Optimal Low-Thrust Transfers Between Halo Orbits in the Circular Restricted Three-Body Problem

Master of Science Thesis

J. Bas Fernandez

Astos Solutions

TU Delft

Optimal Low-Thrust Transfers Between Halo Orbits in the Circular Restricted Three-Body Problem

Master of Science Thesis

by:

J. Bas Fernandez

to obtain the degree of Master of Science
at Delft University of Technology
to be defended publicly on Friday, September 13th, 2024 at 13:30

Student number:	5020239
Project duration:	December 4, 2023 - August 26, 2024

TU Delft thesis committee:

Dr. ir. D. Dirkx,	Chair
Ir. K.J. Cowan MBA,	Independent examiner
Ir. M. Walther,	External supervisor (Astos Solutions GmbH)
Dr. ir. E. Mooij,	Supervisor

[1]Cover Image URL: https://playgroundai.com/post/labyrinthine-moon-clnstqhkd0j7ws601zmy5mrnl [Accessed: 25/02/2024]

Preface

This research work marks the conclusion of my five years at TU Delft, from the beginning of my bachelor's to the completion of my master's. Thanks to my mom sharing her interests with me, I was aware of my passion for physics long before finishing high school — specifically, space physics and engineering. TU Delft then nurtured and shaped this passion, helping me discover my love for space trajectory optimization and control algorithms. Consequently, a Master's thesis on finding optimal transfers within the intriguing and beautiful dynamics of the circular restricted three-body problem was ideal for me.

For this opportunity, I would like to express my deepest gratitude to Astos Solutions GmbH, particularly to my supervisor Maximilian Walther, and the company's managing director, Andreas Wiegand. To the former, I owe the success of this project, thanks to his constant support, sense of humor, patience, problem-solving skills, and remarkable availability. The latter taught me the intricacies of optimizing such complex use cases with the ASTOS software. Additionally, I am grateful to all the company's employees for their relentless willingness to help and their ability to make lunch breaks and free time so enjoyable, making me feel at home even in a foreign country.

I would also like to thank my TU Delft supervisor, Erwin Mooij, who was a key part of this project despite being thousands of kilometers away. His valuable mentorship in countless areas always felt thorough and close. His welcoming attitude and humor made this nine-month project much more enjoyable.

Lastly, I cannot forget my loved ones and friends, both from the Netherlands and Spain, who supported and believed in me throughout this journey. I am especially indebted to my parents, to my siblings, to Zeta, and to Víctor, all of whom probably grew tired of hearing about my struggles but were always there for me. This work is dedicated to you.

I hope you, as a reader, enjoy this report even one percent as much as I did in carrying it out, as it has been an incomparable learning journey.

Juan Bas Fernández
Delft, August 2024

Abstract

Due to the increasing interest of the aerospace industry and the scientific community in missions targeting halo orbits, a specific family of periodic solutions within the circular restricted three-body problem, it is highly desirable to find ways to reduce the cost of transfers between these orbits. To achieve this, depending on the mission characteristics, one could opt for reducing the required propellant mass or the time of flight. As such, with two objectives to be minimized, an ideal implementation for a mission designer would provide a Pareto front of feasible transfers instead of the single trajectory commonly obtained with conventional methods. Moreover, the necessary propellant mass can be minimized even further by means of electric low-thrust propulsion due to the significantly larger exhaust velocities. Therefore, the purpose of this research is to develop, implement, and study a suitable approach to obtain a collection of low-thrust transfers between halo orbits optimized in terms of both propellant mass and time of flight.

The approach employs an optimal control indirect method as thrust law which, combined with a heuristic optimizer based on differential evolution, can find a wide variety of trajectories that minimize the aforementioned objectives within the circular restricted three-body problem. Heuristic optimization is employed to remove the dependency of the solution on the provided initial guess and find trajectories in the region of the global minima. To satisfy the demanding boundary conditions characteristic of indirect methods, these constraints are included as a third objective for the optimizer to minimize as well. Due to the tolerance allowed on the constraints, the trajectories are subsequently refined with direct collocation methods. Then, they can be transitioned to a high-fidelity model, taking advantage of the versatility of direct methods. Furthermore, the implementation allows for the inclusion of invariant manifold phases arising from the departure and target orbits to obtain a wider set of Pareto-optimal solutions.

To assess the suitability of the proposed procedure, a specific transfer between two halo orbits around different Lagrange points of the Earth-Moon system was optimized. The results consisted of 100 Pareto-optimal transfers spanning more than 60 days, offering significant mission design freedom. Moreover, the Pareto front outperforms the trajectory found in the literature for a comparable use case by 30% in all objectives. Next, an optimized trajectory was first successfully verified with the mission analysis software ASTOS and then refined with direct collocation. The refined trajectory exhibited negligible changes in performance, rendering the trajectories obtained with this approach as promising initial guesses for further optimization with direct collocation methods. Moreover, several major perturbations not included in the simplified dynamic system were also corrected with direct collocation. The next steps include: accounting for the eccentricity of the Moon's orbit to fully transition the trajectories to a high-fidelity model, assessing the optimization quality with different use cases, and implementing transfers between different periodic solutions and dynamic systems, such as the Sun-Earth system.

Contents

List of Symbols

Latin Letters	Quantity	Symbol Unit
A_i	Sum of the response values for all experiments with factor A at level i	Case-dependent
A_iB_j	Sum of the response values for all experiments with factor A at level i and factor B at level j	Case-dependent
A_x	Halo orbit amplitude in the x-direction	[m]
A_y	Halo orbit amplitude in the y-direction	[m]
A_z	Halo orbit amplitude in the z-direction	[m]
C	Jacobi constant	[-]
C_{target}	Target Jacobi constant	[-]
$\boldsymbol{C}_x$	Right-handed rotation matrix around the x-axis	[-]
$\boldsymbol{C}_y$	Right-handed rotation matrix around the y-axis	[-]
$\boldsymbol{C}_z$	Right-handed rotation matrix around the z-axis	[-]
CF	Correction factor of the total sum of squares	Case-dependent
$\boldsymbol{c}$	Non-dimensional Coriolis acceleration vector in the CR3BP	[-]
DU	Distance Unit	[m]
e_M	Eccentricity of the Moon's orbit around the Earth	[-]
$\boldsymbol{F}(\boldsymbol{p})$	Generic multi-objective function	Case-dependent
F_e	Variance ratio of the error	[-]
F_i	Variance ratio of factor i	[-]
F_{ij}	Variance ratio of the interaction between factors i and j	[-]
FU	Force Unit	[N]
$\boldsymbol{f}$	Generic acceleration vector	[-]
$f(\boldsymbol{p})$	Generic objective function	Case-dependent
$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$	CR3BP equations of motion	[-]
$\boldsymbol{f}_{acc}$	Non-dimensional thrust acceleration vector	[-]
$f_{const}(\boldsymbol{p})$	Constraints objective function	[-]
f_e	Number of degrees of freedom of the error	[-]
f_i	Number of degrees of freedom of factor i	[-]
f_{ij}	Number of degrees of freedom of the interaction between factors i and j	[-]
f_T	Total number of degrees of freedom	[-]
G	Gravitational constant	[Nm^2/kg^2]
$\boldsymbol{g}$	Gradient of the pseudo-potential	[-]
g_0	Non-dimensional gravitational acceleration at Earth's surface	[-]
$g_{0_{dim}}$	Gravitational acceleration at Earth's surface	[m/s^2]
H	Hamiltonian function of an optimization problem	[-]
h	Hamiltonian of the CR3BP system	[-]
$\boldsymbol{h}$	Inequality constraints vector	[-]
h_c	Jacobi constant constraint function	[-]
h_{pos}	Position constraint function	[-]
h_{vel}	Velocity constraint function	[-]
I_{sp}	Non-dimensional specific impulse	[-]
$I_{sp_{dim}}$	Specific impulse	[s]

i	Imaginary number ($\sqrt{-1}$)	[-]
i_M	Inclination of the Moon's orbit around the Earth	[rad]
J	Performance index	[-]
J'	Augmented performance index	[-]
JD	Julian Date	[days]
k_i	Number of levels of factor i	[-]
L	Integral cost function of the performance index	[-]
$\boldsymbol{M}$	Monodromy matrix	[-]
MU	Mass Unit	[kg]
m	Non-dimensional mass	[-]
$\dot{m}$	Non-dimensional mass flow	[-]
m_0	Initial mass of the spacecraft	[kg]
m_1	Mass of the primary body	[kg]
m_2	Mass of the secondary body	[kg]
m_{dim}	Dimensional mass	[kg]
$\dot{m}_{dim}$	Dimensional mass flow	[kg/s]
m_p	Required propellant mass	[kg]
N	Dimension of the design space	[-]
N_d	Total number of discretization nodes	[-]
n	Dimension of the state space	[-]
n_{exp}	Total number of experiments	[-]
n_{int}	Number of experiments per interaction level	[-]
n_{lev}	Number of experiments per factor level	[-]
n_{const}	Number of constraints	[-]
P_e	Percentage of contribution to the variance of the error	[%]
P_i	Percentage of contribution to the variance of factor i	[%]
P_{ij}	Percentage of contribution to the variance of the interaction between factors i and j	[%]
$\boldsymbol{p}$	Design variable vector (optimizable parameters)	Various
$\boldsymbol{q}$	Equality constraints vector	[-]
$\boldsymbol{R}$	Rotation matrix	[-]
$\hat{\boldsymbol{R}}$	Radial unit vector	[-]
R^2	Coefficient of determination	[-]
R^2_{adj}	Adjusted coefficient of determination	[-]
$\boldsymbol{r}$	Non-dimensional position vector	[-]
$\boldsymbol{r}_1$	Non-dimensional position vector relative to the primary body	[-]
$\boldsymbol{r}_2$	Non-dimensional position vector relative to the secondary body	[-]
$\boldsymbol{r}_{EM}$	Position vector of the Moon relative to the Earth	[m]
$\boldsymbol{r}^*_f$	Target non-dimensional final position vector	[-]
$\boldsymbol{r}_{target}$	Non-dimensional position vector at a specific point in the target trajectory	[-]
S	Switching function	[-]
$\hat{\boldsymbol{S}}$	Along-track or transversal unit vector	[-]
SS_e	Error sum of squares	Case-dependent
SS_i	Sum of squares of factor i	Case-dependent
SS_{ij}	Sum of squares of the interaction between factors i and j	Case-dependent
SS_T	Total sum of squares	Case-dependent
SS'_e	Pure sum of squares of the error	Case-dependent

SS'_i	Pure sum of squares of factor i	Case-dependent
SS'_{ij}	Pure sum of squares of the interaction between factors i and j	Case-dependent
T	Orbital period/Interval of interest	[s]
T_{man}	TOF in the departure manifold phase	[days]
T_{man_2}	TOF in the target manifold phase	[days]
T_{max}	Maximum non-dimensional thrust of the spacecraft	[-]
$T_{max_{dim}}$	Maximum dimensional thrust of the spacecraft	[N]
TU	Time Unit	[s]
$\mathcal{U}$	Admissible region of control inputs	[-]
$\boldsymbol{u}$	Input vector	[-]
$\hat{\boldsymbol{u}}$	Thrust unit vector	[-]
V_e	Variance of the error	Case-dependent
V_i	Variance of factor i	Case-dependent
V_{ij}	Variance of the interaction between factors i and j	Case-dependent
VU	Velocity Unit	[m/s]
$\boldsymbol{v}$	Non-dimensional velocity vector	[-]
$\boldsymbol{v}_f^*$	Target non-dimensional final velocity vector	[-]
$\boldsymbol{v}_{target}$	Non-dimensional velocity vector at a specific point in the target trajectory	[-]
$\hat{\boldsymbol{W}}$	Cross-track or normal unit vector	[-]
w	Weighting parameter in constraint-handling methods	[-]
w_{pos}	Weight of both the position constraint and position state error	[-]
w_{vel}	Weight of both the velocity constraint and velocity state error	[-]
x	First non-dimensional position component	[-]
$\boldsymbol{x}$	Non-dimensional state vector	[-]
$\boldsymbol{x}*$	Target non-dimensional state vector	[-]
y	Second non-dimensional position component	[-]
y_i	Response value of experiment i	Case-dependent
y_{sum}	Total sum of the responses of all experiments	Case-dependent
z	Third non-dimensional position component	[-]

Greek Letters	**Quantity**	**Symbol Unit**
α	In-plane acceleration angle	[rad]
β	Out-of-plane acceleration angle	[rad]
$\boldsymbol{\Gamma}$	Jacobian with respect to the velocity of the acceleration vector in the CR3BP	[-]
ΔV	Spacecraft trajectory delta V	[m/s]
Δ_n	Step in initial state in the pseudo-arclength continuation method	[-]
ϵ	Mean obliquity of the ecliptic	[rad]
ϵ_p	Manifold generation perturbing factor	[-]
ϵ_s	State error	[-]
ζ	Normalized distance between the nominal value and the new bounds in CCD	[-]
θ_M	True anomaly of the Moon's orbit around the Earth	[rad]

$\boldsymbol{\lambda}$	Costate parameter vector	[-]
λ	Eigenvalue	[-]
λ_m	Costate parameter for the non-dimensional mass	[-]
$\boldsymbol{\lambda}_r$	Costate parameter vector associated with $\boldsymbol{r}$	[-]
λ_s	Stable eigenvalue of the monodromy matrix of an orbit	[-]
λ_u	Unstable eigenvalue of the monodromy matrix of an orbit	[-]
$\boldsymbol{\lambda}_v$	Costate parameter vector associated with $\boldsymbol{v}$	[-]
λ_{vx}	Costate parameter for the x-component of the velocity	[-]
λ_{vy}	Costate parameter for the y-component of the velocity	[-]
λ_{vz}	Costate parameter for the z-component of the velocity	[-]
λ_x	Costate parameter for the x-component of the position	[-]
λ_y	Costate parameter for the y-component of the position	[-]
λ_z	Costate parameter for the z-component of the position	[-]
μ	Mass normalization factor	[-]
$\boldsymbol{\nu}$	Lagrange multiplier vector of the boundary conditions	[-]
$\nu_{s/u}$	Stable/unstable eigenvector	[-]
Π	Set of allowable design variables	[-]
ρ_n	Step scaling factor in the pseudo-arclength continuation method	[-]
Σ	Poincaré Section	[-]
τ	Thrust ratio	[-]
τ_{orb}	Orbital period fraction of the periodic solution	[-]
υ	Stability index of a periodic solution in the CR3BP	[-]
$\boldsymbol{\Phi}$	State Transition Matrix	[-]
ϕ	Final cost function of the performance index	[-]
$\boldsymbol{\psi}$	Boundary conditions function	[-]
Ω	Pseudo-potential of the CR3BP	[-]
$\boldsymbol{\Omega}$	Jacobian with respect to the position of the acceleration vector in the CR3BP	[-]
Ω_M	Right ascension of the ascending node of the Moon's orbit	[rad]
ω_{EM}	Angular velocity of the Earth-Moon rotation around their barycenter	[rad/s]
ω_{xy}	In-plane oscillatory angular velocity	[rad/s]
ω_z	Out-of-plane oscillatory angular velocity	[rad/s]
ω_M	Argument of periapsis of the Moon's orbit around the Earth	[rad]

List of Abbreviations

2BP	Two-Body Problem
AMO	Adams-MOulton
ANN	Artificial Neural Network
ASTOS	Analysis, Simulation, and Trajectory Optimization Software
BCR4BP	Bi-Circular Restricted Four-Body Problem
BCR	Barycentric Rotating reference frame
CAMTOS	Collocation And Multiple Shooting Trajectory Optimization Software
CCD	Central Composite Design
CMA/ES	Covariance Matrix Adaptation Evolutionary Strategy
COV	Calculus Of Variations
CR3BP	Circular Restricted Three-Body Problem
DE	Differential Evolution
DLL	Dynamic-Link Library
DOE	Design Of Experiments
DOF	Degrees Of Freedom
DOPRI	DOrmand-PRInce
DRO	Distant Retrograde Orbit
ECI	Earth-Centered Inertial reference frame
EM	Earth-Moon
EML_1	First Lagrange point in the Earth-Moon CR3BP
EML_2	Second Lagrange point in the Earth-Moon CR3BP
EOM	Equations Of Motion
ESA	European Space Agency
FFD	Fractional Factorial Design
GACO	Genetic Ant Colony Optimization
GB	Gradient-Based
GNC	Guidance, Navigation, and Control
GRABUSTO	GRAgg-BUlirsch-STOer
GUI	Graphical User Interface
HIHA	HIgham-HAll
HJB	Hamilton-Jacobi-Bellman
IPOPT	Interior Point OPTimization
KKT	Karush-Kuhn-Tucker
MEE	Modified Equinoctial Elements
MHACO	Multi-objective Hypervolume-based Ant Colony Optimization
MIDACO	Mixed Integer Distributed Ant Colony Optimization
MO	Multi-Objective
MOEA/D	Multi-Objective Evolutionary Algorithm with Decomposition
NLP	Non-Linear Programming
NMS	Nelder-Mead Simplex
NRHO	Near-Rectilinear Halo Orbit
NSGA	Non-dominated Sorting Genetic Algorithm
OD	Orbit Determination
ODE	Ordinary Differential Equations

OGT	Orbit Generator Tool
OOP	Object-Oriented Programming
PMP	Pontryagin's Minimum Principle
PSO	Particle Swarm Optimization
RSW	Radial-Transversal-Normal reference frame
SA	Simulated Annealing
SE	Sun-Earth
SO	Single-Objective
SCP	Sequential Convex Programming
SNOPT	Sparse Non-linear OPTimizer
SQP	Sequential Quadratic Programming
STM	State Transition Matrix
TOF	Time Of Flight
TT	Terrestrial Time
TPBVP	Two-Point Boundary Value Problem
WFD	Work Flow Diagram
WBS	Work Breakdown Structure
WORHP	We Optimize Really Huge Problems optimizer

Introduction 1

In this chapter, an introduction to this research is provided. It starts in Section 1.1 with an explanation on what motivates this investigation and the scientific gap to be filled, and ends in Section 1.2 with a description of the research questions to be answered. This last section also includes the structure of the remaining report chapters.

1.1 Motivation and Scientific Gap

Over the past years, the aerospace industry has developed a great interest in missions targeting a special type of orbit: the halo family of orbits. These differ from the conventional Keplerian orbits and only appear within the so-called Circular Restricted Three-Body Problem (CR3BP). The CR3BP is a dynamical model in which two massive bodies, such as the Earth and the Moon, revolve around their common barycenter in circular orbits, exerting a gravitational attraction onto a third body (often the spacecraft) of negligible mass. Such a system can approximate spacecraft trajectories in numerous places across the Solar System, as its bodies usually orbit each other in approximately circular orbits. Well-known examples are the Earth-Moon (EM) system, the Sun-Earth (SE) system, the Sun-Jupiter system, and even the Jupiter-Ganymede system (Q. Li et al., 2022). The CR3BP presents five equilibrium points in which the spacecraft would stay in a fixed relative position in relation to the two massive bodies. From these points, known as the *Lagrange libration points*, multiple families of periodic solutions arise. Additionally, from these solutions, invariant manifold structures emerge, which serve as pathways to enter and leave these orbits with virtually no effort.

Periodic solutions around the Lagrange points, such as halo orbits, are interesting for future missions as they can maintain an almost constant location with respect to the Earth and the Moon, facilitating telecommunications as well as improving the scientific output by measuring consistently in a set place. Missions such as ISEE-3[1] or Adytia-L1[2] took advantage of this fact by placing their spacecraft in a halo orbit around the L_1 Lagrange point of the Sun-Earth system (SEL_1) to take consistent measurements of the Sun by always staying between the Sun and the Earth at a set distance from both. Moreover, the famous James Webb Space Telescope[3] is currently in a halo orbit around SEL_2, thus always having the same lighting conditions.

Halo orbits or other periodic solutions in the Earth-Moon system are also of interest especially for future space stations around the Moon (Kokou et al., 2014), like, for instance, the proposed Deep Space Gateway[4]. These stations can serve as gateways for other locations such as the Moon or Mars due to the available low-cost trajectories from and to these orbits. As an example, a set of CubeSats could be sent together to a specific space station, and then, each of them would travel independently to the desired location around the Moon with greatly reduced requirements in their propulsion systems. In addition, they can serve as continuous communication links to the far side of the Moon, which may well be useful in the future. Other applications arise from the fact that some specific halo orbits can ensure lunar south pole coverage (Ozimek and Howell, 2010).

Furthermore, as explained by Sentinella and Casalino (2006), the success of the Smart-1[5] mission to the Moon and the Deep-Space-1[6] mission to the asteroid belt, proved solar-electric propulsion

[1]URL: https://science.nasa.gov/mission/isee-3-ice/ [Accessed: 16/01/2024]
[2]URL: https://www.isro.gov.in/Aditya_L1.html [Accessed: 16/01/2024]
[3]URL: https://webb.nasa.gov/content/about/orbit.html [Accessed: 16/01/2024]
[4]URL: https://www.nasa.gov/mission/gateway/ [Accessed: 16/01/2024]
[5]URL https://www.esa.int/Enabling_Support/Operations/SMART-1 [Accessed: 31/01/2024]
[6]URL: https://science.nasa.gov/mission/deep-space-1/ [Accessed: 31/01/2024]

to be suitable to explore the solar system. A more recent example would be ESA's Bepi Colombo[7] mission, which is currently on its way to Mercury. Since electric low-thrust engines present a higher specific impulse compared to chemical propulsion, low-thrust trajectories have gained extreme interest over the past years to reduce the required propellant mass, thus triggering the development of better and more refined trajectory optimization methods. An example of a low-thrust mission targeting Earth-Moon halo orbits is the ARTEMIS (Acceleration, Reconnection, Turbulence, and Electrodynamics of the Moon's Interaction with the Sun)[8] mission. Two satellites, originally part of the THEMIS mission, transitioned from their respective Earth orbits to distinct halo orbits around the first and second Lagrange points, making use of optimal control methods to design the trajectories. However, no low-thrust mission has ever shifted between halo orbits, highlighting the cutting-edge nature of this research.

The most successful and commonly used methods for low-thrust trajectory design are the robust direct collocation methods, within the branch of optimal control. These optimizations are solved with local, gradient-based optimizers because of their outstanding convergence capabilities and ability to effectively handle the thousands of design variables required for collocation methods. Nonetheless, they show a high dependence on the provided initial guess, which must be a complete trajectory, usually a rough and infeasible one. As such, they tend to converge to suboptimal minima, making it an arduous task to find the global optimum, with the obtainment of promising solutions dependent on the astrodynamicist's expertise. Heuristic optimizers, such as genetic algorithms, do not require an initial guess, and they offer a much wider exploration of the design space, increasing the odds of finding trajectories in the region of the global minimum. However, they possess poor numerical accuracy, and their convergence capabilities are greatly influenced by the selected constraint-handling method (Sentinella and Casalino, 2006). Therefore, a sensible approach would be to use heuristic optimization to obtain preliminary solutions, hopefully in the vicinity of the global optimum, in combination with a direct optimization method that refines these solutions until a minimum is reached and transitions these trajectories to a high-fidelity model. The use of heuristic optimizers to obtain optimal low-thrust trajectories within the CR3BP is still in its early phases, especially when combining it with multi-objective optimization. Moreover, assessing the benefits of using these trajectories as initial guess for direct collocation methods is a matter in which there is still much that remains to be investigated.

Another disadvantage of direct methods with local optimizers is that they only provide a single trajectory. Thus, if a different performance is desired, continuation methods must be applied starting from this probably suboptimal trajectory. Furthermore, as the mission cost can be reduced (or the mission performance enhanced) by decreasing the required propellant mass or the mission flight time, it would be beneficial to minimize both, as suggested by Lee et al. (2005). As these two figures of merit are competing parameters in trajectory optimization problems with coasting phases, there is not one, but a set of equally optimal minima (a so-called Pareto front). Considering that the choice of a certain time of flight is determined by characteristics specific to each mission, it would be interesting to provide a range of minimum propellant mass trajectories as a function of the time of flight, i.e., the Pareto front, such that any of them is a suitable initial guess to obtain a refined trajectory near the requested time of flight (TOF). Henceforth, with all the previous information, the purpose of this research is to develop, implement, and study a suitable method to generate Pareto-optimal low-thrust transfers between periodic solutions within the CR3BP. To achieve said objective, the developed strategy will be implemented within the *Orbit Generator Tool* (OGT) (Walther and Wiegand, 2023) from Astos Solutions GmbH.

[7]URL: https://www.esa.int/Science_Exploration/Space_Science/BepiColombo_overview2 [Accessed: 31/01/2024]
[8]URL: https://science.nasa.gov/mission/themis-artemis/ [Accessed: 09/08/2024]

1.2 Research Objectives

This M.Sc. thesis aims to contribute to the scientific community by providing an answer to a specific research question. From the explanations provided in Section 1.1, the main research question for this work can be derived:

> **Main Research Question**
>
> *What would be a suitable approach to obtain a variety of low-propellant-mass low-thrust transfers between periodic solutions in the cislunar space as a function of the TOF?*

As already explained, the idea is to generate a set of Pareto-optimal trajectories with respect to the TOF and the propellant mass within the CR3BP such that a user without extensive experience in mission trajectory design and astrodynamics can obtain useful initial guesses at desired times of flight to use with conventional trajectory optimization methods (e.g., direct collocation). The obtained solutions are meant to provide a preliminary thrust strategy to design real-life space missions in the cislunar space. From this research question, several sub-questions arise to aid in answering it:

Sub-Question 1

To what extent is the outcome of the implemented method verified?

Firstly, the implemented method will only be suitable as long as the obtained trajectories are consistent with the mathematical models used in the space scientific community. Knowing what the accuracy of the solutions is can help identify the range of applications of the method and its usefulness.

Sub-Question 2

What is the quality of the solutions obtained?

Secondly, the suitability of the approach is also measured by the performance of the trajectories compared to the ones obtained by other authors with different methods. Poor performance leads to questioning whether the solutions are promising initial guesses in every context, even if they are very effective initial guesses for a given situation.

Sub-Question 3

What is the performance of the solutions when used as initial guess for direct collocation methods?

Thirdly, the obtained trajectories are intended to serve as promising initial guesses for further optimization with direct collocation methods, with the goal of outperforming the trajectories that would be obtained after optimizing with the commonly employed rough and fast initial guesses. In addition, if without excessive effort the performance of the optimized trajectories corresponds to a great extent to the performance of the trajectories in the initial Pareto-front, then the proposed initial guesses can be considered effective since the mission designer can easily find promising trajectories at very different times of flight to suit the mission.

Sub-Question 4

How accurate is the CR3BP approximation to obtain preliminary trajectories to design real-life missions in the cislunar space?

Fourthly, the method presented in this report is derived purely within the context of the CR3BP. Hence, several steps, such as a sensitivity analysis, ought to be carried out to assess the difficulty

in transitioning the trajectories to a high-fidelity model and their usefulness when designing a real mission.

Sub-Question 5

To what extent can the use of manifold theory improve the obtained solutions?

Fifthly, the literature supports the idea that the use of manifold theory enables the obtainment of trajectories that require a lower propellant mass. It would be of value to quantify how much benefit the use of manifold theory compared to a direct transfer provides.

The report continues in Chapter 2 with a detailed elucidation of the required theoretical background on the CR3BP for the method presented. Next, the existing functionalities and capabilities of the *Orbit Generator Tool* are discussed in Chapter 3, followed by a trade-off between the available methods to solve the problem at hand given in Chapter 4. Chapter 5 explains the chosen method to solve the main research question, including its implementation within the tool. Subsequently, in Chapter 6 the use cases under study are presented, and the problem is analyzed in detail, whereas Chapter 7 presents the results of the optimizations carried out. The resulting optimized trajectories are analyzed in Chapter 8 and the report ends with the conclusions and recommendations in Chapter 9. Lastly, the research planning can be found in Appendix A.

Background Theory on the CR3BP 2

This work takes place in the context of the Circular Restricted Three-Body Problem (CR3BP), in which the convenient halo orbits arise. Firstly, the problem at hand and its equations will be explained in Section 2.1 and Section 2.3, making use of the reference frames defined in Section 2.2. Then, the existence of Lagrange Points and periodic solutions around them will be discussed in Section 2.4 and Section 2.5, respectively. Lastly, Section 2.6 will explore the generation and use of manifolds originating from these periodic solutions.

2.1 History

As explained by Wakker (2015), after Newton (1642-1727) formulated the general gravitational equations of motion and derived the analytical solutions for the Two-Body Problem (2BP), many astronomers, such as Euler (1707-1783), Lagrange (1736-1813), Laplace (1749-1827), Jacobi (1804-1851), Hamilton (1805-1865), Poincaré (1854-1912) and Birkhoff (1884-1944) attempted to solve the Three-Body Problem, in which the gravitational attraction of the three bodies is the only force. This problem was of interest to predict the movement of the Moon in the Sun-Earth environment, for instance. It is still of great relevance for modern-day astrodynamics to predict the behavior of a spacecraft in the cislunar regime. However, it exhibits a chaotic behavior and analytical solutions could only be found under certain symmetry conditions.

The problem was then simplified under a specific set of assumptions. The problem then became "restricted", as the mass of the third body was assumed to be negligible compared to the other two, allowing these two to always move in the same plane, with their movement not being affected by the third mass. Moreover, the first two masses were assumed to move in circular orbits with constant angular velocities about their common barycenter. These two assumptions lead to the CR3BP, reducing the system from nine second-order equations of motion (order 18) to three (order six), as the motion of the first two masses is always known. However, even though the motion is accurately described, as the mass of the third body is not strictly zero, conservation laws for energy and angular momentum do not hold.

The CR3BP configuration is interesting since it can be found in numerous places in the Solar System as the planets move in almost circular orbits around the Sun. Also, the eccentricity of the Moon's orbit around the Earth is almost zero. Euler, Lagrange, Jacobi, Hill (1828-1914), Poincaré, and others contributed considerably to this problem, which is still far from being solved. The combination of the equations of two CR3BP systems led to the creation of the Bi-Circular Restricted Four-Body Problem (BCR4BP) (Maisch, 2022), which is out of the scope of this thesis.

2.2 Reference Frames and Transformations

The equations of motion (EOM) of the CR3BP are derived in a specific non-inertial reference frame that needs to be defined. Moreover, it is interesting to visualize the trajectories in a more intuitive inertial frame centered around the Earth. Lastly, the thrust profiles are usually provided as attitude commands to the spacecraft, due to the thrusters commonly being fixed to the spacecraft bus. As such, a third reference frame is utilized.

2.2.1 Earth-Centered Inertial Reference Frame

The Earth-Centered inertial (ECI) reference frame is defined with its origin placed at the Earth's center of mass. The x-axis, with $\hat{\boldsymbol{I}}$ as its unit vector[1], points towards the intersection of the Earth's

[1] Unit vectors are always denoted with "^".

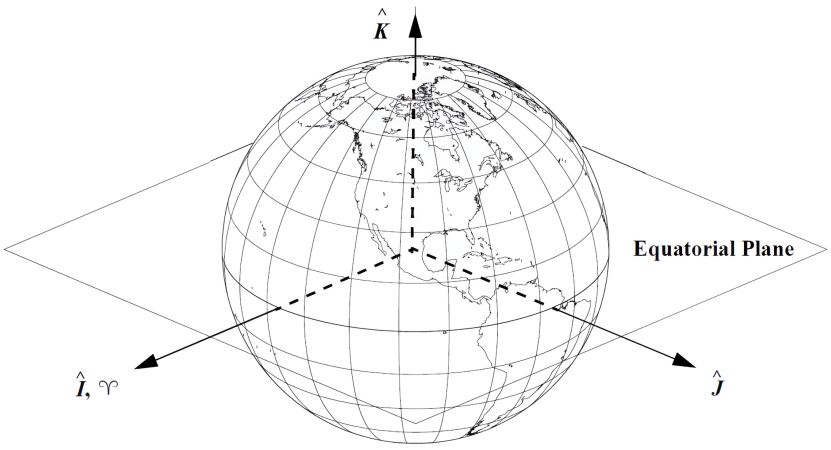

Figure 2.1: Diagram of the ECI reference frame (Vallado, 2013).

equatorial plane with the ecliptic (plane of the Earth's orbit around the Sun) at the epoch J2000. This date corresponds to the 1^{st} of January 2000, at noon in the time zone of the Greenwich meridian (JD = 2451545.0 days). This intersection corresponds to the direction of the Vernal Equinox or First Point of Aries (♈). Then, the z-axis ($\hat{\boldsymbol{K}}$) points towards the Earth's geographic north pole. Lastly, the y-axis ($\hat{\boldsymbol{J}}$) completes the right-handed system. Figure 2.1 depicts the aforementioned coordinates with respect to the Earth and the equatorial plane.

2.2.2 Barycentric Rotating Reference Frame

Under the assumptions of zero gravitational pull from the third mass and circular motion of the primary (the one with the largest mass) and secondary bodies[2] around their barycenter, a rotating reference frame centered around said barycenter can be defined. This frame was found convenient for expressing the equation of motion of the problem.

In the Barycentric Rotating reference frame (BCR), the x-axis points towards the Moon, rotating with it at all times. Then, the z-axis is perpendicular to the plane containing the orbits of the Earth and the Moon around their common barycenter. Lastly, the y-axis again completes the right-handed system.

2.2.3 Normalization of the BCR Frame

It is common to normalize the CR3BP equations and make them non-dimensional for the sake of computational accuracy and efficiency. Hence, the BCR frame ought to be normalized first. The chosen normalization is the one presented by Wakker (2015), which is by far the most common. Firstly, the masses of the primary, m_1, and the secondary, m_2, are normalized with respect to their sum, yielding the mass normalization factor, μ:

$$\mu = \frac{m_2}{m_1 + m_2} \implies 1 - \mu = \frac{m_1}{m_1 + m_2} \tag{2.1}$$

Then, the distance between the primaries (primary and secondary bodies), r_{EM}, is chosen as the unit of distance, DU, and the inverse of the angular velocity of the Earth and the Moon around their barycenter, ω_{EM}, becomes the unit of time, TU:

$$DU = r_{EM} \qquad\qquad TU = \frac{1}{\omega_{EM}} \tag{2.2}$$

[2] In this research, the Earth is the primary body and the Moon the secondary body.

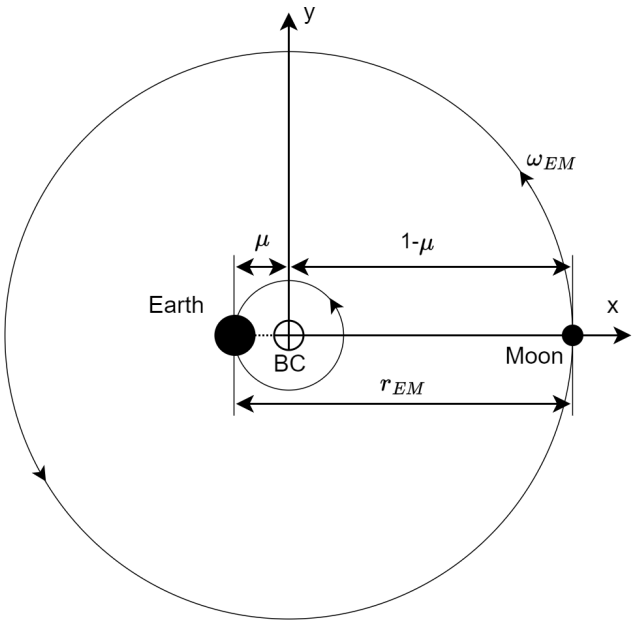

Figure 2.2: Diagram of the barycentric rotating reference frame.

The x-location of the primaries relative to the barycenter (x_1 for the primary and x_2 for the secondary) follows from their normalized distance and the barycenter equation:

$$x_1 + x_2 = 1 \qquad\qquad x_1(\mu - 1) + x_2\mu = 0$$

yielding the following position vectors for the Earth, $\boldsymbol{r}_E$, and the Moon, $\boldsymbol{r}_M$:

$$\boldsymbol{r}_E = \begin{pmatrix} -\mu & 0 & 0 \end{pmatrix}^T \qquad\qquad \boldsymbol{r}_M = \begin{pmatrix} 1-\mu & 0 & 0 \end{pmatrix}^T \tag{2.3}$$

acknowledging that the primaries and the barycenter are by definition aligned at all times. Next, to obtain ω_{EM}, it should be noted that in this configuration the angular velocity around the barycenter is equivalent to the one of the Moon around the Earth. Hence, being a an arbitrary vector, and with the following notation

$$||\boldsymbol{a}|| = a$$

the inertial equations of the relative motion between the Earth and the Moon, can be expressed as

$$\ddot{\boldsymbol{r}}_{EM} = -G\frac{m_1 + m_2}{r_{EM}^3}\boldsymbol{r}_{EM}$$

with G being the gravitational constant. Through the Two-Body Problem (2BP) analytical solution equations, one obtains (Aziz, 2018)

$$\omega_{EM} = \sqrt{G\frac{m_1 + m_2}{r_{EM}^3}} \tag{2.4}$$

A diagram of the described non-dimensional rotating system can be found in Figure 2.2 (not to scale), showcasing the location of the primaries (the Earth and the Moon in this case), their orbits, their distance, r_{EM}, and the barycenter (BC). Lastly, Table 2.1 displays the values used to define

Table 2.1: EM CR3BP defining parameters.

Parameter	Value	Unit
Gm_1	398600.435507	km^3/s^2
Gm_2	4902.800118	km^3/s^2
r_{EM}	384400.0	km
ω_{EM}	13.194253	°/day

the Earth-Moon system parameters, with the first three values given by Park et al. (2021). These parameters lead to a DU, TU, and unit of velocity, $VU = \frac{DU}{TU}$, of:

$$DU = 384400.0 \text{ km} \qquad TU = 375190.2703 \text{ s} \qquad VU = 1024.546824 \text{ m/s} \tag{2.5}$$

2.2.4 Radial-Transversal-Normal Reference Frame

The radial-transversal-normal frame, commonly referred to as the RSW frame or the local-vertical, local-horizontal frame, is centered at the spacecraft center of mass. Its unit vectors are defined by the spacecraft's position and velocity vectors relative to a celestial body. As such, depending on the chosen body, the frame's definition will vary. For this research, it is decided to employ the RSW frame defined with respect to the Earth (the primary body). Then, the *radial* unit vector, $\hat{\boldsymbol{R}}$, corresponds to the x-axis and points in the direction of the spacecraft's position relative to the primary's center of mass. The *transversal* or *along-track* unit vector, $\hat{\boldsymbol{S}}$, corresponds to the y-axis and lies in the plane defined by $\hat{\boldsymbol{R}}$ and the velocity vector, at 90° from $\hat{\boldsymbol{R}}$. Finally, the *normal* or *cross-track* unit vector, $\hat{\boldsymbol{W}}$, completes the right-handed coordinate system.

The explained frame can be visualized in Figure 2.3, where the aforementioned unit vectors are showcased, and the ECI coordinates and a generic spacecraft trajectory are included for reference. The diagram also includes a generic acceleration, $\boldsymbol{f}$, and shows how it can be related to the RSW unit vectors by means of two angles, α and β. The former is the *in-plane* angle, measured from $\hat{\boldsymbol{S}}$ within the $\hat{\boldsymbol{R}}\hat{\boldsymbol{S}}$-plane and positive in the direction of $\hat{\boldsymbol{W}}$. The latter corresponds to the *out-of-plane* angle, positive when the deflection is in the direction of $\hat{\boldsymbol{W}}$.

2.2.5 BCR to ECI Transformation

The procedure for switching between the frames is provided by Walther (2022) and is explained in this subsection. Defining the (non-dimensional) position $\boldsymbol{r}$ and (non-dimensional) velocity $\boldsymbol{v}$ of the spacecraft in the BCR frame with the following notation:

$$\boldsymbol{r} = \begin{pmatrix} x & y & z \end{pmatrix}^T \qquad \boldsymbol{v} = \dot{\boldsymbol{r}} = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T \tag{2.6}$$

the vectors can be then translated to an auxiliary rotating frame, denoted with " $'$ ", centered around the Earth (the primary) instead of around the barycenter. The inertial velocity is considered in this frame, hence, the rotation of the BCR frame ought to be taken into account:

$$\boldsymbol{r}' = \boldsymbol{r} - \boldsymbol{r}_E = \begin{pmatrix} x + \mu & y & z \end{pmatrix}^T \tag{2.7}$$

$$\boldsymbol{v}' = \boldsymbol{v} + \omega_{EM} \times \boldsymbol{r}' = \begin{pmatrix} v_x - \omega_{EM} y & v_y + \omega_{EM}(x + \mu) & v_z \end{pmatrix}^T \tag{2.8}$$

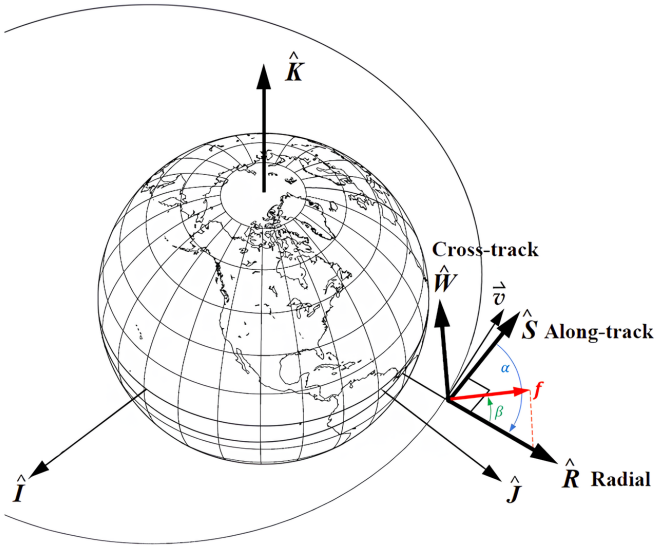

Figure 2.3: Diagram of the RSW reference frame, including the in-plane and out-of-plane angles, α and β, provided by Walther (2022) and based on Vallado (2013).

The last step is to rotate this auxiliary frame to the desired ECI frame by means of rotation matrices. Defining the right-handed rotations, $\boldsymbol{C}_x$, $\boldsymbol{C}_y$, and $\boldsymbol{C}_z$, by an arbitrary angle α respectively around the x, y, and z-axes as

$$\boldsymbol{C}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \quad \boldsymbol{C}_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad \boldsymbol{C}_z(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

the full rotation matrix between the frames can be obtained by successive rotations around specific axes. For this, the Moon's Kepler elements around the Earth with respect to the ecliptic plane can be used: the true anomaly θ_M, the argument of periapsis ω_M, the right ascension of the ascending node Ω_M, and the inclination i_M. Lastly, a rotation around the resulting z-axis is required to translate from the ecliptic plane to the equator plane. For this, the mean obliquity of the ecliptic, ϵ, is required. As such, the transformation matrix between the auxiliary frame and the ECI frame, $\boldsymbol{R}^{ECI/'}$, is obtained by:

$$\boldsymbol{R}^{ECI/'} = \boldsymbol{C}_x(\epsilon)\boldsymbol{C}_z(\Omega_M)\boldsymbol{C}_x(i_M)\boldsymbol{C}_z(\omega_M + \theta_M) \tag{2.10}$$

Hence, the position and velocity vectors in the ECI frame, $\boldsymbol{r}_{ECI}$ and $\boldsymbol{v}_{ECI}$, can be computed as:

$$\boldsymbol{r}_{ECI} = \boldsymbol{R}^{ECI/'}\boldsymbol{r}' \qquad \boldsymbol{v}_{ECI} = \boldsymbol{R}^{ECI/'}\boldsymbol{v}' \tag{2.11}$$

It should be noted that the resulting vectors are still non-dimensional. The values for all these parameters are taken from the United States Naval Observatory et al. (2008). Some of them use polynomial approximations as a function of the initial epoch t_0 in days since December 31, 2007, at 0:00.00, again in Terrestrial Time (TT). The eccentricity of the Moon's orbit is, of course, forced to be zero due to the assumptions of the CR3BP. With JD being the Julian Date, the aforementioned values are

$$\begin{aligned}
\epsilon &= 0.409092808 \text{ rad} \\
e_M &= 0.0^\circ \\
i_M &= 5.1453964^\circ \\
\Omega_M &= -0.05295376t_0 + 330.393098^\circ \\
\omega_M &= 0.16435724t_0 + 78.314065^\circ \\
\theta_M &= 13.06499299t_0 + 131.275374^\circ \\
t_0 &= JD - 2454465.5 \text{ days}
\end{aligned} \tag{2.12}$$

Although the considered approximations are meant to only be used in 2008, Walther (2022) explains that even in 2030 the prediction is accurate enough for the purpose of presenting the results of the method, considering that the trajectory is computed in the time-invariant CR3BP. However, depending on the application of the obtained solution, a better estimation may be needed when translating the trajectory to the ECI frame.

2.2.6 BCR to RSW Transformation

The algorithm is retrieved from the work by Walther (2022). The first step is to define the RSW unit vectors within the auxiliary geocentric rotating frame with inertial velocity from Subsection 2.2.5, which includes the transformation between the BCR frame and this auxiliary frame:

$$\begin{aligned}
\hat{\boldsymbol{R}}' &= \frac{\boldsymbol{r}'}{||\boldsymbol{r}'||} \\
\hat{\boldsymbol{S}}' &= \frac{(\boldsymbol{r}' \times \boldsymbol{v}') \times \boldsymbol{r}'}{||\,(\boldsymbol{r}' \times \boldsymbol{v}') \times \boldsymbol{r}'||} \\
\hat{\boldsymbol{W}}' &= \frac{\boldsymbol{r}' \times \boldsymbol{v}'}{||\boldsymbol{r}' \times \boldsymbol{v}'||}
\end{aligned} \tag{2.13}$$

Figure 2.3 can help visualize these definitions. Then, the acceleration vector in the RSW frame, $\boldsymbol{f}_{RSW} = \begin{pmatrix} f_R & f_S & f_W \end{pmatrix}^T$, (in this case the thrust) can be transformed to the BCR frame by:

$$\boldsymbol{f}' = \boldsymbol{f} = f_R\hat{\boldsymbol{R}}' + f_S\hat{\boldsymbol{S}}' + f_W\hat{\boldsymbol{W}}' = \begin{bmatrix} \hat{\boldsymbol{R}}' & \hat{\boldsymbol{S}}' & \hat{\boldsymbol{W}}' \end{bmatrix} \boldsymbol{f}_{RSW} \tag{2.14}$$

noting that an acceleration vector would be identical in both the auxiliary ($\boldsymbol{f}'$) and BCR ($\boldsymbol{f}$) frames.

Lastly, as the required transformation direction is the opposite of the one presented, by pre-multiplying by the inverse of the matrix in Equation (2.14), the acceleration vector in the auxiliary frame can be transformed to the RSW frame through:

$$\boldsymbol{f}_{RSW} = \begin{bmatrix} \hat{\boldsymbol{R}}' & \hat{\boldsymbol{S}}' & \hat{\boldsymbol{W}}' \end{bmatrix}^{-1} \boldsymbol{f} \tag{2.15}$$

2.3 Equations of Motion

Now that the BCR frame has been defined, the equation of motion can be obtained in non-dimensional form. The complete derivation is shown by Wakker (2015). Firstly, the spacecraft position relative to the primary and secondary bodies, respectively, is defined by

$$\boldsymbol{r}_1 = \begin{pmatrix} x+\mu & y & z \end{pmatrix}^T \qquad \boldsymbol{r}_2 = \begin{pmatrix} x-(1-\mu) & y & z \end{pmatrix}^T \tag{2.16}$$

Then, the so-called pseudo-potential Ω of the system can be defined as

$$\Omega = \frac{1}{2}(x^2+y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2.17}$$

This conservative, non-central force field accounts for the gravitational and centrifugal accelerations and not for the Coriolis acceleration present in rotating reference frames. The complete equations of motion, including arbitrary acceleration terms, f_i, in each axis that represent the thrust acceleration are then (Aziz, 2018)

$$\begin{aligned} \ddot{x} - 2\dot{y} &= \frac{\partial \Omega}{\partial x} + f_x \\ \ddot{y} + 2\dot{x} &= \frac{\partial \Omega}{\partial y} + f_y \\ \ddot{z} &= \frac{\partial \Omega}{\partial z} + f_z \end{aligned} \tag{2.18}$$

The acceleration terms must be of course non-dimensional. As such, the thrust and the mass must be normalized. As they are decoupled from the CR3BP accelerations, they can be scaled independently. Thus, the following mass unit MU and force unit FU are selected:

$$MU = m_0 \qquad FU = MU\frac{DU}{TU^2}, \tag{2.19}$$

where m_0 is the dimensional initial mass of the spacecraft. Hence, with $\hat{\boldsymbol{u}} \in [0,1]^3$ being the thrust attitude unit vector, $\tau \in [0,1]$ the thrust ratio, and T_{max} the maximum non-dimensional thrust of the spacecraft[3], the thrust acceleration vector, $\boldsymbol{f}_{acc}$, is obtained via:

$$\boldsymbol{f}_{acc} = \frac{T_{max_{dim}}\tau}{m_{dim}}\hat{\boldsymbol{u}}\frac{MU}{FU} = \frac{T_{max}\tau}{m}\hat{\boldsymbol{u}} \tag{2.20}$$

Then, as the (non-dimensional) mass varies over time to generate a thrust acceleration, an additional EOM must be satisfied:

$$\dot{m} = -\frac{T_{max_{dim}}\tau}{I_{sp_{dim}} g_{0_{dim}}}\frac{TU}{MU} = -\frac{T_{max}\tau}{I_{sp} g_0} \tag{2.21}$$

where $\dot{m}$ is the non-dimensional mass flow, $g_{0_{dim}}$ is the dimensional gravitational acceleration at the Earth's surface (9.81 m/s^2) and $I_{sp_{dim}}$ is the dimensional specific impulse (with g_0 and I_{sp} being the non-dimensional variables, respectively).

When excluding the additional forcing term from the equations (and thus Equation (2.21)), a single analytical integral of motion can be derived, the so-called Jacobi constant C, directly related to the Hamiltonian h of the system (equal to the sum of the kinetic and potential energy since the EOM are time-independent and the potential energy is velocity-independent (Cline, 2017)) by:

$$C = 2\Omega - v^2 = -2h \tag{2.22}$$

As predicted in Section 2.1, if this expression is translated to an inertial reference frame, the total energy of the third body (the spacecraft) is not conserved. However, the sum of the total energy and the angular momentum around the z-axis does remain constant, coinciding with h.

[3] In these equations, the subscript "*dim*" denotes a dimensional quantity.

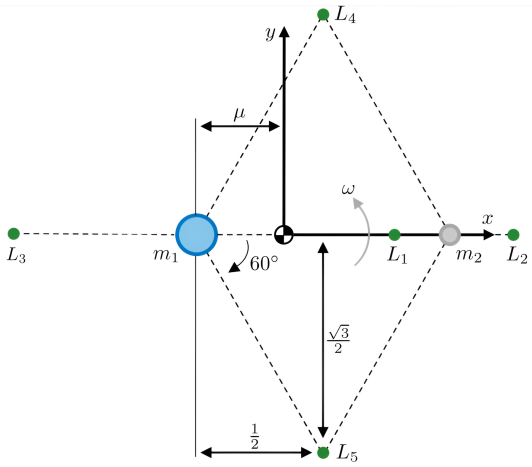

Figure 2.4: Diagram of the Lagrange libration points in an arbitrary CR3BP (Eslinger, 2023)

2.4 Lagrange Libration Points

In the CR3BP there are a set of equilibrium points for which the partial derivatives of the pseudo-potential with respect to the spatial coordinates is zero:

$$\frac{\partial \Omega}{\partial x} = \frac{\partial \Omega}{\partial y} = \frac{\partial \Omega}{\partial z} = 0 \tag{2.23}$$

If both the velocity in the BCR frame and the $\boldsymbol{f}_{acc}$ acceleration vector are also zero, then the acceleration experienced by the spacecraft will also be zero. This means that the spacecraft will rotate with the Moon around the Earth and maintain the same relative position with respect to them. To obtain these points, the following equations thus ought to be solved:

$$\begin{aligned} x - \frac{1-x}{r_1^3}(\mu + x) + \frac{\mu}{r_2^3}(1 - \mu - x) &= 0 \\ y\left(1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3}\right) &= 0 \\ z\left(\frac{1-\mu}{r_1^3} + \frac{\mu}{r_2^3}\right) = 0 \implies z &= 0 \end{aligned} \tag{2.24}$$

The third equation restricts all solutions to lie in the xy-plane. Then, if the y-coordinate is forced to be zero, the first equation (solved numerically or with a series approximation) yields the three collinear Lagrange points (L_1, L_2, and L_3). On the other hand, if y is strictly non-zero, the first two equations yield

$$r_1 = r_2 = 1 \qquad x = \frac{1}{2} - \mu \qquad y = \pm\frac{1}{2}\sqrt{3}, \tag{2.25}$$

meaning that the fourth and fifth points (L_4 and L_5) form an equilateral triangle with the primaries. Figure 2.4 illustrates the location of the five points within the context of the BCR frame. The location of the primaries is also included in the diagram. As explained by Wakker (2015), if $\mu \leq 0.0385$ (which is usually the case in the Solar System CR3BPs), L_4 and L_5 are stable points. However, L_1, L_2, and L_3 are always unstable.

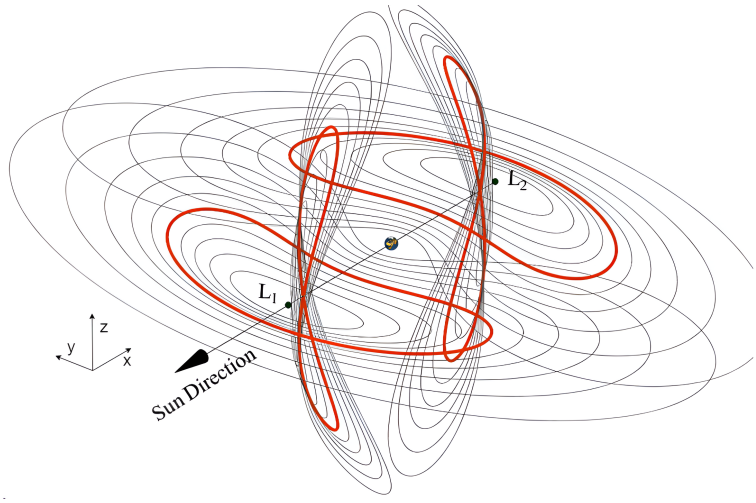

Figure 2.5: Horizontal and vertical Lyapunov orbits around L_1 and L_2 of the SE system (García Yárnoz et al., 2013).

2.5 Periodic Solutions

As explained in Chapter 1, periodic solutions around these Lagrange points are interesting for the future of space exploration. This chapter will focus on some of the periodic solutions around L_1 and L_2, as they are located close to the secondary body, being thus the most useful. In addition, when linearizing the equations of motion around these two points (Eslinger, 2023), the six obtained eigenvalues of the continuous linear system, which determine the stability of the motion (Olsder et al., 2011), are of the form $\pm\lambda, \pm i\omega_{xy}, \pm i\omega_z$. This is a dynamic behavior of the type *saddle-center-center* (Langemeijer, 2018). The real eigenvalues give rise to invariant hyperbolic stable $(-\lambda)$ and unstable $(+\lambda)$ manifolds, which offer extremely cheap routes to enter and leave these points. This does not occur with L_4 and L_5, making the first two Lagrange points more attractive. Section 2.6 will elaborate more on the invariant manifolds and show that the periodic solutions themselves also have associated (un)stable manifolds. It should be noted that quasi-periodic solutions also exist, but they are out of the scope of this study.

2.5.1 Lyapunov Orbits

The in-plane oscillatory motion determined by the $\pm i\omega_{xy}$ eigenvalue pair hints at the existence of periodic solutions around L_1 and L_2 that oscillate strictly within the xy-plane. Even though the linearized equations are only accurate in the vicinity of the equilibrium points, the so-called horizontal Lyapunov family of orbits can be found using the non-linear equations of motion. The horizontal Lyapunov families around L_1 and L_2 of the Sun-Earth system are depicted in Figure 2.5, corresponding to the kidney-shaped orbits within the orbital plane of the Earth. The figure shows that this family extends far away from the Lagrange points. They are symmetric with respect to the xz-plane.

In contrast, the $\pm i\omega_z$ eigenvalue pair yields the vertical Lyapunov family of orbits. These eight-shaped orbits can also be seen in Figure 2.5. These always pass very close to the equilibrium point when crossing the xy-plane, which happens twice per orbit, and are symmetric with respect to this plane and the xz-plane.

2.5.2 Halo Orbits

As explained by Marsden (1978), Langemeijer (2018), and Q. Li et al. (2022), a certain periodic solution of a family can coincide with a periodic solution of a different family for a specific value of a parameter p (e.g., the period or the z-amplitude). For the rest of the values of the parameter, the periodic solutions of the two families must be different. This periodic solution that serves as

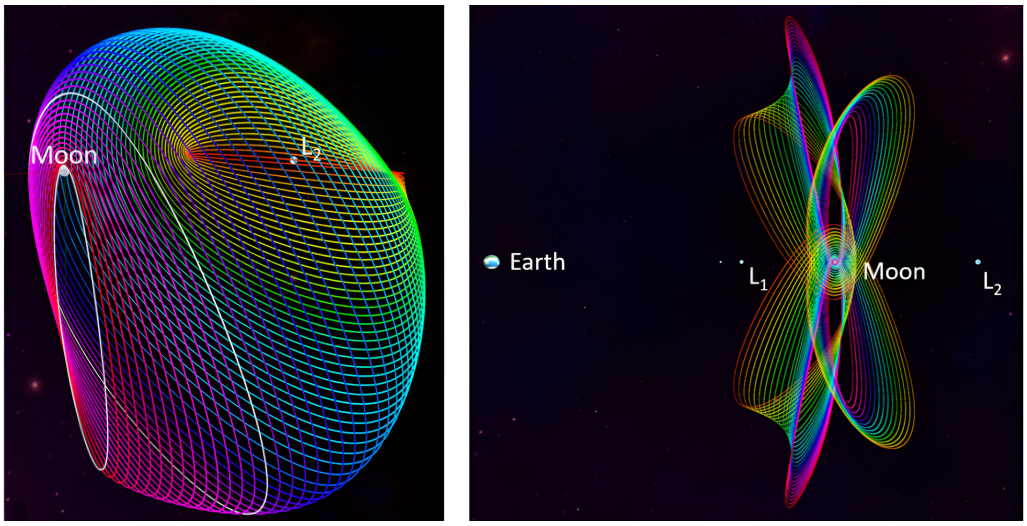

Figure 2.6: Southern halo family around EML_2 (left), and northern and southern NRHOs around EML_1 and EML_2 (right) (Davis et al., 2017).

the intersection between two families of orbits is called a bifurcation point, which is the foundation of *bifurcation theory*. Bifurcation points are identified by a change in the orbit's stability properties compared to the rest of the family. Specifically, there is a change in the eigenvalues of the orbit's monodromy matrix, explained in Section 2.6.

The so-called halo orbits then arise from the horizontal Lyapunov family by introducing a non-zero value for the z-coordinate at the appropriate bifurcation point. Due to the *theory of image trajectories* (Miele, 2010), every halo with a certain positive z-amplitude (A_z) will have an associated trajectory with a negative A_z of the same magnitude. This fact gives rise to the northern and southern halo families, which are mirror images of each other with respect to the xy-plane. The L_2 southern halo family is displayed in the illustration on the left of Figure 2.6, with the orbits (each described by one color) gradually getting further away from the second Lagrange point and closer to the Moon, while becoming more and more eccentric.

2.5.3 Near-Rectilinear Halo Orbits

Above a certain orbital energy, the stability properties (eigenvalues of the monodromy matrix) of the halo family become more favorable. This can be determined by means of the stability index υ, computed by

$$\upsilon = \frac{1}{2}\left(|\lambda_{max}| + \frac{1}{|\lambda_{max}|}\right) \tag{2.26}$$

where λ_{max} is the largest eigenvalue of the monodromy matrix from Equation (2.41), in Section 2.6. The larger υ is, the more unstable the orbit. The Near-Rectilinear Halo Orbits (NRHOs), which still belong to the halo family, are characterized by lower stability indices, leading to more stable orbits. Due to this, NRHOs are interesting as target orbits for space stations, since they require less propellant for station-keeping. The northern and southern families of NRHOs of the Earth-Moon system around L_1 and L_2 can be seen in the illustration on the right of Figure 2.6. Four groups of orbits can be recognized, all of them displaying highly eccentric trajectories. These orbits are characterized by short periods and a perilune considerably close to the Moon.

