Astos Solutions

Optimal Trajectories in the Earth-Moon CR3BP system

Master of Science Thesis

J. Tatay Sanguesa

Technische Universiteit Delft

TU Delft

The image on the cover page has been retrieved from https://www.esa.int/ESA_Multimedia/Images/2019/05/Gateway_and_Moon. Image credit: ESA.

Optimal Trajectories in the Earth-Moon CR3BP system

Master of Science Thesis

by

J. Tatay Sanguesa

to obtain the degree of Master of Science
at Delft University of Technology
to be defended publicly on Wednesday July 7th, 2021 at 9:00 AM.

Student number: 5022207
Project duration: December 2nd, 2020 – June 1st, 2021
Supervisors: Ir. R. Noomen, TU Delft, supervisor
Herr A. Wiegand, Astos Solutions GmbH

An electronic version of this literature study is available at http://repository.tudelft.nl/.

TU Delft

Preface

This research determines the final step in my university education. Although it feels like yesterday, long behind are those first days at the Universitat Politècnica de València. And the truth is, as they say, that time flies when you are having fun. Besides the university parties and celebrations, we will come back to that, I really have enjoyed my learning experience. Even though most of the topics I have learned I will probably not use in my professional life they are fulfilling, at least, an equally important role in my life: satisfying my curiosity of how things work, how planes fly and how bodies move in space.

It was not until the end of my Bachelors that I realized that this was my true passion, space trajectories. I remember learning about this topic and imagining all the planets, satellites, stars and other bodies moving by the law of gravitation as if they were dancing an eternal waltz. I was determined to learn as much as I could about the topic and I knew TU Delft was the right place for it.

Although I was confident that the education here would be outstanding, my expectations were exceeded. Not only by the specificity of the courses and the level of detail in the lectures, but for a much simpler and powerful reason. There was no barrier between professors and students, and the university actively promoted activities to help us not just become colleagues, but friends. Of course, this was also possible thanks to the attitude of the professors themselves, who are willing to share their knowledge, experience and even a can of chips with everyone. This brings me to the first person I would like to thank, my thesis supervisor Ron Noomen, whom I believe is the maximum expression of this behavior I just described. From the very first day he wanted us to know that students and professors were at the same level. He also made sure to build a relationship with the human behind the Thesis report, always showing genuine interest for other aspects of my life, not only discussing work.

I would also like to thank Andreas Wiegand, my Thesis supervisor and future director from Astos Solutions. He trusted me, a master student with no working experience, to collaborate with his company on developing this Master Thesis project. He also had the patience to teach me and support me along the way every time I needed help. It was very reassuring to have this support, especially during Corona times, when it could have been easy to lose focus.

Of course, I could never forget my friends, all of them. From the very old to the newest ones. But I would especially like to mention Marc and Irene, which have been by my side during every important moment, every tough deadline and every celebration afterwards, and I have no doubt they will continue to be there whatever may come our way.

Finally, I would like to express my sincerest gratitude to my family. To my parents, who have supported all my projects since the first moment. To my siblings, all three of them, who I know will be at this same moment very soon. And also to my grandmother, with her mailings both digital and physical, which made me feel home a little closer.

Thank you very much to all of you. I hope you enjoy this work as much as I have done.

J. Tatay Sanguesa
Delft, June 2021

Contents

List of Abbreviations

ACE	Advanced Composition Explorer
ACO	Ant Colony Optimization
ASTOS	Analysis, Simulation and Trajectory Optimization Software for Space Applications
CR3BP	Circular Restricted Three-Body Problem
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DRO	Distant Retrograde Orbit
DSCOVR	Deep Space Climate Observatory
ER3BP	Elliptical Restricted Three-Body Problem
ESA	European Space Agency
GEO	Geostationary Earth Orbit
GTO	Geostationary Transfer Orbit
GUI	Graphical User Interface
HALO	Habitation and Logistics Outpost
ISEE-3	International Sun-Earth Explorer-3
JPL	Jet Propulsion Laboratory
JWST	James Webb Space Telescope
LEO	Low Earth Orbit
LLO	Low Lunar Orbit
MAP	Microwave Anisotropy Probe
MATLAB	MATrix LABoratory
MIDACO	Mixed Integer Distributed Ant Colony Optimization
NASA	National Aeronautics and Space Administration
NRHO	Near-Rectilinear Halo Orbit
OOP	Object-Oriented Programming
PDF	Probability Density Function
PPE	Power and Propulsion Element
RAAN	Right Ascension of the Ascending Node
SOHO	Solar and Heliospheric Observatory
SRP	Solar Radiation Pressure
STM	State Transition Matrix
TOF	Time Of Flight
ZVC	Zero Velocity Curve

List of Notations

Greek Symbols

Δv	Velocity change, measure of total propellant [m/s]
λ	Eigenvalue
μ	Primaries mass ratio [-]
ν	Stability index [-]
Ω	Right Ascension of the Ascending Node (RAAN) [rad]
ω	Argument of periapsis [rad]
θ	True anomaly [rad]

Latin Symbols

$\dot{x}$	Normalized velocity component along the X-axis [-]
$\dot{y}$	Normalized velocity component along the Y-axis [-]
$\dot{z}$	Normalized velocity component along the Z-axis [-]
T	Orbit period [s]
v	Eigenvector
$\mathbf{x}_0$	Initial state
a	Semi-major axis [m]
A_x	Periodic solution normalized X-amplitude [-]
A_y	Periodic solution normalized Y-amplitude [-]
A_z	Periodic solution normalized Z-amplitude [-]
a_{pert}	Perturbing acceleration [m/s^2]
C	Jacobi Constant [-]
d	Manifold perurbation [-]
e	Eccentricity [-]
f_m	Objective function
G	Universal Gravitational constant = $[6.67408 \cdot 10^{-11}$ m^3/kg/s]
g_j	Inequality constraint
h_k	Equality constraint
i	Inclination [rad]
I_{sp}	Specific impulse [s]
L_i	Lagrangian point i, where $i = 1 \dots 5$

M Monodromy matrix

m_1 First primary. Most massive body of the system

m_2 Second primary

m_3 Third (massless) body, the spacecraft

m_i Mass of the body of interest [kg]

m_j Mass of external body [kg]

m_q Mass of the body at the origin of the inertial system [kg]

p Parameter used in continuation schemes

r Distance [m]

r_a Radius of the apoapsis [m]

r_p Radius of the periapsis [m]

r_{ij} Distance from i to body j [m]

STM State transition matrix

v Velocity [m/s]

x Normalized position component along the X-axis [-]

x_i Design variable

y Normalized position component along the Y-axis [-]

z Normalized position component along the Z-axis [-]

Sub- and Superscripts

^{s} Stable branch

^{u} Unstable branch

$^{(L)}$ Lower limit

$^{(U)}$ Upper limit

Abstract

The aim of this investigation is to improve the first stage of mission analysis on trajectories in the Circular Restricted Three-Body Problem (CR3BP). This will be achieved by developing a versatile tool that can optimize any transfer in the CR3BP in terms of ΔV and time-of-flight. Hence, the output will not be a single solution, but a Pareto front with multiple non-dominated solutions that range in duration and propellant consumption, such that the user can easily identify the one that better suits his/her particular mission requirements.

The tool considers three types of arcs: direct transfers, manifold trajectories and flyby arcs which can be combined in any way to connect the departure and destination orbits. These can be defined as Keplerian orbits or CR3BP periodic solutions, including Lyapunov, Halo orbits or NRHOs. This is done in a very intuitive way thanks to the careful selection of design variables. Therefore the developed tool can be used by designers that do not have a large amount of experience solving trajectory optimization problems in the CR3BP.

In order to demonstrate the capabilities of the tool, two specific transfers were optimized. In both cases, the Pareto front is obtained with the combination of optimal solutions that include direct transfers, and manifold and flyby arcs. First, a LEO to L_2 Halo orbit with $A_z = 2000$ km was selected to compare the results obtained by the software with those from literature. The tool not only obtained more results than the previous research, but the solutions found were also improved in terms of ΔV and time of flight. The second transfer was a GTO to lunar Gateway NRHO trajectory, which was chosen to show the applications of this study to problems of high scientific and industrial interest in present times, as well as to prove the versatility of the algorithm. Again, the results are better than the ones found in literature, if only by the number of different solutions that are obtained. Moreover, these results can easily be exported into a higher-fidelity software such as ASTOS.

1 Introduction

This chapter contains the introduction to this investigation. First, a background on space exploration and the upcoming lunar activities is shown. Then, the objectives of this study are described. Finally, a quick outline of the following chapters is presented.

1.1. Background

The Earth and the Moon region in space offers a unique dynamical environment in the near-by part of the universe. As the mass ratio of both primaries is relatively large when compared to other known planet-moon systems, the motion of a spacecraft in the Earth-Moon proximity can be heavily influenced simultaneously by both bodies. Hence, the Circular Restricted Three-Body Problem (CR3BP) can be especially effective and a powerful formulation for mission design in this region. By making use of multiple gravity fields at the same time, the CR3BP reflects more accurately the actual dynamical environment and also smooths a transition to higher-fidelity models. Hence, the trajectories identified during this initial design phase, are a suitable initial guess for the final trajectories computation, as they retain most of their characteristics. The CR3BP also offers the possibility of very cheap transfers, as the gravitational pull of both bodies can be considered to favor the transfer, rather than as perturbations that need to be corrected.

Besides the mathematical advantages and challenges that the CR3BP presents, the Moon being the body closest to planet Earth is a fact that has not been ignored through history. From astronomers who tried to explain the motion of the Moon by observing it since early times, to the culmination of the space race in 1969 with the Apollo 11 lunar landing, the proximity of the Moon has made it an object of interest throughout human history. Although humans last walked on the Moon surface in 1972, the interest in manned lunar missions has been renewed in recent times. Orbits in the vicinity of the Earth-Moon equilibrium points have also garnered interest in the last few years. Mainly, because their orbital characteristics offer applications as both scientific observation and communications network to support human facilities on the Moon surface. The pinnacle of these missions will most likely be the Artemis program developed by the National Aeronautics and Space Administration (NASA) whose main achievement is to develop an orbital outpost around the Moon that provides vital support for a sustainable, long-term human return to the lunar surface, as well as a staging point for deep space exploration [55]. This outpost, known as the lunar Gateway will set a new chapter in humanity's exploration history.

In the same way as early explorers that sailed far from their home countries to satisfy their curiosity about an unknown world, humanity is now starting to explore other planets. Human exploration of Mars has become a strong possibility to come in the following decades, but for now, the focus is placed on the Moon, and that is where this investigation will place its focus on as well. In the next section, the main research objectives are presented and summarized into a single research question.

1.2. Research Objectives

The aim of this work is to develop a tool that can improve the space exploration process. To do so, the mission analysis task regarding trajectory optimization in the CR3BP was studied and several potential advancements were observed.

Firstly, most of the previous work in the field considers the propellant consumption (usually represented by the ΔV) to be the only relevant objective in mission analysis, and thus focus on minimizing this ΔV. Nevertheless, there are other relevant objectives in trajectory optimization: at the very least, the time of flight should also be considered. Hence, this investigation will approach transfer optimization in the CR3BP from a multi-objective perspective, rather than a single-objective one. The aim of the tool, then, is not to obtain a single solution, but rather provide the mission analyst with a converged set of optimal solutions, with different ΔV and TOF values, such that the user can decide which of them better satisfies the requirements. This also restricts the usefulness of the tool to the first stages of the mission analysis process, where it is preferable to obtain a first approach result of a large number of solutions, rather than a very detailed single solution.

Another problem that was identified is that most of the work is focused on finding optimal results for a single trajectory, in some cases a very specific one, such as the transfer from a LEO to the Gateway orbit. Although this is understandable given the relevance of such transfers, a versatile tool where the user could easily change any of the transfer characteristics and adapt it to his or her own problem would be a clear improvement for the mission analysis process. Therefore, the flexibility of the tool and its capability of adapting to all transfers within the CR3BP problem became the second objective.

Also, it was recognized that the tool output should be compatible with higher-fidelity tools, such that the optimal trajectories obtained by the algorithm that were selected by the mission analyst, could be easily studied more in depth. Thus, the next objective is to make the tool compatible with more advanced software.

Finally, in an attempt to reach a wider range of users, it was decided to add a last objective: to construct an intuitive tool such that the user would not need to have a large prior knowledge nor be a specialist in the CR3BP.

These objectives can be summarized in the following research question:

> Is there a user-intuitive way to optimize a wide range of trajectories in the CR3BP model from a multi-objective perspective?

This question can be split in a series of subquestions:

> How will the tool be user-intuitive?
>
> What constitutes a wide-range of trajectories in the CR3BP? Is this limited to departure and destination orbits options, or also to transfer arcs?
>
> Which multi-objective algorithm is best suited to solve this problem?

1.3. Project Outline

In order to answer the previous research question, several steps were followed. The present report was made to summarize the procedure and conclusions of the investigation. It is organized in the following way:

- **Chapter 2** summarizes the state of the CR3BP trajectory optimization in the literature. This includes a brief historical review from the first mathematical descriptions of such problem, to remarkable space missions that took place in recent years; both in the Earth-Moon CR3BP and in other systems. Also, the ASTOS software is introduced, as this advanced software will be used to further investigate the solutions obtained.
- **Chapter 3** provides a complete description of the background knowledge required to understand the trajectory optimization concepts exposed. This includes the assumptions, geometry and equations of motion in the CR3BP and ephemeris physical model, as well as the coordinate transformation calculations between them. Moreover, a discussion on multi-objective optimization and optimization algorithms is given here.

- **Chapter 4** lists the different orbits that have been included into the tool capabilities. These range from Keplerian orbits, to CR3BP periodic solutions, including Lyapunov, Halo orbits and NRHOs.
- Once the orbits have been defined, **Chapter 5**, describes the different trajectory arcs that can be employed to connect such orbits. Three possibilities are investigated in detail: direct transfers, manifold solutions, and flyby transfers.
- **Chapter 6** presents the optimization problem and explains how to define the characteristics of the transfer that will be optimized. This is done with input files. Then, the output files are shown, describing how the information is received by the user, and how can it be exported into external tools such as ASTOS.
- **Chapter 7** explains the inner workings of the developed software, mainly focusing on the design variables and the relations between them. Furthermore, it also describes the optimization process as a whole.
- Once the entire procedure followed has been explained, **Chapter 8** consists of the verification and validation of the different parts of the tool. In the end, it reproduces a result from literature to prove that the tool functions correctly.
- After the tool is verified and validated, **Chapters 9 and 10** present the results for the selected trajectory problems. This includes a multi-objective optimization with the corresponding Pareto front, and a deeper analysis on the most promising solutions.
- Finally, **Chapter 11** contains the conclusions of the investigation and a set of recommendations for future work.

2

Heritage

This chapter summarizes the scientific development evolution with regard to the CR3BP as well as its current state. Firstly, the general development of the model is described, since the seventeenth century until the latest advances. Secondly, a review of the principal space missions that have made used of the CR3BP, both for scientific and for industrial purposes, is presented. Moreover, a separate section is dedicated to the particular applications of this model in the Earth-Moon system, especially with respect to the Gateway project. Finally, the ASTOS software is introduced, summarizing its history and listing its capabilities.

2.1. History of the CR3BP

The CR3BP has been of interest to the scientific community for several centuries. It was first examined by Newton in the *Principia Mathematica* (1687) [41], where he also stated his Law of Gravitation. However, the general three-body problem was first formally stated by Euler in 1727. He recognized its complexity and was able to identify the three collinear equilibrium solutions after formulating the problem in a rotating frame. Lagrange followed this work and was able to find the location of the triangular equilibrium points in 1772. Since then, the libration points have also been known as Lagrangian points.

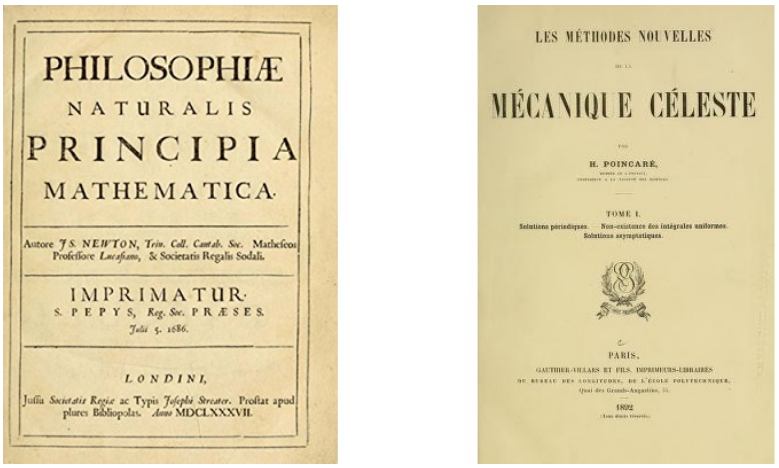

Figure 2.1: Newton's *Principia Mathematica* (left) [41] and *Les Méthodes Nouvelles de la Mécanique Celeste* by Poincare (right) [51].

Further renowned mathematicians have had their contributions to the development of the CR3BP, its formulation and solutions. Jacobi [2] found an integral of motion for Euler's problem formulation, which is named after him (Jacobi constant). Hill showed in 1878 that this constant can be used to show restricted regions in the CR3BP, the Zero Velocity Curves (ZVCs). Poincaré proved that the Jacobi integral is the only integral of motion of the system, and introduced in *Les Méthodes Nouvelles de la Mécanique Celeste* [51] a new qualitative analysis for the chaotic behavior.

In the 1960s and 1970s, with the space race, the CR3BP saw a great development in its theoretical studies for mission applications. It is worth to note the work by Farquhar in 1968 [10] where he found Halo orbits while studying the use of collinear Lagrangian points for lunar communication and station-keeping. Also, recent decades saw the development of manifold trajectories associated with these libration points and the periodic solutions around them, where Howell [20, 27] and Simó [65] should be mentioned. In 1967, Szebehely published the book *Theory of Orbits* [66], compiling all the information available at the time regarding the CR3BP.

However, it has only been during the last decades, especially due to the improvement in numerical integration and computing power, that these types of motions have been fully understood for all kinds of dynamical systems, as explained in [69]. These advances have enabled the use of the CR3BP model to develop several space missions, past and future, which will be discussed in the next section. Nowadays, the CR3BP is mainly used in the initial stages of mission design, as the accuracy requirements of most missions demand higher fidelity models. Nonetheless, there is ongoing investigation in alternative three-body models, such as the Elliptical Restricted Three-Body Problem (ER3BP), which are not constrained by so many assumptions.

2.2. Representative Missions

One of the main advantages of the CR3BP with respect to the two-body model, is the existence of equilibrium solutions called libration points (see Chapter 3). In principle, they allow the spacecraft to maintain the same relative position with respect to two bodies with hardly any stationkeeping cost. As would be expected, these locations have attracted scientific interest in the last years because they offer great advantages as scientific observation posts and as part of a bigger communications network between the two bodies, for instance the Earth and the Moon. In the following paragraphs, a collection of the most important missions to and around libration points is presented.

The ISEE-3 spacecraft [11] launched in 1978 was the first spacecraft that used a libration point as mission destination, as it was delivered to the vicinity of the Sun-Earth L_1. Besides visiting a comet for the first time, its main objective was to study the interaction between Earth's magnetic field and solar wind, as part of a joint effort between ESA and NASA. There have been four other missions deployed to the Sun-Earth L_1 point since the ISEE-3 mission, whose purpose has also been to study the Sun's composition, solar wind and space weather, among other topics. They are WIND [12], SOHO [53], ACE [63], and DSCOVR [3].

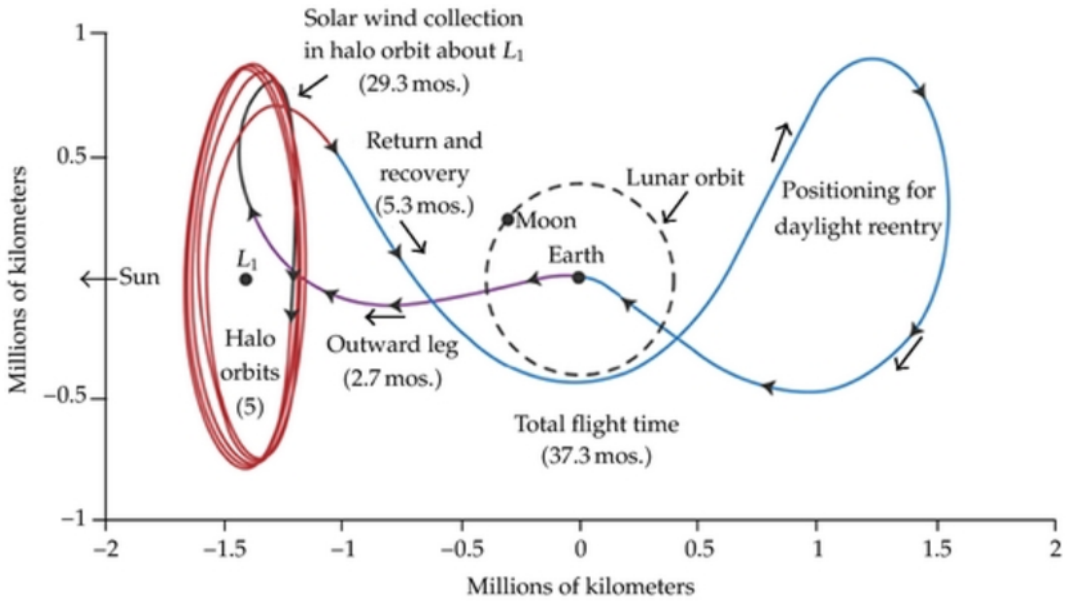

Figure 2.2: Genesis trajectory as viewed in the Sun-Earth rotating frame. [19]

Nevertheless, there exist other spacecraft that have been sent to different Lagrangian points over the last decades or will be in the near future. One of the most notable examples is the James Webb Space Telescope (JWST) [30], successor of the Hubble Space Telescope. By being located around the Sun-Earth L_2 point, it will be able to look into outer space without any obstruction from the Earth or the Sun. Other missions that have been deployed to the Sun-Earth L_2 point are MAP [29] and the Gaia mission [8].

Finally it is important to mention the Genesis mission [27], as it not only had an L_1 Lissajous orbit as its destination, but it also employed a heteroclinic transfer by traveling to L_2 before returning to Earth. This, apart from reducing the propellant cost, enabled a daytime recovery of the capsule.

Overall, it can be seen that the applications of the three-body dynamics are numerous. Advantages range from reducing transfer costs to increasing launch and return windows, or facilitating the achievement of mission objectives.

2.3. Earth-Moon Three-Body Problem

Although the majority of the CR3BP related missions have been performed in the Sun-Earth system, NASA has recently proposed to build a new platform for space, Earth and lunar science: the Gateway. This station will also be used as a staging point for missions to the Moon surface and beyond cis-lunar space [71], and as a testing site for deep-space technologies [16]. An illustration of the Gateway space station can be seen in Figure 2.3.

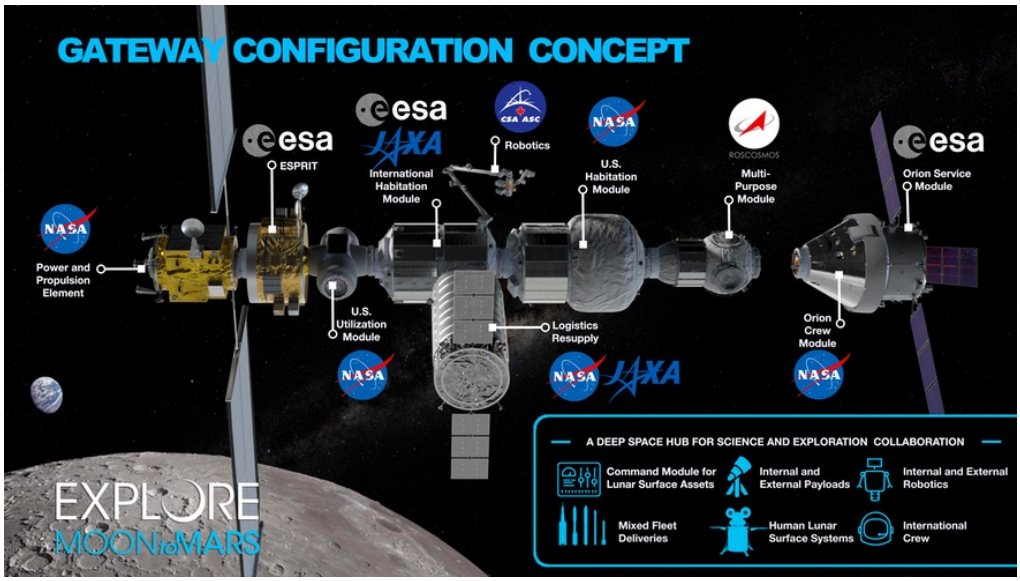

Figure 2.3: Illustration of the lunar Gateway [39].

All of these applications translate in a great variety of potential transfers from the Gateway to other orbits and viceversa. This has lead to a great development in the Earth-Moon CR3BP theory and research, as the number of missions planned within this system has increased significantly. In this section, the transfers of major scientific and industrial interest are presented, as they embody the key trajectories that the developed tool should be able to compute and optimize.

The main trajectory of interest is be the transfer from Earth to the Gateway orbit. Starting in 2023, the first two modules, the Power and Propulsion Element (PPE) and the Habitation and Logistics Outpost (HALO), of the Lunar Gateway will be launched, followed by a great number of missions, including crewed ones. Hence, it logically follows that the next key trajectory needed would be the return trajectory from the Gateway to the Earth. Nonetheless, as the Gateway will be a scientific platform and a staging point for lunar surface missions, transfers from the NRHO to all sorts of orbits in the CR3BP will be required. These may include, among others, Low Lunar Orbits (LLOs), Distant Retrograde Orbits (DROs), Halo and Lyapunov orbits, and Lissajous orbits. Of course, the return trajectories from these to the Gateway are also required. Trajectories to other destinations, such as Mars, are not considered in the current research.

A brief look at literature shows that transfers from Earth to the Gateway are, by far, the most studied topic, followed by transfers from the Gateway to the lunar surface [31, 72]. However, few papers study the return trajectory from the NRHO, with the majority focusing on other destination orbits, such as LLOs, or referring to the *Theorem of Image Trajectories* [33, 34]. Nevertheless, although optimal return trajectories can be obtained as the mirror image of the optimal departure trajectory, it must be reminded that there is no single optimum in space travel, as the ΔV and time of flight requirements may differ. Finally, missions from the Gateway orbit to other CR3BP orbits, such as Lissajous, have barely been studied, as they are also the least significant for industry.

Given the foreseeable demand and the amount of available research that can be used to compare the results obtained, it has been decided to focus this research towards transfers within the most remarkable orbits in the CR3BP, which include Lyapunov and Halo orbits (including NRHOs), as well as Keplerian orbits.

2.4. ASTOS Software

To conclude this chapter, the ASTOS software will be introduced. ASTOS stands for Analysis, Simulation and Trajectory Optimization Software for Space Applications, and as its acronym indicates, it is a commercial software used for space mission and trajectory optimization [13, 70]. Its development started at the Deutsches Zentrum für Luft- und Raumfahrt (DLR) and continued at the University of Stuttgart, before becoming further developed under an independent company, Astos Solutions GmbH.

The main strength that ASTOS has is its multi-purpose capability, which enables it to provide solutions for the whole space project life-cycle. Among its wide range of supported scenarios, some include launch and re-entry safety analysis, multi-disciplinary trajectory optimization, operational life-time prediction, etc.

Regarding this project, ASTOS was the tool selected to improve the capabilities of the developed software and overcome the limitations of the CR3BP assumptions. Therefore, not only the resulting trajectory is valid in the CR3BP system, but the output is designed such that it is completely compatible with an ASTOS scenario. In this way, after a multiple-shooting algorithm converts the CR3BP trajectory into a higher-fidelity model, with the specifications that the user decides, a refined optimization can take place inside ASTOS, improving not only the validity of the results, but also easily adding other objectives such as minimizing eclipse or coverage time.

Moreover, the trajectory analysis can go another step further and include a more accurate propulsive model such as continuous thrust, or analyze the maneuvers necessary for guidance, navigation and control.

Overall, being able to combine the results with ASTOS does not limit the results obtained to the theoretical CR3BP, but rather provides a very accurate initial guess for further optimization and study in a higher fidelity model.

It is important to be noted, however, that the tool has been developed completely independently from the ASTOS software, which can be used for more detailed follow-up studies.

3

Theoretical Background

This chapter presents the main theoretical knowledge required to understand the calculations performed by the optimizing tool. First, the CR3BP model is presented, its accuracy is analyzed and further alternatives are presented. The next section deals with differential corrections methods that are used throughout the project. Finally, the last section discusses the optimization algorithms employed.

