Space exploration and the design of complex interplanetary missions towards distant celestial objects often require very high orbital velocities which the state of the art of the propulsive systems cannot provide yet. Therefore, alternative solutions must be explored to make interplanetary missions feasible endeavours. For this reason, it is essential to include gravity assist maneuvers and planetary flybys in the trajectory design. This strategy is widely applied, such as in the Voyager and Cassini missions.

This thesis presents an innovative procedure to design the interplanetary legs connecting close encounters, by means of a generalization of the *syzygy* functions for planetary alignment, together with the B-plane characterization of flybys, developed by Öpik.

In the second part of the thesis, the preliminary design and optimization of the multiple gravity assist trajectories is tackled by means of techniques of combinatorial optimization. A dynamic programming approach to the problem is proposed, that enables the selection of the most precise and feasible solution with great computational efficiency. The proposed algorithm is tested by reproducing Voyager-like trajectories.

Sommario

L'esplorazione spaziale e il disegno di missioni interplanetarie complesse spesso richiede altissimi livelli energetici. Il livello tecnologico propulsivo è ancora inadeguato per ambire a traiettorie complesse, per cui è necessario ricercare delle strategie alternative che portino a delle missioni effettivamente realizzabili. Per questo motivo è diventato essenziale integrare nel profilo della missione manovre di fionda gravitazionale e flyby planetario, come ad esempio nelle missioni Voyager e Cassini.
In questa tesi si propone una procedura alternativa per il disegno delle traiettorie interplanetarie che collegano le diverse manovre di fionda, che consiste in una generalizzazione delle funzioni di *syzygy* per l'allineamento planetario, insieme alla descrizione del flyby planetario tramite il formalismo del B-plane di Öpik.
La seconda parte della tesi affronta il problema della progettazione e dell'ottimizzazione delle fionde gravitazionali multiple. Si propone un approccio di dynamic programming al problema, che permette di selezionare le soluzioni più precise e realizzabili, mantenendo contenuto il costo computazionale. L'algoritmo proposto è successivamente testato riproducendo traiettorie nello stile delle missioni Voyager.

Ringraziamenti

Ringrazio la professoressa Colombo, che ha ideato questo progetto di tesi, dandomi la possibilità di dedicare il mio tempo e le mie energie a ciò che mi appassiona.
Grazie agli amici conosciuti all'università: a Luca, per tutte le risate e le piacevoli distrazioni durante le lezioni, a Marco per la gentilezza e sensibilità che mi ha sempre dimostrato, a Mettiu e Sam, che mi hanno fatto appassionare all'arrampicata, a Fabio, fidato compagno di allenamenti, a Christian per le innumerevoli giornate alle terme, a Dave per essere stato un punto di riferimento come ingegnere ma soprattutto come uomo, e ad Alessandro, che mi è stato vicino nei momenti più bui, come un fratello.
Grazie agli amici nuovi, a Carlo, Michi, Fabio, Teo, Mezzi e Raffo, che mi hanno fatto sentire ancora giovane.
Grazie agli amici di una vita, a Toni, Cita e Homer, che mi hanno sempre sostenuto nelle scelte che ho preso, e che ci sono sempre stati per me.
Grazie a Mamma e Papà, per avermi cresciuto, per avermi permesso di coltivare tutte le mie passioni e per tutto l'amore che mi avete donato. Grazie a Maggie per essere stata la spalla su cui appoggiarmi nei momenti duri.
Grazie infine a Siria, che con il suo amore, come una luce nel buio mi guida e mi accompagna nella vita.

Contents

List of Figures

List of Tables

List of Acronyms

GAM Gravity Assist Maneuvers
MGA Multiple Gravity Assist
tof Time Of Flight
LVLH Local Vertical - Local Horizontal
DCM Direction Cosine Matrix
SOI Sphere of influence
MCTS Monte Carlo Tree Search
CR3BP Circular Restricted 3 Body Problem
KM Keplerian Map

1 | Introduction

In the past 60 years interplanetary spaceflight has shown the great scientific value of data gathered from the observation of the Solar System. Certain observations can be performed efficiently from Earth's surface or from orbits around the Earth, but most of the times it is required to leave the Earth and travel deep inside of the solar system to gather adequate information. It is the case of missions like Cassini, Voyager and Pioneer, that deepened human knowledge about various planets, moons and celestial bodies of the Solar System. A great issue arose from the need of interplanetary exploration: in order to travel deep in the solar system, very large velocity changes are required. Currently, launch systems and propulsion technologies are still inadequate to reach targets whose orbits have much higher or lower energy levels with respect to the Earth; an alternative strategy has been found by exploiting Gravity Assist Maneuvers (GAMs). A gravity assist manoeuvre (also called swing-by or gravitational slingshot), is the use of the relative movement and gravity field of a planet or other massive celestial body to change the velocity of a spacecraft. This is achieved with a close proximity swing-by of the celestial body so that its gravity produces a change in the velocity vector of the spacecraft. [26]. This idea has been exploited widely in the last decades, in missions such as Bepicolombo [38], Cassini-Huygens [29], Juice [19] and many others.

However, in many cases, a single GAM is not sufficient to achieve the mission target. This is due to the fact that a close encounter can generate a limited variation of the orbital parameters of the spacecraft, depending on it's velocity magnitude and orientation and on the mass of the close encounter body. To overcome this limitation, it is necessary to perform a definite number of GAMs to reach the objective. The idea behind this strategy is to obtain the prescribed energy variation by diluting it into an arbitrary number of GAMs, each of which causes feasible variations of the orbital parameters. These kind of missions are identified as Multiple Gravity Assist missions (MGA).

The problem at hand has increased greatly in complexity and the design of the trajectory involves several challenging aspects. Firstly it is necessary to identify a planetary sequence, which has a great impact on the optimality of the solution. This is due to the fact that different planetary sequences can lead to extremely different trajectories, in

terms of initial Δv, overall time of flight and departure time window. Once the sequence is defined, a favorable spatial positioning of the selected planets must be sought, in order to actually reach the target planet.
For these reasons, efficient methods based on simplified dynamics models are required to quickly explore such complex solution spaces in the early stages of the design.
Over the years, several methods were proposed to simplify the design of the flyby, with the Keplerian Map [34], the Tisserand-Poincarè map [10] and the flyby [35] and kick map for low energy flyby [5]. The Keplerian Map (KM) is a common method to study the flyby effects in the Ciruclar Restricted 3 Body Problem (CR3BP). Based on perturbation theory, this method was initially developed to investigate the dynamics of comets around Jupiter, whereas Tisserand-Poincarè graphs provide an efficient way to tackle the combinatorial part of the MGA problem, by allowing a simple computation of the effect of different sequences of gravity assists, based only on energy considerations [25]. The Flyby Map is also a typical approach to study the CR3BP, which is fully numerical and valid for a wide range of energy levels. It extended the functionality of the Tisserand-Poincarè graph over the applicability of the patched-conics model.
Nevertheless, the above methods offer a solution which depends on the parameter state vector (e.g. initial epoch, position, velocity) and therefore require to be repeated for each initial condition to give a representation of the solution in "global" map. [14]. Instead, it is possible to efficiently simplify the case, following D. Menzio et al. [14] idea: the trajectory design issue is treated simply as the study and search for the zero of a function that describes the problem, the *syzygy* function, that will be presented in detail in Chapter 3.

1.1. Contribution of the thesis work

In the first part of the thesis, the procedure presented by Menzio et al. [14] is further improved by removing one of its main simplifying assumptions, namely that the transfer leg must either start or end at one of it's absidal points. In this way, the designed solutions will be more realistic, and the solution search more generalised than the ones proposed by Menzio et al.. Furthermore, the model presented by Menzio et al. is extended to hyperbolic orbits, in order to include open trajectories to the possible design strategies. The initial model is then combined with the B-plane formalism introduced by Öpik [15], in order to implement efficiently the close encounter effect on the spacecraft's orbital parameters. This innovation to the initial model allows the design process to identify only the flybys that are feasible from an orbital mechanics point of view.
In the second part of the thesis, once an optimality policy is chosen, the developed model

is combined with a dynamic programming approach to address the search of the optimal trajectory between the solutions computed by means of the *syzygy* algorithm. Dynamic programming in fact, provides a very efficient optimisation method for problems involving combinatorial elements, and allows to find the global optimal solution by investigating only a reduced portion of the solution space. Overall, the objective of the thesis is to provide a fast and reliable algorithm for computing preliminary solutions to MGA design problems, whose results could serve as reasonable starting point for numerical methods to faster converge to more complete trajectory solutions.

2 | The multiple gravity assist problem

Especially in the early times, the preliminary design of MGA missions was approached mainly relying on the intuition of mission designers in unison with analysis tools, based on a number of simplifying assumptions, like the CR3BP and patched conics frameworks, such as in the Tisserand's graph [4] [30]. These techniques were mainly used to scan a large range of options in order to identify a restricted number of potentially good solutions (or first guesses). A more accurate design of these solutions was performed only as a second step, by using higher fidelity models with more complex dynamics, optimal control theory, and optimisation techniques [2].
Regarding the latter, a great number of possible approaches can be selected to tackle this issue. In this Chapter a brief outline of the typical solution strategies for this problem is presented.

2.1. Interplanetary trajectory design strategies

The MGA problem can be roughly represented by a series of two tasks: establishing an effective planetary sequence and evaluate the possible trajectories that arise from the particular sequence considered. Then, the trajectories can be obtained with different strategies: ballistic arcs, low-thrust or low-energy transfers and impulsive maneuvers.
In the last decades global optimization techniques were extensively used towards the solution of complex interplanetary trajectory transfers, together with methods including neurocontrollers [13], shooting [12], and collocation [22], showing variable effectiveness.

2.1.1. Optimisation algorithms

Commonly used optimization algorithms can be classified into three main classes: stochastic algorithms, which involve a suitably chosen random sample of points and subsequent manipulation of the sample to find good local minima; guaranteed algorithms, which are

deterministic algorithms that guarantee the identification of a global optimum with a required accuracy; metamodel algorithms that exploit the construction of metamodels, and do not perform the global search on the real objective function, but on a metamodel of it, that is, a model of the model, or also a simplified model of an actual model of a circuit, system, or software like entity.
Among these, stochastic algorithms have shown to be the generally suitable for the problem at hand, and they comprehend a great variety of methods, but the most exploited ones in the trajectory design framework are Simulated Annealing, which performs the global search based on successive update steps, and Evolutionary Algorithms, which globally search the solution space by simulating the self-optimising natural process of evolution. Evolutionary Algorithms can be further divided in three main branches: Genetic Algorithms, Evolutionary Programming and Evolutionary Strategies [32].
In the last decades, different forms of stochastic search methods have been applied to orbit design, such as, Vasile in 2003 [39] proposed a stochastic global optimiser (EPIC) that was tested on interplanetary transfer, and later used by Vasile and De Pascale as a component of the interplanetary design tool IMAGO [28].
In 2008, Vasile et al. [27] proposed a generalisation of Differential Evolution (DE) and Particle Swarm Optimization [21] in the form of discrete-time dynamical system.
In order to select more easily a solution strategy, Di Lizia et al. tested different stochastic algorithms, and identified a limited number of global optimization methods to be applied, depending on the type of trajectory design chosen [32], since the efficiency, both computational and performance-wise, of these approaches are strongly linked to the type of problem considered. The tested algorithm set embraces classical genetic algorithms including different Genetic Operators for performing the global search (GAOT and GATBX), Genetic Algorithms with sharing and migration operators (GAOT-shared and GATBX-migr respectively), Evolutionary Programming (Fast Evolutionary Programming, FEP), Differential Evolution (DE), an improved Simulated Annealing (Adaptive Simulated Annealing, ASA), branching methods (glbSolve and MCS), response surface based optimisation algorithms (rbfSolve) and, an innovative hybrid systematic-heuristic method combing branching techniques and evolutionary programming (EPIC).
It emerged from the tests that, for example, in the simple case of 2-impulse direct planet to planet transfer problem, the Monte Carlo Search algorithm (MCS), a branching method, has shown to be the best performing one in terms of computational effort (number of objective function evaluations to reach the solution). For low thrust transfers instead DE and GATBX-migr showed good performances.
Ballistic trajectories are characterized by a lower number of design variables, since instantaneous maneuvers and control laws for thrust profile determination are not considered,

and show a highly combinatorial nature. Dynamic programming is especially suitable for problems involving combinatorial elements, such as ballistic trajectories, and therefore has been chosen for this thesis work [18].
All the strategies presented so far only deal with the trajectory design issue, leaving the selection of the planetary sequence as an unresolved task. This problem has been addressed with different strategies: from intuition and heuristics, to, in the more recent years, what is named automatic MGA design.
Regarding the automatic design of MGA trajectories (that is, both the planetary sequence and trajectory has to be determined), the literature shows that the problem has been approached with several different techniques: from deterministic approaches, to graphical tools like the Tisserand-Poincarè Map or the Flyby Map [4] [30], from stochastic approaches to hybrid methods, like the procedure presented by Hennes and Izzo [20], based on a heuristic free approach to automated trajectory planning (including the encounter sequence planning) based on Monte Carlo Tree Search (MCTS). MCTS is a technique widely applicable in domains that require sequential decision making, including game-tree search and planning problems. The MCTS paradigm combines informed tree search with the generality of Monte Carlo simulations
Nevertheless, all of them can be classified in two main categories: two level approaches and integrated approaches. Two-level approaches define the planetary sequence independently of the trajectory itself and split the problem into two sub-problems which lay at two different levels: one sub-problem is to find a suitable set of sequences of planetary encounters; the other is to find at least one optimal trajectory for each sequence. The STOUR software, for example, is based on a two-level approach [32]. As opposed to the two-level approaches, integrated approaches define a mixed integer-continuous optimisation problem, which tackles both the search of the sequence and the optimisation of the trajectory, using a single model, at the same time [28].
The planetary sequence determination in this thesis is completely arbitrary, since the main objective of this work deals strictly with the preliminary design of MGA ballistic trajectories. A possible future direction for this work could be indeed the implementation of a fully automatic MGA design strategy.

3 | Generalized *Syzygy* functions theory

3.1. Multiple gravity assist trajectory design strategy

This Chapter presents an overview of the work developed by Menzio et al. [14] regarding the adapted *syzygy* functions for the preliminary design of MGA ballistic trajectories. These concepts provide the necessary theoretical basis for understanding the innovative model extension developed in Chapter 5.

3.1.1. *Syzygy* functions for sighting

The term *syzygy* comes from suzugos, "conjunction" in the Ancient Greek, and indicates the alignment of three or more planets, that is, when the three planets form a straight line while revolving around the Sun.
The idea consists in using the *syzygy* function to find the planetary alignment considering the dynamics of the bodies, in order to model the interplanetary trajectory. First of all the sighting problem is considered, under the assumptions of co-planar, circular planetary orbits. The line condition can be imposed by expressing the planar position (in terms of x and y) of the three planets in terms of orbital radius.

$$\begin{cases} x_1 = r_1 cos(n_1 t + \phi_1(t, t_0)) \\ y_1 = r_1 sin(n_1 t + \phi_1(t, t_0)) \end{cases} \tag{3.1}$$

$$\begin{cases} x_2 = r_2 cos(n_2 t + \phi_2(t, t_0)) \\ y_2 = r_2 sin(n_2 t + \phi_2(t, t_0)) \end{cases} \tag{3.2}$$

$$\begin{cases} x_3 = r_3 cos(n_3 t + \phi_3(t, t_0)) \\ y_3 = r_3 sin(n_3 t + \phi_3(t, t_0)) \end{cases} \tag{3.3}$$

Where r_i is the i-th planet's orbital radius, n_i is the i-th planet's angular velocity and $\phi_i(t, t_0)$ is the initial angular position of the planet.
Imposing the passage of the line by 3 points, that is:

$$\frac{x_1 - x_2}{y_1 - y_2} = \frac{x_3 - x_2}{y_3 - y_2} \tag{3.4}$$

rearranging and exploiting the trigonometric identity, the *syzygy* function can be written as the summation of sine of the angular difference of the planets [14]:

$$f_{123} = \frac{sin[(n_3 - n_2)t + \phi_{32}]}{r_1} + \frac{sin[(n_1 - n_3)t + \phi_{31}]}{r_2} + \frac{sin[(n_2 - n_1)t + \phi_{21}]}{r_3} \tag{3.5}$$

The goal now is to find the zeros of the function, that represents the planetary alignment. The solution will be given in terms of departure time. The *syzygy* doesn't account for orbital eccentricities and inclinations, but nevertheless gives a general idea of the patterns of alignment expected for planets with roughly circular and co-planar orbits [14].
"Perfect" solutions will make the *syzygy* function tend to zero, meaning a precise alignment is taking place, but for the sake of sighting alone, also solutions close to zero can be considered good enough ones.

3.1.2. Time augmentation: the Hohmann-*syzygy* functions

The line condition hypothesis is preserved, however, because of the required time of flight to reach all the planets, an updated definition of *syzygy* function must be introduced. This constraint forces the transfer to follow an Hohmann trajectory. The *syzygy* function can be written this time as [14]:

$$F(t) = \sum_{j=1}^{N_{pl}-1} (r_j r_{j-1} sin(n_{j+1} tof_j + \phi_{j,j+1}(t_j)))^2 \tag{3.6}$$

This strategy has a crucial drawback: satisfying the line condition doesn't ensure that the planets are in a feasible Hohmann configuration, which requires to ride on an unfeasible high-eccentricity orbit, that leads to clearly unfeasible close encounters, see figure 3.1.

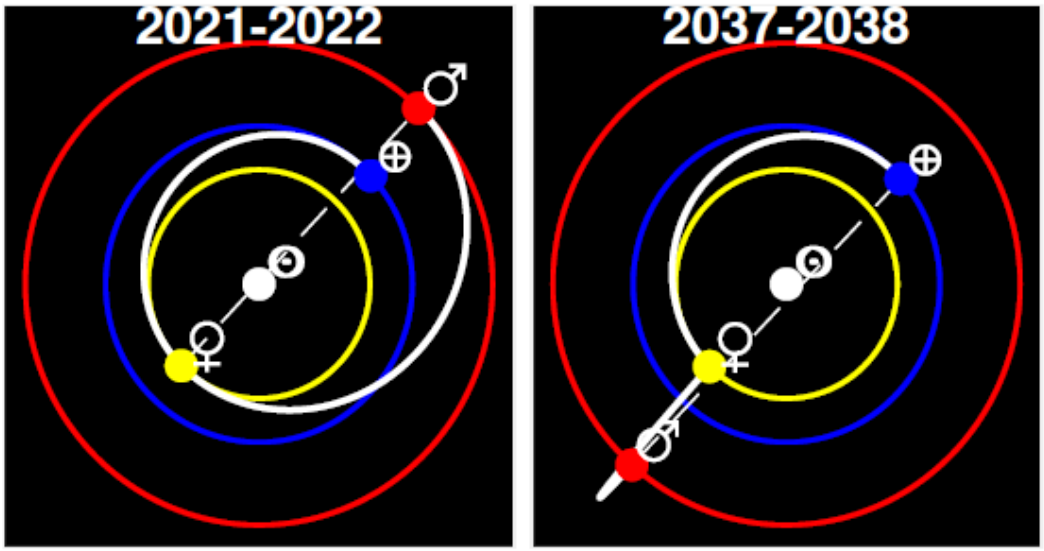

Figure 3.1: The solutions identified from the Hohmann-syzygy function: on the left the correct disposition of the planets allows to perform two Hohmann to reach Mars, on the right having Venus and Mars from the same side results the bouncy trajectory. The image presented is taken from Menzio et al. work [14].

Since Hohmann transfers are characterized by tangential and parallel velocities at arrival and departure, the flybys would give no contribution. Indeed, a ballistic (unpowered) close encounter's effect is limited to a pure rotation of the incoming planetocentric velocity, but in the case described, both pre and post encounter velocities share the same direction, meaning that the effect of the flyby is overall null.
To obtain the wanted Hohmann GA the planetocentric velocity vector would need a variation in magnitude, making it necessary to perform powered manoeuvres, making the trajectories not optimal design-wise for the problem at hand, since only ballistic trajectories are being considered.
For not perfect solutions (imprecise alignment and therefore not perfectly tangential terminal points), the flyby does give a contribution by rotating the incoming planetocentric velocity, but it changes the overall geometry not respecting the line condition constraint.

3.1.3. Conic-*syzygy* functions

In order to solve the problems highlighted in the section above, the line condition must be abandoned, in favour of a conic one. The idea behind the "new" *syzygy* developed by Menzio et al. is going to be to find where the time of flight meets the time of interception, to shape the trajectory [14].