2.5.4 Distant Retrograde Orbits

A fundamentally different family of periodic solutions are the Distant Retrograde Orbits (DROs), as they orbit the secondary body instead of the Lagrange liberation points, which can be seen in

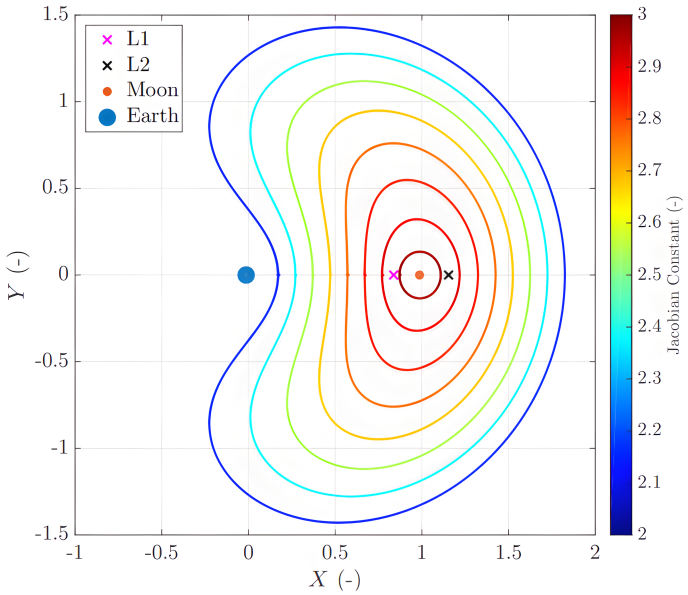

Figure 2.7: Distant Retrograde Orbit (DRO) family in the EM system (Bucci and Lavagna, 2016).

Figure 2.7. In this figure, the Earth-Moon family of DRO orbits is displayed, clearly showing that they are centered around the Moon. The name "retrograde" arises from the fact that the spacecraft would orbit the secondary body in a clockwise direction when viewing the xy-plane from above. They are symmetric with respect to the xz-plane, and they are planar and kidney-shaped, much like horizontal Lyapunov orbits. As explained by Q. Li et al. (2022), these orbits are stable ($\upsilon = 1$), so the use of manifold theory is not advantageous in this case.

There are other periodic solutions around the first two Lagrange points not explained in this section for conciseness. For instance, from the NRHOs, the butterfly family bifurcates (Spreen et al., 2017), and the horizontal and vertical Lyapunov orbits are connected by the axial orbits (Langemeijer, 2018). Lastly, higher-order DROs exist which revolve several times around the Moon before completing the orbit. These are referred to as PmDRO_n, with "Pm" being the order of period multiplication (e.g., P4DRO means that the orbit's period is roughly four times the one of the original DRO), and "n" being the subfamily identifier for the same period (Q. Li et al., 2022).

2.5.5 Generation of Periodic Solutions

Initial Solution - State Transition Matrix

An initial periodic solution must be found first to find the complete family of solutions (or the desired member of the family). To achieve this, a single-shooting differential correction method like the one explained by (Langemeijer, 2018) or (Tatay Sanguesa, 2021) is adopted. Firstly, the state vector, $\boldsymbol{x}$, can be defined in this case as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{r}^T & \boldsymbol{v}^T \end{pmatrix}^T \tag{2.27}$$

It should be noted that the state vector will contain the spacecraft mass when thrusting is included in the dynamics, as this mass will change with time according to an additional equation of motion. Secondly, the unperturbed (no thrust) equations of motion of the state vector are:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \implies \begin{cases} \dfrac{d\boldsymbol{r}}{dt} = \boldsymbol{v} \\ \dfrac{dv_x}{dt} = \ddot{x} = \dfrac{\partial \Omega}{\partial x} + 2v_y \\ \dfrac{dv_y}{dt} = \ddot{y} = \dfrac{\partial \Omega}{\partial y} - 2v_x \\ \dfrac{dv_z}{dt} = \ddot{z} = \dfrac{\partial \Omega}{\partial z} \end{cases} \tag{2.28}$$

These equations correspond to six first-order ordinary differential equations. The partial derivatives of Ω can be found in Equation (2.24). However, this system is highly non-linear. The system can be linearized by means of the State Transition Matrix (STM) $\boldsymbol{\Phi}(t, t_0)$ such that an initial state deviation from the reference trajectory $\delta\boldsymbol{x}_0$ can be related to a deviation $\delta\boldsymbol{x}(t)$ at time t:

$$\delta\boldsymbol{x}(t) \approx \frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{x}(t_0)} \delta\boldsymbol{x}(t_0) = \boldsymbol{\Phi}(t, t_0)\delta\boldsymbol{x}_0 \tag{2.29}$$

Now, the STM for all times t can be computed numerically via the following equation, being, of course, initialized as the identity matrix:

$$\frac{d\boldsymbol{\Phi}(t, t_0)}{dt} = \left.\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}(t)} \boldsymbol{\Phi}(t, t_0) \qquad\qquad \boldsymbol{\Phi}(t_0, t_0) = \boldsymbol{I}_6, \tag{2.30}$$

where the derivative of Equation (2.28) with respect to the states is the following Jacobian matrix:

$$\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \Omega_{xx} & \Omega_{xy} & \Omega_{xz} & 0 & 2 & 0 \\ \Omega_{yx} & \Omega_{yy} & \Omega_{yz} & -2 & 0 & 0 \\ \Omega_{zx} & \Omega_{zy} & \Omega_{zz} & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{3x3} & \boldsymbol{I}_3 \\ \boldsymbol{\Omega} & \boldsymbol{\Gamma} \end{bmatrix}, \tag{2.31}$$

with (Out, 2017)

$$\begin{aligned} \Omega_{xx} &= 1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3} + 3\left(\frac{1-\mu}{r_1^5}(x+\mu)^2 + \frac{\mu}{r_2^5}(x-1+\mu)^2\right) \\ \Omega_{yy} &= 1 - \frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3} + 3y^2\left(\frac{1-\mu}{r_1^5} + \frac{\mu}{r_2^5}\right) \\ \Omega_{zz} &= -\frac{1-\mu}{r_1^3} - \frac{\mu}{r_2^3} + 3z^2\left(\frac{1-\mu}{r_1^5} + \frac{\mu}{r_2^5}\right) \\ \Omega_{xy} &= \Omega_{yx} = 3y\left(\frac{1-\mu}{r_1^5}(x+\mu) + \frac{\mu}{r_2^5}(x-1+\mu)\right) \\ \Omega_{xz} &= \Omega_{zx} = 3z\left(\frac{1-\mu}{r_1^5}(x+\mu) + \frac{\mu}{r_2^5}(x-1+\mu)\right) \\ \Omega_{yz} &= \Omega_{zy} = 3yz\left(\frac{1-\mu}{r_1^5} + \frac{\mu}{r_2^5}\right) \end{aligned} \tag{2.32}$$

Initial Solution - Differential Correction Method

Now, having the linearized EOM, one can develop an iterative algorithm to achieve a desired state from a given starting point. Due to the theory of image trajectories, a periodic trajectory must have even or odd symmetry with respect to the x-axis (disregarding the time direction). Thus, the idea is to find an unpowered trajectory that, starting from the xz-plane, intersects with an appropriate state the Poincaré section corresponding to the xz-plane and all velocity components at a certain time. This flight time would then be the half-period ($T/2$) of the orbit. A Poincaré section Σ is usually an (n-1)-dimensional subset of the state space (where n is the dimension of the state space) defined by $S(\boldsymbol{x}) = 0$ (Henon, 1982). In this case, Σ is a 5-dimensional surface defined by $y = 0$. As such, both the starting point and the half-period point lie within the Poincaré section. Then, as the period is also a variable, the linearization becomes

$$\delta\boldsymbol{x}(T/2) = \boldsymbol{x}(T^*/2) - \boldsymbol{x}(T/2) \approx \boldsymbol{\Phi}(T/2, t_0)\delta\boldsymbol{x}_0 + \dot{\boldsymbol{x}}\delta T/2, \tag{2.33}$$

where T^* is the period of the actual orbit. Hence, $\boldsymbol{x}(T^*/2)$ is the state of the orbit at the half-period point and $\boldsymbol{x}(T/2)$ is the state obtained with an initial guess after integrating the EOM for an interval equal to the guessed half-period. The state as a function of time $\boldsymbol{x}(t)$ obtained with the initial guess is the one used to compute $\boldsymbol{\Phi}(t, t_0)$ in Equation (2.30). If the trajectory were to be divided into multiple time arcs and integrated separately, it would be a multiple-shooting differential correction method instead of a single-shooting one.

A horizontal Lyapunov orbit, for instance, would require at both the starting point and the Poincaré section intersection $y = z = 0$ and $v_x = v_z = 0$ to ensure symmetry, meaning that

$$\boldsymbol{x}(T^*/2) = \begin{pmatrix} x^*_{T/2} & 0 & 0 & 0 & v^*_{y_{T/2}} & 0 \end{pmatrix}^T \tag{2.34}$$

Then, as $x^*_{T/2}$ and $v^*_{y_{T/2}}$ are unknown, and only x_0, v_{y0} and $T/2$ are free, Equation (2.33) reduces to

$$\begin{aligned} \boldsymbol{\Phi}_{21}\delta x_0 + \boldsymbol{\Phi}_{25}\delta v_{y0} + v_{y_{T/2}}\delta(T/2) &= -y_{T/2} \\ \boldsymbol{\Phi}_{41}\delta x_0 + \boldsymbol{\Phi}_{45}\delta v_{y0} + \ddot{x}_{T/2}\delta(T/2) &= -v_{x_{T/2}} \end{aligned} \tag{2.35}$$

This system of equations is underdetermined, meaning that one of the coordinates can be fixed (usually δx_0 is fixed to zero for this family of orbits). However, these are linearized equations of motion that only hold very close to the reference trajectory, meaning that the computed initial correction vector will not lead to the desired final state unless the reference trajectory is almost the solution. The easiest way to solve this issue is to iterate. Thus, every time an initial state correction $\delta\boldsymbol{x}_0$ is computed, the trajectory gets re-propagated, and this new trajectory is used as the reference trajectory for the next iteration (needing to recompute the STM again). This ought to be repeated until a certain convergence tolerance is reached.

Nevertheless, due to the strongly non-linear dynamics, this method tends to diverge if the initial guess is not good enough. For planar (horizontal) Lyapunov orbits, as explained by Tatay Sanguesa (2021), an accurate enough approach would be to slightly perturb the Lagrange point in the x and v_y directions. Other approaches would be to find a suitable orbit from literature or to use a Richardson linear approximation (Richardson, 1980a,b).

As another example, to obtain a halo orbit, the initial and final z-coordinate are allowed to be different from zero at the Poincaré section, however, as for horizontal Lyapunov orbits, $y_0 = y^*_{T/2} = v_{x0} = v^*_{x_{T/2}} = v_{z0} = v^*_{z_{T/2}} = 0$. Hence,

$$\boldsymbol{x}(T^*/2) = \begin{pmatrix} x^*_{T/2} & 0 & z^*_{T/2} & 0 & v^*_{y_{T/2}} & 0 \end{pmatrix}^T \tag{2.36}$$

Then, with z_0 now being free to vary, Equation (2.33) in this case reduces to

$$\begin{aligned}
\boldsymbol{\Phi}_{21}\delta x_0 + \boldsymbol{\Phi}_{23}\delta z_0 + \boldsymbol{\Phi}_{25}\delta v_{y0} + v_{y_{T/2}}\delta(T/2) &= -y_{T/2} \\
\boldsymbol{\Phi}_{41}\delta x_0 + \boldsymbol{\Phi}_{43}\delta z_0 + \boldsymbol{\Phi}_{45}\delta v_{y0} + \ddot{x}_{T/2}\delta(T/2) &= -v_{x_{T/2}} \\
\boldsymbol{\Phi}_{61}\delta x_0 + \boldsymbol{\Phi}_{63}\delta z_0 + \boldsymbol{\Phi}_{65}\delta v_{y0} + \ddot{z}_{T/2}\delta(T/2) &= -v_{z_{T/2}}
\end{aligned} \tag{2.37}$$

This system is also underdetermined, having to keep one variable constant throughout the iteration process. To obtain an initial guess, again a Richardson approximation can be used or, as explained by Langemeijer (2018), a bifurcation point can be identified in the horizontal Lyapunov family, in which a slight perturbation in the z direction is introduced to transition towards the halo family.

The same procedure of adapting Equation (2.33) can be applied for the rest of the orbit families through bifurcation theory to obtain an initial guess, and more details are provided by Langemeijer (2018). As a separate note, for this research, however, the initial orbit (already converged) is retrieved from a database generated by previous authors who worked on Astos Solutions' "Orbit Generator Tool" (Walther and Wiegand, 2023). The chosen orbit corresponds to the one closest to the requested periodic solution.

Family Continuation

Once an orbit from the family has been obtained, the rest of the members of the family can be generated via continuation methods. One option is natural parameter continuation, which simply increases a chosen parameter (the period or A_z, for example) by a constant step and applies the differential correction method at each step to find the orbit corresponding to the parameter value. Thus, the previous member of the family with a perturbation in the chosen parameter is used as initial guess for the differential correction method (leaving the value for the continuation parameter constant in Equation (2.33)). To ensure convergence and a uniform distance between the family members, the parameter that has the most monotonous behavior ought to be the one used as continuation parameter. Consequently, for halo orbits specifically (the main focus of this thesis), x-continuation will be used for L_2 and z-continuation for L_1, as demonstrated by Tatay Sanguesa (2021).

However, natural parameter continuation may lead to highly non-uniform distributions of members for some families of orbits, such as the vertical Lyapunov family. Although not directly relevant to this thesis, pseudo-arclength continuation can be used to solve this issue. As explained by Langemeijer (2018), instead of only changing a single parameter, the goal is to adjust the step in several parameters depending on the difference in the previous two members. If the difference in parameters is very large, the step is reduced to correct for the strong gradient and follow the parameter curve tangentially. This idea, which aims to correct for steep changes in the z-parameter, is depicted in Figure 2.8. It displays a generic profile of the z-parameter as a function of the x-parameter, with sample approximation steps that aim to follow the gradient of the curve. In this case, n is the step index. As is evident from the figure and the previous explanation, this method requires two family members to be initialized. The required step can be computed via the difference in parameters of the previous two family members (Δ_n) and a scaling factor ρ_n to account for strong gradients in A_z:

$$\Delta_n = \boldsymbol{x}_0^{n+1} - \boldsymbol{x}_0^n \tag{2.38}$$

$$\rho_n = \sqrt{(x_0^{n+1} - x_0^n)^2 + (z_0^{n+1} - z_0^n)^2} \tag{2.39}$$

$$\boldsymbol{x}_0^{n+2} = \boldsymbol{x}_0^{n+1} + \Delta_n \frac{10^{-4}}{\rho_n} \tag{2.40}$$

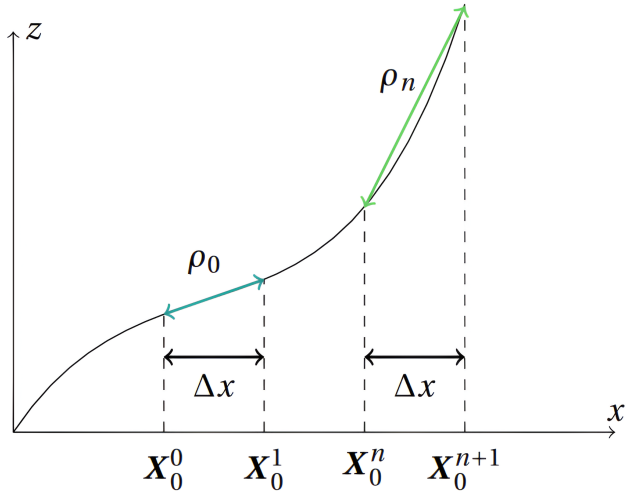

Figure 2.8: Depiction of the pseudo-arclength continuation method (Langemeijer, 2018)

2.6 Manifold Theory

As mentioned in Section 2.5, a convenient feature of the periodic solutions around the first two Lagrange libration points is the fact that stable and unstable manifolds arise from them. In the context of the CR3BP, manifolds are trajectories that asymptotically go towards or depart from periodic solutions with virtually the same Jacobi constant (orbital energy) as these solutions. Hence, they are pathways to leave or reach these orbits for "free" (almost zero fuel consumption). They are very useful in trajectory optimization for this reason, as the manifold departure or insertion point could be optimized to reduce the required propellant mass, greatly increasing design freedom.

These structures arise from the stability characteristics of these orbits. By means of the STM ($\Phi(t, t_0)$), the system dynamics can be linearized, as explained in Section 2.5. Furthermore, by considering the STM exactly one period T apart, the behavior of the spacecraft after each orbital revolution can be characterized by a linearized discrete system:

$$\delta \boldsymbol{x}(t_0 + T) \approx \boldsymbol{\Phi}(t_0 + T, t_0)\delta \boldsymbol{x}_0 = \boldsymbol{M}\delta \boldsymbol{x}_0 \tag{2.41}$$

$\boldsymbol{M}$ is called the monodromy matrix, and it linearly maps the periodic solution. From the stability of linear systems' theory (Olsder et al., 2011), the eigenvalues of $\boldsymbol{M}$ define the stability characteristics of the system. As it is discrete and not continuous, for the system to be stable, the eigenvalues must lie within the complex unit circle instead of in the complex left-half plane. From the work by Langemeijer (2018), the eigenvalues of any STM in the CR3BP always come in reciprocal pairs, meaning that if λ is an eigenvalue, so is $\frac{1}{\lambda}$. Moreover, specifically for the monodromy matrix, if an eigenvalue is complex, its complex conjugate must also be an eigenvalue. Lastly, one eigenvalue must be exactly one for a linear mapping of a periodic solution (a reciprocal pair in this case, due to a previous requirement). As a result, the eigenvalues can be ordered as

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| = 1 = |1/\lambda_3| \geq |1/\lambda_2| \geq |1/\lambda_1| \tag{2.42}$$

Then, the stability properties of the system are mostly determined by the largest and smallest eigenvalues, which give rise to the unstable and stable subspaces, respectively. This fact provides a sensible explanation for Equation (2.26). Using as nomenclature for the stable and unstable eigenvalues

$$\lambda_u = \lambda_{max} \qquad \lambda_s = \lambda_{min} = 1/\lambda_{max}, \tag{2.43}$$

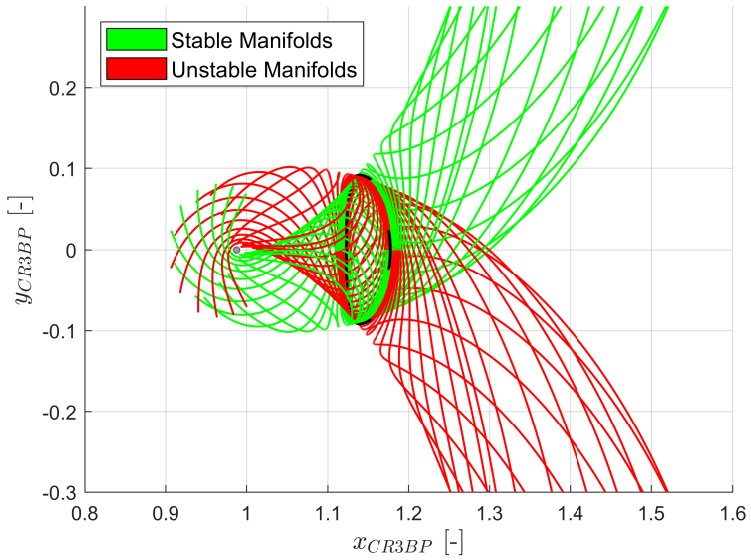

Figure 2.9: Stable and unstable manifolds emanating from a EML_2 northern halo of A_z = 5000 km.

the stable and unstable manifolds at a specific point in the orbit can be triggered by a slight perturbation in the direction of the eigenvector ($\boldsymbol{\nu}_s$ or $\boldsymbol{\nu}_u$) associated with the desired eigenvalue:

$$\boldsymbol{x}_m = \boldsymbol{x}_0 \pm \epsilon_p \frac{\boldsymbol{\nu}_{s/u}}{||\boldsymbol{\nu}_{s/u}||} \tag{2.44}$$

The perturbing factor ϵ_p is suggested to be 10^{-6} (Gómez et al., 1991) in non-dimensional units, as it needs to be small enough to achieve almost asymptotic departure but large enough to avoid an accumulation of numerical errors. This is the approach followed by Langemeijer (2018). Nevertheless, for trajectory optimization, authors usually choose values between 10^{-5} and 10^{-4} to secure faster departures from the periodic solutions while preserving considerably small perturbations that can be handled by the GNC (Guidance, Navigation, and Control) algorithm. Moreover, these manifold structures must be corrected anyway when transitioning the trajectory to a high-fidelity model. For instance, P. Zhang et al. (2013) employ 10^{-5}, whereas Ozimek and Howell (2010) and Kokou et al. (2014) use values slightly larger than 10^{-4}. For this research, a value of 10^{-4} is selected for ϵ_p.

Unstable manifolds must be propagated forward in time, whereas stable manifolds must be propagated backward in time. In addition, perturbing in a certain eigenvector direction leads to the inner manifolds (which go toward the secondary body), whereas perturbing in the opposite direction generates the outer manifolds. As such, a combination of four different manifolds can be generated at any point in the orbit, yielding the four different manifold families when all points of the orbit are considered. These families, as could be deduced from the name, present a pipe- or tube-like shape and, as explained by Koon et al. (2001), "these tubes partition the energy manifold and act as separatrices for the flow through the equilibrium region: those inside the tubes are transit orbits" (trajectories that go through the periodic solution from which the manifolds emanate) "and those outside the tubes are non-transit orbits". This is of course only true for trajectories of the same Jacobi constant as the periodic solution. Figure 2.9 shows the four families of manifolds associated with a certain halo around L_2 (by discretizing the orbit in 20 steps), generated with the *Orbit Generator Tool* (Walther and Wiegand, 2023). The inner manifolds correspond to the ones on the left of the halo, and the outer to the ones on the right side of the figure. In this case, the invariant manifolds were obtained by perturbing only the velocity components, with no discontinuity in the position. This decision was made by Tatay

Sanguesa (2021) to make the generated impulsive trajectories feasible within the CR3BP. It is evident from the figure that the manifolds obtained with this strategy maintain the tubular shape, display the stable/unstable symmetry, and are spaced in a considerably uniform way. Nonetheless, for this research, Equation (2.44) will be followed, with a perturbation in the position and velocity to guarantee uniformity, acknowledging that in low-thrust trajectories a perturbation solely in the velocity would render the trajectory infeasible anyway.

In conclusion, to propagate trajectories and generate periodic solutions within the CR3BP, a large set of equations and delicate procedures must be followed. Luckily, most of these have already been implemented and verified within the *Orbit Generator Tool* (Chapter 3), which is highly advantageous for the development of this research. All the theory presented in this chapter, including the frames and transformations, is employed for obtaining the optimized transfers, except for the pseudo-arclength continuation method. Moreover, the research will focus on transfers between halo orbits, so the rest of the explained periodic solutions can be disregarded.

Trajectory Optimization with the OGT 3

Over the past years, Astos Solutions GmbH has been developing a tool called the *Orbit Generator Tool* (OGT, previously called "HaloTool") (Walther and Wiegand, 2023) meant to provide optimized trajectories between Keplerian orbits and periodic solutions in the two-, three-, and four-body problems. These trajectories ought to serve as initial guesses for further optimization using the Analysis, Simulation, and Trajectory Optimization Software (ASTOS). This tool will be especially attractive in the future for mission design due to the advantages of these periodic solutions and the recent interest of the aerospace industry in using them in missions. For instance, the James Webb Space Telescope is currently in a halo orbit around the Sun-Earth second Lagrange point, and the Artemis I mission orbited the Moon in a DRO.

In this chapter, the capabilities of the OGT prior to the proposed implementations will be explained in Section 3.1, as they are important factors in the decision-making process due to the fact that this research ought to extend the tool. Then, the programming architecture of the OGT will be discussed in a very general way in Section 3.2, followed by a description of the chosen optimization methods and available optimizers in Section 3.3. Lastly, Section 3.4 explains how the user is meant to use the OGT to obtain a desired optimal trajectory, and the chapter ends in Section 3.5 with an overview of the tool's verification.

3.1 Purpose and Capabilities

The objective of the tool is to provide optimized trajectories within the Earth Two-Body Problem (2BP, neglecting the gravitational contribution of the spacecraft), the Earth-Moon Circular Restricted Three-Body Problem (CR3BP), or the Sun-Earth-Moon Bi-Circular Restricted Four-Body Problem (BCR4BP). It can optimize both impulsive and low-thrust trajectories, which serve as initial guesses for further optimization using ASTOS with the same model or with a high-fidelity one. The idea is to find solutions hopefully in the region of the global optimum (using heuristic optimization), such that ASTOS can then refine these solutions with direct collocation methods that use local optimization (with the solution being dependent on the initial guess), and thousands of optimizable parameters efficiently. The OGT is able to minimize the ΔV (single-objective optimization) as well as generate Pareto fronts with the ΔV and the Time of Flight (TOF) as objectives (multi-objective optimization), which will be explained in Subsection 3.3.4. As such, it provides the user with a wide variety of initial guesses as a function of the TOF to choose from in order to design a mission. A detailed explanation of the BCR4BP and how it is implemented into the OGT is provided by Maisch (2022).

The OGT can handle two kinds of trajectories: impulsive and low-thrust. The impulsive transfers can be between Earth/Moon Keplerian orbits, or to go to or come back from a periodic solution in the CR3BP, such as a halo orbit or a DRO. Moreover, it can handle trajectories between periodic solutions as well. For the BCR4BP, the so-called bounded motions are used, which are an adaptation of these CR3BP orbits to account for the Sun's influence. A famous example would be the 9:2 synodic resonant orbit around the Earth-Moon second Lagrange point (EML_2). A body in this orbit would complete nine revolutions at the time the Moon completes two revolutions around the Earth. These bounded motions are also available around the Sun-Earth Lagrange points (SEL). Intermediate phases for the impulsive transfers are Lambert arcs, stable or unstable invariant manifolds arising from the periodic solutions, flybys around the Earth or the Moon, and Weak Stability Boundary (WSB) points in the BCR4BP to take advantage of the Sun's influence on the system dynamics.

The periodic solutions are obtained by applying a single-shooting differential correction method to the orbit closest to the desired one, retrieved from a database. As explained in Subsection 2.5.5,

a z-continuation scheme is used for L1 halo families and an x-continuation scheme for L2 halo families. Then, the Lambert arcs are generated using a multiple-shooting differential correction (Tatay Sanguesa, 2021). The manifolds are obtained by a velocity perturbation in the positive or negative direction of the (un)stable eigenvector of the monodromy matrix at the specific point in the orbit. This procedure does not lead to the rigorous invariant manifolds explained in Section 2.6, as the position is not perturbed. Nevertheless, this decision ensures that the trajectories are continuous, and the obtained trajectories are considerably similar to the actual invariant manifolds, preserving the tubular shape and the symmetry, which can be seen in Figure 2.9. The flybys and WSBs are simply points in the configuration space connected to two Lambert arcs.

The low-thrust trajectories are only currently available in the 2BP and CR3BP. Unfortunately, transfers from halo orbit to halo orbit (or between any other periodic solutions) are not implemented yet, which is the purpose of this research. To connect to Keplerian orbits, the low-thrust law used by the OGT is the Q-law, explained by Walther (2022) and Petropoulos (2004), which is a control law that corrects for the difference in the Kepler elements with respect to the target orbit. This law cannot effectively target the true anomaly, meaning that a specific position and velocity cannot be reached, impeding transfers strictly between CR3BP periodic solutions. To connect to periodic solutions, invariant manifolds are used again. However, as previously explained, these manifolds are generated in the tool employing an instantaneous velocity difference at a specific point in the orbit. This cannot be achieved with low-thrust, so a "low-thrust perturbation phase" is added to enter the manifolds, which consists of having the spacecraft continuously thrust in the positive or negative direction of the (stable/unstable) eigenvector of the orbit's monodromy matrix at that point (Walther, 2022).

3.2 Programming Architecture

3.2.1 Programming Language

The OGT is written in Java, which offers a comparable speed to other languages extensively employed for mathematics, such as Python or MATLAB. Speed is a key requirement for the tool as during the optimization process thousands of function evaluations will be carried out, with some of them requiring the computation of trajectories in the order of several months, especially for low-thrust optimization. In addition, Java is especially suitable due to its compatibility with the ASTOS software (Tatay Sanguesa, 2021), as the goal is to completely integrate the tool within that software. Lastly, as this thesis is meant to extend the functionalities of the OGT, it would not be sensible to change the programming language unless strictly necessary. Hence, Java is the chosen language for this research as well.

3.2.2 Programming Paradigm

The OGT takes full advantage of the Object-Oriented Programming (OOP) capabilities of Java. The code is structured in classes, instantiated as objects that share functions and variables. As an example, all trajectory phases are instantiated as objects that connect to each other depending on the requested trajectory, making it effortless to optimize trajectories that use any reasonable combination of phases. Hence, the versatility of the tool is greatly increased.

Mathematical operations such as computing the STM, normalizing the variables, or propagating the EOM are also defined in classes. This makes extending the tool with new trajectory phases straightforward since just a class for the new phase and a few classes for the new mathematical operations would need to be added to the program. This is ideal in the context of this thesis, as the goal is to add a new functionality to the tool: a low-thrust phase between periodic solutions. The fact that the phases and mathematical operations are structured in classes is also extremely convenient for testing and debugging purposes, since code repetition is mostly avoided. As such, a certain functionality would only need to be modified in one place to fix the program in any optimization context. Moreover, readability is enhanced.

3.2.3 Propagators

The OGT uses the Cowell propagator[1], which employs the traditional cartesian coordinates and tends to be the least accurate. Due to the highly perturbed motion (CR3BP with trajectories passing close to the secondary body), most of the alternative propagators lose their advantages. Specifically, Encke, which defines the state as the difference with respect to a reference Keplerian orbit defined by the initial state, must be immediately discarded because the resulting trajectories greatly differ from Keplerian orbits. Consequently, the use of Kepler Elements is discouraged, as some trajectories may well encounter singularities with these formulations. In contrast, Modified Equinoctial Elements or the Unified State Model (Vittaldev et al., 2012), both of which can be singularity-free, may show advantages in terms of accuracy and/or speed. More information on these propagator alternatives is provided by Hintz (2008). However, the equations of motion of the CR3BP are in a non-inertial frame (as this is where the periodic solutions and the equilibrium points arise), making it difficult to derive the equations for different propagators. Moreover, the method chosen for this work, explained in Chapter 5, is fully derived within the context of the Cowell propagator equations, meaning that the method would need to be adapted if the equations were to be changed, or a transformation would need to be carried out at every integration step (and sub-steps). Due to the uncertainty in the complexity and drawbacks of this task and the time constraints of this research, it is decided to not pursue this approach.

The large variation in the state derivatives can indeed decrease the efficiency of the OGT as the integrator tolerances will need to be reduced to have an acceptable error. Nevertheless, the use of non-dimensional quantities helps with coping with the cumulative error arising from the large state derivative values. Henceforth, Cowell is the most sensible choice.

3.2.4 Integrators

As explained by Tatay Sanguesa (2021), the use of variable step size integrators is encouraged as the trajectories are expected to get close to the Moon, drastically changing the accelerations. Hence, reducing the step size only in these regions to maintain the required accuracy will decrease the computational effort of the propagation. As the OGT is written in Java, a choice will be made from the *Apache Commons* toolbox[2], since implementing one from scratch is probably not worth the effort as it is not possible to determine a priori which is the best for the problem. As thousands of orbital propagations are expected to be carried out throughout the optimization, it is crucial to find the fastest integrator that provides sufficient accuracy. As such, an integrator trade-off will be performed in Section 6.2. Specifically, Dormand-Prince, Adams-Moulton, Higham-Hall, and Gragg-Bulirsch-Stoer integrators and integrator settings will be subjected to a trade-off between speed and accuracy.

3.3 Optimization

The OGT uses static optimization to find minimum ΔV trajectories (or minimum propellant mass for low-thrust) by choosing an appropriate set of optimizable parameters or design variables $\boldsymbol{p}$, of dimension N, for which the cost or objective (scalar) function $f(\boldsymbol{p})$ is minimized. In addition, the vector of parameters must lie within a specified constraining set Π, and the solution must satisfy certain equality ($\boldsymbol{q}(\boldsymbol{p})$) and inequality ($\boldsymbol{h}(\boldsymbol{p})$) constraints. From the work by Papalambros and Wilde (2000), this problem can be mathematically expressed in *negative null form* as

[1] In this research, the term *propagator* is used as analogous to the equations of motion formulation.

[2] URL:https://commons.apache.org/ [Accessed: 30/01/2024]

$$\begin{aligned} \underset{\boldsymbol{p}}{min} \quad & f(\boldsymbol{p}) \\ subject\ to \quad & \boldsymbol{q}(\boldsymbol{p}) = \mathbf{0} \\ & \boldsymbol{h}(\boldsymbol{p}) \leq \mathbf{0} \\ & \boldsymbol{p} \in \Pi \subseteq \mathbb{R}^N \end{aligned} \tag{3.1}$$

In static optimization, as opposed to dynamic optimization, the design variables are constant with time, i.e., constant for each evaluation of $f(\boldsymbol{p})$. However, in dynamic optimization, the design variables are functions of time and modify the equations of motion, which are included in the problem as additional equality constraints (Bryson and Ho, 1975). This is the basis of optimal control. Static optimization is generally easier to solve, as will be made clear in Section 4.5. Thus, it is convenient to formulate optimization problems as static optimization problems as much as possible, which has been the case for all OGT transfer types so far. There are several families of methods to solve optimization problems: local, heuristic, and hybrid methods. They are common to both static and dynamic problems, as long as the dynamic optimization problems are put into an appropriate form (more on this in Section 4.5).

3.3.1 Local Optimization

Local optimization methods, as the name suggests, search the design space until a local minimum is reached (or maximum, if desired). They can be gradient-based, like Sequential Quadratic Programming (SQP), or gradient-free, like Nelder-Mead Simplex (NMS), depending on whether they use gradient information. Nevertheless, they basically follow the direction in which the objective function is minimized while attempting to satisfy all constraints.

As the search space is usually narrow, unless the problem is convex the found solution is not guaranteed to be a global minimum. It is common for these methods to converge to different solutions depending on the initial guess (they get "stuck" in local minima). In addition, they may experience convergence problems for difficult objective functions, making it cumbersome to find an appropriate initial guess when the feasible domain (the subset of design variables that satisfy all constraints) is small or non-smooth. For these two reasons, a usual approach is to run them numerous times with different initial guesses. The main advantage of these methods is their convergence speed, with modern computers being able to handle >100,000 design variables. Moreover, they converge to an actual mathematical minimum, so with a suitable initial guess in the region of the global minimum, a solution can be refined very efficiently.

The most popular local optimization algorithms are SQP and Interior Point Optimization (IPOPT). These have proven to be two of the best-performing and most robust local algorithms for constrained, highly non-linear problems in a wide variety of realms. For instance, for trajectory optimization in the three- and four-body problems, SQP is used by Ozimek and Howell (2010) and Pergola et al. (2009) and IPOPT is employed by Parrish et al. (2016). Furthermore, SQP is used by both the SNOPT (Sparse Non-linear OPTimizer) (Gill et al., 2005) and WORHP (We Optimize Really Huge Problems) (Büskens and Wassel, 2013) optimizers, which are implemented within the ASTOS software. WORHP even makes use of IPOPT to solve the quadratic sub-problems of SQP. Even though no stand-alone local optimizers are currently implemented in the OGT, nor will they be utilized to generate trajectories with the OGT, the refinement of the trajectories with the ASTOS software will be carried out with local optimization.

3.3.2 Heuristic Optimization

Another approach to solving optimization problems is heuristic optimization, sometimes referred to as global optimization. They aim to find the global optimum of the problem by applying search

strategies that explore the design space in a much broader sense than local methods, attempting to avoid getting "stuck" in local minima. These are the optimization methods generally used in the OGT as they are the most likely to find solutions in the region of the global optimum (the aim of this thesis). The main drawbacks of these optimizers are the fact that convergence to a minimum is not ensured unless it is set as a constraint in the cost function, and that the computational cost greatly increases with the number of design variables (in order to obtain a good solution). Thus, they are usually suitable for problems with a low number of design variables (usually <20 for difficult problems), especially recommended for non-smooth problems with many local minima. In addition, the quality of the solution has proven to be dependent on the pseudo-random seed used, as well as on the optimizer settings and the optimizer itself. Therefore, to obtain a solution to a specific problem the general approach is first to choose an optimizer, then to tune the optimizer settings to suit the problem characteristics and behavior, and lastly to run the optimization with several seeds. Lastly, constraint handling is a point of attention in these methods, as there is a wide variety of strategies to deal with them, many of which are highly dependent on the tunable parameters.

Most of the heuristic optimizers implemented in the OGT come from the Pagmo scientific library for parallel optimization (Biscani and Izzo, 2020). This open-source software, greatly supported by ESA and Google, offers a wide variety of local and heuristic optimizers with the possibility of using several islands that exchange information in order to parallelize and broaden the search (Ahlborn, 2023). The strategies for the different heuristic methods[3] usually come from mimicking nature and adding probabilities to specific decisions throughout the optimization problem.

For instance, the most famous methods are evolutionary and genetic algorithms, which aim to follow the principles of evolution in nature closely by creating a dynamic population that ensures the survival of the fittest candidates (the ones with lower values for the cost function) and makes the new "offspring" come from these fit individuals[4]. A famous variant of these algorithms based on the same principle is Differential Evolution (DE). On the other hand, ant colony optimizers such as Genetic Ant Colony Optimization (GACO), also referred to as extended ant colony optimization, try to mimic the behavior of ants when they find the shortest route to food by following the paths with the most pheromones. As such, new individuals are generated by means of Gaussian probability distributions with numerous kernels that resemble pheromone paths. Another ant colony optimizer implemented in the OGT is the Mixed Integer Distributed Ant Colony Optimization (MIDACO)[5]. Other examples of heuristic optimizers are Particle Swarm Optimization (PSO), which mimics the behavior of food-searching swarms (in this case the individuals "move around" instead of being substituted by new ones), Covariance Matrix Adaptation Evolutionary Strategy (CMA/ES), which combines probability distributions with evolutionary algorithms, and Simulated Annealing (SA), which is an iterative random search that narrows down its search with each iteration. SA can be considered a special case of Tabu Searches, which enhance local search by using memory structures (tabu lists) to avoid revisiting previously examined solutions and explore the solution space better. MIDACO, SA, and CMA/ES have shown to generally be the best-performing ones in the OGT, with SA being usually difficult to tune (Ahlborn, 2023).

3.3.3 Hybrid Optimization

Hybrid optimization methods aim to combine the strengths of both local and heuristic methods. The idea is to run a heuristic optimizer and let some or all individuals be further optimized with a local optimizer to fix the issue of not converging to minima. Within the OGT, taken from Pagmo, hybrid

[3]URL: https://esa.github.io/pagmo2/ [Accessed: 10/07/2024]

[4]In the context of heuristic methods, an individual is simply a certain combination of design variables, i.e. a specific point in the design domain.

[5]URL: http://www.midaco-solver.com/ [Accessed: 22/01/2024]

optimization is easily implemented by making one of the aforementioned parallel islands run a local optimizer (Nead-Melder Subplex) which optimizes the best solutions of the rest of the heuristic islands and then feeds the new solutions back to them. This is done by placing the islands in a rim configuration (Ahlborn, 2023), but only works for SO optimizations. Another approach to hybrid optimization is to use heuristic optimization to find solutions that satisfy the constraints, which will then be optimized locally to minimize the actual cost function (Sentinella and Casalino, 2006). Moreover, heuristic optimization can be used to evaluate the convergence capabilities of each individual when used as initial guess for a local optimization method (Coverstone-Carroll et al., 2000). However, this method can be rather inefficient, as assessing the fitness of an individual would require running a local optimizer for several iterations. For this research, however, hybrid optimization is not employed.

3.3.4 Multi-Objective Optimization

A distinction between single-objective (SO) and multi-objective (MO) optimization should be made. In the context of trajectory optimization and space mission design, it is equally desirable to reduce the required propellant mass (Lee et al., 2005), as it will greatly determine the design of the spacecraft subsystems and drive the mission cost, as it is to reduce the time of flight (TOF) since it will decrease the operational costs and increase both the reliability of the mission and the spacecraft's operational lifetime. Thus, a method that could optimize both the propellant mass and the TOF would be highly convenient. The optimization problem can be reformulated as

$$\begin{aligned} \underset{\boldsymbol{p}}{min} \quad & \boldsymbol{F}(\boldsymbol{p}) = \begin{pmatrix} f_1(\boldsymbol{p}) & f_2(\boldsymbol{p}) & \dots & f_m(\boldsymbol{p}) \end{pmatrix}^T \\ subject\ to \quad & \boldsymbol{q}(\boldsymbol{p}) = \boldsymbol{0} \\ & \boldsymbol{h}(\boldsymbol{p}) \leq \boldsymbol{0} \\ & \boldsymbol{p} \in \Pi \subseteq \mathbb{R}^N, \end{aligned} \tag{3.2}$$

where m in this case is the number of objectives to be optimized. This problem would then not converge to a single minimum but to a set of optimal parameters called a "Pareto front" of dimension $m-1$. This set comprises all non-dominated feasible solutions found by the optimizer (Papalambros and Wilde, 2000), i.e. all individuals that are not worse than any other individual in all objectives simultaneously. A schematic of a Pareto front for a MO optimization with two objectives can be found in Figure 3.1. The utopia (ideal) and nadir points can be obtained with the extreme points of the Pareto set, as shown in the figure. The hypervolume would then be the volume in between the Pareto front and a chosen reference point, as depicted in Figure 3.2. Note that for $m = 2$ the hypervolume corresponds to an area.

To solve MO optimization problems, two approaches are generally used: *decomposition* and *non-dominated sorting*. *Decomposition* decomposes the problem into several weighted functions of the objectives and searches for a minimum of each of these functions independently. With enough functions, a homogeneous Pareto front can be achieved that spans a wide range of objective values. The simplest decomposition method is the *weighted sum decomposition*, with the performance index (cost function) being a weighted sum of the objectives:

$$F(\boldsymbol{p}) = \sum_{i=1}^{m} a_i f_i(\boldsymbol{p}),\ a_i \geq 0,\ \sum_{i=1}^{m} a_i = 1 \tag{3.3}$$

However, not all Pareto-optimal points can be obtained as a minimum of a weighted sum of the objectives if the Pareto front is not convex. A famous decomposition method that solves this issue would be Tchebycheff decomposition, explained by Ahlborn (2023). Optimizers in the OGT (including Pagmo) that use decomposition are MIDACO or Multi-Objective Evolutionary Algorithm with Decomposition (MOEA/D), based on DE.

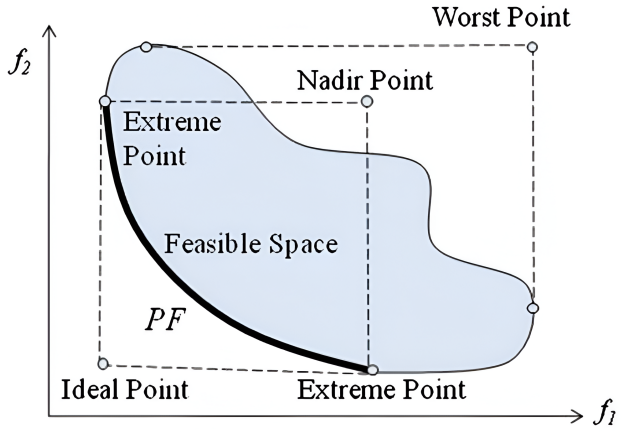

Figure 3.1: Diagram of a Pareto front for a MO optimization of two objectives, with its utopia or ideal point and its nadir point (H. Wang et al., 2017).

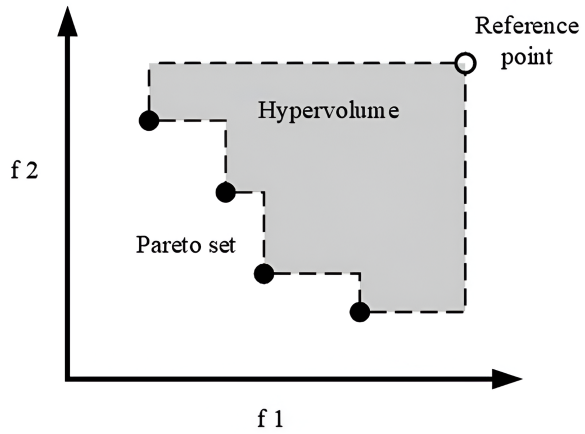

Figure 3.2: Diagram of the hypervolume with respect to a reference point for a MO optimization of two objectives (Wu et al., 2023).

Then, *non-dominated sorting* ranks the fitness of the individuals depending on the order of the non-dominated (Pareto) front to which they belong. The fittest individuals correspond to the first Pareto front. Next, the second Pareto front is computed by ignoring the first Pareto front and the third is obtained by ignoring the first two. The fitness of the second front is ranked lower than the first but higher than the third. This algorithm goes on until all individuals in the population have been ranked. Within the same non-dominated front, the individuals can be ranked, for instance, by how densely populated the front is in that region (Coverstone-Carroll et al., 2000). This is called *fitness sharing* or *niching*. Like this, uniformity in all regions of the front is promoted. Another option would be to rank them depending on how much they contribute to the hypervolume taking the nadir point as the reference point (Acciarini et al., 2020). As such, the points further away from the nadir are ranked as fitter. This method is called *hypervolume-based non-dominated sorting*. In the OGT, Multi-Objective Hypervolume-Based Ant Colony Optimization (MHACO), Non-dominated Sorting Particle Swarm Optimization (NSPSO), or Non-dominated Sorting Genetic Algorithm (NSGA) use non-dominated sorting to solve MO problems. The last two are not yet interfaced with the OGT, as such, they will be added to the tool in this research to have more MO optimizers available for performance comparison.

3.4 Program Flow

The Orbit Generator Tool aims to reduce the user input and required knowledge of the problem as much as possible, which is reflected in the brief input files. However, in MO optimization all Pareto-optimal solutions are equally promising. The choice is subjected to factors and constraints dependent on the intended mission. Therefore, it makes sense that after the user is presented with the computed Pareto front, they manually have to choose a solution that they would like to refine. For this refinement, a subsequent SO optimization on the ΔV or propellant mass will be carried out with the selected TOF as a constraint. This new optimization can use heuristic, or hybrid optimization as previously described. Of course, if this new optimization is not deemed necessary, one can simply run the chosen solution and generate the trajectory files.

The output files of the SO optimization include files to automatically generate an equivalent ASTOS scenario to better visualize the trajectory, to optimize further with direct collocation, or to transition the trajectory to a high-fidelity model. The OGT input and output files will be explained in

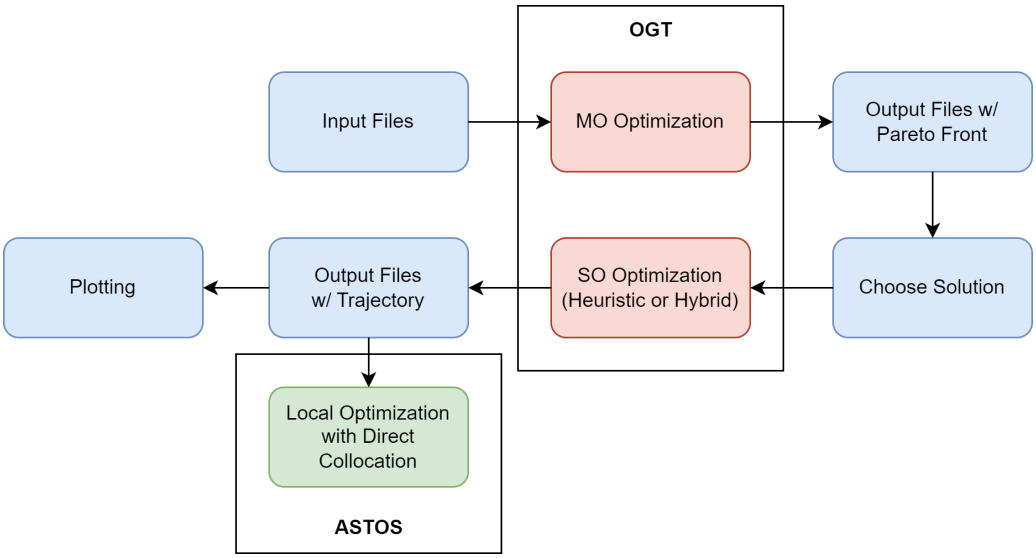

Figure 3.3: General OGT user workflow diagram.

Section 5.3, with the required input files to optimize low-thrust transfers between periodic solutions (the addition of this research to the OGT). A workflow diagram of the explained procedure can be found in Figure 3.3.

3.5 Verification

The OGT has been under development for several years by authors such as Walther (2022), Tatay Sanguesa (2021), or Maisch (2022). Each of them performed unit and system testing on their respective implementations. Moreover, verification procedures were carried out, such as a comparison between the OGT halo orbits and the ones obtained in literature, yielding maximum errors in the order of 10^{-5} (in the non-dimensional state space) (Tatay Sanguesa, 2021). Via the comparison made by Walther (2022) between the OGT low thrust trajectories with the ones that would be obtained with ASTOS and GMAT[6] for the same initial conditions and thrust strategy, the orbit and manifold phases are considered to be verified for this thesis. Furthermore, the coordinate frames and transformations have also been tested.

As such, for this work, only the new implementations (the new thrusting phase, its interface with the optimization procedure, the new optimizers, and the new mathematical tools) and the final complete trajectories ought to be verified. The unit and system testing overview of the implementation can be found in Subsection 5.3.4, whereas the trajectory verification is carried out in Section 8.1.

[6]URL: https://software.nasa.gov/software/GSC-17177-1 [Accessed: 31/01/2024]

Two-Point Boundary Value Problem 4

This chapter aims to provide an overview of the existing approaches to solving low-thrust trajectory optimization problems and illustrate the reasons for the final choice. Each of the methods will be briefly explained, with the chosen method (indirect optimization) elucidated in greater depth. Then, their advantages and drawbacks and their suitability to the problem at hand will be assessed. The collection of methods is largely built from the work by Morante et al. (2021).

The chapter starts with a general description of the problem to be solved in Section 4.1, as well as general requirements for the tool to be developed in Section 4.2. Then, Section 4.3 explains ways to obtain initial guesses, whereas Section 4.4 focuses on the different closed-loop control laws. Lastly, optimal control is discussed in Section 4.5, with a summary of the trade-off present in Section 4.6.

4.1 The Problem

In Newtonian mechanics, the complete state of a body with fixed mass can be described with six parameters: three position components and three velocity components (the dimension of the state space). In the 2BP, any trajectory without thrust can be fully described analytically with only five parameters (the first five Kepler elements), as they are conic sections. The sixth parameter (the true anomaly) determines the location along the periodic solution. Hence, the first five parameters remain constant (or vary slowly with time, depending on the assumptions), with the true anomaly being the only fast-varying parameter. This enables effective transfers between Keplerian orbits using closed-loop control laws where only five parameters need to be targeted (such as the Q-Law used in the OGT). However, something like this has not been found for the CR3BP. Therefore, the full state must be targeted to enter a desired periodic solution.

Regardless of the number of parameters targeted, the transfer problem between two periodic solutions in the CR3BP is a so-called *Two-Point Boundary Value Problem* (TPBVP), meaning that the goal is to achieve a final state $\boldsymbol{x}_f^*$ from an initial state $\boldsymbol{x}_0^*$ while satisfying a set of dynamic constraints (the EOM) throughout the whole trajectory. This problem can be described mathematically by a set of equality constraints, with $\boldsymbol{\psi}$ being a boundary conditions vector function:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) \\ \boldsymbol{\psi}_0(\boldsymbol{x}_0) &= \boldsymbol{x_0} - \boldsymbol{x}_0^* = 0 \\ \boldsymbol{\psi}_f(\boldsymbol{x}_f) &= \boldsymbol{x_f} - \boldsymbol{x}_f^* = 0 \end{aligned} \tag{4.1}$$

Following Poulsen (2012), if $\boldsymbol{x}_0^*$ and $\boldsymbol{x}_f^*$ define the full state of the body, they are called *simple start and end point constraints*. However, as explained in Section 2.3, when thrust is active the EOM contain an extra equation for the mass flow. Consequently, the state vector must include the spacecraft's mass. As no constraint is imposed on the final spacecraft mass, but $m_0 = m_0^*$, the final constraint is a *simple partial end point constraint*.

4.2 Implementation Requirements

On top of satisfying the constraints in Equation (4.1), the goal is to find Pareto-optimal trajectories that minimize both the propellant mass consumption and the flight time. Overall, a suitable method to solve this problem must satisfy the TPBVP constraints, be versatile enough to handle very different orbit transfers and spacecraft specifications and have an efficient thrusting strategy to increase performance as much as possible. Moreover, as explained in Section 3.1, the method needs to have

a low enough number of design variables to be suited for heuristic optimizers and MO optimization in order to hopefully find solutions in the region of the global minimum that trade-off between TOF and required propellant mass, providing a wide variety of initial guesses for further optimization with ASTOS.

Additional requirements for the implementation include being compatible with the OGT. This means that on top of following the same coding architecture, it should have comparable optimization times, require a similar number of input parameters, be compatible with other OGT phases such as the manifold and halo orbit phases, and be user-friendly. Lastly, as explained in Chapter 1 and Section 3.1, the obtained trajectories need not be strictly feasible but serve as promising initial guesses for direct collocation methods.

4.3 Rough Initial Guesses

Depending on the chosen method, an initial trajectory may be helpful or required to solve the problem. For instance, local optimization using direct methods or successive convex optimization requires a complete trajectory to initialize the algorithm, with the quality of the solution greatly depending on the initial guess. Furthermore, indirect methods can be constructed upon a specific promising trajectory to improve the performance of the obtained minimum, even if heuristic optimization is used.

4.3.1 Simple Orbit Propagation

The simplest trajectory to use as initial guess would be to propagate the starting periodic solution for several orbital periods, then "jump" to the target periodic solution at a certain point in time, and lastly to propagate the target periodic solution for several periods as well. This trajectory will certainly not satisfy the dynamic constraints but, since it is continuous in time, it is good enough for an appropriate optimization method to eventually find a trajectory that satisfies all constraints and minimizes the fuel. This kind of initial guess was used in combination with a direct collocation method by Parrish et al. (2016), obtaining promising results. Nevertheless, it is explained in the article that the quality of the results and their time of flight are greatly dependent on the number of periods that each orbit is propagated. Also, this method takes no advantage of manifold theory.

4.3.2 Manifold Patching

Another more refined way that makes favorable use of the dynamics would be to attempt to connect an unstable manifold from the starting periodic solution to a stable manifold of the target orbit. Then, provided they are connected in configuration space, the only discontinuity would be a velocity difference at the patch point. If the manifolds are connected without any velocity difference, the connection is called *heteroclinic*, which is only possible if the starting and target orbits have the same Jacobi constant. This is the method chosen by P. Zhang et al. (2013), connecting two invariant manifolds to generate a trajectory around which to build the indirect optimization method to smooth the transition between them with continuous thrust[1].

A strategy to find manifold connections between orbits of the same energy is described by Langemeijer (2018). It is to find the best connection by varying the location of the connecting Poincaré section. This method could be used to find connections between manifolds of different energy, however, it is not guaranteed that the manifolds that happen to intersect each other are the ones that lead to the cheapest trajectory in terms of fuel consumption.

[1]Not strictly an initial guess, but a "guide" for how the boundary conditions change as a function of the initial and final thrusting times.

4.3.3 Intermediate Orbits

In the particular case where the transfer takes place between periodic solutions of the same family, a common method is to connect several intermediate orbits of the same family until the desired orbit is reached. Again this trajectory would not satisfy the dynamic constraints as it is not continuous in position, but is enough for other methods to refine it. Preferably, the energy of the intermediate orbits should be in between the energies of the start and target orbits. In addition, the gradient of the energy throughout the different intermediate orbits should not change sign to avoid inefficiencies. This method is used by Kayama et al. (2021) to generate initial guesses for a successive convex optimization procedure. Hiraiwa et al. (2022) propose a beam search strategy to find the best combination of intermediate orbits to minimize fuel. The method could in principle be adapted to transfers between orbits of different families with the proposed beam search algorithm to refine the initial guess, although no literature supporting this idea has been found so far. Nevertheless, a considerable drawback of this method is the fact that invariant manifolds are not considered.

4.4 Direct Control Laws

Direct control laws are convenient strategies to generate initial guesses for other optimal control methods (Morante et al., 2019), as they are usually simple and easy to implement. Furthermore, they often have a low number of optimizable parameters. Some of them are closed loop, which is convenient to reduce the required number of correction maneuvers in a real mission, and others are able to target all position and velocity components. However, due to the nature of how they are generated, the obtained trajectories are suboptimal, being nevertheless suitable as initial guesses for further optimization using ASTOS (Walther, 2022). It should be noted that, as explained in Section 4.1, a control law will only be valid for the problem at hand if it is able to target the full state disregarding the mass (from now on, the full state). For instance, if the law is defined with respect to the cartesian state, all components of the position and velocity vectors would need to be targeted, whereas, if defined with respect to the Kepler elements or the Modified Equinoctial Elements (MEE), it would need to target all six elements.

4.4.1 Blended Control

The control strategy for blended control is based on a weighted sum of the thrusting directions that maximize the change in each of the Kepler elements independently. This is achieved by manipulating the variation of parameter equations of the Kepler elements (Lagrange Planetary Equations (Vallado, 2013)). For instance, Gao (2007) proposes a control strategy that targets the semi-major axis, the eccentricity, and the inclination of a Keplerian orbit. Zuiani et al. (2013) explain another blended control approach to raise the semi-major axis by means of tangential thrusting. None of them can target the full state, rendering blended control unsuitable for the problem.

4.4.2 Lyapunov Control

As the name suggests, Lyapunov control is based on Lyapunov stability. The goal is to find a control strategy that minimizes the gradient of the Lyapunov function. From Hatten (2012), for a system of the form

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{f}(\Delta\boldsymbol{x}) \qquad\qquad \Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_t \tag{4.2}$$

the target state x_t is said to be Lyapunov stable if there exists a function $V(\Delta\boldsymbol{x})$ such that:

$$\begin{cases} V(\mathbf{0}) = 0 \\ V(\Delta\boldsymbol{x}) > 0 \quad \forall \ \Delta\boldsymbol{x} \neq \mathbf{0} \\ \dot{V}(\Delta\boldsymbol{x}) < 0 \quad \forall \ \Delta\boldsymbol{x} \neq \mathbf{0} \\ \lim\limits_{||\Delta\boldsymbol{x}|| \to \infty} V(\Delta\boldsymbol{x}) = \infty \end{cases} \tag{4.3}$$

Hence, the best thrust strategy (which need not be optimal in terms of minimizing fuel consumption) would make $V(\Delta \boldsymbol{x})$ go to zero as fast as possible, i.e., make $\dot{V}(\Delta \boldsymbol{x})$ as negative as possible at all times.

This is the framework in which the famous Q-Law used in the OGT for transfers between Keplerian orbits and CR3BP periodic solutions was developed (Petropoulos, 2004). The Q-Law is extremely convenient due to its versatility, as it can target up to five Kepler elements, but fewer if desired. Moreover, it has a smart coasting strategy, which is lacking in many other control laws, to only thrust in regions in which it is efficient enough (e.g. when increasing the semi-major axis, only thrust close to the periapsis) using the effectivity cut-off value. Lastly, only the weights of the targeted parameters and the effectivity cut-off value need to be optimized, making this law very suitable for heuristic optimization.

The Q-Law defined with respect to the Kepler elements has been extensively used in literature, such as by Petropoulos (2004) or Falck et al. (2014), especially for the 2BP. As proved by Walther (2022), the Q-Law performs considerably well also in the CR3BP, even though the Q-law does not account for the gravitational perturbations of the secondary body. It has even been successfully formulated in terms of MEE by Varga and Sánchez Pérez (2016). Nonetheless, the Q-Law can only target five Kepler elements, the true anomaly (or the true longitude in MEE) cannot be reached efficiently as it is a fast-varying parameter, extremely difficult to effectively drive it to the desired value. In addition to being highly dependent on time, it is also sensitive to thrusting in the radial and along-track directions. Hence, unless the desired true anomaly is achieved at the same point in time as the rest of the Kepler elements, having the Q-Law focusing on the true anomaly early on would be highly suboptimal.