3.1. Circular Restricted Three-Body Problem

In this first section, the physical model that will be used throughout the thesis is described. First, the assumptions required and the reference frames used are shown. Then, the physical model is further explored to look for equilibrium and periodic solutions and analyze their stability.

3.1.1. Equations of Motion

The CR3BP studies the motion of a body with negligible mass m_3 in the gravity well formed by the two primaries with mass m_1 and m_2. These massive bodies move in circular orbits around their common barycenter. As this problem is *circular*, the orbits of the primaries have constant angular velocity. It is also *restricted* because the spacecraft does not influence the motion of the primaries. Moreover, all bodies are assumed to be perfectly symmetric and with uniform density, so they can be approximated as point masses.

Putting together all these conditions the CR3BP is analyzed. In order to build a coordinate system, some considerations are made. First, the magnitudes are nondimensionalized with respect to some relevant quantities in the problem. Table 3.1 summarizes the main physical quantities (mass, distance and time) that are used in the Earth-Moon CR3BP system. All quantities have been obtained from the Jet Propulsion Laboratory (JPL) ephemeris database [54]. These transformations give the universal gravitational constant G a dimensionless value of one.

- The distance unit is the distance between the primaries.
- The angular velocity of the relative motion of the primaries is assumed to be one. Hence, the orbital period of the primaries motion is 2π.
- The mass unit is the sum of the two primaries' masses. A dimensionless parameter μ is obtained as $m_2/(m_1 + m_2)$, such that $m_2 = \mu$ and $m_1 = 1 - \mu$.

Secondly, a rotating reference frame is chosen to represent the CR3BP. The angular rotation rate of the system n is set to one and the primaries are fixed on the X-axis. The origin is defined at the barycenter of the massive bodies. The XY-plane is defined as the orbital plane of the primaries, with the Y-axis being perpendicular to X. Finally the Z-axis is perpendicular to the orbital plane, and aligned with the angular momentum vector of the system. Figure 3.1 represents the reference frame described. Here note that μ is not only the mass ratio, but also the distance from the first primary (Earth) to the barycenter.

	Value	Units	Dimensionless Value
Earth Gravitational Parameter	398600.4356	km^3/s^2	0.9878494
Moon Gravitational Parameter	4902.801	km^3/s^2	0.0121506
Earth Radius	6378.1363	km	0.0165924
Moon Radius	1738.0	km	0.0045213
Earth-Moon Distance	384400.0	km	1.0
Earth-Moon Sidereal Period	27.32166	days	2π
Earth-Moon Angular Velocity	0.21277	rad/day	1.0

Table 3.1: Earth-Moon CR3BP Physical Parameters [54].

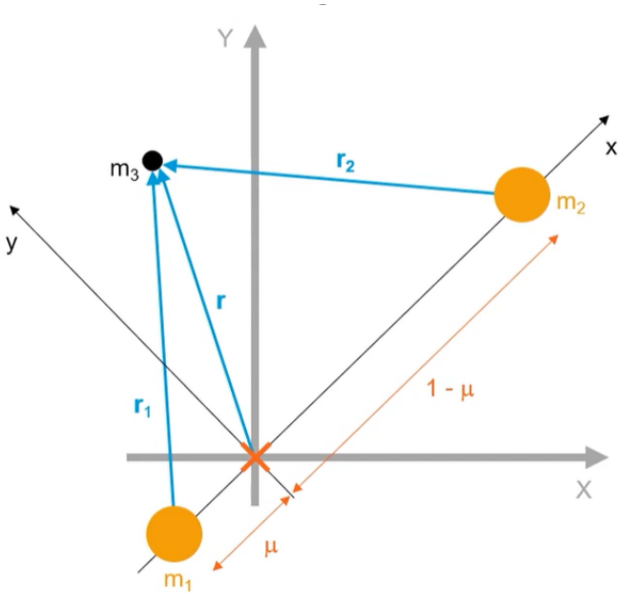

Figure 3.1: Geometry of the problem, the sidereal $(X; Y)$ and the synodic $(x; y)$ reference frames. [17]

Using elementary mechanics, the three second-order differential equations that govern the motion of the third body can be found. They are shown in Equation (3.2). For a full derivation and explanation of the equations of motion please refer to Szebehely, 1967 [66].

$$\begin{aligned} \ddot{x} - 2\dot{y} &= \Omega_x \\ \ddot{y} + 2\dot{x} &= \Omega_y \\ \ddot{z} &= \Omega_z \end{aligned} \tag{3.1}$$

where the subscripts denote partial derivatives of the function:

$$\Omega(x, y, z) = \frac{x^2 + y^2}{2} + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2} \tag{3.2}$$

and:

$$\begin{aligned} r_1^2 &= (x + \mu)^2 + y^2 + z^2 \\ r_2^2 &= (x - 1 + \mu)^2 + y^2 + z^2 \end{aligned} \tag{3.3}$$

As this is a non-inertial coordinate system, note that some apparent forces show up, namely the centrifugal (first term) and Coriolis potential (second term). This system will have only one integral of motion: the Jacobi constant, which is given by Equation (3.4).

$$C = 2\Omega(x, y, z) - \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) \tag{3.4}$$

Since $\left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2\right) \geq 0$, the Jacobi constant determines some forbidden regions in the CR3BP space which are known as ZVCs or Hill surfaces.

3.1.2. Libration Points

The accelerations, gravitational and apparent, cancel each other in five different points, known as Lagrangian points. In order to obtain their coordinates, the following steps can be followed.

Substituting velocity and acceleration coordinates by zero in the equations of motion yields the result $\nabla\Omega = 0$. Deriving the gradient of the potential in Equation (3.5) it can be seen from the third equation that $z_{eq} = 0$. Hence, the equilibrium solutions will lie in the orbital plane of the primaries.

$$\begin{aligned}
\frac{\partial\Omega}{\partial x} &= -\frac{(1-\mu)\left(x_{eq}+\mu\right)}{r_{1_{eq}}^3} - \frac{\mu\left(x_{eq}-1+\mu\right)}{r_{2_{eq}}^3} + x_{eq} = 0 \\
\frac{\partial\Omega}{\partial y} &= -\frac{(1-\mu)y_{eq}}{r_{1_{eq}}^3} - \frac{\mu y_{eq}}{r_{2_{eq}}^3} + y_{eq} = 0 \\
\frac{\partial\Omega}{\partial z} &= -\frac{(1-\mu)z_{eq}}{r_{1_{eq}}^3} - \frac{\mu z_{eq}}{r_{2_{eq}}^3} = 0
\end{aligned} \tag{3.5}$$

Moreover, it can be recognized that there will be some solutions for which $y_{eq} = 0$. In fact, a fifth-order polynomial appears, although after iterative solving, only three solutions are real. These equilibrium solutions are known as the collinear libration points, as they all lie on the x-axis. The collinear libration points were first identified by Euler in 1765 [66]. Today, by convention L_1 is located between the primaries, L_2 is situated beyond the second primary and L_3 appears to the left of the first primary. See Figure 3.2 for reference.

The remaining two libration points are known as triangular points, and were first discovered by Lagrange in 1772 [66]. They can be found by setting $r_1 = r_2 = 1$ and they appear at the vertex of two equilateral triangles with the primaries, hence the name.

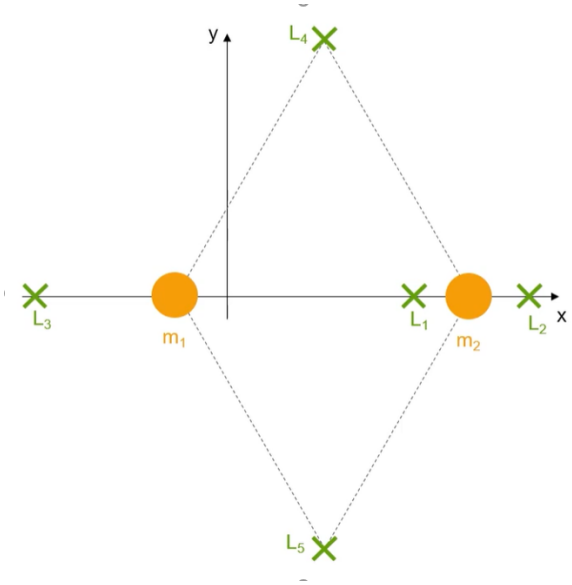

Figure 3.2: Lagrangian Points in a general CR3BP system. [17]

3.1.3. Stability

The stability properties of the Lagrangian points can be analyzed by linearizing the equations of motion around them. This gives place to six eigenvalues, whose real part is of interest. If at least one $Re_\lambda > 0$, the equilibrium solution will be unstable. For L_1, L_2 and L_3, one has one positive eigenvalue, one negative eigenvalue and two pure imaginary conjugate eigenvalue pairs [47]. However, L_3 has slow dynamics and a mild instability [14]. On the other hand, L_4 and L_5 are stable, as all eigenvalues have a negative real part. Figure 3.3 shows the gravitational well around the Lagrangian points, although with inverse signs, hence the peaks correspond to stable equilibria. It can be seen that L_4 and L_5 are at the highest points, hence being stable, contrary to the co-linear points.

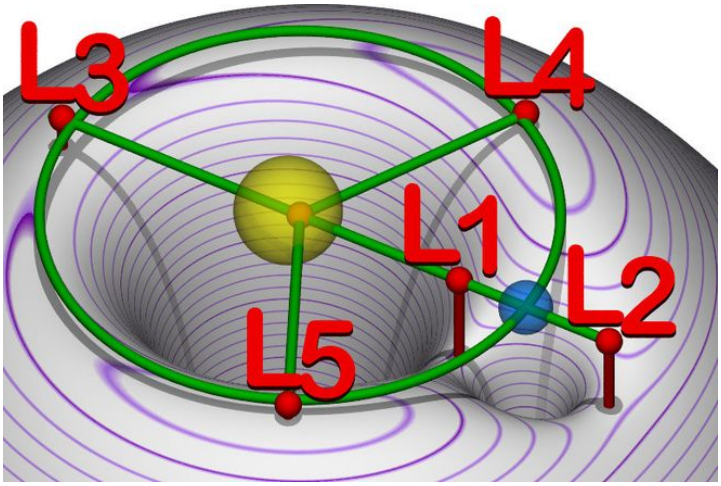

Figure 3.3: Lagrangian points stability well. [73]

Opposite to what it may seem, it is the instability properties that will be key for this study, as they may result in natural, cheaper ways towards or from the equilibrium points, as well as lead to the existence of periodic solutions around them. Hence, the Lagrangian points on which the study will focus are L_1 and L_2. Moreover, these libration points are associated to low energy levels (high values of the Jacobi constant), as Hill curves open earlier for these points, allowing motion around them with less propellant consumption.

3.2. Ephemeris Model

While the assumptions of the CR3BP are precise enough for several applications, such as the general dynamic behavior of a system, actual missions require higher-fidelity models to better simulate the trajectory dynamics. Although such level of accuracy is out of the scope of the tool to be developed, it may be interesting to have a more accurate dynamical system to compare the results against and check the accuracy of the CR3BP model. Therefore, the tool's output is also intended to serve as an appropriate initial guess for higher-fidelity optimization tools, such as the ASTOS software.

3.2.1. Perturbations and Model Accuracy

Reviewing the assumptions made in the CR3BP model, it can be stated that the accuracy limitations come from these main sources:

- **Lunar Eccentricity.** Although a *Circular* Restricted Three-Body Problem is being considered, the Moon's orbit is actually slightly eccentric (e = 0.05490 [54]). This can cause significant differences in the results and justifies the use of other models such as the ER3BP.

- **External Bodies Gravitational Influence.** The Sun is the main perturbing body in this section. Depending on the application and time of flight, the gravitational forces of other bodies, like Jupiter or Mars, may affect the trajectory results considerably.

- **Solar Radiation Pressure.** Especially depending on the design of the spacecraft, the radiation pressure from the Sun can have a considerable effect and deviate the spacecraft from its determined course.

- **Gravitational Irregularities.** The gravity field of the primaries may also have an effect if the designed orbits are close to any of the primaries for a long period of time. This is especially the case for Earth orbits and LLOs.

- **Atmospheric Drag.** Due to the lack of an atmosphere around the Moon, this consideration is only necessary for LEOs. Hence, it will not be important in most of the applications.

The effect of the main perturbing forces has been studied in Pavlak et al. [49] by analyzing the Z-amplitude difference between the CR3BP solution and the higher-fidelity models on a 200 day propagation of the ARTEMIS P2 mission trajectory [74]. Figure 3.4 represents the difference between the nominal and perturbed CR3BP trajectories and the same propagation with an Earth-Moon point mass model (blue) and with an Earth-Moon-Sun model (red). As can be seen, neglecting the lunar eccentricity affects the position results in the order of 1000 km, whereas the Sun has an accuracy impact in the order of 100 km.

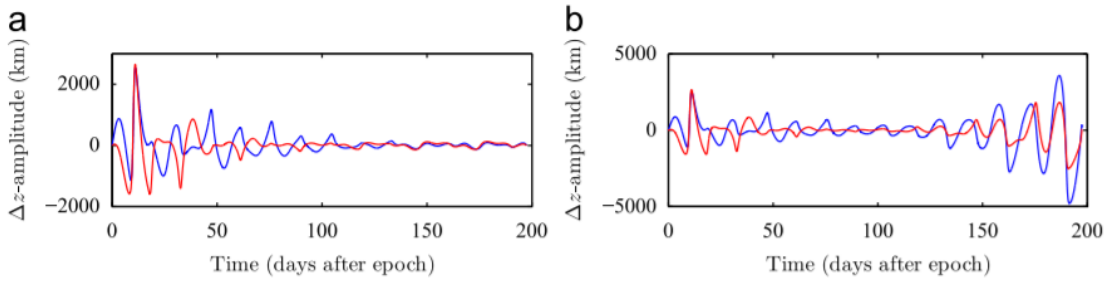

Figure 3.4: Effect of lunar eccentricity (blue) and lunar eccentricity and solar gravity (red) on z-amplitude. (a) Reference and (b) perturbed reference. [49]

The effect of the Solar Radiation Pressure was also studied in [49], although it is considerably less important than the previous two for this application. Figure 3.5 shows the difference between the Earth-Moon-Sun model z-amplitude time history and that of the same model with Solar Radiation Pressure (SRP). As can be seen, the difference is just in the order of kilometers.

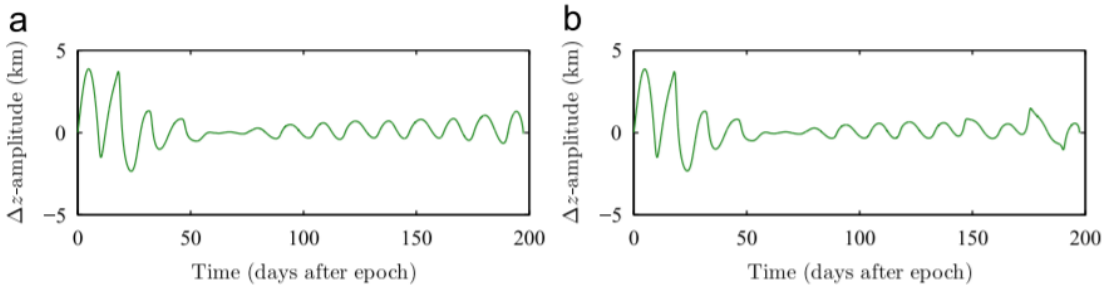

Figure 3.5: Effect of SRP on z-amplitude evolution. (a) Reference trajectory and (b) perturbed trajectory. [49]

3.2.2. Equations of Motion

Most of the inaccuracies can be minimized, as proposed by [48], by using an $N-body$ propagator with the JPL DE405 planetary ephemerides values [54], which locate the positions of any celestial body during the simulation. The $N-body$ dynamics are determined by Equation (3.6). Here the motion of the body of interest (i) is represented in an inertial reference frame centered in one of the bodies (q) and subject to the gravitational pull of m_q and external bodies j.

$$r''_{qi} + \frac{\tilde{G}\left(m_i + m_q\right)}{r_{qi}^3} r_{qi} = \tilde{G} \sum_{\substack{j=1 \\ j \neq i,q}}^{n} m_j \left(\frac{r_{ij}}{r_{ij}^3} - \frac{r_{qj}}{r_{qj}^3} \right) + \sum a_{pert} \tag{3.6}$$

The precision of this model is further improved if additional force models, such as solar radiation pressure and gravity harmonics, are incorporated. Thus, these additional forces are considered in the last summation term of Equation (3.6). As this project was performed together with Astos Solutions GmbH, it has been possible to make use of the ASTOS software [13, 70] to obtain this higher-fidelity model to improve the accuracy of the resulting trajectories in a further stage.

3.3. Coordinate Frame Transformation

Transformations between different coordinate frames are often required when computing and visualizing trajectories in multi-body systems. For example, additional insight can be gained by plotting solutions from a CR3BP in an inertial reference frame. Also, to consider the velocity changes between the trajectory phases and compute the ΔV, the states should be expressed in the inertial frame, to avoid accounting for the rotating components. In addition, a CR3BP converged solution does not usually satisfy all the requirements when it is integrated in a higher-fidelity model. In order to produce a satisfactory propagation in the more accurate model a transformation from the barycenter rotating frame to an inertial J2000 frame is required.

Provided that the differential equations that govern the problem are expressed in the barycentric rotating frame, the transformation from this frame to the inertial coordinate system is detailed below. The opposite transformation can also be computed by inverting the steps and rotation matrices.

3.3.1. From Barycentric Rotating Frame to J200 Inertial System

The first step when transforming a six-dimensional state from the barycentric rotating frame to an inertial J2000 model is to translate the position coordinates from the barycenter to the first primary, in this case the Earth. As the first primary is fixed in the $x-axis$ at the coordinate $x=-\mu$, the transformation is trivial for the position components, as can be seen in Equation (3.7) where the subscript $_{PC}$ corresponds to primary centered.

$$\mathbf{r}_{PC} = \mathbf{r} + \begin{Bmatrix} \mu \\ 0 \\ 0 \end{Bmatrix} \tag{3.7}$$

For the velocity components, however, the transformation is slightly more complex as the rotating rate of the system must be taken into account. Equation (3.8) shows the procedure employed. In this equation, $\Omega = [0\ 0\ 1]$ is the dimensionless rotation rate of the system, and $\mathbf{v}_{PC}^{B}$ is the velocity of the barycenter with respect to the first primary.

$$\mathbf{v}_{PC} = \mathbf{v} + \mathbf{v}_{PC}^{B} + \Omega \times \mathbf{r} = \mathbf{v} + \Omega \times \mathbf{r}_{PC}^{B} + \Omega \times \mathbf{r} = \mathbf{v} + \Omega \times \mathbf{r}_{PC} = \mathbf{v} + \begin{Bmatrix} -y_{PC} \\ x_{PC} \\ 0 \end{Bmatrix} \tag{3.8}$$

Once the translation has been considered, the rotation of the state must be analyzed. To do so, the orbital elements of the Moon are considered.Equation (3.9) shows the approximations used, obtained from [44]. It is important to note that the quantities are expressed in degrees and that M refers to the mean anomaly, which needs to be transformed to the true anomaly.

$$\begin{gathered} i = 5.1453964 \\ \Omega = 330.393098 - 0.05295376 * T \\ \omega = 78.314065 + 0.16435724 * T \\ M = 131.275374 + 13.06499299 * T \end{gathered} \tag{3.9}$$

In the previous equation T refers to the date when the trajectory takes place and is obtained as seen in Equation (3.10).

$$T = JD - 2454465.5 \tag{3.10}$$

where JD is the Julian date in days.

Once the Keplerian angles have been obtained, a rotation matrix is built by composing different rotations as shown in Equation (3.11). Note that as all ephemeris are defined with respect to the ecliptic, a rotation by the obliquity of the ecliptic (ϕ) has to be added.

$$R = [R_Z(-\omega - \theta) \times R_X(i) \times R_Z(\Omega) \times R_X(\phi)]^T \tag{3.11}$$

Finally, the position and velocity coordinates with respect to the first primary ($\mathbf{r}_{PC}, \mathbf{v}_{PC}$ are multiplied by the rotation matrix and the result is dimensionalized using the characteristic quantities of the problem.

If instead of a single point, a whole trajectory solution were to be transformed, the result should be discretized and these steps should be repeated for each point.

In case that the inertial coordinate system uses a higher-fidelity model, this solution should be re-converged with a single or multiple-shooting strategy using the desired differential equations. This process is out of the scope of this work, but it is important to mention it as this is the method employed by ASTOS to load the trajectory.

3.4. Optimization Methods

In this chapter several optimization strategies that have been used in previous work to tackle the CR3BP trajectory optimization are described. First, a comparison between gradient-based and global search methods is presented, including the advantages and disadvantages of each technique. Then, a brief section is included justifying the need for multi-objective optimization. Finally, the MIDACO software is described, as it will be the main optimizer used in this work.

3.4.1. Gradient-Based Methods

The first group of optimization algorithms to be described is gradient-based methods. As indicated by its name, these algorithms are based on the either the analytical or the numerically approximated derivatives of the objective function to find the optimal combination of variables. The particular use of the gradient, Jacobian or Hessian matrices will depend on the order of the method and the number of design variables.

The main advantage of gradient-based methods is that given a sufficiently accurate initial guess, the algorithm will converge to the global optimum. Nonetheless, this optimum is not a single point in a multi-objective problem, but rather a whole region, so several initial guesses would be required.

Moreover, there are other issues with respect to gradient-based algorithms. First, these methods are not suited to optimize multi-modal functions (those with a large number of local minima) or discontinuous functions. As can be seen in Figure 3.6, this is usually not the case in space trajectory optimization applications. Although the figure represents a two-body model trajectory optimization problem, it can serve as an example for this application.

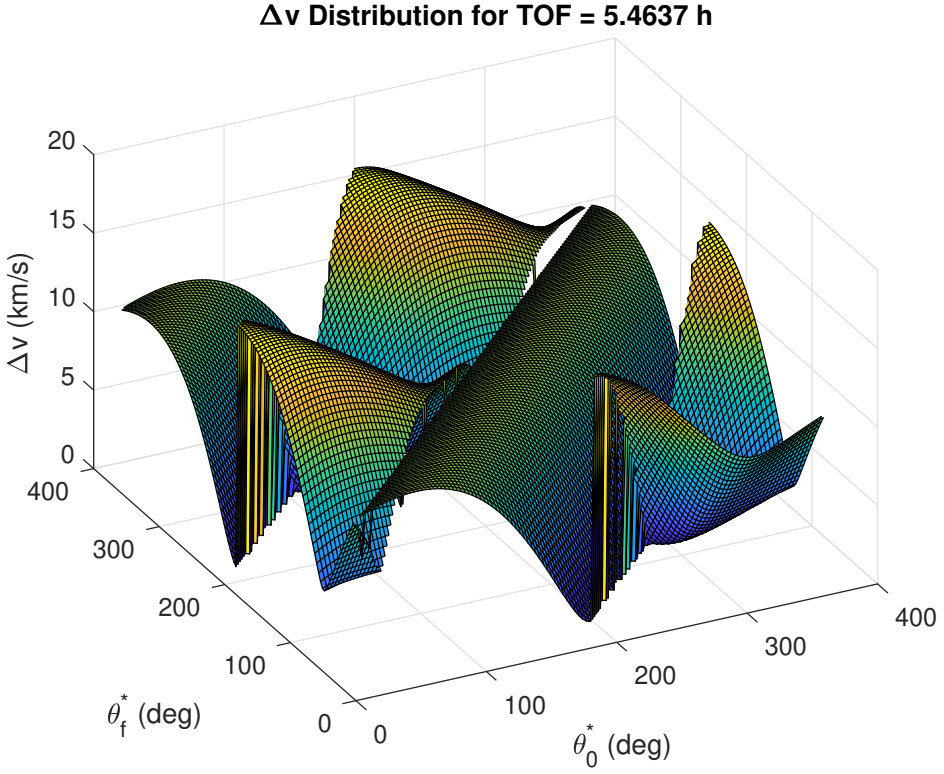

Figure 3.6: Earth-centered two-body transfer design space. [67]

Moreover, gradient-based algorithms would require a good initial guess in order to converge to the global optimum and not get stuck in any of the various local minima. However, this is far from ideal as the tool is intended to be designed such that the user can obtain a range of optimal trajectories without any prior knowledge about the solution. Also, the fact that a range of solutions is delivered is not compatible with the existence of only one initial guess, but rather several would be required for each solution in the Pareto front.

3.4.2. Global Search Algorithms

On the other hand there is the family of global search algorithms. Instead of finding the optimal solution by making use of derivatives, these methods try different variable combinations and refine their search space in the successive iterations. Often, they mimic nature and its behavior, such as ant communication based on pheromones (ant colony optimizer), Darwinism (genetic algorithm), or bird movement (particle swarm optimization), among multiple other options. In this way, their search becomes more efficient than a Monte Carlo simulation or grid search.

For optimizing trajectories in the CR3BP, these algorithms have been used many times [1, 4, 26, 35, 40, 50], as they are more advantageous than gradient-based methods. In the first place, they can easily work with discontinuities, multi-modal functions and non-linearities, so they are not affected by the characteristics of the design space or the objective function. In addition, although an initial guess is sometimes required by some methods, the algorithm will still look in the whole design space, so it is not very determinant.

On the other hand, global search algorithms also have disadvantages. The principal one is that they do not offer a guarantee that the global optimum will be located. It is required to run the optimization with multiple series of quasi-random numbers (imitated by the so=called seed number), to compensate for the random process. Although this increases the confidence on the results, identification of the global optimum is never guaranteed. Moreover, this also translates into a considerable computational time increase when compared to the alternative.

A usual compromise between both methods, that is often used in optimization problems, is to employ the global search algorithm to find a feasible solution expected to be close to the global optimum, and that can be used by a gradient-based method as an initial guess. In principle, the objective here is to build the tool with a global search method so that the solution can be refined in the ASTOS software with a gradient-based algorithm. Refinement with a gradient-based method will not be a part of this work, although it can be studied depending on the time constraints.

3.4.3. Multi-objective Optimization

Multi-objective optimization is a field of study where the design variables result in two or more objective functions. Thus, the optimal solution defines the best trade-off between competing objectives [9], and is composed of a group of optimal solutions, rather than a single point. Equation (3.12) shows a multi-objective problem described mathematically, where **x** represents the design variables, f the objective functions, and h and g the equality and inequality constraints, respectively.

$$\begin{array}{ll} \min / \max \quad f_m(\mathbf{x}), & m = 1, 2, ..., M \\ \text{subject to} \quad g_j(\mathbf{x}) \geq 0, & j = 1, 2, ..., J \\ h_k(\mathbf{x}) = 0, & k = 1, 2, ..., K \\ x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, ..., N \end{array} \tag{3.12}$$

It is important to define another concept with respect to multi-objective optimization: the concept of dominance. A solution will be dominant over another solution if it is strictly better (more optimal) than the other solution. This may not always be the case, as a solution with higher propellant consumption but lower time of flight can be better suited for the mission than another with opposite characteristics. In general, a solution x_1 will dominate over another solution x_2 if x_1 is no worse than x_2 in all objectives and x_1 is strictly better than x_2 in at least one objective [6].

In order to build the Pareto front with all the non-dominated solutions, several options exist. The simpler choices consist of constraining one objective except one and performing a single optimization on the other one. This process is repeated for several values of the constrained objective to build the Pareto front. This approach is known as ε-constraint method. Another option is using a weighted sum of the individual objectives and changing the weights to obtain different solutions. Nevertheless, these

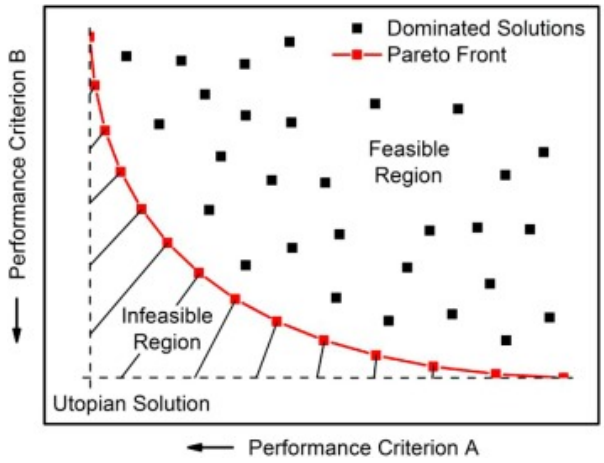

Figure 3.7: Example of a general Pareto front [62]

approaches increase the number of function evaluations considerably, as every optimization returns only one single point on the Pareto front. Hence, to reduce computational time, it is best to consider built-in multi-objective algorithms.