Considering now the case of 2 interplanetary legs and a flyby, the number of variables has increased: three parameters for each leg, two times of flight, the departure date and one for the flyby, either the pericenter radius, the turn angle δ, or the impact parameter b. Since the number of variables is greater than the number of unknowns, the problem formulated this way admits infinite solutions. Therefore an additional constraint should be taken into account to balance the two and an orbital parameter must be chosen parametric to shape the orbit: only tangential arcs are considered at the terminal points (departure and arrival, this imposes the true anomaly at departure and at arrival that will be apsidal points) and the eccentricity is chosen as a parameter.
Under these assumptions, by defining the eccentricity and the departure date, the whole interplanetary leg is fully defined:

$$\theta(t_1) = 0 \tag{3.7}$$

$$a_1(e_1) = \frac{r_1}{1 - e_1} \tag{3.8}$$

$$\omega(t_1) = L_1(t_1) \tag{3.9}$$

Where $\theta(t_1)$ is the true anomaly on the transfer orbit (perigee), a_1 is the semi-major axis, e_1 is the eccentricity, ω_1 is the argument of perigee and L_1 is the true longitude of the departure planet.
It is then possible to define the aperture of the space triangle, identified by the difference in true anomaly [14]:

$$cos(\Delta L_{21}(e_1)) = \frac{1}{e_1}(\frac{r_1}{r_2}(1 + e_1) - 1) \tag{3.10}$$

Considering the equation above, it is possible to derive the minimum and maximum values that e_1 can assume (the right term of the equation has boundary values of +/- 1):

$$\begin{cases} e_{min} = \frac{r_2 - r_1}{r_2 + r_1} \\ e_{max} = 1 \end{cases} \tag{3.11}$$

Now, by means of Kepler equation, it is possible to compute the time of flight associated to the space triangle of aperture ΔL_{21} and therefore, the arrival time:

$$L_1 = n_1 t_1 + \phi_{01} \tag{3.12}$$

Where n_1 is the angular velocity of the departure planet and ϕ_{01} is the initial ($t_0 = 0$) angular position of the departure planet.

$$t_2 = t_1 + tof_{short} \tag{3.13}$$

This above represents the short transfer arc. It is possible to compute also the long transfer arc by considering the complementary tof with respect to the transfer leg orbital period.

$$tof_{long} = T_{orbit} - tof_{short} \tag{3.14}$$

The time of flight computation can lead to negative values for the former. These cases represent retrograde orbits.
Finally, the *syzygy* function, also called Feasible Transfer Condition FTC is defined as [14]:

$$FTC_{12} = cos(n_2 tof_{short} + \phi_{21}(t_1)) - cos(L_2(t_2) - L_1(t_1)) \tag{3.15}$$

The above function is subject to:

$$sin(n_2 tof_{short} + \phi_{21}(t_1)) > 0 \tag{3.16}$$

Where, n_2 is the angular velocity of the arrival planet, $\phi_{21}(t_1)$ is the phasing between the two planets at departure and $L_i(t_i)$ is the true longitude of the spacecraft on the transfer orbit.
When the above constraint is violated, the tof_{short} must be replaced with it's complementary, becoming:

$$FTC_{12} = cos(n_2 tof_{long} + \phi_{21}(t_1)) - cos(L_2(t_2) - L_1(t_1)) \tag{3.17}$$

As the FTC tends to zero, it will provide the feasible solution trajectory, in terms of a_1, e_1, t_1, t_2 and L_1.
For all the consequent interplanetary legs, the procedure followed is the same.

4 | Close encounters in the B-plane and flyby characterization

This chapter presents an overview of Öpik's theory of planetary close encounters, and an overview of the Orbital Mechanics relations for the flyby characterization. These concepts provide the necessary theoretical basis to understand the solution strategy presented in Chapter 7. A more thorough view on these subjects is presented in the works of Öpik [15], Carusi et al. [1] and Curtis [11].

4.1. Close encounters in the B-plane

Öpik's theory was derived based on a two-body linked-conics approximation and by including Tisserand's criterion [37]. In this framework, a massless particle is assumed to move on a Keplerian orbit relative to the Sun, until it enters the Sphere Of Influence (SOI) of a perturbing planet moving on a circular orbit. From a heliocentric perspective, the SOI is assumed to have an infinitesimal radius, implying that the orbits of the two objects are actually intercepting [26]. Once the particle reaches the SOI, the close encounter is modeled as an instantaneous rotation of the planetocentric velocity, without any change in its magnitude. The effect of the flyby is, therefore, a sudden variation of the keplerian parameters of the particle due to momentum exchange between the celestial body and the object.
Despite these simplifying assumptions, the true nature of planetary encounters is approximated quite well by this model and provides accurate initial estimates of the actual trajectory [6].

4.1.1. Encounter geometry

In this framework, the reference frame is planetocentric and all the orbital parameters are computed by using Jacobi normalized units. The period of the planet is 2π and its distance to the Sun is 1. It is also assumed that both the mass of the Sun and the

gravitational constant are equal to 1, such that the planet's velocity is 1 as well [18]. Let a, e, i, Ω, ω be the orbital elements of the particle's heliocentric orbit.
A planetocentric reference frame (X, Y, Z) is introduced, centred in the planet's center of mass such that the X-axis is directed from the Sun to the planet's position, the Y-axis is aligned with its direction of motion and the and Z-axis completes the right-handed triad. In this reference frame, the components of the planetocentric velocity vector of the particle are [1]:

$$\begin{cases} U_x = \pm\sqrt{2 - \frac{1}{a} - a(1-e^2)} \\ U_y = \sqrt{a(1-e^2)}cos(i) - 1 \\ U_z = \pm\sqrt{a(1-e^2)}sin(i) \end{cases} \tag{4.1}$$

$$U = \sqrt{U_x^2 + U_y^2 + U_z^2} \tag{4.2}$$

Identifying with T the Tisserand parameter of the particle's orbit, it can be shown a relation between T and the magnitude of the planetocentric velocity U:

$$U = 3 - T = \sqrt{3 - \frac{1}{a} - 2\sqrt{a(1-e^2)}cos(i)} \tag{4.3}$$

T is an invariant for the CR3BP, therefore, both U and T are conserved during a close encounter.
It must be noticed that, since normalized units are being used, the Tisserand parameter will depend on the particular planet on which the flyby is being performed. The encounter can be further represented geometrically by means of two angles, θ and ϕ, as in figure 4.1 where θ is the angle between U and the y-axis, and ϕ, the angle between the y-z plane and that containing U and the x-axis [17].
The velocity vector can be therefore expressed as:

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} Usin(\theta)sin(\phi) \\ Ucos(\theta) \\ Usin(\theta)cos(\phi) \end{bmatrix} \tag{4.4}$$

So that:

$$\begin{bmatrix} cos(\theta) \\ tan(\phi) \end{bmatrix} = \begin{bmatrix} \frac{U_y}{U} \\ \frac{U_x}{U_z} \end{bmatrix} \tag{4.5}$$

By substituting equations 4.1 and 4.3 into 4.5, it is possible to obtain the following expression [18]:

$$cos(\theta) = \frac{\sqrt{a(1-e^2)}cos(i) - 1}{\sqrt{3 - \frac{1}{a} - 2\sqrt{a(1-e^2)}cos(i)}} \tag{4.6}$$

Which can be reduced to:

$$cos(\theta) = \frac{1 - \frac{1}{a} - U^2}{2U} \tag{4.7}$$

Equation 4.6 shows that in a close encounter, the potential energy of the particle is fixed by the distance of the planet relative to the Sun. Instead, the kinetic energy is linked to the magnitude of the vectorial sum of the heliocentric velocities of the particle and the planet. For fixed U, this magnitude does only depends on θ. Therefore, the total energy (potential plus kinetic), and thus the semi-major axis, are only function of θ, as evidenced from equation 4.6 [18].

It is possible to derive also an expression for ϕ as a function of the particle's orbital elements. However this expression suffers from sign ambiguity and doesn't depend only on the particles semi-major axis:

$$tan(\phi) = \pm\sqrt{\frac{2a-1}{a^2(1-e^2)} - 1}\frac{1}{sin(i)} \tag{4.8}$$

If $0 < \phi < \frac{\pi}{2}$ the encounter is from inside the planet's orbit, and vice versa if $-\frac{\pi}{2} < \phi < 0$.

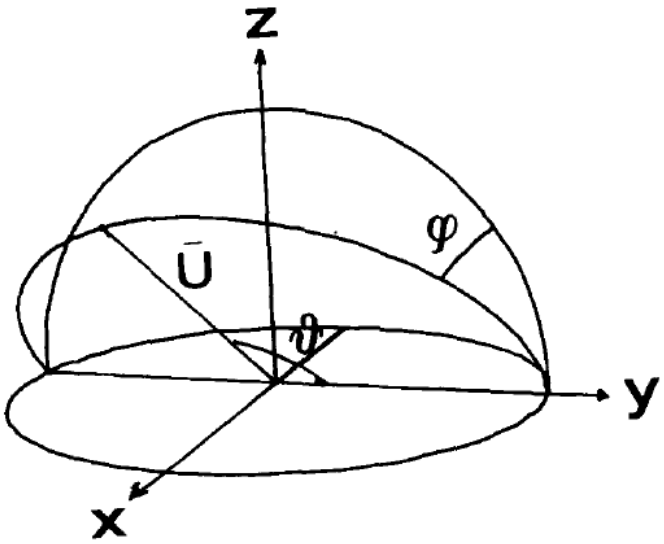

Figure 4.1: The frame of reference of the vector U. The origin is at the planet's centre, the y-axis is oriented in the direction of motion of the planet, the x-axis is in the opposite direction with respect to the sun, the z-axis is parallel to the planet's angular momentum vector. The direction of U is provided by the two angles θ and ϕ. This figure is taken from Carusi et al. work [1].

4.1.2. B-plane frame

The B-plane is defined as the plane orthogonal to **U** and containing the centre of the planet. In this context it is introduced the B-plane reference frame (ξ, η, ζ) such that the (ξ, ζ)-axes lie on the B-plane and η is perpendicular to it. In particular, ζ is parallel to the projection of the planet's velocity V_{pl} on the B-plane but with opposite direction and ξ completes a right-handed reference system, as shown in figure 4.2:

$$\hat{\eta} = \frac{U}{|U|} \tag{4.9}$$

$$\hat{\xi} = \frac{U \times v_{pl}}{|U \times v_{pl}|} \tag{4.10}$$

$$\hat{\zeta} = \hat{\xi} \times \hat{\eta} \tag{4.11}$$

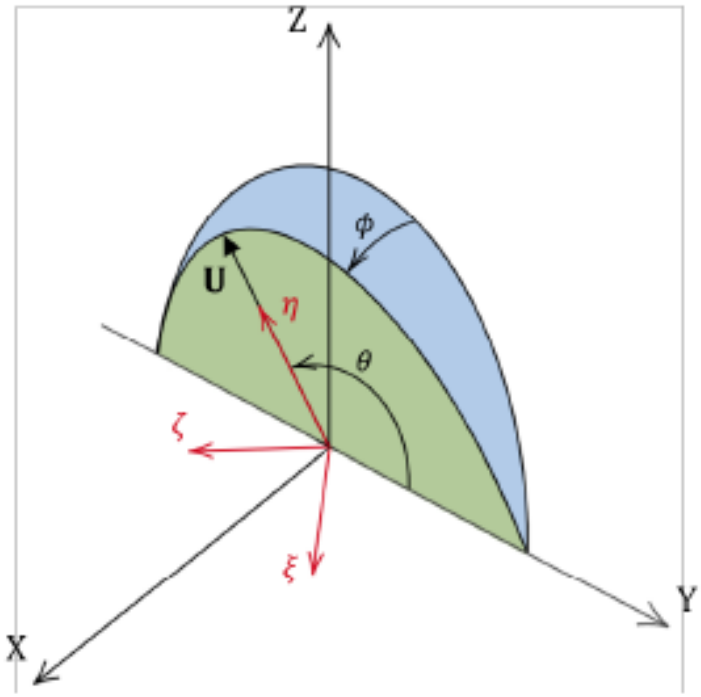

Figure 4.2: The figure shows the two reference frames presented: B-plane and planetocentric frame. This figure is taken from Campiti et al. work [18]

The ζ coordinate is associated to the shift in the time of arrival of the particle with respect to the planet, whereas ξ is related to the minimum distance between the two orbits [18]. When the particle crosses the B-plane, $\eta = 0$, and the quantity called impact parameter identifies the closest distance between the object and the planet:

$$b^2 = \xi^2 + \zeta^2 \tag{4.12}$$

4.1.3. B-plane deflection model

As already highlighted in the previous chapter, the flyby effect on the particle's velocity **U** is a pure, instantaneous rotation which doesn't alter it's magnitude.
The magnitude of such rotation depends on the value of $|\mathbf{U}|$, on the impact parameter and on the planet's mass m in units of the Sun's mass.
Introducing the characteristic length $c = \frac{m}{U^2}$, the deflection angle γ can be computed as:

$$tan(\frac{\gamma}{2}) = \frac{c}{b} \tag{4.13}$$

The encounter generates a rotation of $\mathbf{U}$ of γ in the direction given by the direction angle ψ:

$$\begin{bmatrix} bsin(\psi) \\ bcos(\psi) \end{bmatrix} = \begin{bmatrix} \xi \\ \zeta \end{bmatrix} \tag{4.14}$$

Following the procedure proposed by Valsecchi et al. [17], it is possible to compute the post encounter angle ϕ', by means of the angle χ.:

$$tan(\chi) = \frac{sin(\psi)sin(\gamma)}{cos(\gamma)sin(\theta) - sin(\gamma)cos(\theta)cos(\psi)} \tag{4.15}$$

$$\chi = \phi - \phi' \tag{4.16}$$

Finally, the post encounter angle θ' is computed as:

$$cos(\theta') = cos(\theta)cos(\gamma) + sin(\theta)sin(\gamma)cos(\psi) \tag{4.17}$$

At this point, the flyby can be fully characterized. Starting from the geometry considerations made so far, it is possible to identify the point on the B-plane corresponding to the close encounter considered:

$$\zeta = \frac{(b^2 + c^2) * cos(\theta')}{2csin(\theta)} - \frac{(b^2 - c^2) * cos(\theta)}{2csin(\theta)} \tag{4.18}$$

$$\xi = \sqrt{b^2 - \zeta^2} \tag{4.19}$$

Furthermore, it is possible to exploit the computations made so far to evaluate the post encounter $\mathbf{U}'$ components, which is represented in figure 4.3, together with the post encounter orbital parameters a', e' and i':

$$\begin{bmatrix} U'_x \\ U'_y \\ U'_z \end{bmatrix} = \begin{bmatrix} Usin(\theta')sin(\phi') \\ Ucos(\theta') \\ Usin(\theta')cos(\phi') \end{bmatrix} \tag{4.20}$$

$$a' = \frac{1}{1 - U^2 - 2U'_y} \tag{4.21}$$

$$e' = \sqrt{U^4 + 4U_y^{2\prime} + U_x^{2\prime}(1 - U^2 - 2U_y^{2\prime}) + 4U^2U'_y} \tag{4.22}$$

$$i' = atan(\frac{U'_z}{1 + U'_y}) \tag{4.23}$$

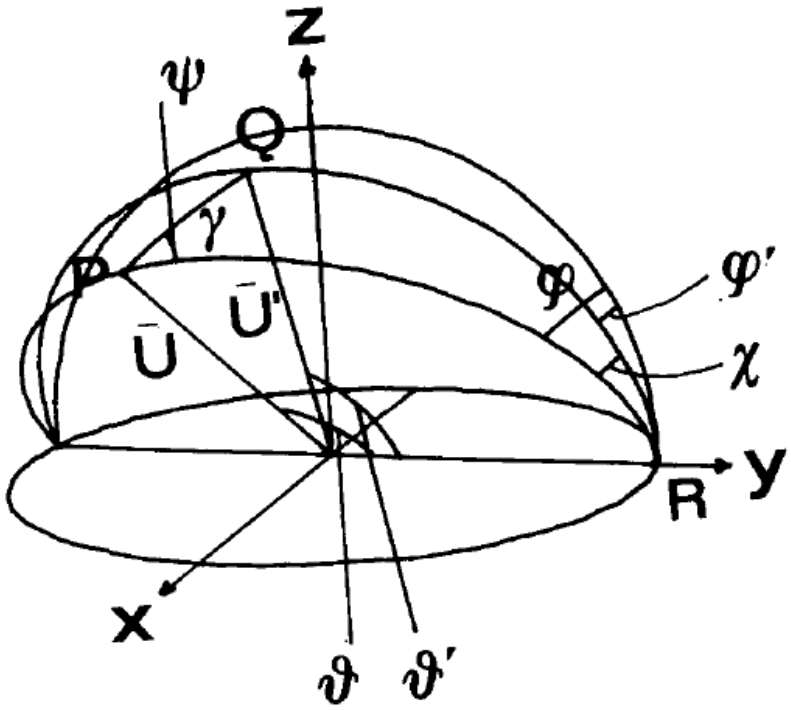

Figure 4.3: After the encounter the vector **U** is rotated by an angle γ in the direction given by ψ. This last is the angle (counterclockwise) from the meridian RP, containing the velocity vector. After rotation, the direction of **U** is given by the angles θ' and ϕ'. This figure is taken from Carusi et al. work [1].

For the problem at hand, which considers all the planets on co-planar, circular orbits, the close encounter characterization assumes a simplified description, since the rotation of the incoming velocity vector **U** is such that also the outgoing velocity vector **U**$'$ lies on the ecliptic plane. This means that the direction angle ψ is fixed and is equal to π. This way:

$$\chi = 0 \tag{4.24}$$

$$\phi' = \phi \tag{4.25}$$

$$cos(\theta') = cos(\theta)cos(\gamma) - sin(\theta)sin(\gamma) \tag{4.26}$$

$$\begin{bmatrix} U'_x \\ U'_y \\ U'_z \end{bmatrix} = \begin{bmatrix} Usin(\theta')sin(\phi') \\ Ucos(\theta') \\ 0 \end{bmatrix} \tag{4.27}$$

$$i' = i = 0 \tag{4.28}$$

Indeed, equation 4.26 represents a planar rotation of the angle θ by an angle γ. A similar result could have been obtained by applying the rotation of γ to vector $\mathbf{U}$ by means of a rotation matrix. The obtained result is also in accordance with the corresponding trigonometric expression of $cos(\theta + \theta')$.

4.2. Flyby characterisation

The B-plane and Öpik's formalism characterised the flyby in terms of variation of a, e (planar case). The values of a' and e' (post-encounter values) computed by these means automatically respect the conservation of the Tisserand parameter.
The argument of perigee uniquely identifies the orientation of the transfer orbit, and the true anomaly uniquely identifies the spacecraft's flight path angle at encounter, limiting the feasible variation of orbital parameters obtainable by means of the flyby.
The idea behind this section is to employ the classical Orbital Mechanics relations for the characterisation of the flyby, in order to verify if the deflection studied by means of the B-plane is actually feasible for the unique incoming interplanetary leg designed with the *syzygy* algorithm.
In order to do so, the angular momentum of the planet encountered and of the spacecraft pre and post flyby must be evaluated:

$$h_{s/c} = \sqrt{\mu a(1 - e^2)}; \tag{4.29}$$

$$h'_{s/c} = \sqrt{\mu a'(1 - e^{2\prime})}; \tag{4.30}$$

$$h_{pl} = \sqrt{\mu r}; \tag{4.31}$$

From these quantities it is possible to evaluate the radial and orthogonal components of the spacecraft and planet's velocities. Note that in the assumption of planetary circular orbits, the radial component of the planet velocity is always null:

$$v_r = \frac{\mu}{h} esin(\theta) \tag{4.32}$$

$$v_o = \frac{\mu}{h}(1 + ecos(\theta)) \tag{4.33}$$

At this point, it is possible to compute the planetocentric relative velocity (velocity at infinity) of the spacecraft, first evaluating it's radial and orthogonal components:

$$v_{r\infty} = v_{rpl} - v_{rsc} \tag{4.34}$$

$$v_{o\infty} = v_{opl} - v_{osc} \tag{4.35}$$

$$\mathbf{v}_\infty = \begin{bmatrix} v_{r\infty} \\ v_{o\infty} \end{bmatrix} \tag{4.36}$$

By fixing the impact parameter, it is possible to completely define the close encounter hyperbolic trajectory in terms of pericenter radius r_p:

$$r_p = \frac{-\mu_{pl}}{v_\infty^2} \pm \sqrt{\frac{\mu_{pl}^2}{v_\infty^4} + b^2} \tag{4.37}$$

Where μ_{pl} is the gravitational parameter of the flyby planet, v_∞ is the magnitude of $\mathbf{v}_\infty$ and b is the impact parameter. The sign ambiguity can be ruled out by remembering that the perigee radius r_p must be a positive value, so:

$$r_p = \frac{-\mu_{pl}}{v_\infty^2} + \sqrt{\frac{\mu_{pl}^2}{v_\infty^4} + b^2} \tag{4.38}$$

The flyby hyperbola will feature an eccentricity given by:

$$e = 1 + \frac{r_p v_\infty^2}{\mu_{pl}} \tag{4.39}$$

and turn angle δ computed as:

$$\delta = 2asin(\frac{1}{e}) \tag{4.40}$$

The data gathered by means of these computations can be exploited in order to evaluate the feasibility of the close encounter prescribed by the B-plane deflection model, by checking if the perigee radius r_p is actually greater than the planetary radius of the flyby planet. The feasibility condition therefore reads:

$$r_p > r_{planet} \tag{4.41}$$

Any flyby that respects this condition is considered feasible.

5 | Conic *Syzygy* generalisation

This Chapter presents an innovative generalisation of the approach described in Chapter 3 for hyperbolic orbits and for not-tangential terminal points. The aim is to obtain a more flexible algorithm, able to find a higher number of feasible solutions and adapt it to use it jointly with the B-plane formalism described in Chapter 4.

5.1. Generalized *syzygy*

5.1.1. Elliptic case

The conic *syzygy* algorithm developed by Menzio et al. [14] and proposed so far is based on a strong assumption: the transfer trajectory has one of it's terminal points in the absidal points. The aim at this point is to remove this hypothesis, in order to broaden the solution space and perform a more thorough search for the optimal trajectory.
Removing the constraint leaves the problem with an additional degree of freedom, therefore, a new parameter must be chosen to fully determine the transfer leg. The departure true anomaly on the transfer orbit θ_1 is chosen, enabling departure and arrival not in the absidal points. The proposed procedure is the same as the one presented in section 3.1.3, but some adjustments must be taken in order to successfully implement it.
Two consideration must be made regarding the computation of the aperture of the space triangle $\Delta\theta_{12}$, that was defined in equation 3.9:

$$\Delta\theta_{21}(e_1) = acos(\frac{1}{e_1}(\frac{r_1}{r_2}(1+e_1)-1)) \tag{5.1}$$

This expression is valid only if r_1 is actually the pericenter radius of the transfer orbit. In this generalized approach therefore, r_1 must be replaced by:

$$r_p = a_1(1-e_1) \tag{5.2}$$

Leading to:

$$\Delta\theta_{21}(e_1) = acos(\frac{1}{e_1}(\frac{a_1}{r_2}(1-e_1^2)-1)) \tag{5.3}$$

$$\theta_2 = \Delta\theta_{21}(e_1) \tag{5.4}$$

The inverse cosine function in equation 5.3 returns values in the interval $[0, \pi]$. The real values of $\Delta\theta_{12}$ that satisfy equation 5.3 are instead two, the "lost" value being in the interval $[-\pi, 0]$ and it's value is:

$$\Delta\theta_{2lost} = -\Delta\theta_{21} \tag{5.5}$$

Not considering this second value halves the possible solutions produced by the algorithm and therefore must be taken in consideration. This value is labeled as "lost" because the inverse cosine function, by the way it is defined, is not able to identify it. Once the aperture $\Delta\theta_{21}$ has been determined (therefore also $\Delta\theta_{lost}$), as shown in Chapter 3, it is possible to evaluate the correspondent time of flights by means of Kepler equation. In this work, only prograde orbits will be considered, and the retrograde orbits will be considered as unfeasible solutions.