Falck et al. (2014) also propose a Directional Adaptive Guidance control law that is again unable to target the sixth Kepler element. Furthermore, a collection of seven low-thrust control laws (including the Q-Law again) is provided by Hatten (2012). Only one of them, explained by Naasz (2002), can target all six Kepler elements, however, it is a rather simple control law with constant weights and a poor coasting mechanism, and, thus, greatly suboptimal.

Lastly, Parrish et al. (2016) deem these kinds of control laws ineffective for generating initial guesses for transfers between periodic solutions in the CR3BP. Henceforth, it is safe to discard Lyapunov control as a fitting method.

4.4.3 Fourier Expansion

As explained by Hudson and Scheeres (2009), "any piecewise-smooth function $f(\alpha)$ with a finite number of jump discontinuities on the interval (0, T) can be represented by a Fourier series that converges to the periodic extension of the function itself" (Fourier's theorem). As such, each of the components of the thrust (including coast arcs) can be expressed as a Fourier series of the form:

$$f(\alpha) = \sum_{k=0}^{\infty} \left[a_k \cos\left(\frac{2\pi k \alpha}{T}\right) + b_k \sin\left(\frac{2\pi k \alpha}{T}\right) \right] \tag{4.4}$$

where α is an arbitrary variable and a_k and b_k the constant coefficients of the series. The more terms k (and therefore coefficients) the series has, the better the approximation of the thrusting function. Furthermore, with this approximation of the thrust profile, by averaging the Gaussian form of the Lagrange planetary equations over one orbit, the 2BP EOM can be greatly simplified, using only 14 of the Fourier coefficients (regardless of how many are used to represent the thrust profile). Nevertheless, these 2BP equations are probably not applicable to the CR3BP. In addition, even if the Fourier expansion were to be used to approximate the thrust profile, no technique is proposed to compute the optimal coefficients for reducing the required propellant mass, or even reaching the target orbit. Thus, it would be a blind search of coefficients for which heuristic optimization would not be suitable, as probably a large number of coefficients are required to obtain good enough results.

4.4.4 Shape-Based Methods

Shape-based methods simply assume a specific shape for the position coordinates and analytically compute the required thrust profile such that the trajectory satisfies the equations of motion in the specified TOF. If, additionally, the shape is forced to satisfy the boundary conditions, this method would solve the TPVBP. Moreover, they are suitable as initial guesses for direct collocation methods, as supported by Wall and Conway (2009) and shown by Morante et al. (2019).

The problem is choosing a suitable function to ensure low propellant consumption. Shaped-based approaches are usually employed in interplanetary transfers (rendezvous), assuming the gravitational attraction of the Sun as the only acceleration from the environment. As such, generalized spirals (Morante et al., 2019) or exponential sinusoid shapes (Petropoulos, 2004) (Wall and Conway, 2009) are enough to find good enough initial guesses. De Pascale and Vasile (2006) take as shaping functions first order deviations of the MEE. However, none of these shapes are general enough to tackle transfers between periodic solutions in the CR3BP.

Transfer shapes can be generalized by using a Fourier series to describe the desired shape. This is done by Taheri and Abdelkhalik (2016) for interplanetary transfers. The first twelve coefficients would be forced to analytically satisfy the twelve boundary conditions (initial and final position and velocity) and the rest of the coefficients (as many as desired for generality) would be used to optimize the propellant mass. Something similar is done by Gondelach and Noomen (2015), where the velocity profile is shaped instead of the position (*hodographic shaping*), but the chosen functions are less general.

Heuristic optimization would be suitable for these methods to optimize the extra coefficients. However, as seen in the work by Taheri and Abdelkhalik (2016), relatively simple shapes already require >20 optimizable coefficients. Thus. the problem may become too large for heuristic optimization when complex transfers are considered. As the velocity profiles might have simpler shapes than the position, it might be interesting to combine the methods from Taheri and Abdelkhalik (2016) and Gondelach and Noomen (2015) into a holographic shaping method that uses Fourier series to shape the velocity profile. Nonetheless, these methods pose other problems such as the inclusion of thrust constraints since the required thrust cannot be predicted in advance, the lack of coasting arcs, or the variable non-zero thrust profile (not *bang-bang*[2]), therefore leading to a suboptimal trajectory.

4.4.5 COV-Based Methods

As explained by Yang (2009), the method of Calculus Of Variations (COV), explained in Subsection 4.5.3, is used to obtain the optimal thrusting direction as a function of the costate parameters. However, the costate equations are not used to determine the behavior of the costate parameters as a function of time, instead, the trajectory is discretized at specific points in time, optimizing the costates at each discretization point via Non-Linear Programming (NLP). As such, at each discretization node, the costate parameters required to compute the optimal thrusting direction are estimated. Between the nodes, the estimated parameters are interpolated. The optimization must also satisfy the boundary constraints. With this strategy, the small radius of convergence of indirect methods is greatly increased, and it has fewer optimizable parameters than direct collocation methods (Subsection 4.5.2), without the need for a trajectory as initial guess.

In the article, these equations are derived in terms of the MEE, only targeting the first five, being thus unsuitable for the TPBVP considered if not adapted. Furthermore, no coasting mechanism is included in this method, leading to suboptimal trajectories. Lastly, the number of design variables would be too large for a reasonably refined mesh, being then incompatible with heuristic optimization.

[2] *bang-bang* control switches between two states: *off* and *on* at maximum power.

4.4.6 Neural-Control

Carnelli et al. (2009), based on the work of Dachwald (2004), propose a direct control law governed by machine learning. By means of a feedforward Artificial Neural Network (ANN) of three layers (input, output, and a single hidden layer), the best thrust magnitude and direction can be obtained as a function of the initial and target states, the current distance to the target, and the propellant mass, for instance. This strategy is extremely versatile, as the law itself does not require any information on the system dynamics. However, this means that the law does not take advantage of them either.

The weights for each of the neurons are optimized using heuristic optimization (an evolutionary algorithm in the article), from which the term *evolutionary neurocontroller* comes. Unfortunately, as the solution is not known a priori, only the fitness (whether a certain trajectory is better than the rest), the problem is a so-called *delayed* machine learning problem. Consequently, the ANN cannot be trained, but the weight optimization must be carried out for each specific transfer.

In addition, the number of design variables (optimizable weights) increases proportionally to the number of weights and inputs. As such, considering that a simple interplanetary trajectory requires 30 neurons (Carnelli et al., 2009), the design variables would exceed 100. As there is not enough literature on this method applied to low-thrust transfers, it is difficult to assess how many would be needed to achieve promising results for the problem considered in this thesis, let alone to converge to a minimum. Nonetheless, probably the number would be too high for the OGT to handle effectively.

4.5 Optimal Control

In a general sense, the optimal control problem is an optimization problem in which the control input as a function of time $u(t)$ is chosen along the time interval $[t_0, t_f]$ to minimize a performance index[3] (cost function) J of the form (Lewis et al., 2012):

$$J = \phi(\boldsymbol{x}_f, t_f) + \int_{t_0}^{t_f} L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)dt, \tag{4.5}$$

where ϕ is the final cost function and L is the integral cost function (both scalars). This minimization problem is subject to the state equations (EOM) and the boundary conditions (assuming a set initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{4.6}$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), \boldsymbol{u}(t_f), t_f) = 0 \tag{4.7}$$

Additional equality and inequality constraints as a function of time can of course be included in the problem, but this simplified version is enough to explain the method and solve the TPBVP in this thesis.

Common performance indexes are the minimum time problem ($J = \int_{t_0}^{t_f} 1dt$) or the minimum control effort problem ($J = \int_{t_0}^{t_f} \boldsymbol{u}^T\boldsymbol{u}dt$), which in the case of this thesis is equivalent to minimizing the required propellant mass. Finding a global optimum for the optimal problem thus leads to obtaining the best possible trajectory for the given objective. However, since the design variables are the input vector at every point in time, the problem has an infinite number of optimizable parameters. To solve this issue, the conditions for (local) optimality can be derived as a function of time from the performance index, the EOM, and the boundary conditions as in indirect methods, or the trajectory and control history can be discretized to have a finite number of design variables as in direct methods.

[3]This is the case for continuous systems, discrete systems would have a different formulation (Poulsen, 2012)

4.5.1 Dynamic Programming

As explained by Morante et al. (2021) and Kelly (2015), dynamic programming consists of solving the Hamilton-Jacobi-Bellman (HJB) equations [4], which discretize the complete state space to obtain the globally optimal solution. Moreover, once a solution is obtained, the control inputs are precomputed, leading to a closed-loop control approach instead of open-loop (being thus also suited for stochastic problems). A more thorough elucidation is provided by Poulsen (2012).

Nevertheless, these advantages come with a large drawback: the curse of dimensionality, i.e., the exponential increase in computational complexity and sparsity of data as the number of dimensions of the state grows. Discretizing a seven-dimensional state space (including the spacecraft mass) and optimizing with such complex equations may well be impractical for many applications. To reduce the toll of the dimensionality issue, techniques such as Differential Dynamic Programming were developed, however, this method locally approximates the value function, therefore not converging to the global optimum anymore. In addition, as concluded by Morante et al. (2021), this method does not have a considerable heritage yet in trajectory optimization. Lastly, this is by far the most complex method to use, being more complicated than the one used by ASTOS itself (direct collocation), so the risks associated with using this method just to generate an initial guess may outweigh the benefits, especially because the suitability of dynamic programming for multi-objective optimization still would need to be explored.

4.5.2 Direct Methods

Direct methods are based on the discretization of the states and controls of a continuous trajectory to reduce the optimal control problem with infinite design variables to a Non-Linear Programming (NLP) problem with finite variables. With N_d being the total number of discretization nodes, the design variable vector, p, adopts the following shape:

$$\boldsymbol{p} = \begin{pmatrix} \boldsymbol{x}_1^T & \boldsymbol{u}_1^T & \boldsymbol{x}_2^T & \boldsymbol{u}_2^T & \dots & \boldsymbol{x}_{N_d}^T & \boldsymbol{u}_{N_d}^T \end{pmatrix}^T \tag{4.8}$$

As such, the calculus of variations (Subsection 4.5.3) is not used and the Karush-Kuhn-Tucker (KKT) conditions are solved instead. Hence, the problem is effectively transformed into a static optimization problem. The obtained local optima tend towards the optimal control problem local optima as the number of discretization points tends to infinity. These methods require a trajectory as initial guess, which ought to be discretized to initialize the optimization.

Direct Collocation

As explained by Parrish et al. (2016) and Mingotti et al. (2007), in which direct collocation is used to compute optimized transfers in the CR3BP, direct collocation aims to replace the propagation of the EOM with a continuous function that satisfies the dynamic constraints at a set of nodes (the discretization points). Therefore, the differential defects at each grid point, i.e., the difference between the analytical derivative of the continuous function and the actual differential EOM, are added as equality constraints (instead of the continuous dynamic constraint of Equation (4.6)). Two examples of common direct collocation methods are Hermite-Simpson approximation and Legendre Pseudospectral approximation, which approximates each element of the state and control at the nodes by an $N_d{}^{\text{th}}$ order Lagrange polynomial.

[4] The HJB equations use a different performance index based on the recursion of the one given by Equation (4.5)

Sequential Convex Optimization

Sequential Convex Programming (SCP) aims to "convexify" the highly non-linear problem by making both the performance index and the constraints convex in order to make the local optimization considerably more efficient than for collocation methods. Consequently, as described by Kayama et al. (2021), the EOM are linearized around a reference trajectory $\boldsymbol{x}_{ref}$ as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_{ref}}) + \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_{ref}) + \boldsymbol{B}\boldsymbol{u} \tag{4.9}$$

being $\boldsymbol{A}$ and $\boldsymbol{B}$ generic matrices. Then, the discretized trajectory can be put in the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{c}_k \tag{4.10}$$

where $\boldsymbol{c}$ is a generic vector. Moreover, being h the sampling step for the time variable, the performance index, J, can also be convexified by

$$J = \sum_{k=1}^{N_d} ||\boldsymbol{u}_k|| h_k \tag{4.11}$$

However, this approximation of the actual non-linear problem only holds close enough to the reference trajectory, meaning that a local minimum found for the convexified problem will not be a local minimum in the actual problem. To solve this issue, it is necessary to iterate by using each optimized trajectory as reference trajectory for a subsequent optimization (successive convex optimization) until the trajectory satisfies the non-linear dynamics.

This is the approach used by Kayama et al. (2021) and Hiraiwa et al. (2022) to find optimized trajectories between halo orbits of the same family, leading to very promising results. This method has also been applied to interplanetary transfers in a similar fashion by Z. Wang and Grant (2018) or Morelli et al. (2021). The latter showed faster convergence and higher robustness to poor initial guesses by using Legendre-Gauss-Lobato discretization instead of Equation (4.10) as discretization scheme, thus combining SCP with collocation methods.

Both collocation methods and SCP display similar advantages and disadvantages, although SCP appears to be more efficient, sacrificing robustness. The biggest advantage is the fact that they obtain excellent approximations of the optimal control problem's local optima without the need for the derivation of the often complicated COV equations. Thus, these methods can be used in very different dynamical systems with few modifications. Nonetheless, as this work takes place solely in the context of the CR3BP, the COV equations can be analytically obtained without excessive effort, and there is no need to adapt them to different contexts. The main drawback of these methods is that, since thousands of design variables are usually required to obtain an accurate optimal trajectory, they are unsuited for heuristic optimization, and thus highly dependent on the initial guess (as proved by Parrish et al. (2016), for instance). Because the purpose of the OGT is to provide a wide variety of promising initial guesses as a function of the TOF to be further optimized with collocation methods using the ASTOS software, it would be rather ineffective to use direct methods to also obtain the initial guesses, as they show the same disadvantages as ASTOS would regarding global optimality.

4.5.3 Indirect Methods

As previously mentioned, indirect methods employ COV to derive the necessary conditions for a local minimum constrained by the specific problem dynamics. Instead of the KKT conditions, Pontryagin's Minimum Principle (PMP) is used, based on the minimization of the problem's Hamiltonian and guaranteeing a minimum of the performance index (not simply an approximation like in direct methods). The quality of the solution does however depend on the truncation error when propagating the equations of motion and the costate parameters.

By assuming t_0 and $\boldsymbol{x}_0^*$ to be fixed and $\boldsymbol{x}(t_0) = \boldsymbol{x}_0 = \boldsymbol{x}_0^*$ to be automatically satisfied, the initial boundary constraint in Equation (4.1) can be neglected. Following the derivation provided by Lewis et al. (2012), by treating the EOM as equality path constraints to be satisfied at all times, and assuming the final boundary conditions $\boldsymbol{\psi}$ to be independent of $\boldsymbol{u}$, the *augmented performance index* or *Lagrange function* J', built upon Equation (4.5), is expressed as

$$J' = \phi(\boldsymbol{x}_f, t_f) + \boldsymbol{\nu}^T \boldsymbol{\psi}(\boldsymbol{x}_f, t_f) + \int_{t_0}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \left[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \dot{\boldsymbol{x}}\right] dt \tag{4.12}$$

The subscript "f" denotes the final value and $\boldsymbol{\nu}$ is the Lagrange multiplier vector of the boundary conditions, whereas $\boldsymbol{\lambda}$ corresponds to the Lagrange multiplier vector of the dynamic constraints, often referred to as *costate* parameters or *adjoint* multipliers. Then, defining the *Hamiltonian function* H as

$$H(\boldsymbol{x}, \boldsymbol{u}, t) = L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{4.13}$$

the augmented performance index can be rewritten:

$$J' = \phi(\boldsymbol{x}_f, t_f) + \boldsymbol{\nu}^T \boldsymbol{\psi}(\boldsymbol{x}_f, t_f) + \int_{t_0}^{t_f} H(\boldsymbol{x}, \boldsymbol{u}, t) - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}} dt \tag{4.14}$$

Now, making use of the *fundamental lemma of calculus of variations* and *Leibniz's rule for functionals*, and manipulating, the increment of the augmented performance index, dJ', as a function of increments in $\boldsymbol{x}$, $\boldsymbol{u}$, t (as the final time is free and thus optimizable), $\boldsymbol{\lambda}$, and $\boldsymbol{\nu}$ can be obtained:

$$\begin{aligned} dJ' = &\left(\frac{\partial \phi}{\partial \boldsymbol{x}} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}} - \boldsymbol{\lambda}\right)^T d\boldsymbol{x}\big|_{t_f} + \left(\frac{\partial \phi}{\partial t} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial t} + H\right) dt\big|_{t_f} \\ &+ \boldsymbol{\psi}^T\big|_{t_f} d\boldsymbol{\nu} + \int_{t_0}^{t_f} \left[\left(\frac{\partial H}{\partial \boldsymbol{x}} + \dot{\boldsymbol{\lambda}}\right)^T \delta \boldsymbol{x} + \frac{\partial H}{\partial \boldsymbol{u}}^T \delta \boldsymbol{u} + \left(\frac{\partial H}{\partial \boldsymbol{\lambda}} - \dot{\boldsymbol{x}}\right)^T \delta \boldsymbol{\lambda}\right] dt \end{aligned} \tag{4.15}$$

Then, the necessary condition for a local minimum with equality constraints is that the solution must be a stationary point of the augmented performance index. Therefore, by setting $dJ' = 0$ for all independent increments in its arguments, i.e., setting the coefficients of the differentials to zero, the necessary condition is given by the Euler-Lagrange equations:

$$State\ \ Equation \qquad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{4.16}$$

$$Costate\ \ Equation \qquad -\dot{\boldsymbol{\lambda}} = \frac{\partial H}{\partial \boldsymbol{x}} \tag{4.17}$$

$$Stationarity\ \ Condition \qquad \boldsymbol{0} = \frac{\partial H}{\partial \boldsymbol{u}} \tag{4.18}$$

the boundary conditions, which are split into:

$$\boldsymbol{x_0} = \boldsymbol{x}_0^* \qquad\qquad \boldsymbol{\psi} = 0 \tag{4.19}$$

and the *transversality* conditions[5]:

[5] $\boldsymbol{x}_f$ is assumed to be independent of t_f.

$$\boldsymbol{\lambda}_f = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_f} + \boldsymbol{\nu}^T \left.\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{t_f} \tag{4.20}$$

$$\left.\frac{\partial \phi}{\partial t}\right|_{t_f} + \boldsymbol{\nu}^T \left.\frac{\partial \boldsymbol{\psi}}{\partial t}\right|_{t_f} + H_f = 0 \tag{4.21}$$

However, in real-life problems, the input vector is usually constrained to a set of *admissible* inputs $\mathcal{U}$. For this kind of problem, Pontryagin's Minimum Principle is applied, stating that, when the input is constrained, a minimum can only be obtained by choosing the admissible input parameters that minimize the Hamiltonian function at all times. Henceforth, the stationarity condition, Equation (4.18), must be replaced with

$$\boldsymbol{u} = arg \min_{\boldsymbol{u} \in \mathcal{U}} H \tag{4.22}$$

The costate equation, Equation (4.17), dictates the behavior of the costate parameters as a function of time. This means that, if they are integrated together with the EOM, Equations (4.16) and (4.17) will be automatically satisfied. With the aforementioned equations, the optimal control problem can be solved with conventional local optimization methods by estimating the initial value of the costate parameter vector. As such, instead of *directly* optimizing $\boldsymbol{u}$ (as in direct methods), the input is *indirectly* optimized via the initial costate parameters.

The most difficult part of implementing indirect methods is analytically obtaining the partial derivatives and choosing the control inputs as a function of time such that the Hamiltonian is minimized. Nonetheless, for transfers between periodic solutions in the CR3BP, the required equations were successfully derived by numerous authors, such as Stuart et al. (2010), Ozimek and Howell (2010) or X. Pan and B. Pan (2020). Moreover, they have been obtained for transfers between halo orbits within the four-body problem by P. Zhang et al. (2013) (SEL halo to MEL halo) and Pergola et al. (2009) (Earth halo to Mars halo), even using manifold theory.

Another key drawback of indirect methods is the small radius of convergence of the method due to the strong non-linearities and the discontinuous thrust profile. This means that the provided initial guess ought to be close enough to a minimum for the method to arrive at a solution that satisfies the conditions for a local minimum in a finite number of iterations. To solve this issue, some authors, such as Oberle and Grimm (1989), propose multiple shooting, whereas X. Pan and B. Pan (2020) employ homotopy techniques to avoid divergence. These techniques consist of solving an easier optimization problem and using the obtained solutions to solve gradually more restrictive problems until the actual optimal control problem is reached. A simple example would be to increase the maximum allowable thrust and keep reducing it until the actual spacecraft maximum thrust is used. However, as the number of design variables is greatly reduced compared to direct methods since the only optimizable parameters are the TOF and the seven initial costate parameters (in the case of a single-shooting approach), indirect methods are suitable for heuristic optimization, for which having a narrow convergence radius is in principle not an issue.

Heuristic approaches have been successfully used for interplanetary transfers by Sentinella and Casalino (2006), and Pontani and Conway (2010), for the four-body problem by Pergola et al. (2009), and for the CR3BP by Lee et al. (2005). The latter employs EA and SA for multi-objective optimization, proving the suitability of indirect methods for MO optimization, also explored by Coverstone-Carroll et al. (2000) for interplanetary transfers. In addition, most of these authors, like Pergola et al. (2009), employ parameters, such as the TOF in the manifolds or the starting point in the initial periodic solution, that cannot be analytically included in the transversality conditions. For such a problem, the equations explained in this section only guarantee a local minimum for the TPBVP, not for the whole problem with the extra parameters. Nonetheless, convergence to the real minima can

still be achieved with local/heuristic optimizers in the conventional way, thus disregarding the complicated and often sensitive transversality conditions. This method would be considered a *mixed* indirect optimization approach.

Nevertheless, constraint handling is one of the weak points of heuristic optimization. This is a problem in indirect methods since the boundary and transversality conditions must be satisfied, the former to obtain a feasible trajectory, and the latter to ensure a minimum. To solve this, hybrid optimization (Subsection 3.3.3) is generally used. Pergola et al. (2009) propose to simply include the constraints in the performance index with weighting factors, and then perform a subsequent local optimization with the best solution. On the contrary, Coverstone-Carroll et al. (2000) locally optimize each of the individuals in the population for a set number of iterations, and the convergence capabilities of the iterated solutions are used for the heuristic optimization. Sentinella and Casalino (2006) deem the previous strategy rather inefficient, as most of the individuals will not converge to a better solution after a few iterations. Nonetheless, Sentinella and Casalino (2006) suggest only using heuristic optimization to satisfy the boundary conditions and then using local optimization on the obtained solutions to converge to a minimum.

Olympio (2008) proposes an interesting indirect optimization method for N-bodies, centered around the Sun. This method converges by means of an extended STM, instead of with optimization methods. Making use of an indirect method for the control law, and an iterative single-shooting differential correction to meet the transversality conditions, a minimum is ensured. The biggest problem is that the iterative procedure rarely converges, especially when the trajectory is rather long or gets close to a body. This is solved by a so-called *gradient method*[6]. In a very rough way, a second-order Lagrangian is introduced, with second-order optimality conditions, and guaranteed solutions are obtained by a set of equations that require a forward and a backward integration run. Although the convergence capabilities of this gradient method are outstanding with respect to not diverging, convergence is considerably slow, which would be a problem when using heuristic optimization to optimize the manifolds, for instance. Moreover, the obtained trajectory is dependent on the initial guess (which would need to be optimized, adding even more computational cost) and the mathematics behind the method are extremely cumbersome. For all these reasons, the gradient method is discarded.

4.6 Trade-off Summary

Overall, the following conclusions were drawn from the literature study regarding the suitability of the explored methods to the given problem:

- Rough initial guesses can be discarded because, although their simplicity is outstanding, the quality of the trajectories would be very poor and the trajectories would be far from feasible.
- Blended and Lyapunov control laws are not suitable as they cannot target all position and velocity components required to enter the final orbit.
- Neural-control laws are not considered for this work due to the lack of heritage in orbit design and the high number of optimizable parameters required (being thus inadequate for heuristic optimization).
- Although feasible trajectories can in principle be easily obtained with shape-based methods, they do not include effective coasting mechanisms (required for MO optimization), they may require too many optimizable parameters, and the solution performance is not predicted to be remarkable.

[6]Not to be confused with gradient-based local optimization methods.

- Direct methods can be discarded for their incompatibility with heuristic optimizers due to the large number of design variables. Direct collocation will anyway be used for further optimization.
- There are numerous ways of solving the TPBVP with indirect methods. They have shown to be the most promising to solve the problem at hand due to their reduced number of design variables, their suitability for multi-objective heuristic optimization (thus also removing the need for an initial guess), and the superior quality of the attainable solutions. The latter is true because, if convergence is achieved, they guarantee a local minimum for the TPBVP. Nevertheless, difficulties may arise especially in finding trajectories that satisfy the constraints and in ensuring convergence to actual Pareto-optimal solutions.

Mixed Indirect Optimization Approach 5

Having chosen indirect methods to solve the TPBVP, the next step is to derive the relevant equations specifically for the problem at hand, which is done in Section 5.1. Then, Section 5.2 explains the optimization procedure for the complete problem, with its design variables and constraints. Lastly, the implementation of the complete method within the OGT is discussed in Section 5.3, including the new additions to the software and their testing.

5.1 Calculus of Variations

To solve the TPBVP, the objective function and the constraints must be defined, and Equations (4.13), (4.16), (4.17), and (4.20) through (4.22) ought to be derived for the problem at hand[1]. The chosen equations are mostly based on the work by X. Pan and B. Pan (2020).

Defining the state vector, $\boldsymbol{x}$, and the input vector, $\boldsymbol{u}$, as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{r}^T & \boldsymbol{v}^T & m \end{pmatrix}^T \qquad \boldsymbol{u} = \begin{pmatrix} \tau & \hat{\boldsymbol{u}}^T \end{pmatrix}^T \tag{5.1}$$

the state vector derivative with respect to time can be obtained with the EOM in Section 2.3:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \implies \begin{cases} \dot{\boldsymbol{r}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = \boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{c}(\boldsymbol{v}) + \dfrac{T_{max}\tau}{m}\hat{\boldsymbol{u}} \\ \dot{m} = -\dfrac{T_{max}\tau}{I_{sp}g_0} \end{cases} \tag{5.2}$$

$$\boldsymbol{g}(\boldsymbol{r}) = \begin{pmatrix} \frac{\partial \Omega}{\partial x} & \frac{\partial \Omega}{\partial y} & \frac{\partial \Omega}{\partial z} \end{pmatrix}^T \tag{5.3}$$

$$\boldsymbol{c}(\boldsymbol{v}) = \begin{pmatrix} 2v_y & -2v_x & 0 \end{pmatrix}^T \tag{5.4}$$

As explained in Subsection 4.5.3, if t_0 is chosen to be set, and $\boldsymbol{x}_0 = \boldsymbol{x}_0^*$ is known and fixed, then, the second constraint in Equation (4.1) can be excluded. Acknowledging the final mass to be free, the optimal control problem can be thus formulated as

$$\begin{aligned} \underset{\boldsymbol{u}(t)}{min} \quad & \frac{T_{max}}{I_{sp}g_0}\int_{t_0}^{t_f} \tau dt \\ subject\ to \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ & \boldsymbol{\psi}_f = \begin{pmatrix} \boldsymbol{r}_f - \boldsymbol{r}_f^* \\ \boldsymbol{v}_f - \boldsymbol{v}_f^* \end{pmatrix} = 0 \\ & \boldsymbol{u} \in \mathcal{U} \end{aligned} \tag{5.5}$$

In the proposed problem, $\phi = 0$ and L does not explicitly depend on time, greatly simplifying the equations. The Hamiltonian,H, of the optimization problem is

$$\boldsymbol{\lambda}_r = \begin{pmatrix} \lambda_x & \lambda_y & \lambda_z \end{pmatrix}^T \qquad \boldsymbol{\lambda}_v = \begin{pmatrix} \lambda_{vx} & \lambda_{vy} & \lambda_{vz} \end{pmatrix}^T \tag{5.6}$$

[1] These equations constitute the necessary and sufficient conditions for a local minimum.

$$H = \boldsymbol{\lambda}_r^T \boldsymbol{v} + \boldsymbol{\lambda}_v^T \left[\boldsymbol{g}(\boldsymbol{r}) + \boldsymbol{c}(\boldsymbol{v}) + \frac{T_{max}\tau}{m}\hat{\boldsymbol{u}} \right] - \lambda_m \frac{T_{max}\tau}{I_{sp}g_0} + \frac{T_{max}\tau}{I_{sp}g_0} \tag{5.7}$$

λ_x corresponds to the component of the Lagrange multiplier vector $\boldsymbol{\lambda}$ that is multiplied with the EOM for x, i.e., the first component of the state derivative vector, $\dot{\boldsymbol{x}}$. The same logic is applied to the remaining costate parameters. This Hamiltonian yields the following costate differential equations (using the nomenclature from Equation (2.31)), which are integrated together with Equation (5.2) via a direct single-shooting approach:

$$\begin{cases} \dot{\boldsymbol{\lambda}}_r = -\dfrac{\partial H}{\partial \boldsymbol{r}} = -\boldsymbol{\Omega}^T \boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\dfrac{\partial H}{\partial \boldsymbol{v}} = -\boldsymbol{\lambda}_r - \boldsymbol{\Gamma}^T \boldsymbol{\lambda}_v \\ \dot{\lambda}_m = -\dfrac{\partial H}{\partial m} = \dfrac{T_{max}\tau}{m^2} \boldsymbol{\lambda}_v^T \hat{\boldsymbol{u}} \end{cases} \tag{5.8}$$

Now, according to the PMP (Equation (4.22)), the control vector $\boldsymbol{u}$ shall be chosen such that the Hamiltonian is minimized at all times. By only including the thrust ratio $\tau \in [0, 1]$ and its direction unit vector $\hat{\boldsymbol{u}}$ as input, the thrust can never exceed T_{max}, being thus always within the admissible set. It is clear from Equation (5.7) that to minimize the Hamiltonian the thrust must be parallel to $\boldsymbol{\lambda}_v$ and in the opposite direction. Hence,

$$\hat{\boldsymbol{u}} = -\frac{\boldsymbol{\lambda}_v}{||\boldsymbol{\lambda}_v||} \tag{5.9}$$

Then, since Equation (5.7) is linear with τ, the thrust must be set to its maximum value when the coefficient multiplying τ is negative, and to its minimum value when it is positive. This renders bang-bang control as the optimal strategy, governed by the switching function S (the coefficient):

$$\tau = \begin{cases} 1 & S < 0 \\ 0 & S > 0 \end{cases} \tag{5.10}$$

$$S = 1 - \lambda_m - \frac{I_{sp}g_0}{m}||\boldsymbol{\lambda}_v|| \tag{5.11}$$

in which the optimal $\hat{\boldsymbol{u}}$ is of course assumed. This control strategy presents a singularity when $S = 0$, making the optimal thrust magnitude undefined. Nevertheless, this singularity can be neglected, as suggested by X. Pan and B. Pan (2020), and P. Zhang et al. (2013), since this phenomenon only occurs at finite isolated points.

From these equations, one can observe that the mass costate parameter, λ_m, only plays a role in the thrust magnitude, τ, not in the thrust direction. As such, this design variable does not modify the trajectory unless S changes sign (only if coasting periods are present). Furthermore, if it were not for λ_m and S, the output trajectory would not depend on the specific initial values of the costate parameters, but on their ratio instead, removing one degree of freedom. This is because the first two equations of Equation (5.8) would be scaled consistently, and the costate values are normalized in Equation (5.9), removing the effect of the scaling. Another noteworthy conclusion is the fact that, according to Equation (5.8), λ_m is either constant or decreases monotonously (derivative always negative) if the optimal $\hat{\boldsymbol{u}}$ is used, since

$$\dot{\lambda}_m = \frac{T_{max}\tau}{m^2} \boldsymbol{\lambda}_v^T \hat{\boldsymbol{u}} = -\frac{T_{max}\tau}{m^2}||\boldsymbol{\lambda}_v|| \tag{5.12}$$

in which all variables are positive or zero.

Lastly, with the provided performance index and boundary conditions, noticing that $\boldsymbol{\psi}_f$ is not an explicit function of t_f, the transversality conditions (Equations (4.20) and (4.21)) simplify to:

$$\begin{cases} \boldsymbol{\lambda}_{rf} = \boldsymbol{\nu}_r \\ \boldsymbol{\lambda}_{vf} = \boldsymbol{\nu}_v \\ \lambda_{mf} = 0 \end{cases} \tag{5.13}$$

$$H_f = 0 \tag{5.14}$$

In these equations, $\boldsymbol{\nu}_r$ and $\boldsymbol{\nu}_v$ are the boundary condition Lagrange multiplier vectors corresponding to $\boldsymbol{r}$ and $\boldsymbol{v}$, respectively. These vectors do not interfere with the rest of the equations, so the equations for $\boldsymbol{\lambda}_{rf}$ and $\boldsymbol{\lambda}_{vf}$ can be disregarded as long as $\boldsymbol{\lambda}_{r0}$ and $\boldsymbol{\lambda}_{v0}$ are chosen such that the boundary constraints $\boldsymbol{\psi}_f$ are satisfied. With these equations and a direct single-shooting approach, the dynamic optimization problem can be solved with conventional optimization methods, with the optimizable parameters being the initial costate parameter vector ($\boldsymbol{\lambda}_0$). Whereas $\boldsymbol{\psi}_f = 0$ is required to obtain a feasible trajectory, the transversality conditions for λ_{mf} and H_f only guarantee that the trajectory is a local minimum with respect to the mass state and the TOF, respectively. Therefore, if these two conditions are not satisfied by a certain trajectory, but the rest are, the trajectory will still be feasible.

5.2 Mixed Optimization Logic

5.2.1 Optimization Method and Design Variables

The procedure described in Section 5.1 can only ensure local minima of the TPBVP with fixed initial and final points. In this research, the aim is to connect the initial and target orbits utilizing any of their points. Moreover, to take advantage of manifold theory, the transfers will employ unstable manifolds to depart from the target orbit and stable manifolds to converge toward the target orbit, with these manifolds being connected by the explained optimal control strategy. This is predicted to increase design freedom, easing the obtainment of solutions and enabling the finding of better solutions. As such, the transfer strategy will consist of five phases in sequence. A sketch of the five phases is displayed in Figure 5.1. According to the figure, the spacecraft leaves the initial halo orbit via an unstable manifold, subsequently entering an optimal control phase, i.e., a thrusting phase governed by the control law of the indirect method. Then, the optimal control phase connects to a stable manifold that converges to the target halo orbit.

Hence, in the complete problem, the required initial and final positions and velocities of the optimal control phase are optimizable as well and the described indirect optimization method does not ensure a minimum of the full problem. Since the necessary initial and final states are functions of parameters such as the initial orbit point and the TOF in the manifolds, extra transversality conditions arise from these parameters that are often computed numerically due to their difficult analytical derivation. For instance, the transversality condition arising from the variation in $\boldsymbol{r}_f^*$ and $\boldsymbol{v}_f^*$ in Equation (5.5) due to the dimensional TOF in the target manifold, T_{man_2}, would be (Stuart et al., 2010):

$$\begin{pmatrix} \boldsymbol{\lambda}_r^T & \boldsymbol{\lambda}_v^T \end{pmatrix} \begin{pmatrix} \frac{\partial \boldsymbol{r}_f^*}{\partial T_{man_2}} TU \\ \frac{\partial \boldsymbol{v}_f^*}{\partial T_{man_2}} TU \end{pmatrix} = \begin{pmatrix} \boldsymbol{\lambda}_r^T & \boldsymbol{\lambda}_v^T \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_f^* \\ \dot{\boldsymbol{v}}_f^* \end{pmatrix} = \mathbf{0} \tag{5.15}$$

whereas the one due to the point at which the manifold attaches to the target orbit, τ_{orb_2}, would be

$$\begin{pmatrix} \boldsymbol{\lambda}_r^T & \boldsymbol{\lambda}_v^T \end{pmatrix} \begin{pmatrix} \frac{\partial \boldsymbol{r}_f^*}{\partial \tau_{orb_2}} \\ \frac{\partial \boldsymbol{v}_f^*}{\partial \tau_{orb_2}} \end{pmatrix} = \mathbf{0} \tag{5.16}$$

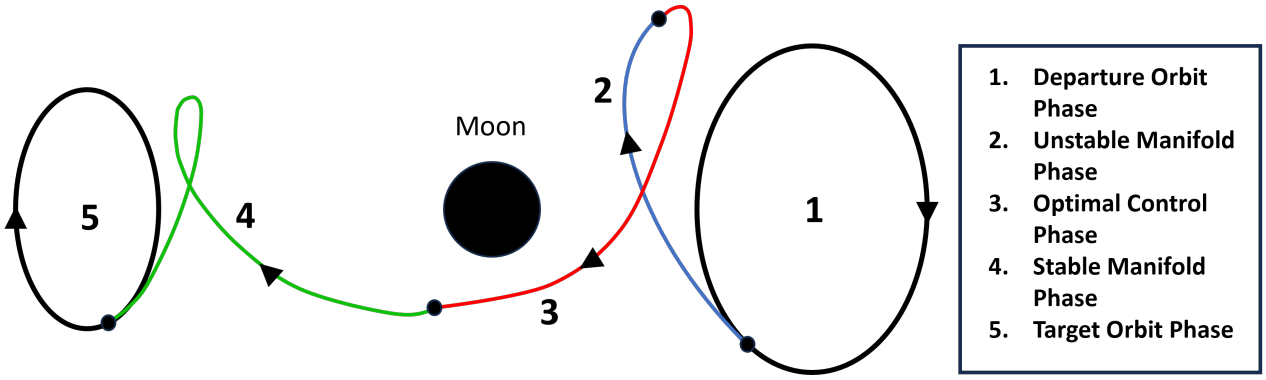

Figure 5.1: Sketch of the five phases employed in this research to optimize transfers between halo orbits.

The aforementioned transversality conditions are often considerably sensitive equations. Numerous authors, such as Pergola et al. (2009) and Ozimek and Howell (2010), solve this issue by neglecting the transversality conditions (also Equations (5.13) and (5.14)), *directly* minimizing the performance index instead of a root-solving process to satisfy all conditions for a minimum. As such, the transversality conditions need not be included because they will eventually be satisfied by optimizing the performance (Ozimek and Howell, 2010) (Lee et al., 2005). For the problem at hand, this idea of disregarding the transversality conditions is also of interest because the goal is to obtain a wide variety of minimum propellant mass solutions as a function of time with a single optimization (minimizing both propellant mass and TOF), such that the mission designer can have numerous alternatives. For this matter, heuristic optimization methods will be employed due to the advantages described in Subsection 3.3.2. Then, having to satisfy the transversality conditions on top of the very constraining boundary conditions may make the optimizer converge (if at all) to only a few solutions. Furthermore, even though these solutions would be rigorous local minima, they might be far from the global minimum, so a heuristic optimization that explores more solutions is predicted to arrive at better results. For this research, a single-shooting approach was chosen to carry out such a task, as the use of multiple-shooting would increase the number of optimizable parameters beyond the capabilities of heuristic optimization. Therefore, the proposed approach is not a pure indirect method, but a mixed (hybrid) direct/indirect optimization method.

The strategy to generate a single trajectory such that it is easier for the optimizer to find trajectories is as follows:

1. Propagate the initial and target orbits for one period.
2. Choose the starting point in the initial orbit, parametrized with respect to its orbital period, $\tau_{orb_1} \in [0, 1]$.
3. Perturb this point in the selected direction of its unstable eigenvector (with $\epsilon_p = 10^{-4}$) and propagate the corresponding unstable manifold for a specific TOF, T_{man}. The mechanism to choose the eigenvector direction is explained in Section 5.3.
4. Choose the final point in the target orbit, also parametrized with respect to its orbital period, $\tau_{orb_2} \in [0, 1]$.
5. Perturb this point in the selected direction of its stable eigenvector and propagate the corresponding stable manifold backward in time for a certain amount of time defined before the optimization.

6. Starting from the end-point of the unstable manifold and with the chosen $\boldsymbol{\lambda}_0$, propagate the so-called *optimal control phase* using the equations from Section 5.1 for a certain amount of time defined before the optimization.

7. Find the closest state between the optimal control trajectory and the stable manifold or the target orbit (Subsection 5.2.3).

8. Adjust the states and times of flight of the optimal control and stable manifold phases accordingly. If the closest state to the optimal control phase lies within the target orbit, the stable manifold phase is removed and τ_{orb_2} is updated.

In this method, it is assumed that the propellant mass required to transition between the manifold and orbit phases is negligible compared to the one required throughout the optimal control phase (Stuart et al., 2010) (P. Zhang et al., 2013). This assumption is also necessary as it is not possible to predict how much mass the spacecraft would still have at the target orbit before propagating the optimal control phase, so the acceleration provided by the thrusters would be unknown and iteration would be required. Nonetheless, the validity of this assumption is explored in Section 8.2.

Therefore, the extra parameters that need to be added to the optimization problem are the initial and target points of the periodic solutions, τ_{orb_1} and τ_{orb_2}, and the TOF in the unstable manifold, T_{man}. Beforehand, the maximum TOFs for the stable manifold and the optimal control phases must be defined. Then, the complete design variable vector would consist of ten parameters:

$$\boldsymbol{p} = \begin{pmatrix} \tau_{orb_1} & T_{man} & \boldsymbol{\lambda}_0^T & \tau_{orb_2} \end{pmatrix}^T \tag{5.17}$$

It should be noted that several trajectory combinations are possible:

- From periodic solution directly to the target periodic solution (case $T_{man} = T_{man_2} = 0$).
- From periodic solution to a stable manifold connecting to the target periodic solution (case $T_{man} = 0$).
- From an unstable manifold arising from the initial periodic solution to the target periodic solution (case $T_{man_2} = 0$).
- From an unstable manifold arising from the initial periodic solution to a stable manifold connecting to the target periodic solution.

Hence, not all the described parameters need to be used for every optimization problem. If the unstable manifold phase is not used, T_{man} would not be needed. Moreover, if the stable manifold phase is disregarded, τ_{orb_2} can be neglected as the complete orbit is used to find the closest state anyway. Consequently, accounting for all combinations, the maximum number of design variables would be ten, and the minimum eight. These numbers, although not small, are suitable for heuristic optimization. Removing manifold phases can be convenient for obtaining faster transfers and for more rapid optimization convergence (fewer design variables). In addition, for marginally stable orbits such as DROs or less unstable periodic solutions such as NRHOs, the benefits of manifold theory disappear.

On a separate note, depending on the mission and the mission design stage, the dry mass of the spacecraft or the mass required at the target orbit may be known instead of the total initial mass at the start of the transfer. As such, to increase design freedom, the optimal control phase can also be propagated backward in time if desired. Thus, the supplied total mass will actually be the final mass, and the optimal control phase will attempt to connect the stable manifold arising from the target orbit to the unstable manifold arising from the initial orbit. In addition, T_{man} would then correspond to the TOF of the stable manifold and T_{man_2} to the TOF of the unstable one.

5.2.2 Boundary Constraints

As mentioned in Subsection 3.3.2 and Subsection 4.5.3, heuristic optimizers may struggle with finding feasible solutions if the constraint handling strategy is not appropriate. Although the constraint handling strategy will be treated in Section 7.1, due to the highly restrictive equality boundary conditions from Equation (5.5), it is crucial to define an appropriate tolerance level to help the optimizer explore a diverse set of trajectories. Having disregarded the transversality conditions, the only constraints required are the six boundary constraints to connect the optimal control phase to the target manifold-orbit structure (three for position and three for velocity).

The trajectories generated by the OGT are meant to be suitable initial guesses for further optimization with ASTOS. Thus, the allowable discontinuities ought to be small enough to not only be able to be corrected with the software without excessive effort but also to ensure that the performance (TOF and required propellant mass) does not significantly change. In this framework, the obtained initial guesses would be considered promising for mission design. As a first estimation for appropriate position and velocity constraint values, considering that the perturbation to enter the manifold phase is already 10^{-4} in non-dimensional units, a value of 10^{-3} for both position and velocity was deemed fitting. If dimensionalized with DU and VU, this value amounts to 384.40 km and 1.02 m/s. It should be noted that this discontinuity would be several orders of magnitude smaller than the ones that usually need to be corrected with the rough initial guesses commonly used with collocation methods, such as the ones employed by Pritchett et al. (2017) or Parrish et al. (2016).

However, close to the Moon, due to the fast varying dynamics, a difference of 384 km may lead to a completely different trajectory. If the connection point between the target manifold and optimal control phase takes place close to a lunar flyby, even when the position and velocity constraints are satisfied, the difference in Jacobi constant (related to the orbital energy in the rotating frame) might be too large to be easily corrected with direct collocation methods. To circumvent this issue, an additional constraint for the Jacobi constant error is introduced, such that the majority of the required orbital energy change is supplied. As the Jacobi constant in the EML_2 halo family ranges between 3.02 and 3.17 (Hiraiwa et al., 2022), a value of 10^{-2} is deemed appropriate. Given this number, problem analysis showed that when the phase connection does not take place near a flyby, the Jacobi constraint is not active. Nevertheless, for some use cases, it may be convenient to revisit this value.

Although most of the optimization tuning in Section 7.1 is carried out with the aforementioned constraint limits, for the OGT optimizations it would be convenient to relax the constraints as much as possible. As will be demonstrated in Section 8.2, the proposed discontinuities are easily corrected with ASTOS without changing the trajectory performance at all. Moreover, Subsection 8.2.2 will show that larger discontinuities can also be rectified without penalizing the performance. Therefore, for the final optimizations and results, the constraints were relaxed to a value of $3{\cdot}10^{-3}$, corresponding to an error of 1153.20 km and 3.07 m/s. To summarize:

- **Position and velocity constraints before relaxation:** 10^{-3} [-]
- **Position and velocity constraints after relaxation:** $3{\cdot}10^{-3}$ [-]
- **Jacobi constant constraint:** 10^{-2} [-]

Lastly, to provide the user with more control over the search areas of the optimizations, two additional constraints are included: one for the maximum allowable propellant mass and another for the maximum time of flight. The values for these constraints are defined before the optimization. Evidently, these two constraints are not close to being as restrictive as the boundary conditions, provided that the selected values are sensible. It should be noted that for the use case explored in this research, the values are chosen such that these two constraints are never active.

5.2.3 Closest State Search Strategy

Due to the extended durations of the optimal control phase and the target phases (manifold and periodic solution), finding an efficient way to retrieve the closest state between the phases is essential. With such demanding constraints, it is expected that a very large number of runs are required for the optimizations to converge. Hence, any reduction in the computational cost of each run is valuable. However, the step size ought to be small enough such that the difference in state error between two adjacent steps is not more than the constraint values. Otherwise, the discretized trajectories will not have enough resolution to satisfy the constraints. For this matter, the maximum step size for the manifold and orbit phases when acting as *target* phases is half of the nominal maximum step size of the OGT. The nominal maximum step size was chosen by Tatay Sanguesa (2021) to be 10^{-3} in non-dimensional units (roughly 375 seconds). Thus, $5{\cdot}10^{-4}$ is the new maximum step size for the target phases. For the optimal control phase, the maximum step is kept at 10^{-3} as the location of these points is more flexible to design variable changes. Interpolation between the steps was explored but deemed unnecessary and too time-consuming.

The general strategy to find the closest state is simply to evaluate, at each step of the optimal control trajectory, the position and velocity errors with respect to every point of the target trajectory (consisting of the target manifold and target orbit phases) and store the minimum value. The overall minimum difference is found by taking the lowest of all stored errors. As iterating through thousands of points in every step of the optimal control phase propagation can be very time-consuming, a coarse search is first carried out by taking (at each step of the optimal control phase) every ten steps of the target trajectory. Once the minimum is obtained, a fine search is performed by evaluating the twenty points surrounding this coarse minimum.

To prove that the maximum step size is appropriate and that taking every ten steps is enough to not miss any close approach, the nominal trajectory that will be presented in Section 6.2 (which is a feasible transfer) was analyzed. At every step of the optimal control phase of the nominal trajectory, the state error with respect to all discretization points of the target trajectory was computed simply by taking the norm of the six-dimensional vector difference in position and velocity. The sharpest minimum of this function, i.e., the minimum with the largest second derivative, is found at the connection point between the phases. Disregarding this point for now, one can find the second-sharpest minimum by examining the second derivative of the state error relative to the rest of the optimal control steps. Figure 5.2 displays the state error between the target trajectory points and the optimal control step that leads to this second-sharpest minimum in state error. From this figure, it can be seen that the local minima are very well described, even by the coarse grid (taking every ten steps). The oscillations follow the general dynamics of manifolds. As manifolds gradually depart from the halo orbit while revolving similarly to the original orbit, appreciable periodicity in the state error is exhibited. Moreover, the first manifold revolution tends to be very similar to the original halo orbit (especially for less unstable halos such as the ones considered in this use case), which explains why the last two minima in the plot are almost identical.

Then, Figure 5.3 shows the state error with respect to the target trajectory taking the actual connection point. It should be noted that in these figures, the portion of the target trajectory that is not part of the final trajectory is not displayed. As would be expected, the error starts from a value close to zero (the connection point). Again, the coarse grid describes the state error evolution accurately, even at the beginning, where the change in state error is most severe. Focusing on the cut-out, one can see that the difference in state error between the first two orange points (the ones describing the minimum) does not exceed $3{\cdot}10^{-3}$. As there are nine additional points in between, several points satisfy the boundary constraints. Furthermore, the computation of this state error groups the position and velocity errors, which is not the case in the constraint computations, making this analysis conservative by a factor of $\sqrt{2}$. In the search for the minimum state error, the following equation is employed:

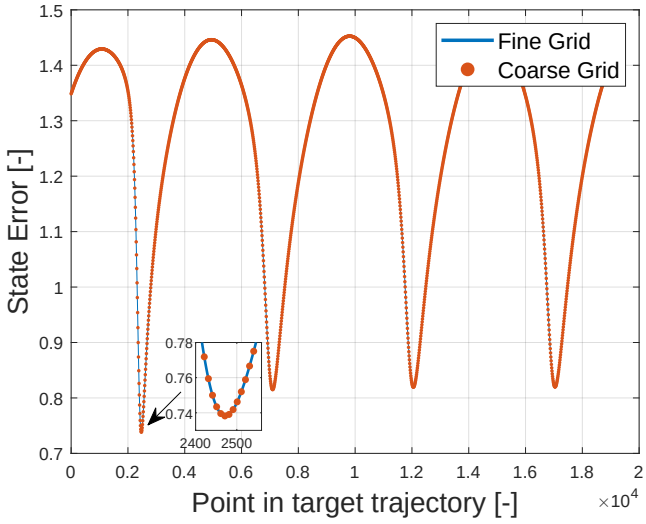

Figure 5.2: State difference between the target manifold and halo discretization points and the optimal control phase point that leads to the sharpest minimum excluding the connection point.

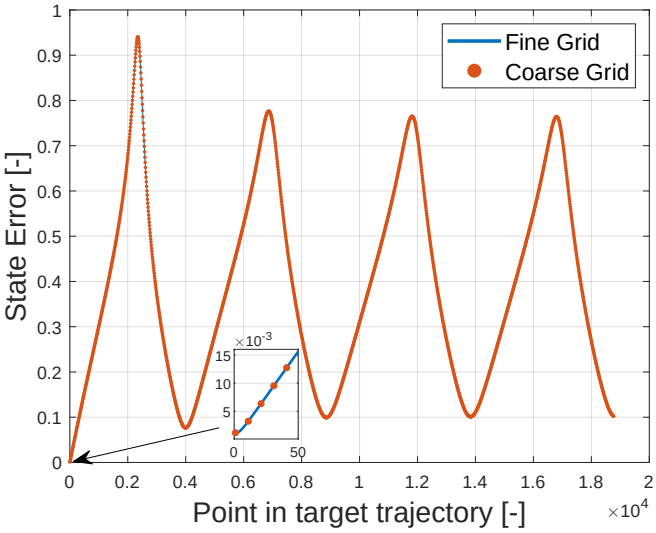

Figure 5.3: State difference between the target manifold and halo discretization points and the optimal control phase point that connects to said target trajectory.

$$\epsilon_s = w_{pos}||\boldsymbol{r} - \boldsymbol{r}_{target}|| + w_{vel}||\boldsymbol{v} - \boldsymbol{v}_{target}|| + |C - C_{target}| \tag{5.18}$$

where ϵ_s is the state error, w_{pos} and w_{vel} are weights, and the subscript "target" refers to the point being evaluated in the target trajectory. The values for the weights will be defined in Section 7.1. Following this nomenclature, the relaxed position, velocity, and Jacobi constant constraints, symbolized by h_{pos}, h_{vel}, and h_c, respectively, are expressed as:

$$\begin{aligned} h_{pos} &= ||\boldsymbol{r} - \boldsymbol{r}_{target}|| - 3 \cdot 10^{-3} \\ h_{vel} &= ||\boldsymbol{v} - \boldsymbol{v}_{target}|| - 3 \cdot 10^{-3} \\ h_c &= |C - C_{target}| - 10^{-2} \end{aligned} \tag{5.19}$$

where the minimum state error between the optimal control phase and the target trajectory according to Equation (5.18) is selected. With these definitions, a trajectory is considered feasible provided that h_{pos}, h_{vel} and h_c are negative.

Further proof that the adopted settings are suitable is the fact that the optimizer can find vast numbers of extremely similar feasible trajectories around the nominal solution by slightly modifying the design variables, even with the tight constraints. However, due to the aforementioned similarity between the target orbit and the first revolution of the stable manifold, on numerous occasions, the coarse search may not accurately identify where the actual minimum is. To solve this and any other special cases, in the coarse search the second-lowest state error point is also stored, and, if it is not adjacent to the lowest, the region corresponding to the second lowest is explored with a fine search as well. After both fine searches (storing the minimum and the second-lowest state errors), if the second-lowest error is still not adjacent to the lowest, the point that leads to the smallest TOF is selected (the propellant mass does not vary for a given point in the optimal control phase).

To further increase the search speed, it is desirable to not evaluate the state error when the spacecraft has a state that is considered "not useful" to find the minimum state error within the trajectory.

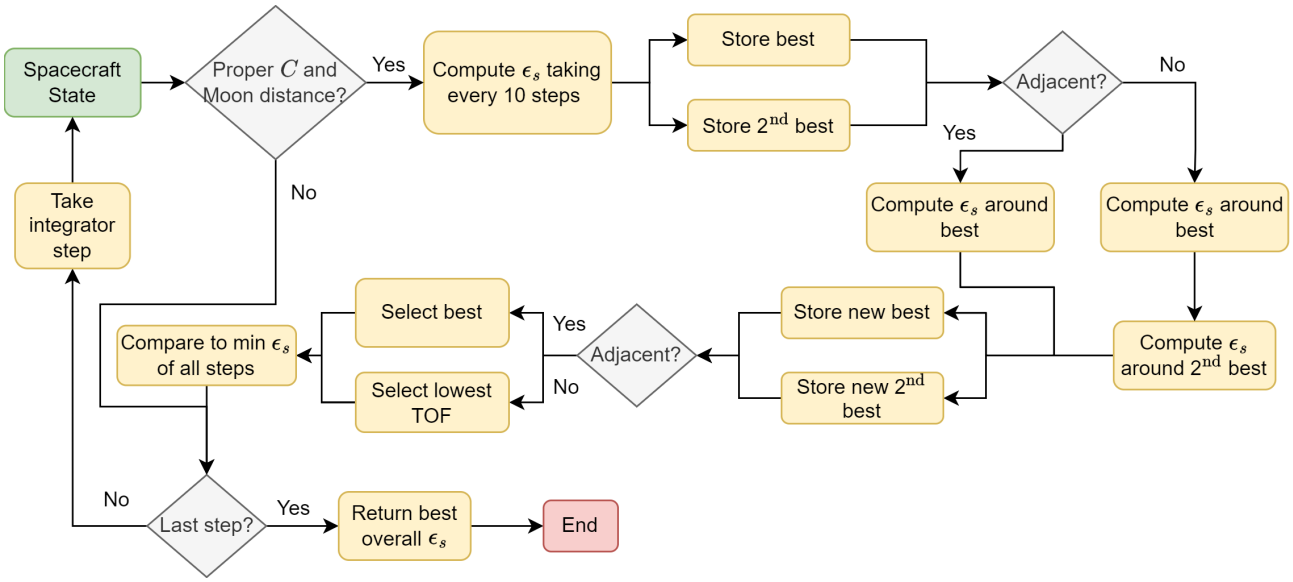

Figure 5.4: Workflow diagram of the strategy to search for the closest state between the optimal control phase and the target trajectory for a single run.

The state error at the initial propagation state is always evaluated. However, when the spacecraft is too far from the Moon or the Jacobi constant is too far from the target value (virtually constant for the complete target trajectory), the search for the closest state is not carried out. A conservative value of 0.3 in non-dimensional units was selected for the Moon distance. In contrast, for the Jacobi constant, the state error is not evaluated unless the difference with the target value is less than $3{\cdot}10^{-2}$. These numbers are intended to be suitable for a general transfer between halo orbits, but may not be appropriate in other cases (such as for DROs). In addition, they should not be too constraining to avoid most of the trajectories returning the initial propagation state as the best state [2], providing no help to the optimizer on how close they got to the target. A workflow diagram that summarizes the computational logic is provided in Figure 5.4.

5.2.4 Termination Conditions

As already explained, each of the phases is propagated separately. Aside from the predefined times of flight, several termination conditions have been added to avoid infeasible trajectories and to increase efficiency:

- The periodic orbits are propagated for one period each.
- The departure manifold is propagated for a TOF of T_{man}.
- The target manifold phase is propagated for a maximum TOF predefined by the user before the optimization.
- The Optimal control phase is propagated for a maximum TOF predefined by the user before the optimization, until the distance to the Moon exceeds a non-dimensional value of 0.45 (being 0.5 half of the distance between the Earth and the Moon), or until the spacecraft acceleration due to the thrust is more than 100 times the initial acceleration (the spacecraft mass is too close to zero).
- The propagation of any of the phases is stopped if the distance to the Moon falls below 50 km.

[2] Performance-wise, these trajectories would require a propellant mass of 0.0 kg and a TOF of 0.0 days for the optimal control phase. Nevertheless, the constraint violations would be substantial.

5.3 Implementation Within the OGT

5.3.1 Architectural Design

As already explained, the method developed in this research had to be integrated as part of the OGT. As such, the implementation was made to be compatible with the already existing transfer strategies (such as impulsive transfers), even when the optimal control phase is not used. Figure 5.5 displays a simplified diagram of the architectural design of the OGT, focusing solely on the trajectory optimization problem considered in this research. In the figure, all blocks within the Orbit Generator Tool correspond to a separate class. As the diagram suggests, the user only needs to modify the two input files to set up the desired optimization problem. Then, the main program is run, which instantiates the *Optimization Problem* class, in which the design variables, objectives, and constraints are set up. This class is linked to the Pagmo library and carries out the optimization by instantiating the *Mission Configuration* class with the specific connected phases and optimizable parameter values. For the transfers considered in this research, three phases are used: the *Periodic Orbit* phase (the halo orbits), the *Manifold* (stable or unstable) phase, and the newly implemented *Optimal Control* phase. The first two phases employ the general class for the CR3BP equations of motion, whereas the latter requires a new class for the ordinary differential equations (ODE) according to the optimal control law. This class modifies the state EOM to include the thrust law and adds the seven costate EOM. Lastly, the *Dynamic System* class includes the necessary information on the CR3BP and celestial bodies.