It is also important to mention that multi-objective optimization, although not always considered, is a key concept when developing a trajectory optimization tool, as mission designers take into account considerable objectives when selecting a space transfer. At the very least, propellant consumption and time of flight should be considered, although other objectives such as eclipse time, coverage or number of maneuvers can also be important for some missions.

On the other hand, only few papers appear to provide the full Pareto front when computing optimal trajectories in the CR3BP [1, 24, 68], and even a smaller quantity of them are considering manifold transfers to the gateway orbit. In fact, not a single reference of multi-objective optimization specifically focused on lunar Gateway trajectories was found. This makes the current research a unique contribution to the general knowledge on such transfer.

3.4.4. MIDACO Software

As previously introduced, the MIDACO software [60] will be the optimizer used in this work. MIDACO was developed by Martin Schlueter and is a very capable program that can be applied to problems with continuous and/or discrete variables. It supports all kinds of constraints and can solve optimization problems with thousands of variables and over one hundred objectives.

Its acronym, MIDACO stands for *Mixed Integer Distributed Ant Colony Optimization*, because it solves multi-objective problems with an Ant Colony Optimization (ACO) [58] extended evolutionary algorithm. The ACO algorithm employed by MIDACO is based on multi-kernel Gaussian Probability Density Functions (PDFs), that produce samples of individuals (also called ants or iterates). For integer design variables, a discretized version of the PDF is applied. The constraint-handling inside MIDACO is done by means of the *Oracle Penalty Method* [57] which is an advanced self-adaptive method developed for global search algorithms.

However, the feature that makes this algorithm best suited for this problem is its built-in multi-objective capability. For these problems, MIDACO employs the novel *Utopia Nadir-Balance* concept [61] , which was developed specifically for multi-objective problems that appeared in aerospace applications. This concept is especially suited for problems with four or more objectives, and hence is ideal to improve the tool in the future by adding other objectives such as coverage time and eclipses.

4

Orbits and Periodic Solutions in the CR3BP

An orbit is defined as the "*path of a body revolving around an attracting centre of mass*" [43]. However, this definition only holds for the two-body problem, where there is only one attracting body and the spacecraft follows the Kepler laws and the conic equation. In the CR3BP, however, when switching from an inertial to a barycentric rotating frame, other orbits appear. The spacecraft can orbit not only each of the primaries, but also the Lagrangian points.

In this chapter, the main orbits and periodic solutions that can be found in the CR3BP are described and analyzed. The focus is placed on closed orbits. In particular, this chapter discusses the following main characteristics of each orbit: what do they orbit around, how are their constructed, their periodicity conditions in the CR3BP, and their stability properties. Also, plots for each type of orbit are included to facilitate the properties explanation.

4.1. Keplerian Orbits

As said, Keplerian orbits are those trajectories that arise in the two-body problem, which follow the conic equation. There exist three kinds of conics: ellipses, parabolas and hyperbolas, circular orbits being a special case of elliptical conics. This report will focus mainly on elliptical orbits, as they are closed conics and have considerably more applications. Keplerian orbits are often characterized using the Keplerian elements, as they are physically more intuitive than Cartesian coordinates. This set of variables is able to determine the concrete size, shape and orientation of the orbit in space, as well as the particular position of the spacecraft along the orbit. The list of Keplerian elements is shown in Table 4.1, whereas Figure 4.1 shows a visual representation.

Symbol	Name	Units
a	Semi-major axis	km
e	Eccentricity	-
i	Inclination	°
Ω	Right Ascension of the Ascending Node	°
ω	Argument of Periapsis	°
θ	True Anomaly	°

Table 4.1: Keplerian Elements.

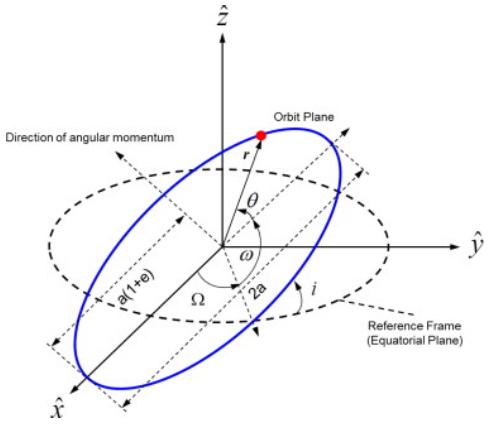

Figure 4.1: Keplerian Elements [64].

Although Keplerian elements offer an intuitive representation of the spacecraft state in space, the information is often needed to be expressed in Cartesian coordinates, especially for propagation and coordinate transformation purposes. Hence, it has become a common engineering problem to establish a transformation between them. Due to the extensive literature on the topic [69], the mathematics involved are not exposed here.

It is important to note, however, that the simple transformation mentioned is not enough, as the CR3BP uses a barycentric rotating frame rather than a inertial, primary-centered one. Therefore, a coordinate transformation, such as the one explained in Section 3.3, must take place. It is important to note that due to the change in reference frame, elliptical orbits do not maintain the same representation in space after each revolution, mainly because the origin is not the primary, but the barycenter of the system, and because the frame is rotating. This does not apply to circular orbits ($e = 0$), such as LEO and GEO because the system rotation does not affect the representation of a circle.

4.1.1. Earth Orbits

Earth orbits are very important as they determine the parking orbit for most mission scenarios. Moreover, they can also be used as the destination orbit for missions that return samples or humans. In this work three main orbits will be studied: Low Earth Orbit (LEO), Geostationary Earth Orbit (GEO), and Geostationary Transfer Orbit (GTO), as they are the most common parking orbits used in space missions. Due to the variety of definitions for these orbits, Table 4.2 summarizes the Keplerian parameters of the orbit that will be used for every case. As can be seen, the GTO is the elliptical orbit connecting the LEO and the GEOs. Note that the orientation angles for LEO and GTO are not fixed, as they can become design variables in order to further improve the solutions. Furthermore, the true anomaly (θ) is omitted as there is no restriction to the position inside the orbit. The choice of the 200 km altitude LEO was made to more accurately compare the results with those of the literature.

	Semi-major axis (a) [km]	**Eccentricity** (e) [-]	**Inclination** (i) [°]	**RAAN** (Ω) [°]	**Argument of Periapsis** (ω) [°]
LEO	6578.1	0.0	0-180	0-360	0-360
GEO	42164.0	0.0	0.0	0.0	0.0
GTO	24367.5	0.73033754	0-180	0-360	0-360

Table 4.2: Earth Orbits Keplerian Parameters.

4.1.2. Moon Orbits

As for the previous section, the smaller primary of the system, in this case the Moon, may also have spacecraft orbiting around itself. These orbits may be of interest as intermediate steps towards lunar surface operations or for closer lunar observation missions. The focus of this work will be on the Low Lunar Orbit (LLO).

LLOs have a maximum altitude of around 300 km above the Moon surface and their orbital period is usually about two hours. These characteristics suit them perfectly for lunar observation missions, due to the close surface proximity. However, gravitational perturbations due to the Moon irregularities make them very unstable, with only a limited number of inclinations available for long-term missions. As they are very close to the Moon, the two-body model assumption holds, and they can be defined by means of Keplerian elements. As explained in the previous section, to build them in the barycentric rotating frame, a coordinate transformation must take place.

As a representation of a lunar landing at the poles, a 100 km altitude, circular and polar ($i = 90°$) LLO will be the objective of this investigation. This choice enables a comparison with previous literature results [72]. Table 4.3 summarizes the Keplerian elements used.

4.2. CR3BP Periodic Solutions

Being able to orbit the Lagrangian points is very beneficial for mission design. First, because it allows to maintain the scientific and observation advantages that Lagrangian points themselves have, without limiting the number of satellites that are present at a Lagrangian point, nor requiring an extreme level

	Semi-major axis (a) [km]	**Eccentricity** (e) [-]	**Inclination** (i) [°]	**RAAN** (Ω) [°]	**Argument of Periapsis** (ω) [°]
LLO	1837.1	0.0	90.0	0.0	0.0

Table 4.3: Lunar Orbits Keplerian Parameters.

of precision in position and velocity both at the arrival and during the maintenance and station-keeping. Secondly, because these orbits may offer unique characteristics that make them more suitable for the mission requirements than the Lagrangian points themselves. And, thirdly, because the existence of periodic orbits around the equilibrium points increases the transfer possibilities when traveling to or from them, as will be explained in Chapter 5.

This section will focus on the most remarkable periodic solution families: Lyapunov and Halo orbits. Their main properties and characteristics, as well as the strategy developed to compute them is presented. For a more thorough description of this and other periodic orbit families around the collinear Lagrangian points, the reader is referred to [18].

4.2.1. Lyapunov Orbits

The simplest example of periodic orbits in the CR3BP are Lyapunov orbits. The main characteristic that defines them is that they are planar, i.e. contained in the plane of the primaries. An example of a Lyapunov orbit family with varying values of the Jacobi constant can be found in Figure 4.2. The most common parameters to describe a planar Lyapunov orbit are A_x, Jacobi constant, period and A_y.

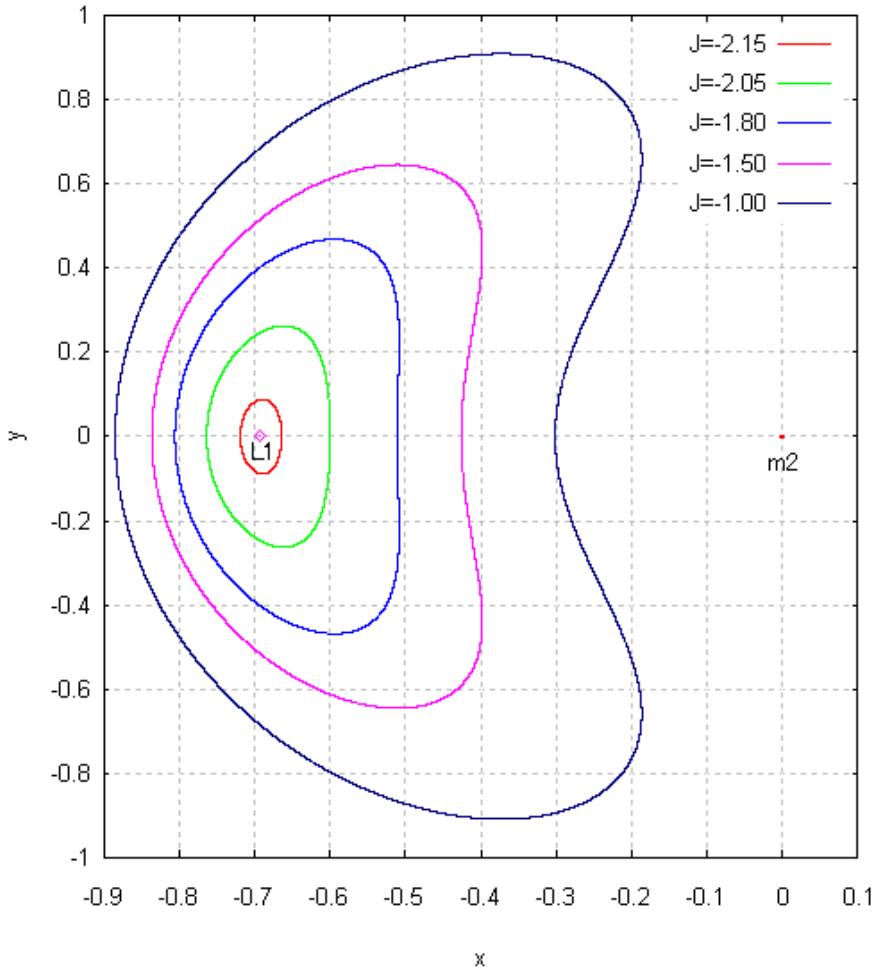

Figure 4.2: Lyapunov orbit around L_1 in the Sun-Earth system. [37]

4.2.2. Halo Orbits

Halo orbits are perhaps the most known periodic solutions in the CR3BP. Discovered by Farquhar in his 1968 thesis [10], their main characteristic that distinguishes these orbits from the planar Lyapunov family, is their out-of-plane motion. When the Z-amplitude (A_z) is greater than a critical value, it can make the in-plane and out-of-plane frequencies match. A_z also becomes a parameter used for the continuation schemes, together with the Lyapunov parameters of A_x, Jacobi constant, period and A_y. The use of other parameters such as periapsis and apoapsis radius, and stability index is discussed in the next section.

Over the last decades, Halo orbits have been the objective of several missions as commented in Chapter 2, due to their unique conditions to satisfy scientific and operational mission requirements. Figure 4.3 represents the Earth-Moon southern L_2 Halo orbit family. As can be seen in the figure, the Halo orbit family bifurcates from the planar Lyapunov orbit into the three-dimensional space. It is important to mention that a symmetrical family of Halo orbits exists, which bifurcates to the positive Z direction and conforms the northern family. This same behavior is repeated for Halo orbits around the L_1 point.

In the following sections, the procedure followed to build an accurate Halo orbit initial guess, obtain the whole family, and get the precise member that fulfills a given requirement, for instance its period, is explained.

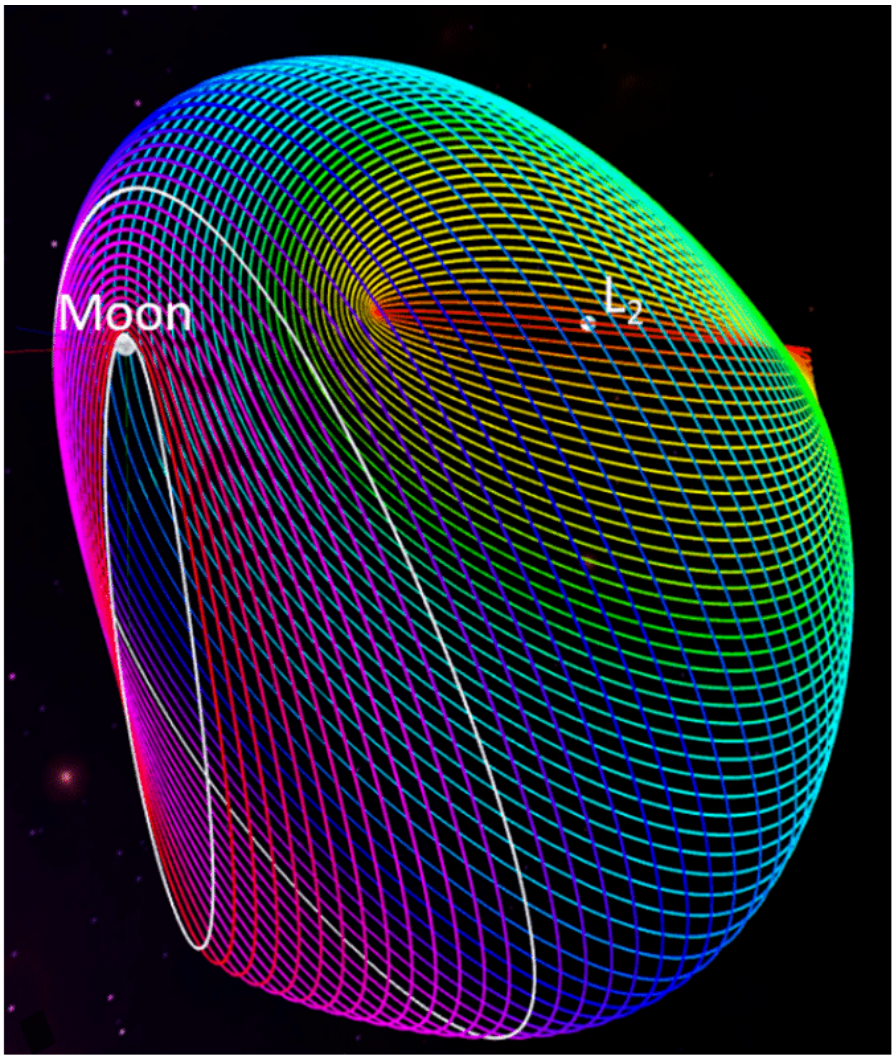

Figure 4.3: Earth-Moon L_2 southern Halo orbit family. [7]

4.2.3. Near-Rectilinear Halo Orbits

A special type of Halo orbits family that has gained special interest over the last years is the Near-Rectilinear Halo Orbit (NRHO) family. It comprises the portion of Halo orbits closer to the smaller primary, as seen in Figure 4.4. An example of NRHOs inside the southern L_2 Halo family can be seen in the leftmost region of Figure 4.3.

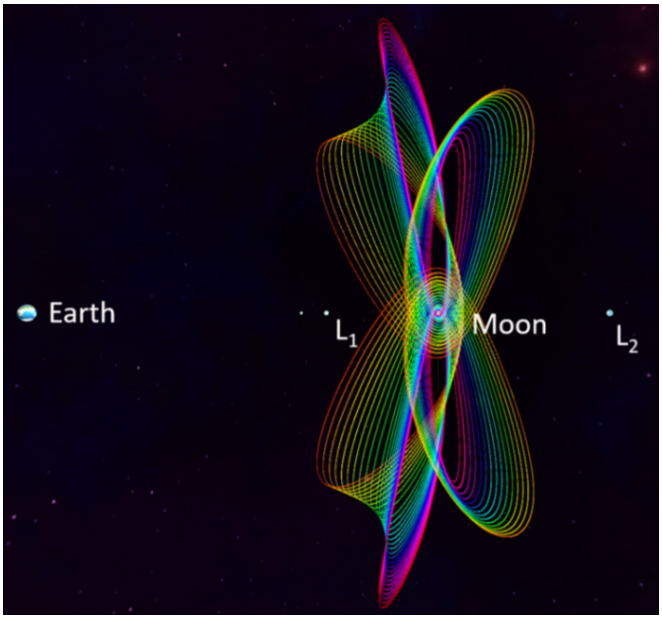

Figure 4.4: L_1 and L_2 Earth-Moon Northern and southern NRHOs. [7]

The boundary between the Halo and NRHO orbits is determined by their stability properties. The stability index ν, represented in Equation (4.1), characterizes this family. As shown by Figure 4.5, low perilune radius have stability indices close to unity, meaning that the periodic orbit is almost stable. On the contrary, nearly-planar Halo orbits can reach stability indices of several hundreds. This higher stability property of NRHOs is very useful for long-term missions, such as the Gateway station which will be built on an L_2 NRHO in 2024 [25] and which is the main target trajectory of this investigation. When building NRHOs, the periapsis radius is another parameter that can be used in continuation schemes. In fact, this investigation has shown that periapsis radius is the only parameter that can fully characterize Halo orbits in general. Stability indices are not recommended as they do not always offer a unique solution, as it can be seen in Figure 4.5. The same occurs with the other parameters.

$$\nu = \frac{1}{2}\left(\lambda_{\max} + \frac{1}{\lambda_{\max}}\right) \tag{4.1}$$

4.2.4. Initial Guess Formulation

In order to obtain a Lyapunov orbit, a single-shooting strategy given a reasonable initial guess can be developed as proposed in [4, 32]. In this case, the initial guess can be the Lagrangian point state with a small perturbation in the x-position and the y-velocity, as the motion will be planar. An initial guess can also be obtained from the literature or from a linear approximation as explained by Richardson [52].

Hence, consider an initial state vector $\vec{x}_0 = [x_0 \;\; 0 \;\; 0 \;\; 0 \;\; \dot{y}_0 \;\; 0]^T$ that lies on the x-axis and whose velocity is perpendicular to the xzZ-plane. According to the *Mirror Theorem* [56], symmetry will be guaranteed by constraining a second perpendicular crossing on the xz-plane. Hence, by using a single-shooting strategy, the trajectory will return to the converged initial condition, producing a periodic orbit. Figure 4.6 shows the initial guess (red) and the converged planar L_2 Lyapunov trajectory.

In order to obtain the perpendicular crossing the free variables are $\Delta\vec{x} = [\Delta x \;\; \Delta\dot{y} \;\; \Delta t]$, whereas the constraints are $\vec{x}_c = [x_c \;\; 0 \;\; 0 \;\; 0 \;\; \dot{y}_c \;\; 0]$, leaving x_c and $\dot{y}_c$ unconstrained. Using a Taylor first-order expansion of the flow of the perturbed state ($\vec{x}_0 + \Delta\vec{x}$) at time $t_0 + T/2$ ($\vec{x}_{T/2}$), where T would be the orbit full period, as shown in Equation (4.2).

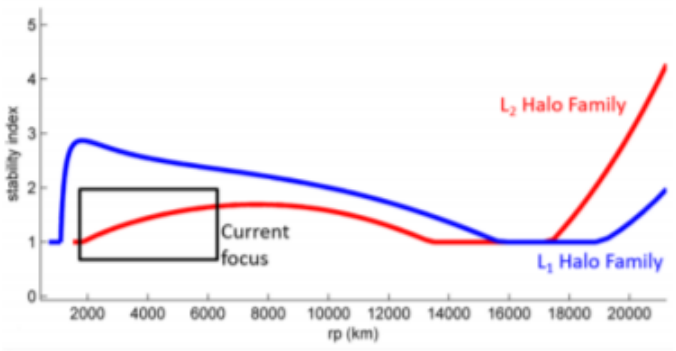

Figure 4.5: L_1 and L_2 Earth-Moon Halo stability indices vs. perilune radius [7]. NRHOs constitute the region with low stability indices.

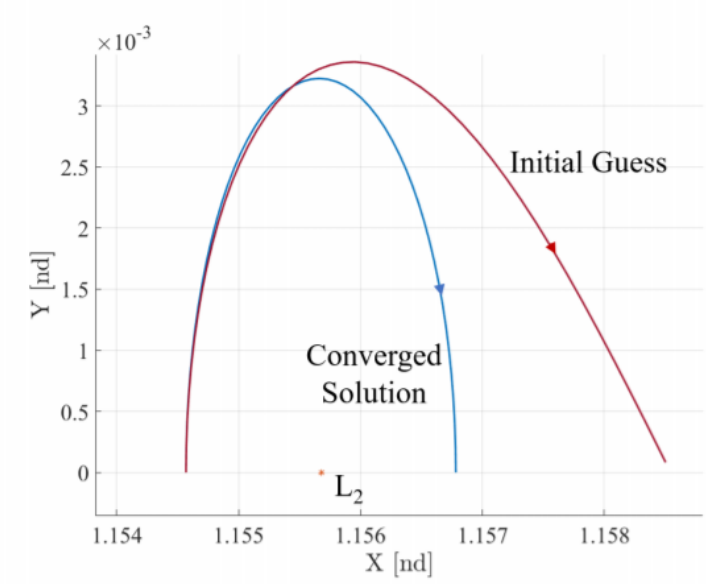

Figure 4.6: Initial guess (red) and converged solution (blue) in a perpendicular crossing differential corrections scheme for an Earth-Moon L_2 Lyapunov orbit [32].

$$\Phi\left(\vec{x}_0 + \Delta\vec{x}, T/2 + \Delta t\right) = \vec{x}_{T/2} + \left[\frac{\partial \vec{x}_{T/2}}{\partial \vec{x}}\right] \cdot \Delta\vec{x} + \frac{\partial \vec{x}_{T/2}}{\partial t} \cdot \Delta t \tag{4.2}$$

Noting that $\left[\frac{\partial \vec{x}_{T/2}}{\partial \vec{x}}\right]$ is the State Transition Matrix (STM), called *Monodromy Matrix* (M) for periodic solutions, and considering the free variables and constraints the system of equations can be reduced to the expression shown in Equation (4.3).

$$M\begin{Bmatrix} \Delta x \\ 0 \\ 0 \\ 0 \\ \Delta\dot{y} \\ 0 \end{Bmatrix} + \mathrm{f}(\Phi)\Delta t = \begin{Bmatrix} x^* \\ 0 \\ 0 \\ 0 \\ \dot{y}^* \\ 0 \end{Bmatrix} - \begin{Bmatrix} x_{T/2} \\ 0 \\ 0 \\ \dot{x}_{T/2} \\ \dot{y}_{T/2} \\ 0 \end{Bmatrix} \tag{4.3}$$

which can be reduced to Equation (4.4), considering that the motion is planar and the constraint values are known.

$$\begin{aligned} m_{21}\Delta x + m_{25}\Delta\dot{y} + f_2\Delta t &= 0 \\ m_{41}\Delta x + m_{45}\Delta\dot{y} + f_4\Delta t &= -\dot{x}_{T/2} \end{aligned} \tag{4.4}$$

As shown in Equation (4.4), there are two equations for three unknowns. Hence, one must parametrize the solution in terms of one of the variables. It is common practice [4] to fix the value of Δx to zero and parametrize in terms of the x-amplitude of the Lyapunov orbit, although other options are possible, depending on the parameter that guides the continuation algorithm.

To compute Halo orbits, the same strategy explained for planar Lyapunov orbits can be used. However, it is noted that due to the out-of-plane motion some differences arise.

First, it is important to distinguish between northern and southern families, which are just mirror images of one another, but whose characteristics change depending on which region of the smaller primary is the periapsis located, for instance. Moreover, it is important to note that having out-of-plane motion translates in one additional unknown Δz, but also one additional equation, as the velocity in the Z-direction is also constrained to zero and there is three-dimensional motion. Hence, the system of Equation (4.4) is rewritten to Equation (4.5). The initial guess is obtained from a planar Lyapunov orbit with a slight out-of-plane perturbation, or from a Richardson approximation [52]. In this work the latter approach was used.

$$\begin{aligned} m_{21}\Delta x + m_{23}\Delta z + m_{25}\Delta\dot{y} + f_2\Delta t &= 0 \\ m_{41}\Delta x + m_{43}\Delta z + m_{45}\Delta\dot{y} + f_4\Delta t &= -\dot{x}_{T/2} \\ m_{61}\Delta x + m_{63}\Delta z + m_{65}\Delta\dot{y} + f_6\Delta t &= -\dot{z}_{T/2} \end{aligned} \tag{4.5}$$

The parametrization, in this case, can be done in terms of x, z, or the period, depending on the parameter that has been chosen to perform the continuation algorithm.

4.2.5. Family Continuation

The idea from the initial guess differential correction process is to obtain only one periodic solution, even though it will probably not fulfill the requirements of the orbit. Then, there are two strategies to compute the whole family of periodic solutions by applying a continuation method. The two most common methods of orbit continuation are natural parameter continuation and pseudo-arclength continuation. The idea behind both methods is to vary the initial guess supplied to the single-shooting algorithm slightly, so that it converges to another periodic solution until the solution whose characteristics are desired is found.

Natural parameter continuation is simpler to implement, however it can yield computational issues when the slope of the chosen parameter p becomes too steep. Hence, smaller variations of p will be needed, increasing the computational time [32].

Moreover, it is important to understand how the different Halo orbit families evolve, as the parameter chosen for the continuation algorithm will depend on the precise family of interest. As seen in Table 4.4, L_1 Halo orbits have a monotonically decreasing z_0 (in absolute value), hence this family should be

computed using a z-continuation scheme. However, Table 4.5 shows that in the L_2 family is x_0 the parameter with monotonic behavior, thus implying that an x-continuation method should be employed here.

JC	Period	x_0	z_0	$\dot{y}_0$
2.7650	2.7159986012	0.8627102723	-0.4702996003	0.1351811306
2.8108	2.6303896376	0.8824867412	-0.4295792682	0.1168235652
2.8526	2.5253862155	0.8994896761	-0.3893500236	0.1017096929
2.8852	2.4186055725	0.9116731376	-0.3558608940	0.0916962219
2.9148	2.2965495319	0.9214943426	-0.3238011029	0.0848204819
2.9384	2.1788225624	0.9279940378	-0.2972096703	0.0819024094
2.9608	2.0492350902	0.9323498107	-0.2711617980	0.0829745365
2.9797	1.9284597334	0.9334331905	-0.2483007402	0.0899286329
2.9987	1.8126475049	0.9271716074	-0.2221003493	0.1154680681
3.0038	1.8551101606	0.9086559874	-0.2047314903	0.1643943779
3.0012	1.9744679048	0.8939591428	-0.1982839590	0.1969078447
2.9989	2.0945289103	0.8827711645	-0.1942766955	0.2187424072

Table 4.4: Initial Conditions for the L_1 Halo Orbits [75]

JC	Period	x_0	z_0	$\dot{y}_0$
3.0591	1.3632096570	1.0110350588	-0.1731500000	-0.0780141199
3.0493	1.4748399512	1.0192741002	-0.1801324242	-0.0971927950
3.0411	1.5872714606	1.0277926091	-0.1858044184	-0.1154896637
3.0341	1.7008482705	1.0362652156	-0.1904417454	-0.1322667493
3.0283	1.8155211042	1.0445681848	-0.1942338538	-0.1473971442
3.0236	1.9311168544	1.0526805665	-0.1972878310	-0.1609628828
3.0199	2.0474562565	1.0606277874	-0.1996480091	-0.1731020372
3.0171	2.1741533495	1.0691059976	-0.2014140887	-0.1847950147
3.0155	2.2915829886	1.0768767277	-0.2022559057	-0.1943508955
3.0152	2.4093619266	1.0846726654	-0.2022295078	-0.2027817501
3.0162	2.5273849254	1.0925906981	-0.2011567058	-0.2101017213
3.0188	2.6455248145	1.1007585320	-0.1987609769	-0.2162644440

Table 4.5: Initial Conditions for the L_2 Halo Orbits [75]

On the contrary, pseudo-arclength continuation uses the nullspace of the Jacobian matrix from the previous periodic solution to build the initial guess in the direction tangent to the slope of the family, hence better adapting to the curve. Figure 4.7 shows how the initial guess from each of the continuation schemes is obtained. The most common parameters to describe a planar Lyapunov orbit are A_x, Jacobi constant, period and A_y. However, its implementation is not so intuitive. Nonetheless, this will be the approach followed.