If $\theta_1(t_1) < \pi$:

$$\begin{cases} if\theta_1(t_1) > \theta_2(t_2) \\ tof_{short} = tof_{arrlost} - tof_{dep} \\ tof_{long} = tof_{arr} + T_{orbit} - tof_{dep} \end{cases} \tag{5.6}$$

$$\begin{cases} if\theta_1(t_1) < \theta_2(t_2) \\ tof_{short} = tof_{arr} - tof_{dep} \\ tof_{long} = tof_{arrlost} - tof_{dep} \end{cases} \tag{5.7}$$

And, if $\theta_1(t_1) > \pi$:

$$\begin{cases} if\theta_1(t_1) < \theta_{2lost}(t_2) \\ tof_{short} = tof_{arrlost} - tof_{dep} \\ tof_{long} = tof_{arr} + T_{orbit} - tof_{departure} \end{cases} \tag{5.8}$$

$$\begin{cases} if\theta_1(t_1) > \theta_{2lost}(t_2) \\ tof_{short} = tof_{arr} + T_{orbit} - tof_{departure} \\ tof_{long} = tof_{arrlost} + T_{orbit} - tof_{departure} \end{cases} \tag{5.9}$$

Where tof_{dep} is the time of flight from pericenter to the departure true anomaly, tof_{arr} to arrival true anomaly, $tof_{arrlost}$ to the "lost" arrival true anomaly θ_{2lost} and T_{orbit} the transfer orbit period, that depends only on the already computed semi-major axis.
Lastly, an improvement is performed on the shape of the FTC. Due to the harmonic nature of the cosine functions, the presented *syzygy* function tends to zero either when the arguments of the two cosines are equivalent, or when one it's the opposite of the other. This second scenario represents transfer orbits which satisfies the FTC but leads to the target planet only if travelled in retrograde motion, and must be therefore discarded for the model presented, since only prograde orbits are considered.
In order to avoid such issue, the cosines are removed from the FTC, leading to:

$$FTC_{12} = n_2 tof_{short} + \phi_{21}(t_1) - \theta_2(t_2) + \theta_1(t_1) \tag{5.10}$$

$$FTC_{12} = n_2 tof_{long} + \phi_{21}(t_1) - \theta_2(t_2) + \theta_1(t_1) \tag{5.11}$$

The first representing the short transfer arc (aperture $< \pi$) and the latter the long transfer arc. Both solutions might lead to a feasible transfer (such that the FTC tends to zero), so both are considered as solutions of the presented algorithm.
By removing the cosines from the FTC expression, the FTC will tend to zero only if the spacecraft and the target planet share the same angular position, or in other words, the apertures of the space triangles spanned by both the planet and the spacecraft are equal. Indeed, $n_2 tof + \phi_{21}(t_1)$ represents the angle spanned by the target planet with respect to the position of the departure planet. $\phi_{21}(t_1)$ is the initial phasing (initial angle) between target and departure planet; $n_2 tof$ instead represents the angle travelled by the target planet during the duration of the transfer leg. The summation of the two identifies the overall aperture of the space triangle, that must be matched by the angle travelled by the spacecraft, namely $\Delta\theta_{21}$. To conclude, the improved FTC represents the difference, in terms of angular position, between the spacecraft and the target planet. As the FTC tends to zero, so does the difference of angular position between the spacecraft and the target planet, meaning that the two objects share the same spatial position.

5.1.2. Hyperbolic case

Among the conic sections often exploited during interplanetary missions, a very important role is played by hyperbolic orbits. Apart from being fundamental in the study of close approaches to planets within the SOI, they are the only solution, being open orbits, for missions whose objective is reaching as far as possible from the Sun. For this reason they have been included in the generalization of the algorithm proposed in Chapter 3.

The algorithm in the hyperbolic case is essentially equivalent to the elliptic case, with the only differences being in the selection of departure and arrival true anomalies and time of flight computations. This is a consequence of the fact that hyperbolic orbits are open.
Regarding the first issue, $\theta_1(t_1)$ must be chosen so that it respects the asymptotes constraint, being:

$$\theta_\infty = acos(-\frac{1}{e_1}) \tag{5.12}$$

The true anomalies of the asymptotes are therefore $(-\theta_\infty, \theta_\infty)$ and both θ_1 and θ_2 must be inside this bounds.
As for the time of flight, since the orbit is open and only prograde solutions are considered, only certain configurations will lead to feasible transfers. More in particular, if $\theta_1 > 0$:

$$\begin{cases} if \theta_1(t_1) < \theta_2(t_2) \\ tof_{short} = tof_{arr} - tof_{dep} \\ tof_{long} = retrograde \end{cases} \tag{5.13}$$

$$\begin{cases} if \theta_1(t_1) > \theta_2(t_2) \\ tof_{short} = retrograde \\ tof_{long} = retrograde \end{cases} \tag{5.14}$$

Instead, if $\theta_1 < 0$:

$$\begin{cases} if \theta_1(t_1) > \theta_{2lost}(t_2) \\ tof_{short} = tof_{arr} - tof_{dep} \\ tof_{long} = retrograde \end{cases} \tag{5.15}$$

$$\begin{cases} if \theta_1(t_1) < \theta_{2lost}(t_2) \\ tof_{short} = tof_{arrlost} - tof_{dep} \\ tof_{long} = tof_{arr} - tof_{dep} \end{cases} \tag{5.16}$$

Again, tof_{short} representing the short transfer arc (aperture $< \pi$) and tof_{long} the long transfer arc. Both solutions might lead to a feasible transfer (such that the FTC tends to zero), so both are considered as solutions of the presented algorithm.

5.2. Solution space pruning

Having outlined the procedure used to evaluate the FTC, it is possible now to exploit it to search for the solution.

Starting from the first interplanetary leg, the aim is to parameterize the problem by means of the departure time, eccentricity and true anomaly measured on the transfer orbit. By generating a discretization for the chosen parameters it is possible, for any given departure time, to consider all the possible combinations of eccentricity and true anomaly. The conic *syzygy* algorithm then returns the orbital parameters, FTC and tof of the computed transfer orbit.
At this point, the number of computed solutions can be very large, especially when a thorough search is performed. In order to reduce the computational weight of the algorithm, a tolerance on the maximum acceptable value of the FTC is added, to discard all the solutions that violate it. Care must be taken when choosing the tolerance value, to avoid discarding acceptable solutions.
Some simple considerations can be made in order to select reasonable values for the FTC tolerance values. The length of a circular arc associated to an angle at center α of a circle of radius r can be evaluated as:

$$l = \alpha r \tag{5.17}$$

By the way the FTC is defined, the function is dimensional, and shows a value in radians. As already shown in section 5.1.1, the FTC represents the difference in angular position between the spacecraft and the target planet. It is therefore possible to evaluate the error in terms of distance along the planet's circular orbit, associated to a particular value of the FTC by:

$$err = aFTC \tag{5.18}$$

Where a is the semi-major axis of the target planet.
For the sake of this thesis work, whose objective is only a preliminary design of the trajectory, to be used later with higher fidelity and more complex models, also small distance errors can be viewed as valuable solutions. In order to define a threshold value for the FTC that leads to acceptable distance errors, some simple considerations can be made.
In order for the close encounter to take place, the spacecraft must enter the SOI of a planet. It is therefore reasonable to select as a threshold for the FTC only values that lead to distance errors smaller than the SOI radius of the planet considered for the close encounter. The SOI radius of a given planet can be evaluated as:

$$r_{SOI} = a(\frac{m_{pl}}{m_{sun}})^{0.4} \tag{5.19}$$

Where m_{sun} is the Sun's mass, m_{pl} is the planet's mass and a is the planetary semi-major axis. It is then possible to evaluate the correspondent FTC value that leads to a distance

error equal to the SOI radius:

$$FTC_{thr} = \frac{r_{SOI}}{a} \tag{5.20}$$

In table 5.1 the values of the the planetary SOI radius and the correspondent FTC threshold that guarantees the distance error to be smaller than the SOI radius are reported:

Planet	SOI radius	FTC threshold
Mercury	$1.17 \cdot 10^5$ km	$2.02 \cdot 10^{-3}$
Venus	$6.16 \cdot 10^5$ km	$5.7 \cdot 10^{-3}$
Earth	$9.29 \cdot 10^5$ km	$6.2 \cdot 10^{-3}$
Mars	$5.78 \cdot 10^5$ km	$2.54 \cdot 10^{-3}$
Jupiter	$4.82 \cdot 10^7$ km	$6.19 \cdot 10^{-2}$
Saturn	$5.45 \cdot 10^7$ km	$3.8 \cdot 10^{-2}$
Uranus	$5.19 \cdot 10^7$ km	$1.8 \cdot 10^{-2}$
Neptune	$8.62 \cdot 10^7$ km	$1.91 \cdot 10^{-2}$

Table 5.1: SOI radius and correspondent FTC threshold values for the main celestial bodies in the Solar System.

Once the planetary sequence to be followed is defined, it is possible to select any value reported in table 5.1 to guarantee that the associated solution will lead the spacecraft within the SOI of the target planet.
As already outlined at the beginning of this section, the number of computed solutions can be very large. Any value smaller than the threshold values computed can be used to trim the solution space, by not considering solutions that violate this new limit value. For the single leg case presented in the section 5.3 the solution space can be considered as a hyper-rectangle of dimension 3. This is because the chosen parameters are three and each parameter is discretized on a arbitrarily selected interval. Without applying any space pruning and just using the threshold value of the FTC for selecting acceptable solutions, the number of computed solutions is 1633, compared to the pruned space (with threshold value for the FTC of 10^{-4}) that counts only 75 solutions, showing a reduction factor of 21.77.
Care must be taken in the selection of the pruning threshold for the FTC, since an aggressive pruning can lead to discarding feasible solutions. This process can be applied to any subsequent interplanetary leg, in order to keep the computational load reduced.

5.3. Algorithm validation: single transfer case

The proposed algorithm is validated against the solution of the Lambert problem.
As a first step, the generalized *syzygy* is used to solve the following problem: consider a mission from Earth to Mercury, with departure window during year 2023 and departure time counting 360 discrete values; the transfer orbit must be chosen with a specific eccentricity value, counting 200 discrete values spanning between $(0, 1)$, and must start at a selected true anomaly on the transfer trajectory, that can vary between $[0, 2\pi]$, counting 360 discrete values. A space pruning on the FTC value is performed, with FTC threshold selected at 10^{-4}. The solution must be computed in terms of tof and keplerian parameters of the transfer orbit.
Using the generalized *syzygy* algorithm and then Lambert solver, the solutions to the problem are computed. Among the computed solutions, the optimal one in terms of FTC minimization is reported in table 5.2, whose trajectory is shown in figure 5.1:

a	e	i	Ω	ω	θ	tof
$8.6137 * 10^8$ km	0.99	0	0	-0.2071	3.6054	$7.5475 * 10^6$s

Table 5.2: Keplerian parameters and tof of the optimal solution transfer orbit, computed by means of generalized *syzygy* algorithm.

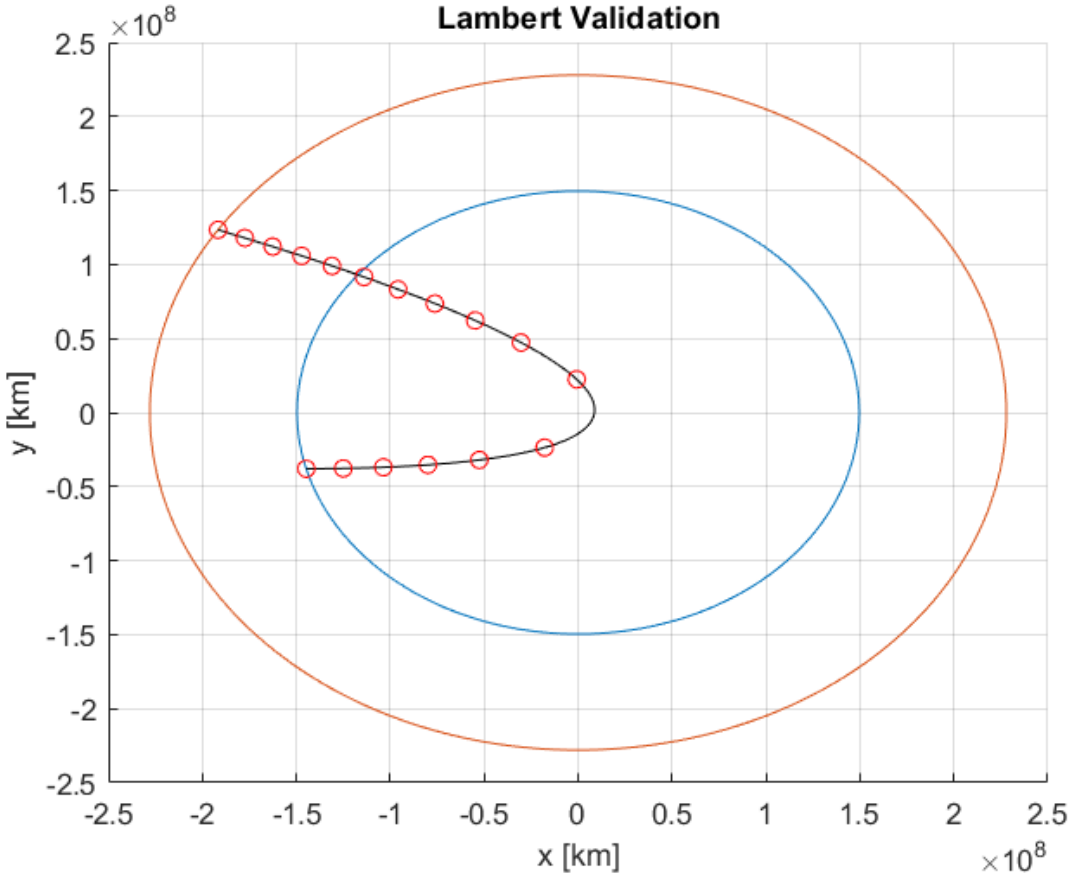

Figure 5.1: The trajectories computed by means of Lambert solver and generalised *syzygy* algorithm. In blue Earth's orbit, in red Mars' orbit, in black the transfer leg computed by means of the generalized *syzygy* algorithm and the red dots represent the Lambert arc.

The algorithm took 64.799 seconds to generate the presented solution, and the optimization process lasted only 0.0058 seconds.
It is then possible to evaluate the relative error of every *syzygy* solution with respect to the correspondent Lambert solution. For the semi-major axis for example, the relative error is evaluated as:

$$err_{rel} = \frac{a_{Lambert} - a_{syzygy}}{a_{Lambert}} \tag{5.21}$$

The relative error is evaluated for the semi-major axis, eccentricity and argument of perigee values and is presented in figures 5.2, 5.3 and 5.4:

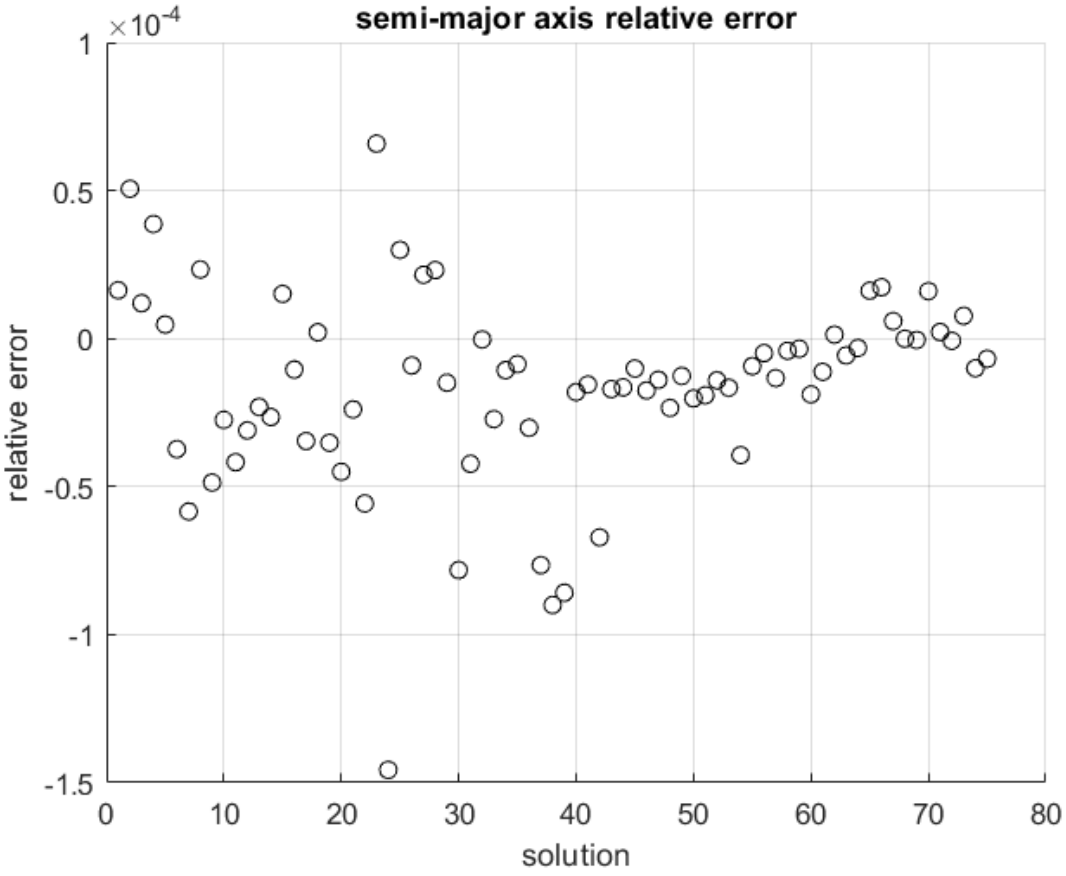

Figure 5.2: Relative error of the *syzygy* algorithm in the computation of the orbit's semi-major axis with respect to the Lambert solution's semi-major axis.

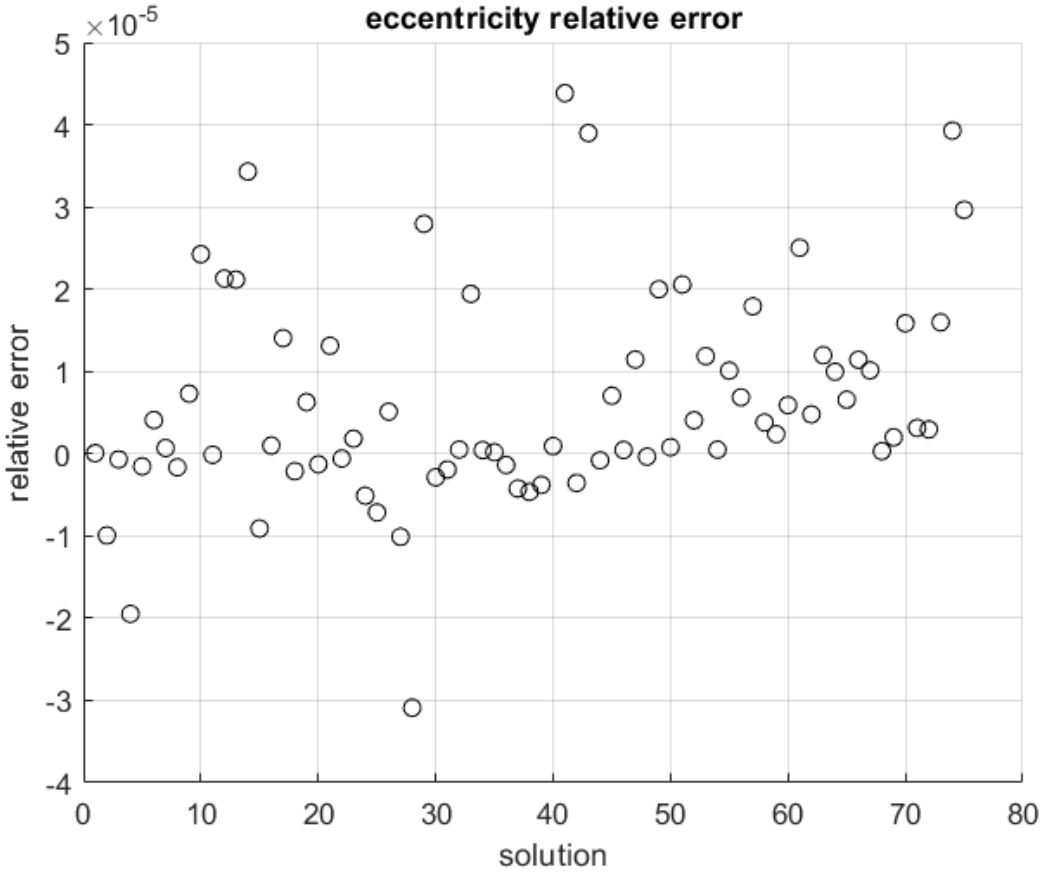

Figure 5.3: Relative error of the *syzygy* algorithm in the computation of the orbit's eccentricity with respect to the Lambert solution's eccentricity.