After the optimization is complete (or after the user chooses to stop it), a set of output files is generated with the solution set and detailed information on the optimization. As opposed to Figure 3.3, in which a single-objective optimization is usually carried out around a chosen trajectory to refine it and obtain the final trajectory file, for the trajectories including optimal control phases all optimizations in this research must be multi-objective since the constraint violations are minimized as a separate objective, as will be explained in Section 7.1. Moreover, the trajectories are meant to be refined with ASTOS to correct the instantaneous discontinuities between the phases. This is because the OGT has a certain tolerance on the boundary conditions, meaning that a reduction in propellant mass around a given solution, may well entail an increase in the final position and velocity errors (with the constraint

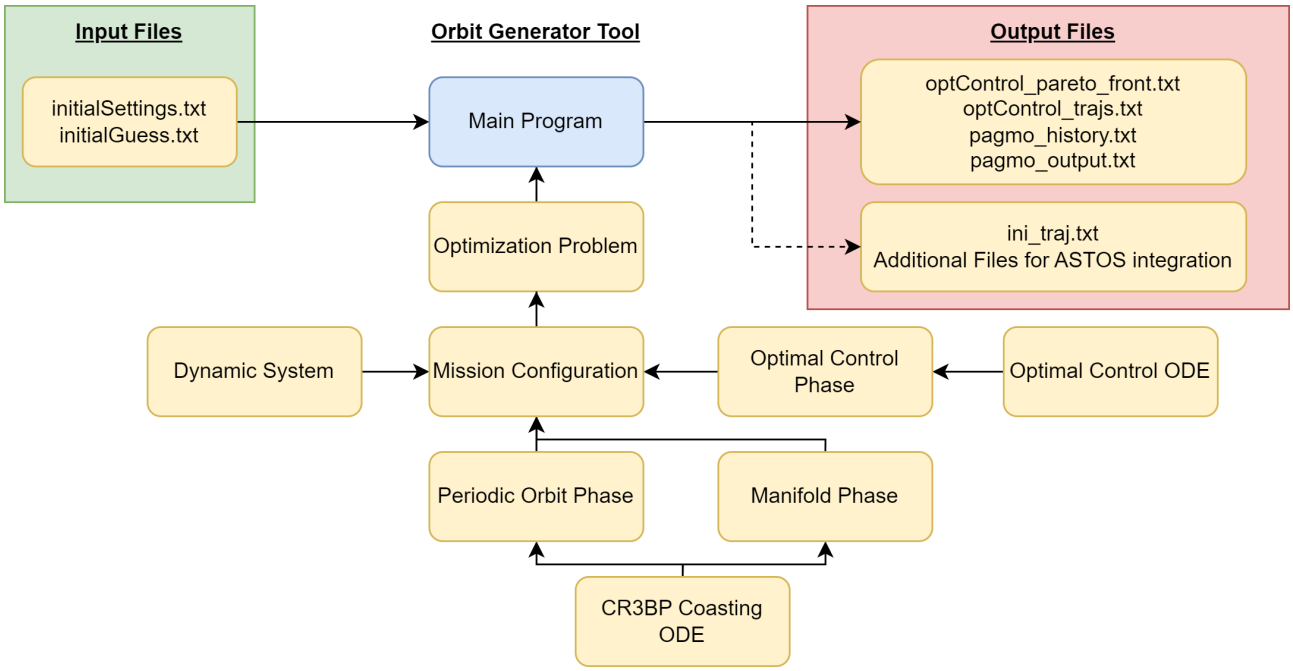

Figure 5.5: Architectural design of the OGT with a focus on the implementation of this research.

Table 5.1: Example of the parameters required within the *initialSettings.txt* file for a transfer between two halo orbits that employs an optimal control phase connecting an unstable and a stable manifold.

Parameter	**Value**
Trajectory String [-]	PMOMP
Julian Date [days]	2460000
Maximum TOF [days]	200
Maximum Propellant Mass [kg]	100
Dynamic System [-]	EM
Maximum Evaluations [-]	999999999
Maximum Time [min]	999999
Seed [-]	11
Solver [-]	Pagmo
Islands [-]	10
Population per Island [-]	120
Generations [-]	5000
Migrations [-]	1000
Topology [-]	ring
Replacement Rate [-]	1
Selection Rate [-]	1
Balance [-]	30
Algorithm [-]	moead
Algorithm-specific Parameters...	
Departure Orbit [-]	HALO-L2-NORTHERN
Defining Parameter [-]	Periapsis
Value [km]	34660.33
Arrival Orbit [-]	HALO-L1-NORTHERN
Defining Parameter [-]	Periapsis
Value [km]	27086.02
Thrust Force [N]	0.1
Initial/Final Mass [kg]	500
Exhaust Speed [m/s]	19620
Backward Propagation [-]	false
Maximum Optimal Control Phase TOF [days]	80

violations still being nominally 0). Hence, the SO optimization is not carried out and the final trajectory file (*ini_traj.txt*) is obtained by commanding the tool to only run the initial guess, which ought to be the selected optimized trajectory from the final Pareto front.

5.3.2 Input Files

All the required setup information is summarized in two input files: one with the use case and optimizer settings (*initialSettings.txt*), and one with the optimizable parameters and their bounds (*initialGuess.txt*). These files include different parameters depending on the transfer type, however, in this section only the required configuration for transfers with an optimal control phase is described.

initialSettings.txt

The *initialSettings.txt* file is divided into three main sections: the general transfer information, the optimization data, and the use case information, which consists of the orbital parameters and the spacecraft specifications. Table 5.1 displays the necessary parameters for the optimization carried out in this research, using the first use case, explained in Section 6.1.

In the table, the *trajectory string* lists the (in this case five) transfer phases in order. The letter "P" corresponds to a periodic solution phase (in this case, a halo orbit), "M" to a manifold phase, and "O" to the new optimal control phase. Then, *maximum evaluations* dictates the allowable number of trajectory computations carried out by the optimizer, whereas *maximum time* specifies the maximum optimization time. The balance parameter determines the relative weighting between the propellant mass and TOF objectives[3]. As such, a balance of, for instance, 30 means that the TOF is weighed 30 times less than the propellant mass. A negative value would make the optimizer maximize the TOF instead (with the specified relative weighting). Moreover, the strings "mass" and "TOF" would make the optimizer focus solely on the propellant mass and the TOF, respectively, having only two objectives when including the constraints objective. The rest of the settings related to the optimizer are explained along Chapter 7. In addition, there are several parameters not included in the table that depend on the chosen optimizer. For MOEA/D, these parameters can be found in Table 7.1.

Regarding the use case parameters, for transfers with optimal control phases, the only possible departure and arrival orbits are currently halo orbits within the Earth-Moon system (around L_1 or L_2 and northern or southern) or DROs. To define the specific orbit, six different parameters can be employed: the period, the distance between the Moon and the periapsis, the Jacobi constant, the out-of-plane orbit amplitude, A_z, the orbit amplitude in the x-direction relative to the pertinent Lagrange point, A_x, or the orbit amplitude in the y-direction, A_y. Then, *value* must be set according to the chosen defining parameter. The provided spacecraft mass can be the initial or the final mass depending on if the transfer is propagated forward or backward, respectively (specified by the *backward propagation* boolean). Lastly, *Maximum Optimal Control Phase TOF* specifies the maximum time of flight for which the optimal control phase can be propagated if no other termination condition is triggered.

initialGuess.txt

The *iniGuess.txt* file consists of three lines: the first one contains the design variables of the initial guess, the second the lower bounds for the design variables, and the third the upper bounds. A table with sample values for the required parameters of a "PMOMP" trajectory can be found in Table 5.2. As the optimal control phase ("O") can only connect manifold ("M") or periodic orbit ("P") phases, this example file encompasses all possible optimizable parameters within an optimization problem with an optimal control phase (the problem explored in this research).

The periodic solutions only have one optimizable parameter each, the orbit point (τ_{orb_1} and τ_{orb_2}), parametrizing the orbit between 0 and 1 with respect to its orbital period. This parameter is common to both halo orbits and DROs.

Lastly, according to Equation (5.17), only the departure manifold phase has an optimizable parameter, which is the phase's TOF in days, T_{man}. As such, in the other parameter, the lower and upper bounds coincide with the initial guess. Namely, the base-ten logarithm of the perturbation parameter, $\log(\epsilon_p)$, is forced to be -4 (as explained in Section 2.6), but this parameter could be optimized as well if desired. For the stable manifold, the TOF is in this case fixed to 38 days. As explained in Section 5.2, because it is the target manifold, this value corresponds to the maximum propagation time of the manifold and is not an optimizable parameter. In every trajectory evaluation, the TOF of the target manifold, T_{man_2}, is automatically updated once the closest state to the optimal control phase

[3]Only when Pagmo is used. MIDACO has its own balance parameter.

Table 5.2: Example of the parameters required within the *initialGuess.txt* file for a transfer between two halo orbits that employs an optimal control phase connecting an unstable and a stable manifold.

Phase	**Parameter**	**Initial Guess**	**Lower Bound**	**Upper Bound**
Departure Halo	τ_1	0.5	0.0	1.0
Unstable Manifold	T_{man}	15.0	12.0	22.0
	$\log(\epsilon_p)$	-4.0	-4.0	-4.0
Optimal Control	λ_x	0.05	-0.1	0.1
	λ_y	0.05	-0.1	0.1
	λ_z	0.05	-0.1	0.1
	λ_{vx}	0.05	-0.1	0.1
	λ_{vy}	0.05	-0.1	0.1
	λ_{vz}	0.05	-0.1	0.1
	λ_m	0.0	-1.0	1.0
Stable Manifold	Max. T_{man_2}	-38.0	-38.0	-38.0
	$\log(\epsilon_p)$	-4.0	-4.0	-4.0
Target Halo	τ_2	0.5	0.0	1.0

is found. If *backward propagation* in Table 5.1 were set to "true", the roles of the manifolds would be reversed (becoming the unstable manifold the target one), and the bounds ought to be changed accordingly. It should be noted that the direction of the manifold (interior or exterior) can be adjusted by setting negative times of flight. For example, for the departure manifold, the bounds can be set from -10 days to 10 days to include both the interior and exterior manifolds in the optimization, with a maximum TOF of 10 days. For the target manifolds, only one direction can be used, again selected based on the sign. For reference, the general convention followed by the OGT is that the manifolds that go toward the Earth (regardless of the Lagrange point from which the periodic solution arises) have positive TOFs.

Then, the optimal control phase has the seven initial (non-dimensional) costate parameters, $\boldsymbol{\lambda}_0$, as the optimizable parameters. In Table 5.2, the subscript "0" is dropped for visualization purposes. There are no physical bounds for these design variables, making the selection of a suitable set of optimization bounds a demanding task, which will be addressed in Chapter 6.

5.3.3 Output Files

The main output files, specified in Figure 5.5, are the following:

- *optControl_pareto_front.txt*: It consists of a list of all Pareto-optimal solutions in terms of propellant mass and TOF that are feasible (they satisfy the three constraints). Each line corresponds to a trajectory and displays the dimensional objective values, the five values for the constraint functions (propellant mass, TOF, position error, velocity error, and Jacobi constant error), and the design variables of the given trajectory. This file is updated every time a new feasible trajectory is found, so it is safe to stop the optimization at any time.

- *optControl_trajs.txt*: It includes a list of all feasible trajectories found throughout the optimization. The trajectories are presented in the same format as in *optControl_pareto_front.txt*, and this file is also updated every time a feasible trajectory that is sufficiently different from the rest in terms of performance is found. Sufficiently different in this case corresponds to a 0.5% variation in any of the objectives.

- *pagmo_history.txt*: It consists of a list of all trajectory evaluations carried out by the optimizer, regardless of their performance or constraint violations. The trajectories are displayed in the same format as the previous two files.
- *pagmo_log.txt*: This file displays general information on the optimization and specifies the optimizer and the optimization topology. Moreover, according to the *LogPeriod* parameter, it displays in subsequent lines the elapsed time and the utopia point of each island.
- *ini_traj.txt*: Includes the output data of a single trajectory generated with the OGT. For each step in time, it includes in a line the phase, the date, the ECI and BCR states, the thrusting attitude vector (in the RSW frame, as well as with the in-plane and out-of-plane angles), and the normalized thrust magnitude. For transfers with an optimal control phase, this file also includes the costate parameters and the Jacobi constant. This file is helpful to plot trajectories and to use the trajectory as initial guess for ASTOS optimization.
- Additional files: They include data and variables to automatically create the ASTOS scenario for a given trajectory and to accurately replicate it.

5.3.4 Unit and System Testing

To verify the implementation, a series of unit and system tests were carried out. These tests ought to be specific and account for all allowable trajectory options, as this multipurpose tool must not be only limited to the use cases explored in this research. As such, tests were performed for trajectories propagated forward and backward, including and excluding manifolds. In addition, the closest state search strategy was thoroughly tested to account for all situations. Since the OGT is meant to eventually be incorporated within the ASTOS software, user-friendliness checks were also implemented[4]. Although the outcomes of several tests were checked via visual inspection of the data, most of them were evaluated numerically. A summary of the main tests can be found in Table 5.3. The table includes the titles of the tests, an identifier, and a general description of their purpose. The purpose of this table is simply to provide an overview of the verification process. Specific information on the tests would require a much deeper understanding of the program's internal structure and is more suited to the developer's documentation of the tool. The tests themselves are divided into five general categories of increasing scope: testing of the equations of motion (*EOM*), testing of the phases (*PHASE*), evaluation of the complete trajectories (*TRAJ*), assessment of the optimization procedure (*OPT*), and user-friendliness tests (*USER*).

After the implementation, a readily available testing tool for the complete OGT was run to ensure that the new code did not negatively interact with the existing software. The outcome was successful, and new tests were added to this testing tool to now account for the new implementation (transfers with an optimal control phase).

[4]Most of the necessary user-friendliness checks have already been implemented by previous developers.

Table 5.3: Overview of the most relevant unit and system tests carried out on the implementation of this research.

Test ID	Title	Description
EOM-01	*State Derivative Computation Test*	Ensures that the 14 equations of motion are correct by comparing their results for a given state vector with those from an independent computation.
EOM-02	*State Derivative Limits Test*	Ensures that *NaN* values are never returned, even when all states are zero.
EOM-03	*Thrust Computation Test*	Ensures that the provided thrust acceleration vector is correct by comparing their results for a given state vector with those from an independent computation.
EOM-04	*Thrust Limits Test*	Ensures that *NaN* values are never returned, which can occur when all components of $\boldsymbol{\lambda}_v$ are zero.
EOM-05	*Jacobi Constant Test*	Ensures that the Jacobi constant is computed correctly by comparing their results for a given state vector with those from an independent computation.
PHASE-01	*Parameter Setting Test*	Ensures that the optimizable and other necessary parameters are set correctly when instantiating an optimal control phase by commanding a specific set of parameters, also with backward propagation.
PHASE-02	*Collision Test*	Ensures that the optimal control phase propagation is stopped when the spacecraft gets too close to a celestial body by running a sample case, also with backward propagation.
PHASE-03	*Minimum Mass Test*	Ensures that the optimal control phase propagation is stopped when the spacecraft mass is too close to zero by running a sample case.
PHASE-04	*Maximum Distance Test*	Ensures that the optimal control phase propagation is stopped when the spacecraft is too far from the Moon by running a sample case.
PHASE-05	*Manifold Direction Tests*	Testing tool that checks the manifold generation implementation to ensure that the direction of the manifold phases (interior/exterior) is the one desired for a range of orbit points, orbits, and perturbation parameters.
TRAJ-01	*Closest State Tests*	Collection of tests that ensure that the closest state between the optimal control phase and the target phases is found correctly for all trajectory cases ("PMOMP", "PMOP", "POMP", and "POP") by propagating sample trajectories. Backward propagated trajectories are also included, as well as trajectories that ignore the target manifold and directly connect to the target orbit, which must remove the manifold phase altogether. The time and state continuity and the phase durations were also checked.

TRAJ-02	*Backwards Propagation Test*	Ensures that the backward propagation feature is implemented correctly by replicating a forward-propagated trajectory with backward propagation. Continuity in states and time was also checked.
TRAJ-03	*Target Phase Discretization Test*	Ensures that the target phases (halo orbit and manifold) have a maximum step size reduced by a factor of two (also with backward propagation) by checking a sample trajectory.
OPT-01	*Pagmo DLL Tests*	Collection of tests that ensure that the new Pagmo Dynamic-Link Library (DLL) that includes NSGA2 and NSPSO works correctly and provides the same output as the original when using the rest of the optimizers. This was done by directly comparing the complete output of a sample optimization with that obtained with the older DLL.
OPT-02	*Pareto Front Test*	Ensures that the Pareto front is computed correctly from the array of feasible trajectories by providing a sample array of trajectories and comparing the result to the expected outcome.
OPT-03	*Trajectory Storing Test*	Ensures that feasible trajectories are only stored when the performance is sufficiently different compared to the previously found feasible solutions.
USER-01	*Allowable Transfer Check*	Ensures that the supplied transfer phases are logical and fall within the scope of the implementation. Thus, the program throws an error when incorrect trajectory strings are provided.
USER-02	*Manifold Parameters Check*	Ensures that the upper and lower bounds for the fixed manifold phase parameters are equal to the initial guess. If this is not the case, the program automatically fixes the bounds and notifies the user. For the target manifold (including backward propagation), the manifold TOF bounds are also fixed to the initial guess as it corresponds to the maximum TOF.
USER-03	*Target Orbit Parameter Check*	Ensures that the target orbit point's bounds coincide with the initial guess when no target manifold is used, as it is not needed (including backward propagation). If this is not the case, the program automatically fixes the bounds and notifies the user.
USER-04	*Optimizer Parameters Check*	Ensures that the optimizer parameters are within the appropriate bounds for NSGA2 and NSPSO (the two new interfaced optimizers) and notifies the user if this is not the case.

Analysis of the Problem 6

Having finalized all code implementations to generate and optimize trajectories, the next step before optimizing is to analyze the problem to assess its difficulty and understand its behavior to hopefully draw relevant conclusions that will aid the optimization process. Firstly, two use cases explored by other authors are chosen in Section 6.1. Secondly, the integrator and its settings to maximize performance are selected in Section 6.2. Thirdly, Section 6.3 carries out a design space exploration of the problem, exploiting the design of experiments theory. Lastly, the final optimization bounds and the selected constraint-handling strategy can be found in Section 6.4.

6.1 Proposed Use Cases

To explore the design space, two fairly complicated transfers between halo orbits are selected from published work. These use cases are sufficiently different from each other to make the drawn conclusions from the analysis applicable to a wide range of scenarios. Nevertheless, only the first use case will be optimized in Chapter 7 due to the time constraints of this research. However, the developed tool can in principle be employed to optimize any use case within the CR3BP, also for orbital families other than the halo family.

6.1.1 First Use Case

The first transfer under consideration aims to imitate the one proposed by Pritchett et al. (2017). As they utilize a direct collocation method based on Gauss-Legendre discretization to obtain their optimized trajectory, it is an interesting use case as it enables a comparison between the proposed indirect approach and the commonly used collocation methods with rough initial guesses.

Table 6.1 displays the orbital parameters of both the departure and target orbits. It should be noted that the parameters moderately differ from the ones provided by Pritchett et al. (2017). Hence, although the transfers are still similar and comparable, small variations in performance lose their significance[1]. In addition, the use case in this research is arguably more demanding due to the larger difference in Jacobi constants. As can be seen from the table, the transfer takes place between halo orbits of similar orbital energies but belonging to families around different Lagrange points. In addition, because both of them are northern halos, the spacecraft will not only need to move from one side of the Moon to the other but also to change the direction of rotation around the Moon in the BCR frame. The spacecraft thrust, its initial mass, and the specific impulse of the thrusters are shown in Table 6.2, leading to an initial acceleration of $2{\cdot}10^{-4}$ m/s^2 and a mass flow of $5.10{\cdot}10^{-6}$ kg/s or 0.44 kg/day. Even though the selected mission Julian Date (JD) will not affect the CR3BP trajectories (as it is time-invariant), it plays a role when translating it to the inertial frame. An arbitrary date of 2460000 days (February 24$^{\text{th}}$, 2023, at noon) was selected for both use cases.

6.1.2 Second Use Case

The second use case can be found in the work by Kayama et al. (2021), specifically, it corresponds to "case 3" of the article. In this case, they use SCP to obtain the optimized transfer instead of collocation methods. From the orbital parameters of the orbits presented in Table 6.3, it can be seen that both

[1]The discrepancy in orbital parameters is attributed to the fact that the perilune radii were used to generate the halo orbits because they uniquely parameterize the halo families (Tatay Sanguesa, 2021). However, the radii provided by Pritchett et al. (2017) may have been accidentally swapped between the departure and target orbits in their article. This seems to be the cause, as using the opposite perilune radii yields halo orbits with Jacobi constants that match those reported by Pritchett et al. (2017). Unfortunately, this difference was discovered too late in the research project.

Table 6.1: Parameters of the departure and target orbits in the first use case.

Parameter	**Departure**	**Target**
Orbit Type	Northern EML_2 Halo	Northern EML_1 Halo
Period [days]	13.43	10.85
C [-]	3.05327	3.00387
Perilune [km]	34660.33	27086.02

Table 6.2: Spacecraft specifications of the first use case.

Parameter	**Value**
Thrust [N]	0.1
Initial Mass [kg]	500
Specific Impulse [s]	2000

are southern halos around the second Lagrange point, meaning that the spacecraft need not change rotation directions and that it will stay on the far side of the Moon. Nonetheless, the difference in the Jacobi Constant is greater than for the first use case, which may make the problem more difficult.

Table 6.4 shows the spacecraft specifications, now having a larger specific impulse but half the acceleration than in the first use case. As Kayama et al. (2021) restrict the maximum allowable acceleration throughout the transfer to 10^{-4} m/s^2, the final mass is supplied to the OGT tool instead of the initial mass to ensure that this value is never exceeded. As such, the equations of motion will be propagated backward in time, and the optimal control phase will aim to connect to the unstable manifold of the departure orbit starting from the stable manifold of the target orbit instead of the opposite. With these values for the parameters, the mass flow will be $3.40 \cdot 10^{-6}$ kg/s or 0.29 kg/day. Visualization of the orbits of both use cases is provided along Sections 6.2 and 6.3.

6.2 Integrator Analysis

One of the main factors that determine the speed of each simulation is the EOM integrator speed, especially when the trajectories can last longer than a hundred days. Other parameters are, for instance, the termination conditions, the calculation of the thrusting attitude and magnitude, and, in this

Table 6.3: Parameters of the departure and target orbits in the second use case.

Parameter	**Departure**	**Target**
Orbit Type	Southern EML_2 Halo	Southern EML_2 Halo
Period [days]	14.75	12.83
C [-]	3.14011	3.03644
Perilune [km]	49092.79	30392.27

Table 6.4: Spacecraft specifications of the second use case.

Parameter	**Value**
Thrust [N]	0.1
Final Mass [kg]	1000
Specific Impulse [s]	3000

Table 6.5: Design variable values and performance of the obtained nominal trajectory.

Parameter	Value	Unit
τ_{orb_1}	0.367882	-
T_{man}	14.431031	days
λ_x	0.103116	-
λ_y	-0.032238	-
λ_z	0.209432	-
λ_{vx}	-0.460699	-
λ_{vy}	-0.612191	-
λ_{vz}	0.964264	-
λ_m	-2.695675	-
τ_{orb_2}	0.379583	-
m_p	6.93	kg
TOF	60.13	days
TOF (Opt. Control Phase)	15.71	days
T_{man_2}	30.00	days
Position Error	308.40	km
Velocity Error	0.93	m/s
C Error	$7.30 \cdot 10^{-4}$	-

case, the computation of the closest state between the thrusting and target trajectories. Reducing the computational cost of each simulation is of great interest as the optimization runs require more than 100,000 trajectory simulations, especially with such tight constraints. The focus of this section is then to choose the fastest integrator and integrator settings for the given dynamic system while ensuring that the trajectories have an appropriate level of accuracy to be useful initial guesses for further optimization.

6.2.1 Nominal Trajectory

The first step is to find a nominal trajectory that can be used to carry out the integrator analysis. This trajectory ought to be "challenging" enough such that the error estimation can be generalized to the whole set of trajectories, i.e., to ensure that the optimized transfers also satisfy the accuracy requirement. In addition, it should satisfy the constraints imposed in Section 5.3 (it should be a feasible solution). This is not a strict requirement for the integrator analysis, however, this nominal trajectory will be used in the design space exploration (Section 6.3), for the trajectory refinement with ASTOS (Section 8.2), and in the sensitivity analysis (Section 8.3). Thus, to draw generalized conclusions for the whole solution set, the nominal trajectory must be feasible.

A feasible solution was obtained by running a preliminary optimization with MIDACO using the first use case. All phases were integrated with the *DormandPrince853* integrator, using relative and absolute tolerances of 10^{-9}, a maximum non-dimensional step size of 10^{-3} ($\sim$375 s), and a minimum of 10^{-15} (numerical accuracy of the Java *double* data type). These are the settings employed by Tatay Sanguesa (2021), Walther (2022), and Maisch (2022) for the other OGT functionalities. Its design variable values and performance parameters are shown in Table 6.5[2]. In the table, sufficient significant figures were included for each optimizable parameter to ensure accurate reproducibility. It should be noted that the subscript "0" has been removed from the initial costate parameters for conciseness. The

[2]These performance values already correspond to the nominal trajectory re-propagated with the final integrator and integrator settings from Subsection 6.2.4. The differences were found to be very small.

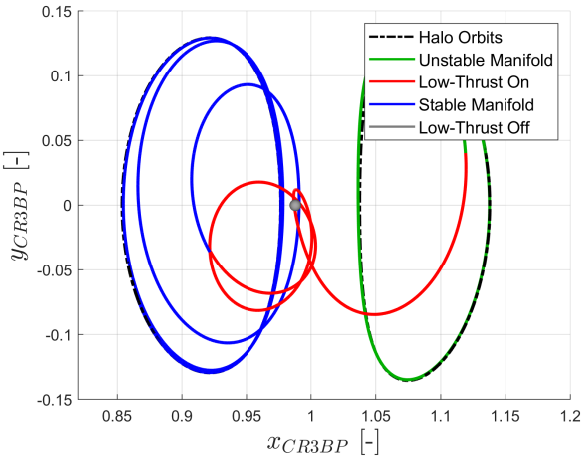

Figure 6.1: Top view of the nominal trajectory used for the integrator analysis.

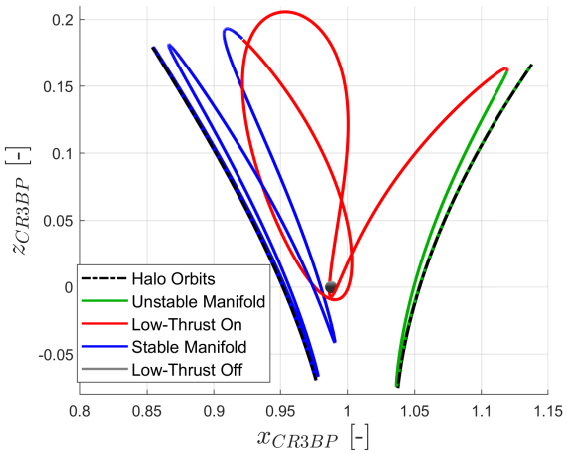

Figure 6.2: Side view of the nominal trajectory used for the integrator analysis.

complete transfer takes 60.13 days, divided into the times of flight of the (unstable) departure manifold (T_{man}), the optimal control thrusting phase, and the (stable) target manifold (T_{man_2}). Compared to the one obtained by Pritchett et al. (2017), with a TOF of 47.5 days and a required propellant mass of 8.02 kg, the nominal trajectory has a longer TOF but a lower propellant consumption, thus being competitive. For reference, the required propellant mass, m_p, of the nominal trajectory corresponds to $\sim$1.4% of the total spacecraft mass.

If made non-dimensional with the values from Equation (2.5), the position and velocity discontinuities between the optimal control and the target manifold phases lie within the non-dimensional constraints of 10^{-3} (constraint values before the constraint relaxation (Subsection 5.2.2)). Moreover, the Jacobi Constant constraint of 10^{-2} is comfortably satisfied. Hence, the trajectory is deemed feasible.

The top and side views of the trajectory in the BCR frame can be found in Figure 6.1 and Figure 6.2, respectively. These figures include the departure (parking) and target orbits as well. The departure halo corresponds to the one on the right side of the Moon. Thus, five phases are present: the initial EML_2 halo, the departure manifold (green), the optimal control phase (red), the target manifold (blue), and the target EML_1 halo. It can be seen in the figures that this transfer includes two flybys considerably close to the Moon during the optimal control (thrusting) phase. As the accelerations and acceleration rates are generally larger the closer the spacecraft is to the Moon (the dynamics of the system are faster, being the most critical regions of the transfer), a nominal trajectory with such characteristics is considered to be "challenging" enough for useful analysis. In addition, 60 days of travel time is a long enough period to properly evaluate the error propagation. It should be noted that the 308.35 km discontinuity between the thrusting phase and the stable manifold is present in the figure, even though it is difficult to see, suggesting that the trajectories that satisfy the constraints require a considerably small refinement. The 10^{-4} discontinuities to transition between the manifold and halo phases are also present but cannot be discerned in the figure.

During the optimal control phase, one can notice that the spacecraft is continuously thrusting, presenting no coasting periods during this phase. From the theory presented in Section 4.1, this entails that λ_m can take any value without modifying the trajectory at all, as long as the switching function, S, in Equation (5.11) does not become positive. By examining the equation, one can conclude that any value higher than the one in Table 6.5 would never modify the trajectory. Moreover, as long as

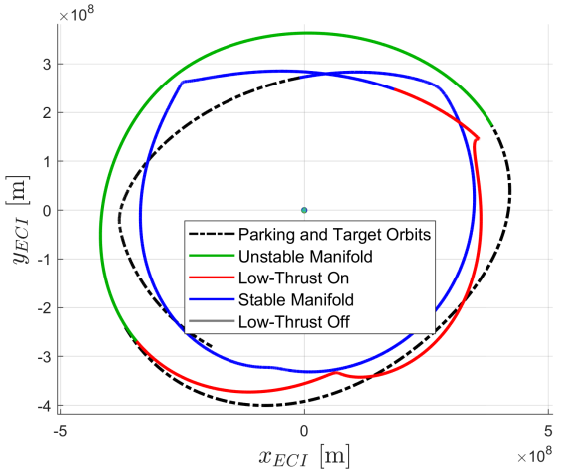

Figure 6.3: Top view of the nominal trajectory used for the integrator analysis in the inertial frame (ECI).

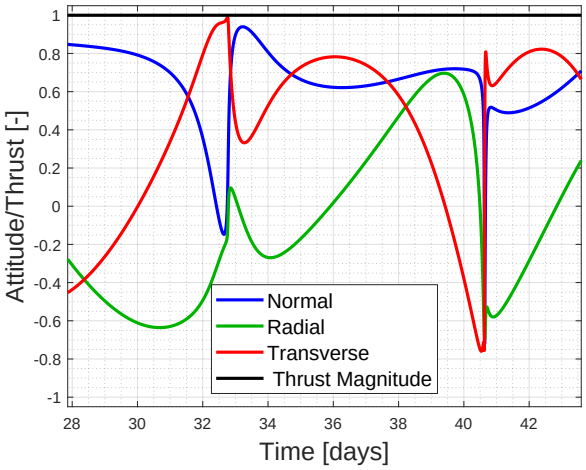

Figure 6.4: Attitude unit vector components and thrust profile for the nominal trajectory as a function of time.

S does not switch signs along the trajectory, changing λ_m by a certain amount will simply shift the switching function upwards or downwards by exactly the same amount at all points in time (since $\dot{\lambda}_m$ is independent of λ_m). Further analysis of S as a function of time for this transfer led to the conclusion that the trajectory will not be affected provided that $\lambda_m \geq -5.652505$. Any value below this one will trigger a coasting phase in the second flyby.

The top view of the trajectory is also displayed in the ECI frame in Figure 6.3, with dimensional quantities. As expected, the halos are not periodic in this frame and appear like arbitrary motions. All phases are connected in this figure (ignoring the 308 km discontinuity), verifying that the trajectory is indeed continuous in time and that the transfer between the orbits is carried out successfully while rotating with the Moon around the Earth (assuming an eccentricity of 0). The flybys, and the portions of the transfer that are closer to the Moon in general, can be identified in the figure as the sharp "peaks" that deviate from a quasi-circular motion since the spacecraft changes position considerably faster in these regions. Lastly, the length of the phases can be estimated in this frame, with the stable manifold (blue) being clearly the longest phase, since it performs more than one revolution around the Earth.

The normalized thrust magnitude, τ, and the thrust attitude unit vector components, $\hat{\boldsymbol{u}}$, in the RSW frame (with respect to the Earth) are plotted in Figure 6.4 as a function of time. These are only shown for the optimal control phase since the spacecraft coasts during the other phases. It should be pointed out that in the displayed trajectory, the departure and target halos were propagated for one orbital period each, and this propagation time was included in the time axes of the figures. This explains why the optimal control phase starts on day 27.86, instead of on day 14.43, as would be deduced from Table 6.5. Nonetheless, at the chosen JD, the spacecraft is at the starting point of the transfer itself, i.e., the starting point of the unstable manifold phase.

Since the spacecraft is thrusting during the full optimal control phase, τ has a constant value of one, as expected. A more interesting behavior can be found in the components of the attitude vector, which exhibit a slowly changing behavior all along the trajectory, except on days $\sim$33 and $\sim$40.5. These points in time correspond to the aforementioned lunar flybys during this phase. The rapid change in the unit vector direction can be attributed to the faster dynamics close to the Moon. Nevertheless, the attitude change is still gradual when considering the large timescale, having the

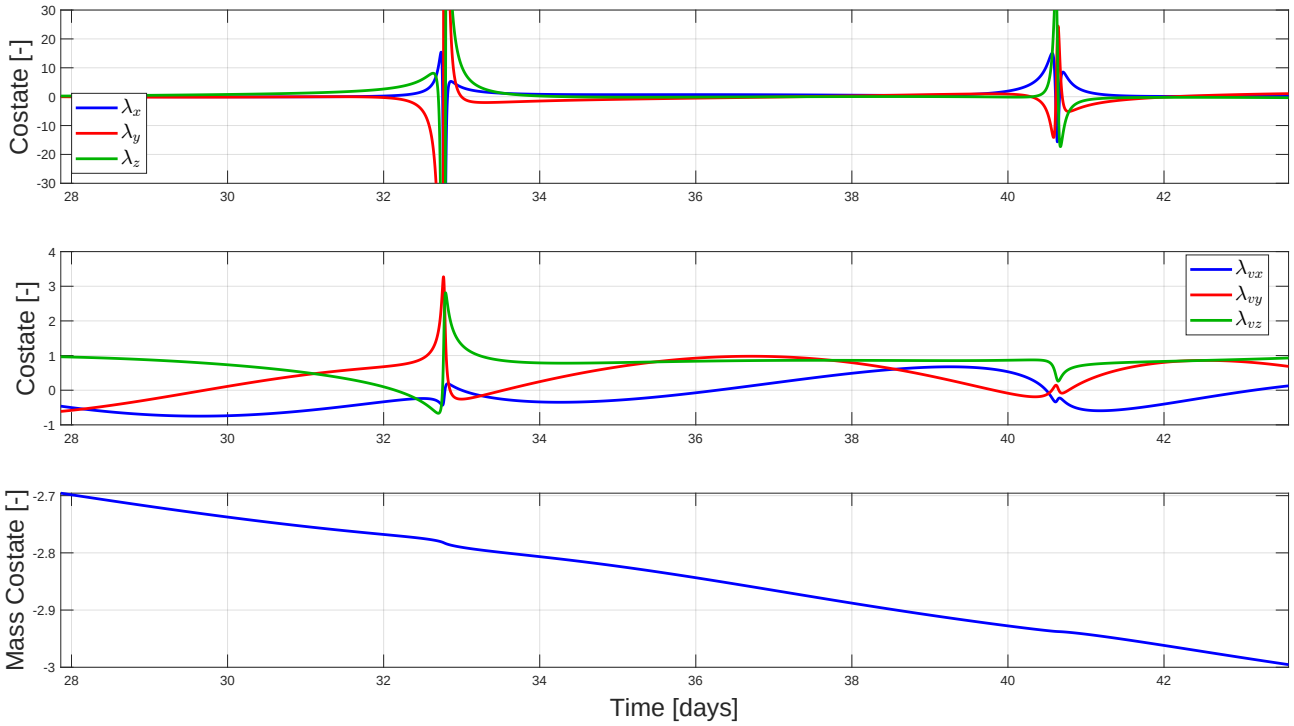

Figure 6.5: Costate parameter values for the nominal trajectory as a function of time.

spacecraft more than an hour to turn not even 140°. This is important from a mission design point of view because flybys tend to be the most critical part of transfers as they are the places where the orbital energy is changed the most in the shortest amount of time. Overall, the attitude displays a smooth behavior, which is one of the usual characteristics of optimal transfers (Pritchett et al., 2017).

Since the attitude components are related to the costate parameters according to Equation (5.9), one would expect to observe a similar sudden change in the costate parameter values during the flybys, specifically in the costates related to the velocity components (λ_{vx}, λ_{vy}, and λ_{vz}). The seven costate parameters as a function of time are displayed in Figure 6.5, grouping the position and velocity costates into two plots. As predicted, there are "peaks" present in the behavior at the points in time at which the flybys take place. The effect is more severe in the position costates than in the velocity ones, and this rapid variation affects the velocity costates following Equation (5.8). Interestingly, the magnitudes of the costates tend to return to the original order after leaving the flyby points. Finally, the mass costate decreases strictly monotonously since there are no coasting phases.

6.2.2 Required Accuracy Level

For the trajectory to be a suitable initial guess for optimization with ASTOS, the OGT transfers must ensure a low enough error in the integration of the EOM, such that they are realistic enough to be transitioned to a real-life scenario. Due to the simplicity of the mass equation of motion (the derivative is either 0 or a negative constant), the leading errors will be present in the position and velocity components. Having, from Subsection 5.2.2, non-dimensional constraint values for both position and velocities of 10^{-3} (before relaxation), the propagation accuracy should be at least an order of magnitude lower to properly quantify the error. However, the imposed perturbing factor to enter the manifold from the periodic solutions, ϵ_p, (Section 2.6) is chosen to be 10^{-4}. Therefore, to correctly describe this discontinuity, the position and velocity errors ought to be below 10^{-5} (in non-dimensional quantities). In dimensional values, this would correspond to a maximum position and velocity error norms of 3.84 km and 1.02 cm/s, respectively.

Having lower errors would in principle not be of use considering that, firstly, the trajectory would still need to be transitioned to a high-fidelity model using ASTOS, being this error correction negligible in comparison with the transitioning error, and, secondly, because the state-of-the-art Orbit Determination (OD) capabilities in the cislunar environment are on the order of several kilometers for the position, and several centimeters per second for the velocity (Newman et al., 2022) (Scorsoglio et al., 2023). As such, it will probably not be possible for the guidance, navigation, and control (GNC) subsystem to correct for such small deviations anyway. Henceforth, 10^{-5} is the value chosen for the allowable position and velocity errors of the trajectory. Now, the goal is to find the integrator that guarantees this level of accuracy for the lowest computational cost.

6.2.3 Benchmark Generation

To estimate the error of the trajectories, a benchmark needs to be first generated. This benchmark will be taken as the ground truth, the zero-error trajectory. For the estimation to be accurate, the maximum error of the benchmark itself must be at least one order of magnitude lower than the error that is to be estimated. However, in an integrator analysis, one would like to examine the behavior of the integrators for a wider range of accuracy levels, going below the allowable value of 10^{-5}. Therefore, the selected benchmark maximum error is two orders of magnitude below the allowable one, namely 10^{-7} in non-dimensional quantities.

For the benchmark, it is interesting to propagate the trajectory with a constant step size to have the points more evenly distributed, making the calculation of the difference in position and velocity more accurate when interpolating. A common approach to estimating the benchmark error is to compute the difference with respect to the same trajectory propagated with half the step size. Thus, to obtain the position error, for instance, as a function of time, the three position components of the benchmark are interpolated to estimate the value at each of the time steps of the trajectory for which the error is to be approximated. Then, at each point in time, the error is taken as the norm of the vector difference in position. The same procedure applies to the velocity computation. It should be noted that the benchmark is interpolated instead of the other trajectory as it takes more steps, making the interpolation more accurate. For the interpolation, Hermite cubic splines are employed (Fritsch and Carlson, 1980), as not only do they ensure continuity and differentiability, but also piecewise monotonicity, greatly reducing the typical overshoots of the conventional cubic splines. Moreover, piecewise interpolation is preferred to avoid Runge's phenomenon for higher degrees. Thus, this interpolation method is considered to be the most accurate over the entire trajectory.

In the OGT, the costate EOM, Equation (5.8), are integrated together with the state EOM, Equation (2.28). As these costate equations describe the necessary conditions for local dynamic optimality, it is desirable to solve them as accurately as possible. Hence, for this analysis, the decision was made to not fix the thrust strategy but to compute the thrust according to the costate EOM. Consequently, the estimated error will be a combination of the actual trajectory error and the change in the thrust strategy due to the error in the computation of the optimality equations. With this approach, the trajectory with the ideal (zero-error) thrust profile will not differ from the nominal trajectory by more than 10^{-5} in position and velocity at any point.

A preliminary inspection of the nominal trajectory propagation led to the conclusion that the leading errors take place during the optimal control phase. The observed difference is in the order of several orders of magnitude. Moreover, this phase is the one with the largest computational cost, even with the search strategy for the closest state from Subsection 5.2.3 not active. These observations make sense since for this phase fourteen EOM are propagated instead of the seven state EOM. Moreover, the presence of flybys drastically increases the error in the trajectory. Therefore, the focus of this integrator analysis will be on the optimal control phase, which is not connected to the target manifold anyway.

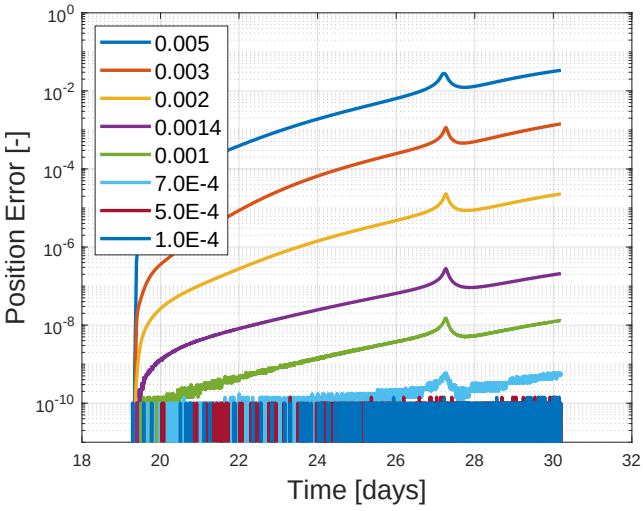

Figure 6.6: Position error profiles as a function of time for different constant integrator step sizes.

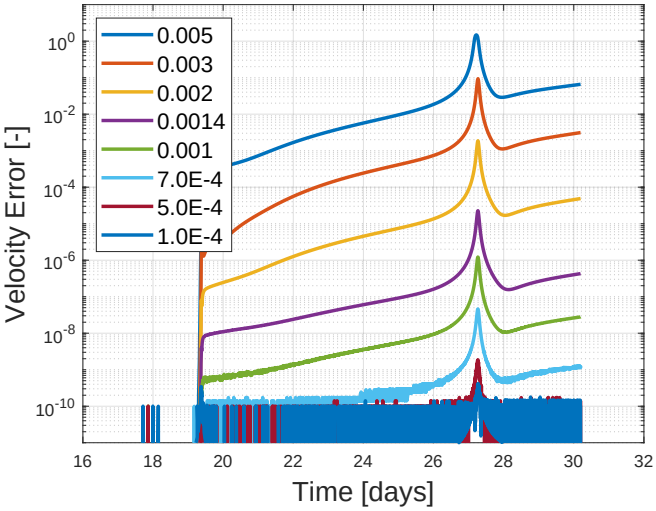

Figure 6.7: Velocity error profiles as a function of time for different constant integrator step sizes.

For the periodic orbit and manifold phases, the error with the nominal settings (DormandPrince853 with tolerances of 10^{-9}, a non-dimensional maximum step size of 10^{-3} and a minimum of 10^{-15} (Tatay Sanguesa, 2021)) is in the rounding error regime at all times, thus being negligible and not affecting the optimal control phase. As these phases are used in the OGT for other types of transfers (such as impulsive), for the sake of consistency within the phases, the integration settings for the orbits and manifolds are kept the same (except for the reduction in maximum step size for the target phases explained in Subsection 5.2.3).

Having explained the approach, Figures 6.6 and 6.7 display the position and velocity error profiles as a function of time for different (constant) step sizes, in non-dimensional units. To generate these, DormandPrince853 was used again. In this case, the time axis does not include the halo phases, so the optimal control phase occurs at a different time. Both position and velocity errors behave similarly over time: the error is practically zero before the first flyby (day $\sim$19), which then steadily increases over time until experiencing a sharp peak at the second flyby (day $\sim$27). Then, the error appears to decrease down to the value before the flyby. By examining the amount of noise in the error profiles as a function of the step size, it can be concluded that the computer rounding error does not play a role above errors of 10^{-9} for both position and velocity. Below 10^{-10}, the error is completely dominated by the rounding error. In between, the obtained profiles seem to be a combination of the rounding error and truncation error (error arising from the integrator approximations when solving the EOM).

The error in velocity for a given step size is consistently larger than in position, presenting the biggest differences along the flyby regions. As such, the velocity errors will determine the chosen benchmark. Since the maximum trajectory error ought to be below 10^{-7}, any step size below $7{\cdot}10^{-4}$ would satisfy this requirement (looking at Figure 6.7). However, it is also desirable that the trajectory error is dominated by the truncation error to avoid random "jumps" in the error and such that the results of the analysis are independent of the hardware. Hence, the trajectory with a constant step of $7{\cdot}10^{-4}$ is selected as the benchmark.

Figure 6.8 displays the maximum position and velocity errors as a function of the step size. Two major quasi-linear slopes can be identified in this figure: an almost horizontal slope below a step size of $5{\cdot}10^{-4}$ (which would probably be noisy if more step sizes were added), and a considerably steep positive slope above said step size. This is the behavior to be expected when plotting the data in a *log-log* plot, with a clear separation between the regime dominated by the truncation error (above $5{\cdot}10^{-4}$)

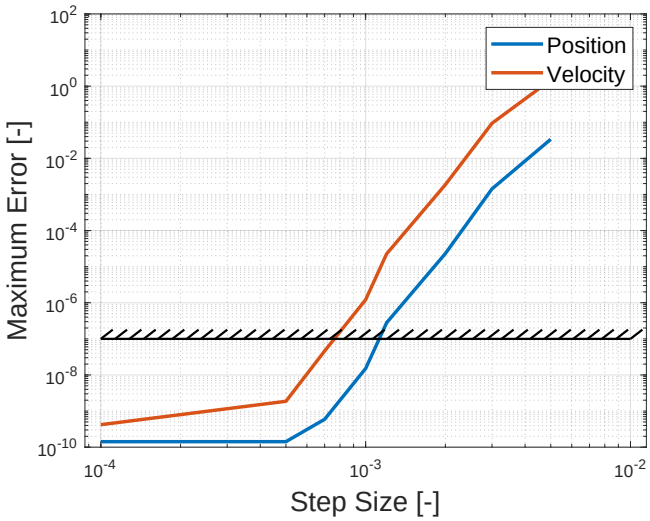

Figure 6.8: Maximum position and velocity errors as a function of the integrator step size.

and the one dominated by the rounding error. This figure acts as further verification for the obtained error profiles and helps to confirm the fact that the benchmark is indeed in the regime dominated by the truncation error.

6.2.4 Integrator Trade-off

Different integrators and integrator settings can be compared now that an appropriate benchmark has been generated. The error profiles are computed with respect to the benchmark, and the computational effort of each run is quantified through the number of times the integrator calls the differential equations (i.e., the number of function evaluations). This number includes the sub-steps and any correction steps that the integrator computes. This is deemed a more reliable figure of merit than the CPU time because the latter depends on factors external to the integration method, such as the computer background processes. The goal is then to find the integrator and integrator settings that provide the required accuracy level with the lowest number of function evaluations.

The analysis is carried out for the maximum and velocity errors, even though the velocity error will probably be the determining factor error source. Only variable step size integrators are considered in this analysis as they are generally more efficient when dealing with dynamical systems of highly varying dynamics (Tatay Sanguesa, 2021). For instance, as can be deduced from Figure 6.7, the step size required close to the Moon ought to be much smaller than far from it to obtain the same level of accuracy. The following integrators from the *Apache Commons* toolbox[3] will be put to assessment:

- DormandPrince54 (DOPRI54) and DormandPrince853 (DOPRI853)
- HighamHall54 (HIHA54)
- Adams-Moulton, taking four (AMO4), seven (AMO7), and twelve (AMO12) interpolating points.
- Gragg-Bulirsch-Stoer (GRABUSTO)

In this list, DOPRI and HIHA are multi-stage integrators of varying order, AMO is an implicit multi-step integrator, and GRABUSTO employs Richardson extrapolation to solve the ordinary differential equations. For all these, the relative and absolute tolerances were varied to span several

[3]URL:https://commons.apache.org/ [Accessed: 21/06/2024]

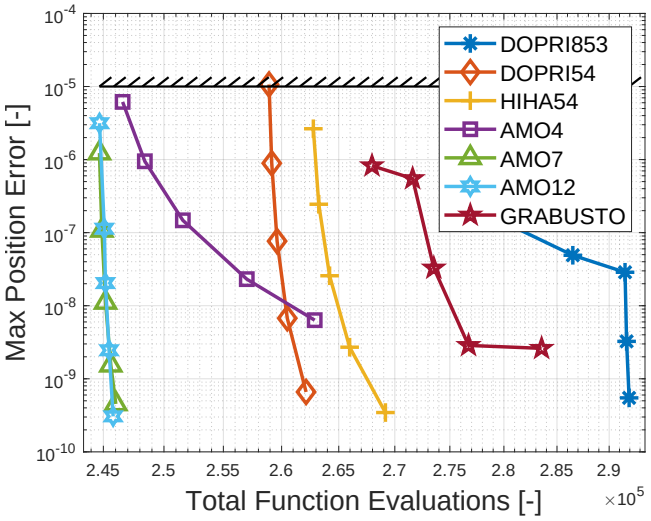

Figure 6.9: Trade-off between the maximum position error and the total number of function evaluations as a function of time for different integrators and integrator settings.

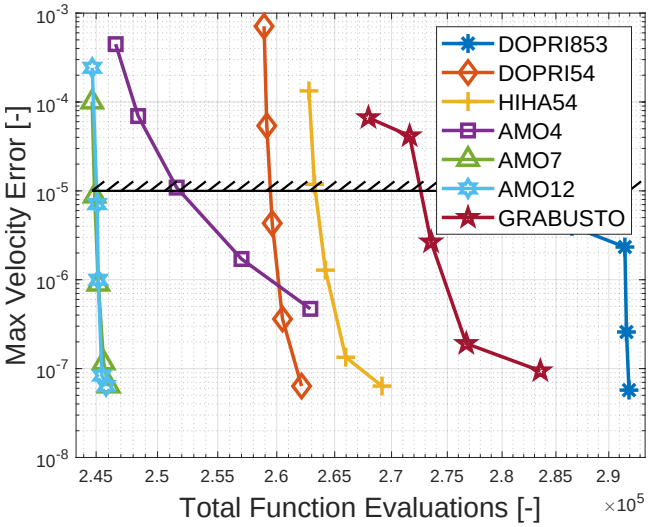

Figure 6.10: Trade-off between the maximum velocity error and the total number of function evaluations as a function of time for different integrators and integrator settings.

orders of magnitude in maximum error values. For every single run, it was decided to make the absolute tolerance equal in value to the relative one. The maximum time step was kept as 10^{-3} ($\sim$375 seconds) to ensure that the number of steps is large enough to accurately find the closest state to the target trajectory (Subsection 5.2.3).

The outcome of the analysis is displayed in Figure 6.9 for the position error, and in Figure 6.10 for the velocity error. In these figures, the maximum errors of each trajectory are plotted against the required number of function evaluations for the different integrators and integrator settings. The boundary line at an error of 10^{-5} corresponds to the required accuracy. As predicted, the maximum velocity errors dominate over the position errors in all cases. Thus, the velocity error will be the deciding factor. Nonetheless, the error behavior of the integrators compared to each other is considerably similar in both figures, being Figure 6.9 roughly equivalent to Figure 6.10 if it were scaled down by two orders of magnitude in the error axis. It should be noted that for the top point of GRABUSTO and the top two points of DOPRI853, the maximum step size was slightly increased to obtain trajectories of lower accuracy (for those integrator settings, the step size taken was always the maximum allowable value). This fact explains the sudden change in slope. Lastly, except for AMO4, all integrators comfortably reached the accuracy level of the benchmark (10^{-7} in Figure 6.10).

By examining the figures, it is safe to conclude that the best-performing integrators are AMO7 and AMO12. Not only are they the ones that require the fewest function evaluations for all accuracy levels, but they also exhibit a robust trend. It is also interesting to note that the default integrator of the OGT, DOPRI853, although extremely accurate, it is the worst performing of all for the given nominal trajectory and EOM. This fact showcases the benefits of the integrator analysis: aside from estimating the error of the trajectories, two integrators 16% faster in terms of function evaluations have been found.

To choose between AMO7 and AMO12, the error profiles as a function of time can be inspected. Figure 6.11 and Figure 6.12 show, respectively, the position and velocity error behavior as a function of time for the different integrators. For each integrator, the run with a maximum error closest to the required accuracy level was selected. One can conclude from these figures that all integrators

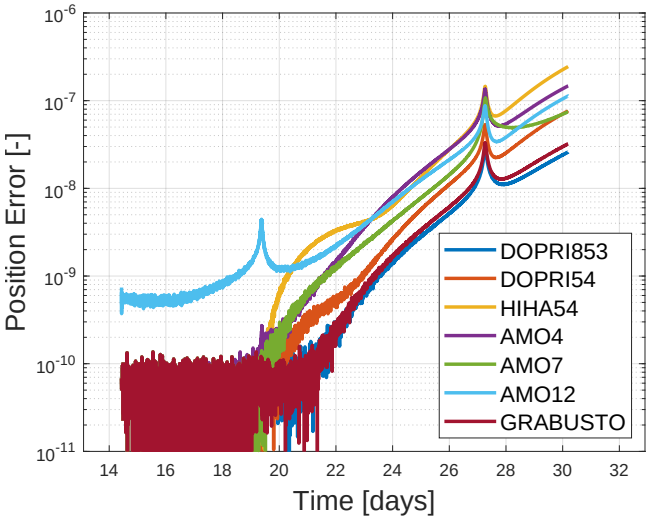

Figure 6.11: Selected position error profiles as a function of time for different integrators.

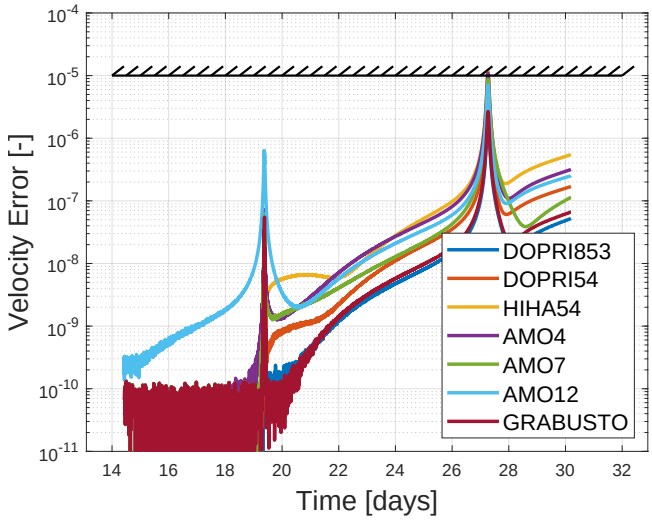

Figure 6.12: Selected velocity error profiles as a function of time for different integrators.

accumulate errors in a very similar fashion, displaying peaks at the flyby points. As experienced in Figure 6.7, the second flyby error decreases when moving away from the Moon and the rounding error dominates the profile before the first flyby. Focusing on AMO7 and AMO12, although the error behavior towards the end is roughly the same, AMO12 shows a considerably larger error at the beginning of the trajectory, before the first flyby, something not experienced by any other integrator. This may be because multi-step integrators need to be initialized with a different integration strategy to obtain the number of points used in the interpolation. As AMO12 requires an initialization of twelve steps (compared to the seven required by AMO7), the accuracy at the beginning of the trajectory may be affected, especially when the error is dominated by the rounding error.

Henceforth, even though AMO12 would still be a promising integrator, AMO7 is chosen as the final integrator, with relative and absolute tolerances of 10^{-8}, a minimum step size of 10^{-15}, and a maximum of 10^{-3}. Looking at Figure 6.10, one could argue that for roughly the same computational effort, higher accuracy can be achieved just by choosing tighter integration settings. Nonetheless, it is uncertain that this steep slope in accuracy can be extrapolated to all trajectories in the transfer (the accuracy level may be maintained, but more function evaluations might be required). Moreover, being the flybys so detrimental to the trajectory accuracy, the error estimation may be poor in trajectories with three or more flybys. This topic will be addressed in the verification section (Section 8.1). However, it can be safely concluded that AMO7 and AMO12 will generally be the least computationally expensive integrators for this dynamic system and these equations of motion.

6.3 Design Space Exploration

In this section, the problem itself is analyzed using the *Taguchi method* (Taguchi, 1987), theory belonging to the branch of *Design Of Experiments* (DOE). Then, the variance of the outcome of the experiments for both use cases is subsequently statistically examined with the method of Analysis of Variance (ANOVA). As such, conclusions can in principle be drawn to identify promising design variable bounds or suitable constraint-handling strategies. Moreover, understanding the problem and its complexity provides insight into how to overcome the optimization challenges, or even to assess if optimization of a fitted response surface would be appropriate.

6.3.1 Design of Experiments: The Taguchi Method

Problems with numerous design variables (factors) make their analysis extremely cumbersome, as a large number of simulations (experiments) are required to explore the design space effectively. A usual approach is the so-called *Monte Carlo simulation*, which randomly samples the design domain according to a set of probability density functions for each factor, which depend on the physical behavior of the factors. Uniform and normal distributions are common examples of these probability functions. The main advantage of this method is its supreme suitability to identify trends in the data. However, as demonstrated by Mistree et al. (1993), the outcome of the Monte Carlo method is highly dependent on the chosen pseudo-random seed and the number of experiments. As a result, statistics such as the variance or the mean value of the response significantly vary according to these parameters. Furthermore, a vast number of experiments are usually required to properly estimate these statistics, especially for problems with many factors. To solve these issues, an alternative approach, DOE, can be used.

DOE is defined as "the technique of defining and investigating all possible conditions in an experiment involving multiple factors" (Roy, 2010), also referred to as *factorial design*. For instance, if one has a problem with four factors each of which is varied over two levels, the problem would have a total of 2^4 = 16 combinations. Then, one or more experiments for each combination would be carried out, and the results would be analyzed. In this case, as the outcome of the experiments is constant for a given set of design variables, there is no need for experiment repetition.

Because most engineering problems deal with multiple factors with continuous values, it is desirable to find a balance between the accuracy of the output and the number of experiments by selecting an appropriate number of factor levels. As explained by Roy (2010), for preliminary analysis, two levels are usually used, however, to obtain an estimation for the curvature behavior of the problem on top of the linear one, in this study an extra level is selected, since the problem is predicted to be highly complex and non-linear. Hence, the factorial design for this problem would consist of ten factors varied over three levels: a minimum value, a maximum value, and the average of the two (the nominal value). This configuration would lead to a total of 3^{10} = 59,059 combinations, which is considered to be too large for the necessary computational costs of this problem, especially if several runs with different configurations (design variable bounds) and for various use cases are to be carried out.