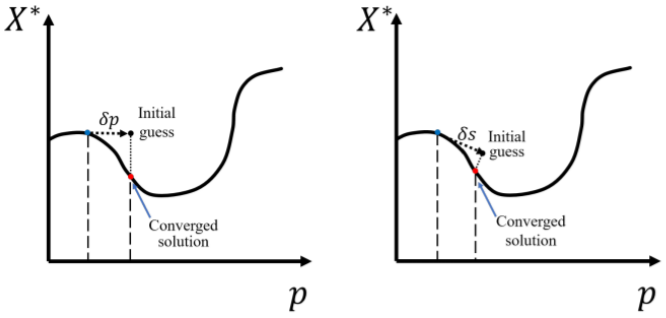

Figure 4.7: (a) Natural parameter continuation and (b) pseudo-arclength continuation schemes. [32]

5

Transfers in the CR3BP

In Chapter 4 the main orbits and periodic solutions that are of interest in the CR3BP, were presented. In this chapter, how to travel from one of these orbits to another one will be discussed, together with the kinds of transfers that are used in the CR3BP, the main characteristics of each of them in terms of flight time and propellant consumption, and the strategies to compute them.

5.1. Direct Transfers

To begin with, direct transfers will be discussed. They are referred as *direct* because the trajectory is performed *directly*, not using any stability feature of the CR3BP. At a minimum, two impulses are required to make a direct transfer: one that departs the initial orbit, and a second one with which the spacecraft is inserted into the destination orbit. Nevertheless, three- or four-impulse transfers are also possible, and may be of interest in some occasions. In this investigation, however, only two-impulse direct transfers will be studied. Figure 5.1 shows an example of a direct transfer profile from an Earth LEO to a Moon L_1 and L_2 Halo orbit in the barycentric rotating reference frame.

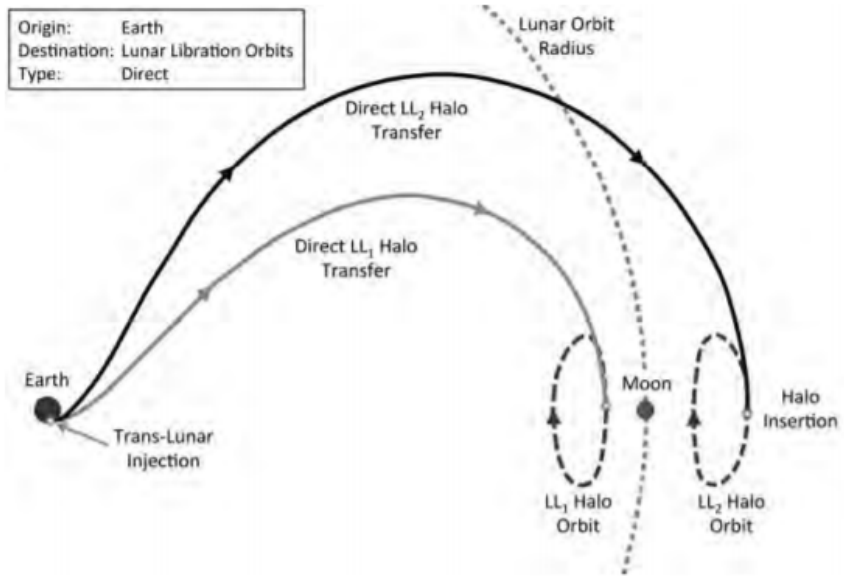

Figure 5.1: Profile for a simple direct transfer from Earth to a lunar libration orbit about either the Earth–Moon L_1 or L_2 point. [46]

Regarding the trajectory objectives, optimal LEO-to-Halo direct transfers are usually characterized by short time of flights (3-10 days) and higher propellant consumption (3600-4100 m/s) than other transfer alternatives [36]. Thus, they are commonly used in crewed missions, such as the Apollo missions in the 1960s space race. Furthermore, the two-body solution can also be used as an initial guess for

the refined CR3BP optimal solution if the optimizer requires one, as there are several similarities between the two models. Unfortunately, this is only the case between Keplerian orbits, as other periodic solutions such as Lyapunov or Halo orbits cannot be obtained from a two-body model.

The main advantage of direct transfers, apart from their low time of flight, is that they can connect any two states in space, as they do not make use of the dynamic features of the CR3BP. Thus, it is common to combine this arc with other methods. This concept will be further studied in the following sections.

5.1.1. Lambert's Problem

To compute direct transfers, the most practical way is by means of Lambert's Problem. With this method, a transfer arc can be built with only the initial and final state as inputs, as well as the time of flight. Although it has an analytic solution in the two-body problem [23], the CR3BP requires an iterative procedure to solve it. The idea described in [4], is to propagate an initial guess $x = [x_0\ y_0\ z_0\ \dot{x}_0\ \dot{y}_0\ \dot{z}_0]$ during the specified time of flight t so that the desired final position is reached. Using the difference between the propagated and desired position, the initial velocity is corrected with a first-order Taylor approximation as shown in Equation (5.1). The process is repeated until the algorithm converges to the desired accuracy.

$$\mathbf{X}^{j+1} = \mathbf{X}^{j} - \mathbf{DF}\left(\mathbf{X}^{j}\right)^{-1} \mathbf{F}\left(\mathbf{X}^{j}\right) \tag{5.1}$$

where $\mathbf{DF}(\mathbf{X})$ is the Jacobian matrix.

In essence, it is a fixed-time single-shooting strategy with three free variables $(\dot{x}_0\ \dot{y}_0\ \dot{z}_0)$ and three constraints $(x_f\ y_f\ z_f)$. However, the CR3BP is known to be a chaotic system in which the sensitivity to initial conditions is high. Experience has shown that the single-shooting strategy has difficulty converging, especially when the trajectory gets close to any of the primaries. Hence, a multiple-shooting method was used.

The idea behind multiple-shooting is to discretize the transfer arc into n patch points, and integrate multiple segments. In this way, each arc is propagated over a shorter time and the linear approximation becomes less sensitive. However, the problem also becomes more complex as constraints need to be applied at the segment intersections. In order to simplify the approach, continuity constraints between all patch points were introduced, such that no intermediate maneuver is required and the trajectory is continuous. The end point maintains the same constraints as the single-shooting case. Figure 5.2 shows the general framework for a multiple-shooting method.

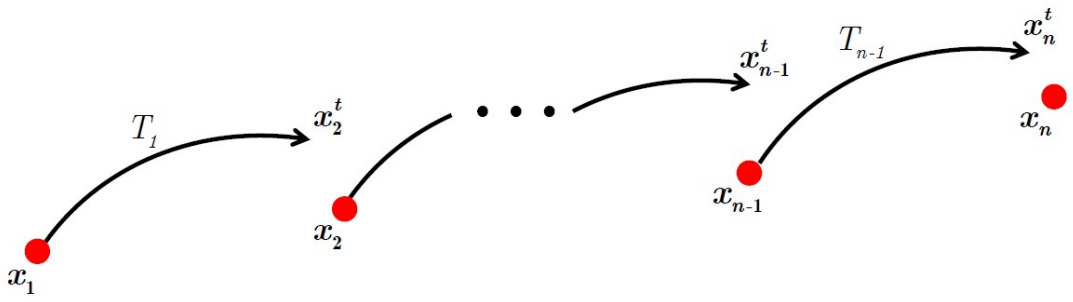

Figure 5.2: General multiple-shooting diagram [48].

In the fixed-time multiple-shooting the free variables are the initial velocity components plus the states on the intermediate patch points which can be stacked in the free variables vector X as shown in Equation (5.2), of dimensions $(6n-3)\times 1$, where n is the number of arcs.

$$\mathbf{X} = \begin{Bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n-1} \end{Bmatrix} \tag{5.2}$$

The constraints are used to ensure the continuity of the trajectory ($\mathbf{x}_i^t = \mathbf{x}_{i+1}$) and the connection of the final state with the required position, forming the $(6n-3)\times 1$ constraint vector **F(X)** that can be seen in Equation (5.3).

$$\mathbf{F(X)} = \begin{Bmatrix} \mathbf{x}_0^t - \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n-2}^t - \mathbf{x}_{n-1} \\ x_n - x_f \\ y_n - y_f \\ z_n - z_f \end{Bmatrix} = \mathbf{0} \tag{5.3}$$

These expressions are used together with the algorithm of Equation (5.1) until the tolerance requirements are met. It is important to note that the Jacobian matrix $\mathbf{DF}(\mathbf{X})$ is often sparse, which offers many numerical advantages when computing its inverse.

5.2. Manifold Transfers

Manifold transfers, apart from the equilibrium solutions and the periodic orbits around them, are one of the most beneficial features that arise after switching from a two-body to a three-body model. Before discussing deeply these transfers, the concept of manifold will be explained in detail.

5.2.1. Definition

Recall from Chapter 3, where the Lagrangian points were presented and their stability properties were defined, that L_4 and L_5 were deemed stable, as the linearization of the equations of motion yielded eigenvalues with only a negative real part, whereas the co-linear libration points were unstable, as they presented at least one eigenvalue with a positive real part. However, the instability of L_3 was mild [14], hence its associated dynamics slow and was left out of further analysis in this work. L_1 and L_2 are the the focus of this project and will be used to explain the concept of manifolds.

As explained in Chapter 4, periodic solutions around Lagrangian points are obtained by perturbing the state in the direction of the eigenvector associated with the pure imaginary eigenvalues, responsible of the periodic motion dynamics. However, perturbing the state in the direction of the other eigenvectors, those corresponding to eigenvalues with negative and positive real parts, will result in stable and unstable manifolds, respectively. This is shown in Equation (5.4).

$$\mathbf{x}_0^s = \mathbf{x}_0 \pm d\mathbf{v}^s \qquad\qquad \mathbf{x}_0^u = \mathbf{x}_0 \pm d\mathbf{v}^u \tag{5.4}$$

Superscripts ^{s,u} represent the stable and unstable branches and parameter d, the amount of perturbation imposed. $\mathbf{x}_0$ is the initial state (position and velocity) and **v** is the associated eigenvector. As can be seen, the $\pm$ sign indicates that two branches emerge from each point.

By perturbing in the direction of the stable eigenvector and propagating backwards in time, the stable manifold branch is found. If the spacecraft is located in any point of that branch (with the corresponding position and velocity), by means of the dynamics of the system, the Lagrangian point will be reached with almost no arrival maneuver. Conversely, the unstable manifold is obtained when the unstable eigenvector is slightly perturbed. From here, the vicinity of the equilibrium point can be left with virtually no propellant consumption. Figure 5.3 shows the one-dimensional[1] stable and unstable manifolds associated to the L_1 point of the Sun-Earth system after a dimensionless perturbation of 10^{-5} was applied. As it can be seen, the manifolds are closer to the Earth (located at coordinates

[1] Note that one-dimensional does not refer to the spatial coordinates in which the manifold extends, but rather the mathematical entity from which it is formed, a point.

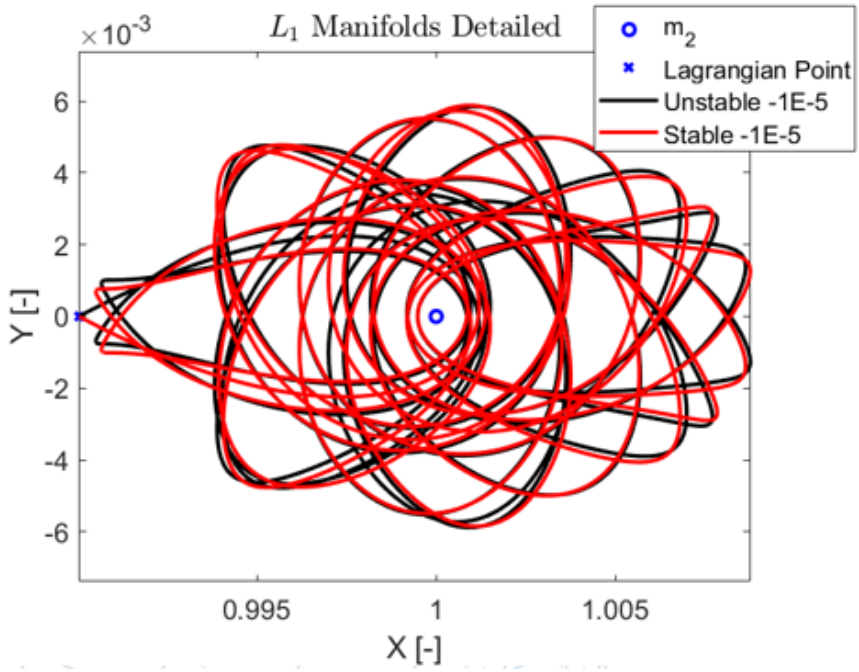

Figure 5.3: 1D Manifolds associated to the L_1 point of the Sun-Earth system after a 10^{-5} perturbation.

[1,0], approximately) than the Lagrangian points. This can reduce time of flight and ΔV consumption considerably.

In the same way as libration points, some periodic solutions such as Lyapunov or Halo orbits also have stable and unstable manifolds. However, in this case they are two-dimensional manifolds as the orbit are mathematically one dimension higher than the equilibrium point. In Figure 5.4, it can be seen how the manifold shape resembles that of a tube, hence it is usual to call orbit manifold, ”manifold tubes”. To obtain one manifold arc, one simply perturbs a particular point of the Halo orbit in the direction of the desired monodromy matrix eigenvector (stable or unstable). The monodromy matrix is the state transition matrix after one revolution. In the next section, the strategies on how to use these manifold structures to compute and integrate efficient transfers will be explained.

5.2.2. Transfers

Depending on the stability properties of the arrival and departure orbits, manifold transfer strategies change considerably. Because they only appear on unstable orbits, one must realize that this kind of transfer will not be possible for two stable orbits, such as two Keplerian orbits.

On the other hand, if the parking orbit were stable and the arrival orbit unstable, the idea would be to connect the departure orbit to one stable manifold of the arrival orbit. This would be the case, for instance, when traveling from the Earth to the Gateway orbit. In the opposite trajectory, from an unstable to a stable orbit, the objective would be to place the spacecraft on an unstable manifold branch that approaches the destination orbit.

Finally, if both orbits were unstable it could be possible to locate the spacecraft on the departure orbit unstable manifold and try to get close not to the destination orbit, but to any stable manifold branch that originates from it. If both orbits have the same Jacobi constant it could be possible to switch from one manifold to the other without any ΔV consumption [45].

Even though it would be ideal that the manifold tube intersected the opposite orbit from which it comes from, this is almost never the case. Therefore, a Lamber-arc direct transfer is usually added to connect the gap between the orbit and the manifold, as shown in Figure 5.5. It is important to note that when adding a connecting arc, ending the manifold closer to the Earth does not necessarily offer any improvement, and this extra constraint may reduce the optimality of the solutions found.

In conclusion, manifold transfers are critically defined by the manifold arc. In order to have a complete definition of the orbit point that is being perturbed, the dimensionless amount of perturbation, and the propagation time of the manifold are the variables required.

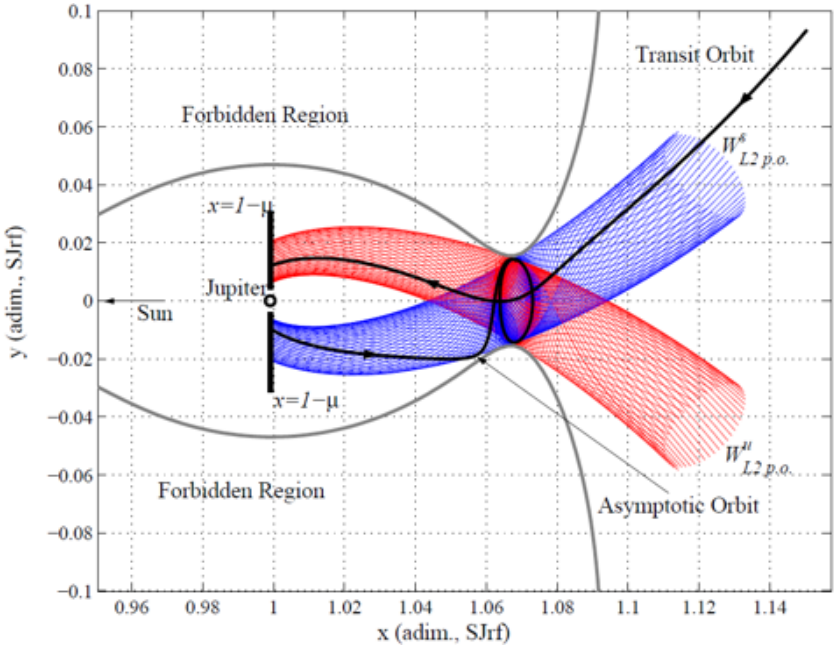

Figure 5.4: 2D Manifold tubes associated to a L_2 Halo orbit in the Sun-Jupiter system. [4]

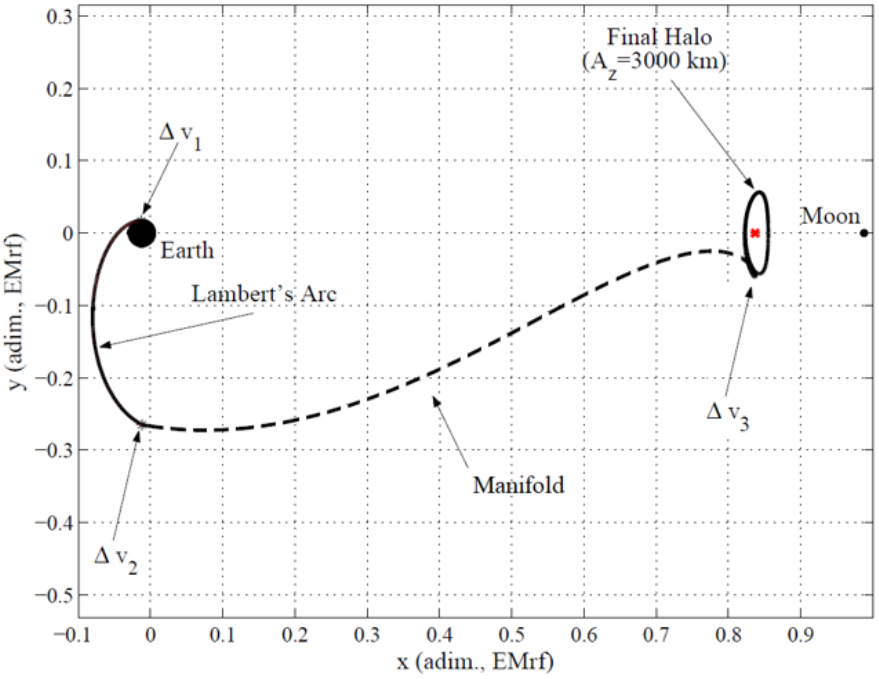

Figure 5.5: Earth-Moon Halo Sample Manifold Trajectory. [4]

However, manifold arcs just guarantee that the last maneuver will be zero, which despite potentially beneficial to minimize the overall ΔV consumption, is a sub-optimal approach. Hence, following previous work [36], a better approach is to consider the manifold arc as an initial guess that can be slightly perturbed, such that even though the final maneuver will become greater than zero, the overall propellant consumption is decreased. Nonetheless, whereas Mingtao et al. [36] perturb the manifold to ensure a flyby passing at the Moon, this work will simply introduce the velocity perturbation components as design variables, without imposing any restrictions on the trajectory transfer.

5.3. Transfers Including Lunar Flybys

Flyby maneuvers are often used in interplanetary travel to reduce the mission ΔV consumption by taking advantage of the gravitational pull of massive bodies. Some authors [36] have found that this concept also provides lower ΔV trajectories in the Earth-Moon CR3BP. Although the manifold approach does not exclude flybys as a possibility, is it often beneficial for global search methods if the search space is reduced around the optimal solutions. Therefore, a different approach will also be studied, in which flybys are included as a requirement.

Mingtao et al. [36] propose an approach where the manifold is perturbed to ensure a flyby. However, this constraints the design space in excess, causing that some potentially optimal solutions are missed by the optimizer. Hence, this investigation will use a different approach by which the flyby point will be set around the Moon using spherical coordinates of radial distance (r), polar angle (θ) and azimuth (ϕ), as represented by Figure 5.6. This point is then connected to the parking orbit and the destination orbit by means of two Lambert arcs.

Although the flyby will be more effective the closer to the Moon the point is chosen, it is important to establish a minimum safety distance from the lunar surface to avoid complications and maintain the gravitational model assumptions as valid. Hence, a limit of 100 km altitude will be set, coinciding with the literature results which will be used for comparison, and arguing that the flyby will be a brief instant, so the possible gravitational harmonics will not affect the transfer considerably. The polar (0-180°) and azimuthal (0-360°) angles are not restricted.

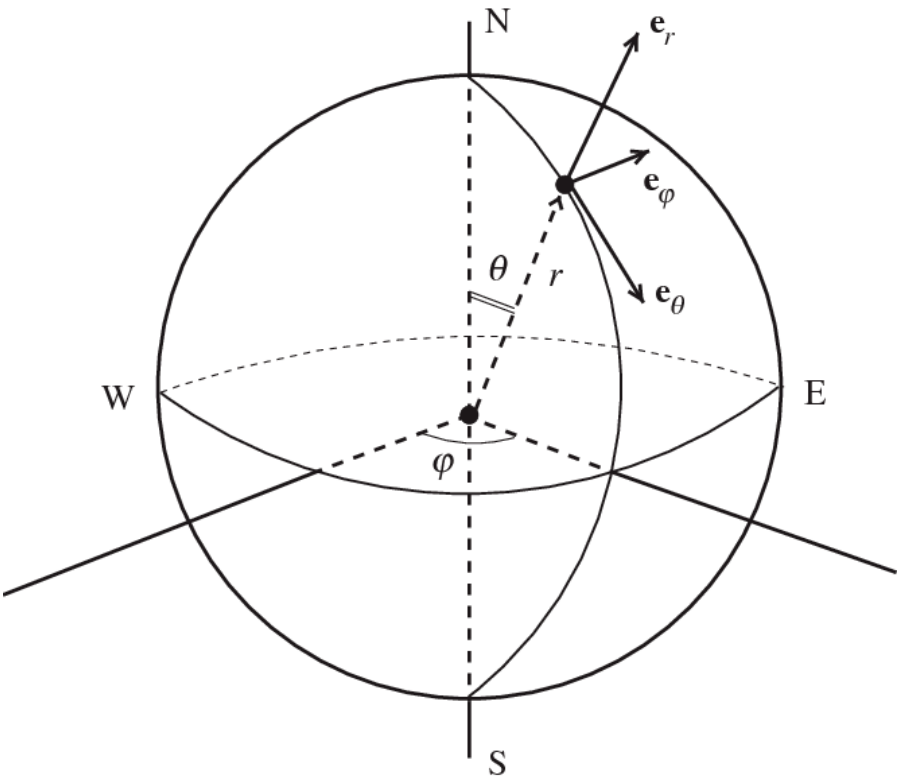

Figure 5.6: Spherical coordinates representation [5].

5.4. Selected Transfers

After the different kinds of transfer arcs have been defined, this section summarizes the possibilities discussed and presents the trajectories that will be studied in the results chapters.

Regarding the transfer possibilities, three main categories should be considered: direct, manifold and flyby transfers. Although direct transfers are available for all trajectories, independently of the parking and destination orbit characteristics, this is not the case for manifold transfers. Flyby transfers are in principle possible regardless of the orbit characteristics, nevertheless, there may be some cases where they do not offer a clear advantage. An example could be a LEO to LLO transfer.

Moreover, as manifold transfers require an additional Lambert arc to connect to the other arcs, it is possible to study direct and manifold transfers simultaneously, by allowing the manifold arc propagation time variable to have a value of zero. In this case, the transfer would effectively become direct. Hence, for the given problems only two optimization runs need to be performed: one for the direct/manifold case, and one for the flyby scenario.

In order to show the capabilities of the developed program, two trajectories will be analyzed. First, a 200 km altitude LEO to an L_2 southern Halo orbit with A_z of 2000 km is chosen. This transfer was selected because there exists previous research [4, 36] to compare the results obtained and discuss whether this work presents any improvement to the body of science. Secondly, a GTO to a 9:2 period resonant NRHO will be studied. The destination orbit coincides with the one proposed for the Gateway concept [25]. Here, the choice was made to show the applications of the tool to problems of great scientific and industrial interest, as this transfer is expected to become very popular in the coming years.

6

Optimization Problem

This chapter describes the general characteristics of the optimization problem. Then, the input files are presented, showing how intuitive is for the user to work with the tool and set up the optimization problem. Finally, the output files are described, which tell the user about the optimization process status and contain final results, including the Pareto front and the trajectory files.

6.1. Objectives and Constraints

As explained in Chapter 3, an optimization problem consists of minimizing a set of objective functions, subject to a series of constraints, by finding the combination of design variables that achieves said optimum. In this case, the objectives consist on the overall ΔV of the trajectory, as a measure of the propellant consumption, and the total time of flight. Although these are the main objectives that appear when dealing with trajectory optimization, other objectives are possible, such as eclipse or coverage time. However, those objectives are out of the scope of this study.

Hence, the aim of the tool is to provide a converged Pareto front containing as many optimal solutions as possible, so as to give the mission analyst several options to decide upon the trajectory that best fits the requirements.

Regarding the constraints, the user only needs to specify the maximum ΔV and TOF desired, which will set the bounds of the Pareto front and limit the design space. There exist other constraints, such as the distance with respect to primaries, which needs to be over a certain threshold to ensure the safety of the mission, but also to maintain the validity of the physical model assumptions.

6.2. Input files

This section summarizes the input files that are employed by the user to declare the trajectory conditions. There are two input files required: the *initialSettings.txt* file, with the run-general settings and the *initialGuess.txt* file, which defines the variables design space.

6.2.1. *initialSettings.txt*

The run settings are defined with the *initialSettings.txt* file, an example of which can be seen in Figure 6.1. The settings can be divided in four sections: general trajectory settings, optimizer parameters, parking orbit definition and destination orbit description.

The first section includes the trajectory string (here *KLMP*) which determines the orbits and arcs that will be used to build the trajectory. In this case, the transfer will start from a Keplerian orbit (*K*), use a Lambert arc (*L*) to join the manifold (*M*) and end at a periodic solution orbit (*P*), particularly a Halo orbit. However other combinations would be possible such as a direct transfer between a Halo and a Lyapunov orbit (*PLP*), for instance. Chapter 7 will explain this in further detail. A name for each arc is added, as well the transfer Julian date, useful for the transformation to inertial coordinates. Finally, the *maxTOF* (in days) and *maxDV* (in km/s) constraints that bound the Pareto front are set.

```
Trajectory      KLMP
Phase names     parkingOrbit, lambert1, manifold, halo
Julian date     2455927.50000
maxTOF  25.0
maxDV   10.0

Seed    823.0
optim   0
maxeval 999999999
maxtime 3000
evalstop        0.0

Semi-major axis 6578.1
Eccentricity    0.0

Orbit Type      HALO-L2-SOUTHERN
Parameter       Az
Value   2000.0
```

Figure 6.1: Initial settings file example.

The second section contains the optimizer settings, which include the random seed number, and the stopping criteria. The *optim* parameter defines whether this will be a multi-objective run (*optim* = 0) or a single-objective refinement (*optim* = 1), the differences of which will be described in the next chapter. Regarding the stopping criteria, the user can choose between a maximum number of function evaluations, a maximum running time (in minutes), or an algorithmic approach which will end the optimization after a certain number of function evaluations do not provide an improved result.

The third section describes the parking orbit, in this case a Keplerian orbit, hence the semi-major axis (in km) and the eccentricity parameters are required. The rest of the Keplerian elements are considered design variables, and are allowed to vary. Nonetheless, they can also be fixed to a desired value by limiting the bounds.