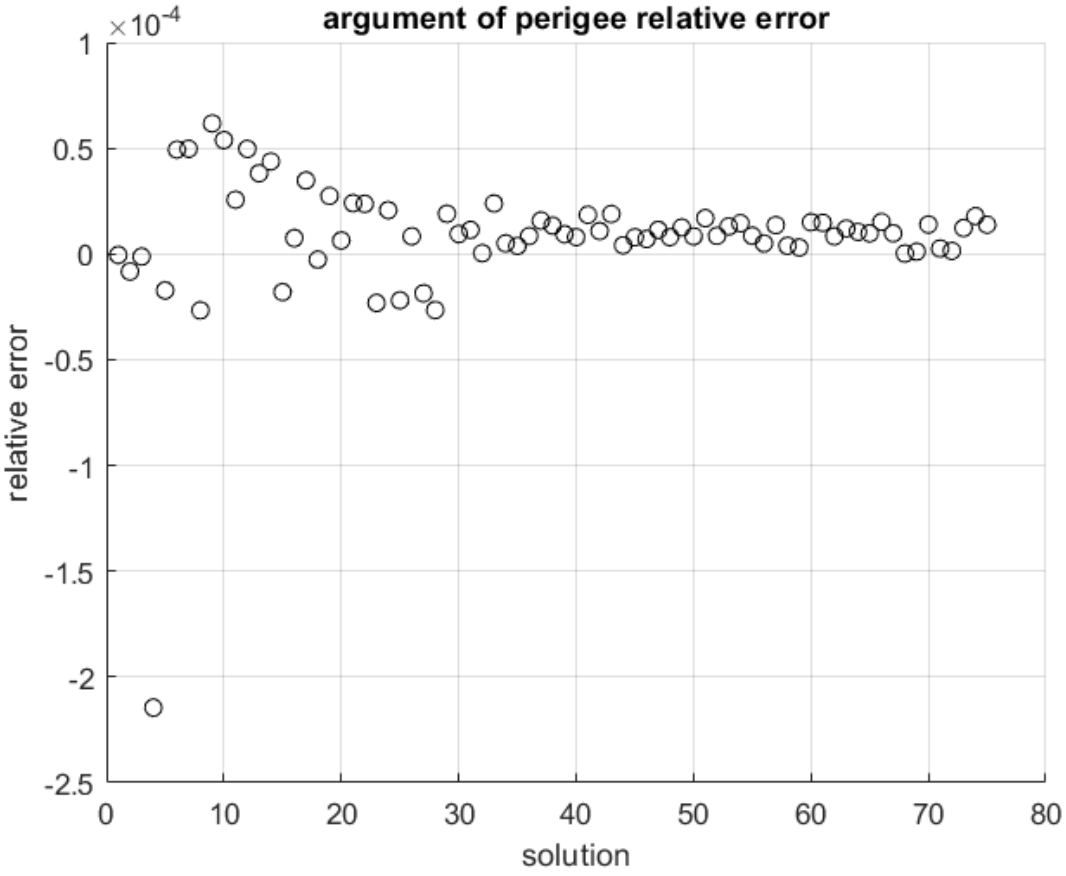

Figure 5.4: Relative error of the *syzygy* algorithm in the computation of the orbit's argument of perigee with respect to the Lambert solution's argument of perigee.

It is clear from the data presented that the relative error committed by the *syzygy* algorithm in the computation of the keplerian parameters of a single interplanetary leg is indeed negligible from a trajectory design point of view.

5.3.1. *Syzygy* algorithm computational efficiency

In the framework of the test problem presented in the previous section it of interest to evaluate the computational quickness of the *syzygy* algorithm compared to the speed of the Lambert solver in the solution of a single interplanetary arc.
Despite the parametrization of the problem must be defined differently for the two methods proposed, it is possible to choose a discretization for the Lambert parameters that spans a solution space of equivalent dimension with respect to the *syzygy* solution space. For the *syzygy* implementation the departure window is set in 2023 and counts 50 discrete values. The eccentricity is set to have 50 discrete values, varying between (0, 1) (only el-

liptical orbits are considered). As for the true anomaly, 50 discrete values between $[0, 2\pi]$ are chosen. A space pruning on the FTC is performed, with threshold value of 10^{-3}. The resulting discretization enables the *syzygy* algorithm to evaluate 125000 combinations of parameters, each one representing a unique interplanetary arc.
The Lambert solver implementation requires to set as parametric only 2 quantities. In this case the departure time and the tof are chosen. The same discretization presented for the *syzygy* solution is used for the departure date, whereas 2500 discrete values are defined for the tof, spanning from half to ten times the Earth's orbital period.
The *syzygy* algorithm took 0.47 s to evaluate all possible combinations, compared to 313.81 s of the Lambert solver. From this data it would appear that the *syzygy* algorithm has the edge on the Lambert solver computationally, but a consideration must be made: the *syzygy* algorithm, out of 125000 trajectories, was able to identify only 5 solutions that comply with the minimum FTC threshold, and that therefore represent feasible solutions. All the orbits generated by the Lambert solver are instead feasible in terms of encounter precision.
It is possible to conclude that the *syzygy* algorithm is able to span wide solution spaces in a very reduced amount of time compared to the Lambert solver, but it doesn't guarantee a high number of solutions compliant with the minimum FTC threshold, whereas the Lambert solver is more efficient in this task. The great quickness of the *syzygy* algorithm must be therefore exploited by performing searches of very wide solution spaces, in order to match the efficiency of the Lambert solver in finding feasible solutions.

6 | Dynamic programming approach

This Chapter is based on the work developed by Campiti et al. [18].
Dynamic programming is a useful mathematical technique for making a sequence of interrelated decisions. It provides a computationally efficient method for finding optimal solutions to problems that can be formulated as multi-stage decision processes, where the decisions made at one time influence the later available choices. In contrast to linear programming, there does not exist a standard mathematical formulation of "the" dynamic programming problem. Rather, dynamic programming is a general type of approach to problem solving, and the particular equations used must be developed to fit each situation [16].
It was first introduced by Richard Bellman in the 1950s and has since found application in a variety of disciplines, from engineering to economics. In space mission design, several works have demonstrated the effectiveness of this technique for different optimisation problems. In Lin [23], a dynamic programming method is used to optimise the total propellant consumption required to control of the orbital altitude of a space station. The optimisation of low-thrust trajectories is a typical problem approached with dynamic programming, as it allows to reduce the high dimensional problem into a succession of low dimensional sub-problems. In this direction, Colombo et al. [9] proposed an efficient algorithm based on differential dynamic programming that computes an optimal feedback control law by discretizing the dynamics in correspondence of a fixed number of decision times. Alternative strategies based on this approach can be found in Nugnes and Colombo [31] and Lantoine and Russel [24]. In Appendix A, one of the most popular dynamic programming problems is presented in order to introduce the reader to the general notation and terminology, and explain the logic behind the approach. Despite the renowned performance of dynamic programming for stochastic problems, this Chapter will focus on deterministic decision processes as they reflect the structure of the problem presented in the thesis.

6.1. General dynamic programming approach

Dynamic programming can be used to solve efficiently a variety of problems, that must all share three essential features in order to be tackled effectively by means of a dynamic programming approach: states, stages and a recursive relation.
Firstly, the procedure requires to structure the problem as a multistage decision process, where the stages identify the points in which a policy decision is required. The problem is tackled starting from either the first or last stage and proceeding one stage at a time, such that the solution to a stage is necessary to solve the next one. Any problem lacking this property cannot be dealt with dynamic programming. At each stage, the system might be in different possible conditions called states, and a policy decision has the effect to transform the current state into a different one associated with the next stage. The definition of the states is a crucial design parameter of the model, as the choice is not unique and the effectiveness of the method may change drastically depending upon which one is made. The problem must comply with the so called Markovian property [36], which prescribes that the chosen state must retain all the necessary information to determine the optimal policy henceforth. In addition, the number of state variables should be kept low, since the computational efficiency of dynamic programming rapidly decreases as the dimensionality of the state space increases. This property is referred to as "curse of dimensionality" [7] and significantly limits the applicability of the method in practise.
The process of making optimal decisions is based on Bellman's principle of optimality [7]: whatever the current state, the remaining decisions must constitute an optimal policy for all successive stages, regardless of the history of decisions made to arrive at that point. In practise, the optimisation is carried out by defining a recursive relationship that provides the optimal policy to any sub-problem, given the solution to all the smaller sub-problems. For the stagecoach problem, presented in Appendix A, the sub-problems are considered starting from the last stage and then moving backward one stage at a time, until the original problem is solved completely. This kind of logic is referred to as backward induction [33]. Some problems are instead approached more conveniently starting from the first stage basing the recursive relation on a forward induction process.

7 | Multiple unperturbed ballistic flyby trajectory design

7.1. Problem statement

The preliminary design problem addressed in this work can be expressed as: given a spacecraft departing from a given position at an initial time, evaluate the optimal trajectory that leads the spacecraft to a target celestial object, performing ballistic flybys on a sequence of selected planets. By exploiting the GAMs, it is possible to reach locations for which a direct transfer would lead to unfeasible high energy levels.

7.1.1. Assumptions and considerations

The problem formerly presented is tackled under the following assumptions:

- patched-conics model
- planets move on circular, co-planar orbits
- no resonant flybys
- unpowered flybys
- no orbital perturbations

The patched conics model, despite the simplifications introduced, has been proven to work efficiently also for real-life missions, such as the Voyagers 1 and 2. It is therefore widely used in the preliminary design phase and once the solution is computed, it can be converted into a n-body trajectory by means of techniques such as differential corrections [8].
Regarding the strong assumption of circular and co-planar orbits, the choice was made in order to reduce the number of state variables to fully define each stage, therefore, to mitigate the "curse of dimensionality" effect that would lead to an extremely high com-

putational cost together with a long computational time. The main driver that led to the choice of dynamic programming as optimization tool was, in fact, to obtain a robust algorithm that produces solutions in a restricted amount of time. Nevertheless, this assumption can still be applied effectively to produce a preliminary solution to the problem. Since neither perturbations nor manoeuvres are considered, the effect of each GAM is a pure rotation of the incoming velocity $\mathbf{U}$, without changing its magnitude. For this reason, the target orbit must be ballistically reachable, otherwise the design would not be effective. This condition is equivalent to saying that the initial and final orbits must feature the same Tisserand parameter [3]. Discarding all kinds of perturbations inevitably leads to moderately approximated solutions, since the planetocentric dynamics can be altered in a non-negligible way by disturbing effects. Still, the main objective of the work presented is to prove the effectiveness of the *syzygy*/dynamic programming couple for solving the preliminary design problem, that deals mostly with heliocentric trajectories, on which the perturbations have a more limited effect. A method to include such effects in the computation of the B-plane quantities has already been developed by Masat et al. [3], so, future works could address the inclusion of perturbations in the model in order to refine the dynamics within the SOI.

In the problem at hand, each interplanetary transfer leg represents a stage of the problem. In order to fully describe each stage, a choice of state variables must be operated. Being each stage a keplerian trajectory, a straightforward idea would be to use keplerian orbital parameters as variables, or a restricted set of them at least. Following Menzio et al. the eccentricity together with the departure date represent the best option [14] as shaping parameter.

This thesis extends the model developed by Menzio et al. [14], relaxing the main constrain introduced in the previous work, namely that only tangential arcs are considered, introducing a new degree of freedom for the problem. Therefore, in order to fully determine the state at each stage, an additional parameter is chosen. More in particular, the departure true anomaly on the transfer orbit θ_1 for the first interplanetary leg. For all the subsequent arcs, the number of variables decreases: by selecting the new departure time and either the impact parameter b or deflection angle γ as parametric, the flyby problem is fully determined using the B-plane formalism. In other words, fixing either the impact parameter or deflection angle fixes the post encounter semi-major axis, eccentricity and therefore true anomaly.

A choice must be taken between using γ or b as a parameter, following some considerations. The deflection angle is defined in the interval $[-\pi, \pi]$, and if taken as a parameter it's value determines uniquely the post encounter angle θ', without relying on any inverse harmonic function, avoiding any quadrant ambiguity. Furthermore, being γ the deflection

angle, there is no need for a normalization process, which is instead required for the b, if chosen as a parameter. The associated value for the impact parameter can be directly evaluated by means of equation 4.12. In this work therefore, the parametrization of all the subsequent interplanetary arcs will consider the deflection angle γ.
Being all the orbits considered co-planar, inclination is null, and argument of perigee can be determined by geometric considerations, thus, the orbit is fully determined.
Finding the optimal sequence of GAMs therefore translates into the search of the optimal triplet (e_1, θ_1, t_1) for the first arc and the optimal series of couplets (t_i, γ_i) that lead from the initial planet to the target one by performing an arbitrary number of flybys:

$$\text{Initial planet} \rightarrow (e_1, \theta_1, t_1) \rightarrow (t_2, \gamma_2) \rightarrow ... \rightarrow \text{Target planet}$$

The approach proposed in the thesis features a structure with a variable number (depending on the number of wanted GAMs) of nested for loops, used to analyse all possible combinations of departure date, eccentricity and departure true anomaly on the first transfer orbit, and combinations of departure date and deflection angle for the following arcs, that satisfy an arbitrary optimality policy.
This thesis extends on Menzio et al.'s work [14] and attempts to introduce an optimisation strategy by avoiding both the use of heuristic methods, which do not guarantee finding the optimal solution, as well as brute-force searches, which are computationally expensive. This is accomplished by proposing a deterministic dynamic programming approach that, on the one hand, always converges to the globally optimal solution of the problem and, on the other hand, allows to find such solution without systematically evaluating all the possible combinations of flyby sequences.

7.2. Dynamic programming approach

An optimal policy must be selected, such as minimum tof, flyby feasibility or encounter precision. The policy selected for the problem at hand consists in the search for precise transfers.
The transfer precision is described by the closeness of the FTC to zero.
Once the interplanetary arcs are computed by means of the $syzygy$ and the flyby effect is modeled with the B-plane formalism, the optimization algorithm evaluates, for each flyby, the precision of the transfer and selects the best one for each sub-problem.
In order to do so, the objective function is defined as:

$$f_n(s_n, x_n) = \sum_{i=1}^{N} |FTC| \tag{7.1}$$

As already stated, the function FTC represents the precision of the interplanetary transfer: the closer it is to zero, the more precisely the transfer arc leads the spacecraft to the target planet. As already mentioned, each stage is completely described by means of either two or three state variables, namely either the departure time and deflection angle or the eccentricity, departure time and departure true anomaly on the transfer orbit. Instead of the eccentricity, the transfer semi-major axis could have been used as a state variable, but the former is more convenient since its easier to define the boundary values between which it can vary.
By the way it is defined, the state of the problem is completely discrete, and the nature of the optimization is fully combinatorial. Therefore, the problem can be tackled by means of a deterministic dynamic programming approach. Thanks to the nature of dynamic programming itself, the optimal solution is not only computed for the whole problem, but also for all the sub-problems into which the model can be divided.

7.3. Solution procedure

First of all, it is necessary to define the layout of the problem, that is, the planetary sequence to be followed by the spacecraft. This task must be approached with care: transfers between planets whose orbits are characterized by very different energy levels (both high and low) would very likely lead to unfeasible flybys.
Once the outer layout is defined, the initial condition at departure is fully determined by propagation of the planetary ephemerides and then translated into a vector of keplerian elements $(a, e, i, \Omega, \omega, \theta)$.
Since the problem is fully combinatorial, the set of possible solutions is finite and so the problem presents a similar structure to the stagecoach problem presented in Appendix A. However, unlike this latter, the possible states through which the system can move are not known in advance. For this reason, the solution strategy consists in practise of two steps, the first of which is aimed at generating the possible intermediate states, the second focuses on searching the optimal solution to the problem.
State generation step: The procedure starts, in order to discretize the domain, by defining a set of departure times, a set on feasible eccentricity values for the transfer trajectory and a set of departure true anomalies on the transfer orbit.
The definition of the departure window is arbitrary, and the choice can be made in order to respect any design constraint selected, if any. As for the eccentricity and true anomaly sets, the former can vary between 0 and an upper limit arbitrarily chosen, and the latter varies between $[0,\ 2\pi]$. Indeed, depending on the particular stage, often only a subset of eccentricity values are feasible for the evolution of the system to the next stage. This

is because the aperture of the space triangle that characterizes the transfer cannot be arbitrarily big, but depends on the initial and consequent orbit chosen, as shown in Chapter 3. Nevertheless, the algorithm automatically discards all solutions with unfeasible values of eccentricity.
This procedure is iterated for every stage of the problem, with the only differences in that the departure times for stage n, with $n > 1$, are not arbitrarily generated but are computed by summing each departure time with it's correspondent computed time of flight, and in that the only other parameter needed for stage n, with $n > 1$, is the deflection angle γ. Thanks to the B-plane deflection model in fact, it is possible to evaluate the post-encounter semi-major axis a', and the eccentricity e' by fixing γ.
Dynamic programming:
The problem is then formulated with dynamic programming as follows:

- Number of stages N : total number of transfer legs (number of planets diminished by one).
- Staged numbering n: number of transfers completed.
- States s: (t_i, e_i, θ_i) for the first leg and (t_i, γ_i) for any subsequent arc.
- Policy decision at state n, x_n: state at stage $n-1$
- Objective function being in state s_n at stage n and making the decision x_n:

$$f_n(s_n, x_n) = \sum_{i=1}^{N} |FTC| \tag{7.2}$$

- Recursive formula:

$$f_n^*(s_n) = \min_{x_n \in S_{n-1}} f_n(s_n, x_n) \tag{7.3}$$

  where S_n is defined as the set of all possible states in which the system could be at stage n. In practise, at each stage and for each state equation 7.3 answers the question: "what is the best pre-encounter state to come from, if the system is currently in state s_n?".
  Starting from the departure, initially S_0 includes only s_0. As the algorithm proceeds, it computes the states s_1, s_2 and so on, and S_0 evolves progressively in S_1, S_2, etc. So, the generic j-th set of stages (S_j), is computed starting from all the stages in S_{j-1}. For each j-th stage, each triplet (t_1, e_1, θ_1) or couplet (t_i, γ_i) computed is saved as an element of S_j. This way, for any stage the problem is at, also the sub-optimal triplets and couplets are still considered as valid options for the next stage since,

similarly to the stagecoach problem, a sub-optimal decision at a previous stage could lead to an overall better (optimal) solution for the consequent stages.

Differently from the stagecoach problem presented in Appendix A, here the sub-problems are solved starting from the first stage and then moving forward one stage at a time, until the original problem is solved completely. This kind of logic is referred to as forward induction [33]. A block scheme representation of the algorithm structure is presented in figure 7.1.

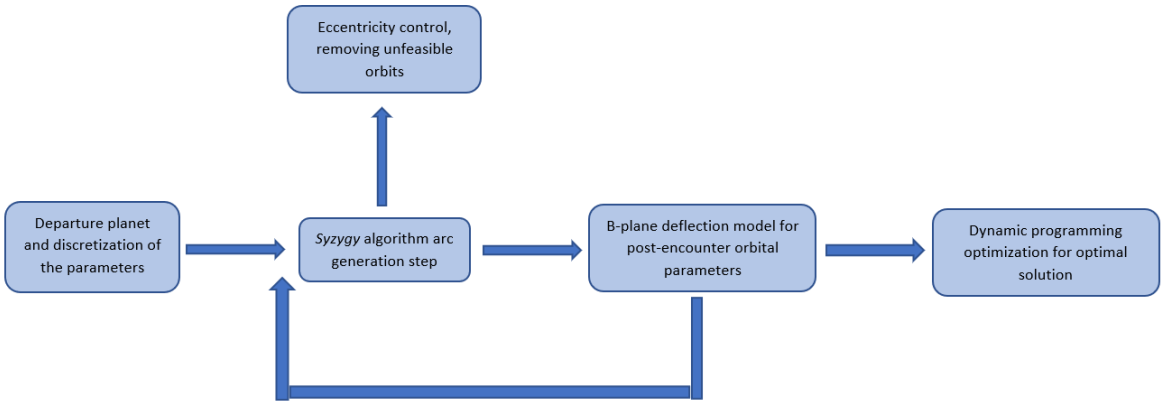

Figure 7.1: Block scheme representation of the algorithm main structure.

8 | Test cases and results

Several tests have been performed to assess the goodness of the procedure proposed so far. This Chapter focuses on three of them, accurately selected to show how the algorithm performs in the case of a single gravity assist and in a MGA scenario. For all reported performances in terms of computational time, please note that all tests have been implemented in Matlab® on a machine with an Intel® Core^{TM} i7-4750HQ CPU @2.00 GHz.

8.1. Voyager-1

Voyager-1 mission's primary objective was the exploration of Jupiter and Saturn. It was later extended to explore the outermost edge of the Sun's domain and beyond.
In order to reach far into interplanetary space without heavily relying on artificial means to vary the spacecraft energy level, gravity assist maneuvers were included in the mission profile. More in particular, the trajectory design is characterized by a close encounter with the giant Jupiter, in order to efficiently alter the orbital parameters of the spacecraft so to reach Saturn.
In this first test, the dynamic programming algorithm is asked to reproduce the design of the interplanetary phase of the mission. The baseline trajectory is the mission profile with launch on September 5 1977, close encounter with Jupiter on March 5 1979 and finally reaching Saturn on August 25 1981.
Since the trajectory involves no powered maneuvers, it offers an ideal test case for the proposed design strategy.
The algorithm is given only the wanted planetary sequence and an arbitrary launch window grid as an input, whereas the departure time and planets' positions at encounter are left as a free variables.
Different optimality policies could have been applied in the problem solution: to start, the optimal transfer in terms of transfer precision (optimality of the FTC) is computed.