Paraphrasing Montgomery et al. (2009), a possible solution to have fewer runs would be to use a so-called *Central Composite Design* (CCD), which consists of a factorial design of the factors varied over only two levels (maximum and minimum), one run with all factors at their nominal level, and a set of runs varying each factor one-at-a-time relative to the nominal, with a value at a normalized distance $\pm\zeta$ from the nominal value for each factor. ζ is normalized with the distance between the nominal and the two-level factorial design bounds. With this design, the factorial design would estimate the linear effects, whereas the extra runs provide a measure for the quadratic effects. Having ten factors, only 2^{10} + 1 + 2×10 = 1045 runs would be necessary. Nevertheless, to accurately estimate the quadratic effects, the CCD ought to be *rotatable*, i.e., the normalized distance to the center of the design (the nominal experiment) is the same for all design points (experiments), entailing that the prediction variance depends only on this distance (Siva Prasad et al., 2012). However, from the NIST/SEMATECH (2003), the required normalized distance to the nominal values for a rotatable design depends on the number of factors, N, according to:

$$\zeta = \left(2^N\right)^{0.25} \tag{6.1}$$

Then, the required value of ζ for ten factors would be 5.66, meaning that the distance has to be 5.66 times greater than that between the nominal and the factorial design bounds. This would be an issue with factors such as T_{man}, as there are physical limits to the bounds (the TOF cannot be shorter than zero). In addition, τ_{orb_1} is necessarily between 0 and 1. Figure 6.13 displays the schematics of the

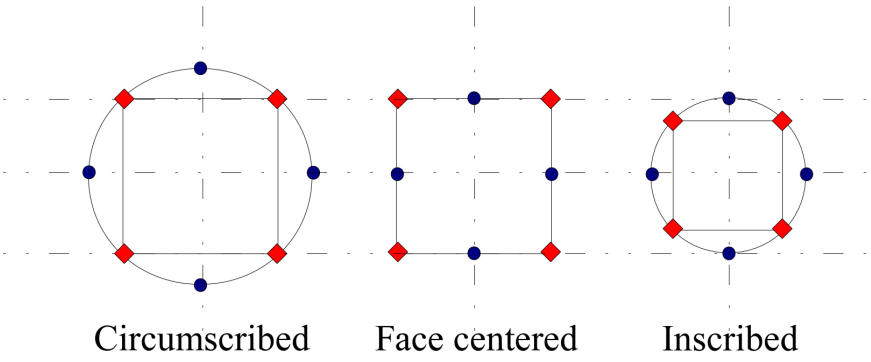

Figure 6.13: Diagram of the different CCD configurations commonly used (Cai, 2013).

three most commonly used CCD configurations. Being the intersection between the vertical dotted lines and the middle horizontal line the nominal point, and the outer horizontal lines the bounds of interest for the analysis, the *circumscribed* configuration would place the extra runs to estimate the curvature "outside" the factorial bounds (the square vertices). As is evident from the figure, this configuration is rotatable since all design points are at the same distance from the center. However, as already explained, it is not suitable for this analysis due to the physical limits of the problem. A potential solution would be to use an *inscribed* configuration (also rotatable), fixing the extra runs to the physical limits and scaling down the factorial design bounds. Nevertheless, even though the curvature behavior may be accurate, the obtained results would only be precise within the region of the factorial design bounds. With such a large value of ζ, the analysis would not be meaningful. A compromise would be to opt for the *face-centered* configuration, having the factorial bounds and the extra runs at the same level. The main disadvantage of this setting is that it is non-rotatable (the square vertices are farther away from the center than the points on the sides), thus, the accuracy of the curvature estimation will probably be poor (Siva Prasad et al., 2012), making it undesirable for the analysis of a problem as complex as this one. In conclusion, CCDs ought to be discarded as a suitable DOE method due to the large number of factors and the physical limits of the problem.

Orthogonal Arrays

Going one step further, one can greatly reduce the required number of experiments while still obtaining statistically accurate results utilizing the aforementioned Taguchi method. As thoroughly explained by Roy (2010), this method is based on the use of the so-called *orthogonal arrays*, which choose a subset of factor combinations out of the full factorial design (thus falling within the category of *Fractional Factorial Designs* (FFDs)), while ensuring that for any pair of factors, all combinations of factor levels occur an equal number of times. Consequently, the number of runs at each level of every factor is also the same. These characteristics of the columns of an orthogonal array constitute the *balancing* property of these arrays. This property guarantees consistent results regardless of the selected orthogonal array, as proven by Mistree et al. (1993), and allows obtaining almost the same degree of insight as would be acquired with a full factorial design (Roy, 2010).

To visualize this balancing property, Table 6.6 displays an orthogonal array consisting of nine experiments with four factors varied over three levels ($L_9(3^4)$). As such, instead of $3^4 = 81$ experiments, only nine are carried out, and, in this case, each level combination between any two columns takes place only once. For instance, the combination 1-2 between factors A and B can only be seen in experiment 2, and the same goes for the combination 3-1 between columns B and D in experiment 9 or any other combination. Lastly, it is evident from the table that for every factor three experiments are always carried out at each level (1, 2, and 3). If the problem had three factors instead of four, this orthogonal array could still be used simply by choosing three of the columns.

Table 6.6: Possible orthogonal array for four factors varied over three levels ($L_9(3^4)$) (Roy, 2010).

Experiment	Factor A	Factor B	Factor C	Factor D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

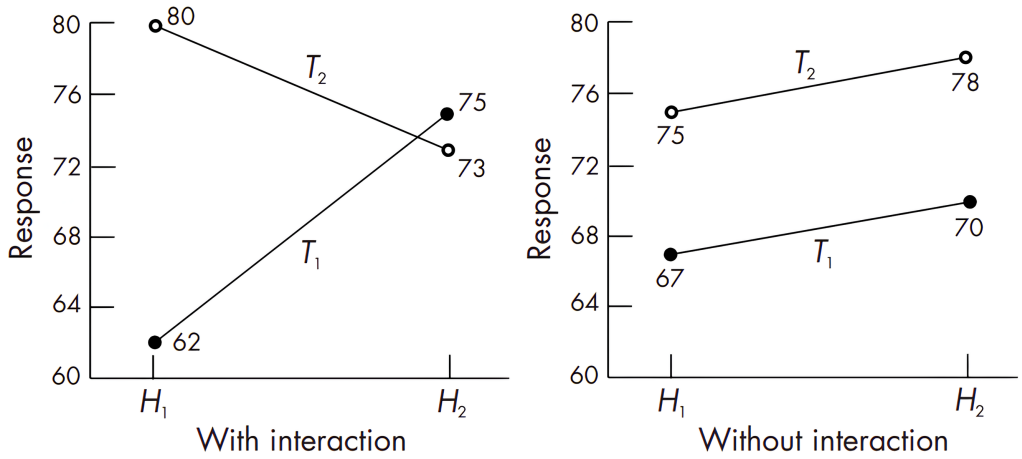

Figure 6.14: Effect on the response of an interaction between two parameters, T and H, at two different levels, compared to a case without interaction (Roy, 2010).

As previously stated, once the orthogonal array experiments have been set up and executed, the data can then be analyzed to obtain accurate estimations of statistics such as the mean or the standard deviation for the given design bounds. Furthermore, the contribution of each of the factors to the variability of the output or response can be investigated, as well as the effect of the interactions between said factors. The interaction contribution, AB, between two factors A and B is a measure of the change in the contribution of factor A to the response depending on the value of factor B (and vice-versa). This effect is showcased in Figure 6.14, where it can be seen that when there is interaction between the factors H and T, the effect of varying H from H_1 to H_2 differs depending on the value of T. This is not the case when no interaction is present, as seen in the second plot. The estimation of the interaction contributions in addition to the main factor contributions is of interest as strong interactions can make a certain factor look irrelevant when, in reality, its interaction with another parameter (or parameters) may be "eclipsing" the effect in the data (Montgomery et al., 2009). The problem at hand, apart from being expected to be highly non-linear, is predicted to have significant interaction contributions between most of the parameters.

Confounding

One of the main difficulties when analyzing problems with numerous interacting parameters with FFDs is that, since not all possible factor combinations are explored, when computing the contri-

bution of factor A, for instance, the estimation may include the contribution of, for example, the first-order interaction CD, meaning that the obtained value is actually a measure of $A + CD$ instead of simply the contribution of factor A, making the estimations unreliable. This is known as *confounding* or *aliasing*. A fractional factorial design with the explained characteristic would be said to have *resolution* III (NIST/SEMATECH, 2003), as there is confounding between the main factor contributions and the first-order interactions (I + II = III). Subsequently, if the main factor contributions are free of confounding, but the first-order interactions are confounded with each other, the resolution of such an array would be IV (II + II). If confounding between first-order interactions is also not present, but they are aliased with second-order interactions (e.g., ABC), the resolution would be V; and so on. Of course, a full factorial design would have maximum resolution. For this problem, to keep the complexity of the computations at an acceptable level and to make it easier to draw conclusions, only the main factor contributions and the first-order interactions will be estimated. In addition, it is hypothesized that the higher-order interactions are less relevant than the aforementioned contributions. Therefore, an orthogonal array of resolution V is sufficient for this design space exploration. If this hypothesis is not correct, a Monte Carlo simulation or even a CCD would be more suited due to the computational complexity required to estimate the contributions of the higher-order terms.

As explained by Roy (2010), Taguchi himself dedicated strong efforts to identifying the confounding between the columns of the orthogonal arrays. Consequently, he developed *triangular* or *interaction tables*, which display for a given orthogonal array what contributions will be confounded in each column. Hence, by using these tables, one could choose the array columns for the factors (also formulated as "assigning the factors and interactions to appropriate columns") so that no undesirable confounding takes place and the required array resolution is achieved. These tables are provided by Taguchi (1987) and Roy (2010), however, they are not large enough for a problem with ten factors and 45 first-order interactions. As generating these tables requires a very high level of expertise in the fields of DOE and statistics, it is not feasible to generate a suitable one for this research. Hence, an alternative approach will be followed to deal with confounding, which will be explained in Subsection 6.3.3.

To conclude, FFDs allow for an accurate estimation of the output's statistical behavior, with a significantly reduced number of simulations, greatly outperforming CCDs and Monte Carlo simulations when many factors are present. However, selecting an appropriate orthogonal array with enough resolution is crucial.

6.3.2 Analysis of Variance (ANOVA)

One of the biggest advantages of full factorial designs is the fact that Analysis Of Variance (ANOVA) can be applied without the need for a response surface fitting, unlike Monte Carlo methods. Fractional factorial designs with orthogonal arrays also possess this characteristic, thanks to the balancing property of the arrays. ANOVA is employed to analyze, for the given design bounds, the variance of the response and compute the percentage of contribution of the factors and interactions, i.e., a measure of how much a factor or interaction explains the variability in a data set. In addition, this method provides a measure of the confidence level as well. The following ANOVA equations are retrieved from the explanations by Roy (2010).

Sum of Squares

The first step of the method is to compute the total sum of squares, SS_T, via:

$$SS_T = \sum_{i=1}^{n_{exp}} (y_i - \frac{y_{sum}}{n_{exp}})^2 = \sum_{i=1}^{n_{exp}} y_i^2 - \frac{y_{sum}^2}{n_{exp}} = \sum_{i=1}^{n_{exp}} y_i^2 - CF \tag{6.2}$$

where y_i is the outcome of each experiment, y_{sum} is the sum of all response values, n_{exp} is the number of experiments, and CF is denoted as the correction factor. Then, defining SS_A as the sum of squares of factor A (for instance), and SS_{AB} as the sum of squares of the interaction between factors A and B, the sum of squares of the factors and the first-order interactions can be obtained through

$$SS_A = \sum_{i=1}^{k_A} \frac{A_i^2}{n_{lev}} - CF \tag{6.3}$$

$$SS_{AB} = \sum_{i=1}^{k_A} \sum_{j=1}^{k_B} \frac{(A_i B_j)^2}{n_{int}} - CF - SS_A - SS_B \tag{6.4}$$

In these equations, k_A is the number of levels of factor A (in this problem common to all factors), n_{lev} is the number of experiments at each factor level (common for all factors), and A_i is the sum of the response values for all experiments with factor A at level i. In addition, the number of experiments per interaction level is represented by n_{int} and the sum of the response values for all experiments with factor A at level i and factor B at level j is denoted as $A_i B_j$. An interaction level is any combination of factor levels between the interacting factors (2-1, 2-3, 3-3...). Having three levels for each factor in this problem, the number of interaction levels would be $3^2 = 9$ for the first-order interactions. The last sum of squares to be computed is the error sum of squares, SS_e, which is simply the sum of squares not accounted for by the modeled factors and interactions:

$$SS_e = SS_T - \sum_i SS_i - \sum_{ij} SS_{ij} \tag{6.5}$$

Degrees of Freedom

The DOF indicate the number of independent quantities that can vary in the analysis, with more degrees of freedom allowing for a better fit of the data, thus improving the statistical estimates. The total number of degrees of freedom (DOF), f_T, is, as usual, the number of output data points minus one (one degree of freedom is used to estimate the mean). Then, the number of DOF of each factor f_A corresponds to the number of factor levels minus one and the interaction DOF are obtained through the product of the degrees of freedom of each factor. Similarly to SS_e, the DOF of the error, f_e, is the remainder of the total DOF minus the modeled DOF. Mathematically, all this would be written as:

$$f_T = n_{exp} - 1 \qquad f_A = k_A - 1 \qquad f_{AB} = f_A f_B \tag{6.6}$$

$$f_e = f_T - \sum_i f_i - \sum_{ij} f_{ij} \tag{6.7}$$

Variance and Variance Ratio

Once the sums of squares and DOF have been computed, all variances, V, which quantify how much a set of values spreads out from their mean, reflecting the degree of variability in the data, can be obtained by a simple division:

$$V_A = \frac{SS_A}{f_A} \qquad V_{AB} = \frac{SS_{AB}}{f_{AB}} \qquad V_e = \frac{SS_e}{f_e} \tag{6.8}$$

Furthermore, by dividing the variances of the factors and interactions by the variance of the error, V_e, one can obtain the variance ratios or *F-ratios*, F:

$$F_A = \frac{V_A}{V_e} \qquad F_{AB} = \frac{V_{AB}}{V_e} \qquad F_e = \frac{V_e}{V_e} = 1 \tag{6.9}$$

The F-ratio measures "the significance of the factor under investigation with respect to the variance of all of the factors included in the error term" (Roy, 2010). The higher the value of this ratio, the more confidence one has that said factor is the one contributing to the sum of squares of the response (assuming no confounding). Thus, by increasing the DOF of the error, f_e, the error estimation becomes more accurate, reducing its variance, V_e, and consequently increasing the variance ratio and the confidence in the results. To quantify the confidence level, several tables (*F-tables*) depending on the desired percentage of confidence are provided in Appendix B of the work by Roy (2010). Then, given the number of DOF of the numerator (factor or interaction contribution) and the denominator (error), the table provides a value that ought to be lower than the value of the F-ratio, otherwise, the factor does not contribute to the sum of squares within the confidence level.

Percentage of Contribution

Having computed the DOF and the error variance, the *pure sum of squares* of, for instance, factor A, SS'_A, of interaction AB, SS'_{AB}, and of the error, SS'_e, would be obtained as:

$$SS'_A = SS_A - f_A V_e \tag{6.10}$$
$$SS'_{AB} = SS_{AB} - f_{AB} V_e \tag{6.11}$$
$$SS'_e = SS_e + \left(\sum_i f_i + \sum_{ij} f_{ij} \right) V_e \tag{6.12}$$

The pure sum of squares aims to estimate the "true" contribution of the factors by correcting for the error variance and the degrees of freedom. Then, the percentage of contribution, P, of the factors, the interactions, and the error term can be computed by dividing each pure sum of squares by the total sum of squares:

$$P_A = 100 \frac{SS'_A}{SS_T} \qquad P_{AB} = 100 \frac{SS'_{AB}}{SS_T} \qquad P_e = 100 \frac{SS'_e}{SS_T} \tag{6.13}$$

As P refers to the percentage of contribution to the variance (which deals with sums of *squares*), when comparing the relative influence between parameters the square root of the ratio ought to be taken. For example, if P_A = 27% and P_B = 3% with enough confidence level, it can be concluded that factor A influences the response (not the variance) $\sqrt{\frac{27}{3}}$ = 3 times more than factor B for the provided bounds.

To finalize the analysis, the quality of the model can be assessed with the *coefficient of determination*, R^2. This coefficient represents the proportion of the total variance that is explained by the employed model. In addition, the *adjusted* coefficient of determination, R^2_{adj}, is equivalent to R^2 but corrected for the degrees of freedom in the model (the number of predictors), making comparisons between models with different numbers of predictors fairer. These two coefficients are obtained by:

$$R^2 = 1 - \frac{SS_e}{SS_T} \qquad R^2_{adj} = 1 - \frac{f_T SS_e}{f_e SS_T} = 1 - \frac{P_e}{100} \tag{6.14}$$

Pooling

By examining Equations (6.11) and (6.12), one can realize that the pure sum of squares of a factor or interaction can become negative, meaning that its contribution to the variance of the response is negligible. When this is the case, that factor or interaction needs to be *pooled*: it should be removed from the modeling (thus increasing SS_e), its degrees of freedom added to the error DOF, and the ANOVA analysis recomputed. Factors ought to also be pooled if the F-ratio for a given factor or interaction is lower than the required value for the desired confidence level of the analysis (obtained with the F-tables). In addition, as suggested by Taguchi (1987), the least significant terms should be pooled until $f_e \approx \frac{f_T}{2}$ to ensure that the estimation of the error term is accurate enough and that significant factors are not eclipsed by a too large number of modeling variables.

It is likely that for a given ANOVA calculation run, several factors or contributions do not satisfy the criteria to not be pooled. Then, the recommended procedure is to only pool the factor or interaction with the lowest sum of squares and recompute everything again. Hence, factors are pooled one at a time to avoid removing too many modeling variables at once. If not automated, this procedure can be considerably time-consuming.

6.3.3 Experiments Set-Up

The first step is to define the list of experiments to be carried out, i.e., choose an appropriate orthogonal array. Having ten variables varied over three levels and 45 first-order interactions, the required number of degrees of freedom is 10×2 + 45×4 = 200 (Equation (6.6)). Thus, the minimum number of experiments is 201, leaving no simulations to estimate the error. In addition, the number of columns of such an array would be obtained by (Leung and Y. Wang, 2001):

$$\#\text{Columns} = \frac{\#\text{Rows} - 1}{\#\text{Levels} - 1} \tag{6.15}$$

where the number of rows equals n_{exp}. As such, said orthogonal array would have 100 columns, each of which with two degrees of freedom (Forestryani et al., 2022). However, to obtain greater statistical confidence in the results, a larger array is required. Moreover, having more experiments may help to identify trends in the response. To generate the Taguchi orthogonal arrays, the algorithm described by Leung and Y. Wang (2001) is used, by means of the MATLAB function *oa_permut* (Jeppu, 2024).

To obtain a resolution V orthogonal array with so many factors and interactions, the proposed approach is to generate an orthogonal array much larger than required to minimize the probability of confounding when sampling ten random columns to assign the factors. The chosen number of experiments is 3^9 = 19,683, leading to a $3^{10\text{-}1}$ FFD and a complete orthogonal array with 9,841 columns, from which ten are selected to assign the factors and set up the experiments. With this array, all experiments will be run with the following test function:

$$y = \sum_{i=1}^{N} x_i + \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j \tag{6.16}$$

where N is the dimension of the design space, i.e., the number of factors (ten), y is the response value, and x_i corresponds to the value of factor i at the level specified by the experiment. Provided that the experiment bounds are the same for all factors, the contribution of each factor to the variance should be equal for all factors, as well as the contribution of each interaction should be equal for all interactions. This is because the coefficients (weights) of the proposed function are equal for all linear and quadratic terms (x_i and x_i^2), regardless of the factor. The same goes for all first-order linear interaction terms ($x_i x_j$). If differences in the contributions are found, there is confounding between the factors and/or interactions with the chosen array columns. Then, ten new columns are

selected from the complete orthogonal array and the process is repeated until the outcome of the test is satisfactory. For further verification, three different orthogonal arrays were generated following this approach, and the first ANOVA analysis of Subsection 6.3.4 was run once with each array. Comparing the three sets of results, no factor or interaction contribution varied by more than 1%, suggesting that the results are reliable.

As the orthogonal array has three times fewer runs than a full factorial design, more ANOVA analyses can be carried out, allowing for additional bounds combinations to be assessed. In this research, two sets of ANOVA analyses will be conducted: the so-called *general runs*, with the factor bounds aiming to span the feasible domain, and the *local runs*, which are centered around the nominal solution from Section 6.2, varying its factor values firstly by 1%, then by 10%, and lastly by 50%. For the general runs, both use cases will be utilized to aim to draw more general conclusions.

For the general runs, the maximum TOF of the optimal control phase was chosen to be 80 days. To define the maximum TOF for the target manifold phases, the approach was to discretize the halo orbits in 20 points, spaced by equal flight times, and propagate the corresponding stable or unstable manifolds for a long enough time such that the regularity in the manifolds is already lost due to the perturbation introduced by ϵ_p. The outcome for the first use case is displayed in Figure 6.15, where one can see the departure orbit on the right side of the Moon, the target orbit on the left, and the 20 stable manifolds, chosen to go in the direction of the Moon (interior manifolds) due to the geometry of the problem. Each color denotes a manifold arising from a different orbit point. As can be concluded from the figure, after 38 days the manifold "tube" loses its regularity and all manifolds pass close to the Moon at least once, being this TOF thus suitable for the optimization.

Figure 6.16 shows the result of the equivalent analysis for the second use case. Even though the right-most orbit is the departure orbit, as in this use case the optimal control phase uses backward propagation, it becomes the target orbit for the optimization, and the unstable manifolds become the target manifolds. Again, due to the use case geometry, the manifold direction was chosen towards the Moon. The target orbit of the first use case is evidently more stable than the one for the second case since in the latter all manifolds reach the lunar vicinity in less than 18 days (compared to 38 days, describing more revolutions). The regularity is also lost after the chosen period. Nonetheless, because of the lower stability in the second use case, the manifolds display a more uniform tube-like shape (showcased by the smooth color gradient), presenting a higher resemblance to the invariant manifolds of Section 2.6. Regarding the departure manifold phases, T_{man} was chosen in a similar fashion. 22 days were deemed appropriate for the first use case, choosing again the interior manifolds, whereas 18 were selected for the second use case, opting for the outer manifolds in this instance (the ones heading away from the Moon). For a real optimization, it would probably be convenient to use both the interior and exterior manifolds as it is uncertain which are more advantageous, however, as for this analysis only three levels are used, more insight will be obtained by focusing on a single manifold direction.

τ_{orb_1} and τ_{orb_2} were both bounded between 0 and 1, expecting a periodic result. Choosing the bounds for the costate parameters requires more engineering judgment than the rest of the factors. In principle, there are no physical bounds for the values of these parameters, and depending on the problem the solution variables can be significantly different. However, finding a suitable design domain for the variables is important to force the heuristic optimizer to focus its search on promising regions. If the bounds are too wide, the optimizer may struggle to thoroughly explore these regions and get "stuck" in worse-performing feasible points, especially if solutions can be found at several orders of magnitude (the optimizer will tend to find solutions within the order of magnitude of the bounds).

As λ_m cannot increase over time, and the transversality condition for the mass costate in Equation (5.13) requires $\lambda_{m_f} = 0$, the initial value for this costate parameter ought to be positive. Nevertheless, when using low-thrust bang-bang control, the only way to obtain lower propellant mass solutions

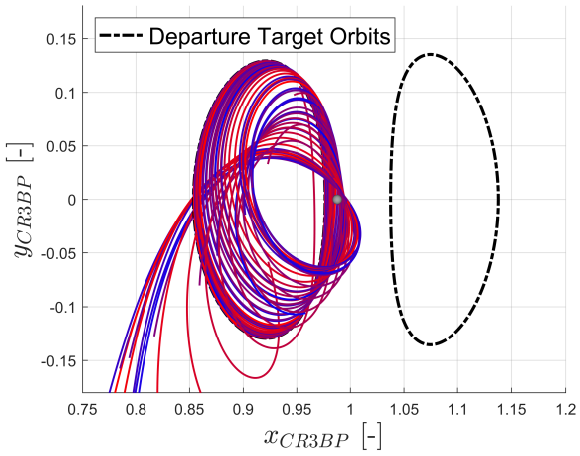

Figure 6.15: Subset of stable target manifolds propagated for 38 days for the 1st use case.

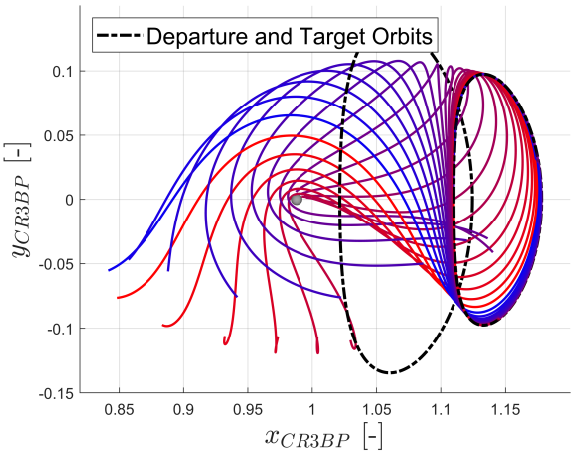

Figure 6.16: Subset of unstable target manifolds propagated for 18 days for the 2nd use case.

at larger TOFs is to include coasting phases in the trajectory. Thanks to manifold theory, this can already be achieved without the need for coasting periods during the optimal control phase. However, incorporating these will provide a wider range of trajectories of even better performance. Then, looking at Equation (5.11), which governs the coasting periods (present when $S > 0$), one realizes that a negative value for λ_m will tend to trigger more coasting phases. Hence, even though the solutions will not be rigorous mathematical minima with respect to the mass state, acknowledging the difficulty in finding solutions that satisfy the constraints, it is predicted that by allowing negative values for λ_{m_0} the number and quality of the results will be increased.

Furthermore, in Equation (5.11) the orders of magnitude of λ_m and $\boldsymbol{\lambda}_v$ play a big role in triggering coasting phases. As the last term of the equation is always negative, the lower the order of magnitude compared to 1 (the first term), the more and longer coasting phases will be obtained. Then, λ_m will promote the spacecraft to coast or to thrust depending on if it is positive or negative. The approach followed in this research to achieve a balance between thrusting and coasting periods is to make all terms in Equation (5.11) of the same order of magnitude. Thus, λ_m should not be much larger than 1, whereas $||\boldsymbol{\lambda}_v||$ should be one order lower since $\frac{I_{sp} g_0}{m} \approx 19$. Smaller orders of magnitude may restrict the design space too much, excessively prioritizing coasting phases, while all trajectories obtained with larger orders of magnitude that do not coast can be replicated with these bounds by having a positive enough value for the mass costate [4].

Regarding the three components of $\boldsymbol{\lambda}_{r0}$, since the behavior of $\boldsymbol{\lambda}_r$ and $\boldsymbol{\lambda}_v$ strongly depend on each other (Equation (5.8)), it is desirable to choose the bounds such that the order of magnitude of the velocity costates does not vary too much. Examining the equation for $\dot{\boldsymbol{\lambda}}_v$, one can observe that both $\boldsymbol{\lambda}_r$ and $\boldsymbol{\lambda}_v$ contribute to the derivative on the same order of magnitude, provided that both vectors also have the same order. Hence, $\boldsymbol{\lambda}_{r0}$ was chosen to have the same bounds as $\boldsymbol{\lambda}_{v0}$. Lower bounds may be too restrictive, and larger bounds might excessively modify $\boldsymbol{\lambda}_v$. Moreover, as the position and velocity have been normalized in this system, it is sensible to assume bounds of the same order. It is noteworthy that with this configuration, $\boldsymbol{\lambda}_r$ tends to be one order of magnitude larger than $\boldsymbol{\lambda}_v$, especially when flybys are present (as can be seen in Figure 6.5). Lastly, no compelling reason has been found to specify different bounds within the components of $\boldsymbol{\lambda}_{r0}$ or $\boldsymbol{\lambda}_{v0}$.

[4] Recall that if no coasting phases are present in a trajectory, λ_m does not play a role and the trajectory is fully defined by the ratio between the six other costate parameters instead of by their actual value (Section 5.1).

Table 6.7: Parameters and their level values for the general runs of both use cases.

Parameter	**1st Use Case Levels**	**2nd Use Case Levels**	**Unit**
τ_{orb_1}	0.0, 0.5, 1.0	0.0, 0.5, 1.0	-
T_{man}	0.0, 11.0, 22.0	0.0, 9.0, 18.0	days
$\boldsymbol{\lambda}_{r0}$	-0.1, 5.0·10^{-3}, 0.11	-0.1, 5.0·10^{-3}, 0.11	-
$\boldsymbol{\lambda}_{v0}$	-0.1, 5.0·10^{-3}, 0.11	-0.1, 5.0·10^{-3}, 0.11	-
λ_{m0}	-1.0, 0.0, 1.0	-1.0, 0.0, 1.0	-
τ_{orb_2}	0.0, 0.5, 1.0	0.0, 0.5, 1.0	-

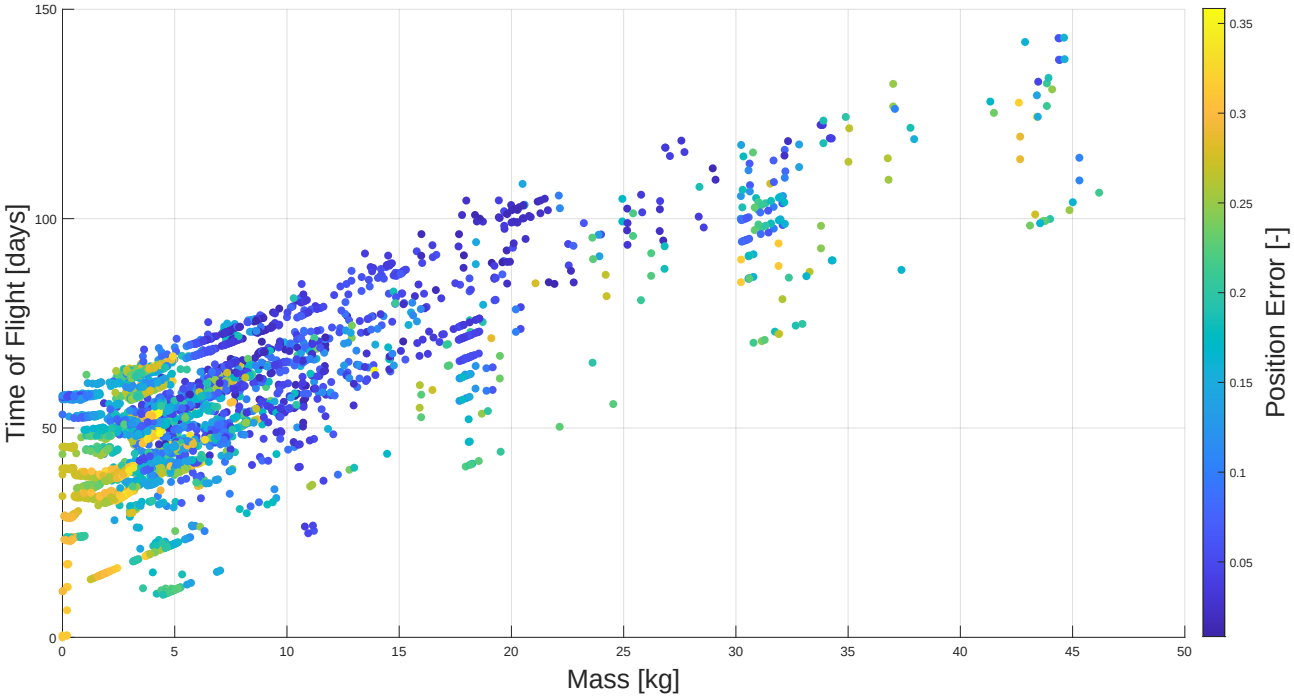

Figure 6.17: Comparison between propellant mass, TOF, and position error for the different experiments of the general FFD for the 1st use case.

The factor bounds for the general runs of the use cases are indicated in Table 6.7. The position and velocity costate bounds are not centered around zero to avoid the case in which all costate parameters are exactly zero. If this is the case, the thrust direction is undefined, and $\dot{\boldsymbol{\lambda}}_r$ and $\dot{\boldsymbol{\lambda}}_v$ will always be zero. To assess the suitability of the chosen bounds, which are only an initial guess, several general runs will be carried out varying the order of magnitude of the bounds of $\boldsymbol{\lambda_0}$ (and keeping the rest constant). Hence, if the bounds of Table 6.7 are referred to as $\boldsymbol{\lambda}_0^*$, for each use case, three extra runs will be conducted with differing bounds: $0.1\boldsymbol{\lambda}_0^*$, $10\boldsymbol{\lambda}_0^*$, and $100\boldsymbol{\lambda}_0^*$.

Finally, for all ANOVA runs, w_{pos} and w_{vel} in Equation (5.18) are chosen to be equal to 1, as preliminary values. In addition, the Jacobi constant error is omitted in the equation, and the closest state computation explained in Subsection 5.2.3 is performed at every spacecraft state, regardless of the Jacobi constant error or the distance to the Moon. This is because these functionalities were added later on in the research.

6.3.4 Results

General Runs

Figure 6.17 plots the required propellant mass against the TOF of all experiments for the first use case (using the bounds from Table 6.7). In addition, the non-dimensional position error is color-coded within the points themselves. As one would expect, there is an identifiable linear trend in the data, since the longer the spacecraft flies, the longer it will tend to thrust. This relationship can be modified by adding coasting phases, which explains the width of the data stream in terms of TOF. As such, the lower limit corresponds to the trajectories that do not use manifolds or coast during the optimal control phase. This is a physical limit because, for a given TOF, T, the thrusters can only expel a maximum mass of $\dot{m}_{dim}T$, where $\dot{m}_{dim}$ is the dimensional mass flow. The upper limit corresponds to the trajectories that make use of the upper bound of T_{man} and of most of the target manifold. Moreover, some coasting periods may well be present during the optimal control phase. Nevertheless, this is not a physical limit, as longer TOFs can be obtained by longer coasting periods during the optimal control phase. This also explains why the upper line presents some outliers in comparison to the lower line. However, the fact that an upper limit can be identified hints at the fact that either not enough coasting phases were triggered, or that they increased the state error relative to the target trajectory, stopping prematurely. This does not necessarily suggest that the bounds are not adequate since only $\sim$20,000 experiments were performed, but it should lead to setting a focus on generating these coasting periods.

Regarding the position error, it can be seen that relatively low error trajectories can be obtained at a wide range of times of flight and propellant masses, which is something promising to generate an extensive Pareto front. Then, numerous trajectories of zero propellant mass were obtained, corresponding to trajectories for which the initial state of the optimal control phase (the end-point of the departure manifold) is the best. As it is logical, these exhibit large position errors. Still, for the optimizations, it may be beneficial to penalize these trajectories in terms of propellant mass and TOF, as it would be desirable to exclude these individuals from the population as fast as possible. Overall, due to the concentration of the data, it seems to be easier to obtain low TOF trajectories, and connecting to target manifolds appears to be more convenient than directly patching to the target orbit since most of the points are close to the upper bound in Figure 6.17.

To evaluate the quality of the trajectories in terms of state error, the position error of the trajectories of the first use case is compared to their velocity error in Figure 6.18. Both the position and velocity errors span a wide range, limited by the use case architecture, without revealing any clear trends. Relatively low velocity errors can be paired with relatively low position errors and vice versa. From the figure, it can be seen that the largest errors tend to correspond to trajectories with low propellant mass, i.e., the ones whose best state is the initial state or somewhere close. Furthermore, low position and velocity error trajectories (bottom-left corner) can be obtained at considerably different propellant mass values.

Unfortunately, none of the trajectories satisfy the boundary constraints (errors below $3{\cdot}10^{-3}$), evidencing the highly restrictive nature of the constraints. Thus, the behavior that is being analyzed here corresponds to the behavior of the unfeasible transfers, without acquiring any insight into the feasible domain. This fact strengthens the need for the local runs. Nevertheless, a sensible conclusion is that it is generally easier to find lower velocity error transfers than lower position error ones. This can be identified in Figure 6.18 by the large number of points near the x-axis compared to the few that approach the y-axis. Intuitively, this phenomenon can be understood by considering that the thrusters (and the gravity forces) only directly modify the acceleration, which in turn changes the velocity over time. Only the integral of this change with respect to time will cause a change in position. Hence, there is a "delay" in varying the position compared to altering the velocity, making it more difficult to arrive at the desired position value or to explore a wider range of values. Therefore, for the

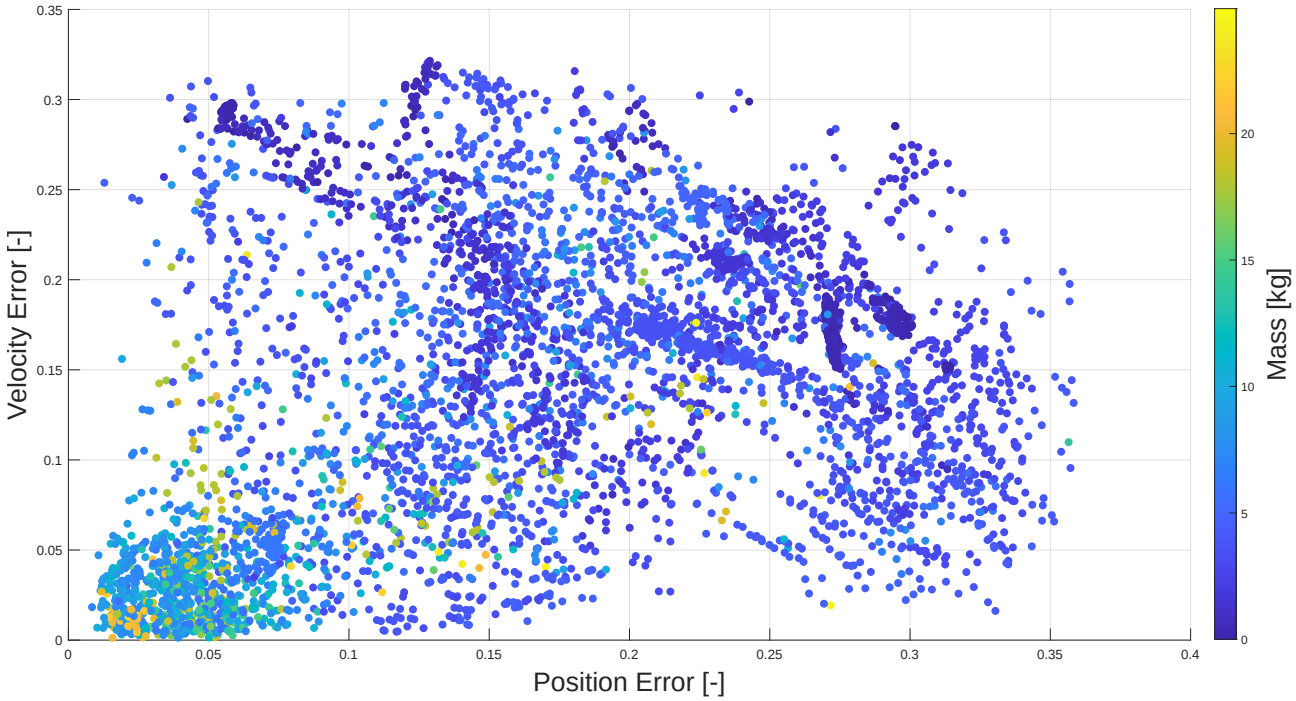

Figure 6.18: Comparison between position error, velocity error, and propellant mass for the different experiments of the general FFD for the 1st use case.

optimizations, it will be better to prioritize the position constraint over the velocity constraint, a fact that will ease the weight tuning.

Figure 6.19 displays the equivalent position versus velocity error plot but for the general run of the second use case instead. Low velocity error trajectories were again less challenging to obtain, generalizing the previous conclusion. Due to the location of the departure and target orbits in space, the maximum position and velocity errors are half the ones found for the first use case. However, the second use case appears to be more demanding since fewer promising trajectories were obtained in this run, as manifested by the sparsely populated lower left corner of the figure. Furthermore, a large fraction of the data is grouped in several clusters of very low mass, meaning that the best state was found near the initial state. The higher complexity of the transfer can be attributed to the larger difference in Jacobi constant between the departure and target orbits compared to the first use case. Probably, including interior manifolds on top of the exterior manifolds for the departure trajectory will help the optimizer find feasible trajectories. The linear relationship between propellant mass and TOF is also present in this use case, with a however narrower width compared to Figure 6.17.

As already described, the influence of the factors and interactions on the response was analyzed through ANOVA. Specifically, four objectives were investigated: propellant mass, TOF, position error, and velocity error. Unfortunately, in none of the use cases nor objectives, an R^2 or R^2_{adj} value above 0.5 was obtained, going as low as 0.16 for the propellant mass. Thus, the modeling error is always primary, suggesting that the problem is highly non-linear and complex, with leading contributions of higher-order interactions. This strong non-linearity may be attributed to several reasons. Firstly, the highly complex nature of the CR3BP can yield rather different trajectories with similar initial states or thrust profiles. Secondly, the strong coupling between the equations of motion, as evident from Equations (2.28) and (5.8), can make the contributions of higher-order interactions leading. As a last hypothesis, the closest state between the optimal control phase and the target phases can greatly differ for similar trajectories, since several connection points along the target trajectory may have very similar state errors, resulting in pronounced differences in propellant mass and TOF.

No general conclusions can be retrieved from the data itself, as it might be confounded with

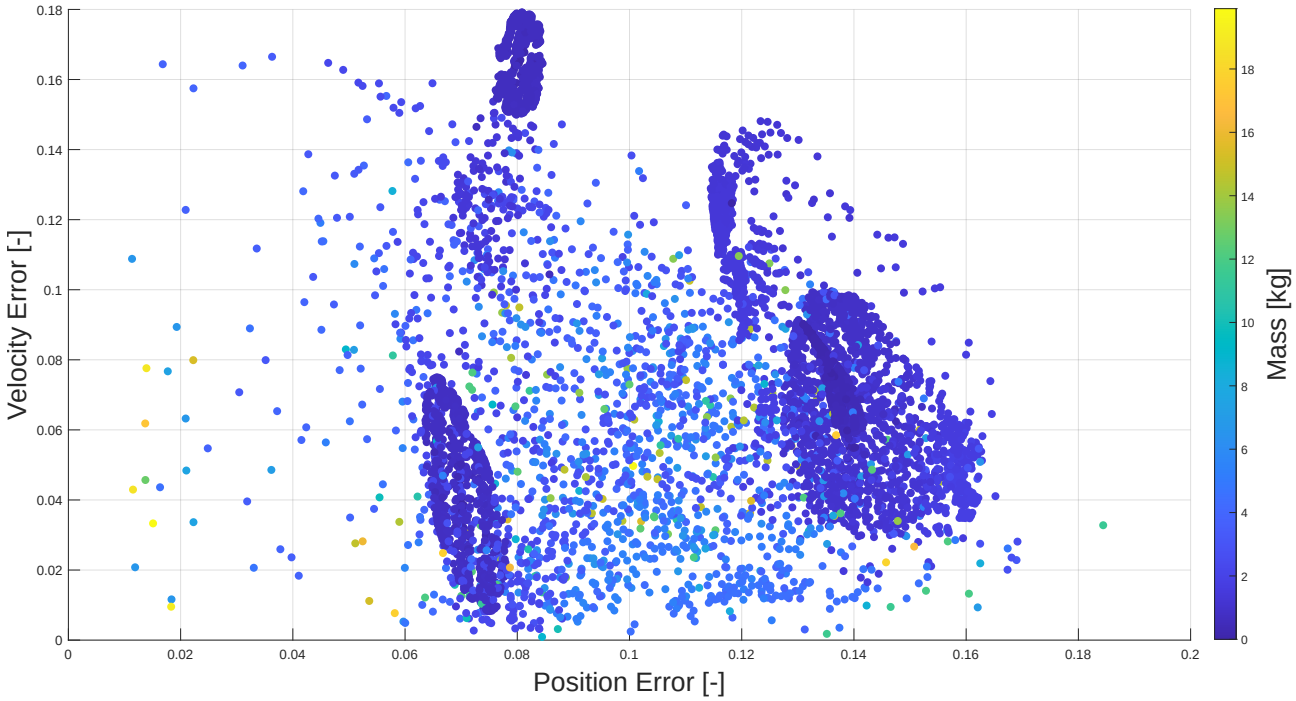

Figure 6.19: Comparison between position error, velocity error, and propellant mass for the different experiments of the general FFD for the 2nd use case.

the contributions of higher-order interactions and the quality of the fit is poor. Moreover, for such a complex problem, an FFD of three levels may well not be sufficient to accurately describe the behavior. Therefore, not only is the optimization challenging due to the very restrictive boundary conditions but also the behavior of the objectives is exceptionally non-smooth and highly complex. Future steps to further study the general problem would be to conduct Monte Carlo simulations to hopefully identify trends in the response as a function of the design variables thanks to the larger amount of data points. In addition, response surface methods may be interesting to recognize relevant behaviors. However, this process may become very time-consuming (a driving factor in this research) and not rewarding enough for the optimizations.

Since no information could be retrieved from the factor contributions to evaluate the design bounds for $\boldsymbol{\lambda}_0$, the approach was to carry out three extra runs with the bounds for $\boldsymbol{\lambda}_0$ set at different orders of magnitude, as described at the end of Subsection 6.3.3. Then, the response averages and standard deviations were compared. The outcome of the analysis for the second use case can be seen in Figure 6.20, where the average position and velocity errors and their standard deviations are plotted against the costate bounds. As usual, the position error average is always considerably larger than the velocity error one. The run with the costate bounds set at 0.1 corresponds to the original run from Table 6.7. Recall that the bounds for λ_{m0} are one order of magnitude larger than for $\boldsymbol{\lambda}_{r0}$ and $\boldsymbol{\lambda}_{v0}$. From the figure, it can be observed that the runs with the costate bounds of 1 and 10 are identical in terms of average and standard deviation, and very similar to the run with costate bounds of 0.1. An examination of the output data revealed that most of the obtained trajectories were equal. The performance is significantly different for the fourth run. These observations can be explained by the fact that with the larger bounds, no coasting periods are triggered during the optimal control phases, so changing the order of magnitude of the costates will not change the trajectories (as explained in Subsection 6.3.3). However, decreasing the order of magnitude led to new trajectories with longer coasting periods and different performances. Overall, the original bounds are deemed to be the most promising, as they present trajectories that promote coasting without excessively constraining the design space.

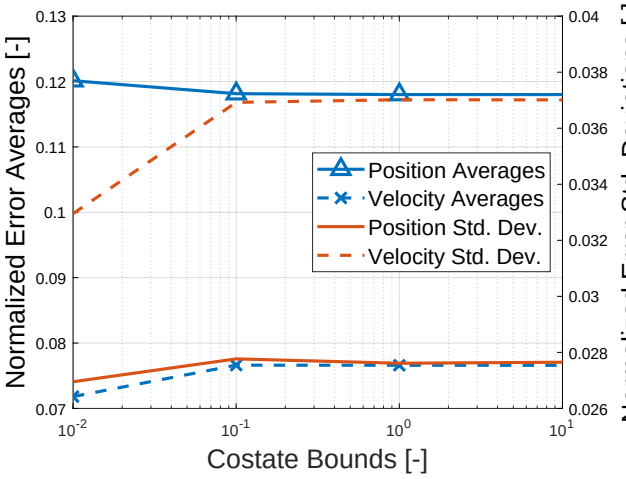

Figure 6.20: Mean and standard deviation of the position and velocity errors as a function of the costate parameter bounds.

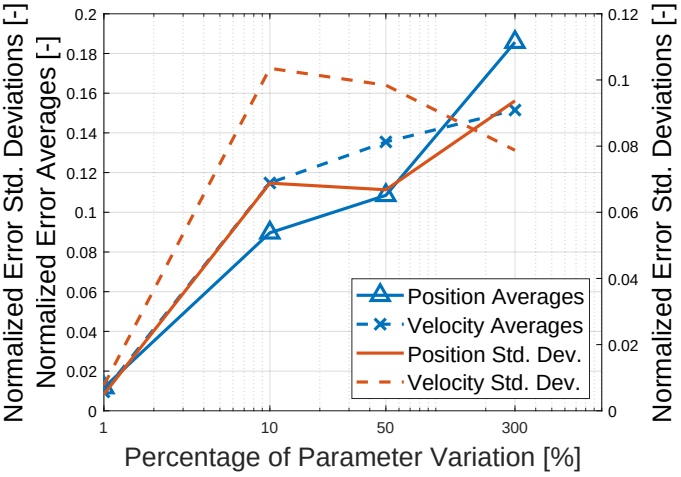

Figure 6.21: Mean and standard deviation of the position and velocity errors as a function of the percentage of parameter variation with respect to the nominal trajectory.

Local Runs

As already mentioned, the local ANOVA rounds are centered around the nominal solution, perturbing the design variables by 1%, 10%, and finally by 50%. Recall that the nominal trajectory corresponds to the first use case. These runs will help analyze the sensitivity of the trajectories, their local behavior, and the span of the feasibility regions.

Firstly, the evolution of the position and velocity error averages and standard deviations as a function of the parameter variation is displayed in Figure 6.21. The 300% variation corresponds to the general run from Table 6.7 (the percentage is an estimation as the actual relative change varies per parameter). The figure shows that the average errors sharply increase with the percentage of variation. The standard deviations display a more unpredictable behavior, although there is a strong increase between the parameter variation of 1% and the one of 10%. Interestingly, even though at 1% the position error is larger than the velocity error, the roles reverse at 10% and 50%, getting the velocity error average again surpassed by the position error average in the general run, not centered around a trajectory. Nevertheless, it would be imprudent to generalize this result to all trajectories.

Secondly, the propellant mass is plotted against the TOF for the run with 1% variation on Figure 6.22. The position error is again added in a color bar. As would be expected, the trajectories with the minimum position error present a similar performance to the nominal trajectory: 6.93 kg and 60.13 days. In this case, the linear trend found in Figure 6.17 cannot be identified, as the variations are very small. It is present however for the runs with variations of 10% and 50%. The interesting characteristic of Figure 6.22 is the difference in ranges between the TOF and the propellant mass. As this trajectory never coasts during the optimal control phase, a reduction in mass requires a reduction in the TOF of the optimal control phase. However, the reduction in mass that some of the trajectories exhibit (the ones with larger position errors) does not translate to the required reduction in total TOF. The reason for this is that the perturbed trajectories find the new minimum state error somewhere else in the target manifold, effectively exchanging TOF in the optimal control phase with TOF in the target manifold phase. This phenomenon of attaching somewhere else to the manifold decreases the error of the trajectories in general, something that will greatly benefit the optimization and is the whole purpose of the proposed search strategy in Subsection 5.2.3. Moreover, the patterns of roughly par-

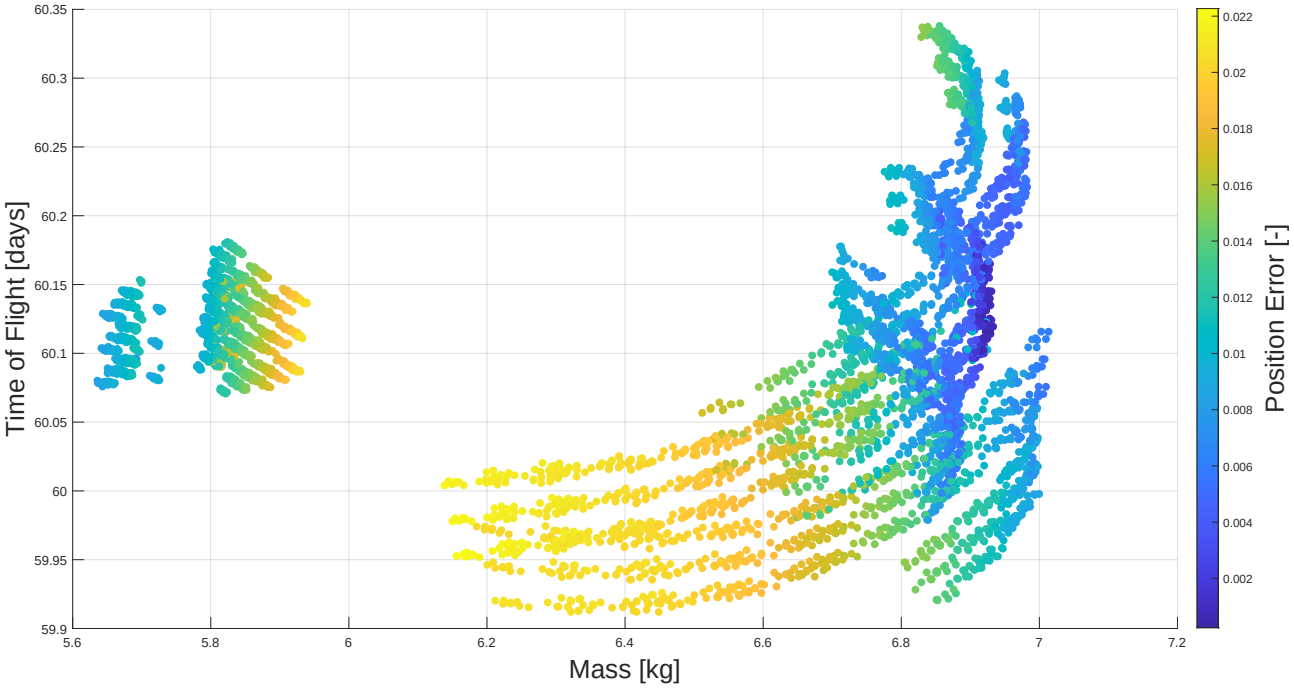

Figure 6.22: Comparison between propellant mass, TOF, and position error for the different experiments of the FFD around the nominal trajectory with a 1% variation (1st use case).

allel curves that can be identified in the figure may also be due to the search strategy, as the optimal control phase seeks the best connection point among a discrete set of states, removing homogeneity in the response.

Then, the position and velocity errors again for the 1% variation run are shown in Figure 6.23. Unfortunately, most of the trajectories do not satisfy the boundary constraints (indicated by the red lines). However, a fraction of them still do, and, since 1% is a significant perturbation, this suggests that the trajectories are not extremely sensitive and that the spacecraft can still remain close to the target trajectory when there is uncertainty in the parameters (which is convenient for a real transfer). This will be analyzed more deeply in Section 8.3. Nevertheless, this also shows that the optimizer will need to vary all parameters by less than 1% to find a single feasible solution. With such a narrow and non-smooth feasible domain, not only will the heuristic optimizer struggle to find feasible solutions, but it will probably prematurely converge to the found solution, without appropriately exploring the remainder of the design space. Hence, the constraint handling strategy is crucial in this problem, with a focus on diversification.

As usual, there are many more trajectories with a low velocity error than with a low error in position, with the velocity constraint being satisfied by a considerable fraction of the trajectories. In addition, a linear trend can be identified for a significant portion of the trajectories, which is expected since, to find the closest state, the same penalty is given to the position and velocity errors, i.e., w_{pos} and w_{vel} in Equation (5.18) are both equal to 1. Nevertheless, not all points follow this trend due to the highly non-linear problem. All linearity is lost in the 10% variation run, with the plot looking similar to Figure 6.18.

Regarding the ANOVA analyses of the local runs, the adjusted coefficients of determination of the position error, velocity error, TOF, and propellant mass are displayed Table 6.8. For a 1% variation, the accuracy of the fit for the propellant mass is already below 80%, making it the most difficult objective to estimate, followed by the TOF, the position error, and, lastly, the velocity error. For all objectives, the quality of the fit tends to decrease as a function of the variation percentage around the nominal solution. Even with a 1% variation, the modeling error is above 10% in all objectives, again

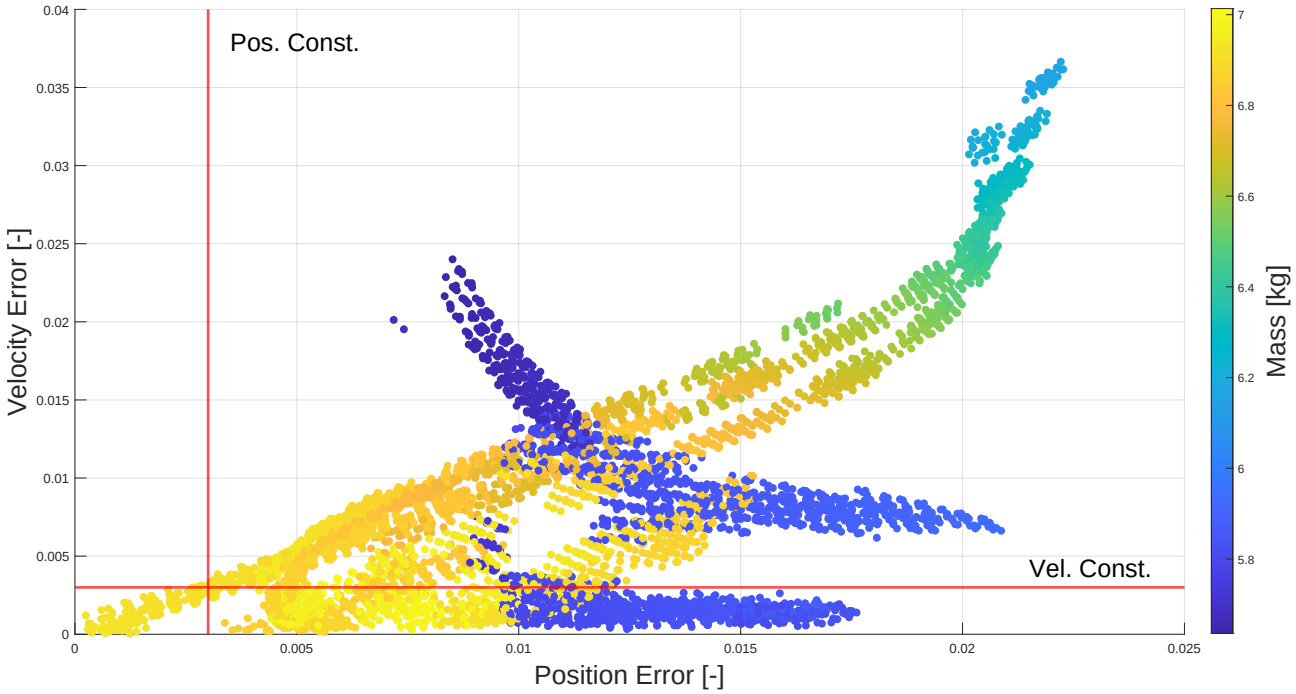

Figure 6.23: Comparison between position error, velocity error, and propellant mass for the different experiments of the FFD around the nominal trajectory with a 1% variation (1st use case). The red lines correspond to the boundary constraints of 10^{-3} in both position and velocity.

Table 6.8: R^2_{adj} values of the ANOVA analyses for the different objectives as a function of the run. The fraction of promising trajectories is added for each run as well.

Objective \ Run	**1%**	**10%**	**50%**	**General**
Position Error R^2_{adj} [-]	0.86	0.87	0.8	<0.5
Velocity Error R^2_{adj} [-]	0.89	0.87	0.84	<0.5
TOF R^2_{adj} [-]	0.85	0.63	0.63	<0.5
propellant Mass R^2_{adj} [-]	0.71	0.52	<0.5	<0.5
Fraction Promising Points [%]	88.2	5.1	3.4	0.4

proving the problem to be highly non-linear. As already stated, for the general runs, all R^2_{adj} values are below 0.5. A row was added to Table 6.8 incorporating the percentage of points that have both position and velocity errors below 20 times the constraint values. This value was chosen to denote points that could be relatively promising in terms of feasibility because only very few points satisfy the constraints. Even though almost 90% of trajectories satisfy this measure for the 1% variation run, the amount of promising points is very low for the rest of the runs. Once again, this shows the difficulties that the optimizer will encounter in obtaining a feasible transfer, being also the specific heuristic method an essential factor.

Focusing on the 1% variation run, Table 6.9 displays the ANOVA table for the position error[5], which has strong similarities to the one for the velocity error. Comparable percentages of contribution are obtained for the rest of the local runs, although not shown here. Nevertheless, these variance analyses correspond to runs that only explore a narrow region of the design space, meaning that their conclusions cannot be safely extrapolated to the general case. It should be noted that in Table 6.9, a

[5]Again, the subscript "0" is dropped when talking about the initial costate parameters for conciseness.

Table 6.9: ANOVA table of the position error corresponding to the 1% variation local run. The terms that were pooled due to insufficient statistical confidence are not included.

Term	**DOF** [-]	SS [Mm^2]	**Variance** [-]	**F-Ratio** [-]	SS' [Mm^2]	**Contrib.** [%]
τ_{orb_1}	2	1054	527	1044	1053	1.45
T_{man}	2	27982	13991	27731	27981	38.71
λ_{vx}	2	28	14	27	27	0.04
λ_{vz}	2	152	76	150	151	0.21
τ_{orb_2}	2	17	8	17	16	0.02
$\tau_{orb_1} \times T_{man}$	4	14209	3552	7041	14207	19.66
$\tau_{orb_1} \times \lambda_x$	4	14	4	7	12	0.02
$\tau_{orb_1} \times \lambda_z$	4	33	8	16	31	0.04
$\tau_{orb_1} \times \lambda_{vx}$	4	360	90	178	358	0.50
$\tau_{orb_1} \times \lambda_{vy}$	4	825	206	409	823	1.14
$\tau_{orb_1} \times \lambda_{vz}$	4	4400	1100	2180	4398	6.08
$\tau_{orb_1} \times \tau_{orb_2}$	4	8	2	4	6	0.01
$T_{man} \times \lambda_x$	4	30	8	15	28	0.04
$T_{man} \times \lambda_z$	4	47	12	23	45	0.06
$T_{man} \times \lambda_{vx}$	4	1543	386	765	1541	2.13
$T_{man} \times \lambda_{vy}$	4	1794	449	889	1792	2.48
$T_{man} \times \lambda_{vz}$	4	9625	2406	4769	9623	13.31
$\lambda_{vx} \times \lambda_{vy}$	4	9	2	4	7	0.01
$\lambda_{vx} \times \lambda_{vz}$	4	60	15	30	58	0.08
$\lambda_{vy} \times \lambda_{vz}$	4	181	45	90	179	0.25
$\lambda_{vz} \times \tau_{orb_2}$	4	10	3	5	8	0.01
Error	19608	9897	0.5	1.00	9936	13.75
Total	19682	72278	*NA*	*NA*	*NA*	**100.00**

good fraction of the estimated terms were pooled due to the statistical confidence of the results not being above 90% (their contribution is too low). As already explained, the statistical confidence is measured with the F-ratio, and for a confidence level of 90%, according to the tables provided by Roy (2010), an F-ratio value of 2.2 is required for the factors, and a value of 1.9 for the interactions.