Regarding the destination orbit, which is a periodic solution, the last section defines its characteristics. First, the orbit type is described, including the kind of orbit (*HALO* or *LYAPUNOV*), the Lagrangian point that it orbits (*L1* or *L2*) and, for the Halo orbits, the family (*NORTHERN* or *SOUTHERN*). Then, the parameter used to specify the particular orbit inside the family is declared, as well as its value. The characteristics of both types of orbits will be further explained in Chapter 7.

It is important to mention that all parameters need to be separated from its description by a tabulator.

6.2.2. *initialGuess.txt*

Regarding the description of the design variable space, it is important to note that the variables will change depending on the trajectory that is desired. Therefore, each transfer arc (Keplerian orbits, Lambert arcs, invariant manifolds and periodic solutions) will have a different set of variables associated to them. A summary of the design variables for each arc together with a description, range and the units employed can be seen in Chapter 7.

As the tool purpose is to obtain as many optimal solutions as possible, one might consider leaving the variable bounds as open as possible. However, a wide design space implies the algorithm will have more trouble finding the optimal solutions. Therefore, it is recommended to set the variable bounds as tight as possible without losing any potential optimal solution. Furthermore, not only the lower and upper bounds are required, but also an initial guess must be provided. Although the quality of this guess is not very important for the multi-objective run, it is recommended to use a Pareto-optimal solution in the single-objective refinement run to reduce the convergence time.

6.3. Output Files

Once the user input files have been described, the files used by the optimizer to communicate the optimization state and the final result are presented.

6.3.1. *MIDACO_SCREEN.txt* and *MIDACO_SOLUTION.txt*

These files are generated by MIDACO to convey the state of the optimization process and, for this tool, are updated every 50 iterations. As the printing command introduces some overhead processes that increase the computational cost, this value was chosen so that an update is provided every 2-10 minutes, depending on the application.

Figure 6.2 shows an example of the *SCREEN* and *SOLUTION* files. The former contains a summary of the optimizing parameters that are being used, including the stopping criteria. Moreover, it shows the optimization evolution and can give the user an idea of when the optimization has converged. At the end of the optimization process, the best solution is printed too. The *SOLUTION* file, on the other hand, prints the best solution found by the algorithm as soon as an improvement is made, including the detailed value of the design variables, objectives and constraints. This is an important feature as it gives the user the chance to access the full solution already during run time and adds security in case an optimization run gets interrupted for some reason. This is another novel and useful feature that has not been encountered in any other optimization tool.

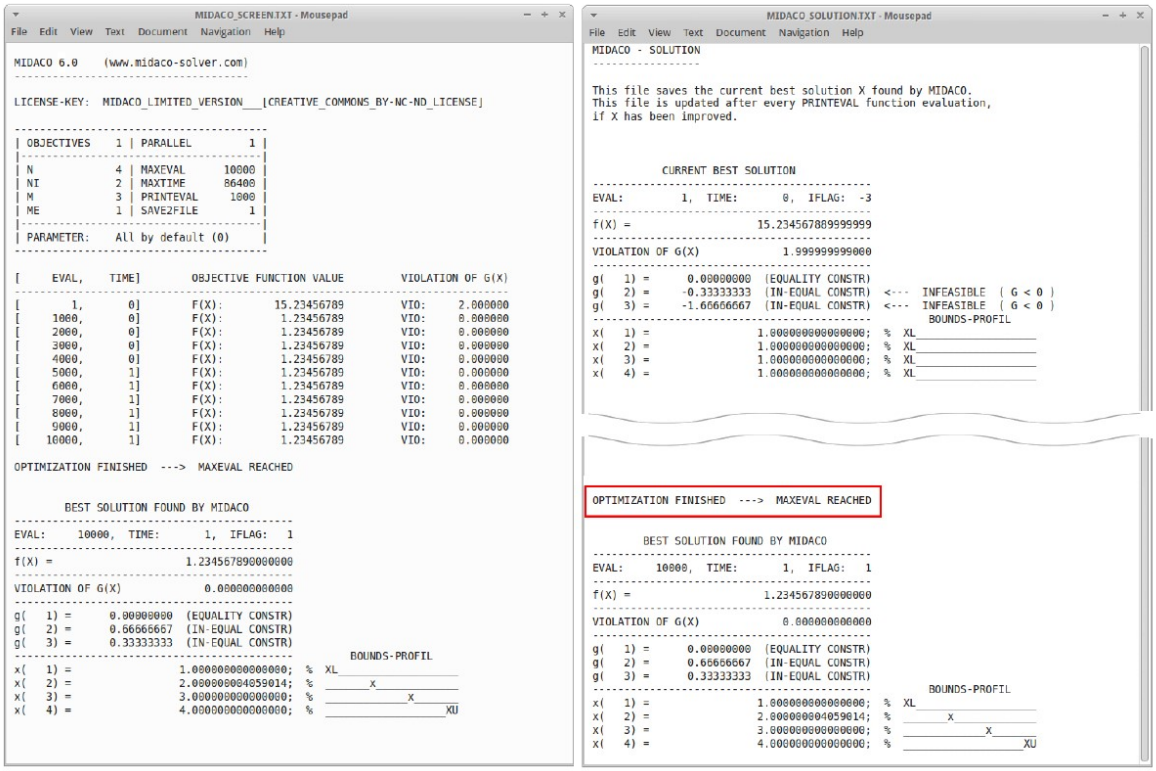

Figure 6.2: MIDACO *SCREEN* and *SOLUTION* file examples [59].

6.3.2. *MIDACO_PARETOFRONT.txt*

The *MIDACO_PARETOFRONT.txt* is the first output file that already gives very valuable information to the user, as it contains the set of non-dominated solutions that conform the optimal result of the multi-objective optimization. In essence, it is the main output of this tool as it will contain all the possibilities from which the mission analyst should choose.

It is also updated every 50 iterations, and in order to provide a more user-friendly visualization of the results, the *PlotTool* executable is included. It is used to graphically illustrate the Pareto front data. For more information on the functionality of this tool, the user is referred to [59].

6.3.3. *MIDACO_HISTORY.txt*

The final output file generated by MIDACO is the *HISTORY* file. This document provides a complete history of all function evaluations tried by the search algorithm. This output is very useful for debugging purposes, but also to double check that the Pareto front provided by MIDACO did not miss any optimal

solution. This can be very important, as two different trajectories could present the same ΔV and TOF values.

6.3.4. Trajectory Output Files

Besides the optimization process files generated by MIDACO, once the user has selected a single solution from the Pareto front to perform a more detailed analysis on it, the trajectory files associated with it are created. These files can then be plotted with an external program, or be imported into the ASTOS software for further analysis. The set of output files include:

- **ini_traj.txt:** Consists of the full trajectory expressed in the inertial J2000 such that it can be imported into ASTOS.
- **Variables.xml:** Set of additional variables such as the Julian date or the ΔV components for each phase. It is also required by ASTOS to load the trajectory.
- **Barycentric rotating trajectory files:** Separate files for each trajectory arc which have the name that was set in the *initialSettings.txt* file (see Figure 6.1). These files can be plotted using an external program such as the Matplotlib library.

7

Internal Code Structure

In previous chapters, the individual elements that conform the developed tool have been described in detail. This chapter will explain how all of these elements have been put together in the code and how they interact within each other to perform the trajectory calculation and optimization. Moreover, an in-depth description of the different design variables is presented.

7.1. Programming Architecture

The tool that is being presented here was developed in the Java programming language using an object-oriented model. It is completely functional as a standalone application, although its output is best intended to be use in combination with the ASTOS software. In this section, the main reasons that justify these choices will be presented.

7.1.1. Programming Language

Although there exist numerous programming languages that would have been suitable for this work, such as MATLAB, Python, Java or C++, it was finally decided that Java would be the chosen language. The main reason was the compatibility that this programming language has with the commercial software ASTOS, with which this tool is expected to merge in the future. In fact, the use of this programming language was a requirement from Astos Solutions GmbH, the company that has been supervising this master thesis.

The use of Java has the advantage of being computationally faster than most of the alternatives, especially MATLAB and Python. This is very important in the case of global search optimization algorithms where a multitude of function evaluations is performed. Another advantage of choosing Java has to do with its object-oriented programming capabilities, which will be further commented in the next section.

Regarding the disadvantages, one could mention the low experience and familiarity with Java prior to this work, which was greatly solved during the internship period. Also, the lack of existing orbital mechanics libraries such as Pykep and Pygmo [22], which are present in Python, should be mentioned here.

7.1.2. Programming Paradigm

Another important element to consider regarding the programming aspects of this work is the programming paradigm. In this case, it was decided to use an object-oriented approach in order to increase the flexibility of the tool so that it can be employed to optimize all kinds of trajectories within the CR3BP.

Object-Oriented Programming (OOP) bases the code in ”objects” that share some information or functions. In Java, objects are known as classes. It is important to mention two object types present in this tool: departure/destination objects, and the intermediate arc objects. The idea is that any trajectory will be defined by a departure object, a series of (or one) intermediate arc objects, and a destination object. However, depending on the transfer characteristics, the particular class of each of those objects will differ. Thanks to OOP, the user will only need to select which objects compose the trajectory and the tool will dynamically change the variables and function logic associated with them.

In addition, OOP also results in a much more organized and structured code, making it easier to add new functionalities (for instance, new periodic solutions) or for debugging purposes, as it improves the code readability. In the next sections, the particularities of each object types are explained.

7.2. Departure and Destination Objects

The first group consists of departure and destination objects. The main difference with respect to the next group is that these classes require some kind of information from the user to define them, and this information will be constant during the whole optimization. However, other attributes may be varied by the optimizer. Furthermore, these objects are self sufficient, meaning that they do not require interaction with other objects to be completely defined. The aim of these objects is to return a state in the barycentric rotating frame, so that it can be used by the intermediate arc objects.

The easiest way to understand this concept is to think of a LEO departure: its Keplerian elements will need to be defined by the user, whereas the true anomaly (or other angles) will remain free for the optimizer to find the best trajectory. Additionally, the state coordinates associated with a particular set of Keplerian elements can be completely defined independently of what the rest of the trajectory looks like.

After analyzing the most common transfers in the CR3BP, three different classes were defined: independent states, Keplerian orbits and periodic solutions.

7.2.1. Independent State Class

The simplest way to achieve the purpose would be if the user specified the six coordinates (three position, three velocity) that compose the state as the translation would be straightforward.

Furthermore, a propagation time design variable is added, as it could be the case that the user does not require the first (or last) maneuver to be performed at the precise state, but rather has the spacecraft in a given situation and wants to reach another position from there after some time.

Nonetheless, this would hardly ever be useful for the majority of cases, as the barycentric coordinates are not a physically intuitive way of describing states. Additionally, the simplicity of this class also denies the possibility of having a manifold arc associated with it.

In conclusion, the state class does not seem very useful but it was included for completion purposes, so that every possibility is considered. Tables 7.1 and 7.2 summarize the class attributes.

Symbol	Description	Range	Units
x	Position along the x-axis in the barycentric rotating frame	-	-
y	Position along the y-axis in the barycentric rotating frame	-	-
z	Position along the z-axis in the barycentric rotating frame	-	-
$\dot{x}$	Velocity along the x-axis in the barycentric rotating frame	-	-
$\dot{y}$	Velocity along the y-axis in the barycentric rotating frame	-	-
$\dot{z}$	Velocity along the z-axis in the barycentric rotating frame	-	-

Table 7.1: Independent State Class Fixed Parameters.

Symbol	Description	Range	Units
$TOF_{0,f}$	Propagation time until/since the first/last maneuver	0... max TOF	days

Table 7.2: Independent State Class Variable Parameters.

7.2.2. Keplerian Orbits Class

As introduced in Chapter 4, the second kind of departure and destination objects are Keplerian orbits. These orbits arise from the two-body problem and can be expressed by means of the six Keplerian elements. The first five elements describe a conic solution in space, its size, shape and orientation. The sixth one represents the position of the spacecraft within the conic at a given time.

As this tool is intended to be used in mission design scenarios, it was decided that only the semi-major axis and eccentricity would be set as fixed parameters, together with the selection of the primary

at the center of the orbit. On the other hand, the Keplerian parameters that define the orientation of the orbit are kept as free variables, such that the optimizer can vary them to find better solutions. However, the user can always restrict their bounds not to allow such freedom.

Tables 7.3 and 7.4 summarize the class attributes. It is important to note that, in principle, only elliptical Keplerian orbits are being considered for this work. Also, as these orbits are solutions of the two-body model, they will have no manifold arcs associated to them.

Symbol	Description	Range	Units
m_i	Primary around which the Keplerian orbit is computed	1,2	-
a	Semi-major axis of the Keplerian orbit	0...	km
e	Eccentricity of the Keplerian orbit	0...1	-

Table 7.3: Keplerian Orbit Class Fixed Parameters.

Symbol	Description	Range	Units
i	Inclination of the Keplerian orbit	0...180	°
Ω	Right ascension of the ascending node of the Keplerian orbit	0...360	°
ω	Argument of the periapsis of the Keplerian orbit	0...360	°
θ	True anomaly at which the maneuver takes place	0...360	°

Table 7.4: Keplerian Orbit Class Variable Parameters.

7.2.3. CR3BP Periodic Solutions Class

Finally, there is the case of periodic orbits of the CR3BP. At first glance, it may seem difficult to put together all these orbits in one category, but they do share numerous characteristics.

First, the precise type of periodic orbit must be defined. Therefore, an orbit type variable is created. The orbit variable is a string of the form *LYAPUNOV-LAGRANGIAN* for Lyapunov orbits and *HALO-LAGRANGIAN-FAMILY* for Halo orbits and NRHOs. In this string, *LAGRANGIAN* should be substituted by the Lagrangian point around which the periodic solution orbits (*L1* for L_1 and *L2* for L_2). Furthermore, in the Halo cases, *FAMILY* should be substituted by *NORTHERN* or *SOUTHERN* depending on the desired choice.

Although this concept may seem slightly confusing, the orbit type is expressed in this way because not all periodic solutions need a *FAMILY* value, or even a *LAGRANGIAN* value (for instance DROs). Hence, in an attempt to advance for future versions this representation was chosen.

Once the periodic orbit family has been defined by means of the previous variable, the individual member of the family that satisfies the user requirements needs to be selected. In order to define these characteristics a periodic orbit calculator tool was created, as explained in Chapter 4. Now, the user will first input the parameter name that will be used to define the family member on a string variable, and then will introduce the desired parameter value in a variable of type double. The relation between the variables input name and value units can be found in Table 7.5.

Input	Description	Units
Az*	Out-of-plane orbit amplitude	km
Ax	Orbit amplitude in the x-direction	km
Ay	Orbit amplitude in the y-direction	km
Period	Time taken to perform a full revolution	days
Jacobi	Value of the orbit's associated Jacobi constant	-
Stability Index*	Stability index of the orbit (see Chapter 4)	-
Periapsis	Minimum distance with respect to smaller primary's center	km
Apoapsis	Maximum distance with respect to smaller primary's center	km

Table 7.5: CR3BP Periodic Orbit Defining Parameters.

* not available for Lyapunov orbits

With respect to free variables, an equivalent to Keplerian orbit's true anomaly was defined. In this case, it is a normalized orbit period value starting from the y-axis crossing with positive velocity and following in the direction of motion. Tables 7.6 and 7.7 summarize the attributes listed.

Symbol	Description	Range	Units
-	Orbit type description	-	-
-	Periodic orbit defining parameter name	Table 7.5	-
-	Periodic orbit defining parameter value	-	Table 7.5

Table 7.6: CR3BP Periodic Solutions Class Fixed Parameters.

Symbol	Description	Range	Units
p	Periodic orbit point at which the maneuver takes place	0.0…1.0	-

Table 7.7: CR3BP Periodic Solutions Class Variable Parameters.

7.3. Intermediate Arc Objects

The second group of objects worth discussing are intermediate arc objects. The members of this group have no fixed attributes as they have no user restrictions. Another shared characteristic is that they are not completely independent, as they require other objects' information in order to compute their initial and final state.

Their function is to provide a link between the departure and destination objects, be it with a single arc, direct transfer, or with a series of connecting phases, as is the case with manifold or flyby transfers. Here, the main objective is to obtain the initial and final state of each arc, in order to later compute the velocity differences between phases and calculate the total ΔV. In this section, the variables required to define each of this classes will be explained.

7.3.1. Lambert's Arc Class

Lambert arcs behave as linking phases between all the other classes, including other intermediate arcs. Their aim is to ensure that the full trajectory is continuous from the position viewpoint. In order to achieve this, they require that all the other objects have been completely defined in previous steps.

This is because they make use of the final and initial state information of surrounding objects and apply a fixed-time multiple-shooting method to fulfill continuity, as described in Chapter 5. The only variable required for this is the arc's time of flight. Although technically this variable could get as high as the maximum TOF, it is generally more convenient to use lower bounds, as Lambert arcs will only be relevant for direct transfers or for linking intermediate arcs, both situations of considerably shorter duration than a full transfer. In this way, the optimization effort is also reduced. Table 7.8 summarizes the design variables.

Symbol	Description	Range	Units
TOF_L	Lambert's arc time of flight	$0…maxTOF$	days

Table 7.8: Lambert's Arc Class Design Variables.

7.3.2. Manifold Arc Class

As introduced in Chapters 3 and 5, manifolds are three-body dynamic features that are present in unstable periodic solutions of the CR3BP. As such, they can only be present when the departure and/or destination objects consist of a periodic orbit solution. The choice between stable and unstable manifolds is performed automatically by the program by looking at whether the associated periodic orbit corresponds to the destination or departure object, respectively.

To completely define a manifold arc three variables are needed. First, because periodic orbit manifolds are two-dimensional (in the mathematical definition), a single element of the whole manifold tube needs to be chosen. This is achieved by selecting the periodic orbit point which will be perturbed to obtain the manifold. This orbit point, however, will be taken from the periodic orbit design variables (see Table 7.7), hence reducing one variable.

Secondly, the dimensionless perturbation magnitude to be applied to this point is requested. This perturbation should be small, to satisfy the linear approximations. Therefore, values between 10^{-2} and 10^{-5} are usually selected. Due to the exponential distribution of this variable, the $\log_{10}$ values of the perturbation are chosen as the actual design variables, rather than the perturbation magnitude itself. Finally, the last design variable corresponding to this class is the propagation time of the arc, expressed in days. Remember that this propagation will be backwards for stable orbits. However, as explained in Chapter 5, the approach chosen will not use ”pure” manifolds, but rather a perturbation to the velocity components will be added to minimize the overall ΔV, not just the last maneuver. Hence, three more design variables are added, one for each velocity component.

Table 7.9 summarizes the design variables of the manifold arc.

Symbol	Description	Range	Units
Δ	$\log_{10}$ value of the manifold dimensionless perturbation magnitude	$-5.0 \ldots -2.0$	-
Δv_x	Manifold velocity perturbation in the X-axis	$-\Delta v_x \ldots +\Delta v_x$	m/s
Δv_y	Manifold velocity perturbation in the Y-axis	$-\Delta v_y \ldots +\Delta v_y$	m/s
Δv_z	Manifold velocity perturbation in the Z-axis	$-\Delta v_z \ldots +\Delta v_z$	m/s
TOF_M	Manifold arc time of flight	$0 \ldots maxTOF$	days

Table 7.9: Manifold Arc Class Design Variables.

7.3.3. Flyby Arc Class

The last intermediate arc class type is the flyby transfer arc. As explained in Chapter 5, flybys will be computed by defining a point around the Moon using spherical coordinates (r, θ, ϕ) and connecting it to the parking and destination orbits with a Lambert arc.

The Lambert arc connecting the flyby point to the parking orbit can be declared as a Lambert arc class by itself, but the second connecting transfer will require some extra variables. These are the combination of the flyby point (r, θ, ϕ) and the Lambert arc design variables (TOF). Table 7.10 summarizes the design variables used for this arc class.

Symbol	Description	Range	Units
r	Altitude at which the flyby takes place	$100.0 \ldots 1000.0$	km
θ	Polar angle that defines the inclination of the flyby	$0.0 \ldots 180.0$	°
ϕ	Azimuth angle that defines the flyby point	$0.0 \ldots 360.0$	°
TOF_F	Flyby arc time of flight	$0 \ldots maxTOF$	days

Table 7.10: Flyby Arc Class Design Variables.

As can be seen, depending on the trajectory the design variables will vary. This is all managed internally by the tool in order to increase its versatility and be able to solve almost any trajectory optimization problem in the CR3BP.

The user only needs to define the trajectory string in the *initialSettings.txt* file and set the proper variable bounds inside the *initialGuess.txt* file. An example of the file for a manifold trajectory can be seen in Figure 7.1, whereas Figure 7.2 shows the setting for a flyby transfer.

```
      i       RAAN    omega   theta   TOF_L   TOF_M   Delta   dVx     dVy     dVz     p
x     0.0,    0.0,    0.0,    0.0,    5.0,    10.0,   -3.0,   0.0,    0.0,    0.0,    0.50
xl    0.0,    0.0,    0.0,    0.0,    1.0,    0.01,   -5.0,   -10.0,  -10.0,  -10.0,  0.0
xu    35.0,   360.0,  360.0,  360.0,  10.0,   25.0,   -2.0,   10.0,   10.0,   10.0,   1.0
```

Figure 7.1: *initialGuess.txt* example for a KLMP transfer.

```
        i       RAAN    omega   theta   TOF_L   TOF_F   r       polar   azimuth   p
x       0.0,    0.0,    0.0,    0.0,    5.0,    10.0,   100.0,  90.0,   180.0,    0.50
xl      0.0,    0.0,    0.0,    0.0,    1.0,    0.01,   100.0,  0.0,    0.0,      0.0
xu      35.0,   360.0,  360.0,  360.0,  10.0,   25.0,   1000.0, 180.0,  360.0,    1.0
```

Figure 7.2: *initialGuess.txt* example for a KLFP transfer.

7.4. Program Running Workflow

Now that all objects and design variables have been defined, the program workflow will be explained. A top-down approach will be followed, starting from the full optimization and then diving into the steps that take place on each function evaluation.

7.4.1. Full Optimization Workflow

There are two main actions when performing an optimization with this tool. These actions try to achieve different outcomes, however, they also share some behavior. In the first place, the optimization settings and the fixed departure and destination variables are loaded, together with the initial guess and design variable bounds. Then, the MIDACO object is created with the corresponding configuration and the optimization starts. This avoids repeating the costly process of reading the settings from an external file for every evaluation and improves the efficiency of the optimization.

With respect to the first phase of the optimization, a multi-objective run takes place in order to obtain the Pareto front. The objective in this phase is to obtain the largest number of optimal solutions and populate the Pareto front. To do so, rather than the Pareto front file provided by MIDACO it is interesting to perform a Pareto filter through all the MIDACO History file, to ensure that no optimal solution was lost. Moreover, given the random nature of global search algorithms, it is strongly recommended to perform several runs with different seed numbers, at least three, and then combine the results in a single Pareto front.

Once the number of optimal solutions is satisfactory, the mission designer should decide which optimal solution from those in the Pareto front best suits his/her mission requirements. Then, this solution is placed as the initial guess for the next phase: the single-objective refinement solution. In this phase, it is very important to also modify the maximum TOF value, otherwise the optimizer will potentially find an optimal trajectory in an undesired region of the Pareto front. This is because in the second phase the optimization is performed with a single objective: ΔV.

Once this optimization is converged, which should not take more than a few iterations given the high accuracy of the initial guess provided, several output files are created. These files are compatible with the ASTOS software and must be imported into a prepared scenario such that the trajectory is loaded into ASTOS and further analysis is performed.

A visual representation of this whole process is shown in Figure 7.3.

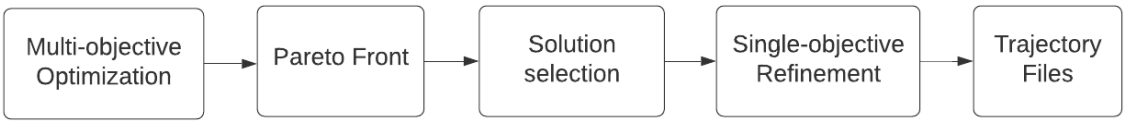

Figure 7.3: Full optimization workflow schematic.

7.4.2. Single Function Evaluation

During any of the optimization runs, several thousands of function evaluations take place with different combinations of variables to try and find the optimal combination. In this section, the steps that take place in every iteration are discussed. In order to show the full capabilities, a *KLMP/KLFP* trajectory has been used.

Having the constant information from the orbits loaded, MIDACO sets the values for the design variables of the current iteration and the tool computes the states associated with the departure and destination objects (*K* and *P*). This is the first step because this information will never depend on any

other object, unlike intermediate arc objects.

The second object to compute its information are manifold objects (*M*) and flybys (*F*), if necessary. This is possible because the periodic orbit object was already created in the previous step, so all the information is available to obtain said arcs.

Finally, Lambert arcs (*L*) are established to connect all previous phases, ensuring continuity within the trajectory. This needs to be the last step as the initial and final state of the Lambert arc must come from the previous phases.

Once all arcs have been computed, each phase initial and final states is transformed to the inertial reference frame so that the velocity difference with respect to the next phase can be computed. Then, the total ΔV is obtained by adding the norm of the velocity changes. The time of flight is obtained by adding the duration of the intermediate arc phases. The departure/destination duration is not counted towards the transfer TOF, but it is used in the output files for plotting purposes.

8

Verification and Validation

8.1. Orbit Calculation

As discussed in Chapter 4, two types of orbits are used in this investigation as departure/destination points: Keplerian orbits and periodic solutions. In this section, the validation and verification for both calculations will be presented.

8.1.1. Keplerian Elements Transformation

The transformation from Keplerian elements to inertial coordinates was not discussed in this investigation, as it is a trivial problem in two-body dynamics and the user was referred to the literature. However, it is important that the code written for this tool is verified. Hence, two sample problems from previous studies were reproduced. The aim in both examples is to transform from the Keplerian elements to inertial coordinates, as this is also the transformation applied in the software.

Input Values	Literature Result	Code Output
a = 6787746.891 [m]	x = -2700816.14 [m]	x = -2700816.139434747 [m]
e = 0.000731104 [-]	y = -3314092.80 [m]	y = -3314092.801019180 [m]
i = 51.68714486 [deg]	z = 5266346.42 [m]	z = 5266346.420678317 [m]
Ω = 127.5486706 [deg]	v_x = 5168.606550 [m/s]	v_x = 5168.606551641139 [m/s]
ω = 74.21987137 [deg]	v_y = -5597.546618 [m/s]	v_y = -5597.546615322460 [m/s]
M = 24.06608426 [deg]	v_z = -868.878445 [m/s]	v_z = -868.878445 19267 [m/s]

Table 8.1: International Space Station on June 12, 2014, 12:00:00 hrs [38].

Input Values	Literature Result	Code Output
a = 7096137.00 [m]	x = 3126974.99 [m]	x = 3126974.985462390 [m]
e = 0.0011219 [-]	y = -6374445.74 [m]	y = -6374445.736050989 [m]
i = 92.0316 [deg]	z = 28673.59 [m]	z = 28673.587563790 [m]
Ω = 296.1384 [deg]	v_x = -254.91197 [m/s]	v_x = -254.911973692548 [m/s]
ω = 120.6878 [deg]	v_y = -83.30107 [m/s]	v_y = -83.301071263987 [m/s]
M = 239.6546 [deg]	v_z = 7485.70674 [m/s]	v_z = 7485.706737958623 [m/s]

Table 8.2: Cryosat-2 on June 13, 2014, 14:59:21 hrs [42].

As can be seen in Tables 8.1 and 8.2, the values obtained by the software are identical to the results offered by the literature to the highest accuracy. This confirms the validation and verification of the Keplerian elements to inertial coordinates transformation. The transformation from the inertial coordinates to the barycentric rotating frame has also been verified by direct comparison with the ASTOS software, which being a commercial software can be used as a reliable source.

8.1.2. Periodic Orbits in the CR3BP

In order to verify the periodic orbit constructor described in Chapter 4, again some examples from literature will be used. In this case, nonetheless, the aim will be to compare the initial state values of several examples of Halo orbits with the parameter values that define them, for instance, the Jacobi constant or the period. To do so, the work by Zimovan [75] will be employed. The comparison between the literature results and the code output can be seen in Table 8.3 for L_1 Halo orbits and in Table 8.4 for solutions around the L_2 point. Please note that the data is in dimensionless values and the last columns represents the absolute difference between both values.