8.1.1. Departure date grid choice

Since the assumptions adopted to model the problem introduce non-negligible differences with respect to the real dynamics of the problem, a more accurate study on the departure dates to use to initialize the algorithm must be taken.
In order to do so, the algorithm is run preliminary to identify the "best" departure times interval to be used for searching the most efficient solutions. Starting the departure time grid in 1975, a broad launch window of 15 years subdivided into 3000 equi-spaced time steps has been selected. As for the parametrization of the eccentricity, only elliptical orbits are considered for the first interplanetary leg, and 200 discrete values are generated spanning between 0 and 1 (excluded). Finally for the true anomaly, the discretization spans between 0 and 2π with 360 discrete values.
For the second interplanetary leg, both elliptic and hyperbolic orbits are considered, and the chosen parametrization for the deflection angle γ are 720 discrete value, spanning from $[-\pi, \pi]$. For both transfer arcs a space pruning technique has been applied to the FTC value, with a maximum violation value of 10^{-3}.
With the specified above values, the algorithm computed the associated feasible solutions, whose time of departure distribution and associated overall FTC value is shown in figure 8.1:

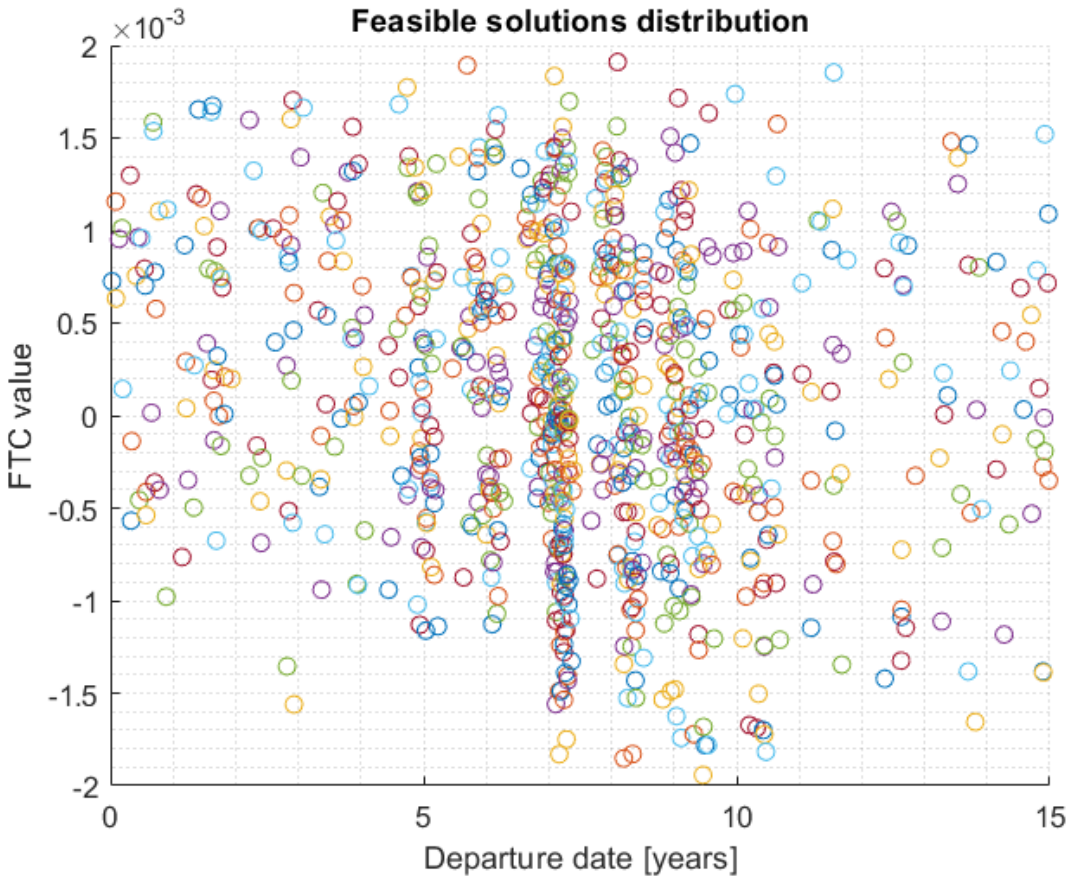

Figure 8.1: Distribution of the departure dates and their relative FTC value for feasible solution for the Voyager-1 mission.

Where the FTC, as already outlined, represents the difference, in terms of angular position, between the spacecraft and the target planet. The overall FTC value reported in figure 8.1 is the summation of each interplanetary leg's FTC, and represents the overall precision error of the computed solution. From figure 8.1 it is clear that a very big cluster of solutions that are composed by a feasible close encounter is located between years 7 to 10 from the initial date considered (1975). This data narrows the efficient launch window, which will be selected accordingly for the optimal solution computation presented in the next section.

8.1.2. Problem setup and results

In order to start the algorithm procedure, the parametrization of the design variables must be specified.
From the data gathered in the prevous section, the departure time grid is started in 1982,

with a launch window of 3 years subdivided into 1100 equispaced time steps (a departure every terrestrial day roughly). Despite the real mission departure date is in 1977, due to the strong simplifying assumptions applied to the model, which completely disregard out of plane motion, it is expected a moderate discrepancy of the optimal solution from the baseline data. For the first interplanetary arc (Earth-Jupiter leg), only elliptic orbits are considered, so the eccentricity can span between 0 and 1 excluded, and the chosen discretization counts 200 discrete values. Finally for the true anomaly θ, the range of variation is between $[0, 2\pi]$, counting 360 discrete values. It is clear from the reported data that the possible combinations that can arise from the discretization is very high.
For the second interplanetary leg, both elliptic and hyperbolic orbits are considered, and the chosen discretization for the deflection angle γ are 720 discrete values, spanning $[-\pi, \pi]$. For both transfer arcs a space pruning technique has been applied to the FTC value, with a maximum violation value of 10^{-3}.
The obtained solution is represented in figure 8.2:

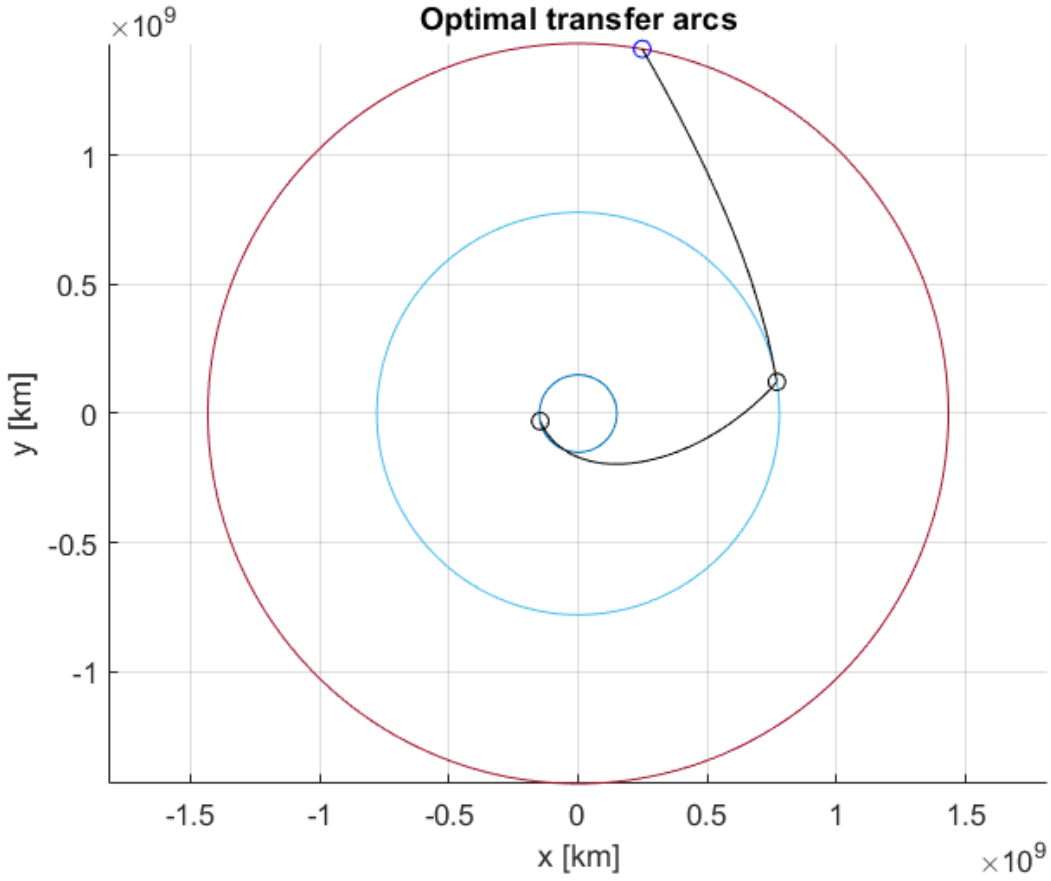

Figure 8.2: Optimal trajectory in terms of transfer precision (FTC). In blue Earth's orbit, in light blue Jupiter's orbit and in red Saturn's orbit. The small circles represent the planetary position at encounter.

The orbital elements of the transfer legs are reported in table 8.1:

a	e	i	Ω	ω	θ
$6.5426 \cdot 10^8$ km	0.7832	0	0	-2.4518 rad	5.7931 rad
$-5.29592 \cdot 10^8$ km	2.3062	0	0	0.1986 rad	-0.0204 rad

Table 8.1: Keplerian parameters of the computed optimal solution's transfer legs.

These results can be compared to the baseline mission orbital prameters, reported in table 8.2:

a	e	i	Ω	ω
$7.4576 \cdot 10^8$ km	0.7977	0.018 rad	-0.306 rad	-0.013 rad
$-5.9324 \cdot 10^8$ km	2.3027	0.043 rad	1.972 rad	-0.027 rad

Table 8.2: Keplerian parameters of the baseline Voyager-1 mission.

Since the presented solution completely disregards out of plane motion, the two trajectories show not negligible differences, despite showing similar values for the semi-major axis and eccentricity.
The presented trajectory shows the minimum value for the FTC function, corresponding to the most precise transfer, that leads to a distance error with respect of the target planets shown in table 8.3:

Planet	FTC value	Distance Error
Jupiter	$2.28 \cdot 10^{-4}$	$1.7749 \cdot 10^5$ km
Saturn	$-7.7369 \cdot 10^{-4}$	$1.108 \cdot 10^6$ km

Table 8.3: FTC value and associated distance error of each interplanetary leg.

The departure and arrival time for each leg, corresponding to the shown planetary configuration are reported in table 8.4:

Departure date	02-01-82
First encounter date	05-08-83
Arrival date	04-08-85

Table 8.4: Departure, first encounter and arrival date for the computed solution.

Finally, the flyby is characterized more in detail. In order to obtain the prescribed variation of keplerian parameters, the close encounter must take place at a specific location in the B-plane, namely (table 8.5):

ξ	η	ζ
0	0	$8.2426 \cdot 10^{-4}$

Table 8.5: B-plane coordinates of the close encounter. Note that all the values are adimensional, due to the fact that all quantities used in the B-plane computations are normalized.

It is also interesting to evaluate the characteristic quantities that define the close encounter, shown in table 8.6:

b	r_p	e	δ
$-6.4167 \cdot 10^5$ km	$1.7225 \cdot 10^5$ km	1.1553	2.0926 rad

Table 8.6: Values of the impact parameter b, perigee radius of the flyby hyperbola r_p, eccentricity and turn angle δ.

Both the values for perigee radius and impact parameter show a value compatible with the minimum flyby altitude (the planetary radius), making the presented close encounter indeed feasible.
The algorithm took 246.69 s for the first arc generation, 9.582 s for the second arc and only 0.085 s for the dynamic programming optimization.
An interesting insight on the computed solutions, not only the optimal one, can be deduced from figure 8.3:

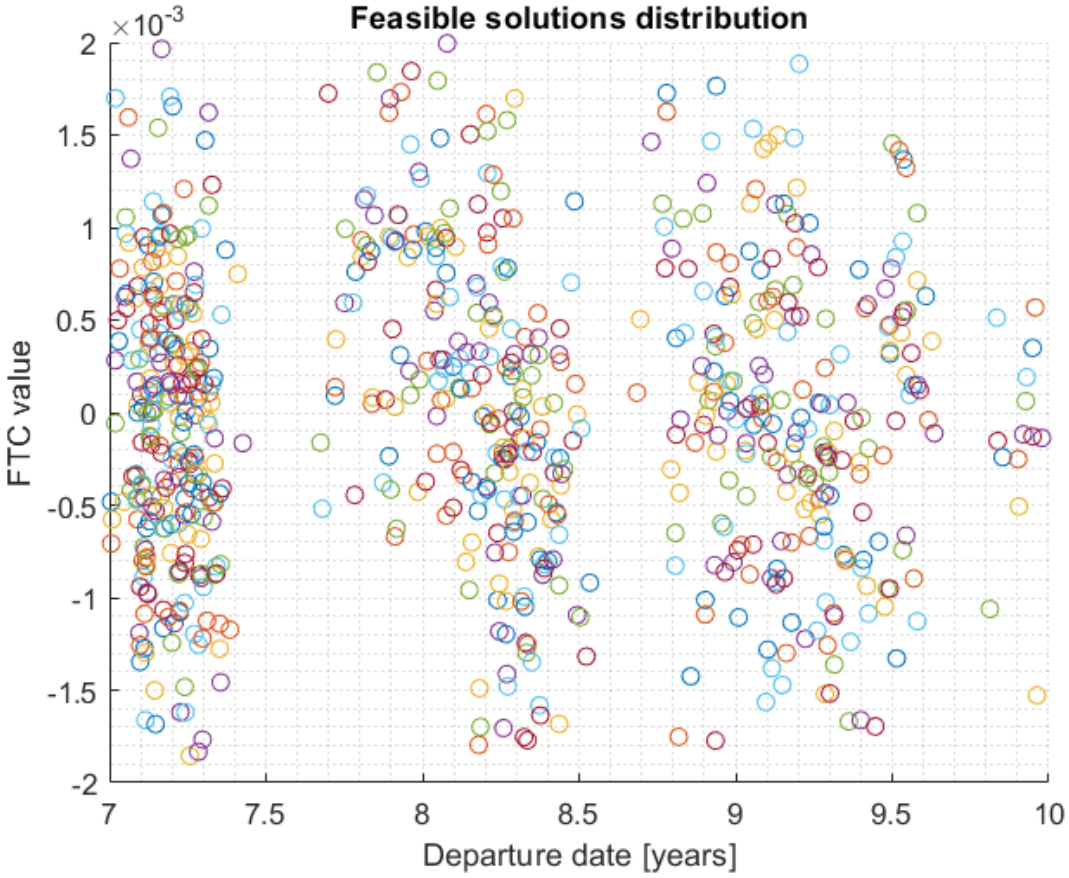

Figure 8.3: Distribution of the feasible solutions' departure dates with their correspondent FTC value.

It appears clear from figure 8.3 that the feasible solutions' departure dates are always concentrated throughout the first half year, that corresponds to a specific position of the Earth in space. Evidently this particular angular position of the Earth is very favorable for reaching easily the gas giants Jupiter and Saturn by following the chosen planetary sequence.

8.2. Voyager-2

Voyager-2 mission objective was even more ambitious than it's close neighbour: following a longer route towards the gas giants Jupiter and Saturn would enable the spacecraft to visit both the icy giants Uranus and Neptune, then leaving behind the Solar System.
The baseline trajectory is the mission profile with launch on August 23 1977, close encounters with Jupiter on July 9 1979, with Saturn on August 26 1981, with Uranus on January 24 1986 and finally reaching Neptune on August 25 1989. Again, since the whole

trajectory involves no powered maneuvers, the mission is ideal for testing the performance of the algorithm, this time in a MGA scenario.
The complexity of the mission profile has increased greatly due to the high number of planetary flybys, so it is expected for the algorithm to decrease performance-wise.
To show how the performance varies with the number on close encounters, only the Jupiter-Saturn-Uranus part of the mission is simulated.

8.2.1. Partial mission problem setup and results

Again, before starting the search for the optimal solution, the algorithm is run preliminary to gain some insight on the most efficient launch window to be used for the final simulation.
The framework of the problem is the same as before, but this time using 3600 discrete time values for a broad launch grid of 15 years, starting from 1975.
The result of this simulation is shown in figure 8.4:

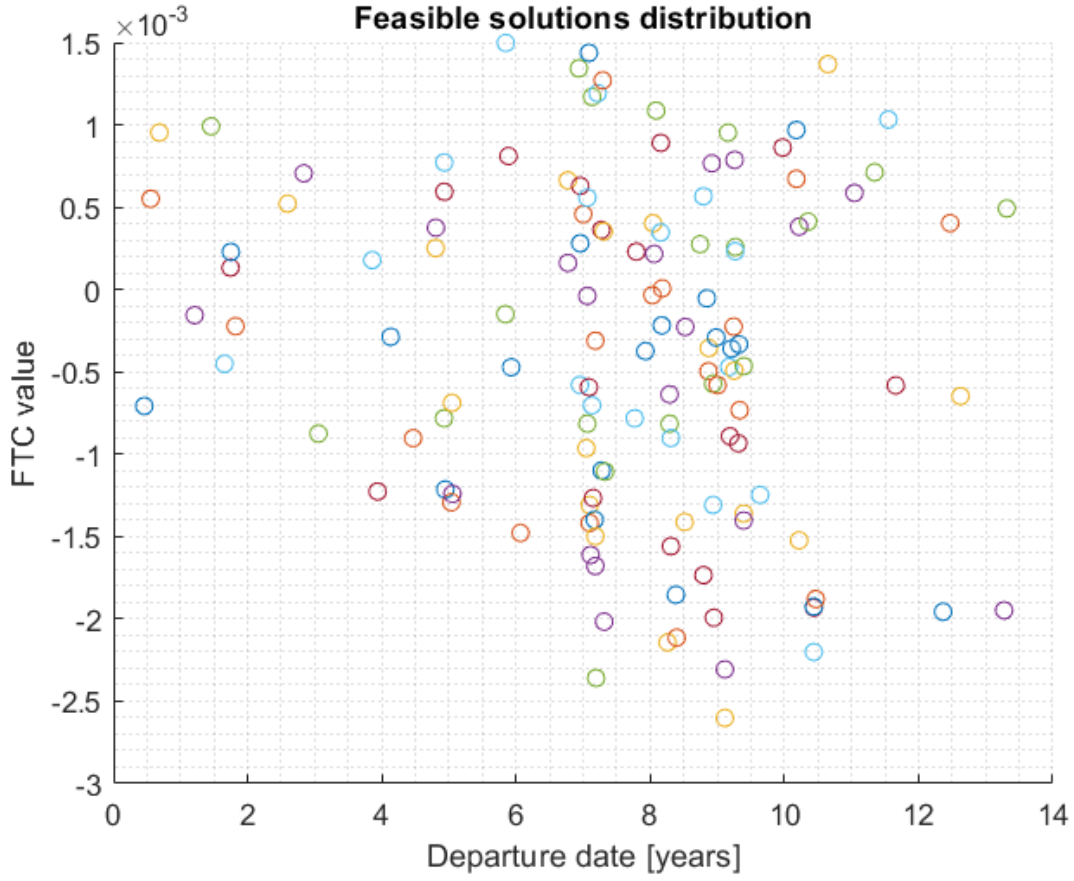

Figure 8.4: Distribution of the feasible solutions' departure dates with their correspondent FTC value for the partial Voyager-2 mission.

It is possible to note that the number of feasible solutions has greatly decreased, which was to be expected, since the complexity of the mission has increased greatly, due to the additional close encounter with Saturn. It is interesting to highlight that despite the new objective of reaching Uranus, the efficient launch time window didn't change drastically, since the highest concentration of feasible solutions is still located between 1982 and 1985. Thanks to this insight, it is now possible to tackle efficiently the problem at hand.
Again, in order to start the algorithm procedure, the discretization of the design variables must be specified.
From the data gathered previously, the departure time grid is started 1982, with a launch window of 3 years subdivided into 1100 equispaced time steps (a departure every terrestrial day roughly). Despite the real mission departure date is in 1977, due to the strong simplifying assumptions applied to the model, which completely disregard out of plane motion, it is expected a moderate discrepancy of the optimal solution from the baseline data. Even more so now that the heliocentric distances have grown considerably.
For the first interplanetary arc (Earth-Jupiter leg), only elliptic orbits are considered, so the eccentricity can span between $(0, 1)$, and the chosen discretization counts 200 discrete values. Finally for the true anomaly θ, the range of variation is between $[0, 2\pi]$, counting 360 discrete values. It is clear from the data reported that the possible combinations that can arise from the discretization is very high.
For the subsequent interplanetary legs, both elliptic and hyperbolic orbits are considered, and the chosen discretization for the deflection angle γ are 720 discrete value, spanning $[-\pi, \pi]$. For all transfer arcs a space pruning technique has been applied to the FTC value, with a maximum violation value of 10^{-3}. The obtained solution is represented in figure 8.5:

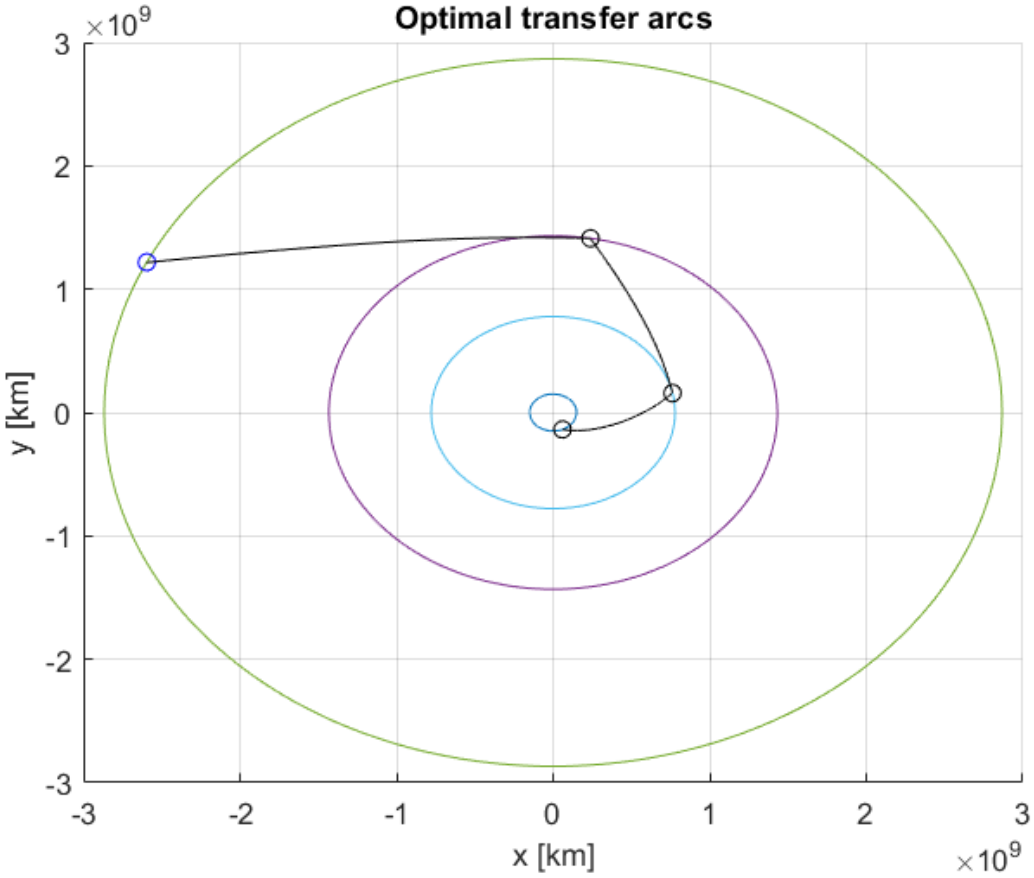

Figure 8.5: Optimal trajectory in terms of transfer precision (FTC). In blue Earth's orbit, in light blue Jupiter's orbit and in purple Saturn's orbit and in green Uranus orbit. The small circles represent the planetary position at encounter.