Looking at the specific values, as expected, λ_m shows no contribution, since, from the explanation in Section 6.2, a much larger variation in λ_m is necessary for it to affect the trajectory at all. In addition, λ_y also seems to have no significant contribution to the variance. However, this can be explained by the fact that the nominal value for this parameter is one order of magnitude below the rest of the position and velocity costates (Table 6.5), meaning that a 1% variation in λ_y is negligible compared to the rest. The table suggests that the most important terms are: τ_{orb_1}, T_{man} (which determines on its own roughly 40% of the variance), and their interaction between themselves and with other parameters. As an example, the contribution of T_{man} is $\sim$14 times larger than λ_{vz}'s contribution[6]. This conclusion about the relevance of τ_{orb_1} and T_{man} in determining position and velocity errors applies consistently across all local runs (1%, 10%, and 50%). These are the most contributing factors for the general runs as well in both use cases, suggesting that either the starting point of the

[6]Recall that to quantify the relative contribution to the objectives instead of the variance, the square root of the ratio must be taken.

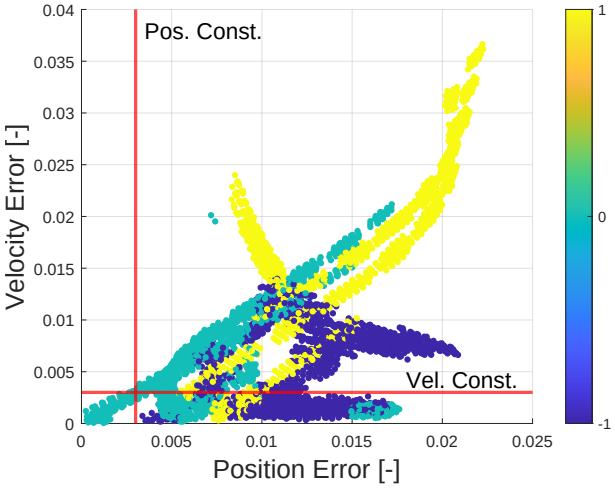

Figure 6.24: Position and velocity errors as a function of the level of T_{man} for the FFD around the nominal trajectory with a 1% variation (1st use case).

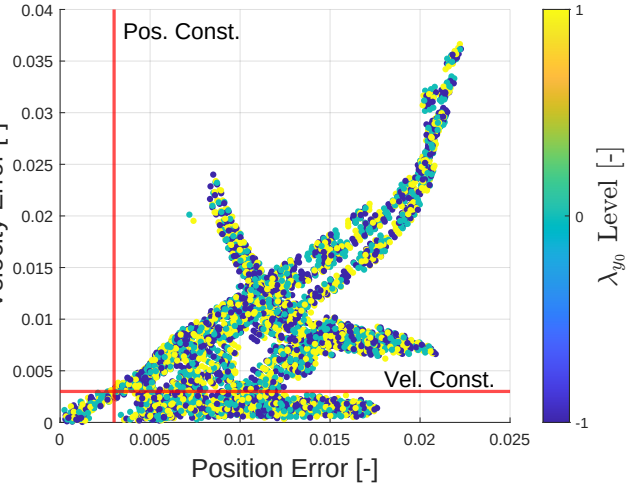

Figure 6.25: Position and velocity errors as a function of the level of λ_{y0} for the FFD around the nominal trajectory with a 1% variation (1st use case).

optimal control phase is paramount in the problem or that the main contribution of these parameters is of lower order (in terms of interactions) than the costates parameters, for instance. It makes sense that τ_{orb_2} is not as important since the complete target manifold and target orbit are taken as suitable end-points to find the closest state. Then, in the local regime, the velocity costates appear to be more influential than the position ones when considering the interactions as well. But this can be attributed to the fact that the nominal values are larger for the velocity costates (Table 6.5), being λ_{vz} the one with the largest value and therefore the largest contributor. Overall, if the design space is too large to find solutions, probably bounding T_{man} or τ_{orb_1} more restrictively would be the most helpful to focus the optimization search.

Now, having found the two most critical factors, it would make sense that the deviations from the linear trend in Figure 6.23 correspond to trajectories with these factors not in their nominal values. The effect of T_{man} on the position and velocity errors can be visualized in Figure 6.24, where it can be seen that the different clusters correspond to T_{man} being at different levels, with the best trajectories being obtained when the factor is at its nominal level. Further analysis led to the conclusion that all trajectories that satisfy the constraints require both T_{man} and τ_{orb_1} to be at their nominal levels. For comparison purposes, Figure 6.25 shows the position and velocity errors as a function of the level of λ_y. As this parameter is barely influential to the response, the distribution appears to be random, with the nominal level not necessarily leading to better trajectories.

To conclude, this design space exploration section provided a better understanding of the problem, denoting that its complexity and non-linearity are severe. As such, an appropriate constraint handling method, optimizer choice, and optimization tuning will be required to obtain a large enough number of feasible trajectories. Nevertheless, suitable ranges for the design variable values were identified, and the most influential parameters and constraints were distinguished.

6.4 Selection of Bounds and Constraint Handling Approach

With the conclusions drawn from the design space exploration, the optimization procedure can be chosen in a more informed way. In this section, the final optimization bounds are selected, as well as a preliminary constraint-handling strategy that will be employed to obtain the results. Due to time constraints, only the first use case from Section 6.1 will be optimized.

Table 6.10: Parameter lower and upper bounds used in the optimizations.

Parameter	Lower Bound	Upper Bound	Unit
τ_{orb_1}	0.0	1.0	-
T_{man}	12.0	22.0	days
$\boldsymbol{\lambda}_{r0}$	-0.1	0.1	-
$\boldsymbol{\lambda}_{v0}$	-0.1	0.1	-
λ_{m0}	-2.0	2.0	-
τ_{orb_2}	0.0	1.0	-

6.4.1 Optimization Bounds

Following the analysis carried out in Section 6.3 to choose adequate bounds for the design space, Table 6.10 displays the final bounds used for tuning the optimizations and to obtain the final results. They are very similar to the ones selected during the problem analysis displayed in Table 6.7. In these optimizations, the initial costate parameter bounds were all centered around zero. The allowable range for λ_{m0} was doubled to hopefully guide the optimizer towards trajectories that present longer coasting periods during the optimal control phase. Moreover, the only other variation is present in the bounds for T_{man}, setting a minimum TOF of 12 days in the unstable manifold phase. This was found to help the optimization, as T_{man} is a very influential optimizable parameter, and restricting the design space in this dimension leads the optimizer to focus its search. The first 12 days in the manifold phase are anyway very similar to the departure orbit in terms of states, showing barely any advantages in using the unstable manifold. With this configuration, all trajectories are "forced" to spend at least 12 days in the unstable manifold, meaning that the fastest transfers will not be found. Nonetheless, optimizations without manifolds will also be carried out to find these faster trajectories.

6.4.2 Constraint-Handling Strategy

Due to the highly demanding boundary constraints that this problem presents, the choice of the constraint handling strategy is paramount to finding a wide variety of feasible solutions to maximize the optimization in terms of propellant mass and flight time. An overview of the main constraint-handling techniques used in heuristic optimization over the years is provided by Mezura Montes and Coello Coello (2011). Due to the time limitations of this research, off-the-shelf optimizers will be employed, namely MIDACO and the ones implemented in Pagmo (with all MO optimizers being unconstrained). Therefore, some promising constraint-handling strategies cannot be used, as their implementation would require the modification of the heuristic philosophy of the algorithms. Moreover, as the idea is to compare the performance of the different optimizers, it is desirable to use a constraint-handling strategy that is not specific to a certain heuristic algorithm. Namely, *stochastic ranking* (which was the approach chosen by Lee et al. (2005) in combination with indirect methods to solve the MO problem in the CR3BP), *feasibility rules*, ϵ*-constrained method*, and *special operators* were not considered for the given problem. For the same reason, hybrid optimization is not employed, although some authors have made use of it in combination with indirect methods, such as Coverstone-Carroll et al. (2000) and Sentinella and Casalino (2006). Nonetheless, the fact that the OGT solutions are meant to serve as initial guesses for direct collocation is already a form of hybrid optimization. Furthermore, due to the narrow and highly non-convex feasible domain, *decoders* to map the feasible domain and *repair algorithms* to turn infeasible solutions into feasible ones are not sensible options. Moreover, for highly constrained problems, the *augmented Lagrangian method*, as described by Mahdavi and Shiri (2015) and Sedlaczek and Eberhard (2006), has proven to be successful. However, this method is complicated, is meant for single-objective optimization, and requires several optimization runs to tune the parameters, thus being undesirable for the OGT.

For the actual constraint-handling strategy trade-off, three families of methods were compared: *penalty functions*, *separation of objective and constraints*, and *multi-objective* concepts. These strategies were compared using MIDACO with default settings as a preliminary optimizer. Penalty methods add a function of the constraint violations to the objective(s), thus penalizing unfeasible trajectories. A common approach, employed with indirect methods by Pergola et al. (2009), Lee et al. (2005), and Pontani and Conway (2010), is for the penalty function to simply be a weighted sum of the constraint violations. As described by Smith et al. (1997), penalty functions are often also a weighted sum of the square of the constraint violations. In addition, Coello Coello (2002) shows that penalty functions can take many different and more complex shapes, with their weights being static or dynamic (changing throughout the optimization). MIDACO has the more complex *Oracle Penalty function* built-in for constraint handling (Schlüter and Gerdts, 2009). After optimizing with numerous penalty function configurations (and combinations of them), including the Oracle penalty, the main conclusion is that with penalty methods, due to the highly restrictive constraints, the optimizer may or may not converge to a feasible solution depending on the tuning parameters, seed, and function shape. The Oracle penalty function appeared to be the most robust, however, the optimizer then failed to find sufficiently different solutions that also satisfied the constraints[7], prematurely converging to suboptimal solutions, and not providing a wide Pareto front. These observations are common disadvantages of penalty functions, according to Coello Coello (2002). Moreover, due to the larger number of tunable parameters and the case-dependency of their values[8], this constraint-handling approach may not be the best for the OGT, since, ideally, it should work for different use-cases without excessive modifications. For the aforementioned reasons, penalty functions are discarded.

To help the optimization converge to a feasible solution, separation of objectives and constraints can be used to ensure that an infeasible transfer is never considered better than a feasible one. The idea is to map the performance of feasible solutions into the interval (-∞, 1), and of infeasible solutions into [1, ∞) (Coello Coello, 2002). Hence, following the nomenclature from Section 3.3 and being the objective function $f(\boldsymbol{p})$ scaled to the interval (-∞, 1), a trajectory would be ranked as:

$$\text{fitness}(\boldsymbol{p}) = \begin{cases} f(\boldsymbol{p}) & \text{if feasible} \\ 1 + a\left(\sum_{i=0}^{n_{const}} w_i \max(0, h_i(\boldsymbol{p}))\right) & \text{otherwise} \end{cases} \tag{6.17}$$

where a and w_i are constant weights, h the specific inequality constraint function, and n_{const} is the number of constraints. Nevertheless, even though this method converges to feasible trajectories more often, the loss in diversity is significant since the objective value is not considered if the trajectory is unfeasible (Mezura Montes and Coello Coello, 2011). Moreover, with this method, infeasible trajectories that may satisfy the constraints with further optimization are generally discarded if a feasible trajectory has already been found.

Lastly, multi-objective constraint-handling methods include the constraints as additional objectives to be minimized by the optimizer. This method is convenient for this research, as MO optimizers are used anyway to minimize the propellant mass and TOF. Then, the solution Pareto-front will be the section of the higher-dimensional Pareto-front for which the constraint violations are zero. The tuning parameters in this case are the relative weighting of the objectives, being hence more applicable to a general case. This method promotes diversity as it favors trajectories that minimize the mass and TOF, even if the constraint violations are large. Then, the optimizer will in principle try to minimize the constraint violations for these promising trajectories. Nevertheless, a balance ought to be found, as these low-cost trajectories may be too unrealistic to ever satisfy all constraints. As such,

[7] Many solutions are obtained, but they are extremely similar to each other.

[8] This is not the case for the Oracle penalty function, as it is self-tuning.

separating the three constraints (position, velocity, and Jacobi constant) as three extra objectives led the optimizer to dedicate excessive efforts to exploring these unrealistic areas. The best configuration found for MIDACO is to perform an optimization with two objectives: the propellant mass and the weighted sum of the constraint violations. However, for MIDACO to find trajectories, the algorithm needed to solely focus on the constraints objective (through the built-in *balance* parameter). Hence, even though with this strategy MIDACO always finds several sufficiently different solutions (unlike the other methods, which would only find solutions extremely close to the first one that was found), the diversity problem still persists. Nonetheless, to move forward and proceed to the optimization tuning in Chapter 7, this last constraint-handling strategy is utilized.

In conclusion, this chapter found the Adams-Moulton integrator with seven interpolating points to be the best-performing option for the given problem. The design space exploration highlighted the highly non-linear nature of the problem and the restrictive nature of the constraints. Consequently, having selected appropriate optimization bounds and a preliminary constraint-handling strategy based on treating the constraints as a separate minimization objective, optimization tuning will be crucial to obtain a diverse set of promising trajectories for the proposed use case.

Use Case Optimization and Results 7

In this chapter, the results of the OGT optimizations with the proposed approach will be presented. As already mentioned, only the first use case of Section 6.1 will be optimized in this research. The optimizer and optimizer settings are chosen in Section 7.1, making a distinction between the results obtained before and after relaxing the constraints. Only then, the final Pareto front is presented in Section 7.2. Lastly, Section 7.3 assesses the effect of manifolds by running optimizations without the use of manifold theory.

7.1 Optimization Tuning

This section is divided between the decisions made with the initial (tight) constraints and the ones determined after relaxing the constraints. Nevertheless, all tuning decisions were re-assessed after the relaxation to ensure their applicability to the final product.

7.1.1 Before Constraint Relaxation

Optimizer Selection

After selecting a preliminary constraint-handling technique in Section 6.4, the performances of different optimizers were compared. As explained in Subsection 3.3.4, the optimizers compared were: MIDACO, MOEA/D, MHACO, NSGA2, and NSPSO (all with default settings). The last four belong to the Pagmo library. The advantage of this library over MIDACO is the fact that it offers the possibility of using several optimization islands (populations) running in parallel that can exchange the most promising trajectories. This analysis showed that the only two optimizers that would consistently find trajectories, even with several islands, were MIDACO (based on ant-colony optimization) and MOEA/D (based on differential evolution), both of which use the decomposition strategy to handle the MO optimization. For MOEA/D to robustly find trajectories, a weight of 10^5 is applied to the (non-dimensional) constraint objective compared to the mass and TOF objectives, both in dimensional quantities. This value is considered to apply to a general case due to the vast difference relative to the other two objectives. As already mentioned in Section 6.4, MIDACO requires the balance parameter (which tunes the focus of the optimizer on the different objectives) to completely concentrate on minimizing the constraints objective.

Special Settings

Following further analysis, several arrangements proved to improve the overall quality of the solutions. Firstly, the DOE showed that with the selected search strategy, a fraction of the trajectories would have zero propellant mass and an optimal control phase of zero TOF due to the initial state of this phase[1] being the closest state to the target phases. When including the Jacobi error limit of $3{\cdot}10^{-2}$ to compute the closest state at a given point in time (Subsection 5.2.3), the number of zero-propellant mass trajectories increased significantly. For the constraints objective this is not a problem as the state error in these trajectories is rather large. However, to prevent MOEA/D from favoring these extreme trajectories, for the mass and TOF objectives (although, for now, only the mass is being used) they are penalized by setting the objective values to twice the maximum allowable propellant mass or TOF.

To further promote diversity, punishing feasible points that have already been found seems to help. The idea is to store every feasible trajectory for the final results, but in the optimization, a non-zero constraint violation will be set for these trajectories to force the optimizer to search somewhere

[1] Recall that this is the end-point of the unstable manifold phase.

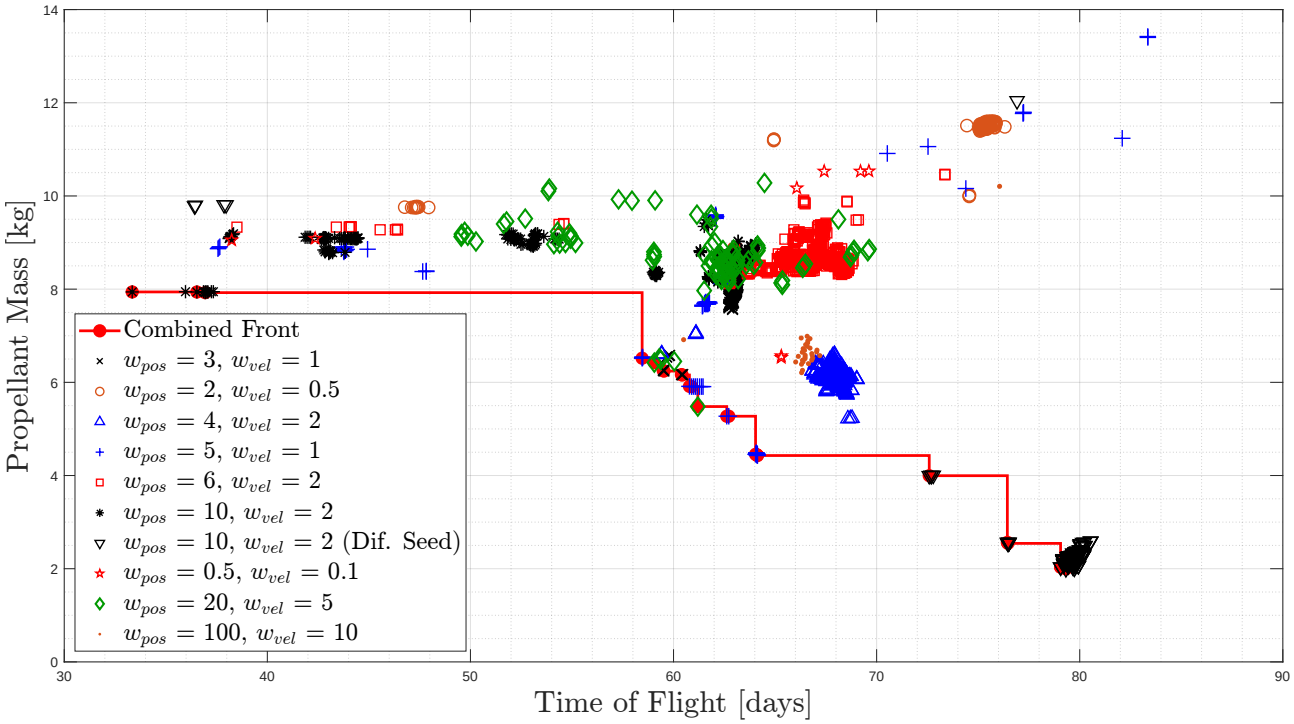

Figure 7.1: Trajectories found by MIDACO in several runs with differing constraint weights and their combined Pareto front.

else. In addition, feasible trajectories that do not have a sufficiently different performance (0.5% in any of the objectives) also receive this penalty. Following the analogy of the ant colony optimization, finding a feasible trajectory would be equivalent to locating food, but at some point, the food in that area runs out, and the ants are urged to explore different regions. This practice is predicted to only be beneficial with highly constrained problems like this one, where the feasible domain is made up of strict and narrow regions of feasibility. The chosen penalty value is 0.01 for the objective of the constraints in non-dimensional units. This number is larger than the surrounding infeasible trajectories to compel the optimizer to leave the area, but low enough to not completely disregard the promising parameters of the trajectory. When using MOEA/D, this value is scaled by 10^5, according to the objective weighting.

Constraint Weights

As all constraints are grouped within a single objective, the weights of each of them within the objective ought to be tuned[2]. For the constraint weighting, the same structure as Equation (5.18) will be used. It is sensible that the search strategy weighs the position, velocity, and Jacobi constant errors equally as much as the optimizer weighs the constraint violations (since the latter is the function to be minimized). As a result, the weights for the position and velocity constraints will be equal to w_{pos} and w_{vel} from Equation (5.18), respectively. As such, the constraints objective function, f_{const}, shall be formulated as:

$$f_{const}(\boldsymbol{p}) = w_{pos} \max(0, h_{pos}(\boldsymbol{p})) + w_{vel} \max(0, h_{vel}(\boldsymbol{p})) + \max(0, h_c(\boldsymbol{p})) \tag{7.1}$$

where w_{pos} and w_{vel} are positive constants.

Figure 7.1 displays the performance of MIDACO for differing values of w_{pos} and w_{vel}, with the position weight always being larger than the velocity one to prioritize the position errors (due to the

[2]If they were separated into three objectives, the relative objective weighting would be tuned instead.

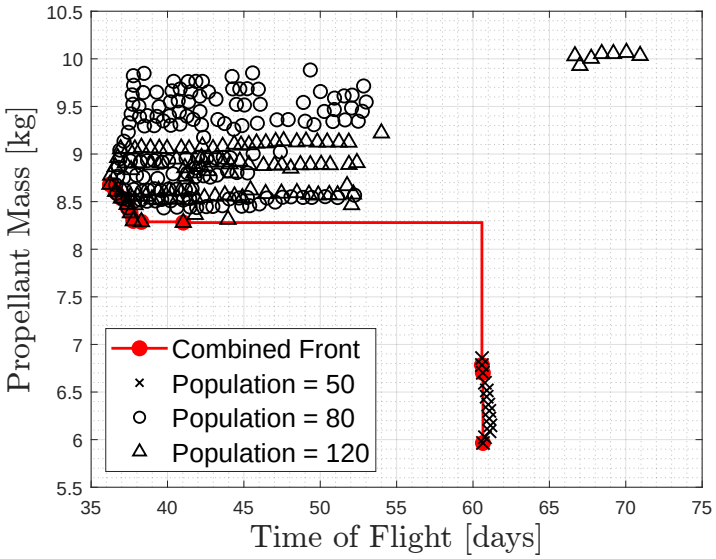

Figure 7.2: Trajectories found by MOEA/D in several runs with differing population sizes and their combined Pareto front.

conclusions from Chapter 6). In this figure, all feasible trajectories are plotted according to their objective values: propellant mass and TOF. From these preliminary optimizations, it can be seen that MIDACO does not find many feasible trajectories with a single run. However, the quality of all of them is outstanding, considering that the transfer can be completed generally with 1-2% of the total spacecraft mass (500 kg). In addition, the combined Pareto front[3] spans a considerably large range of propellant mass and TOF values, although it is scarcely populated and required numerous runs to be generated. Nevertheless, the proposed approach already offers general advantages over conventional methods, as it provides more than one competitive trajectory to choose from.

Regarding the performance with different weights, it is difficult to draw conclusions as all behave similarly in terms of the number of feasible points. In terms of the quality of the solutions, weighing the Jacobi constant constraints more than the other (by having w_{pos} and w_{vel} being smaller than 1) led to slightly poorer results. This is also observed when weighing the Jacobi constraint too little. Moreover, the same settings with distinct pseudo-random seeds lead to very different outcomes, as evident from the two runs with w_{pos} = 10 and w_{vel} = 1, which found trajectories in the opposite extremes of the front. Therefore, the performance of the optimization, at least for MIDACO, appears to be more dependent on the seed than on the constraint weights, as long as they are within a sensible range. For the rest of the tuning and the final optimizations, w_{pos} = 5 and w_{vel} = 1 are chosen.

This analysis showed that MIDACO struggles to preserve diversity and find a large variety of feasible solutions. Hence, MOEA/D may be a more promising choice due to the possibility of using several islands in parallel that exchange information thanks to the Pagmo library. In the next paragraphs, a suitable population size is selected for MOEA/D, and then different optimization architectures are explored to obtain a more diverse set of solutions.

Population per Island

MOEA/D was run with four different population sizes per island: 50, 80, 120, and 200 individuals. The rest of the settings were left as default, taken from the original authors' recommendations (H.

[3]To represent the Pareto fronts, it was decided to use a staircase plot to take the worst-case scenario. This is probably not the case in the real, physical solution, but as trajectories in between the points have not been found, it would be misleading to approximate the front with a smoother shape.

Li and Q. Zhang, 2008). In addition, three islands were used in each optimization with migrations every five generations. As suggested by Ahlborn (2023), only the best member from each island migrates to the others to preserve diversity. Thus, the *selection rate* parameter is set to one member and the *replacement rate* (the probability of accepting an incoming migration) is set to 100% for all optimizations (including the final results).

The results can be visualized in Figure 7.2. The optimization with a population of 200 individuals is not displayed because it did not find any feasible transfer within the allocated number of generations ($\sim$2500), meaning that, although it may eventually obtain results, the required convergence time is too large for a useful tool. It is evident from the figure that, even though the optimization with 50 individuals obtains trajectories of lower propellant mass, the number of feasible trajectories obtained is considerably low compared to the other two configurations. The optimizations with 80 and 120 individuals performed similarly, each of them finding more than 100 sufficiently different feasible trajectories, spanning more than 20 days. A final population size of 120 is selected, as all members forming the Pareto-front belong to this run (disregarding the ones obtained by the run with 50 individuals). Moreover, this run also found several transfers of very different performances (10 kg and 70 days), denoting a more diverse search.

Optimization Architecture

The previous analysis of population sizes demonstrated the better capabilities of MOEA/D (with several islands) in exploring the feasible domain compared to MIDACO, which found more isolated points. Furthermore, looking at the time evolution of the MOEA/D optimizations, the worst-performing points are the first to be found, progressively finding better and better solutions over the generations. This observation is common to all MOEA/D runs presented in this research, proving that the method actually optimizes for propellant mass even with very tight constraints (TOF is not considered in this case) and it is not just randomly finding feasible points.

The goal now is to identify an appropriate number of islands and migrations to maximize performance. How the islands are connected is also of importance, the so-called *archipelago structure*. One can decide to have them unconnected (essentially having no migrations) partially connected, or fully connected. To prevent all islands from prematurely converging to the same set of trajectories, the fully connected configuration is avoided. Then, Pagmo offers a way of connecting them partially by placing them in a *ring* configuration, in which the best members can only migrate to the two adjacent islands, providing a balance between diversity preservation and information exchange to obtain better solutions and delay convergence.

Figure 7.3 shows the results of several runs with differing numbers of islands and migration rates using the ring archipelago structure[4]. "Unconnected" refers to no migrations, "many migrations" corresponds to having migrations every five generations, and "few migrations" to every 20 generations. It can be seen that the run with 20 islands and many migrations performed incredibly similarly to the run with 10 islands and few migrations (but requiring roughly twice as many function evaluations). This makes intuitive sense, since increasing the number of islands can have a similar diversity-preserving effect as decreasing the number of migrations when disposed in a ring configuration. As expected, these runs found the most diverse set of feasible points, however, they are the worst performing in terms of propellant mass and Pareto-optimality overall. The least diverse run in terms of variety of performance in the obtained solutions is the one with five unconnected islands, followed by the one with four islands. This can be attributed to the fact that migrations prevent the islands from prematurely converging by introducing new promising parameters (but too many will remove the diversity

[4]The density of the found feasible points is too high for some of the runs because this correction was not yet implemented.

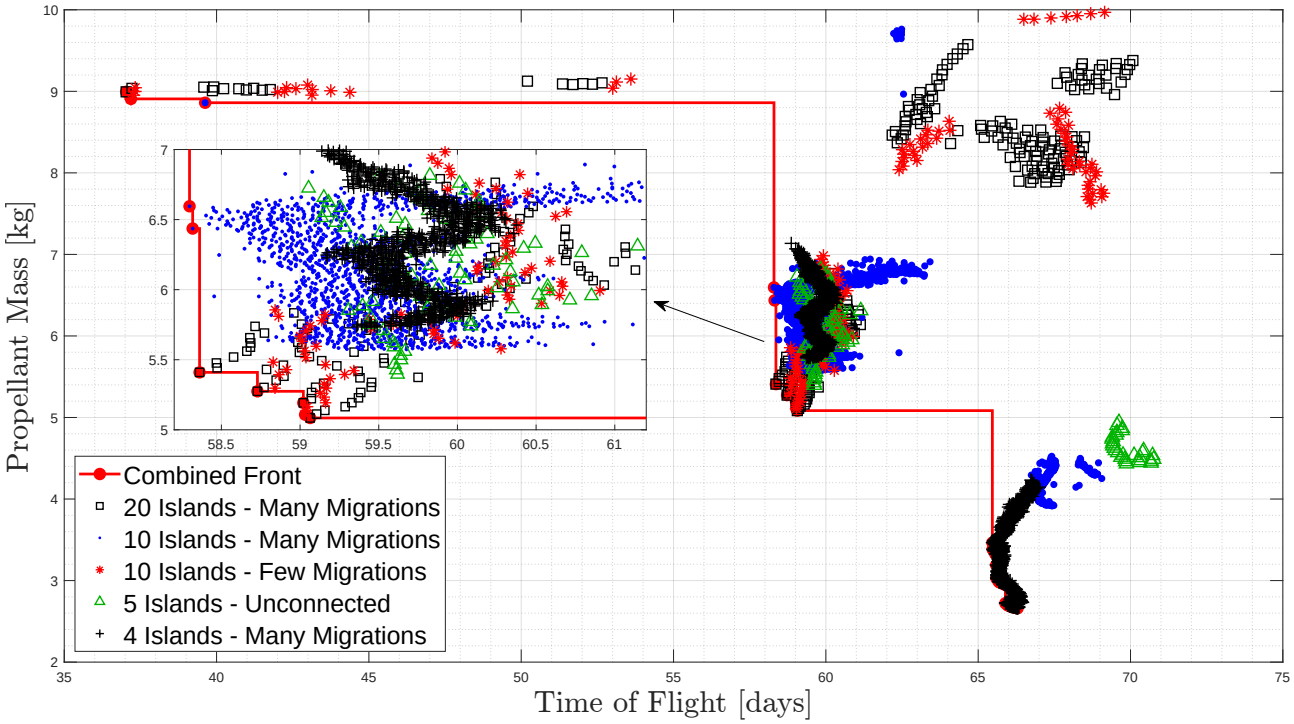

Figure 7.3: Trajectories found by MOEA/D in several runs with differing architectures and their combined Pareto front.

in the populations). In addition, some of the unconnected islands may never find a feasible point. The run with four islands also obtained the trajectories with the lowest propellant mass, since with few islands and a lot of migrations, the exploration capabilities within a promising region are enhanced. Lastly, a good compromise between the diversity of solutions and their quality is attained by using ten islands and migrations every five generations, thus being the chosen configuration.

A more exhaustive tuning could be carried out, also including the selection and replacement rates, however, due to time constraints this was not possible for this research. Moreover, the hardware used for this optimization only has 12 logical processors, making larger numbers of islands inefficient in terms of CPU time (only 12 trajectories can be computed in parallel). For more powerful hardware, the use of more islands with adjusted topology, migrations, selection rates, and replacement rates could be beneficial.

The Pareto-fronts obtained by MIDACO and MOEA/D with all runs are compared in Figure 7.4. With fewer runs, the trajectories found by MOEA/D outperform the ones from MIDACO except for the faster trajectories. MOEA/D also showed better search capabilities and smoother Pareto fronts in the explored regions. As MIDACO can find trajectories of very different performance, in future research it may be interesting to feed populations optimized by MIDACO to MOEA/D for it to thoroughly explore these different regions. Overall, even though the obtained trajectories are of outstanding quality, currently both optimizers provide underpopulated fronts, probably far from the optimal solution and not ideal for the mission designer to choose from. This is attributed to the highly restrictive constraints, which explains the relaxation analysis carried out in Subsection 8.2.2.

7.1.2 After Constraint Relaxation

By increasing the allowable position and velocity discontinuities by a factor of three, MOEA/D's performance improved significantly, consistently finding more than a thousand feasible points instead of barely a hundred. This can be explained by the fact that, with tighter constraints, most of the individuals in a population would need to be very close to a certain feasible point (thus all having very

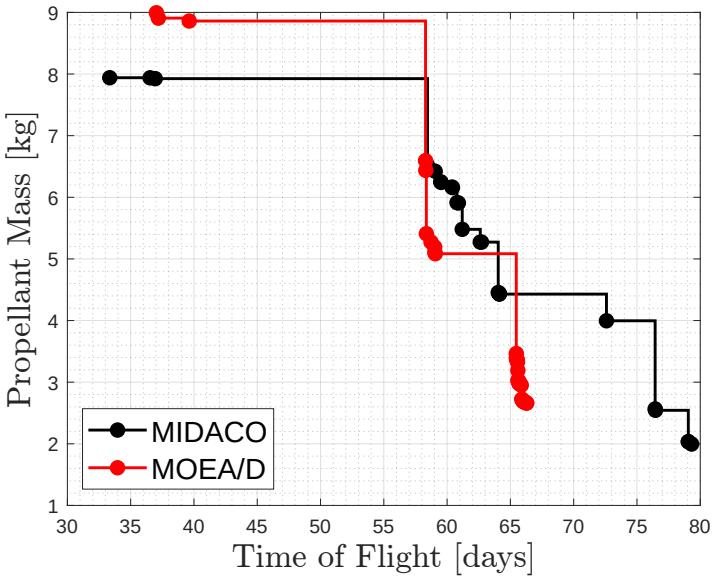

Figure 7.4: Comparison between the Pareto fronts obtained by MIDACO and MOEA/D.

similar parameters) to modify the design variables by a small enough amount to eventually satisfy the constraints. Therefore, most of the diversity is lost once a feasible trajectory is found. By relaxing the constraints, the feasible domain is enlarged such that solutions can be obtained without excessive required accuracy in the design variables, being thus easier for the optimizer to "unlock" new areas of feasibility within a single island and not prematurely discard promising parameter combinations. Nevertheless, this was not the case for MIDACO, which, even though it increased the number of trajectories discovered, it would still get "stuck" around the found solutions, which may be far from optimal. Furthermore, it still displayed a high dependency on the pseudo-random seed. These phenomena demonstrate the benefits of using several connected islands running in parallel when dealing with highly constrained problems.

Weighting of Objectives

So far, the optimization strategy was to only minimize the propellant mass and the constraint violations, completely neglecting the TOF. This was found to work best for MIDACO but was never put to assessment using MOEA/D. With relaxed constraints, other regions of the Pareto-front can in principle be explored better by varying the weighting between the propellant mass and the TOF (having the weighted sum of constraints as a third objective). The outcome of this exercise is plotted in Figure 7.5. The figure shows the performance of all feasible trajectories found with different objective weightings: focusing solely on the TOF, only on the propellant mass (two runs with differing seeds were carried out), on both, or 30 times more on the propellant mass than on the TOF. The value of 30 is chosen as the dimensional TOF values are one order of magnitude larger than the dimensional propellant mass values. As expected, the relative importance of each objective makes the optimizer explore different regions of the Pareto front. With more runs, scarcely populated regions would in principle be filled. When comparing Figure 7.5 to Figure 7.3, one can directly observe the improvement achieved by the relaxation of the constraints. As already mentioned, for all these runs, the optimizer first found the worst-performing trajectories, and it gradually pushed the Pareto front towards the lower left corner through the obtainment of better-performing transfers. Nevertheless, most of the feasible transfers found would be very promising solutions anyway.

The best run in terms of propellant mass was obtained by solely focusing on the propellant mass, as could be predicted. Furthermore, the effect of the seed, although non-negligible, is not severe in

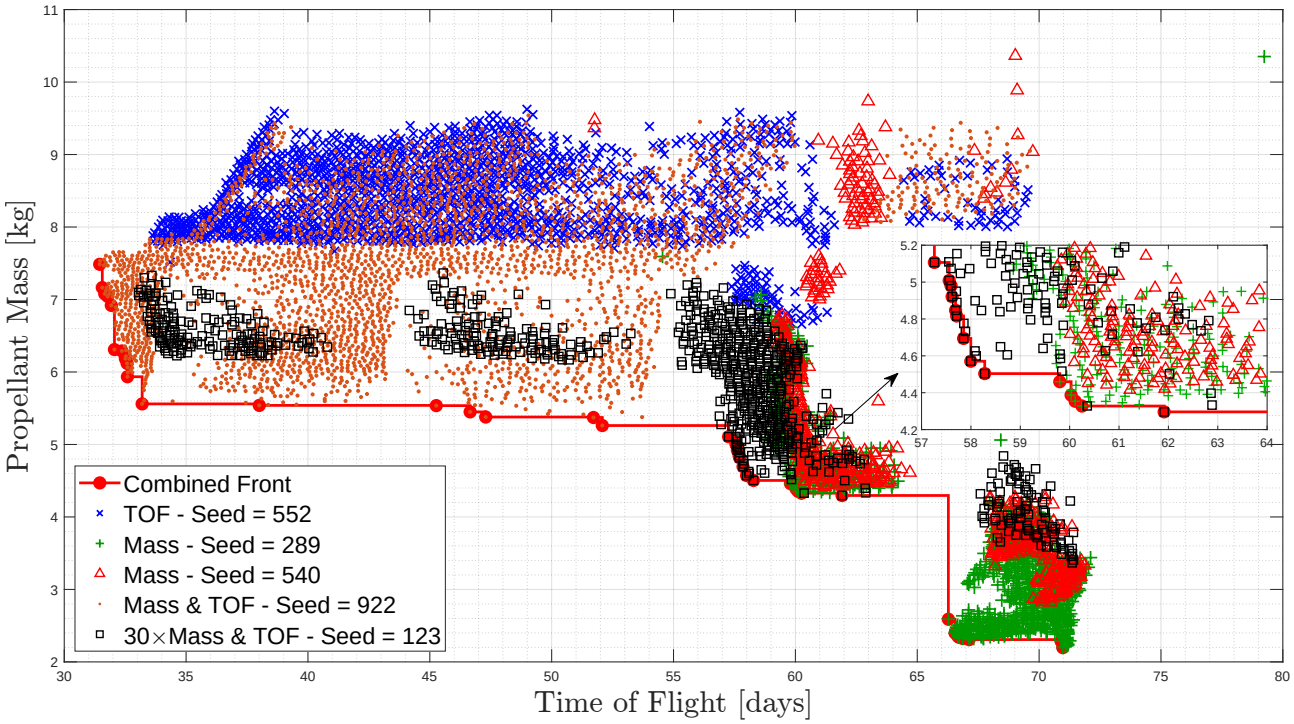

Figure 7.5: Trajectories found by MOEA/D in several runs with differing objective weighting and their combined Pareto front.

terms of performance or the explored regions of the front. However, the lowest times of flight were attained by the run which gave equal importance to both objectives, instead of the one that disregarded the propellant mass. This might be due to the pseudo-random seed or because also focusing on the propellant mass led to finding new trajectories, thus discovering more promising regions of the feasible domain. The most balanced run corresponds to the last one, which, although it only contributed to a specific portion of the combined front, the solutions found span almost the full Pareto front range, exploring all the main regions. As such, with a single run, one would have obtained a front ranging from 33 to 72 days, and from 3 to 7 kg. Overall, this analysis proves that as a mission designer, one can modify the objective balance parameter to obtain optimized solutions with a performance that would suit the mission better.

Effect of Penalizing Already Found Trajectories

The aforementioned effect of penalizing trajectories that have already been found or that have a performance within 0.5% variation relative to these can be visualized in Figure 7.6. Here, two runs were carried out with the same settings, including the seed, but with the punishment mechanism turned on or off. The run with punishment corresponds to the one in which the mass and the TOF are weighted equally, displayed in Figure 7.5. In both runs, the explored regions of the front are the same, starting with an almost identical distribution of solutions. However, over time the run with punishment managed to get the mass down to 5.5 kg, whereas the unpunished one got "stuck" around 6.7 kg. Nonetheless, the unpunished run obtained the fastest trajectories, since the information of the found trajectories can be exploited to obtain more promising runs. The fact that this only occurred with a small fraction of the combined front suggests that an appropriate balance between keeping these trajectories and forcing the optimizer to search somewhere else was found (by penalizing the trajectories by a pertinent amount). To conclude, for the given highly constrained problem, penalizing already-found trajectories seems to enhance the exploration capabilities of the optimization.

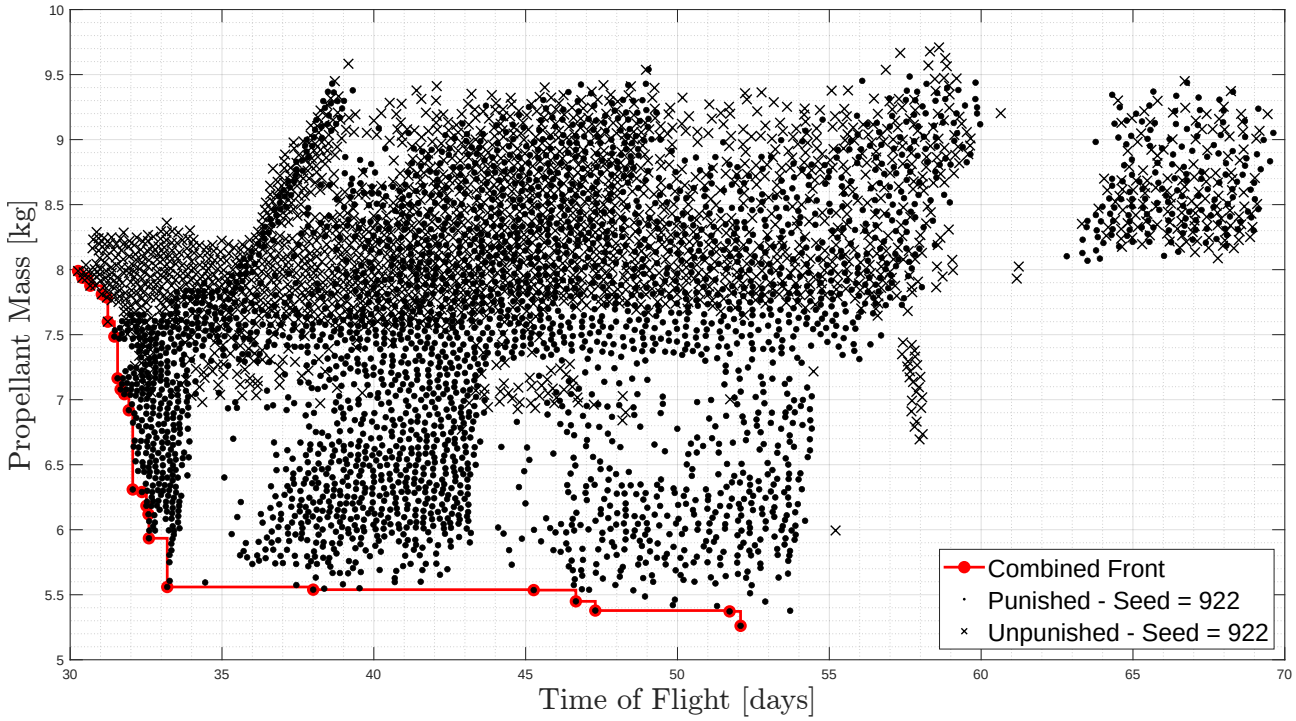

Figure 7.6: Trajectories found by MOEA/D when punishment on the already discovered solutions is applied compared to when it is not. The combined Pareto front of both runs is also included.

Table 7.1: Optimizer settings used to obtain the final Pareto-front.

Archipelago Parameter	**Value**	**MOEA/D Parameter**	**Value**
Number of islands	10	Weight generation	Grid
Population per island	120	Decomposition	Tchebycheff
Migrations	Every 5 gen	Neighbours	20
Topology	Ring	Crossover parameter	1
Replacement rate	1 member	F-parameter	0.5
Selection rate	100%	Distribution index	20
		Realb-parameter	0.9
		Reinsertion limit	2
		Preserve diversity	True

7.2 Final Results

The results presented in this section simply combine all the previously presented runs. No extra runs were carried out for the sake of transparency, although by changing the balance parameter a better and more populated Pareto front would be obtained. The final optimization settings are displayed in Table 7.1. An explanation of the MOEA/D settings[5] is provided by H. Li and Q. Zhang (2008). The design variable bounds correspond to the ones displayed in Subsection 6.4.1. As can be observed in Section 7.1, to obtain most of the Pareto-optimal solutions, the only parameters that were varied over the different runs were the seed and the objective balance parameter (user-definable). These settings are of course catered toward the given use case, although a more exhaustive tuning of the MOEA/D parameters should improve the results to a certain extent. For a different use case, the user can modify these parameters as they see fit.

[5]URL: https://esa.github.io/pagmo2/ [Accessed: 10/07/2024]

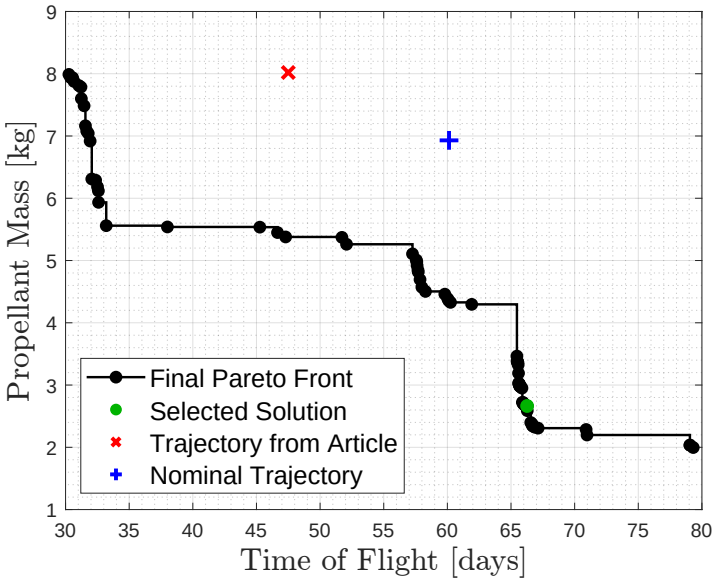

Figure 7.7: Final Pareto front of the chosen use case. The performances of the nominal trajectory and the one found by Pritchett et al. (2017) are included as well. In green, the selected Pareto-optimal solution is highlighted.

7.2.1 Final Pareto Front

The complete Pareto front using manifold theory is presented in Figure 7.7. This front includes 79 feasible transfers from which the mission designer can choose. The diversity of options is large, as the front spans 50 days and 6 kg of propellant mass. Although the front was generated for the given use case, it can be scaled up or down as long as the initial and final orbits are the same, and that the I_{sp} and thrust acceleration are also equal. For instance, a satellite of 50 kg with 10^{-2} N of thrust (instead of the current 500 kg and 0.1 N), would lead to the same Pareto front (and the same trajectory parameters), but with the mass scaled down by a factor of ten. It should be noted that all trajectories except for three were obtained with MOEA/D.

In the figure, the performance of the nominal trajectory from Section 6.2 is displayed, manifesting how much the optimization quality was improved by the tuning process. The optimized solution obtained by Pritchett et al. (2017) for a similar use case is also included. Even though this transfer is already promising, it is greatly outperformed by the obtained Pareto front. For instance, with 47.5 days available, a trajectory was found that requires 5.4 kg of propellant instead of 8.02 kg. On the other hand, with 8.02 kg of propellant consumption, the transfer could be completed in 30 days. As such, a roughly 30% reduction can be obtained in any of the objectives. This demonstrates the advantages of the proposed indirect method in combination with MO heuristic optimization compared to direct collocation with a rough initial guess: not only is a variety of solutions obtained instead of a single one, but their performance can be superior.

Regarding the speed, the optimizations with the provided architecture required roughly $2.5{\cdot}10^6$ trajectory evaluations to converge (no more feasible points are found). Using an 8^{th} generation CPU of six dual-cores, a speed of 2.2GHz, and 16 GB of RAM, each evaluation takes $\sim$0.28 seconds. Thanks to the ten islands running in parallel, the average time per evaluation is around 0.05 seconds, entailing optimization convergence in 35 hours. However, as shown throughout the tuning process, several runs were carried out with different values for the objective balance parameter, with 35 hours corresponding to a single run.

Although the results are incredibly promising from a mission design standpoint, the shape of the Pareto front does not look optimal, at least as expected for a physical problem. There are unpopulated "jumps" in the front, denoting regions that could be explored better. Moreover, looking at the design

Table 7.2: Design variable values and performance of the selected Pareto-optimal trajectory.

Parameter	**Value**	**Unit**
τ_{orb_1}	0.348125	-
T_{man}	15.348539	days
λ_x	0.003460	-
λ_y	-0.002889	-
λ_z	-0.003816	-
λ_{vx}	$6.847047 \cdot 10^{-4}$	-
λ_{vy}	$-4.298818 \cdot 10^{-4}$	-
λ_{vz}	0.001008	-
λ_m	0.930790	-
τ_{orb_2}	0.512709	-
m_p	2.66	kg
TOF	66.29	days
TOF (Opt. Control Phase)	19.82	days
T_{man_2}	31.13	days
Position Error	383.13	km
Velocity Error	0.203	m/s
C Error	$9.576 \cdot 10^{-4}$	-

variable values of the obtained solutions, for a single run, the trajectories around a given region share strong similarities between their design variables. For a different run, completely different trajectories may be obtained (with maybe a similar performance), showing that there is still room for improvement in the exploration capability and diversity of solutions. All this is again attributed to the very demanding boundary conditions. In addition, the design variables of most of the obtained solutions have the same order of magnitude as the bounds, so urging the optimizer to explore lower orders of magnitude may lead to a different set of trajectories.

Nevertheless, the shape of the Pareto front resembles that of the one obtained by Lee et al. (2005) with a similar approach in a transfer between DROs. In addition, the large "jump" in TOF from 33 to 56 days with roughly the same propellant mass can be to some extent explained by the fact that all trajectories below 40 days directly connect to the target orbit, without the use of the stable manifold phase. As the rest of the transfers employ both manifolds (which need a propagation time of at least 15 days to substantially separate from the halo and thus be advantageous), considerably larger TOFs are needed to reduce the required propellant mass. As such, when removing both manifold phases, it is expected to obtain even faster trajectories and extend the Pareto front of Figure 7.7 toward the left.

7.2.2 Selected Pareto-Optimal Solution

The idea now is to choose a solution from the final Pareto front and correct its discontinuities between the phases with direct collocation methods by means of the ASTOS software. The design variable values of the selected Pareto-optimal solution (highlighted in Figure 7.7) and its performance are outlined in Table 7.3. The required propellant mass for this trajectory is minuscule, roughly 0.5% of the spacecraft mass, so several coasting periods are present. For reference, 2.66 kg would correspond to a ΔV of 104.7 m/s. The position, velocity, and Jacobi constant errors are far from the constraint limits, making it easier for the ASTOS refinement. Furthermore, the initial position and velocity costate parameters of this trajectory are at least one order of magnitude lower than the provided bounds. The transfer requires 66.29 days, from which 15.35 days take place in the unstable manifold, 19.82 days in the optimal control phase, and 31.13 days in the stable manifold.

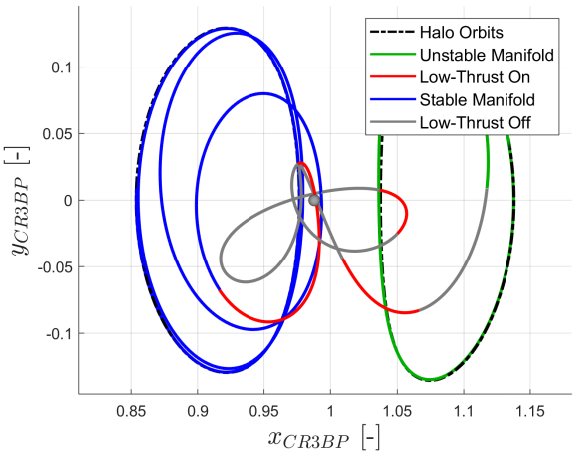

Figure 7.8: Top view of the chosen Pareto-optimal trajectory.

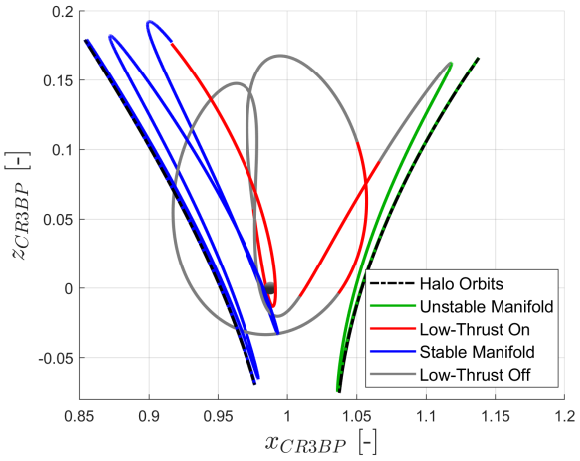

Figure 7.9: Side view of the chosen Pareto-optimal trajectory.

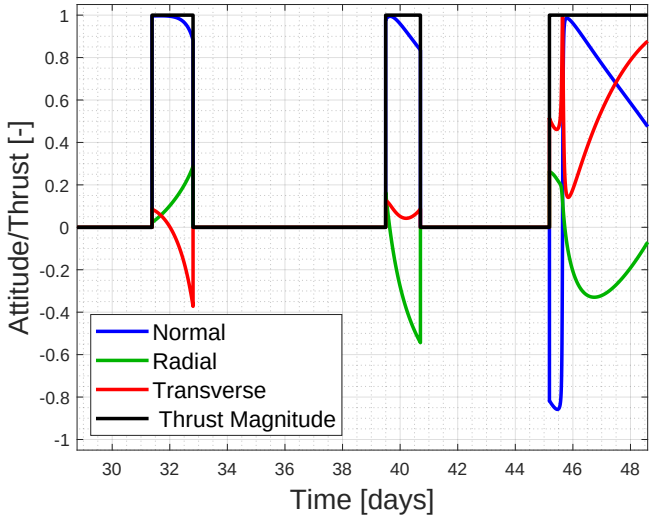

Figure 7.10: Attitude unit vector components and thrust profile for the chosen Pareto-optimal trajectory as a function of time.

The top and side views of the trajectory are portrayed in Figure 7.8 and Figure 7.9, respectively. The manifold phases resemble those present in the nominal trajectory (Figure 6.1). Moreover, many optimized transfers start their optimal control phase when the spacecraft is near the highest point of the unstable manifold, thus being a very promising starting point. The optimal control phase is however rather different from the nominal trajectory, carrying out in this case three lunar flybys, although farther away from the Moon. In addition, three coasting periods can be identified during the optimal control phase, greatly reducing the thrusting time to minimize mass consumption. The first coasting period is connected to the unstable manifold, being essentially equivalent to staying in the manifold phase for longer. The discontinuity between the optimal control and the stable manifold phases can be discerned in Figure 7.9.

The thrust profile (including the normalized attitude vector) is presented in Figure 7.10. For the sake of visualization, the attitude components have been set to zero during the coasting periods. As such, the discrete "jumps" in attitude are not physical and the profile itself is smooth. The fastest change in attitude (on day 45.6) corresponds to the third flyby, and the spacecraft does not thrust

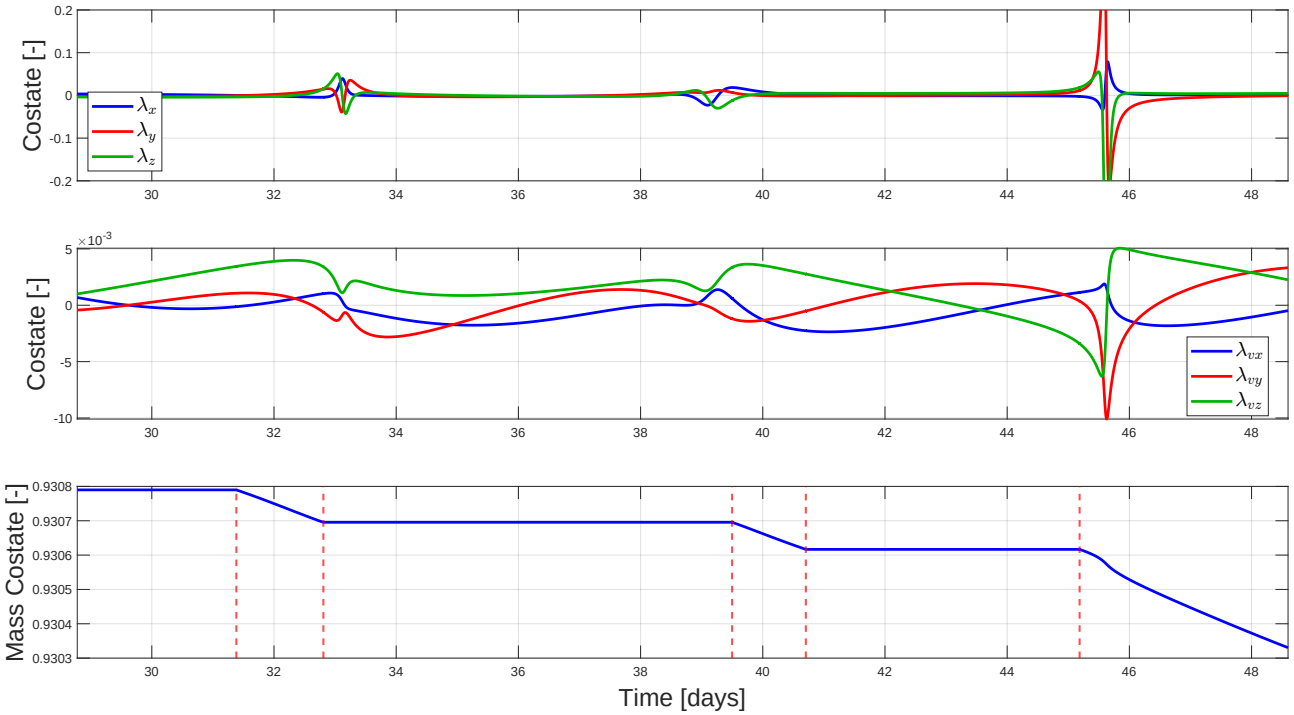

Figure 7.11: Costate parameter values for the chosen Pareto-optimal trajectory as a function of time.

during the other two flybys. The plot shows that the spacecraft coasts during most of the phase and that *bang-bang* control is employed, which is a necessary condition for optimality arising from the indirect method derivation.

The values of the seven costate parameters as a function of time can be found in Figure 7.11. As usual, strong variations in the position and velocity costates are experienced during the three flybys. Differently from the nominal trajectory, in this case, the mass costate remains constant over three distinct intervals, corresponding to the three coasting phases. To prove this statement, vertical dotted lines have been added to the mass costate plot corresponding to the points in time in which the thrusters are turned on or off. At these points, the mass costate either adopts a slope of zero or stops having a slope of zero. This in accordance with the costate EOM (Equation (5.8)). It should be noted that because the final value of λ_m is not 0, this is not a mathematically optimal trajectory with respect to the spacecraft mass, so an even better trajectory for this TOF could in principle be obtained.