Parameter	Literature Result	Code Output	Difference
JC	3.1743	3.17430085	$8.50{\cdot}10^{-7}$
Period	2.7430553931	2.74303483	$2.05{\cdot}10^{-5}$
x_0	0.8233901862	0.8233904445	$2.58{\cdot}10^{-7}$
z_0	-0.0029876370	-0.002934718497	$5.29{\cdot}10^{-5}$
$\dot{y}_0$	0.1264751431	0.12642521	$4.99{\cdot}10^{-5}$
JC	3.0427	3.0427920527	$9.21{\cdot}10^{-5}$
Period	2.7462016488	2.746222909036	$2.13{\cdot}10^{-5}$
x_0	0.8368126154	0.8367920794209	$2.05{\cdot}10^{-5}$
z_0	-0.1474695518	-0.147394718497	$7.48{\cdot}10^{-5}$
$\dot{y}_0$	0.2560040701	0.2559581128	$4.60{\cdot}10^{-5}$
JC	2.9989	2.9988853496	$1.47{\cdot}10^{-5}$
Period	2.0945289103	2.0945929136074093	$6.40{\cdot}10^{-5}$
x_0	0.8827711645	0.88276570758	$5.46{\cdot}10^{-6}$
z_0	-0.1942766955	-0.194274718497	$1.98{\cdot}10^{-6}$
$\dot{y}_0$	0.2187424072	0.218752435046	$1.00{\cdot}10^{-5}$

Table 8.3: Initial Conditions for the L_1 Halo Orbits [75].

Parameter	Literature Result	Code Output	Difference
JC	3.1521	3.152094574	$5.43{\cdot}10^{-6}$
Period	3.4154433338	3.41548563275	$4.23{\cdot}10^{-5}$
x_0	1.1808881373	1.180893165535	$5.03{\cdot}10^{-6}$
z_0	-0.0032736457	-0.00335250641	$7.89{\cdot}10^{-5}$
$\dot{y}_0$	-0.1559184478	-0.1558883411	$3.01{\cdot}10^{-5}$
JC	3.0807	3.0807912545	$9.13{\cdot}10^{-5}$
Period	3.2266000495	3.22662814977	$2.81{\cdot}10^{-5}$
x_0	1.1542349115	1.154278155535	$4.32{\cdot}10^{-5}$
z_0	-0.1379744940	-0.1378827411675	$9.17{\cdot}10^{-5}$
$\dot{y}_0$	-0.2147411949	-0.21470208965	$3.91{\cdot}10^{-5}$
JC	3.0236	3.0236078555	$7.86{\cdot}10^{-6}$
Period	1.9311168544	1.9311542998	$3.74{\cdot}10^{-5}$
x_0	1.0526805665	1.052683155534	$2.59{\cdot}10^{-6}$
z_0	-0.1972878310	-0.1972886777215	$8.47{\cdot}10^{-7}$
$\dot{y}_0$	-0.1609628828	-0.16096700785	$4.13{\cdot}10^{-6}$

Table 8.4: Initial Conditions for the L_2 Halo Orbits [75].

As can be seen from Tables 8.3 and 8.4, all results are found within an absolute tolerance of $1.0{\cdot}10^{-4}$, which would transfer to the order of 100 km for the position values. Considering that the aim of this tool is to provide a first approximation to the trajectory optimization problem, it can be confirmed that the periodic calculator has been verified and validated for the proposed application. This also validates the physical model differential equations, as it would have not been possible to obtain these results if an error was present in that section of the code.

8.2. Trajectory Calculation

After some intermediate steps of the software have been verified, the whole trajectory optimizer will be validated. In order to do so, the results from Mingtao et al. [36] will be reproduced. This will be achieved by selecting one of their solutions as initial guess together with narrow variable bounds in those variables which are known. However, as the approach differs, some variables are unknown and thus were left with wide bounds.

To also assess the validity of the optimizer, the solution with lowest ΔV was chosen and a single-objective optimization process was started. The algorithm required a few hundreds of iterations before finding similarly enough results.

```
          CURRENT BEST SOLUTION
-------------------------------------------
EVAL:      3700,  TIME:14330.00,  IFLAG:-500
-------------------------------------------
f[0] =                     3.328470118293547
-------------------------------------------
VIOLATION OF G(X)              0.000000000000
-------------------------------------------
g[   0] =      6.67152988  (IN-EQUAL CONSTR)
g[   1] =      3.58826319  (IN-EQUAL CONSTR)
-------------------------------------------          BOUNDS-PROFIL
x[   0] =                25.138057398538653;  # _______________x_____
x[   1] =               318.709104018890170;  # ____________________x__
x[   2] =               216.184740198300400;  # _____________x________
x[   3] =                82.004866605017654;  # ____x_________________
x[   4] =                 5.691736807715883;  # _x____________________
x[   5] =                10.720000000000001;  # XL____________________
x[   6] =                -2.782594521448018;  # _____________x________
x[   7] =                 1.726089697284203;  # ____________________x_
x[   8] =                 2.000000000000000;  # ____________________XU
x[   9] =                 0.591615544738124;  # _____________x________
x[  10] =                 0.367020910124833;  # _______x______________
```

Figure 8.1: Transfer validation result.

Parameter	Literature Results	Software Output	Difference [%]
Total ΔV [m/s]	**3311.43**	**3328.47**	**0.51**
ΔV_1 [m/s]	3116.20	3016.89	3.00
ΔV_2 [m/s]	193.30	305.42	3.39
ΔV_3 [m/s]	1.92	6.32	0.13
Total TOF [days]	**16.43**	**16.41**	**0.12**
TOF_1 [days]	5.70	5.69	0.06
TOF_2 [days]	10.73	10.72	0.06

Table 8.5: Initial Conditions for the L_2 Halo Orbits [75].

The characteristics of this solution can be found in Table 8.5.

The closest solution to the 3.31 km/s result of Mingtao et al. had a total ΔV of 3.33 km/s, which represents less than a 1% difference. When analyzing the individual ΔV values for each phase, differences of up to 3% are found. Regarding the time of flight, differences are negligible as this information was a priori and was included when setting the optimization, albeit with a slight range to account for possible rounding errors. Figure 8.1 shows the *MIDACO_SOLUTION.txt* output from the optimizer with the combination of design variables values that lead to the desired trajectory. It is interesting to note how some variables bounds are active, as if the optimization results could be improved if these bounds were widened. A further study with respect to this aspect will be performed in the next chapter. Please note that the relative differences are performed with respect to the total ΔV and TOF.

Overall, it can be concluded that the tool is able to find the optimal solutions from the literature within a reasonable tolerance limit. In the next chapters, whether the results can be improved will be studied, as well as optimizing other trajectories in order to prove the versatility of the developed tool.

9

Transfer Scenario 1: LEO to L_2 Halo orbit

In the previous chapters, all the details regarding the present investigation have been explained. After the theory has been made clear and the implementation has been validated and verified (see Chapter 8) the tool will be used to solve two different scenarios. This chapter will present the results regarding the optimization of a trajectory from a 200 km altitude LEO to a southern Halo orbit around the L_2 point, characterized by an A_z value of 2000 km. The aim is to be able to compare the solutions obtained to those available in literature [4, 15, 24, 36] and to see if the developed work can improve those results in terms of ΔV or time of flight.

9.1. Trajectory Description

As has been said, the trajectory to be optimized starts at a 200 km altitude LEO. This is a very common starting point for space missions, as launchers are often optimized for this configuration given the quantity of Earth observation missions that are launched every year. It is important to note that the overall ΔV results obtained will depend greatly on the LEO altitude. As such, it is not possible to effectively compare two results which start at different altitudes as the major part of the ΔV is used to escape the Earth's influence, which is greater the closer to Earth the spacecraft is located. Thus, the value of 200 km altitude was selected, as it coincides with several results of literature with which the solution obtained will be compared. Moreover, the LEO has been constrained to be a circular orbit ($e = 0.0$) but the orientation parameters (i, Ω, ω) have been left with open bounds, as has been the case in the other references, as so has the true anomaly (θ).

Regarding the destination orbit, an L_2 southern halo orbit has been chosen, as this is by far the most studied transfer in the CR3BP literature. The particular member of the family was defined with an A_z of 2000 km. This choice was also made looking at the literature, as most authors have found solutions for orbits in the 1000 to 8000 km range. Although the difference can be significant for the mission applications, with respect to the total ΔV and time of flight results are normally within a 5% difference. Having this in mind, the results for closer Halo orbits will be also used to compare the results obtained.

Figure 9.1 shows the *initialSettings.txt* files for the multi-objective run of the manifold transfer (left) and the flyby case (right). As can be seen, the only difference is in the trajectory string and in the name of the phases. However, it is important to note that the random number seed is also changed in each of the optimization runs with each configuration, to analyze the robustness of the optimizer and obtain more certain results.

In the next sections, a multi-objective run will be performed for two different trajectory configurations, resulting in a converged Pareto front with a set of optimal solutions. From those, a result will be selected, a refinement run will be performed and a detail visualization of the trajectory characteristics will be shown, both from ASTOS and from an independent plotting tool that was developed with *Matplotlib* [21].

```
Trajectory      KLMP
Phase names     LEO, lambert, manifold, halo
Julian date     2455927.50000
maxTOF  25.0
maxDV   10.0

Seed    107.0
optim   0
maxeval 999999999
maxtime 3000
evalstop        0.0

Semi-major axis 6578.1
Eccentricity    0.0

Orbit Type      HALO-L2-SOUTHERN
Parameter       Az
Value   2000.0
```

```
Trajectory      KLFP
Phase names     LEO, lambert, flyby, halo
Julian date     2455927.50000
maxTOF  25.0
maxDV   10.0

Seed    107.0
optim   0
maxeval 999999999
maxtime 3000
evalstop        0.0

Semi-major axis 6578.1
Eccentricity    0.0

Orbit Type      HALO-L2-SOUTHERN
Parameter       Az
Value   2000.0
```

Figure 9.1: Case scenario 1 *initialSettings.txt* file for the manifold (left) and the flyby case (right).

9.2. Multiobjective Optimization Results

As explained in Chapter 7, the first step in the optimization process is to perform the multi-objective run in order to obtain the Pareto front with the collection of optimal solutions. To do so, four optimization runs, each with one pseudo-random seed number will be used. The seed numbers used are 107, 468, 569 and 823. This was done to compensate for the random heuristics of the global search algorithm.

Moreover, two different configurations will be tested. First, the manifold transfer will be analyzed, then a trajectory where a lunar flyby is ensure will be tested, and finally the best results of the ten runs will be summarized in a common Pareto front.

9.2.1. Manifold Case

After the initial settings have been defined as seen in Figure 9.1, only the design variable bounds need to be selected before starting the optimization. Figure 9.2 shows this selection for the manifold configuration. As can be seen, the Keplerian angles are left with open bounds as the orbit is not restricted to be planar nor have any other predefined inclination.

The Lambert arc is restricted to a TOF between 1 and 10 days, and the manifold arc can be extended up to 20 days. It also has a lower bound of 0.01 days is set to account for the possibility of a direct transfer. Moreover, the velocity perturbation values are set to $\pm$100 m/s to allow for more maneuvers that can minimize the ΔV for any given TOF.

```
        i       RAAN      omega     theta     TOF_L     TOF_M     Delta     dVx       dVy       dVz       p
x       0.0,    0.0,      0.0,      0.0,      5.0,      10.0,     -3.0,     0.0,       0.0       0.0,     0.5
xl      0.0,    0.0,      0.0,      0.0,      1.0,      0.01,     -6.0,     -100.0,   -100.0,    -100.0,   0.0
xu      35.0,   360.0,    360.0,    360.0,    10.0,     20.0,     -2.0,     100.0,    100.0,    100.0,    1.0
```

Figure 9.2: *initialGuess.txt* file for the case scenario 1 manifold configuration.

The four optimization runs took an average of 10 hours to obtain the converged results that can be seen in Figure 9.3. As can be appreciated, in this time they came up with around a dozen solutions each. It can also be observed the importance of running multiple seed numbers, as the results vary considerably depending on the seed number used, especially for shorter transfers. However, there was also one run (seed number 107) which was unable to reproduce the results from Mingtao et al. [36].

Figure 9.4 shows the combination of the best solutions of each of the seed number runs. It is clear that this approach is an intuitive way of showing multiple results such that the user can easily choose the solution that best fits the mission requirements.

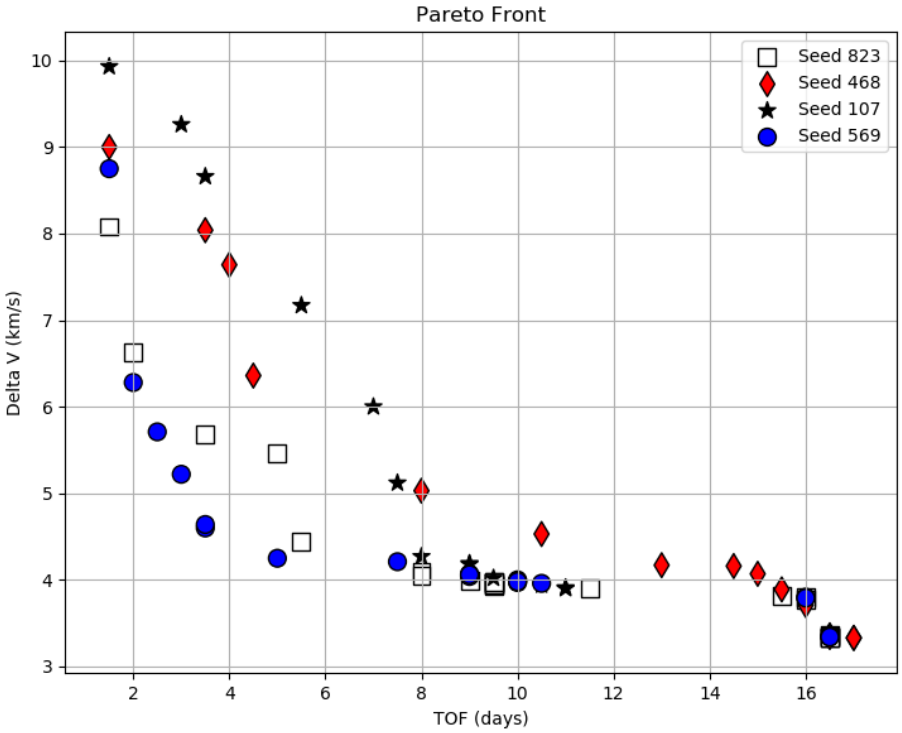

Figure 9.3: Pareto front including all runs for the case scenario 1 manifold configuration.

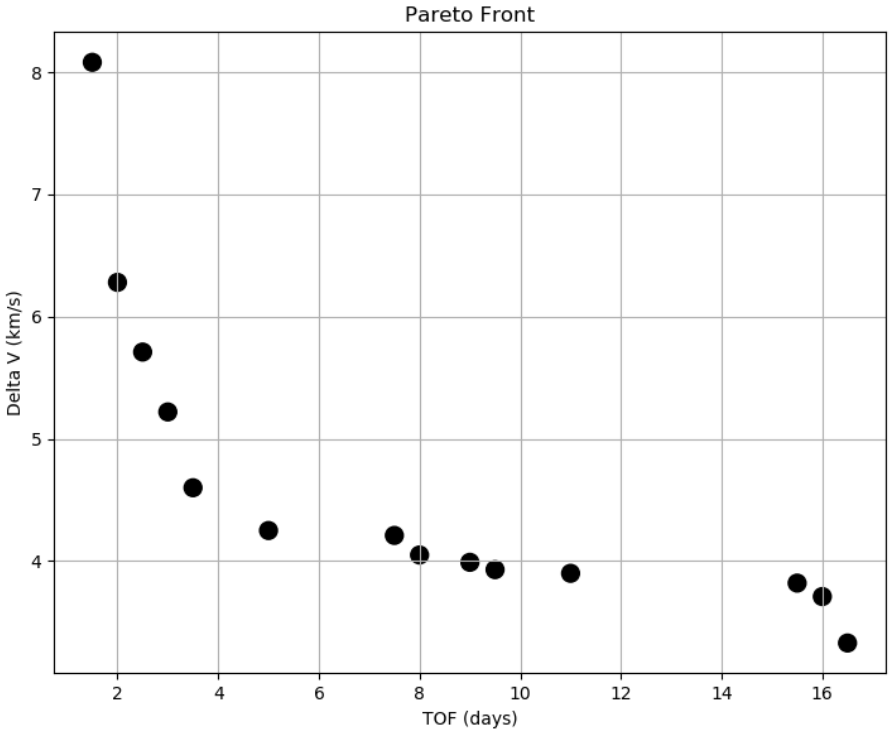

Figure 9.4: Case scenario 1 manifold configuration combined Pareto front.

9.2.2. Flyby Case

With respect to the flyby configuration, slightly different variable bounds were selected. As can be seen in Figure 9.5, the Keplerian elements and Lambert arc maintain exactly the same bounds. For the flyby arc, the time of flight has the same bounds as the manifold arc, but the rest of the variables differ. The altitude is limited to 100 km from the Moon surface, in order to have comparable results to those from literature. The polar and azimuth angles have completely open bounds. This is the same case for the Halo orbit point selected for the insertion.

```
        i       RAAN      omega     theta     TOF_L     TOF_F     r         polar     azimuth   p
x       0.0,    0.0,      0.0,      0.0,      5.0,      10.0,     100.0,    90.0,     0.0,      0.50
xl      0.0,    0.0,      0.0,      0.0,      1.0,      0.01,     100.0,    0.0,      0.0,      0.0
xu      35.0,   360.0,    360.0,    360.0,    10.0,     20.0,     110.0,    180.0,    360.0,    1.0
```

Figure 9.5: *initialGuess.txt* file for the case scenario 1 flyby configuration.

In this case, the four optimization runs took slightly longer computational time to converge, averaging 12 hours. The results from the four runs can be seen in Figure 9.6. Two things can be clearly appreciated. First, the results from the four runs are more similar than in the manifold case. This means that the robustness of this approach is greater than the manifold solution, and hence, the impact of the pseudo-random seed is lower, at least for the trajectory problem analyzed. However there are still notable differences between the runs, and using multiple seed numbers is still recommended. In second place, the results obtained are considerably better than with the manifold approach, as faster transfers with lower ΔV have been obtained. The results from the four runs are combined and filtered, creating a Pareto front which is shown in Figure 9.7.

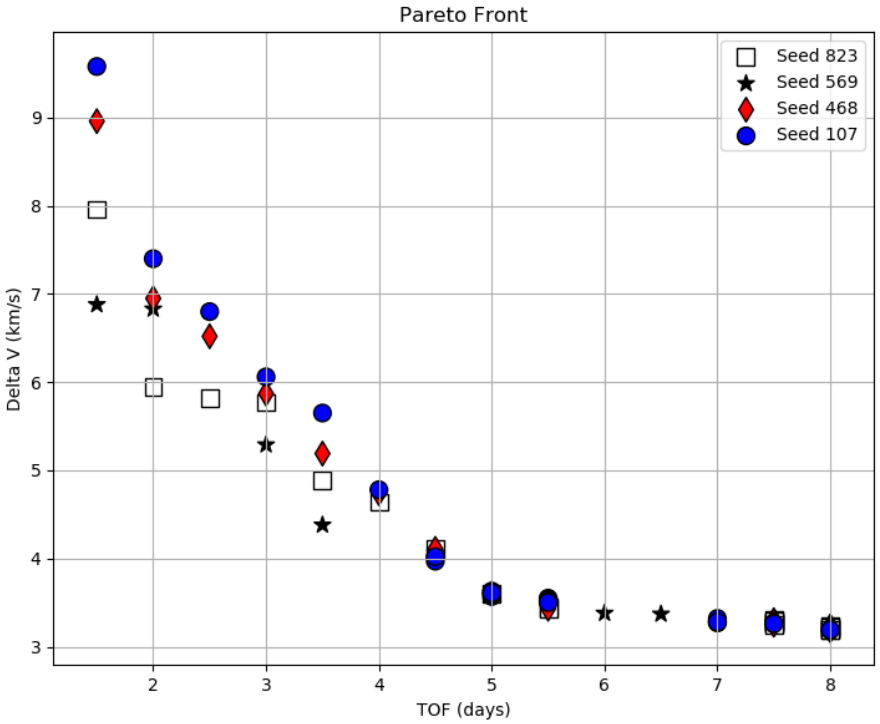

Figure 9.6: Pareto front including all runs for the case scenario 1 flyby configuration.

After combining the results from the manifold and the flyby approach in Figure 9.8, it is clear that the flyby configuration has improved the results of the manifold scenario. Mingtao et al. results have also been added for comparison. The minimum ΔV solution requires 3.26 km/s for the full transfer and only 8 days, versus 3.33 km/s and 16.5 days for the manifold approach. This will be the result used for further analysis, as it is the solution with the lowest ΔV and the time of flight would still be considered low for most applications.

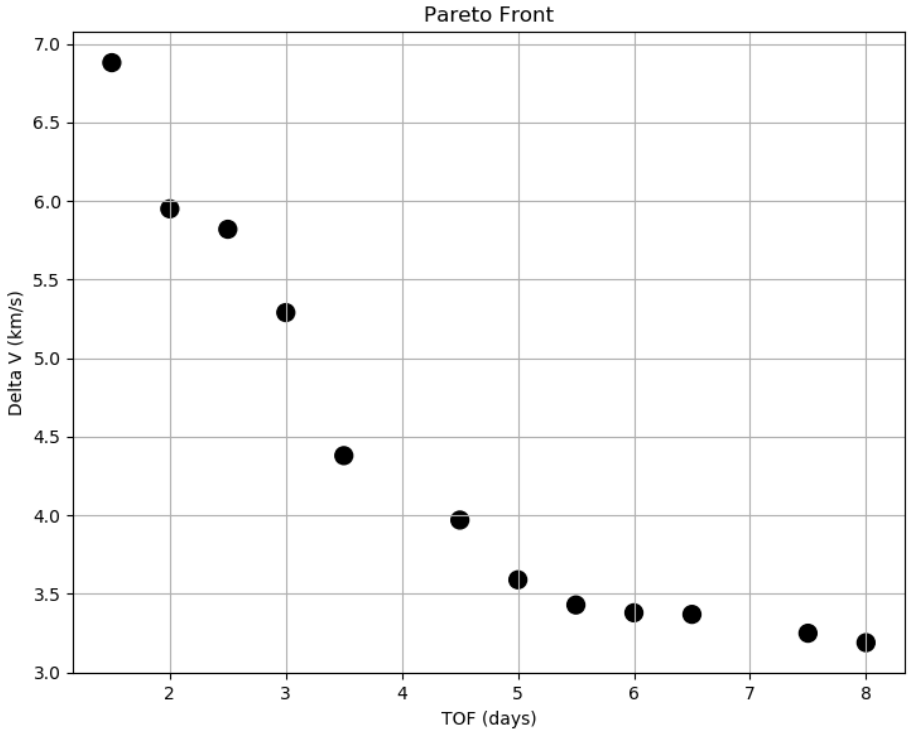

Figure 9.7: Case scenario 1 flyby configuration combined Pareto front.

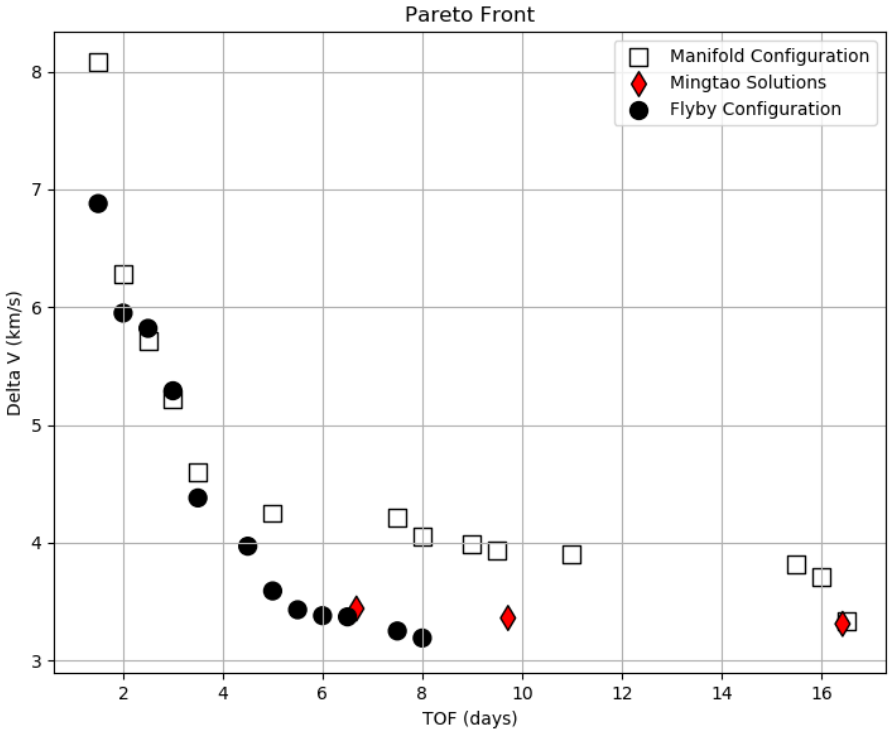

Figure 9.8: Case scenario 1 final Pareto front results

9.3. Refined Solution

For the refined solution, one particular result from the multi-objective optimization is selected as the initial guess. As commented previously, the ΔV = 3.26 km/s and TOF = 8 days is chosen. This is done to increase the convergence speed of the single-objective optimization. The rest of the parameters are maintained unchanged, except for the *optim* value in the *initialSettings.txt* file (see Figure 9.1), which is changed to 1. After another four runs, each with a different random seed number, the single-objective optimizer was able to improve the result found by the multi-objective run by around 2.15% moving from 3.26 km/s to 3.19 km/s. The time of flight was also reduced to 7.75 days.

Table 9.1 shows a compilation of literature results for the selected transfer. As can be seen, the refined solution of 3.19 km/s and 7.75 days improves the majority of prior results, however some exceptions must be discussed. First, Mingtao et al. [36] found a 6.68-day trajectory which is shorter than the presented solution. Nonetheless, the Pareto front in Figure 9.8 shows a 6-day solution which requires only 3.38 km/s in total. Moreover, Zazzera et al. [4] came up with very low ΔV solutions, of 3.15 and 3.12 km/s respectively but the time of flight values were out of the bounds of the design scenario. Overall, it can be affirmed that the results obtained improve those of the literature for this transfer.

Reference	**ΔV Lit.** [km/s]	**TOF Lit.** [days]
Current Work	3.19	7.75
	3.38	6.00
	3.42	5.75
Mingtao [36]	3.31	16.43
	3.36	9.72
	3.44	6.68
Zazzera [4]	3.15	123.1
	3.12	77.4
	3.44	42.8
Gordon [45]	3.44	20
Kokou [24]	3.21	35

Table 9.1: Comparison of obtained solutions with literature results.

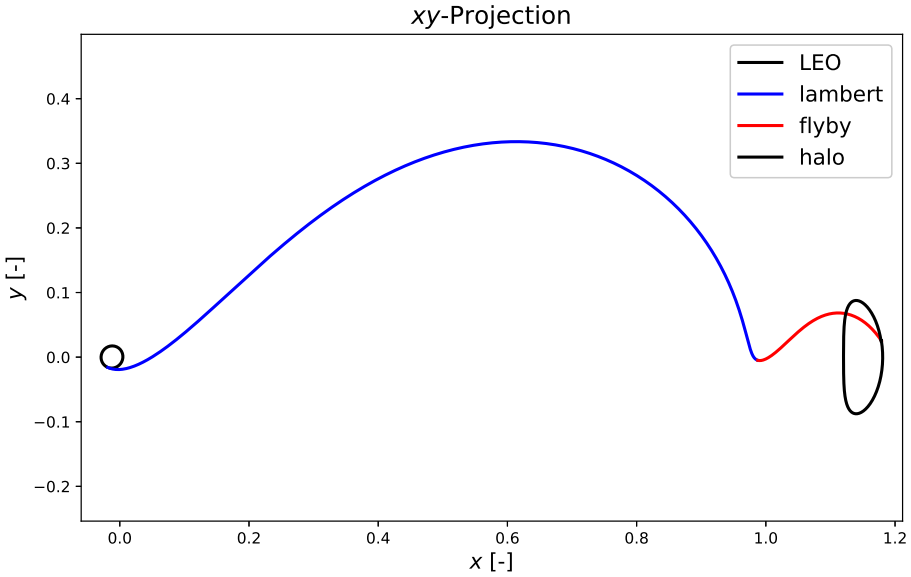

Figure 9.9: Solution trajectory in the barycentric rotating frame XY projection.

This newly optimized trajectory solution is shown in Figures 9.9 to 9.12. Figure 9.9 shows the xy-projection of the transfer in the barycentric rotation frame. This plot has been obtained with an independent tool. This same tool was used to obtain Figure 9.10. It is important to note that the axes in Figure 9.10 are not to scale.