The reported trajectory shows sharp cusps at the close encounters locations. Such intense variations of the orbital elements are obtainable due to the huge planetary masses of the gas giants Jupiter and Saturn. The flyby deflection intensity in fact, is proportional to the mass of the close encounter planet. Nevertheless, the proposed solution is indeed feasible, as shown in table 8.12
The orbital elements of the transfer legs are reported in table 8.7:

a	e	i	Ω	ω	θ
$5.6449 \cdot 10^8$ km	0.8373	0	0	3.7075 rad	1.4177 rad
$-5.7637 \cdot 10^8$ km	2.3506	0	0	0.1906 rad	0.0132 rad
$-2.835 \cdot 10^8$ km	6.0013	0	0	1.555 rad	-0.152

Table 8.7: Keplerian parameters of the computed optimal solution's transfer legs.

These results can be compared to the baseline mission orbital prameters, reported in table 8.8:

a	e	i	Ω	ω
$5.4447 \cdot 10^8$ km	0.724429	0.084 rad	-0.5749 rad	3.7075 rad
$-2.220 \cdot 10^9$ km	1.338264	0.045 rad	2.08	-0.16 rad
$-5.7905 \cdot 10^8$ km	3.4802	0.046 rad	1.34 rad	1.959 rad

Table 8.8: Keplerian parameters of the baseline Voyager-2 mission.

Since the presented solution completely disregards out of plane motion, the two trajectories show not negligible differences, even more so in the presented mission scenario, which presents only the partial mission profile, not reaching Neptune.
The presented trajectory shows the minimum value for the FTC function, corresponding to the most precise transfer, that leads to a distance error with respect of the target planets shown in table 8.9:

Planet	FTC value	Distance Error
Jupiter	$-8.5909 \cdot 10^{-4}$	$6.6878 \cdot 10^5$ km
Saturn	$-9.7358 \cdot 10^{-4}$	$1.394 \cdot 10^6$ km
Uranus	$-9.9902 \cdot 10^{-4}$	$2.8642 \cdot 10^6$ km

Table 8.9: Departure, first encounter and arrival date for the computed solution.

The departure and arrival time for each leg, corresponding to the shown planetary configuration are reported in table 8.10:

Departure date	14-04-82
First encounter date	05-09-83
Second encounter date	15-08-85
Arrival date	07-04-89

Table 8.10: Departure, first and second encounter and arrival date for the computed solution.

In order to obtain the prescribed variation of keplerian parameters, the close encounter must take place at a specific location in the B-plane, namely (table 8.11):

ξ	η	ζ
0	0	$6.6144 \cdot 10^{-4}$
0	0	$1.0303 \cdot 10^{-4}$

Table 8.11: B-plane coordinates of the close encounters. Note that all the values are adimensional, due to the fact that all quantities used in the B-plane computations are normalized.

It is interesting to evaluate the characteristic quantities that define the close encounter, shown in table 8.12:

b	r_p	e	δ
$-5.1492 \cdot 10^5$ km	$1.1674 \cdot 10^5$ km	1.1084	2.2498 rad
$-1.4755 \cdot 10^5$ km	$6.1293 \cdot 10^4$ km	1.4171	1.5667 rad

Table 8.12: Values of the impact parameter b, perigee radius of the flyby hyperbola r_p, eccentricity and turn angle δ.

Both the values for perigee radius and impact parameter show a value compatible with the minimum flyby altitude (the planetary radius), making the presented close encounter indeed feasible.
The algorithm took 249.3 s for the first leg generation, 143.52 s for the subsequent legs generation and only 13.11 s for the dynamic programming optimization.

8.3. Algorithm validation

To further prove the goodness of the implemented procedure, the computed transfers for Voyager-1 and Voyager-2 missions are simulated using a fool-proof procedure, in this case, a Lambert solver. The objective of this section is showing that the relative error of the syzygy solutions is negligible from a trajectory point of view.
From the obtained orbital parameters for the Voyager-1 mission, using the same value for the FTC pruning (10^{-3}), the relative errors are shown in figures 8.6, 8.7 and 8.8:

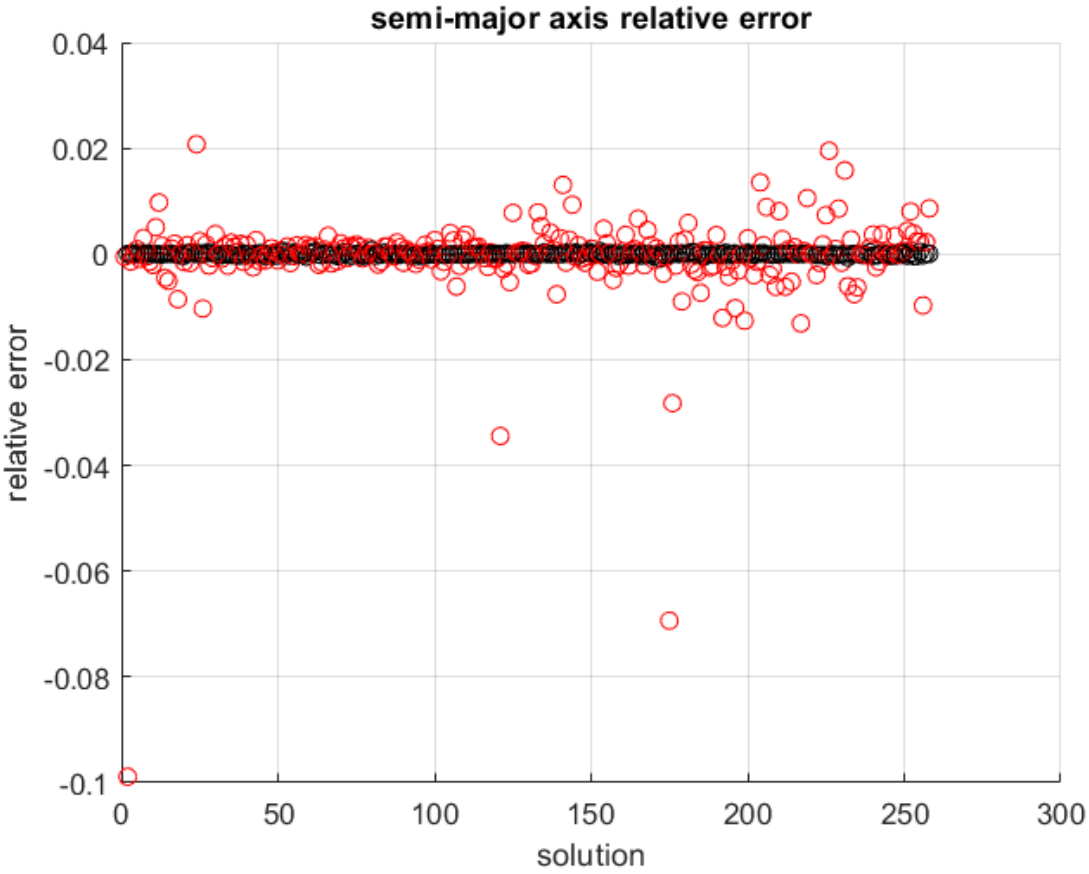

Figure 8.6: Relative error on the semi-major axis computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

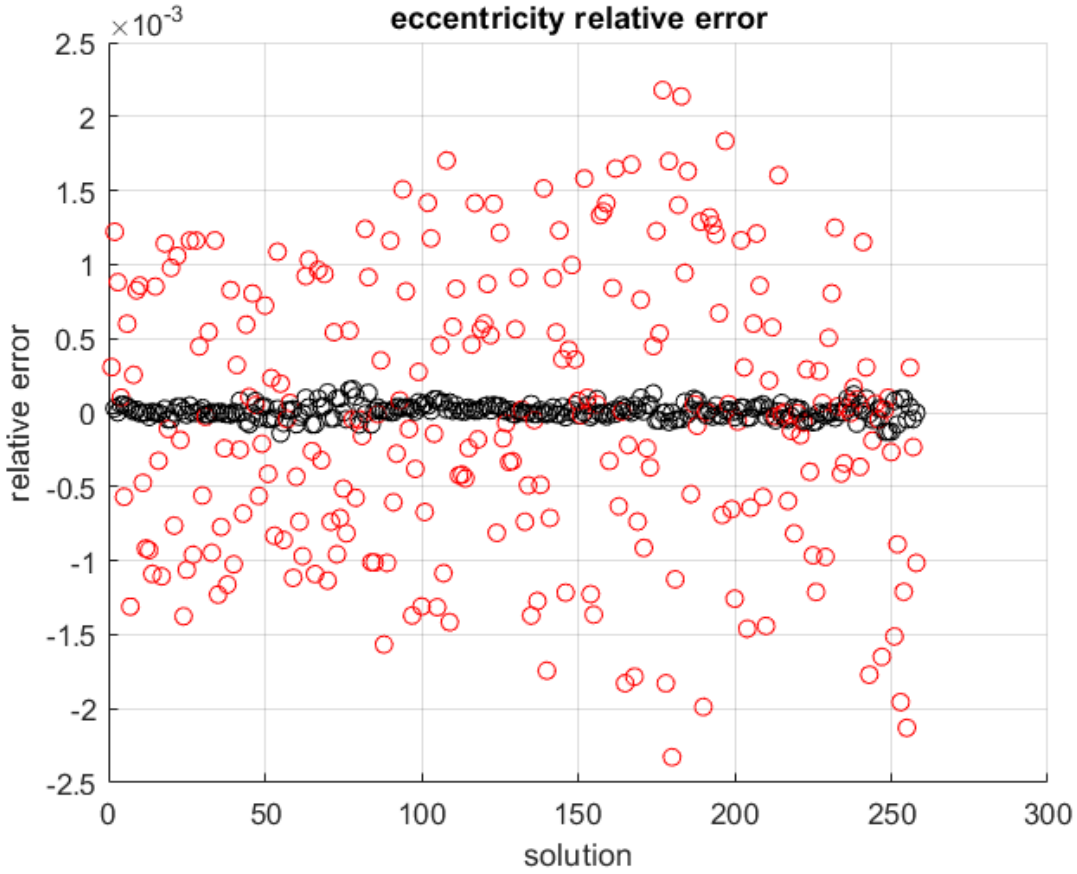

Figure 8.7: Relative error on the eccentricity computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

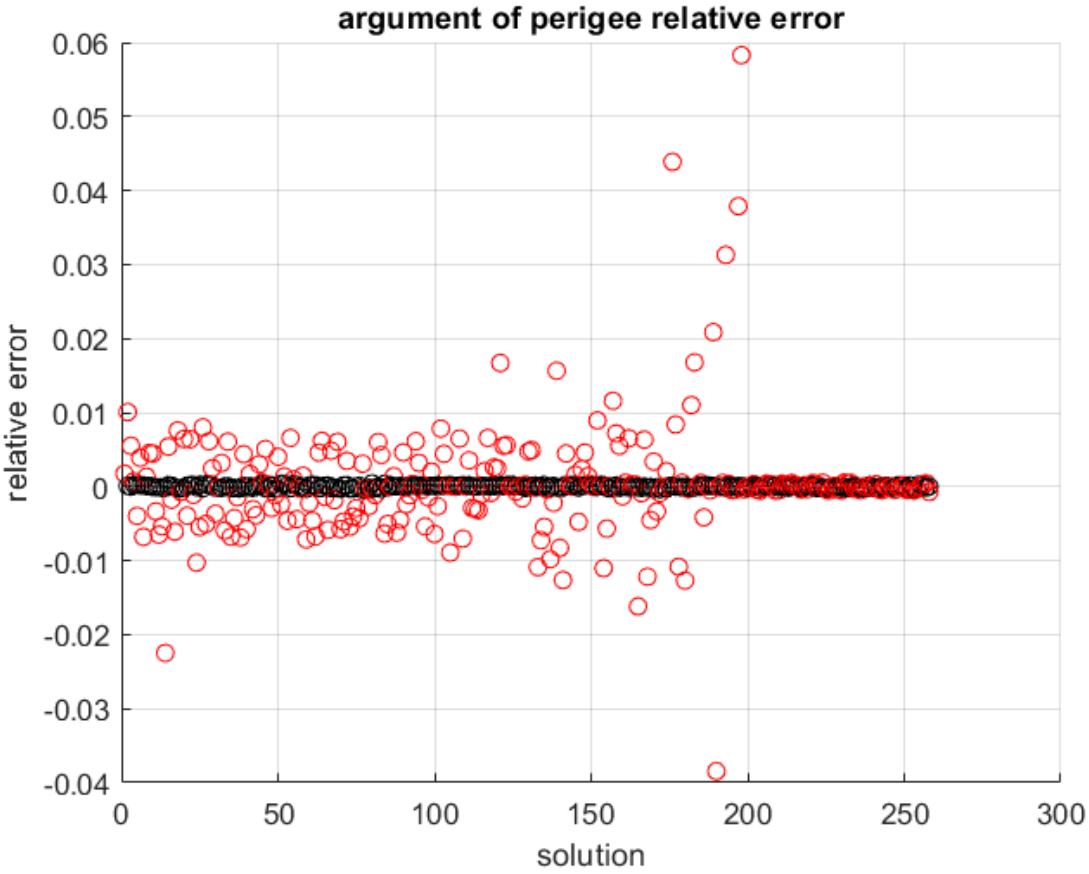

Figure 8.8: Relative error on the argument of periapsis computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

Using instead a more aggressive pruning policy ($5 \cdot 10^{-4}$), figures 8.9, 8.10 and 8.11:

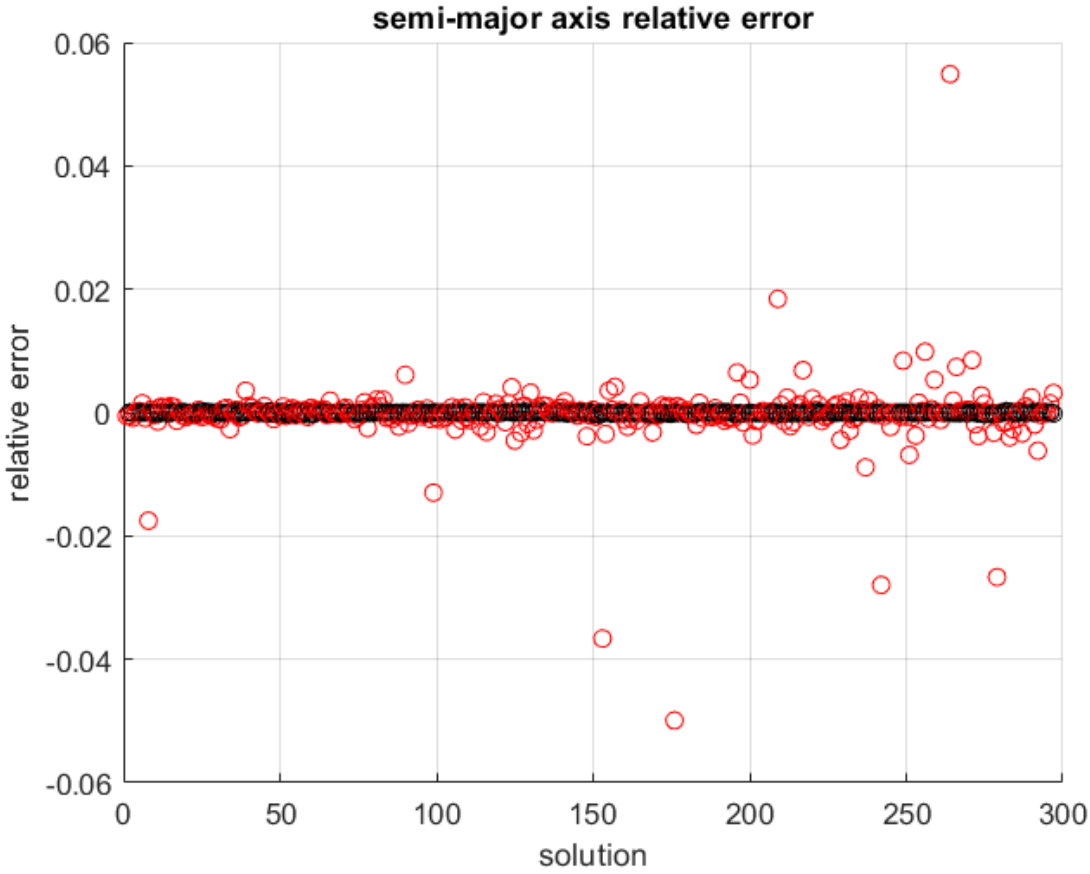

Figure 8.9: Relative error on the semi-major axis computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

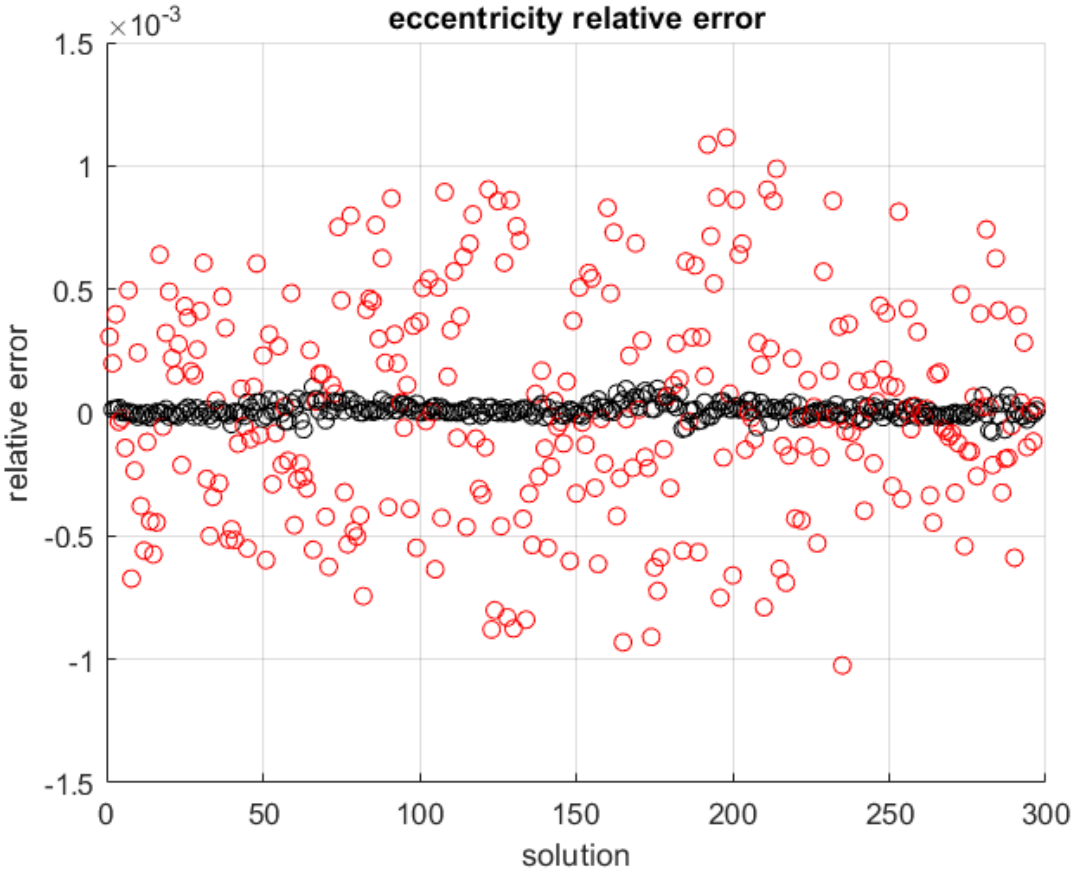

Figure 8.10: Relative error on the eccentricity computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

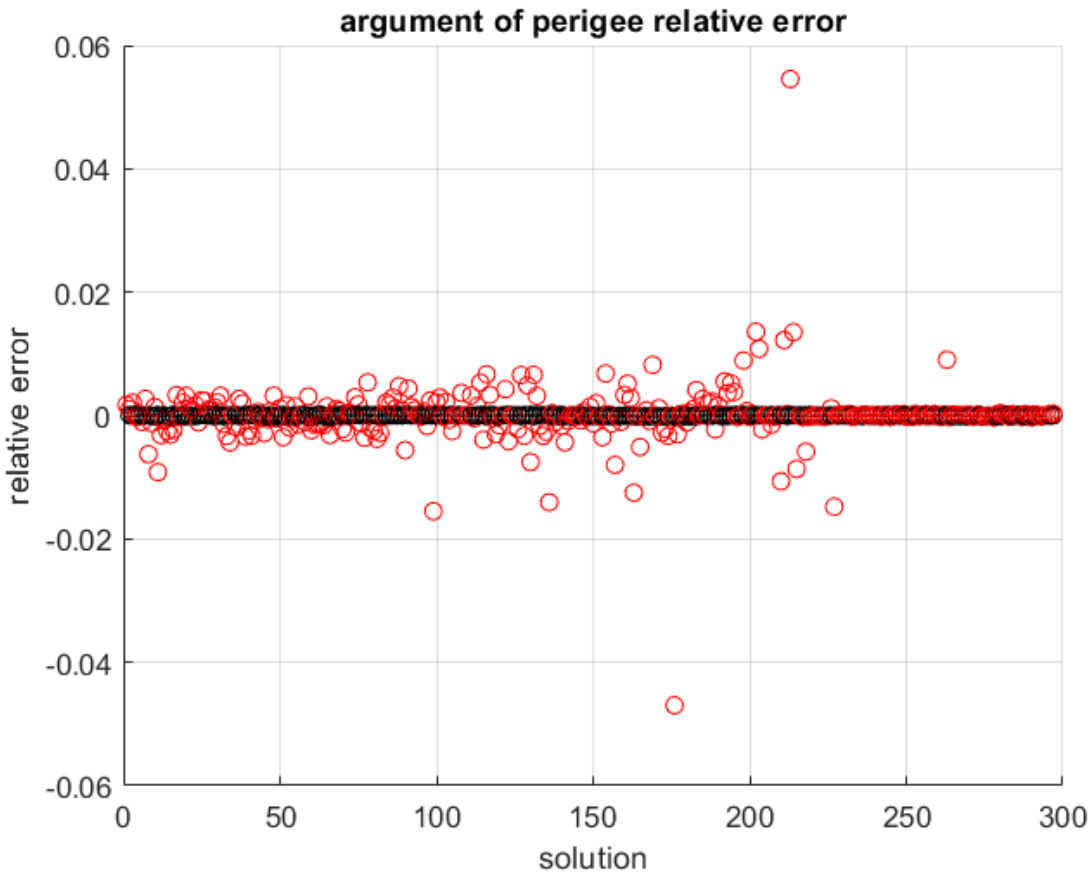

Figure 8.11: Relative error on the argument of periapsis computation committed by the syzygy algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc.