7.3 Optimization without Manifolds

7.3.1 Set-up

For these optimization runs, both manifold phases were removed, having thus no need for T_{man} or τ_{orb_2}. The bounds for the rest of the design variables were kept the same as the ones in Table 6.10. The optimizer settings were also the same as the optimizations with manifolds (Table 7.1). When trying to minimize both the TOF and the propellant mass, the optimizer would always only find a small number of solutions, all around the same very promising trajectory with a TOF of 19.46 days. Said trajectory exhibits no coasting phases, meaning that much faster transfers are not expected to be possible. These poor results are attributed to the fact that, as the transfer without manifolds is more difficult, the islands would get populated with extremely fast trajectories that could never satisfy the constraints (one of the dangers of this constraint-handling strategy). Increasing the weight of the constraints objective even by four orders of magnitude had no effect on the results. Without resorting to a cumbersome re-tuning of the optimization settings, a suitable way of solving this issue was to minimize the propellant mass while maximizing the TOF, something easily done with the objective

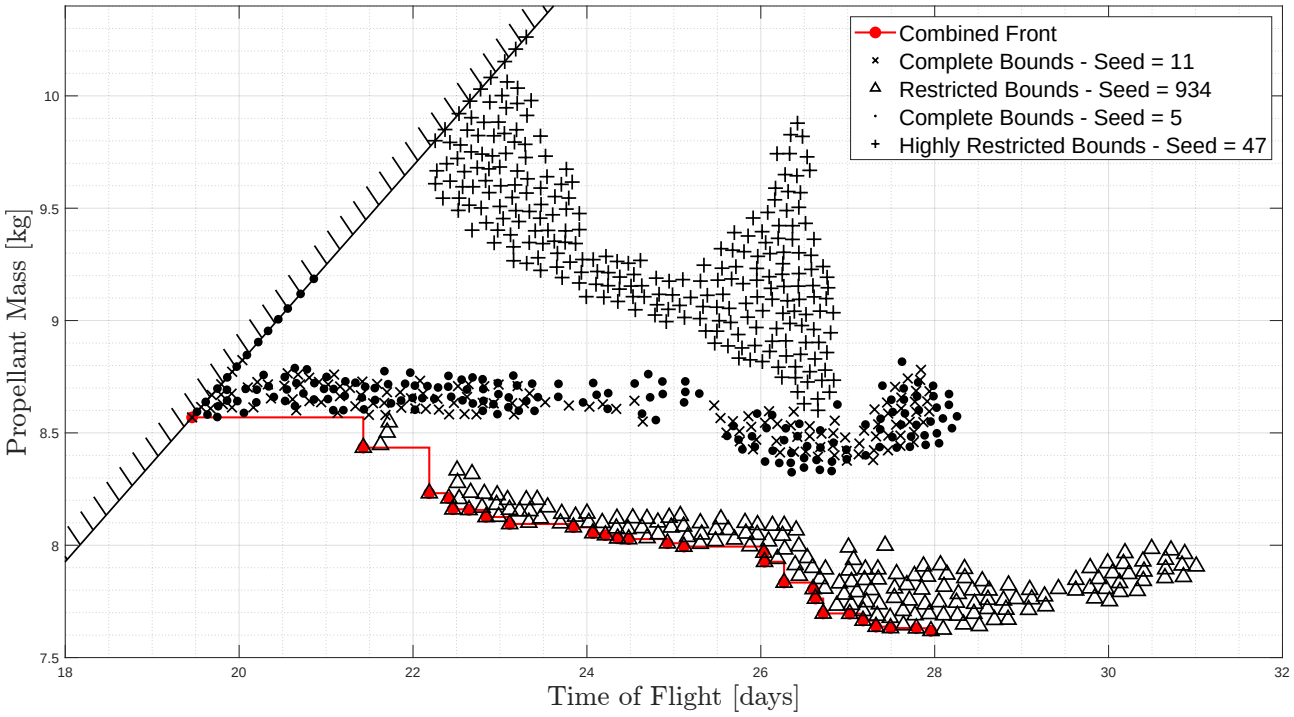

Figure 7.12: Trajectories found by MOEA/D and their combined Pareto front for several runs with differing seeds and departure orbit bounds. The boundary line corresponds to the upper limit due to the thruster capabilities.

balance parameter. This strategy urges the optimizer to find solutions with longer TOFs and discard the utopian trajectories from the populations. The proposed solution is of course not ideal, as the obtained Pareto front will not be optimal, but from a practical point of view, it is interesting as a wider front is obtained.

Furthermore, even with the new balance, all optimization runs found roughly the same solutions, exploring similar design variables. To obtain a more diverse set of solutions, the approach was to adjust the bounds of τ_{orb_1} such that these solutions could not be found. After a new set of trajectories was obtained, τ_{orb_1} was restricted even more also to exclude this set. As such, three runs were carried out, keeping everything the same except for the bounds of the initial orbit point.

7.3.2 Results

Figure 7.12 displays the feasible solutions obtained with the three runs and their combined Pareto front. An extra run was included with complete bounds in which the Jacobi limit for the computation of the closest state (Subsection 5.2.3) was removed. It can be seen from the figure that the Jacobi limit had barely any influence on the performance, even for a different pseudo-random seed. Hence, at least in some cases, the Jacobi constant limit reduces the optimization time without sacrificing performance.

The boundary line corresponds to the physical limit of the spacecraft propulsion system since, for a given TOF, the thrusters cannot expel more mass than the maximum mass flow ($\dot{m}_{dim} = \frac{T_{max_{dim}}}{I_{sp_{dim}}}$) times the TOF. As such, since *bang-bang* control is employed, all trajectories in Figure 7.12 that do not lie in the boundary line present coasting periods within the optimal control phase. This shows the importance of choosing appropriate design bounds to maximize the generation of coasting trajectories as much as possible. The combined front reaches this physical limit, meaning that the fastest solution always thrusts at its maximum level. Consequently, having found this Pareto optimal solution of 8.57 kg, this problem cannot have any Pareto-optimal solution that requires more than 8.57 kg. Moreover,

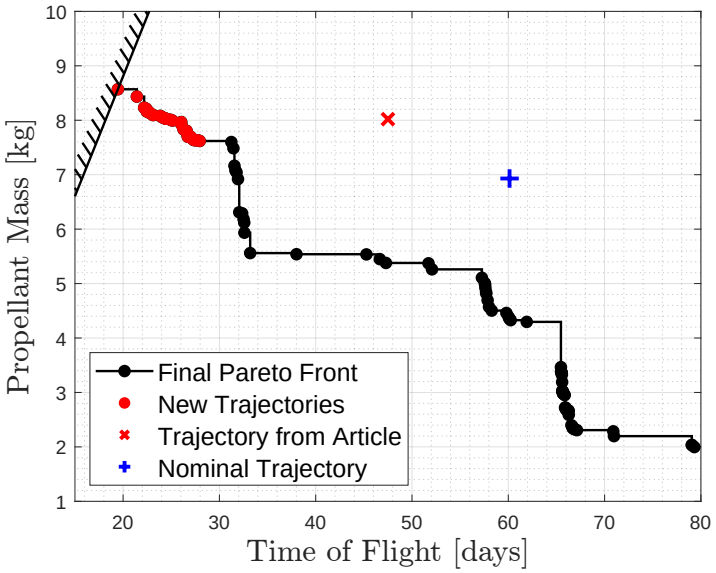

Figure 7.13: Final Pareto front of the chosen use case, including the optimization without manifold theory (with the new trajectories highlighted in red). The performances of the nominal trajectory and the one found by Pritchett et al. (2017) are included as well. The boundary line corresponds to the upper limit due to the thruster capabilities.

as explained by X. Pan and B. Pan (2020), the necessary conditions for the input controls ($\hat{\boldsymbol{u}}$ and τ) to minimize the TOF instead of the propellant mass are the same as the ones presented in Section 5.1 except for τ, which is forced to always be 1 (continuously thrusting). Hence, the proposed indirect optimization method is also capable of obtaining the fastest trajectories possible. A simple way to force the trajectories to not coast would be to set λ_m to an extremely high positive value (thus not affecting the EOM).

Regarding the individual runs, each found roughly 200 trajectories, an order of magnitude lower than the optimizations with manifolds. The obtained Pareto-front is considerably smooth and spans ten days and a kilogram of propellant mass. All trajectories except for the fastest correspond to the run that excluded the fastest trajectory from the design bounds. As such, the optimizer was forced to explore a different region, arriving at better results in terms of propellant mass. When restricting the bounds even more, the worst run was obtained.

Because the manifold phases need not be propagated and because the computation of the closest state is faster as the target trajectory is solely made up of the target orbit (without a 38-day manifold), the average computational time of each trajectory evaluation is only 0.012 seconds when having ten islands in parallel. Since convergence is achieved roughly around $3.5{\cdot}10^6$ evaluations, each run only took 12 hours, three times less than the optimizations with manifolds. Nevertheless, as already mentioned, the number of obtained trajectories and the range of the Pareto front is much lower. Several factors have been identified to probably contribute to this difficulty in finding trajectories:

- With manifolds, the collection of effective target points is one dimension larger, as any point in the manifold "tube" is a suitable connection. Without manifolds, the target trajectory is simply the halo "line". Even for a single trajectory evaluation, the target trajectory is four times longer if a 38-day manifold is added.

- For the same reason, there is a much wider variety of starting points for the optimal control phase.

Table 7.3: Design variable values and performance of the fastest Pareto-optimal trajectory.

Parameter	**Value**	**Unit**
τ_{orb_1}	0.600930	-
λ_x	-0.070783	-
λ_y	-0.091913	-
λ_z	0.023994	-
λ_{vx}	-0.058566	-
λ_{vy}	-0.094567	-
λ_{vz}	0.054386	-
λ_m	1.578137	-
m_p	8.57	kg
TOF	19.46	days
Position Error	725.78	km
Velocity Error	2.894	m/s
C Error	$2.896 \cdot 10^{-4}$	-

- Related to the previous, a different region in space may be a more promising and easier starting point than the initial halo.
- The required TOF of the optimal control phases when manifolds are not used are longer. This complicates the optimization as the indirect control law is very restrictive and "blind" since it does not consider the state error in the EOM.

This new Pareto-front can be patched with the front from Figure 7.7, obtaining Figure 7.13, which highlights the new trajectories in red. The optimization runs without manifold theory extended the original front with faster trajectories, dominating only a few of the original solutions and making them obsolete. Hence, both optimization configurations complemented each other, exploring different regions of the Pareto front and providing the mission designer with 100 optimized trajectories from

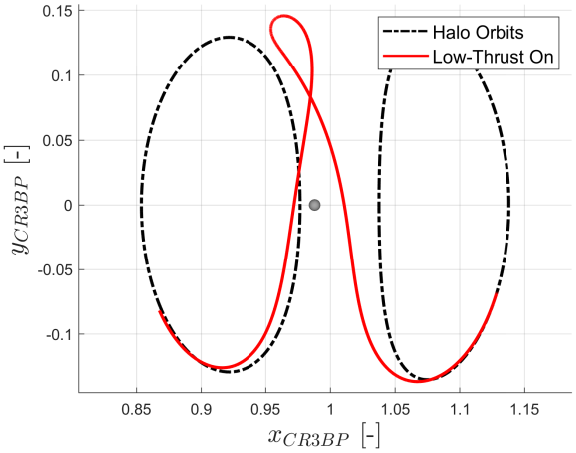

Figure 7.14: Top view of the fastest obtained trajectory.

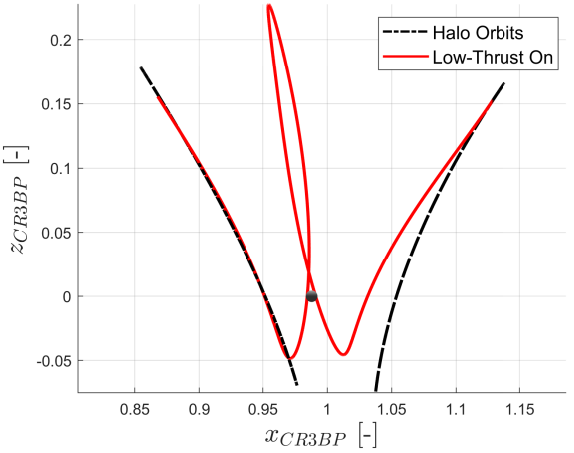

Figure 7.15: Side view of the fastest obtained trajectory.

which to choose. From the figure, it can be seen that both the nominal solution and the trajectory found by Pritchett et al. (2017) are contained in the hypervolume of the Pareto front with respect to the nadir point. Moreover, with 8.02 kg of available propellant mass, the optimal transfer now only requires 24.9 days. For reference, the utopia point of the Pareto front is 1.99 kilograms and 19.45 days, and the nadir point is 8.75 kg and 79.32 days.

The parameters and performance of the fastest trajectory can be found in Table 7.3. The error values lie within the constraints, although the velocity error is close to the allowable limit. Since this solution exhibits no coasting periods, the value of λ_m again does not influence the trajectory as long as it is positive enough. The transfer can be visualized in Figure 7.14 from the top, and in Figure 7.15 from the side. Evidently, there are no manifold phases in this case. Furthermore, the spacecraft does not perform any revolutions around the Moon, as would be expected for the fastest transfer when considering that the rotation direction ought to be inverted.

In conclusion, not only does the proposed approach provide a wide variety of promising initial guesses for the considered use case, but also the trajectories greatly outperform literature values in terms of propellant mass and time of flight for a comparable use case. Nevertheless, the trajectories ought to still be refined with ASTOS to remove the instantaneous discontinuities between the phases. Lastly, manifold theory has proven to discover new areas of the Pareto front on top of easing the obtainment of feasible transfers.

Analysis of Results 8

In this chapter, the obtained trajectories will be analyzed, focusing on their verification and sensitivities. Moreover, the suitability of the transfers to be used as initial guesses for direct collocation methods is studied. The verification process of the nominal trajectory and the selected Pareto-optimal solution is presented in Section 8.1. Then, the trajectories are refined with collocation methods using ASTOS in Section 8.2. Next, a sensitivity analysis of the nominal trajectory is carried out in Section 8.3. Lastly, Section 8.4 explores the main accelerations not modeled by the CR3BP.

8.1 Trajectory Verification

8.1.1 Switching Function Check

To satisfy the necessary conditions for local optimality, the normalized thrust magnitude, τ, ought to obey Equations (5.10) and (5.11). As such, the switching function, S, of a given trajectory must change signs precisely when the thrust is activated or switched off.

Figure 8.1 plots the switching function of the selected Pareto-optimal solution (Subsection 7.2.2) as a function of time. This function is, of course, only meaningful during the optimal control phase. When S is positive, the spacecraft must coast, elsewhere, it must thrust at its maximum level. In the figure, the points in time at which the spacecraft starts or stops thrusting can be identified by the red dotted vertical lines. These lines correspond to the coasting and thrusting intervals of Figure 7.10. As required, these red lines coincide with the points where the switching function changes signs and the thrust is indeed off whenever S is positive.

8.1.2 Jacobi Constant and Mass Check

Whenever the spacecraft is coasting, both the Jacobi constant, C, and the mass of the spacecraft must remain constant. The Jacobi constant of the selected Pareto-optimal solution as a function of time is displayed in Figure 8.2 for all phases. The plot suggests that C is indeed constant during the orbit and manifold phases. Although not distinguishable, there is a slight change in C when transitioning between the orbit and manifold phases, generated by the perturbation parameter ϵ_p required to trigger

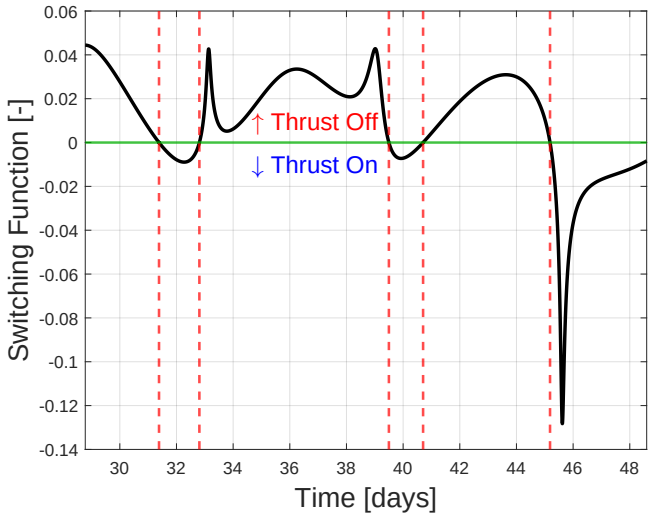

Figure 8.1: Switching function, S, of the selected Pareto-optimal trajectory as a function of time. The red vertical lines correspond to the time points at which the thrust is switched on or off.

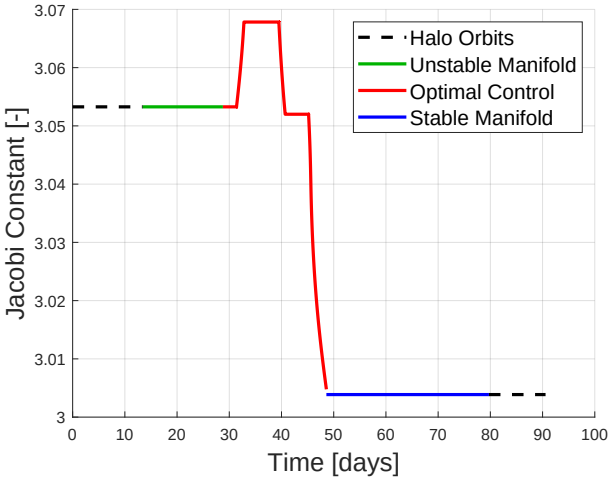

Figure 8.2: Jacobi constant of the selected Pareto-optimal trajectory as a function of time.

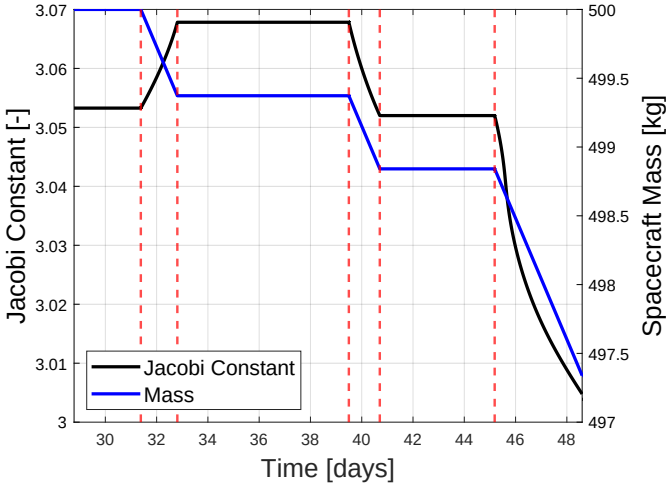

Figure 8.3: Jacobi Constant and spacecraft mass throughout the optimal control phase of the selected Pareto-optimal trajectory. The red vertical lines correspond to the time points at which the thrust is switched on or off.

the manifolds. Then, a strong variation in C can be recognized during the optimal control phase, almost connecting to the target Jacobi constant. The difference is, of course, within the allowable error of 10^{-2}. During the coasting intervals of the optimal control phase, C appears to be constant. Via a close examination of the specific C values when independently computing C using the trajectory states, it was found that the order of accuracy is at least 10^{-11} in non-dimensional units, as no variation is observed up to this order. For reference, the difference in C of the orbits is $4.9396 \cdot 10^{-2}$.

Focusing on the optimal control phase, Figure 8.3 compares the spacecraft mass and the Jacobi constant as a function of time. As mandated by the EOM displayed in Equation (5.2), the mass must remain constant while the spacecraft is coasting, coinciding with the plateaus in C. The figure proves this fact by also pointing out the points at which the thrust is switched on and off (vertical red lines). Moreover, whenever the thrust is on, both C and the mass vary over time, with the mass linearly decreasing with a constant slope of $\dot{m}_{dim} = \frac{T_{max_{dim}}}{I_{sp_{dim}}}$ (checked numerically).

This analysis reveals that not only are the coasting equations of motion implemented correctly, but also that the integrator settings are accurate enough when coasting in any phase, even when flybys are included. In addition, the thrust was proven to indeed not affect the EOM when the spacecraft is meant to be coasting.

8.1.3 Trajectory Error

For the reasons explained in Section 6.2, only the error of the optimal control phase ought to be assessed. The integrator analysis aimed for maximum non-dimensional position and velocity errors of 10^{-5}, including the effect of the error in the thrust strategy computation via Equation (5.8) as well. Taking again the selected Pareto-optimal solution, the approach was to compare the trajectory to one integrated with much tighter tolerances and a significantly smaller maximum step size, i.e., the benchmark.

The position and velocity errors of the optimal solution as a function of time are displayed in Figure 8.4. In the figure, the peaks in error due to the three flybys can be identified, with the third one being the steepest as it is the only one in which the thrust is active. Even though the final position and

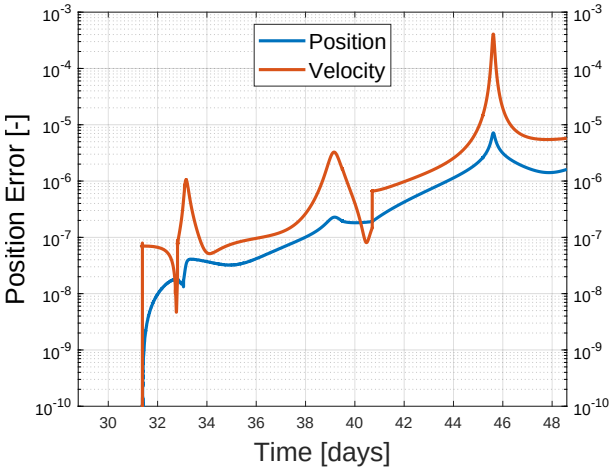

Figure 8.4: Non-dimensional position and velocity errors as a function of time of the selected Pareto-optimal trajectory when allowing the thrust profile to vary.

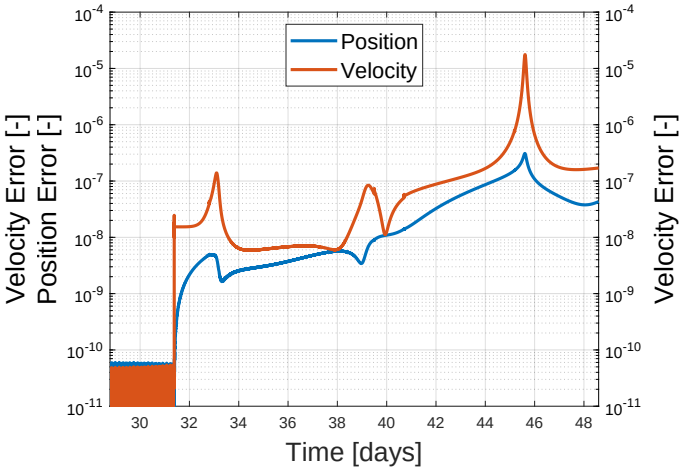

Figure 8.5: Non-dimensional position and velocity errors as a function of time of the selected Pareto-optimal trajectory when interpolating a fixed thrust profile.

velocity errors are below 10^{-5} as desired, the velocity error significantly exceeds this limit in the third flyby, which was not the case for the nominal trajectory. This phenomenon has two main reasons. Firstly, this trajectory exhibits three flybys instead of two, which are the most critical segments in terms of error accumulation. Secondly, discrete "jumps" in velocity can be identified roughly on days 31.5, 32.7, and 40.5. These time points correspond to when the thrust is switched on or off. They can be explained by the fact that the spacecraft starts or stops thrusting slightly earlier or later due to the error when solving Equation (5.8). The sensitivity of the switching function is something that P. Zhang et al. (2013) already warned about.

Nevertheless, the error displayed in Figure 8.4 is not the actual error of the trajectory within the CR3BP, as it includes the deviations due to the thrust strategy not being completely optimal (essentially being a different trajectory since the thrust profile is distinct). To quantify the actual errors in position and velocity due to the integration accuracy, Equation (5.8) can be disregarded, and the original thrust profile is interpolated when generating the benchmark. The resulting errors are plotted in Figure 8.5. Now, only the first "jump" in velocity error is present, which may have to do with interpolation inaccuracies, anyway entailing considerably small errors. As predicted, the other two "jumps" are absent and the maximum error of the velocity is now 10^{-5}.

In conclusion, this proves to an acceptable extent that the chosen integrator settings are appropriate for the thrusting periods as well. Furthermore, even if for some transfers the accuracy requirements are not satisfied, these errors would be an order of magnitude lower than the discontinuity present between the optimal control phase and the target manifold. As such, these trajectory errors can be easily corrected with direct collocation using the ASTOS software, as a much larger deviation is corrected anyway.

8.1.4 Verification with the ASTOS Software

The complete trajectory can be verified by replicating it using exhaustively verified software such as ASTOS. The procedure is to repropagate each phase by solely supplying the initial states of each phase, their duration, and the thrust magnitude and attitude vector as a function of time. Of course, the same dynamic model and Julian Date is used. In addition, the trajectory ought to be propagated

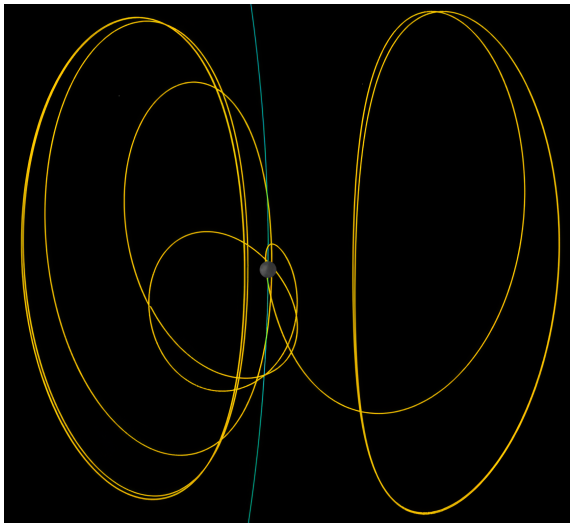

Figure 8.6: Visualization of the replica of the nominal trajectory using the ASTOS software.

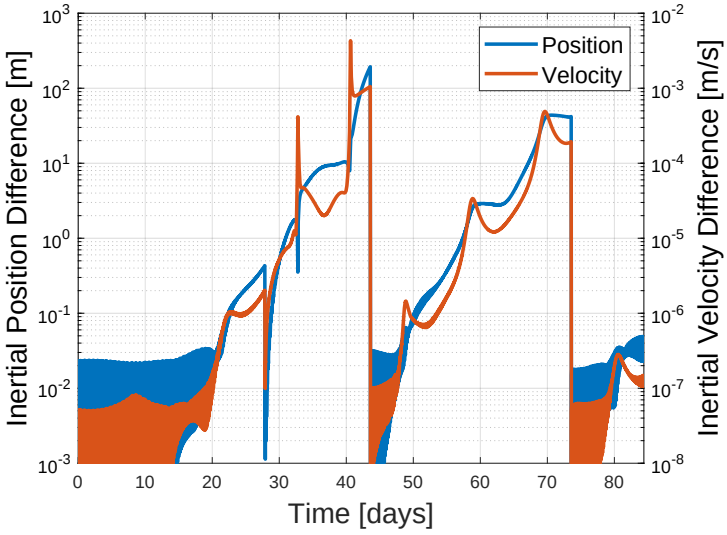

Figure 8.7: Difference in inertial position and velocity between the nominal trajectory generated with the OGT and the replicated trajectory in ASTOS.

with similar settings. As such, DormandPrince78 with absolute and relative tolerances of 10^{-9} and a maximum non-dimensional step size of 10^{-3} was used to replicate the nominal trajectory.

Figure 8.6 displays the top view of the replicated nominal trajectory as visualized in ASTOS. The spacecraft trajectory is shown in yellow, whereas the blue line corresponds to the path of the Moon. The trajectory looks very similar to Figure 6.1. To quantify their degree of similarity, the difference in position and velocity between the nominal trajectory and the replica is plotted in Figure 8.7. It can be seen that during most of the first two phases (up to day 28) the errors are mostly dominated by the computer rounding errors. The same goes for the last phase, the target EML_1 halo. In the optimal control phase (from day 28 to day 44) and the stable manifold phase (from day 44 to 74) the error in both position and velocity gradually increases over time. The two flybys during the optimal control phase can be identified as the two sharp peaks in velocity error, as usual. However, as the nominal trajectory already exhibits these peaks in error in Figure 6.12, the aforementioned peaks in Figure 8.7 are not of relevance. As the nominal trajectory itself has a maximum error of roughly 50 meters and 1 cm/s, the obtained maximum differences are considerably close to the trajectory error. In the case of the velocity, the difference is almost an order of magnitude lower than the trajectory error. Henceforth, the generated trajectories can be considered to be verified with this exercise, proving that the thrusting equations of motion are also implemented correctly.

Fully validating the trajectories is an infeasible task since no low-thrust missions have been launched between halo orbits, let alone within the specific use case employed in this research. The most that can be said is that the performance of the trajectories seems to be in accordance with the one obtained by Pritchett et al. (2017) for a comparable use case, as demonstrated in Section 7.2, and trajectories of similar shape have been found.

8.2 Trajectory Refinement with ASTOS

The goal of this section is to prove that the OGT trajectories serve as promising initial guesses for collocation methods. For this purpose, the Collocation And Multiple Shooting Trajectory Optimization Software (CAMTOS) (Gath, 2002), implemented in ASTOS, is employed. Due to trajectory

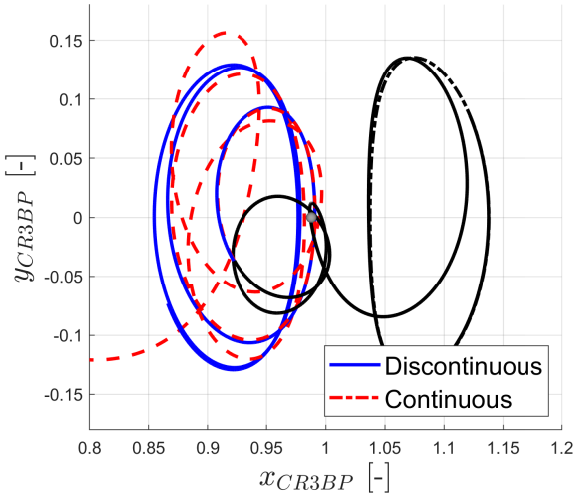

Figure 8.8: Comparison between the nominal trajectory and the trajectory obtained if no discontinuity was applied to enter the target manifold phase.

optimization problems with collocation methods usually requiring thousands of design variables, a local optimizer is the best option. Within CAMTOS, the chosen local optimizer to solve the non-linear programming problem will be WORHP, as it tends to have good performance even with large numbers of design variables and is not too sensitive to set-up settings. It should be noted that in this research, the OGT trajectory discontinuities will be refined with direct collocation without focusing on propellant mass minimization. As such, the transfers, which are already outstandingly promising, could in principle be optimized further. The objective is to make them feasible while maintaining the same TOFs and similar propellant consumptions.

8.2.1 Nominal Trajectory

The first trajectory to be refined is the nominal trajectory, making use of the replica displayed in Figure 8.6. To evidence the need for this refinement process, Figure 8.8[1] shows the (infeasible) nominal trajectory compared to the trajectory that would be obtained if the instantaneous discontinuity of 308 km and 0.93 m/s between the optimal control and the stable manifold phases would not be applied. Due to the Jacobi constant difference being small, both trajectories remain considerably close during the first revolution around the Moon, gradually diverging until the continuous trajectory leaves the halo vicinity. Hence, although the discontinuity is considerably small, due to the system's instability, the correction with ASTOS is necessary for the trajectory to be useful in mission design. Aside from the discontinuity between the optimal control and manifold phases, the two smaller discontinuities to transition from the orbit phases to the manifold phases must also be rectified.

The trajectory was then refined with direct collocation using the Hermite-Simpson approximation. The chosen number of collocation nodes to approximate the states was roughly 30 nodes per day. For the optimal control phase, however, due to the presence of flybys, it is interesting to have a higher grid density where the dynamics are faster to ease the optimization and make it more accurate. Therefore, the grid for this phase was generated by taking every five steps of the OGT trajectory, which uses a variable-step integrator to adapt to the dynamics. For the refinements, the TOFs of each phase were fixed to the initial guess values, and the thrust magnitude was allowed to vary. It should be noted that

[1]None of the displayed trajectories are continuous in the rigorous sense, as the solution of the EOM is approximated with a finite number of steps. Nevertheless, the word "continuous" here refers to a trajectory that exhibits no instantaneous discontinuities between the phases, thus satisfying the EOM.

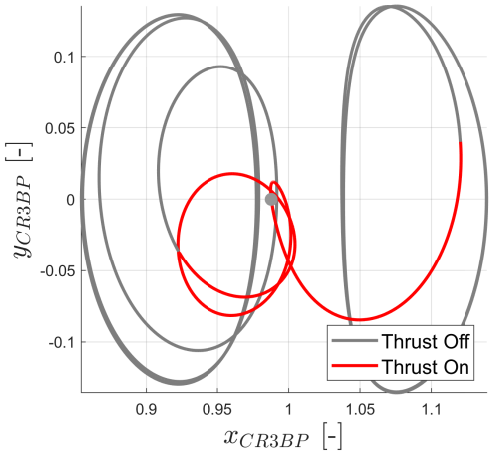

Figure 8.9: Top view of the nominal trajectory refined with the ASTOS software.

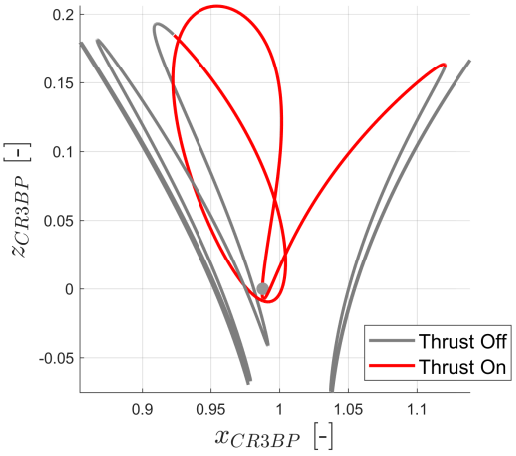

Figure 8.10: Side view of the nominal trajectory refined with the ASTOS software.

the grid generation strategy and the optimization settings are common to all ASTOS refinements in this research.

The refined trajectory can be seen in Figure 8.9, from the top, and in Figure 8.10, from the side. As expected, the trajectory looks very similar to the nominal trajectory (Figure 6.1 and Figure 6.2), however, the aforementioned discontinuities are not present in this case. The thrust only appears to be on during the optimal control phase because, due to the highly unstable system, a negligibly small thrust perturbation at the appropriate time can make the spacecraft transition between the manifold and periodic orbit phases. For reference, the obtained values of τ to carry out this transition never exceeded 10^{-5}, a value far smaller than the thrust magnitude uncertainty. Moreover, in a high-fidelity model, the halo and manifold phases would adopt significantly different shapes, so much larger corrections would be required anyway. For these reasons, the orbit and manifold phases are not explored further in this research within the CR3BP and are left for the GNC subsystem to handle or for when the trajectories are fully transitioned to a high-fidelity model.

As already mentioned, the TOF of the refined trajectory is identical to that of the initial guess. Moreover, the required propellant mass is 6.94 kg, virtually equivalent to the original 6.93 kg. Hence, the performance variation is minimal, making the Pareto-fronts arising from the OGT accurate and helpful in estimating the performance of the final trajectory. Furthermore, the optimization of this trajectory (end-to-end) took less than 20 minutes[2]. The thrust profile during the optimal control phase is displayed in Figure 8.11. This profile also closely resembles the thrust profile of the nominal trajectory (Figure 6.4). The difference between both figures is presented in Figure 8.12, with the difference in attitude components and normalized thrust magnitude being generally below 10^{-2}. This demonstrates that the OGT trajectories are very promising initial guesses that require small corrections to be feasible within the CR3BP.

In terms of states, the position and velocity differences between the refined trajectory and the optimal control phase are shown for the optimal control phase in Figure 8.13. As usual, the differences rise steeply at the flyby points, getting reduced after leaving the lunar vicinity. The difference is already considerable at the beginning as the starting point is not the same, however, at the end,

[2]It should be noted, however, that this optimization time does not account for the challenges involved in correctly setting up the optimization problem in ASTOS to converge to a sufficiently accurate solution.

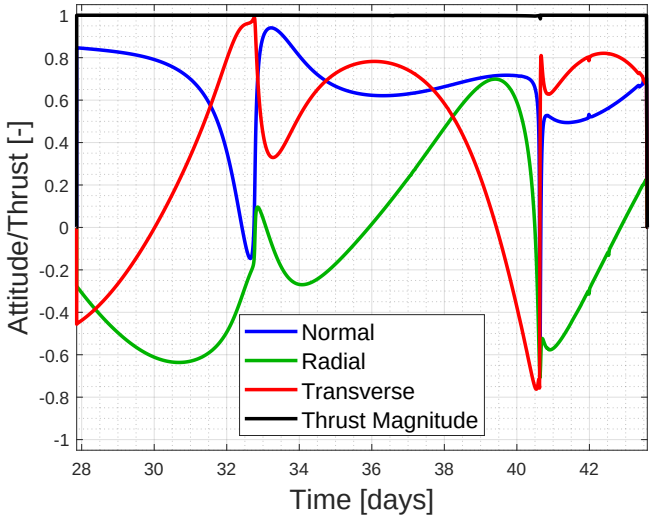

Figure 8.11: Attitude unit vector components and thrust profile as a function of time of the nominal trajectory refined with the ASTOS software.

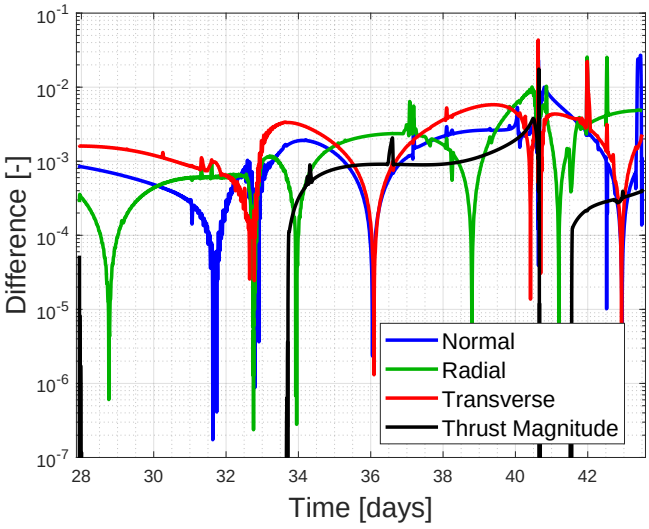

Figure 8.12: Difference in thrust magnitude and thrusting attitude between the nominal trajectory generated with the OGT and the refined trajectory using the ASTOS software.

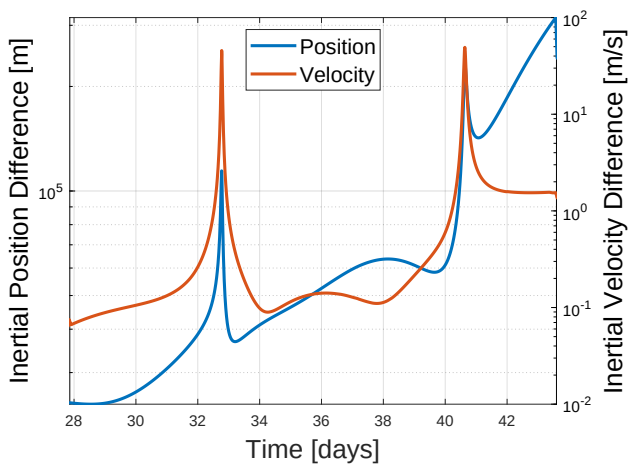

Figure 8.13: Difference in inertial position and velocity between the nominal trajectory generated with the OGT and the refined trajectory.

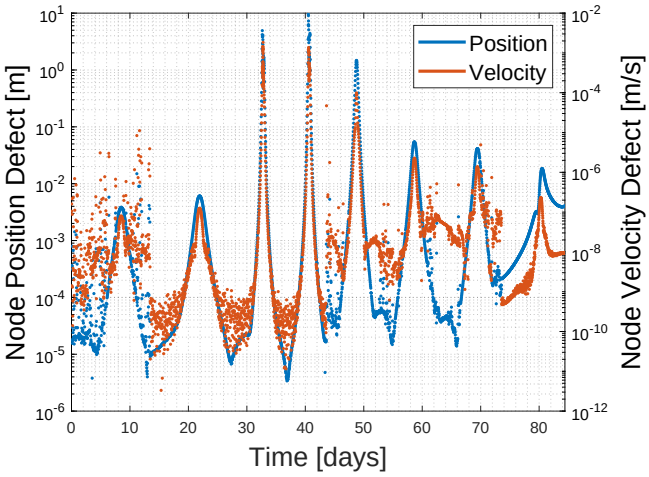

Figure 8.14: Integration position and velocity defects at each collocation node of the refined trajectory.

the difference corresponds to roughly 300 km and 1 m/s, the magnitude of the discontinuity to be corrected.

Lastly, each collocation node's position and velocity defects are shown in Figure 8.14. These defects were obtained by propagating the trajectory in ASTOS using the integrator settings specified in Subsection 8.1.4 in multiple-shooting mode. This means that each node is propagated up to the point in time corresponding to the subsequent node, then, the trajectory "jumps" to the state of this subsequent node and a new propagation between nodes starts. As such, the defects from Figure 8.14 include the propagation error of the integrator. Looking at the figure, the defects are generally larger whenever the spacecraft is closer to the Moon, presenting oscillations according to the trajectory

Table 8.1: Design variable values and performance of the nominal trajectory with relaxed constraints.

Parameter	**Value**	**Unit**
τ_{orb_1}	0.367695	-
T_{man}	14.4479322	days
λ_x	0.090420	-
λ_y	-0.021418	-
λ_z	0.202595	-
λ_{vx}	-0.481627	-
λ_{vy}	-0.632665	-
λ_{vz}	0.986062	-
λ_m	-2.756724	-
τ_{orb_2}	0.379244	-
m_p	6.94	kg
TOF	60.08	days
Position Error	1110.80	km
Velocity Error	1.17	m/s
C Error	$2.06 \cdot 10^{-3}$	-

dynamics. The highest peaks correspond to the two flybys, as could be predicted. Including the integration error, the least accurate node exhibits an error of 10 meters and 1 mm/s, making this trajectory highly accurate considering that the disturbances due to the unmodelled accelerations would be much larger. For reference, when propagating each phase with single-shooting, the final error never exceeds the orbit determination lower limit of a few kilometers and several centimeters per second (Subsection 6.2.2).

8.2.2 Constraint Relaxation

Realizing that the original constraints are tight enough for the OGT trajectories to be easily refined with ASTOS, this subsection aims to determine to what extent the constraints could be relaxed while still obtaining feasible trajectories of performances similar to the initial guess without excessive effort. The purpose of this relaxation is to improve the results of the heuristic optimizations of the OGT, as proven by Section 7.1.

Firstly, the magnitude of the discontinuity between the optimal control and the target manifold phase was doubled. This new transfer was again easily corrected in ASTOS. Then, the constraint values were increased to three times the original, i.e., $3 \cdot 10^{-3}$ both in position and velocity. The parameters and performance of this relaxed trajectory can be found in Table 8.2. This trajectory is profoundly similar to the nominal trajectory, also in terms of thrust profile, with the main difference being the position and velocity errors.

The thrust profile of the refined trajectory[3] is displayed in Figure 8.15. The largest differences in attitude with Figure 8.11 occur after the second flyby. Moreover, the thrust magnitude was reduced during the refinement, requiring 6.91 kg of propellant mass instead of the original 6.94 kg for the same TOF.

The node defects are shown in Figure 8.16, with the peaks again corresponding to the two flybys. In this case, the trajectory is more accurate, with the least accurate node revealing an error of 0.1 meters and 10 μm/s. To further demonstrate the accuracy of the trajectory, Table 8.2 shows the final position and velocity discontinuities with respect to the target manifold when taking the optimized

[3]Recall that only the optimal control phase is corrected here.

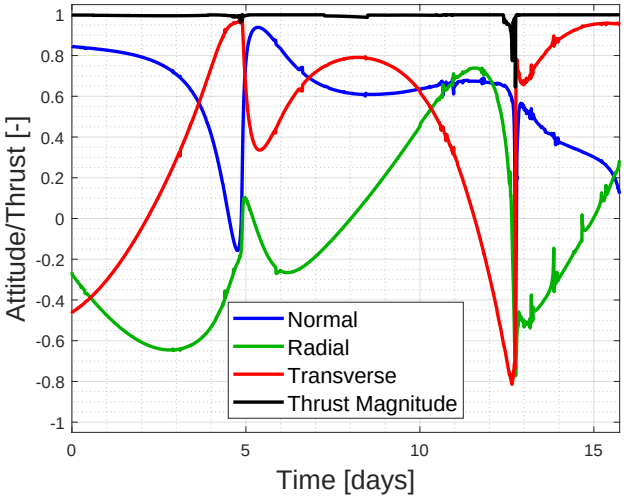

Figure 8.15: Attitude unit vector components and thrust profile as a function of time of the trajectory with relaxed constraints refined with the ASTOS software.

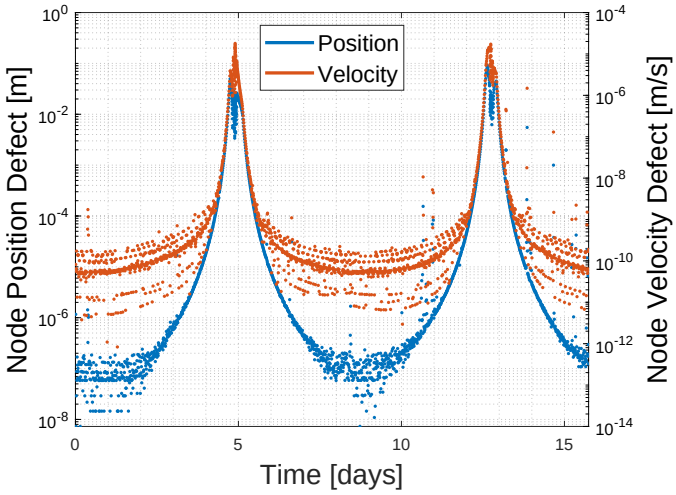

Figure 8.16: Integration position and velocity defects at each collocation node of the refined trajectory with relaxed constraints.

Table 8.2: Design variable values and performance of the nominal trajectory with relaxed constraints.

Propagation Mode	**Final Position Error** [km]	**Final Velocity Error** [m/s]
Collocation Points	$1.24 \cdot 10^{-7}$	$6.65 \cdot 10^{-11}$
Single-Shooting	0.10	$2.54 \cdot 10^{-4}$

collocation points and when propagating in single-shooting from the first node. As expected, the obtained error with the collocation points is null and simply proves that the optimization converged to a feasible solution in terms of final constraints. Of greater interest is the fact that the final errors with single-shooting are again lower than the spacecraft OD capabilities in the cislunar environment. It should be noted that the expected final integration error of this trajectory is 50 meters and 0.1 mm/s, so the actual values of the position and velocity errors are not as meaningful.

This analysis proves that it is appropriate to relax the position and velocity constraints by a factor of three. In addition, it is also possible to reduce the position and velocity errors to the original constraint values (and even further) by means of a refinement run within the OGT itself, to ease the ASTOS optimization. Lastly, the constraints could probably be relaxed even more, but this conservative approach ensures that the trajectories can be refined without excessive effort while maintaining roughly the same performance.

8.2.3 Selected Optimal Solution

A similar trajectory refinement was carried out with the selected Pareto-optimal trajectory displayed in Figure 7.8. This transfer is of interest as it not only is an optimized trajectory that requires a very low propellant mass but also exhibits large coasting intervals during the optimal control phase, unlike the nominal trajectory. It is important to ensure that the refinement process can also easily correct the large leaps in thrust magnitude.

The refined thrust profile is displayed in Figure 8.17. In this case, the complete attitude profile is included for the sake of visualization as the thrust, although low, is not zero during some portions of the coasting intervals. Throughout the original thrusting intervals, the throttle remains at its maximum

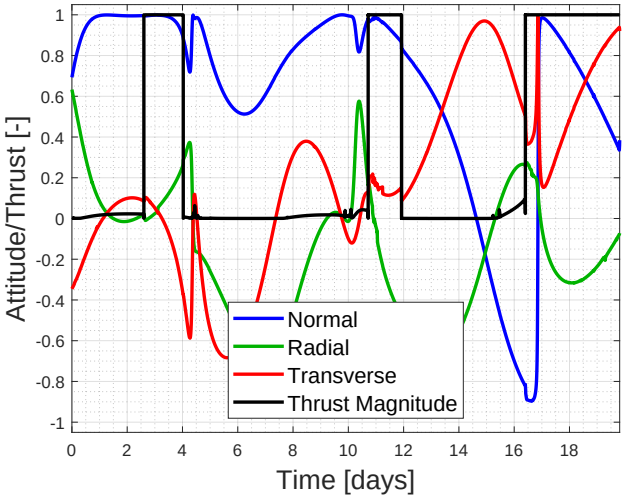

Figure 8.17: Attitude unit vector components and thrust profile as a function of time of the selected Pareto-optimal solution refined with the ASTOS software.

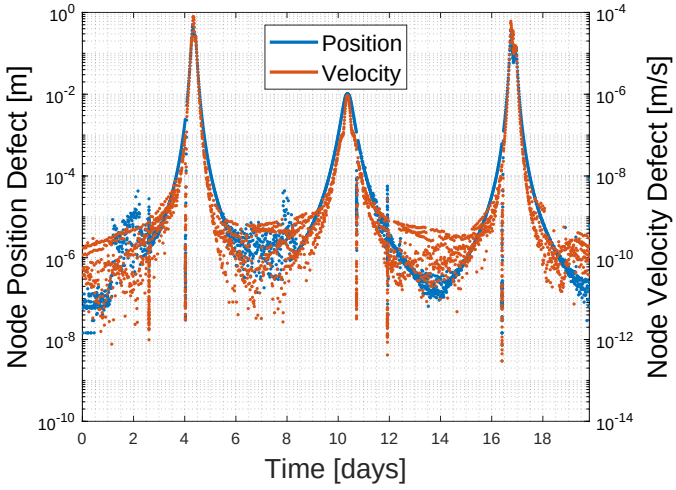

Figure 8.18: Integration position and velocity defects at each collocation node of the refined Pareto-optimal solution.

level. With the same TOF, this trajectory has a slightly worse performance in terms of propellant mass: 2.72 kg instead of the original 2.66 kg. Nevertheless, this suggests that the optimized trajectories can be suitable initial guesses for direct collocation, and by freeing up the TOF and focusing on mass minimization, the performance could in principle be improved even more.

As with the rest of the refinements, the position and velocity defects of each node are shown in Figure 8.18, with maximum defect values similar to those in Figure 8.16, suggesting that the accuracy of the trajectory is sufficient. In this case, three peaks can be recognized, logically corresponding to the three flybys. Furthermore, there are five vertical (due to the scaling) lines present in the figure that correspond to the time points at which the thrusters are switched on or off. These indicate that the optimizer needed to more strongly focus on these events, as they are critical in terms of error propagation.

8.3 Sensitivity Analysis

In this section, the sensitivities of the nominal trajectory (Section 6.2) with respect to the initial state and parameters are analyzed via a Monte Carlo method. In each evaluation, the optimal control phase is propagated with a perturbation, and the resulting position and velocity errors relative to the target trajectory (stable manifold and target halo) are computed. Instead of propagating for a specific TOF, the phase is propagated for a very long time and the closest state to the target trajectory is computed via the procedure described in Subsection 5.2.3 (with w_{pos} and w_{vel} both set to 1). The analysis is carried out in this manner to quantify how far the spacecraft would tend to get from the target trajectory instead of simply from a point in space, exploiting the recovery capabilities of the transfer by attaching somewhere else to the stable manifold. As such, a more realistic measure of the severity of the uncertainties can be acquired. This also means, however, that the thrust strategy will need to be recomputed in each evaluation according to the indirect method theory presented in Section 5.1, being slightly different every time due to the change in states (thus contributing to the error). Nonetheless, this is not predicted to have a great influence as long as the errors are relatively small.

Two Monte Carlo runs will be carried out in this analysis, one in which there is only uncertainty in the initial position and velocity of the optimal control phase and another in which the thrust magnitude

and initial mass uncertainties are also included. For a complete robustness analysis in a real mission, other parameters could be included, such as the thrust alignment, but the proposed variables are considered sufficient to analyze the controllability of the trajectory and the instabilities of the dynamic system. Each run will comprise 40,000 trajectory evaluations to obtain sufficient statistical confidence and to ease the identification of trends. It should be noted that the nominal trajectory already has errors in position and velocity. As such, it was refined with the OGT to get the errors down to $\sim$40 km and $\sim$0.2 m/s. Due to the discrete nature of the search strategy, it would be difficult to obtain trajectories of lower error. Nevertheless, this trajectory is accurate enough for the purposes of this analysis, as the expected errors are considerably larger on average.

For all variables, a normal distribution is assumed, thus sampling from a multivariate normal distribution. The Java class employed for the sampling is *MultivariateNormalDistribution*[4] from the *Apache Commons* library, which employs the algorithm *Well19937c* as pseudo-random number generator and Cholesky decomposition for the multivariate distribution. An arbitrary pseudo-random seed of 31 was selected. For the uncertainty in the initial position, the state-of-the-art OD capabilities within the cislunar environment are estimated from the work by Scorsoglio et al. (2023). Hence, a standard deviation of 3 km was selected for the x-coordinate, 1 km for the y-coordinate, and 300 m for the z-coordinate. Similarly, the standard deviations of the velocity vector are $(4, 2.5, 2.5)^T$ cm/s. The order of magnitude of these values is also in accordance with Newman et al. (2022). The standard deviation of the thrust magnitude is $5 \cdot 10^{-4}$ N (0.5%), as suggested by Maestrini et al. (2023) for accurate thrusters. Lastly, an arbitrary value of 1 kg (0.2%) was selected for the mass standard deviation, as there is no extra information on the spacecraft itself and the mass uncertainty is highly dependent on the mission. Nonetheless, the specific value is not essential because the aim of this analysis is simply to quantify how sensitive the trajectory is to the considered parameters

8.3.1 Initial Position and Velocity Sensitivities

The fitted probability density functions of the position and velocity errors of the first Monte Carlo run are presented in Figure 8.19 and Figure 8.20, respectively. The cumulative distributions and the mean value are also included for reference. Both the position and velocity errors appear to be governed by a *log-normal* probability distribution, which is sensible since the norm of the error components is taken. The figures show that the magnitude of the errors is outstandingly low, with almost all trajectories being within the original tight optimization constraints of 384 km and 1 m/s. Moreover, the average position error is only 109 km, an outstanding value when considering that no corrections were carried out and that the propagation lasted 15.7 days and included two flybys considerably close to the Moon. The ranges of variation in propellant mass and TOF are around 20 grams and 1.5 hours, meaning that the optimal control phase connects to the stable manifold always in practically the same spot.

Figure 8.21 plots the final position error as a function of the initial position and velocity deviations. Figure 8.22 displays the same data with the x-axis and color bar data swapped. From these figures, it can be concluded that the velocity uncertainty is more detrimental to the trajectory than the position uncertainty due to the linear trend that can be identified in Figure 8.26. This trend can also be recognized in Figure 8.21 through the uniform color gradient, positively correlated with the final position error. In addition, low error trajectories can be obtained even at large position deviations. This higher sensitivity to the velocity might be attributed to the fact that the standard deviations of the velocity are roughly three times larger than those of the position in non-dimensional units. The empty spot in the bottom left corners of both figures is due to the inherent error of the original trajectory and to the closest state search strategy. For the final velocity errors, similar trends are identified, and the same conclusions can be drawn.

[4]URL:https://commons.apache.org/proper/commons-math/javadocs/api-3.6.1/org/apache/commons/math3/distribution/MultivariateNormalDistribution.html [Accessed: 15/07/24]

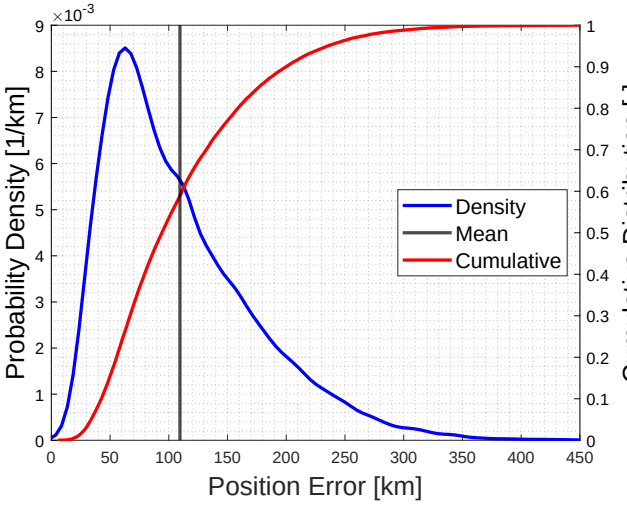

Figure 8.19: Fitted probability density function and cumulative distribution function of the position error of the Monte Carlo analysis considering only initial position and velocity uncertainties. The vertical black line corresponds to the mean value.

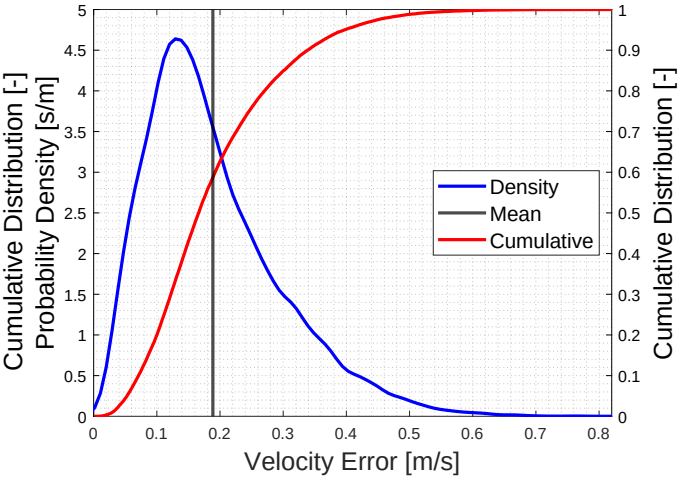

Figure 8.20: Fitted probability density function and cumulative distribution function of the velocity error of the Monte Carlo analysis considering only initial position and velocity uncertainties. The vertical black line corresponds to the mean value.

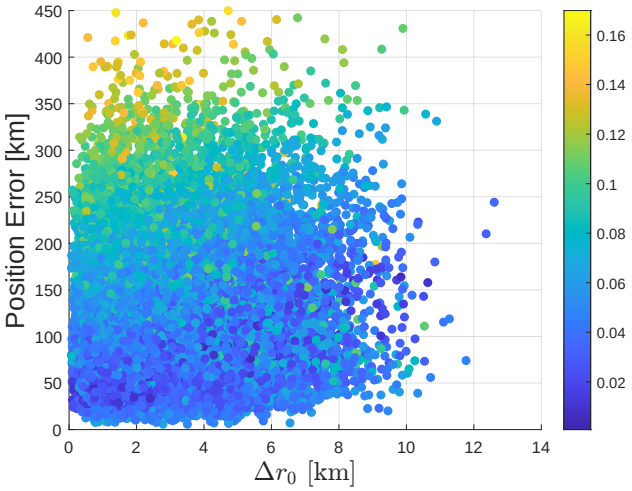

Figure 8.21: Position error as a function of the uncertainty in the initial spacecraft position.

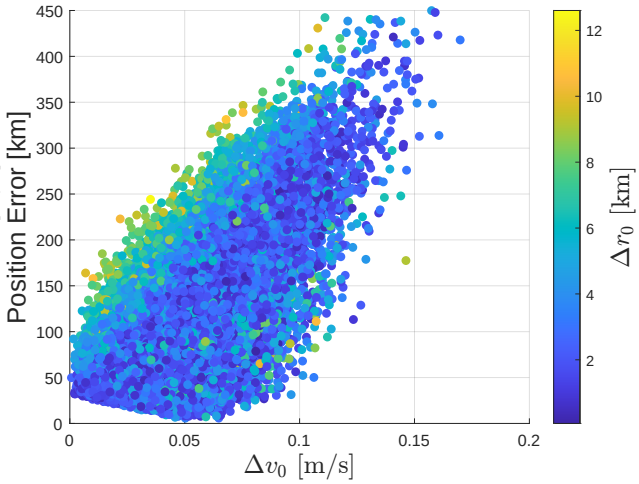

Figure 8.22: Position error as a function of the uncertainty in the initial spacecraft velocity.

Figure 8.23 displays the position error as a function of the initial velocity error components in the RSW frame. The figure shows that the radial velocity perturbation is the dominant one, as the large final errors are only obtained with large initial deviations in this component. This may be explained by the fact that, because the spacecraft is rotating with the Moon around the Earth, the radial component roughly coincides with the x-component of the BCR frame, which is the component with the largest standard deviation in its uncertainty (by a factor of 1.6).

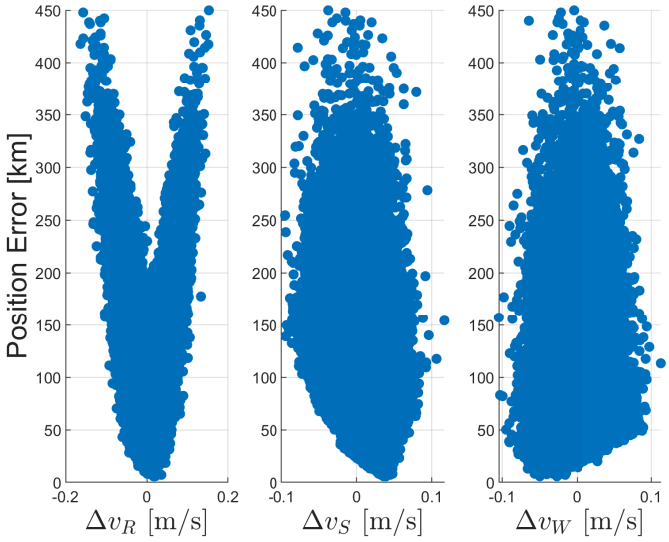

Figure 8.23: Position error as a function of the uncertainty of each velocity component in the RSW frame.