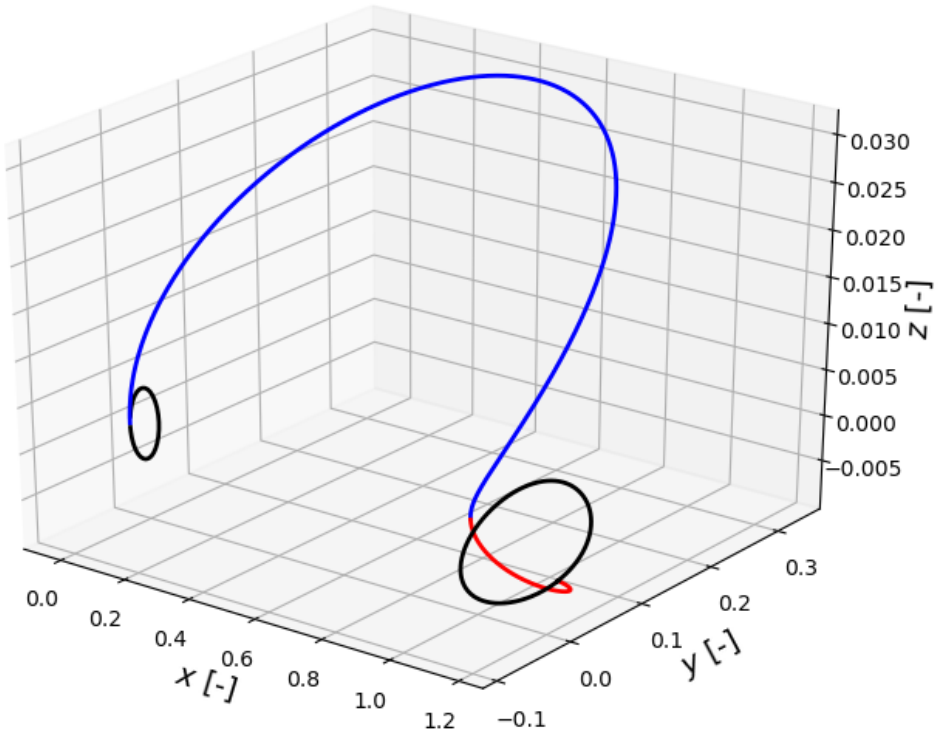

Figure 9.10: Solution trajectory in the barycentric rotating frame 3D projection.

Although the independent tool is useful to obtain a general idea on how the trajectory can be represented in space, exporting the result into ASTOS offers other advantages. As can be seen in Figure 9.11, the same trajectory plots can be obtained.

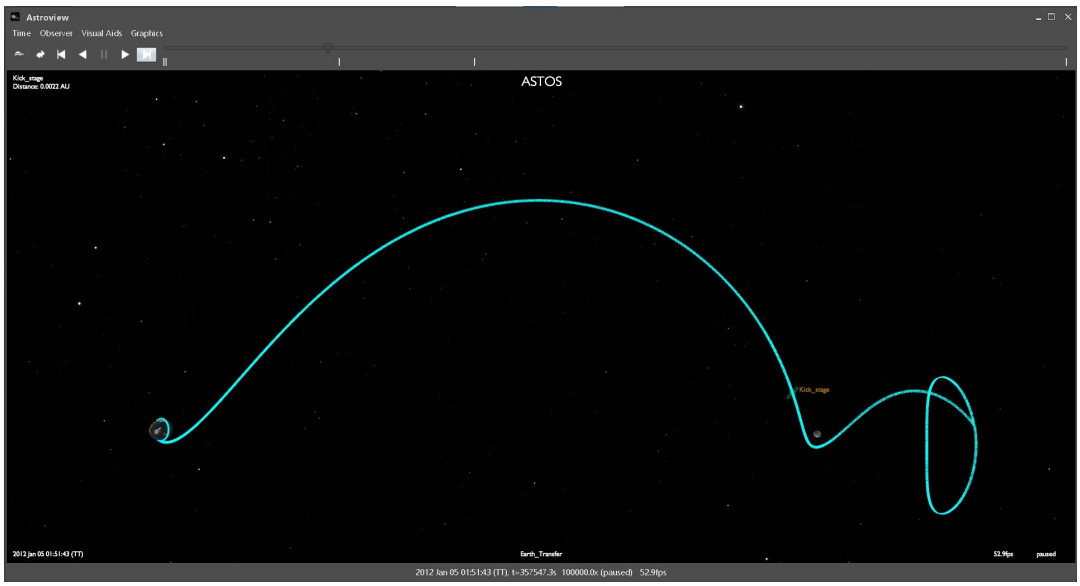

Figure 9.11: Solution trajectory in the barycentric rotating frame as viewed in ASTOS.

However, other interesting results such as the inertial representation of the trajectory seen in Figure 9.12.

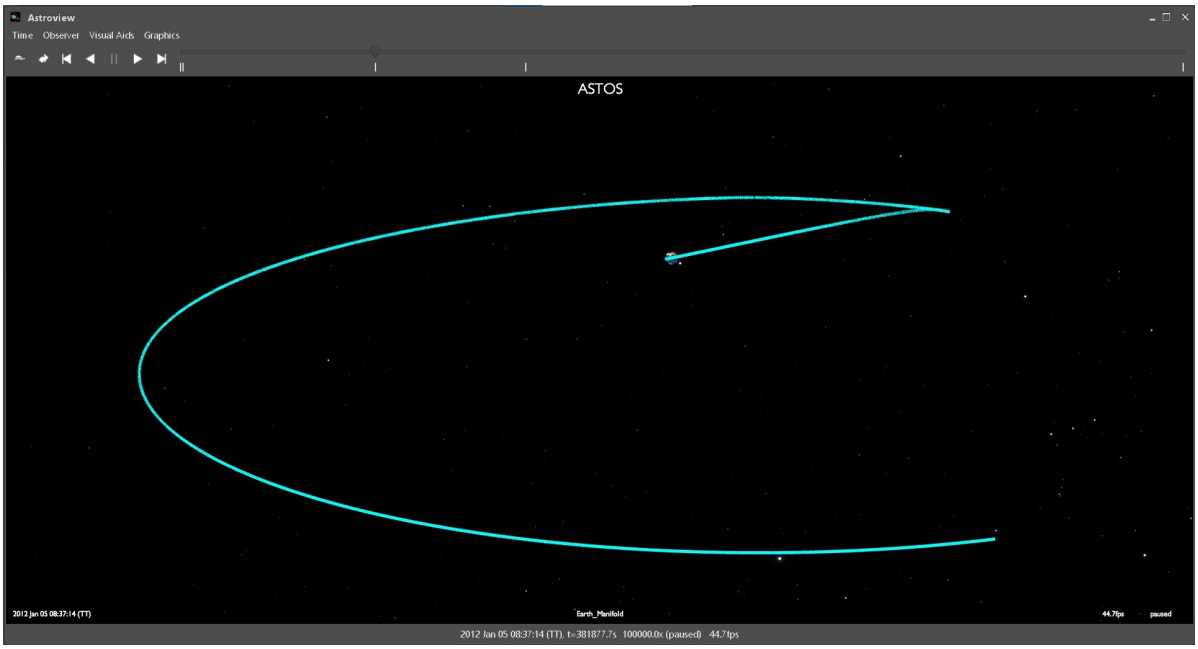

Figure 9.12: Solution trajectory in the inertial J2000 frame as viewed in ASTOS.

10

Transfer Scenario 2: GTO to Gateway NRHO

Whereas Chapter 9 used a transfer problem where multiple solutions were available from literature, this chapter will optimize a trajectory from a GTO to an L_2 9:2 period-resonant NRHO, which corresponds to the orbit that will be used by the lunar Gateway. The objective is to show the versatility of the tool developed, and how intuitive it is to set a new optimization problem. In the same way, the capabilities of the software are presented, proving that it is not limited to circular or planar orbits but that it is completely flexible to adapt to the user needs, without any loss in the accuracy of the results.

10.1. Trajectory Description

As said, the trajectory to be optimized starts at a GTO. The precise definition of the shape of the orbit, which can be seen in Figure 10.1, consists of a semi-major axis of 24239 km and an eccentricity of 0.726845. This is obtained from calculating an intersecting arc between a LEO and a GEO. GTOs are often considered as suitable parking orbits when the destination trajectory is a GEO, but also for space exploration missions. As in the previous example, the orientation angles are left unrestricted, except for a maximum inclination of 35°.

With respect to the destination orbit, the future lunar Gateway orbit was selected. Orbiting the L_2 point in a 9:2 resonant NRHO, the Gateway is expected to become an outpost for many future missions, thus bringing a large amount of attention. In order to define it for the tool, its characteristics are introduced in the *initialSettings.txt* file as shown in Figure 10.1. In this case it is identified by a period of 6.56 days.

Figure 10.1 shows the *initialSettings.txt* files for the multi-objective run of the manifold transfer (left) and the flyby case (right). As can be seen, the only difference is in the trajectory string and in the name of the phases. Once again, four multi-objective runs with different random seed numbers (107, 468, 569, 823) will be performed with each configuration (manifold and flyby), resulting in a converged Pareto front with a set of optimal solutions. From those, a result will be selected, a refinement run will be performed and a detail visualization of the trajectory characteristics will be shown, both from ASTOS and from an independent plotting tool that was developed with *Matplotlib* [21].

10.2. Multi-objective Optimization Results

This section contains the results of the eight optimization runs, four random seed numbers for each of the two configurations. After a brief discussion on the individual Pareto fronts for each approach, the solutions are merged on the final multi-objective optimization result, from where a trajectory is chosen for further investigation.

10.2.1. Manifold Case

Before starting the optimization and as the initial settings have been already set, as seen in Figure 10.1, the design variable bounds need to be chosen. For the manifold configuration, this selection is shown

```
Trajectory      KLMP
Phase names     GTO, lambert, manifold, halo
Julian date     2460676.50000
maxTOF  25.0
maxDV   10.0

Seed    107.0
optim   0
maxeval 999999999
maxtime 3000
evalstop        0.0

Semi-major axis 24239.2040
Eccentricity    0.7268450

Orbit Type      HALO-L2-SOUTHERN
Parameter       Period
Value   6.560
```

```
Trajectory      KLFP
Phase names     GTO, lambert, flyby, halo
Julian date     2460676.50000
maxTOF  25.0
maxDV   10.0

Seed    107.0
optim   0
maxeval 999999999
maxtime 3000
evalstop        0.0

Semi-major axis 24239.2040
Eccentricity    0.7268450

Orbit Type      HALO-L2-SOUTHERN
Parameter       Period
Value   6.560
```

Figure 10.1: Case scenario 1 *initialSettings.txt* file for the manifold (left) and the flyby case (right).

in Figure 10.2. As in the previous case, the Keplerian angles are left with open bounds as the orbit is not restricted to any particular inclination.

```
      i       RAAN     omega    theta    TOF_L    TOF_M    Delta    dVx      dVy      dVz      p
x     0.0,    0.0,     0.0,     0.0,     5.0,     10.0,    -3.0,    0.0,     0.0,     0.0,     0.50
xl    0.0,    0.0,     0.0,     0.0,     1.0,     0.01,    -6.0,    -100.0,  -100.0,  -100.0,  0.0
xu    35.0,   360.0,   360.0,   360.0,   10.0,    20.0,    -2.0,    100.0,   100.0,   100.0,   1.0
```

Figure 10.2: *initialGuess.txt* file for the case scenario 2 manifold configuration.

The Lambert arc is again restricted to a time of flight between 1 and 10 days, and the manifold arc, between 0.01 and 20 days to consider a direct transfer scenario. Regarding the velocity perturbation values, $\pm$1000 m/s were considered, but the results did not improve with the wider range. Thus, they will not be included in this section. Finally the bounds were set to $\pm$100 m/s to allow for more maneuvers that can minimize the ΔV for any given TOF.

The four optimization runs with the different seed numbers took 8 hours to obtain the converged results that can be seen in Figure 10.3 as an average. The computational time was around 20% shorter than in the previous transfer, which can be explained because the GTO is further away from the Earth, which reduces the sensitivity of the trajectory and improves the integration time. Hence, the running times shown should serve as a guidance for future calculations, but they will depend on the optimization problems and the computational power of the computer used.

As can be appreciated, in this simulation the different seed numbers provided very much the same results, except for seed number 569, which does not appear in the right-hand side of the Pareto front. Nonetheless, it is still a good practice to run several seed numbers, as it is not clear beforehand whether the results will be affected.

Figure 10.4 shows the combination of the best solutions of each of the seed number runs. If this was the full solution, it could be interesting to analyze further the 8-day transfer, as the improvement in ΔV seems negligible compared to the increase in time of flight. The lowest ΔV result takes 17 days, which is more than doubling the duration, while it only saves around 7% of propellant, from 1.62 km/s to 1.51 km/s, approximately.

In the next section, it will be seen whether the flyby approach is able to improve the results shown in Figure 10.4 as was the case in Chapter 9, or if by the contrary, in this case the manifold configuration was a better suited approach to solve the problem.

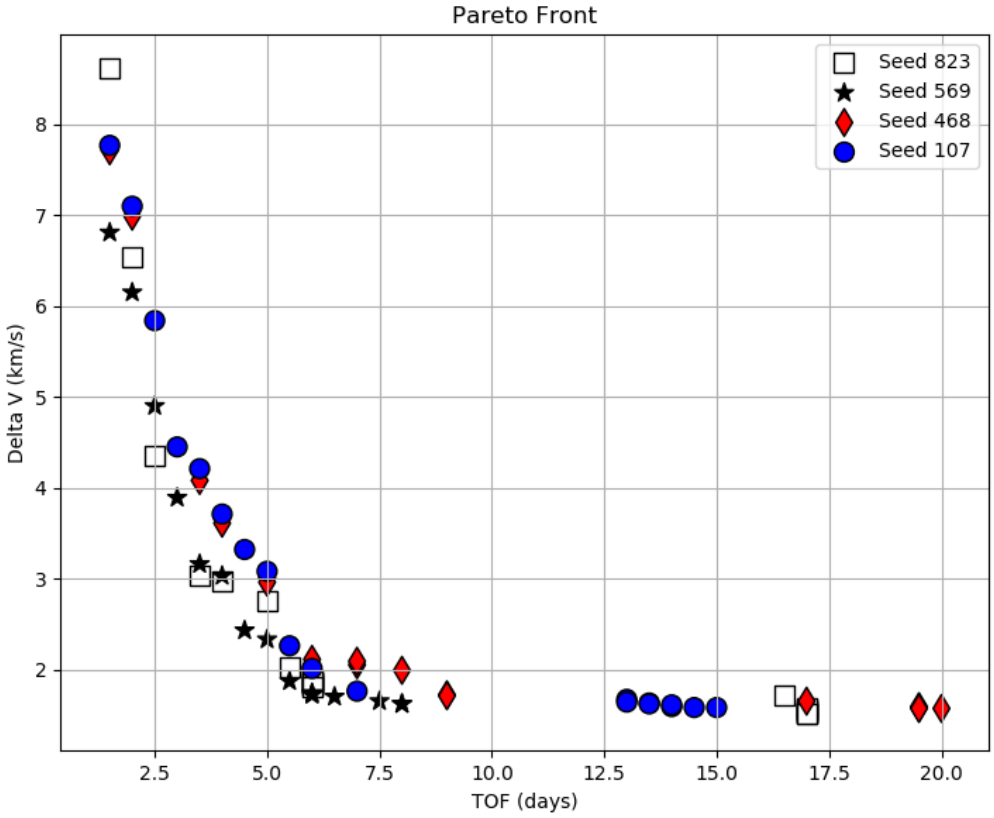

Figure 10.3: Pareto front including all runs for the case scenario 2 manifold configuration.

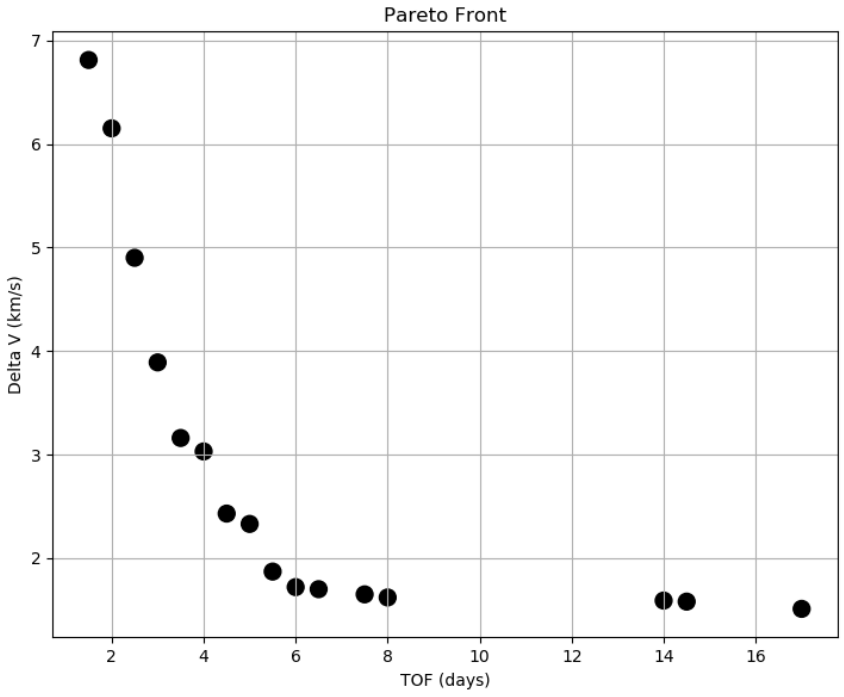

Figure 10.4: Case scenario 2 manifold configuration combined Pareto front.

10.2.2. Flyby Case

With respect to the flyby configuration, the variable bounds were slightly different. The Lambert arc, Keplerian angles and Halo orbit point maintain exactly the same bounds. For the flyby arc, the time of flight bounds are identical to the manifold arc, but the rest of the variables differ. The altitude is again limited to 100 km to have comparable results to those from the literature while keeping a margin of safety from the Moon surface. However, the maximum bound is extended to 3000 km, as it was seen during the optimization process that widening the range of this variable improved the optimization results. The polar and azimuth angles have completely open bounds.

```
        i       RAAN      omega     theta     TOF_L     TOF_F     r         polar     azimuth   p
x       0.0,    0.0,      0.0,      0.0,      5.0,      10.0,     100.0,    90.0,     0.0,      0.50
xl      0.0,    0.0,      0.0,      0.0,      1.0,      0.01,     100.0,    0.0,      0.0,      0.0
xu      35.0,   360.0,    360.0,    360.0,    10.0,     20.0,     3000.0,   180.0,    360.0,    1.0
```

Figure 10.5: *initialGuess.txt* file for the case scenario 2 flyby configuration.

In this case, the four optimization runs took again an average of 8 hours. The results from the four runs can be seen in Figure 10.6. It can be clearly noticed how the results from the four runs are more diverse than in the manifold case. This means that the robustness of this approach is lower than the manifold solution, and hence, the impact of the pseudo-random seed is higher, at least for the trajectory problem analyzed. This is the reason why using multiple seed numbers is still recommended.

The results from the four runs were combined and a dominance filter was applied, creating a Pareto front which is shown in Figure 10.7. In this case, it is not so clear whether the results from the flyby configuration considerably improve those of the manifold approach, especially for lower TOFs.

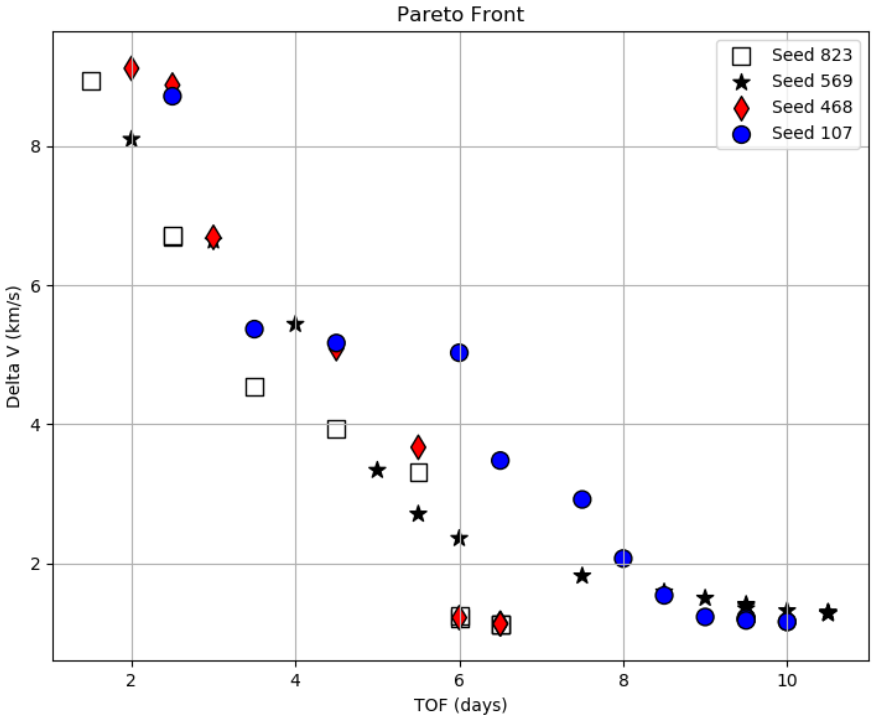

Figure 10.6: Pareto front including all runs for the case scenario 2 flyby configuration.

After combining the results from the both configurations in Figure 10.8, it is clear that the flyby configuration improves the results of the manifold scenario for longer TOFs. The minimum ΔV solution requires 1.12 km/s for the full transfer and only 6.25 days, versus 1.51 km/s and 16.5 days for the manifold approach. However, for shorter duration transfers the manifold configuration obtained better results. Investigating the solutions it is found that they mostly consist of direct transfers. In all cases the manifold arc duration is lower than six hours.

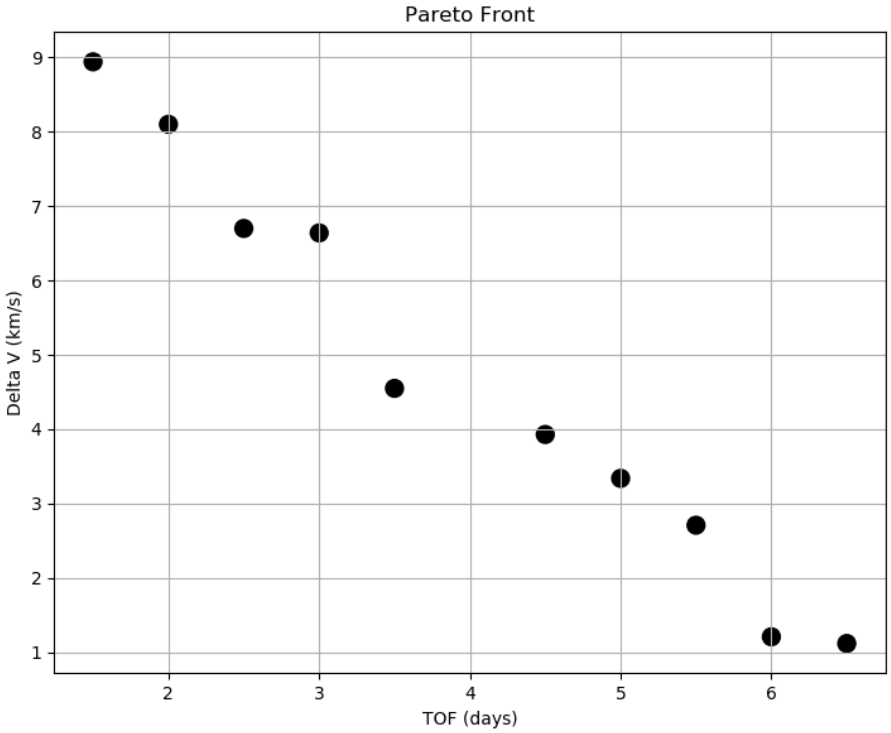

Figure 10.7: Case scenario 2 flyby configuration combined Pareto front.

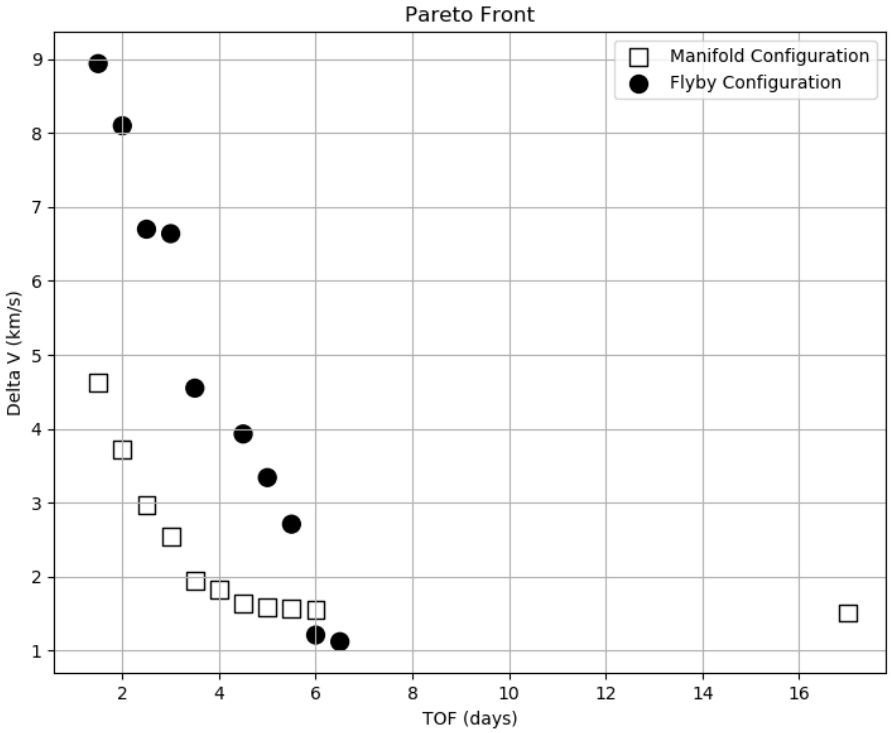

Figure 10.8: Case scenario 2 final Pareto front results.

10.3. Refined Solution

For the refined solution, the chosen result from the multi-objective optimization was the 4.5-days long transfer, with a ΔV consumption of 1.64 km/s. Although this was not the lowest ΔV solution, it has a very good compromise between time of flight and propellant consumption, reaching the desired orbit in 50% less time than the minimum ΔV result.

To increase the convergence speed of the single-objective optimization, the obtained result was set as the initial guess. In the *initialSettings.txt* file, the *optim* value is changed to 1. Moreover, the maximum time of flight limit (*maxTOF*) is set to 4.5 days. This is done to avoid the optimizer finding a better solution in terms of ΔV by increasing the time of flight. The design variable bounds are kept as they were, except for the Lambert and manifold upper TOF limit, which is set to 4,5 days to reduce the search space and improve the speed.

After another four runs, each with a different random seed number, the lowest ΔV solution was selected. The single-objective optimizer was able to improve the result found by the multi-objective run from an initial result of 1.64 km/s to a final solution of 1.56 km/s of total ΔV, which equals a reduction of around 5%. The time of flight remained unchanged, at 4.5 days.

In this case, the selected transfer did not have as many previous solutions available as in Chapter 9. Zazzera et al. [4] studied transfers from a GTO in the CR3BP and their results are shown in Table 10.1. Nevertheless, the destination orbit of these solutions is a southern L_2 Halo orbit with A_z 1000 and 8000 km. Hence the results are not entirely comparable, but can be used to have an approximate idea on the values expected. It can be seen that the minimum ΔV solution obtained (1.12 km/s) is within a 10% margin from these results, albeit the time of flight is considerably less. Therefore, the results found are more suitable for manned missions to the Gateway orbit, whereas Zazzera et al. results could be used for cargo missions.

Reference	**ΔV Lit.** [km/s]	**TOF Lit.** [days]
Current work	1.12	6.25
Zazzera [4]	0.99	123.1
	0.95	77.4
	1.04	42.8

Table 10.1: Comparison of obtained solutions with literature results

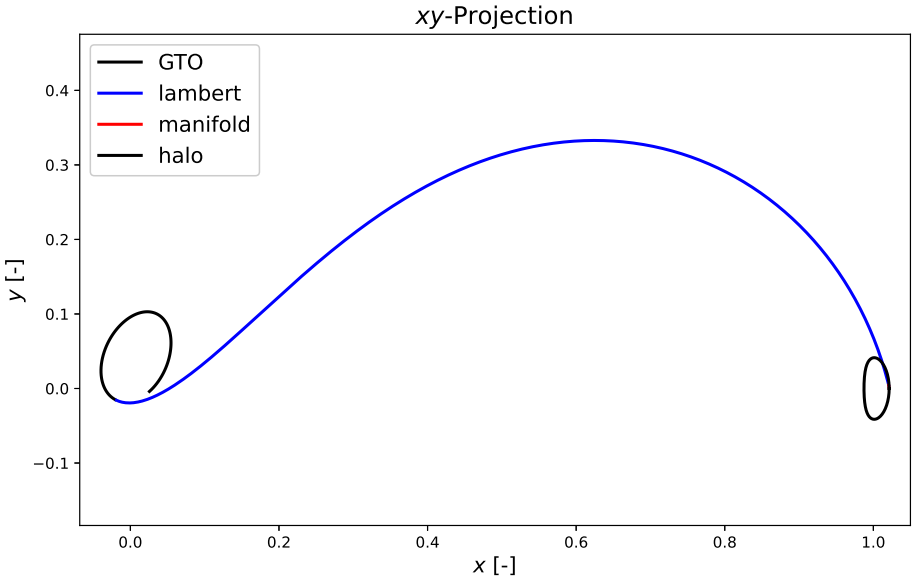

Figure 10.9: Solution trajectory in the barycentric rotating frame XY projection.