The results show that in general, as the FTC value associated to a particular solution decreases, so does the relative error. A compromise must be therefore sought in the choice for the adapt FTC pruning tolerance: a very aggressive pruning will lead to very precise solutions (very low relative error) but will limit greatly the number of acceptable solutions computed by the algorithm; a lax pruning instead will broaden greatly the solution space, at the expense of the goodness of the computed solutions.
More in detail, it is possible to note that the error associated to the first interplanetary leg is very low. This is due to the fact that the parametrization used for the first state generation is much finer and combinatorially complex, since it originates from triplets. For the subsequent legs the generation of the orbital parameters is from couplets instead, which span a much smaller solution space.
Overall, it is safe to conclude that for the single GAM scenario, the relative error obtained using the *syzygy* algorithm is low, and can be certainly neglected in a preliminary

trajectory design task.
It is interesting to evaluate how the relative error varies as the complexity of the mission increases, that is, adding a close encounter. Following the same procedure as for the Voyager-1 mission, the partial Voyager-2 mission is simulated by means of a Lambert solver, using a pruning tolerance on the FTC value of $5 \cdot 10^{-3}$.
The results are shown in figures 8.12, 8.13 and 8.14:

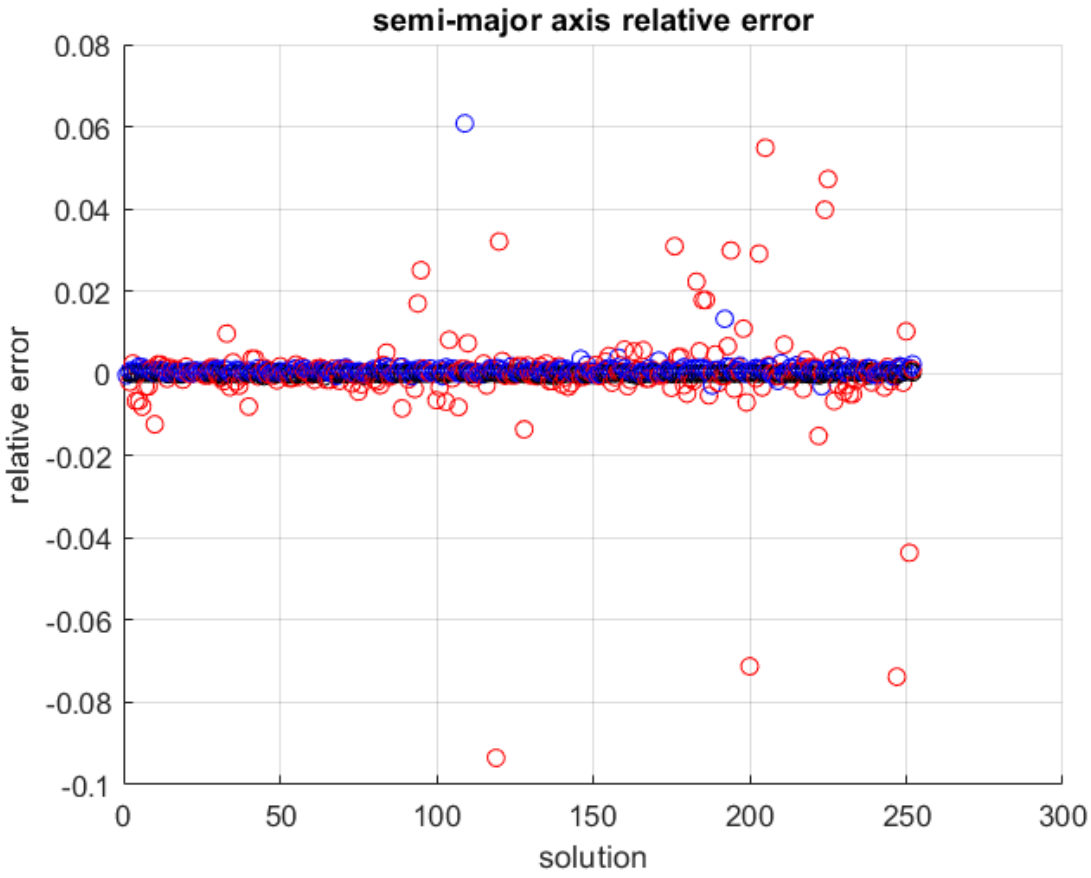

Figure 8.12: Relative error on the semi-major axis computation committed by the *syzygy* algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc and the blue ones the third.

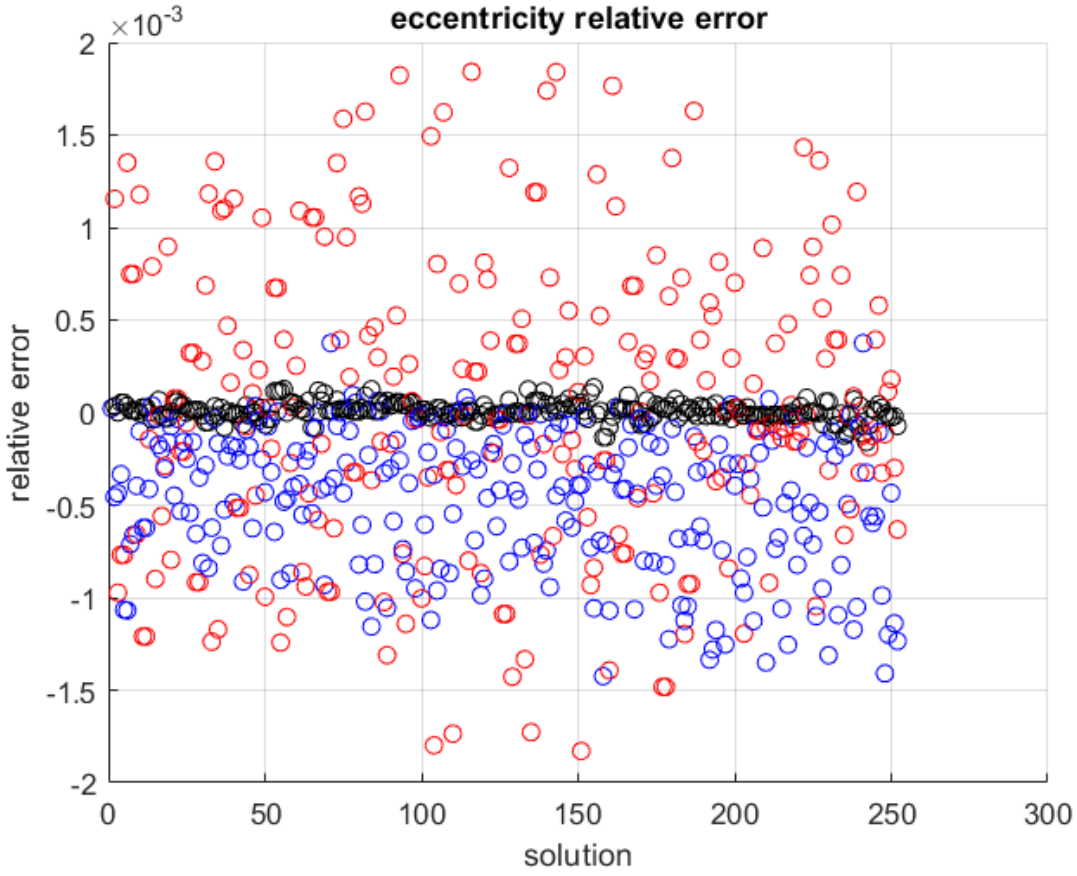

Figure 8.13: Relative error on the eccentricity computation committed by the *syzygy* algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc and the blue ones the third.

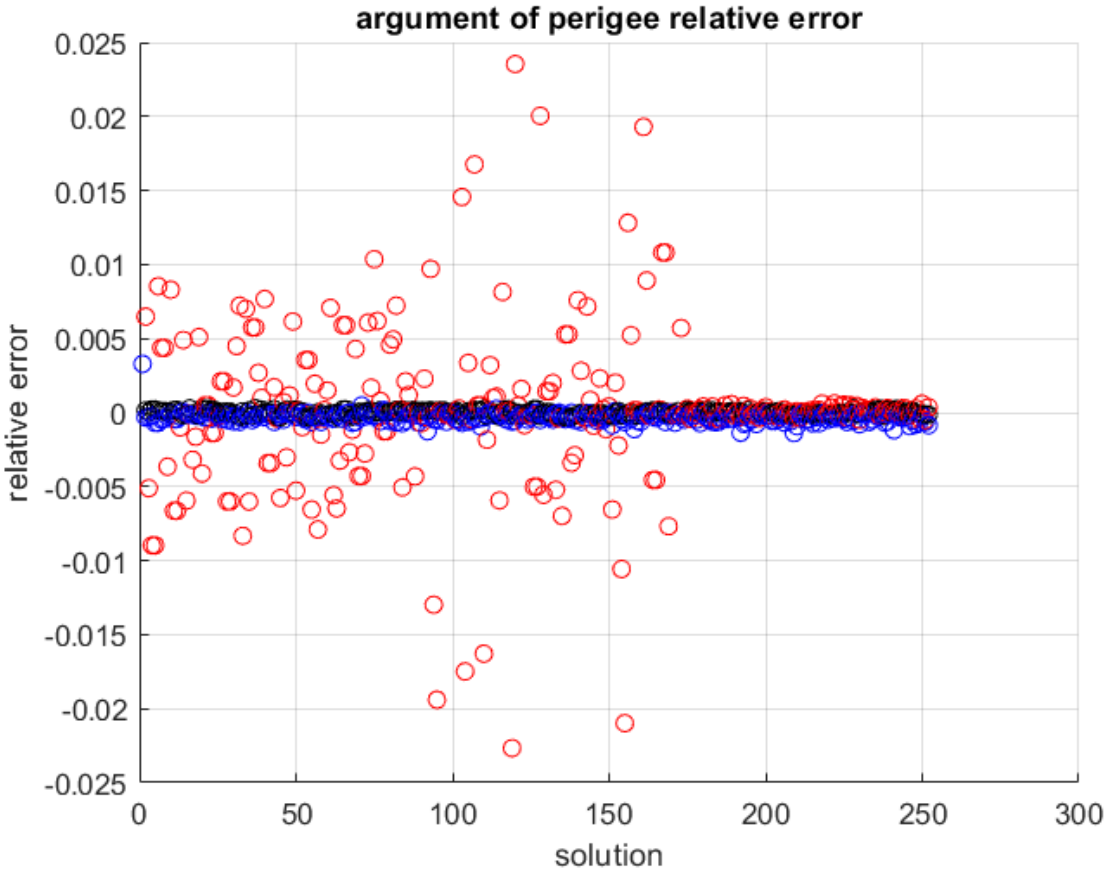

Figure 8.14: Relative error on the argument of periapsis computation committed by the *syzygy* algorithm implemented, with respect to a Lambert solver. The black circles represent the first interplanetary leg, the red ones the second arc and the blue ones the third.

From the figures presented it can be deduced that the relative error doesn't increase as the mission complexity increases, making the proposed algorithm adapt also for the MGA case.

8.4. Inner planets mission

As a final test for the presented solution strategy, an arbitrary mission to the inner planets is simulated.
The selected planetary sequence is completely arbitrary and is not based on any insight or real mission, and it features: departure from Earth, close encounters with Venus, Earth, Venus to finally reach Mars. Since the sequence is essentially randomic, a decrease in the algorithm performance, that is, the ability of finding solutions fully compliant

with the problem constraints, is to be expected. The mission proposed features three close encounters, increasing the complexity with respect to the partial Voyager-2 mission analyzed in the previous section.
The chosen discretization is: 730 discrete departure dates, spanning between 2023 and 2028, only elliptic orbits are considered for the first leg, and the chosen discretization counts 300 discrete values. Finally for the true anomaly θ, the range of variation is between $[0, 2\pi]$, counting 360 discrete values.
For the subsequent interplanetary arcs, both elliptic and hyperbolic orbits are considered, and the chosen discretization for the deflection angle γ are 720 discrete values, spanning $[-\pi, \pi]$. For all transfer legs a space pruning technique has been applied to the FTC value, with a maximum violation value of $5 \cdot 10^{-3}$. The obtained solution is represented in figure 8.15:

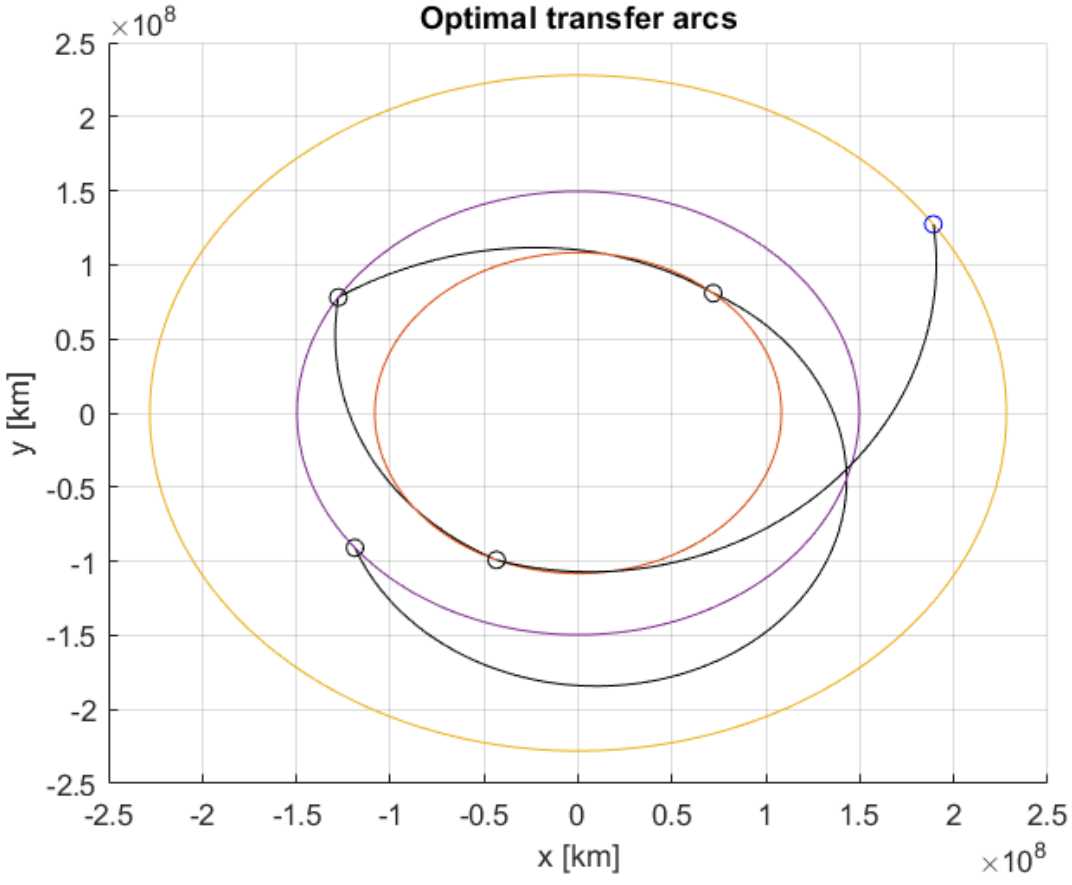

Figure 8.15: Optimal trajectory in terms of transfer precision (FTC). In red Venus' orbit, in purple Earth's orbit and in yellow Mars orbit. The small circles represent the planetary position at encounter.

The orbital parameters of the optimal transfer arcs are presented in table 8.13:

a	e	i	Ω	ω	θ
$1.4185 \cdot 10^8 km$	0.305	0	0	1.7472 rad	2.0477 rad
$2.0509 \cdot 10^8 km$	0.4797	0	0	1.1363 rad	5.9906 rad
$2.0624 \cdot 10^8 km$	0.481	0	0	-2.2442 rad	4.8369 rad
$2.023 \cdot 10^8 km$	0.4722	0	0	-1.6959rad	5.9948 rad

Table 8.13: Keplerian parameters of the computed optimal solution's transfer legs.

The presented trajectory shows the minimum value for the FTC function, corresponding to the most precise transfer, that leads to a distance error with respect of the target planets shown in table 8.14:

Planet	FTC value	Distance Error
Venus	$-7.3078 \cdot 10^{-4}$	$7.9078 \cdot 10^4$ km
Earth	$-2.7062 \cdot 10^{-4}$	$4.0484 \cdot 10^4$ km
Venus	$1.9 \cdot 10^{-3}$	$2.056 \cdot 10$ km
Mars	$5.0663 \cdot 10^{-4}$	$1.1549 \cdot 10^5$ km

Table 8.14: Departure, first encounter and arrival date for the computed solution.

The departure and arrival time for each leg, corresponding to the shown planetary configuration are reported in table 8.15:

Departure date	26-01-25
First encounter date	17-09-25
Second encounter date	17-11-25
Third encounter date	18-01-26
Arrival date	02-06-26

Table 8.15: Departure, first, second and third encounter and arrival date for the computed solution.

The orbital elements reported, show a several orders of magnitude smaller variation induced by the flyby effect compared to the values computed for the close encounters with the giant planets in missions Voyager-1 and Voyager-2. This is due to the fact that the inner celestial bodies are characterized by planetary masses several hundreds times smaller than the gaseous outer planets. The magnitude of the obtainable variation of orbital parameters is in fact proportional to the mass of the flyby planet.

Therefore, for missions towards the inner parts of the Solar System, the weight of the planetary sequence on the ability of the algorithm of finding feasible solutions is even greater. Due to the small attainable variations in the orbital parameters, the celestial bodies must have particularly favorable relative positions to be efficiently reached.
In order to obtain the prescribed variation of keplerian parameters, the close encounter must take place at a specific location in the b-plane, associated to it's correspondent impact parameter, namely (table 8.16):

ξ	η	ζ
0	0	$1.2327 \cdot 10^{-4}$
0	0	$6.4 \cdot 10^{-3}$
0	0	$-1.9 \cdot 10^{-3}$

Table 8.16: B-plane coordinates of the close encounters. Note that all the values are non-dimensional, due to the fact that all quantities used in the B-plane computations are normalized.

It is interesting to evaluate the characteristic quantities that define the close encounter, shown in table 8.17:

b	r_p	e	δ
$-1.3339 \cdot 10^4$ km	$9.4443 \cdot 10^3$ km	3.0103	0.6773 rad
$-9.5307 \cdot 10^5$ km	$9.5099 \cdot 10^5$ km	457.727	0.0044 rad
$2.0202 \cdot 10^5$ km	$1.9722 \cdot 10^5$ km	41.614	0.0481 rad

Table 8.17: Values of the impact parameter b, perigee radius of the flyby hyperbola r_p, eccentricity and turn angle δ.

Both the values for perigee radius and impact parameter show a value compatible with the minimum flyby altitude (the planetary radius), making the presented close encounter indeed feasible.
The algorithm took 370.07 s for the first leg generation, 224.362 s for the subsequent legs generation and only 0.171 s for the dynamic programming optimization.