Using the same independent plotting tool introduced in the previous chapter, the xy-projection of the transfer in the barycentric rotation frame can be obtained and is shown in Figure 10.9. The three-dimensional view, with unscaled axes, is shown in Figure 10.10. It is noticeable how in neither of these two figures the manifold arc is shown, provided that the solution is a direct transfer, as commented.

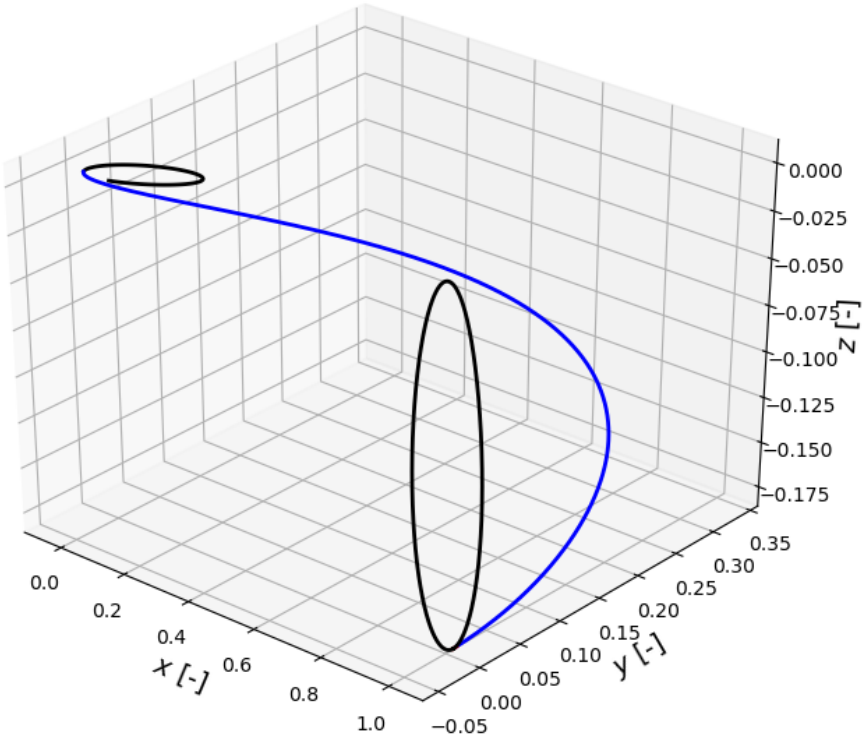

Figure 10.10: Solution trajectory in the barycentric rotating frame 3D projection.

When exporting the result into ASTOS the same trajectory plot in the barycentric rotating frame can be obtained, as seen in Figure 10.11.

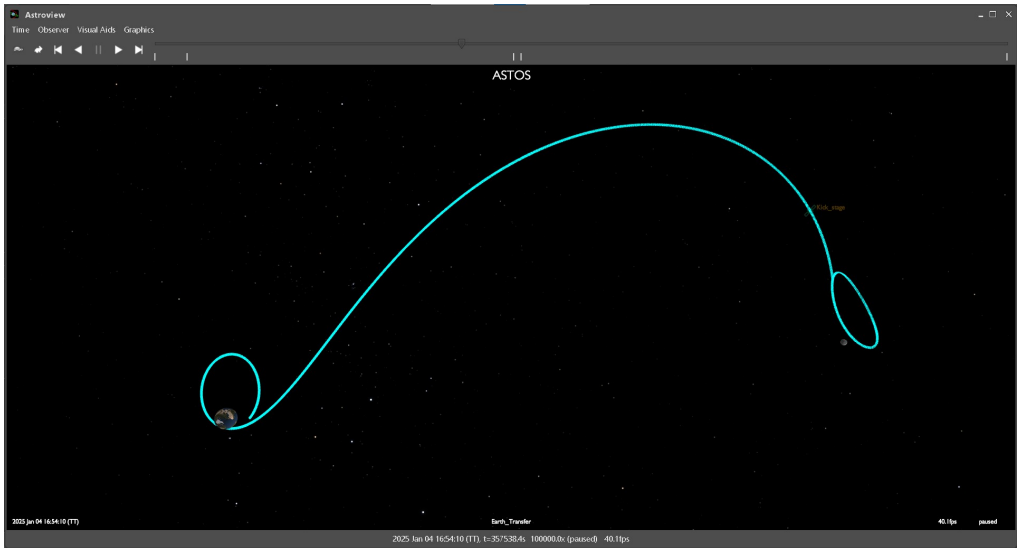

Figure 10.11: Solution trajectory in the barycentric rotating frame as viewed in ASTOS.

The inertial representation of the trajectory can also be obtained, as seen in Figure 10.12. It can be

noted how the NRHO has a large out-of-plane component that has its peak at the periselene.

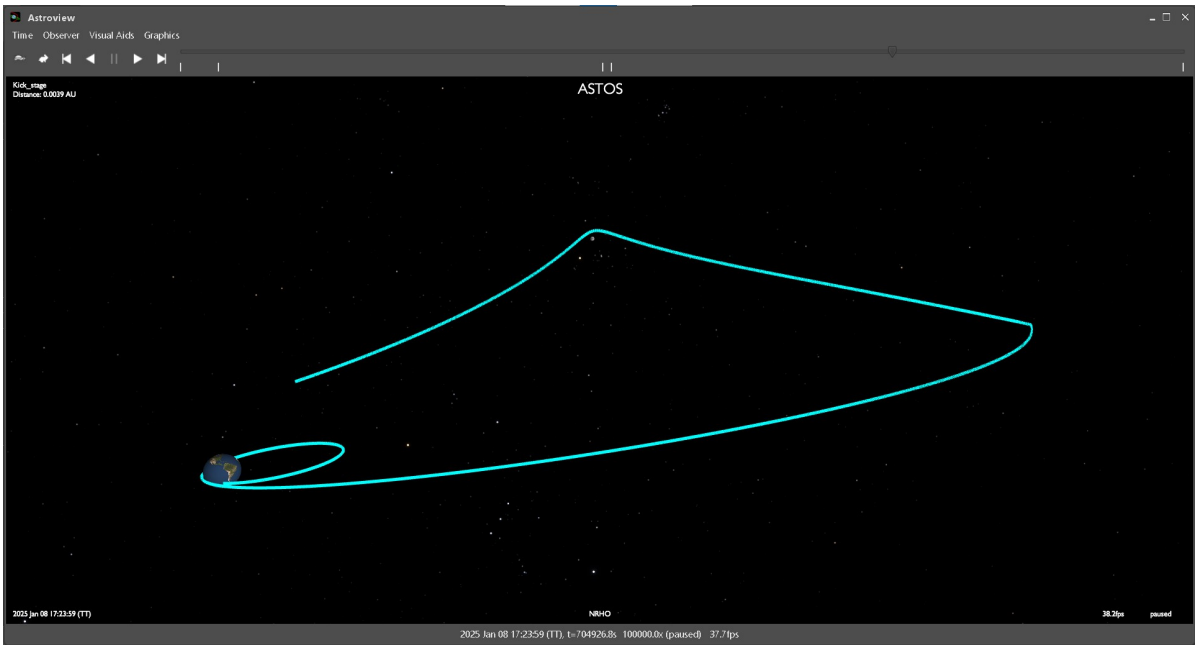

Figure 10.12: Solution trajectory in the inertial J2000 frame as viewed in ASTOS.

In order to test the sensitivity of the solution found, different optimization runs were performed with the selected trajectory as initial guess, but changing the departure date by intervals of 6 hours. The results obtained can be seen in Table 10.2. As it appears, although there are some changes in the ΔV obtained, a trajectory with similar characteristics could be achieved if the launch window is modified.

Δ **Initial Date** [hours]	Δ**V** [km/s]	**TOF** [days]
0	1.56	4.50
+6	1.58	4.50
+12	1.62	4.46
-6	1.60	4.49
-12	1.64	4.48

Table 10.2: Trajectory initial date sensitivity analysis

11

Conclusions and Recommendations

In the final chapter of this report a brief summary of the whole investigation will be presented, emphasizing its most important aspects and reflecting on the knowledge acquired. Furthermore, the research question formulated in Chapter 1 will be recovered to see the extent to which a satisfactory answer was found. Finally, a series of recommendations will be suggested for a future work of this particular area, providing useful ideas for the development of this project.

11.1. Summary

The advantages of this tool compared to the alternatives available in the literature are its versatility and flexibility. Thanks to these features, the user is able to optimize the main trajectories in the CR3BP regardless of the initial and destination orbits. Moreover, the various transfer arc approaches (direct, manifold and flyby) enable different configurations that can be combined to find other optimal solutions. Finally, the multi-objective optimization approach is key to provide the user with a range of solutions such that the compromise between requirements is satisfactorily achieved.

The most relevant factor for the success of this study was the careful selection of design variables. It was very important to choose the transfer configurations such that the design space was not restricted and the global search optimizer could analyze it on its entirety. This was especially important given the multi-objective nature of the project, as the approaches that might be suitable for low ΔV transfers, may not coincide with shorter solutions, and viceversa.

Another crucial decision to increase the design space was to opt for a multiple-shooting strategy inside the Lambert arc calculator. In this way, the sensitivity of the integrator was reduced, obtaining more converged solutions and allowing the global search optimizer to have more information and thus, find optimal solutions with a lower number of iterations. However, this also increased the computational time of each function evaluation.

Moreover, it was a main requirement that the tool was intuitive such that users without a large knowledge on the CR3BP or trajectory optimization are able to make use of it and get a first impression of the transfer requirements for their mission. As an example Tharshan [28] used a beta version of the tool developed in this investigation to compute transfers from Earth (both GEO and GTO) to Halo orbits around the lunar L_2 as one of the stages of a sun-shielding mission investigation. To achieve this user-friendliness, the tool was set up such that the user was required to introduce the least amount of information possible, without losing the generality of the optimization capabilities. Input files *initialSettings.txt* and *initialGuess.txt* are the result of this process.

The decision of choosing a global search algorithm as the optimization method was especially suited for this multi-objective problem. As the design space was analyzed in all regions regardless of the number of objectives, the global search algorithm came with a wider range of solutions without requiring a significant increase of function evaluations with respect to a single-objective approach.

Besides obtaining a wider range of solutions than the previous literature, the results achieved were also better in terms of ΔV and mission duration, as seen in Chapters 9 and 10. This improvement was mainly due to the lack of restrictions in the design space and the tool's ability of trying different transfer configurations simultaneously. This aspect can be considered a complete success with regard to the

tool evaluation.

The results also show how the manifold approach resulted in a sub-optimal configuration, as the aim is to minimize the overall ΔV, rather than the final maneuver. The flyby approach, on the other hand, provided outstanding results, especially regarding the low mission's duration. This makes them very suitable for manned missions. The improvement with respect to Mingtao et al. [36] was mainly due to the flyby arc not being restricted to start from a manifold, thanks to which more possibilities could be analyzed.

In addition, the possibility of exporting the results into a higher-fidelity resource such as the ASTOS software is an added value which allows to easily perform a more accurate analysis of the characteristics of the selected transfer.

Finally, the only aspect that could be questioned is the large computational time cost of this software. 10 hours can appear as a very large time for a first stage tool in the mission analysis process. However, this was expected when the global search optimizer was selected. Secondly, it must be noted that several seeds can be running in parallel, which greatly reduces the computational effort. Finally, the effect of the running time is mitigated by the fact that after the optimization process, not only one but several optimal solutions are obtained, which brings down the effective time cost per solution.

11.2. Research Question

Before judging whether the research objectives of this investigation have been fulfilled, the research question and sub-questions are recalled.

> Is there a user-intuitive way to optimize a wide range of trajectories in the CR3BP model from a multi-objective perspective?
>
> How will the tool be user-intuitive?
>
> What constitutes a wide-range of trajectories in the CR3BP? Is this limited to departure and destination orbits options, or also to transfer arcs?
>
> Which multi-objective algorithm is best suited to solve this problem?

As it has been proven throughout the project, it is perfectly possible to design such said tool, however, some aspects need to be discussed in a deeper way. The analysis of whether the question has been successfully answered in this study will be broken down in three paragraphs: intuitiveness, trajectories and optimization results.

Regarding the intuitiveness of the developed tool, the use of the *initialSettings.txt* and *initialGuess.txt* input files effectively allows the user to introduce the transfer characteristics in a simple way. Moreover, the output files (Pareto front, optimization process state and trajectory data) are very easily understandable and with the additional plotting tools, a visual representation is readily available. Overall it can be said that the tool presents a high level of user-friendliness, although there are some improvements that can be made which will be described in the next section.

With respect to the trajectories of the CR3BP the tool is able to connect Keplerian orbits, and any kind of Lyapunov, Halo orbits or CR3BP in any combination possible. Although this is not an extensive list of all the periodic solutions available in the CR3BP it is more than enough to satisfy a large portion of the missions of scientific and industrial interest of the present days.

Finally, as far as the optimization results are concerned, the tool developed has been able to provide more and better solutions than those available in literature. As such, this aspect can be considered a complete success.

Overall, it can be concluded that the research objectives that were set for this investigation have been met. Nonetheless, there is always room for improvement, and some good ideas related to this topic will be discussed in the next section.

11.3. Future Work Recommendations

This section contains some suggestions which are intended to improve different aspects of the developed tool.

In order to increase the user-friendliness of the program, it could be interesting to create a Graphical User Interface (GUI) such that the design variables are automatically updated depending on the

trajectory configuration used. Moreover, the refinement run could be started by an interactive selection of the region of the Pareto front that is of interest for the user, rather than by setting a new optimization case manually. Finally, the whole tool could be integrated inside ASTOS such that the solution does not need to be exported into the software, but the mission analyst is ready to continue the optimization with the selected trajectory.

Another improvement is expanding the set of periodic solutions that can be introduced as destination or parking orbits. Although Lyapunov, Halo orbits and NRHOs account for the majority of the exploration missions in recent years, other orbits such as DROs or quasi-periodic solutions such as Lissajous orbits could be suitable for future analysis.

Finally, in order to reduce the computational cost of the tool, other optimization algorithms could be studied. As MIDACO treats the objective and constraint functions as a black-box, the code can easily be moved into a different solver.

Finally, a suggestion that could be employed to tackle other type of problems is to include the initial date as one of the design variables. This would be useful to study the launch window options and create pork-chop plots of the trajectory options.

Bibliography

[1] Nima Assadian and Seid H Pourtakdoust. Multiobjective genetic optimization of Earth–Moon trajectories in the restricted four-body problem. *Advances in Space Research*, 45(3):398–409, 2010.

[2] June Barrow-Green. *Poincaré and the three body problem*. American Mathematical Soc., 1997.

[3] Joe Burt and Bob Smith. Deep space climate observatory: The DSCOVR mission. In *2012 IEEE Aerospace Conference*, pages 1–13. IEEE, 2012.

[4] E Canalias, G Gomez, M Marcote, and JJ Masdemont. Assessment of mission design including utilization of libration points and weak stability boundaries. *ESA Advanced Concept Team*, 2004.

[5] A Constantin and RS Johnson. Large gyres as a shallow-water asymptotic solution of Euler's equation in spherical coordinates. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2200):20170063, 2017.

[6] William Crossley. Purdue University. AAE 550 - Multidisciplinary Design Optimization, (Retrieved: October 2020).

[7] C Davis, SA Bhatt, KC Howell, J Jang, R Whitley, FD Clark, D Guzzetti, EM Zimovan, and GH Barton. Orbit maintenance and navigation of human spacecraft at cislunar Near Rectilinear Halo Orbits, 27th AAS. In *AIAA Space Flight Mechanics Meeting, AAS*, pages 17–269, 2017.

[8] JHJ De Bruijne. Science performance of Gaia, ESA's space-astrometry mission. *Astrophysics and Space Science*, 341(1):31–41, 2012.

[9] Michael TM Emmerich and André H Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17(3):585–609, 2018.

[10] Robert W Farquhar. The Control and Use of Libration-Point Satellites. *Ph. D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford University. Stanford, CA, 1968*, 1968.

[11] Robert W Farquhar. The flight of ISEE-3/ICE: origins, mission history, and a legacy. *The Journal of the Astronautical Sciences*, 49(1):23–73, 2001.

[12] Heather Franz, Peter Sharer, Keith Ogilvie, and Mike Desch. WIND nominal mission performance and extended mission design. *The Journal of the Astronautical Sciences*, 49(1):145–167, 2001.

[13] P Gath, A Wiegand, A Markl, and KH Well. Recent improvements in the trajectory optimization software ASTOS. In *High Performance Scientific And Engineering Computing*, pages 305–312. Springer, 2002.

[14] Gerard Gómez and Josep M Mondelo. The dynamics around the collinear equilibrium points of the RTBP. *Physica D: Nonlinear Phenomena*, 157(4):283–321, 2001.

[15] Dawn Perry Gordon. *Transfers to Earth-Moon L2 Halo Orbits*. PhD thesis, Purdue University West Lafayette, 2008.

[16] NASA Hambleton, Kathryn. Deep Space Gateway to open opportunities for distant destinations, (Retrieved: October 2020). URL `https://www.nasa.gov/feature/deep-space-gateway-to-open-opportunities-for-distant-destinations`.

[17] Jeanette Heiligers. TU Delft. AE4889 - Special Topics in Astrodynamics Lecture Notes, (Retrieved: September 2020).

[18] Kathleen C Howell. Families of orbits in the vicinity of the collinear libration points. *The Journal of the Astronautical Sciences*, 49(1):107–125, 2001.

[19] Kathleen C Howell, Diane C Davis, and Amanda F Haapala. Application of periapse maps for the design of trajectories near the smaller primary in multi-body regimes. *Mathematical Problems in Engineering*, 2012, 2012.

[20] Kathleen Connor Howell. Three-dimensional, periodic,'halo'orbits. *Celestial mechanics*, 32(1): 53–71, 1984.

[21] John D Hunter. Matplotlib: A 2D graphics environment. *IEEE Annals of the History of Computing*, 9(03):90–95, 2007.

[22] Dario Izzo. Pygmo and Pykep: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization). In *Proceedings of the Fifth International Conference on Astrodynamics Tools and Techniques, ICATT*, 2012.

[23] Dario Izzo. Revisiting Lambert's problem. *Celestial Mechanics and Dynamical Astronomy*, 121 (1):1–15, 2015.

[24] Pierre Kokou, Bastien Le Bihan, Jean-Baptiste Receveur, and Stéphanie Lizy-Destrez. Computing an optimized trajectory between Earth and an EML2 halo orbit. In *AIAA Guidance, Navigation, and Control Conference*, page 0450, 2014.

[25] David E Lee. White Paper: Gateway destination orbit model: a continuous 15 year NRHO reference trajectory. *NASA*, 2019.

[26] Hanlun Lei and Bo Xu. Transfers between libration point orbits of Sun–Earth and Earth–Moon systems by using invariant manifolds. *Journal of Engineering Mathematics*, 98(1):163–186, 2016.

[27] M Lo, BG Williams, WE Bollman, D Han, Y Hahn, JL Bell, EA Hirst, RA Corwin, PE Hong, KC Howell, et al. Genesis Mission Design, Paper No. *AIAA 98*, 4468, 1998.

[28] Tharshan Maheswaran. *Analysis of logistical construction aspects of a sunshade concept in the vicinity of the Sun Earth L1 Lagrange Point (IRS-21-S-017)*. PhD thesis, University of Stuttgart, 2021.

[29] F Markley, Stephen Andrews, James O'Donnell, and David Ward. The Microwave Anisotropy Probe (MAP) Mission. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4578, 2002.

[30] John Mather. James Webb Space Telescope. In *Space 2004 Conference and Exhibit*, page 5985, 2003.

[31] Zachary D May, Min Qu, and Raymond Merrill. Enabling Global Lunar Access for Human Landing Systems Staged at Earth-Moon L2 Southern Near Rectilinear Halo and Butterfly Orbits. In *AIAA Scitech 2020 Forum*, page 0962, 2020.

[32] Brian P McCarthy. *Characterization of Quasi-Periodic Orbits for Applications in the Sun-Earth and Earth-Moon Systems*. PhD thesis, Purdue University Graduate School, 2019.

[33] A Miele and S Mancuso. Optimal trajectories for Earth–Moon–Earth flight. *Acta Astronautica*, 49 (2):59–71, 2001.

[34] Angelo Miele. Theorem of image trajectories in the Earth-Moon space. In *XIth International Astronautical Congress Stockholm 1960/XI. Internationaler Astronautischer Kongress/XIe Congrès International D'Astronautique*, pages 385–391. Springer, 1961.

[35] LI Ming-Tao and Zheng Jian-Hua. The optimization of transfer trajectory for small amplitude halo orbits. *Measurement and Control*, 41(3):81–84, 2008.

[36] Li Ming-Tao and Zheng Jian-Hua. Impulsive lunar halo transfers using the stable manifolds and lunar flybys. *Acta Astronautica*, 66(9-10):1481–1492, 2010.

[37] Masaki Nakamiya, D. Scheeres, Hiroshi Yamakawa, and Masayuki Yoshikawa. Analysis of capture trajectories to libration points. *Advances in the Astronautical Sciences*, 127, 01 2007.

[38] NASA. International space station on june 12, 2014, 12:00:00 hrs, 2014.

[39] NASA. Gateway configuration concept, (Accessed: 08.04.2021). URL https://www.nasa.gov/feature/multilateral-coordination-board-joint-statement.

[40] Mollik Nayyar. Optimal Earth Return Transfers from Lagrange Point Orbits Using Particle Swarm Optimization. *PennState Univeristy. Master Thesis.*, 2016.

[41] Isaac Newton. *Principia Mathematica*. London Royal Society, 1687.

[42] NORAD. Cryosat-2 on june 13, 2014, 14:59:21hrs, 2014.

[43] The Editors of Encyclopaedia Britannica. Orbit, (Retrieved: October 2020). URL https://www.britannica.com/science/orbit-astronomy.

[44] United States Naval Observatory. Nautical Almanac Office, Great Britain. Nautical Almanac Office, Engineering Research Council (Great Britain), Rutherford Appleton Laboratory, Council for the Central Laboratory of the Research Councils (Great Britain), United States. Department of the Navy, and Great Britain. Hydrographic Office. *The Astronomical Almanac*. US Government Printing Office, 2008.

[45] Jeffrey S Parker. *Low-energy ballistic lunar transfers*. PhD thesis, University of Colorado at Boulder, 2007.

[46] Jeffrey S Parker and George H Born. Direct lunar halo orbit transfers. *The Journal of the Astronautical Sciences*, 56(4):441–476, 2008.

[47] JS Parker and RL Anderson. Low-Energy Lunar Trajectory Design, Vol. 12 of DESCANSO Deep Space Communication and Navigation Series. *Jet Propulsion Laboratory, California Institute of Technology*, 2013.

[48] Thomas A Pavlak. *Mission design applications in the earth-moon system: Transfer trajectories and stationkeeping*. PhD thesis, Purdue University, 2010.

[49] Thomas A Pavlak and Kathleen C Howell. Evolution of the out-of-plane amplitude for quasi-periodic trajectories in the Earth–Moon system. *Acta Astronautica*, 81(2):456–465, 2012.

[50] Lei Peng, Yuanzhen Wang, Guangming Dai, Yamin Chang, and Fangjie Chen. Optimization of the Earth-Moon low energy transfer with differential evolution based on uniform design. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

[51] Henri Poincaré. *The new methods of celestial technique*, volume 3. Gauthier-Villars, 1899.

[52] DL Richardson. Analytical construction of periodic orbits about the collinear points of the Sun-Earth system. *Astrodynamics conference, Paper 79-127, 15 p.*, page 127, 1980.

[53] C Roberts and R Short. Injection contingency recovery strategies for halo orbit transfer trajectories. In *Astrodynamics Conference*, page 3600, 1996.

[54] Ralph B Roncoli. Lunar constants and models document, JPL D-32296. *Jet Propulsion Laboratory, California Institute of Technology*, 2005.

[55] Shane Ross and Martin Lo. The lunar L1 gateway-portal to the stars and beyond. In *AIAA Space 2001 Conference and Exposition*, page 4768, 2001.

[56] AE Roy and MW Ovenden. On the occurrence of commensurable mean motions in the solar system: the mirror theorem. *Monthly Notices of the Royal Astronomical Society*, 115(3):296–309, 1955.

[57] Martin Schlüter and Matthias Gerdts. The Oracle penalty method. *Journal of Global Optimization*, 47(2):293–325, 2010.

[58] Martin Schlüter, Jose A Egea, and Julio R Banga. Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Computers & Operations Research*, 36(7):2217–2229, 2009.

[59] M Schlüter and M Munetomo. MIDACO Solver-User Manual. *IIC, Hokkaido University, Japan*, 2016.

[60] M. Schlüter, S. Erb, M. Gerdts, S. Kemble, and J.J. Ruckmann. MIDACO on MINLP Space Applications. *Advances in Space Research*, 51(7):1116–1131, 2013. doi: 10.1016/j.asr.2012.11.006.

[61] Martin Schlüter, Chit Hong Yam, Takeshi Watanabe, and Akira Oyama. Parallelization impact on many-objective optimization for space trajectory design. *International Journal of Machine Learning and Computing*, 6(1):9, 2016.

[62] Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.

[63] Peter Sharer and Tammy Harrington. Trajectory optimization for the ACE halo orbit mission. In *Astrodynamics Conference*, page 3601, 1996.

[64] Yuchul Shin, Seyoung Yoon, Yongmyung Seo, Ho Jin, and Jongho Seon. Radiation effect for a CubeSat in slow transition from the Earth to the Moon. *Advances in Space Research*, 55(7): 1792–1798, 2015.

[65] C Simó, G Gómez, J Llibre, R Martinez, and J Rodriguez. On the optimal station keeping control of halo orbits. *Acta Astronautica*, 15(6-7):391–397, 1987.

[66] V Szebehely. *Theory of orbits: the restricted three body problem*. Academic Press San Diego, 1967.

[67] Jose Tatay Sanguesa. Multi-objective Optimization of Impulsive Orbital Trajectories. *Bachelor of Science Thesis, Universitat Politècnica de València, Spain*, 2019. URL http://hdl.handle.net/10251/124982.

[68] Francesco Topputo. On optimal two-impulse Earth–Moon transfers in a four-body model. *Celestial Mechanics and Dynamical Astronomy*, 117(3):279–313, 2013.

[69] Karel F Wakker. *Fundamentals of astrodynamics*. TU Delft Library, 2015. URL http://resolver.tudelft.nl/uuid:3fc91471-8e47-4215-af43-718740e6694e.

[70] Sven Weikert, Alexander Dobler, Jochen Teufel, Sven Schäff, Valentino Zuccarelli, Martin Jürgens, Andreas Wiegand, and Sven O Erb. ASTOS 8.1-Mission performance analysis, system analysis and other new features, 2016.

[71] Ryan Whitley and Roland Martinez. Options for staging orbits in cislunar space. In *2016 IEEE Aerospace Conference*, pages 1–9. IEEE, 2016.

[72] Ryan J Whitley, Diane C Davis, Laura M Burke, Brian P McCarthy, Rolfe J Power, Melissa L McGuire, and Kathleen C Howell. Earth-moon near rectilinear halo and butterfly orbits for lunar surface exploration. In *AAS/AIAA Astrodynamics Specialists Conference, Snowbird, Utah*, 2018.

[73] Cmglee Wikimedia. Lagrangian points equipotential, (Retrieved: September 2020). URL https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Lagrangian_points_equipotential.jpg/660px-Lagrangian_points_equipotential.jpg.

[74] Mark Woodard, David Folta, and Dennis Woodfork. ARTEMIS: the first mission to the lunar libration orbits. In *21st International Symposium on Space Flight Dynamics, Toulouse, France*, 2009.

[75] Emily M Zimovan. *Characteristics and Design Strategies for Near Rectilinear Halo Orbits within the Earth-Moon System*. PhD thesis, Purdue University, 2017.