9 | Conclusions and outlook

The conic *syzygy* algorithm was extended removing the limiting constraint of tangential departures and arrivals and integrating also hyperbolic orbits, to gain a more generalized trajectory design tool. The combination of the *syzygy* algorithm with B-plane deflection model and flyby characterization enabled to solve efficiently the problem of patching consecutive solutions, generating interplanetary arcs connected by close encounters that automatically respect the conservation of the Tisserand parameter and with trajectories that respect the minimum flyby height, namely the planetary radius. The original *syzygy* algorithm used as starting point for the presented extension in fact failed at producing consecutive transfer legs that lead to feasible close encounters. The combination of the *syzygy* algorithm with the B-plane deflection model, which is solely used for the generation of the post-encounter orbital parameters doesn't alter the computational quickness of the original *syzygy* formulation, hence using the extended version of the algorithm is always recommended.

The algorithm was finally validated by comparing the goodness of the computed solutions with the typical and fool-proof Lambert solver, showing very small discrepancies between the two solutions. The performance might be comparable in terms of solution goodness, but the *syzygy* algorithm real merit is the great quickness in the solution space search. The same problem, analysed by means of a Lambert solver takes an extremely higher amount of time to be solved, as shown in Chapter 5.

Regarding instead the implementation of an optimization strategy for the solution of the problems considered, a dynamic programming approach was proposed. The complexity of the problems presented lies in the size of the solution space and on the highly combinatorial nature of the latter. The already proven efficiency and computational quickness of the dynamic programming approach, together with the speed of the *syzygy* formulation of the problem, represent a very useful tool for the preliminary trajectory design for MGA mission scenarios.

The developed algorithm was then used to simulate two real-life missions, Voyager-1 and Voyager-2. The computed solutions present differences with the baseline mission profiles, mainly due to the bi-dimensionality of the model, but nevertheless can be considered good

initial guesses for more complex solution strategies.
The main limitation of the proposed procedure is the bi-dimensionality of the model: the out of plane motion leads to non-negligible differences in the trajectory computed with respect to the baseline Voyager missions. The B-plane procedure to evaluate post encounter parameters, explained in detail in Chapter 4, is already defined in the three dimensional case, so future works could focus on the extension of the *syzygy* algorithm to the 3-D case, exploiting considerations on spherical geometry. Another limitation of the model is the total absence of orbital perturbations in the trajectory design process. A step forward could be to include perturbations within the planet's SOI in the model, similarly to what has already been introduced in the resonant close encounter scenario.
A possible direction that could be followed in the future works to fully exploit the presented strategy is looking for more complex and efficient optimality policies for the selection of the optimal trajectory. By keeping the precision of the solution acceptable by means of a space pruning strategy on the FTC value, optimality might be sought in meeting constraints of other nature. Another certainly valuable direction for future work could be the integration of a strategy for the planetary sequence identification, to obtain an automatic MGA mission design tool.
In conclusion, the proposed algorithm provides a fast and reliable tool for the preliminary design of MGA trajectories, which could serve as reasonable starting point for numerical methods to faster converge to more complete trajectory solutions.

A | Appendix A

A.1. Stagecoach problem

The stagecoach problem concerns a fortune seeker in Missouri who decided to go west to join the gold rush in California during the mid-19th century. The journey would require traveling by stagecoach through unsettled country where there was serious danger of attack by marauders. Although his starting point and destination were fixed, he had a considerable number of choices as to which states (or territories that subsequently became states) to travel through en route. The possible routes are shown in Fig.A.1, where each state is represented by a circled letter and the direction of travel is always from left to right in the diagram. Thus, four stages were required to travel from his point of embarkation in state A to his destination in state J. After some thought, he came up with a rather clever way of determining the safest route. Life insurance policies were offered to stagecoach passengers. Because the cost of the policy for taking any given stagecoach run was based on a careful evaluation of the safety of that run, the safest route should be the one with the cheapest total life insurance policy. Which route minimizes the total cost of the policy?

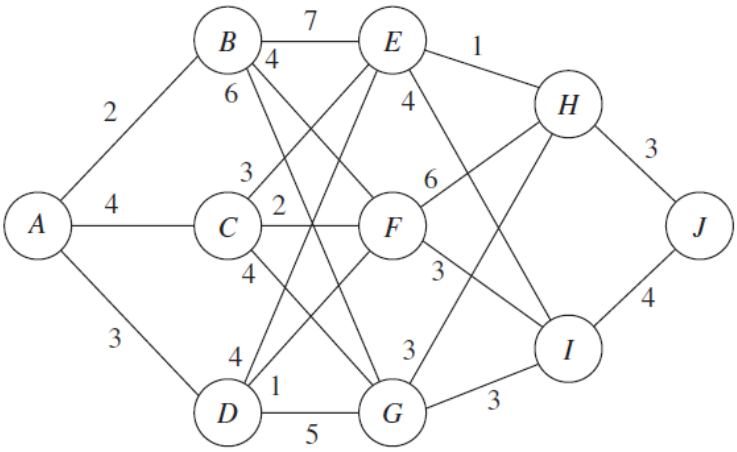

Figure A.1: Stagecoach problem visualization. The figure is taken from Frederick and Hillier work [16].

First, note that the shortsighted approach of selecting the cheapest run offered by each successive stage need not yield an overall optimal decision. Following this strategy would give the route A, B, F, I, J, at a total cost of 13. However, sacrificing a little on one stage may permit greater savings thereafter. For example, A, D, F is cheaper overall than A, B, F.

One possible approach to solving this problem is to use trial and error. In this particular case it might be computationally cheap, but this is not the case for larger problems, that lead inevitably to a huge computational effort. Fortunately, dynamic programming provides a solution with much less effort than exhaustive enumeration, guaranteeing that the solution found is optimal.

The idea behind dynamic programming is to decompose the problem into smaller sub-problems and finding the optimal solution for these smaller problem. The optimal solution to the smaller problems is then used to gradually solve the larger ones, until the original problem is completely solved.

For the stagecoach problem, we start with the smaller problem where the fortune seeker has nearly completed his journey and has only one more stage to go. The obvious optimal solution for this smaller problem is to go from his current state (whatever it is) to his ultimate destination (state J). Then, by moving back one stage at a time (from right to left), each subsequent sub-problem answers the question: "for each possible state at the current stage what is the shortest path to the final destination?". The advantage of this approach is that, given the solution to the first j sub-problems, it takes relatively little effort to find the answer to the (j + 1)-th. The original problem is automatically solved once the optimal solution to the sub-problem associated with the first stage has been found.

The stagecoach problem will be now tackled using the same formalism and approach proposed by Hillier and Lieberman [16].

Let denote with s_n the state of the system at stage n and with x_n the policy decision made at that point, namely what state to go next. Being in state s_n and selecting x_n at the next stage will add a net contribution $c_{s_n} * x_n$, where c_{sn} is the cost associated to the immediate stage of stage n. Let $f_n(s, x_n)$ be the total cost of the best overall policy for the remaining stages, given that the fortune seeker is in state s, ready to start stage n, and selects x_n as the immediate destination. Given s and n, let x_n^* denote any value of x_n (not necessarily unique) that minimizes $f_n(s, x_n)$, and let $f_n^*(s)$ be the corresponding minimum value of $f_n(s, x_n)$. Thus:

$$f_n^*(s) = min f_n(s, x_n) = f_n(s, x_n^*), \tag{A.1}$$

Where, $f_n(s, x_n)$ is the immediate cost (stage n) plus the minimum future cost (stages n + 1 onward). In practice, $f_n^*(s_n)$ represents the minimum distance required to reach J from a node s_n. Starting from the last stage (n = 4) and moving backwards, the solution procedure then consists in iteratively finding $f_4^*(s_4)$, $f_3^*(s_3)$, . . . for each possible state, until computing $f_1^*(s_1 = A)$, which provides the optimal solution to the original problem.

A.1.1. Solution procedure

When the fortune seeker has only one more stage to go (n = 4), his route thereafter is determined entirely by his current state s (either H or I) and his final destination $x_4 = J$, so the route for this final stagecoach run is s $\rightarrow J$. Therefore, since $f_4^*(s) = f_4(s, J) = c_{s,J}$, the immediate solution to the n = 4 problem is:

$\mathbf{s}$	$\mathbf{f_4^*(s)}$	$\mathbf{x_4^*}$
H	3	J
I	4	J

(A.2)

When the fortune seeker has two more stages to go (n = 3), the solution procedure requires a few calculations. For example, suppose that the fortune seeker is in state F. Then, as depicted below, he must next go to either state H or I at an immediate cost of $c_{F,H} = 6$ or $c_{F,I} = 3$, respectively. If he chooses state H, the minimum additional cost after he reaches there is given in the preceding graph as $f_4^*(H) = 3$, as shown above the H node in the diagram. Therefore, the total cost for this decision is 6 + 3 = 9. If he chooses state I instead, the total cost is 3 + 4 = 7, which is smaller. Therefore, the optimal choice is this latter one, x_3^* = I, because it gives the minimum cost $f_3^*(F) = 7$.
Similar calculations need to be made when you start from the other two possible states s = E and s = G with two stages to go. For n = 3, starting from state E:

$$\mathbf{f_3}(E, H) = c_{E,H} + f_4^*(H) = 1 + 3 = 4 \tag{A.3}$$

$$\mathbf{f_3}(E, H) = c_{E,I} + f_4^*(I) = 4 + 4 = 8 \tag{A.4}$$

Making evident that the optimal choice from state E is H, so $f_3^*(E) = 4$.
Starting from state F:

$$\mathbf{f_3}(F, H) = c_{F,H} + f_4^*(H) = 6 + 3 = 9 \tag{A.5}$$

$$\mathbf{f_3}(F,H) = c_{F,I} + f_4^*(I) = 3 + 4 = 7 \tag{A.6}$$

The optimal choice from state F would be going to state I, so $f_3^*(F) = 7$.
Starting lastly from state G:

$$\mathbf{f_3}(G,H) = c_{G,H} + f_4^*(H) = 3 + 3 = 6 \tag{A.7}$$

$$\mathbf{f_3}(G,H) = c_{G,I} + f_4^*(I) = 3 + 4 = 7 \tag{A.8}$$

The optimal choice from state G would be going to state H, so $f_3^*(G) = 6$.
By repeating the same steps, the 2^{nd}-stage sub-problem is solved in a similar fashion. For n = 2, starting from state B:

$$\mathbf{f_2}(B,E) = c_{B,E} + f_3^*(E) = 7 + 4 = 11 \tag{A.9}$$

$$\mathbf{f_2}(B,F) = c_{B,F} + f_3^*(F) = 4 + 7 = 11 \tag{A.10}$$

$$\mathbf{f_2}(B,G) = c_{B,G} + f_3^*(G) = 6 + 6 = 12 \tag{A.11}$$

Note that the optimal policy need not be unique, as is the case for a system in state s_2 = B. The optimal choice is either to go state F or E, in this case $f_2^*(B) = 11$.
Starting from state C:

$$\mathbf{f_2}(C,E) = c_{C,E} + f_3^*(E) = 3 + 4 = 7 \tag{A.12}$$

$$\mathbf{f_2}(C,F) = c_{C,F} + f_3^*(F) = 2 + 7 = 9 \tag{A.13}$$

$$\mathbf{f_2}(C,G) = c_{C,G} + f_3^*(G) = 4 + 6 = 10 \tag{A.14}$$

The optimal choice is to go state E, in this case $f_2^*(C) = 7$.
Starting from state D:

$$\mathbf{f_2}(D,E) = c_{D,E} + f_3^*(E) = 4 + 4 = 8 \tag{A.15}$$

$$\mathbf{f_2}(D,F) = c_{D,F} + f_3^*(F) = 1 + 7 = 8 \tag{A.16}$$

$$\mathbf{f_2}(D,G) = c_{D,G} + f_3^*(G) = 5 + 6 = 11 \tag{A.17}$$

The optimal choice is going to either state E or F, so $f_2^*(D) = 8$.
Lastly, for n = 1, the journey can start only from state A:

$$\mathbf{f_1}(A, B) = c_{A,B} + f_2^*(B) = 2 + 11 = 13 \tag{A.18}$$

$$\mathbf{f_1}(A, C) = c_{A,C} + f_2^*(C) = 4 + 7 = 11 \tag{A.19}$$

$$\mathbf{f_1}(A, D) = c_{A,D} + f_2^*(D) = 3 + 8 = 11 \tag{A.20}$$

An optimal solution for the entire problem can now be identified from the four graphs. Results for the n = 1 problem indicate that the fortune seeker should go initially to either state C or state D. Suppose that he chooses x_1^* = C. For n = 2, the optimal result for s = C is x_2^* = E. This result leads to the n = 3 problem, which gives x_3^* = H for s = E, and the n = 4 problem yields x_4^* = J for s = H. Hence, one optimal route is A → C → E → H → J.
Choosing instead x_1^* = D leads to the other two optimal routes A → D → E → H → J and A → D → F → I → J. They all yield a total cost of $f_1^*(A) = 11$. These results of the dynamic programming analysis also are summarized in Fig.A.2:

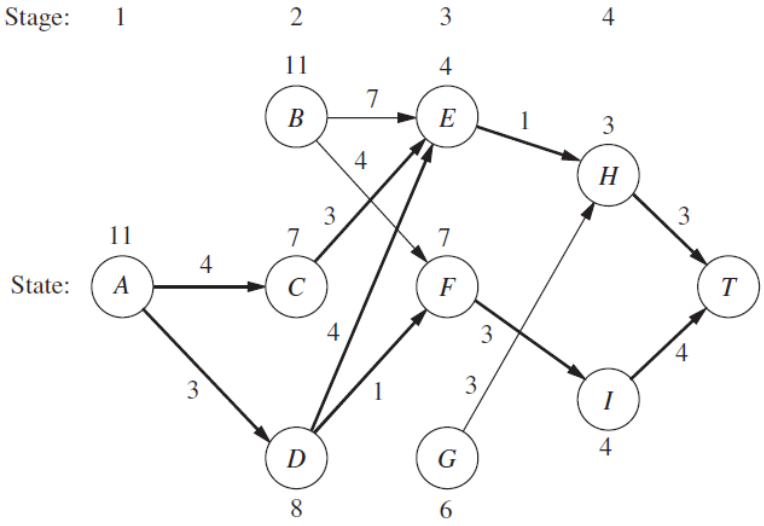

Figure A.2: Stagecoach problem solution visualization. The figure is taken from Frederick and Hillier work [16].

Figure A.2 also highlights one important property of the dynamic programming approach: besides providing the optimal policy for the overall problem, an optimal decision policy is also available for any state in the network, regardless of this state being ever reached. The prescription of an optimal policy for each state and at each stage is a typical feature

of dynamic programming that can be useful in several ways, including sensitivity analysis [16].

Bibliography

[1] G. B. V. A. Carusi and R. Greenberg. Planetary close encounters: Geometry of approach and post-encounter orbital parameters. *Celestial Mechanics and Dynamical Astronomy 49, pp 111-131*, 1990.

[2] Y.-C. H. A. E. Bryson. *Applied optimal control*. Taylor Francis, New York, ISBN: 978-0891162285, 1975.

[3] M. R. A. Masat, C. Colombo. B-plane orbital resonance analysis and applications: Perturbed semi-analytical model for planetary protection and defence applied to ballistic resonant flyby design, scuola di ingegneria industriale e dell'informazione, daer - dipartimento di scienze e tecnologie aerospaziali, politecnico di milano, 2019.

[4] K. G. S. A. V. Labunsky, O. V. Papkov. *Multiple gravity assist interplanetary trajectories*. Earth Space Institute Book Series, Gordon and Breach Science Publishers, ISBN: 90-5699-090-X, 1998.

[5] E. M. Alessi and J. P. Sanchez. Semi-analytical approach for distant encounters in the spatial circular restricted three-body problem. *Journal of Guidance, Control, and Dynamics*. doi: https://doi.org/10.2514/1.G001237.

[6] R. L. Anderson. Resonant flyby and tour design using heteroclinic connections. *American Astronomical Society, 43th annual DDA meeting 53.4*, 2012.

[7] R. Bellman. *Dynamic Programming, chapter 10, pp 317-319*. Dover Publications, 2003. ISBN 978- 0-486-42809-3.

[8] N. Bradley and R. Russell. A continuation method for converting trajectories from patched conics to full gravity models. *Journal of the Astronautical Sciences 152*, 2014. doi: 10.1007/s40295-014-0017-x.

[9] M. V. C. Colombo and G. Radice. Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming. *elestial Mechanics and Dynamical Astronomy*, 2009.

[10] S. Campagnola and R. P. Russell. Endgame problem part 2: multibody technique

and the tisserand-poincarè graph. *Journal of Guidance, Control, and Dynamics*. doi: https://doi.org/10.2514/1.44290.

[11] H. D. Curtis. *Orbital Mechanics for Engineering Students*. Elsevier, 2014.

[12] S. R. V. D. J. Wirthman, S. Y. Park. Trajectory optimisation using parallel shooting method on parallel computer. *Journal of Guidance, Control, and Dynamics, vol. 18, n. 2, p. 377-379*, 1995.

[13] B. Dachwald. Optimisation of solar sail interplanetary trajectories using evolutionary neurocontrol. *Journal of Guidance, Control, and Dynamics, vol. 27, n. 1, p. 66-72*, 2004.

[14] C. C. Davide Menzio. Adapted syzygy functions for the preliminary design of multiple gravity assisted trajectories. *69th International Astronautical Congress, Bremen, Germany, IAC–18–C1.9.12*, 2018.

[15] Ernst J. Öpik. *Interplanetary Encounters Close-Range Gravitational Interactions*. Elsevier Scientific Publishing Company, 1976.

[16] G. J. L. Frederick S. Hillier. *Intro to operation research*. McGraw-Hill, 2010.

[17] C. F. G. B. Valsecchi and R. Gonczi. Modelling close encounters with Öpik's theory. *Elsevier Science Ltd, Planet. Space Sci., Vol. 45, No. 12, pp. 1561-1574,*, 1997.

[18] A. M. G. Campiti, C. Colombo. Resonant flybys in the b-plane: extension of the theory to elliptical planetary orbits and dynamic programming application, Politecnico di Milano, 2021.

[19] O. Grasset, M. Dougherty, A. Coustenis, E. Bunce, C. Erd, D. Titov, M. Blanc, A. Coates, P. Drossart, L. Fletcher, H. Hussmann, R. Jaumann, N. Krupp, J.-P. Lebreton, O. Prieto-Ballesteros, P. Tortora, F. Tosi, and T. V. Hoolst. Jupiter icy moons explorer (juice): An esa mission to orbit ganymede and to characterise the jupiter system. *Planetary and Space Science, volume 78, pages 1-21*, 2013.

[20] D. Hennes and D. Izzo. Interplanetary trajectory planning with monte carlo tree search. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), Noordwijk, The Netherlands*, 2015.

[21] R. E. J. Kennedy. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, Perth, Australia*, 1995.

[22] S. O. E. J. T. Betts. Optimal low thrust trajectories to the moon. *SIAM Journal of Applied Dynamical Systems, vol. 2, n. 2, p. 144-170*, 2003.

[23] Y. Z. L. K. P. Lin and G. J. Tang. Space station orbit design using dynamic programming. *Acta Astronautica 89*, 2013.

[24] G. Lantoine and R. Russell. A hybrid differential dynamic programming algorithm for robust low-thrust optimization. *Astrodynamics Specialist Conference and Exhibit*. doi: 10.2514/6.2008-6615.

[25] Y. Liu, R. Noomen, and P. Visser. A gravity assist mapping for the circular restricted three-body problem using gaussian processes.

[26] M. V. M. Ceriotti. Global optimisation of multiple gravity assist trajectories, department of aerospace engineering, faculty of engineering university of glasgow, 2010.

[27] M. L. M. Vasile, E. Minisci. A dynamical system perspective on evolutionary heuristics applied to space trajectory optimization problems. *Proceedings of IEEE Congress on Evolutionary Computation Trondheim, Norway*, 2009.

[28] P. D. P. M. Vasile. Preliminary design of multiple gravity-assist trajectories. *Journal of Spacecraft and Rockets, vol. 43, n. 4, p. 794-805*, 2006.

[29] S. L. . L. J. Matson, D.L. The cassini/huygens mission to the saturnian system. *Space Science Reviews 104, 1–58*, 2002.

[30] J. M. L. N. J. Strange. Graphical method for gravity-assist trajectory design. *Journal of Spacecraft and Rockets, vol. 39, n. 1, p. 9-16*, 2002.

[31] M. Nugnes and C. Colombo. Low-thrust trajectory optimisation through differential dynamic programming method based on keplerian orbital elements. *70th International Astronautical Congress: IAC Proceedings*, 2019. URL `http://hdl.handle.net/11311/1117583`.

[32] D. I. Pierluigi Di Lizia, Gianmarco Radice and M. Vasile. On the solution of interplanetary trajectory design problems by global optimisation methods. *Proceedings of GO, pp. 1 - 7.*, 2005.

[33] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley Sons, Inc, 2007. ISBN 978-0-470- 17155-4.

[34] S. D. Ross and D. J. Scheeres. Multiple gravity assists, capture, and escape in the re-stricted three-body problem. *SIAM Journal on Applied Dynamical Systems*. doi: https://doi.org/10.1137/060663374.

[35] P. S. Stefano Campagnola and R. P. Russell. *Flybys in the planar, circular, restricted, three-body problem.* Celestial Mechanics and Dynamical Astronomy, 2010.

[36] D. W. Stroock. *An Introduction to Markov Processes.* Springer, Berlin, Heidelberg, 2014. ISBN 978-3-540-26990-8.

[37] F. F. Tisserand. *Traité de Mécanique Céleste. Vol. 4.* Gauthier-Villars, Paris, ISBN: 026639955X., 1896.

[38] M. Van Casteren, J. Novara. Bepicolombo mission. *Memorie della Societa Astronomica Italiana, v.82, p.394*, 2011.

[39] M. Vasile. A global approach to optimal space trajectory design. *Advances in the Astronautical Sciences, vol. 114, n. SUPPL, p. 621-640*, 2003.