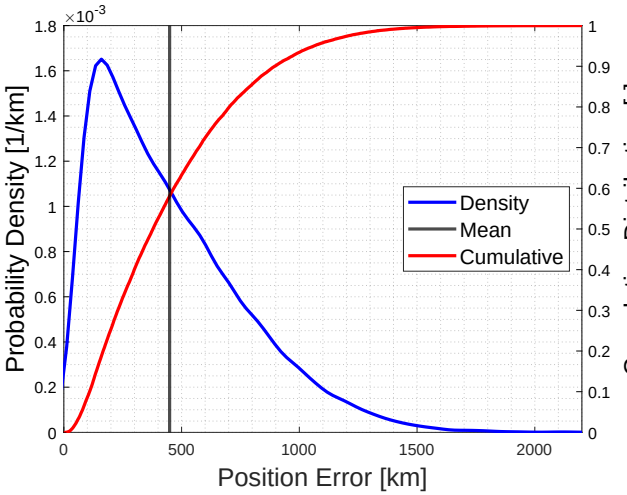

Figure 8.24: Fitted probability density function and cumulative distribution function of the position error of the Monte Carlo analysis including mass and thrust uncertainties. The vertical black line corresponds to the mean value.

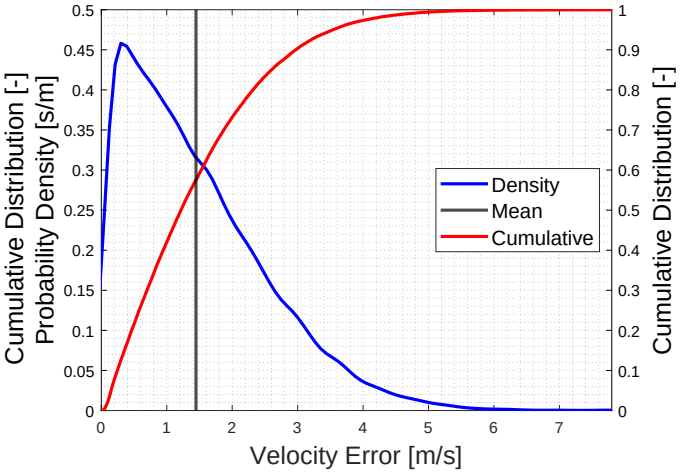

Figure 8.25: Fitted probability density function and cumulative distribution function of the velocity error of the Monte Carlo analysis including mass and thrust uncertainties. The vertical black line corresponds to the mean value.

8.3.2 Sensitivities Including Mass and Thrust

The new fitted probability density functions of the position and velocity errors that include the mass and thrust uncertainties can be found in Figure 8.24 and Figure 8.25, respectively. Even though the shapes of the probability density functions still resemble log-normal curves, the position error mean is roughly five times larger than when not including the mass and the thrust uncertainties. The velocity error mean is seven times as large. Nevertheless, more than 90% of the trajectories have a position error lower than the relaxed constraints ($\sim$1150 km), and the same goes for the velocity error. As such, even though the mass and thrust uncertainties have a significantly stronger impact on the error,

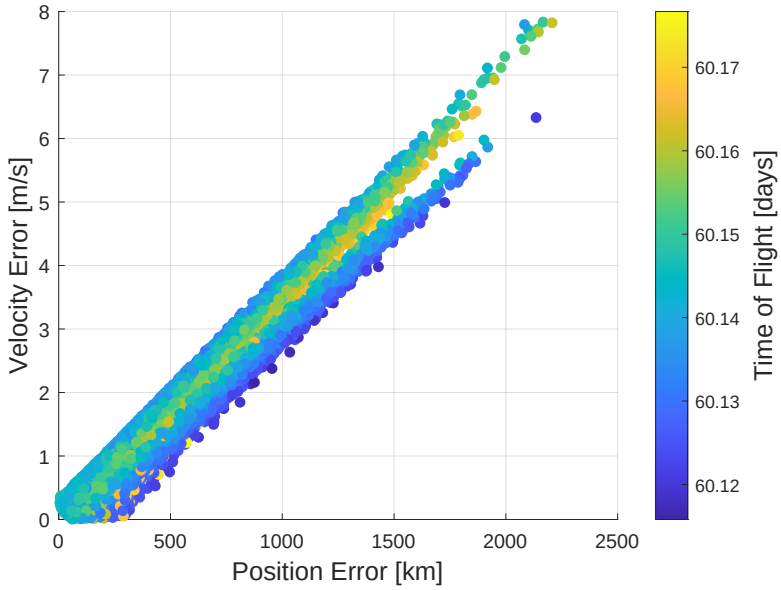

Figure 8.26: Scatter plot of the position and velocity errors of the Monte Carlo analysis including mass and thrust uncertainties.

the thrusting arcs still remain close to the target trajectory even when no correction maneuvers are applied.

Figure 8.26 scatters the position and velocity errors of all trajectory evaluations, as well as the TOF. A clear linear relationship can be identified between the errors, resembling that of Figure 6.23. This can be explained by the fact that to compute the closest state, the function to be minimized is a linear function of the position and velocity errors according to Equation (5.18). As such, when the trajectories are relatively close to the target manifold, the minima will tend to follow a linear relationship. Moreover, in this case, the position and velocity errors were weighted equally. When performing an extra run focusing more strongly on the position error, the resulting linear relationship attains a larger slope, as would be expected (the position errors tend to be lower). As there is a linear trend between the errors, it can be concluded that roughly 90% of the trajectories satisfy both the position and velocity constraints at the same time (from the data presented in the histograms). From the figure, it can be seen that the spread in TOF is rather small, however, the spread in propellant mass is in the order of several kilograms, unlike in the Monte Carlo analysis with only initial position and velocity uncertainties. Hence, in this case, many trajectories connect to a different point of the stable manifold phase, exchanging coasting time in the manifold with thrusting time in the optimal control phase. Nonetheless, this variation in propellant mass considers the case where no correction maneuvers are performed, so it should not be regarded as the uncertainty in the transfer's required propellant mass.

Figures 8.27 and 8.28 show the influence of the thrust and mass uncertainties on the position error. Due to the roughly linear relationship between the errors, the plots for the velocity error look extremely similar. These figures prove the thrust uncertainty to be the dominant one, since when the thrust deviation is zero, the maximum error is only $\sim$900 km and there are recognizable linear lower and upper bounds when ΔT is non-zero. An extra run excluding this uncertainty shows that the initial mass uncertainty has a more severe impact than the initial position or velocity deviations. From the equations of motion, one can observe that the influence of the mass and the thrust should be similar, as doubling the thrust would have the same effect as halving the spacecraft mass in terms of instantaneous acceleration. However, the thrust has more impact as increasing the thrust also increases the mass flow, leading to larger accelerations over time. Nevertheless, the large difference

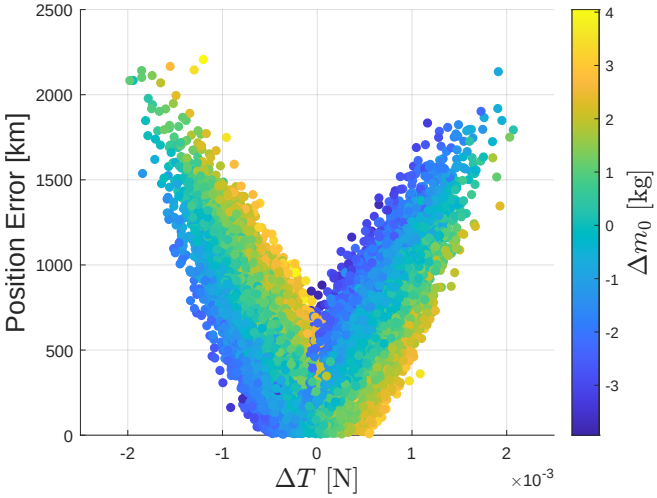

Figure 8.27: Position error as a function of the uncertainty in the spacecraft thrust.

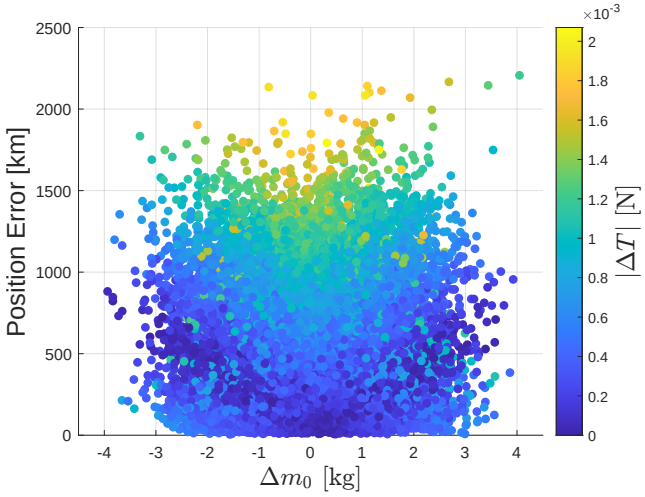

Figure 8.28: Position error as a function of the uncertainty in the spacecraft mass.

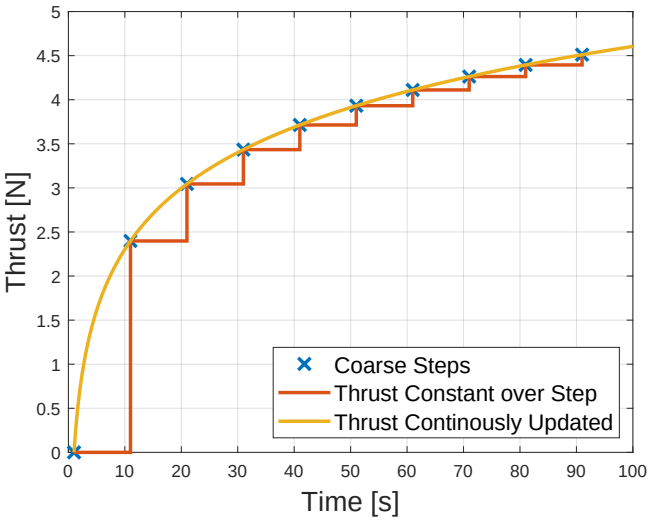

Figure 8.29: Discretization of a generic thrust profile with a coarse step size.

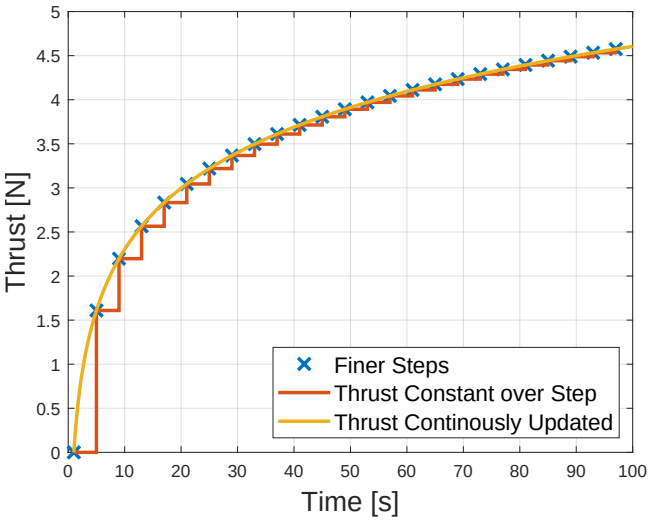

Figure 8.30: Discretization of a generic thrust profile with a finer step size.

in sensitivity can be attributed to the fact that the standard deviation of the thrust is 2.5 larger than the one of the mass. The inverse relationship between the influences of the mass and the thrust can be recognized in the color gradients of Figure 8.27, where, for an increase in thrust, an increase in spacecraft mass is beneficial to lower the position error and vice versa. The color gradient in Figure 8.28 is as expected when the leading factor is the thrust deviation.

Overall, these Monte Carlo results suggest that the trajectories are controllable and not excessively sensitive for a real mission, even when flybys take place highly close to the Moon. For a real mission, a more thorough analysis ought to be performed in a high-fidelity model to determine the reserve propellant mass required for the transfer, depending on the GNC capabilities.

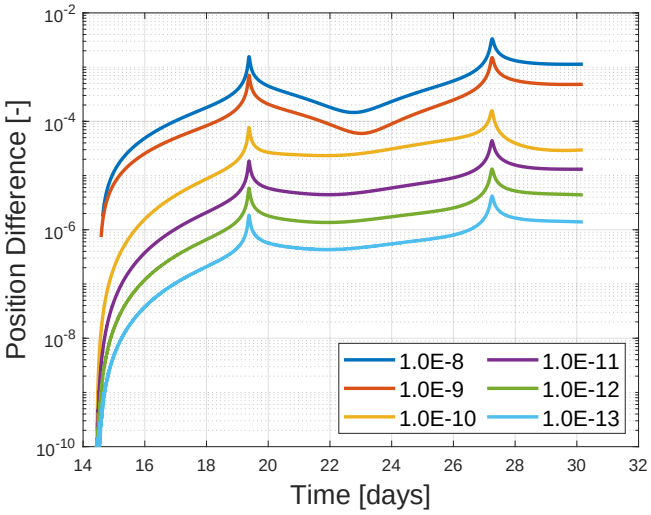

Figure 8.31: Difference in position as a function of time for several sampling step sizes (obtained through different integrator tolerances) using the nominal trajectory.

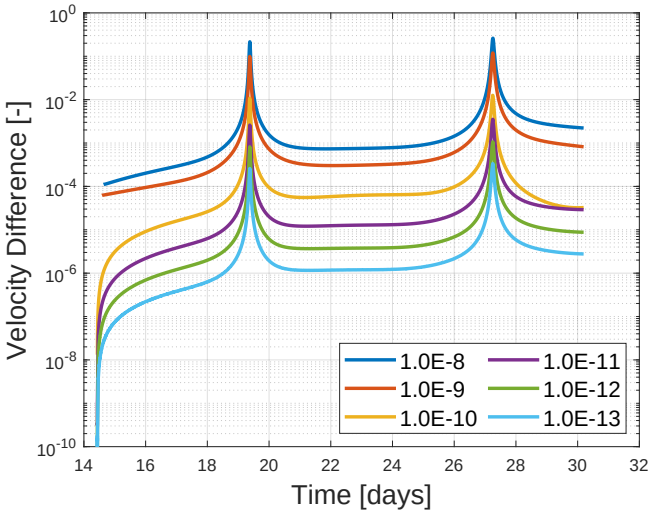

Figure 8.32: Difference in velocity as a function of time for several sampling step sizes (obtained through different integrator tolerances) using the nominal trajectory.

8.3.3 Sampling Sensitivity

The thrust profiles are generally provided in the RSW frame (as their pointing is usually fixed to the spacecraft attitude) in steps of constant attitude. However, the indirect method provides continuous thrusting functions (the "ground truth") that the integrator uses to solve the EOM by approximating said thrusting functions. When transitioning this continuous thrust profile to a step-wise constant profile, a different trajectory is expected to be obtained as the thrust profile changes. The sampling frequency plays a role since the difference with respect to the original thrust profile is reduced the smaller the step size. This can be visualized in Figure 8.29 and Figure 8.30, where a generic thrust profile is sampled with two different frequencies. As Figure 8.30 has a higher sampling frequency, the profile is closer to the continuous one. As the frequency tends to infinity, the step-wise constant profile converges to the original.

To analyze the effect of sampling in the nominal trajectory, Figure 8.31 and Figure 8.32 were generated. These figures display the difference in position and velocity over time with respect to the trajectory obtained with a tolerance one order of magnitude tighter (the relative and the absolute tolerances adopt the same value). This was computed for a variety of tolerances (reducing the tolerance increases the sampling frequency) using the *Dormand Prince* integrator. The strategy to generate the trajectories was to interpolate a given thrust strategy and keep the thrust constant over each integration step (also for the computation of the sub-steps[5].).

Even though these figures were generated by keeping the thrust step-wise constant in the BCR frame instead of the RSW, the analysis is accurate enough to prove that the most critical regions are the flybys, especially regarding the differences in velocity (Figure 8.32). In these points, there are singularities, introducing uncertainty in the resulting differences since to compute the difference the trajectories are compared to others that show the same singularities. For reference, the OD accuracy level is on the order of 10^{-5} in both position and velocity, something only achievable in terms of position difference, requiring integrator tolerances of 10^{-12}.

[5]The thrust is only updated once a new step is accepted. This is achieved through the integrator *step handler*.

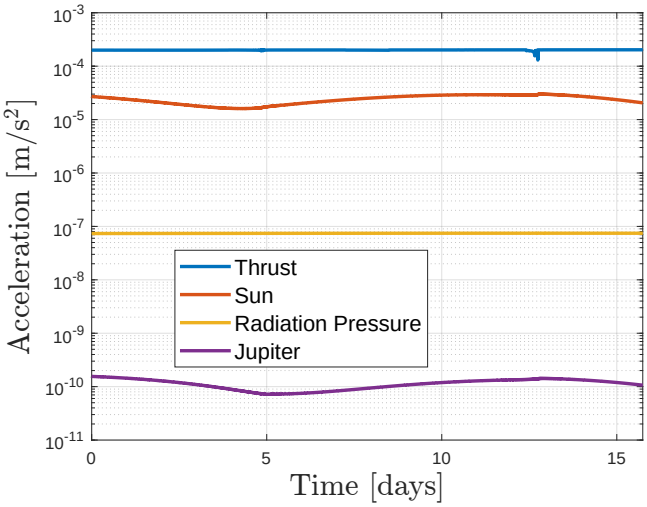

Figure 8.33: Magnitude as a function of time of the main disturbing accelerations for the optimal control phase of the nominal trajectory.

Further analysis is of course required, but this sampling sensitivity is something for the GNC subsystem to take into account. Nevertheless, a practical way of solving this issue would be to refine the trajectory in ASTOS forcing a step-wise constant thrust profile instead of a linearly interpolated one. A different trajectory will be obtained (maybe also in terms of performance), but the trajectory will be feasible.

8.4 Unmodeled Disturbances

So far, all analyses have been performed within the context of the CR3BP. Although the trajectory sensitivities were explored, for the optimized transfers to be useful for a real mission, they ought to be transitioned to a high-fidelity model. As explained by Frueh et al. (2021), the main unmodelled accelerations by the CR3BP within the cislunar space, excluding the eccentricity of the Moon's orbit, are the Sun's and Jupiter's gravitational pulls, and the solar radiation pressure. For the trajectory from Subsection 8.2.2, these disturbances are plotted as a function of time in Figure 8.33. The thrust acceleration is also included for reference. These acceleration profiles were obtained using the ASTOS software, employing a cannonball model for the solar radiation pressure with a diffuse reflection parameter of 0.8 and an area-to-mass ratio of 0.02 m^2/kg (Frueh et al., 2021). The largest perturbation is generated by the Sun, followed by the solar radiation pressure two orders of magnitude below, with Jupiter's disturbance being the lowest, five orders of magnitude below the Sun's. Due to Jupiter's perturbation being so low, it is predicted that any other unmodelled accelerations would have a negligible effect, at least for the purposes of this analysis. Because of the two flybys passing considerably close to the Moon, the J_2 effect of the Moon was also examined, however introducing no perceivable modifications to the trajectory. As the figure evidences, the thrust acceleration for this use case is one order of magnitude larger than the Sun's perturbation. Hence, in cases where the thrust acceleration is lower, correcting for the Sun's third body perturbation can become an arduous task.

The thrust profile of the trajectory corrected for the perturbations in Figure 8.33 is presented in Figure 8.34. For this task, direct collocation in CAMTOS was again used, showing the superior versatility of direct methods to adapt to different dynamic systems compared to indirect methods. The main differences with respect to Figure 8.15 lie in the thrust magnitude profile. The thrust magnitude in this new dynamic model is slightly lower in general, entailing a new required propellant mass of 6.81 kg, 100 g lower than the original (6.91 kg). Thus, for this transfer, the perturbations help the spacecraft reach its final destination more easily, requiring less propellant for the same TOF.

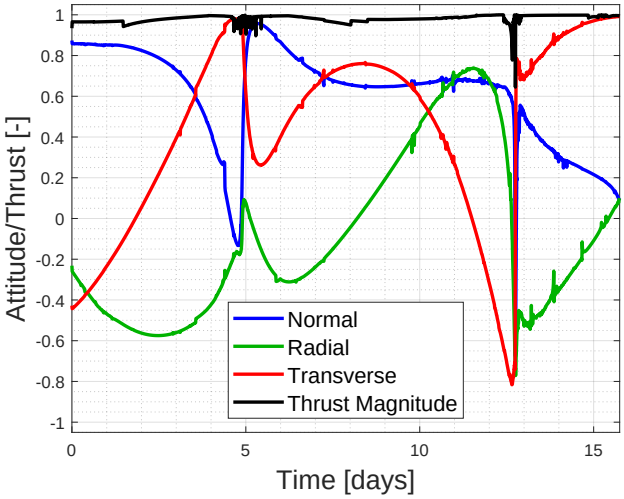

Figure 8.34: Attitude unit vector components and thrust profile as a function of time of the trajectory from Subsection 8.2.2 transitioned to a higher-fidelity model using the ASTOS software.

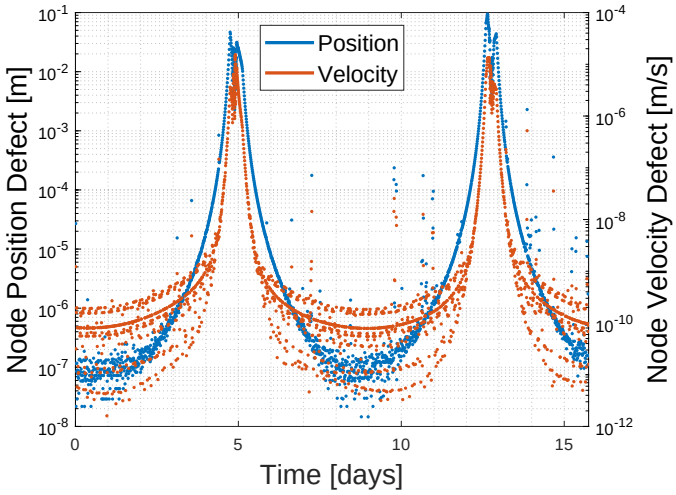

Figure 8.35: Integration position and velocity defects at each collocation node of the trajectory from Subsection 8.2.2 transitioned to a higher-fidelity model using the ASTOS software.

Nevertheless, the profile, although useful, is not as smooth as one would expect for an optimal transfer, hinting that the trajectory could be optimized further. Regarding the node defects in Figure 8.35, the magnitude and behavior resemble those of Figure 8.16, rendering the transfer feasible.

Then, to fully transition this transfer to a high-fidelity model, a more accurate ephemeris model ought to be used for the Moon, since the CR3BP assumes a perfectly circular orbit, whereas the real Moon's orbit has an average eccentricity of 0.0549. This transition to a full-ephemeris model is more demanding than the other acceleration corrections because, at a given Julian Date, the Moon is located somewhere else, making the states of the initial guess inappropriate. This is something especially problematic for the flyby segments (when the spacecraft is closest to the Moon). This task is considered to be out of the scope of this research due to time constraints. Moreover, for transfers between halo orbits, this issue would be an obstacle to overcome regardless of the method employed to generate the initial guesses (the main goal of this research), as these orbits are only purely periodic within the CR3BP. Anyhow, for a similar use case, Pritchett et al. (2017) successfully transitioned an optimized transfer to a full-ephemeris model, proving that it is certainly possible. Moreover, the same was accomplished by Kayama et al. (2021) for a different transfer between halo orbits.

To conclude, this chapter proved the obtained trajectories to be verified and extremely promising initial guesses for direct collocation methods, as they were refined in ASTOS with negligible changes in performance and without excessive effort. Furthermore, although the trajectories seem sensitive to thrust sampling, they remain close to the target phases when including uncertainties in initial state and thrust magnitude, with the latter being the most critical. Lastly, taking advantage of collocation methods, the trajectories can account for most of the unmodelled accelerations without major changes, and literature shows that they can be transitioned to a full-ephemeris model if more time is invested.

Conclusions and Recommendations 9

In this chapter, the main conclusions of this research work are presented, with a focus on answering the main research question and the sub-questions formulated in Chapter 1. This discussion is provided in Section 9.1. Then, Section 9.2 supplies an overview of recommendations and potential investigations to expand this research.

9.1 Conclusions and Answer to Research Questions

The main question of this investigation work was the following:

Main Research Question

What would be a suitable approach to obtain a variety of low-propellant-mass low-thrust transfers between periodic solutions in the cislunar space as a function of the TOF?

After comparing numerous methods to compute low-thrust transfers between periodic solutions within the dynamically complex CR3BP, such as halo orbits, it was concluded that an indirect method, which belongs to the branch of optimal control, is the most suited. The nature of the indirect method ensures the performance of the obtained solutions is competitive, and its low number of required optimizable parameters makes the technique suitable for heuristic optimization, increasing the odds of finding globally optimal solutions. Moreover, a multi-objective optimization was successfully carried out to minimize both the propellant mass and the time of flight of the transfer, obtaining a wide range of Pareto-optimal trajectories for a given use case between two halo orbits, which was the main research goal, as common low-thrust optimization methods such as direct collocation, provide a single trajectory as the solution.

The implementation of the optimization algorithm was made sure to adapt to the architectural design of the *Orbit Generator Tool*, living up to its user-friendliness and generality requirements, and making use of readily-available phases such as the periodic orbit or manifold phases. The integrator analysis proved the obtained trajectories to have an accuracy (within the CR3BP) in the order of the current orbit determination capabilities. Then, the design space exploration demonstrated that the optimization problem is highly non-linear and difficult, making the fulfillment of the demanding boundary constraints a challenge for heuristic optimization. The best-performing algorithm for the proposed use case turned out to be differential evolution, namely, the MOEA/D optimizer implemented in the Pagmo library. To deal with the constraints, a multi-objective approach was found to be the best, in which a weighted sum of the constraints was added as a third objective, with the propellant mass and the TOF being the other two. The tuning process revealed that the relative weighting of the constraints is not critical as long as there is a focus on the position constraint over the rest. The same cannot be said for the weighting of the three objectives, as to maximize the obtainment of feasible trajectories the constraints objective shall have a weight orders of magnitude larger than the other two. In addition, the relative weighting of the propellant mass and TOF leads the optimizer to explore different areas of the Pareto front. This relative weight is thus provided as a user-definable parameter. Furthermore, the optimization topology also proved to play a large role in the quality of the results, as with inappropriate settings the optimizer would tend to converge to a small set of feasible solutions. The outcome of the tuning process deemed ten islands positioned in a ring configuration with migrations of the best solutions every five generations as a suitable architecture to obtain promising results.

The explained strategy in the proposed use case found a wide range of trajectories ranging from 20 to 80 days, and from 2 to 8.5 kilograms of propellant mass, providing the mission designer with 100 Pareto-optimal trajectories to choose from. Moreover, the quality of the solutions is outstanding, rendering the indirect method combined with differential evolution heuristic optimization as a promising approach to answer the main research question.

The conclusions and accomplishments of this research can be understood more deeply by addressing the sub-questions:

Sub-Question 1

To what extent is the outcome of the implemented method verified?

Thorough unit and system testing were carried out on the implementation to ensure that not only does the code work for the analyzed use case but also for any other transfer between halo orbits, with differing numbers of phases and spacecraft specifications. Furthermore, the trajectories themselves were verified via analysis. An appropriate thrusting strategy was ensured by examining the optimal control switching function, whereas the coasting equations of motion and integrator settings were verified by analysis of the mass and the Jacobi constant as a function of time. In addition, the investigation of the trajectory error on optimized transfers proved the integrator settings to be suitable over the thrusting phases as well. Lastly, the trajectories were successfully replicated employing the extensively verified ASTOS software, verifying the equations of motion during the thrusting periods.

Sub-Question 2

What is the quality of the solutions obtained?

For the use case under study, most of the feasible trajectories that were found displayed a very competitive performance compared to the trajectory obtained by Pritchett et al. (2017) in a similar case. Moreover, the Pareto-optimal solutions can outperform said trajectory by roughly 30% in both objectives, proving the superiority of indirect methods with heuristic optimization over direct methods with local optimization in terms of solution performance. The proposed approach also provided a wide variety of low-propellant mass trajectories as a function of the time of flight thanks to multi-objective optimization, again outperforming conventional methods, which only provide one trajectory highly dependent on the initial guess. As such, from a mission design standpoint, the implementation not only provides better solutions but also wider design freedom to suit differing mission requirements.

Sub-Question 3

What is the performance of the solutions when used as initial guess for direct collocation methods?

As the solutions obtained with the indirect method display position and velocity discontinuities in the patch point between the optimal control and the target manifold phases, the general flow is to refine them with direct collocation methods using the ASTOS software. Several trajectories were refined this way, requiring identical times of flight and with negligibly small differences in propellant mass. As such, the trajectories are deemed to be very promising initial guesses for direct collocation methods, greatly outperforming the commonly used rough initial guesses. Hence, the combination of indirect and direct methods for trajectory design is very powerful, as the main advantages of both methods are exploited. However, the trajectories obtained by the OGT are almost final solutions, with the shape or performance barely altered by collocation methods. As such, the constraints could be relaxed even further to obtain more trajectories faster.

Sub-Question 4

How accurate is the CR3BP approximation to obtain preliminary trajectories to design real-life missions in the cislunar space?

The analysis of the results proved the trajectories to not be too sensitive to initial state deviations or parameter uncertainties, even when presenting flybys very close to the Moon, with the thrust uncertainty being the most influential. Moreover, an optimized trajectory was successfully corrected with direct collocation to include some of the main unmodelled perturbations, such as the Sun's gravity pull. What would be left to transition the trajectories to a high-fidelity model would be to account for the eccentricity of the lunar orbit, something not carried out due to time constraints and its higher difficulty. Nonetheless, other authors, such as Pritchett et al. (2017) and Kayama et al. (2021), have accomplished this task using collocation methods also starting from the CR3BP. It should be noted that transitioning the trajectories to a full ephemeris model would be an obstacle regardless of the method selected to generate the CR3BP transfers. As such, even though extra steps such as the GNC algorithm would still be required to finalize the trajectory design for a real mission, the obtained initial guesses in combination with direct collocation can be considered useful for mission design.

Sub-Question 5

To what extent can the use of manifold theory improve the obtained solutions?

The main advantage of manifold theory is that it provides a wider and probably more promising set of initial and final points for the optimal control phase to connect. Since the main difficulty of the method is finding trajectories that satisfy the demanding boundary conditions, the inclusion of manifold phases helped the optimizer find a larger number of feasible trajectories, thus exploring the design domain better. Moreover, although the Pareto-optimal trajectories obtained with manifold theory were in general slower, they greatly outperformed direct transfers in terms of propellant mass, thus "unlocking" new areas of the Pareto front.

Some more general conclusions can be drawn from this research work. Firstly, the Taguchi method can be employed in lieu of Monte Carlo simulations to analyze computationally demanding problems with a fewer number of evaluations, allowing more runs to be carried out. However, for highly interacting and non-linear problems, the Monte Carlo method would be more suitable for identifying trends, acknowledging that the main advantage of the Taguchi method, the calculation of the percentage of contribution of the factors and interactions, does not provide very useful results in this kind of problem. On a separate note, most of the research is focused on dealing with an optimization problem exhibiting a highly non-smooth and narrow feasible domain. Analysis showed that a multi-objective approach that considers the constraint violations as a separate objective can help the optimizer find a wider set of solutions than the commonly used penalty functions. In addition, an essential point is to find an adequate balance between focusing on exploiting the best solutions and preserving diversity in the population. This prevents the optimizer from converging to a sub-set of solutions, while still finding feasible transfers. The aforementioned balance can be obtained via the parallelization of an appropriate number of islands that exchange the most promising trajectories between them. Lastly, even though indirect methods are sometimes used for trajectory optimization, they are rarely combined with heuristic methods, which greatly help in overcoming the problem of the small radius of convergence of these methods and allow for multi-objective optimization.

Overall, the general quality of the results obtained with the proposed approach is more than satisfactory since, even though tuning the optimizations can be a challenging task, a great variety of Pareto-optimal solutions were obtained, all of which with outstanding performance. Moreover, all

research questions were answered, thus contributing to filling the scientific gap, and valuable insight was obtained on the suitability of combining heuristic optimization with direct and indirect methods for mission design within the cislunar space.

9.2 Future Work Recommendations

Even though the results and conclusions are encouraging, some aspects will benefit from additional research. Moreover, the tool can still be enhanced to ease mission design even further.

Firstly, the obtained solution set does not exhibit the characteristic smooth curvature of general optimal Pareto fronts, hence, an investigation to find the reasons for the shape exhibited would be interesting. The optimization performance could in principle be improved with a more exhaustive tuning of the specific MOEA/D parameters or the optimization architecture, also considering constraint-handling strategies that modify the optimization logic, such as stochastic ranking, which has shown to work well with indirect methods (Lee et al., 2005). Additionally, it would be interesting to assess if the use of hybrid optimization to more easily find feasible transfers or a Tabu search to avoid converging to suboptimal trajectories could lead to better results.

Secondly, the optimization tuning process was carried out considering a single use case. Thus, a sensible next step would be to evaluate the quality of the optimizations with different spacecraft specifications and between other halo orbits, or even between different families of periodic solutions such as distant retrograde or Lyapunov orbits. This exercise can provide valuable insight into the limitations of the implementation and into the optimization settings that provide better solutions in general.

Thirdly, regarding the trajectory refinement with collocation methods, it would be desirable to assess if the propellant mass could be reduced even more by focusing on the minimization of this performance parameter, instead of simply aiming to obtain a feasible transfer. In addition, a lot of time is spent on obtaining the Pareto-front with the OGT. It would be desirable to study how much the constraints could be relaxed to obtain transfers that can be successfully refined with ASTOS without excessive effort or change in performance. Moreover, the trajectories are still yet to be transitioned to a full ephemeris model that accounts for the eccentricity of the Moon's orbit, for which a robust strategy that works for any trajectory obtained with the OGT would be desirable.

Then, the implementation is limited to the Earth-Moon (EM) CR3BP. It would be beneficial to include transfers in other systems, such as the Sun-Earth (SE) system, and even transfers between periodic solutions within different systems. For instance, a possible use case could start from an EM halo orbit and finish in one within the SE system.

Lastly, for the sake of user-friendliness, the OGT could be included as part of the ASTOS software, with an appropriate GUI. This would be especially convenient to automatize the refinement of a selected trajectory with direct collocation, and also its transitioning to a high-fidelity model.

References

Acciarini, G., D. Izzo, and E. Mooij (July 2020). "MHACO: a multi-objective hypervolume-based ant colony optimizer for space trajectory optimization". In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. Paper No. 9185694. Glasgow, United Kingdom.

Ahlborn, M. (2023). "Comparison of Global Optimization Methods for Cis-Lunar Transfer Trajectories". BSc Thesis. University of Giessen.

Aziz, J.D. (2018). "Low-Thrust Many-Revolution Trajectory Optimization". PhD thesis. University of Colorado.

Biscani, F. and D. Izzo (2020). "A parallel global multiobjective framework for optimization: pagmo". In: *Journal of Open Source Software* Vol. 5, No. 53, 2338.

Bryson, A.E. and Y.C. Ho (1975). *Applied Optimal Control: Optimization, Estimation, and Control*. Routledge.

Bucci, L. and M. Lavagna (Mar. 2016). "Coupled Dynamics of Large Space Structures in Lagrangian Points". In: *6th International Conference on Astrodynamics Tools and Techniques*. Darmstadt, Germany.

Büskens, C. and D. Wassel (2013). "The ESA NLP solver WORHP". In: *Modeling and optimization in space engineering* Vol. 73, pp. 85–110.

Cai, H. (2013). "Reliability of analog-to-digital Sigma-Delta converters". PhD thesis. Télécom ParisTech.

Carnelli, I., B. Dachwald, and M. Vasile (2009). "Evolutionary neurocontrol: A novel method for low-thrust gravity-assist trajectory optimization". In: *Journal of guidance, control, and dynamics* Vol. 32, No. 2, pp. 616–625.

Cline, D. (2017). *Variational principles in classical mechanics*. Rochester, New York: University of Rochester River Campus Librarie.

Coello Coello, C.A. (2002). "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art". In: *Computer methods in applied mechanics and engineering* Vol. 191, No. 11-12, pp. 1245–1287.

Coverstone-Carroll, V., J.W. Hartmann, and W.J. Mason (2000). "Optimal multi-objective low-thrust spacecraft trajectories". In: *Computer methods in applied mechanics and engineering* Vol. 186, No. 2-4, pp. 387–402.

Dachwald, B. (2004). "Low-Thrust Trajectory Optimization and Interplanetary Mission Analysis Using Evolutionary Neurocontrol". DEng Thesis. Institut für Raumfahrttechnik, Universität der Bundeswehr, München.

Davis, D., S. Bhatt, K.C. Howell, J. Jang, R. Whitley, F. Clark, D. Guzzetti, E. Zimovan, and G. Barton (Feb. 2017). "Orbit maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits". In: *AAS/AIAA Space Flight Mechanics Meeting*. Paper No. 269. San Antonio, Texas.

De Pascale, P. and M. Vasile (2006). "Preliminary design of low-thrust multiple gravity-assist trajectories". In: *Journal of Spacecraft and Rockets* Vol. 43, No. 5, pp. 1065–1076.

Eslinger, J. (2023). "Development and Implementation of Station Keeping Approaches in Halo Orbits". MSc Thesis. University of Stuttgart.

Falck, R.D., W.K. Sjauw, and D.A. Smith (July 2014). "Comparison of low-thrust control laws for applications in planetocentric space". In: *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. Paper No. 3714. Cleveland, Ohio.

Forestryani, V., N. Rosyadi, and M. Ahsan (2022). "Multi-response optimization of dielectric fluid mixture in EDM using grey relational analysis (GRA) in Taguchi method". In: *BAREKENG: Jurnal Ilmu Matematika dan Terapan* Vol. 16, No. 3, pp. 949–960.

Fritsch, F.N. and R.E. Carlson (1980). "Monotone Piecewise Cubic Interpolation". In: *SIAM Journal on Numerical Analysis* Vol. 17, No. 2, pp. 238–246.

Frueh, C., K.C. Howell, K.J. DeMars, and S. Bhadauria (Feb. 2021). "Cislunar space situational awareness". In: *31st AIAA/AAS Space Flight Mechanics Meeting*. Paper No. 290. Charlotte, North Carolina.

Gao, Y. (2007). "Near-optimal very low-thrust earth-orbit transfers and guidance schemes". In: *Journal of guidance, control, and dynamics* Vol. 30, No. 2, pp. 529–539.

García Yárnoz, D., J.P. Sánchez, and C. McInnes (Apr. 2013). "Easily Retrievable Objects among the NEO Population". In: *Celestial Mechanics and Dynamical Astronomy* Vol. 116, No. 4, pp. 367–388.

Gath, P.F. (2002). "CAMTOS - A software suite combining direct and indirect trajectory optimization methods". DEng Thesis. University of Stuttgart.

Gill, P.E., W. Murray, and M.A. Saunders (2005). "SNOPT: An SQP algorithm for large-scale constrained optimization". In: *SIAM Review* Vol. 47, No. 1, pp. 99–131.

Gómez, G., A. Jorba, J. Masdemont, and C. Simó (1991). *Study Refinement of Semi-analytical Halo Orbit Theory*. Contract Report. ESOC Contract No.: 8625/89/D/MD (SC). European Space Agency.

Gondelach, D.J. and R. Noomen (Aug. 2015). "Analytical low-thrust transfer design based on velocity hodograph". In: *AAS/AIAA Astrodynamics Specialist Conference*. Paper No. 594. Vail, Colorado.

Hatten, N.A. (2012). "A Critical Evaluation of Modern Low-Thrust, Feedback-Driven Spacecraft Control Laws". MSc Thesis. University of Texas at Austin.

Henon, M. (1982). “On the numerical computation of Poincaré maps”. In: *Physica D: Nonlinear Phenomena* Vol. 5, No. 2, pp. 412–414.

Hintz, G.R. (2008). “Survey of orbit element sets”. In: *Journal of guidance, control, and dynamics* Vol. 31, No. 3, pp. 785–790.

Hiraiwa, N., M. Bando, and S. Hokamoto (Feb. 2022). “Halo-to-Halo Low-Thrust Transfer via Successive Convex Optimization with Intermediate Orbit Design”. In: *33rd International Symposium on Space Technology and Science*. Paper No. 48. Oita, Japan.

Hudson, J.S. and D.J. Scheeres (2009). “Reduction of low-thrust continuous controls for trajectory dynamics”. In: *Journal of guidance, control, and dynamics* Vol. 32, No. 3, pp. 780–787.

Jeppu, Y. (2024). *MATLAB Central File Exchange*. URL: https://www.mathworks.com/matlabcentral/fileexchange/47218-orthogonal-array (visited on 06/26/2024).

Kayama, Y., K.C. Howell, M. Bando, and S. Hokamoto (Feb. 2021). “Low-Thrust Trajectory Design with Convex Optimization for Libration Point Orbits”. In: *AAS/AIAA Space Flight Mechanics Meeting*. Paper No. 231. Charlotte, North Carolina (Virtual Event).

Kelly, M.P. (2015). *Transcription methods for trajectory optimization*. Tutorial. Cornell University, Ithaca, New York.

Kokou, P., S. Lizy-Destrez, and B. Le Bihan (Jan. 2014). “Computing an optimized trajectory between Earth and an EML2 halo orbit”. In: *AIAA Guidance, Navigation, and Control Conference*. Paper No. 0450. National Harbor, Maryland.

Koon, W.S., M.W. Lo, J.E. Marsden, and S.D. Ross (2001). “Low energy transfer to the Moon”. In: *Celestial Mechanics and Dynamical Astronomy* Vol. 81, No. 1-2, pp. 63–73.

Langemeijer, K. (2018). “Connecting hyperbolic invariant manifolds at variations of the Poincaré section orientation”. MSc Thesis. Delft University of Technology.

Lee, S., W. Fink, R. Russell, P. Von Allmen, A. Petropoulos, and R. Terrile (Sept. 2005). “Evolutionary Computing for Low Thrust Navigation”. In: *American Institute of Aeronautics and Astronautics Space Conference*. Paper No. 6835. New York.

Leung, Y.W. and Y. Wang (2001). “An orthogonal genetic algorithm with quantization for global numerical optimization”. In: *IEEE Transactions on Evolutionary Computation* Vol. 5, No. 1, pp. 41–53.

Lewis, F.L., D. Vrabie, and V.L. Syrmos (2012). *Optimal Control*. 3rd ed. John Wiley & Sons Inc.

Li, H. and Q. Zhang (2008). “Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II”. In: *IEEE transactions on evolutionary computation* Vol. 13, No. 2, pp. 284–302.

Li, Q., Y. Taoand, and F. Jiang (2022). “Orbital Stability and Invariant Manifolds on Distant Retrograde Orbits around Ganymede and Nearby Higher-Period Orbits”. In: *Aerospace* Vol. 9, No. 8, 454.

Maestrini, M., A. De Vittori, J.L. Gonzalo Gómez, C. Colombo, P. Di Lizia, J. Míguez Arenas, M. Sanjurjo Rivo, A. Diez Martín, Pau Gago Padreny, and D. Escobar Anton (Jan. 2023). “ELECTROCAM: assessing the effect of low-thrust uncertainties on orbit propagation”. In: *2nd ESA NEO and debris detection conference*. Paper No. 87. Darmstadt.

Mahdavi, A. and M.E. Shiri (2015). “An augmented Lagrangian ant colony based method for constrained optimization”. In: *Computational Optimization and Applications* Vol. 60, No. 1, pp. 263–276.

Maisch, A. (2022). “Design of trajectories between Lagrangian orbits in the 4-body problem”. MSc Thesis. University of Stuttgart.

Marsden, J.E. (1978). “Qualitative methods in bifurcation theory”. In: *Bulletin of the American mathematical society* Vol. 84, No. 6, pp. 1125–1148.

Mezura Montes, E. and C.A. Coello Coello (2011). “Constraint-handling in nature-inspired numerical optimization: past, present and future”. In: *Swarm and Evolutionary Computation* Vol. 1, No. 4, pp. 173–194.

Miele, A. (2010). “Revisit of the theorem of image trajectories in the earth-moon space”. In: *Journal of optimization theory and applications* Vol. 147, No. 3, pp. 483–490.

Mingotti, G., F. Topputo, and F. Bernelli-Zazzera (2007). “Combined Optimal Low-Thrust and Stable-Manifold Trajectories to the Earth-Moon Halo Orbits”. In: *New Trends in Astrodynamics and Applications III (AIP Conference Proceedings)* Vol. 886, No. 2710047, pp. 100–112.

Mistree, F., U. Lautenschlager, S.O. Erikstad, and J.K. Allen (1993). *Simulation reduction using the Taguchi method*. NASA Contractor Report, CR 93-4542. National Aeronautics and Space Administration.

Montgomery, D.C., G.C. Runger, and N.F. Hubele (2009). *Engineering statistics*. 5th ed. John Wiley & Sons.

Morante, D., M. Sanjurjo-Rivo, and M. Soler (2019). “Multi-objective low-thrust interplanetary trajectory optimization based on generalized logarithmic spirals”. In: *Journal of Guidance, Control, and Dynamics* Vol. 42, No. 3, pp. 476–490.

— (2021). “A survey on low-thrust trajectory optimization approaches”. In: *Aerospace* Vol. 8, No. 3, 88.

Morelli, A.C., C. Hofmann, and F. Topputo (2021). “Robust low-thrust trajectory optimization using convex programming and a homotopic approach”. In: *IEEE Transactions on Aerospace and Electronic Systems* Vol. 58, No. 3, pp. 2103–2116.

Naasz, B.J. (2002). “Classical element feedback control for spacecraft orbital maneuvers”. MSc Thesis. Virginia Tech.

Newman, C.P., J.R. Hollister, D.C. Davis, and E.M. Zimovan-Spreen (Aug. 2022). "Investigating solar radiation pressure modeling for operations in near rectilinear halo orbit". In: *AAS/AIAA Astrodynamics Specialists Conference*. Paper No. 728. Charlotte, North Carolina.

NIST/SEMATECH (2003). *e-Handbook of Statistical Methods*. URL: https://doi.org/10.18434/M32189 (visited on 05/16/2024).

Oberle, H.J. and W. Grimm (1989). *BNDSCO: a program for the numerical solution of optimal control problems*. Tech. rep. Report No. 515. Inst. für Angewandte Math. der Univ. Hamburg.

Olsder, G.J., J.W. van der Woude, J.G. Maks, and D. Jeltsema (2011). *Mathematical Systems Theory*. 4th ed. Delft, The Netherlands: VSSD.

Olympio, J. (Aug. 2008). "Algorithm for low-thrust optimal interplanetary transfers with escape and capture phases". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Paper No. 7363. Honolulu, Hawaii.

Out, I. (2017). "Formation Flying in the Sun-Earth/Moon Perturbed Restricted Three-Body Problem". MSc Thesis. Delft University of Technology.

Ozimek, M.T. and K.C. Howell (2010). "Low-thrust transfers in the Earth-Moon system, including applications to libration point orbits". In: *Journal of Guidance, Control, and Dynamics* Vol. 33, No. 2, pp. 533–549.

Pan, X. and B. Pan (2020). "Practical homotopy methods for finding the best minimum-fuel transfer in the circular restricted three-body problem". In: *IEEE Access* Vol. 8, pp. 47845–47862.

Papalambros, P.Y. and D.J. Wilde (2000). *Principles of optimal design: modeling and computation*. 2nd ed. Cambridge, United Kingdom: Cambridge University Press.

Park, R. S., W. M. Folkner, J. G. Williams, and D. H. Boggs (Feb. 2021). "The JPL Planetary and Lunar Ephemerides DE440 and DE441". In: *The Astronomical Journal* Vol. 161, No. 3, pp. 105–120.

Parrish, N.L., J.S. Parker, S.P. Hughes, and J. Heiligers (Mar. 2016). "Low-thrust transfers from distant retrograde orbits to L2 halo orbits in the Earth-Moon system". In: *6th International Conference on Astrodynamics Tools and Techniques*. Darmstadt, Germany.

Pergola, P., K. Geurts, C. Casaregola, and M. Andrenucci (2009). "Earth-Mars halo to halo low thrust manifold transfers". In: *Celestial Mechanics and Dynamical Astronomy* Vol. 105, No. 1-3, pp. 19–32.

Petropoulos, A. (Aug. 2004). "Low-thrust orbit transfers using candidate Lyapunov functions with a mechanism for coasting". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. Paper No. 5089. Providence, Rhode Island.

Pontani, M. and B.A. Conway (2010). "Particle swarm optimization applied to space trajectories". In: *Journal of Guidance, Control, and Dynamics* Vol. 33, No. 5, pp. 1429–1441.

Poulsen, N.K. (2012). *Dynamic Optimization: Optimal Control*. Kongens Lyngby, Denmark: IMM DTU.

Pritchett, R., K.C. Howell, and D. Grebow (Aug. 2017). "Low-thrust transfer design based on collocation techniques: applications in the restricted three-body problem". In: *AAS/AIAA Astrodynamics Specialist Conference*. Paper No. 626. Stevenson, Washington.

Richardson, D.L. (1980a). "Analytic construction of periodic orbits about the collinear points". In: *Celestial mechanics* Vol. 22, No. 3, pp. 241–253.

— (1980b). "Analytical construction of periodic orbits about the collinear points of the Sun-Earth system". In: *Astrodynamics conference*. Paper 79-127, 15 p., page 127.

Roy, R.K. (2010). *A primer on the Taguchi method*. 2nd ed. Southfield, Michigan: Society of manufacturing engineers.

Schlüter, M. and M. Gerdts (2009). "The oracle penalty method". In: *Journal of Global Optimization* Vol. 47, No. 2, pp. 293–325.

Scorsoglio, A., A. D'Ambrosio, L.Ghilardi, R. Furfaro, and V. Reddy (Sept. 2023). "Physics-informed orbit determination for cislunar space applications". In: *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. Maui, Hawaii.

Sedlaczek, K. and P. Eberhard (2006). "Using augmented Lagrangian particle swarm optimization for constrained problems in engineering". In: *Structural and Multidisciplinary Optimization* Vol. 32, No. 4, pp. 277–286.

Sentinella, M.R. and L. Casalino (July 2006). "Genetic algorithm and indirect method coupling for low-thrust trajectory optimization". In: *42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*. Paper No. 4468. Sacramento, California.

Siva Prasad, K., C. Srinivasa Rao, and D. Nageswara Rao (2012). "Application of design of experiments to plasma arc welding process: a review". In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* Vol. 34, pp. 75–81.

Smith, A.E., D.W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz (1997). "Penalty functions". In: *Handbook of evolutionary computation* Vol. 97, No. 1, C5.

Spreen, E., K.C. Howell, and D. Davis (May 2017). "Near Rectilinear Halo Orbits and their Application in Cis-Lunar Space". In: *3rd IAA Conference on Dynamics and Controls of Space Systems*. Paper No. 125. Moscow, Russia.

Stuart, J., M. Ozimek, and K.C. Howell (Aug. 2010). "Optimal, low-thrust, path-constrained transfers between libration point orbits using invariant manifolds". In: *AIAA/AAS Astrodynamics Specialist Conference*. Paper No. 7831. Toronto, Canada.

Taguchi, G. (1987). *System of experimental design; engineering methods to optimize quality and minimize costs*. New York: UNIPUB/Kaus International.

Taheri, E. and O. Abdelkhalik (2016). "Initial three-dimensional low-thrust trajectory design". In: *Advances in Space Research* Vol. 57, No. 3, pp. 889–903.

Tatay Sanguesa, J. (2021). "Optimal trajectories in the Earth-Moon system". MSc Thesis. Delft University of Technology.

United States Naval Observatory, Nautical Almanac Office, Great Britain, Nautical Almanac Office, Science and Engineering Research Council (Great Britain), Science Research Council (Great Britain), Rutherford Appleton Laboratory, Council for the Central Laboratory of the Research Councils (Great Britain), United States, Department of the Navy, United States, Congress, Great Britain, and Hydrographic Office (2008). *The Astronomical Almanac for the Year 2008: Data for Astronomy, Space Sciences, Geodesy, Surveying, Navigation, and Other Applications*. U.S. Government Printing Office. ISBN: 9780160773969.

Vallado, D.A. (2013). *Fundamentals of astrodynamics and applications*. 4th ed. Hawthorne, California: Space Technology Library.

Varga, G.I. and J.M. Sánchez Pérez (Mar. 2016). "Many-revolution low-thrust orbit transfer computation using equinoctial q-law including j2 and eclipse effects". In: *6th International Conference on Astrodynamics Tools and Techniques*. Paper No. 590. Darmstadt, Germany.

Vittaldev, V., E. Mooij, and M.C. Naeije (2012). "Unified State Model theory and application in Astrodynamics". In: *Celestial Mechanics and Dynamical Astronomy* Vol. 112, No. 3, pp. 253–282.

Wakker, K.F. (2015). *Fundamentals of astrodynamics*. Delft, The Netherlands: TU Delft Library.

Wall, B.J. and B.A. Conway (2009). "Shape-based approach to low-thrust rendezvous trajectory design". In: *Journal of Guidance, Control, and Dynamics* Vol. 32, No. 1, pp. 95–101.

Walther, M. (2022). "Definition and optimization of cislunar low-thrust transfers". MSc Thesis. University of Stuttgart.

Walther, M. and A. Wiegand (June 2023). "ASTOS Orbit Generator – Rapid Creation of Quasi-Optimal Orbital Transfers in the CR3BP". In: *ESA GNC-ICATT*. Paper No. 48. Sopot, Poland.

Wang, H., S. He, and X. Yao (2017). "Nadir point estimation for many-objective optimization problems based on emphasized critical regions". In: *Soft Computing* Vol. 21, No. 9, pp. 2283–2295.

Wang, Z. and M.J. Grant (2018). "Optimization of minimum-time low-thrust transfers using convex programming". In: *Journal of Spacecraft and Rockets* Vol. 55, No. 3, pp. 586–598.

Wu, D., D. Sotnikov, G. Gary Wang, E. Coatanea, M. Lyly, and T. Salmi (2023). "A Dimension Selection-Based Constrained Multi-Objective Optimization Algorithm Using a Combination of Artificial Intelligence Methods". In: *ASME Journal of Mechanical Design* Vol. 145, No. 8, 081704.

Yang, G. (2009). "Direct optimization of low-thrust many-revolution earth-orbit transfers". In: *Chinese Journal of Aeronautics* Vol. 22, No. 4, pp. 426–433.

Zhang, P., J. Li, H. Baoyin, and G. Tang (2013). "A low-thrust transfer between the Earth-Moon and Sun-Earth systems based on invariant manifolds". In: *Acta Astronautica* Vol. 91, No. 10, pp. 77–88.

Zuiani, F., Y. Kawakatsu, and M. Vasile (Feb. 2013). "Multi-objective optimisation of many-revolution, low-thrust orbit raising for destiny mission". In: *23rd AAS/AIAA Space Flight Mechanics Conference*. Paper No. 264. Kauai, Hawaii.

Additional Information: Access to Walther (2022), Maisch (2022), Eslinger (2023), and Ahlborn (2023) can be provided upon request.

Research Structure Outline A

In this chapter, the structure and schedule of this research are outlined. Section A.1 presents the *Work Flow Diagram* (WFD), whereas the *Work Breakdown Structure* (WBS) can be found in Section A.2, with an explanation of the work packages and tasks, followed by a *Gantt Chart* in Section A.3. Lastly, as the presented schedule was of course generated before carrying out the research, a reflection on how closely the planning was followed is provided in Section A.4.

A.1 Work Flow Diagram

The research, excluding the literature study, is divided into five main phases: the implementation of the optimal control phase within the OGT, the optimization of a selected use case, the evaluation of the optimization results, finalizing the report, and preparing the M.Sc. Thesis defense. Each phase consists of several work packages, each of which may be divided into a certain number of tasks. The WFD can be found in Figure A.1.

A.2 Work Breakdown Structure

The WBS is displayed in Figure A.2, with the work packages divided into tasks with allocated durations. Additionally, an explanation of the tasks can be found in Table A.1.

Table A.1: Explanation of the specific research proposal tasks.

Task ID	Explanation
1.1.1	Following the architectural design of the OGT, a class is created with the EOM of the method. A second class is developed that generates the optimal control phase and propagates the EOM.
1.1.2	The class that handles the phases and mission data is updated to include the new optimal control phase.
1.1.3	Having implemented the trajectory computation strategy, the next step is to feed this strategy to the optimization problem class, with the objectives, the design variables, and the constraints.
1.1.4	The selected trajectories must be exported with enough information for them to be plotted and replicated in ASTOS.
1.2	The feedback obtained on the literature review is implemented.
1.3.1	The Pagmo shared library is updated to include NSGA2.
1.3.2	The Pagmo shared library is updated to include NSPSO.
1.4.1	The new phase and interfaces are thoroughly tested to safely proceed.
1.4.2	The new phase and interfaces are tested for user-friendliness such that the OGT can be incorporated within ASTOS in the future.
1.4.3	Check that the implementation and method work via an easy use case optimization, in which a feasible transfer ought to be found.
1.4.4	This preliminary trajectory is used to code a test on the Jacobi constant to verify the EOM.
1.4.5	The goal is to learn to use ASTOS and replicate this preliminary trajectory with the software for verification purposes before proceeding.

2.1	Two promising use cases explored by other authors are chosen for problem analysis.
2.2	The integrator analysis, design space exploration, and optimization sections are reported.
2.3	A nominal feasible trajectory within the first use case is found to be employed in the problem analysis.
2.4.1	The necessary code to generate a benchmark trajectory and to carry out an integrator analysis with the nominal trajectory is written and tested.
2.4.2	An integrator trade-off is carried out to find, using the nominal trajectory, the fastest integrator and integrator settings that still provide a suitable accuracy level in terms of position and velocity.
2.5.1	The goal is to learn about design of experiments and the Taguchi method, and about how it should be applied to analyze the given problem.
2.5.2	Employing the Taguchi method, the setup to analyze the problem via a fractional factorial design and the ANOVA method is coded.
2.5.3	The method is run for both use cases and the output is analyzed with a focus on determining suitable design variable bounds.
2.6.1	Using MIDACO as the reference optimizer and solely focusing on the first use case from now on, optimizations with numerous constraint-handling strategies are carried out and the most promising strategy is selected.
2.6.2	With the chosen constraint-handling strategy, the use case is optimized with all available multi-objective optimizers, each of which with default settings.
2.6.3	After the best optimizer is found, the most promising constraint-handling strategies are re-evaluated. The optimizer selection may also be re-assessed depending on the outcome.
2.6.4	A tuning process is carried out on the chosen constraint-handling strategy for the proposed use case.
2.7.1	The optimization parameters (such as the population or the number of islands) are tuned to maximize the performance of the final optimizations.
2.7.2	With the tuned optimization settings, several optimizations are run with differing seeds to obtain the final Pareto front of the problem with manifolds.
3.1.1	Within the trajectory verification process, to ensure that the propagation error of the trajectories is appropriate, an optimized solution is re-propagated with tighter integrator settings, and the error in position and velocity is quantified.
3.1.2	The EOM are verified by analyzing the behavior of the spacecraft mass and the Jacobi constant over the coasting and thrusting periods.
3.1.3	The trajectory is replicated in high-fidelity space trajectory analysis software such as ASTOS to finalize the trajectory verification.
3.2	The verification, ASTOS optimization, sensitivity analysis, and manifold effect sections are reported.
3.3.1	The goal is to learn how to optimize in ASTOS using direct collocation methods, using as an initial guess a trajectory generated with the OGT.
3.3.2	The instantaneous discontinuities because of the constraint tolerances of the nominal trajectory from 2.3 are corrected with ASTOS.

3.3.3	The instantaneous discontinuities because of the constraint tolerances of a selected optimal trajectory are corrected with ASTOS for further proof that the trajectories are suitable initial guesses for direct collocation methods.
3.4.1	A Monte Carlo analysis is carried out on a chosen optimized solution accounting for the uncertainties in the spacecraft's initial state.
3.4.2	A Monte Carlo analysis is carried out on a chosen optimized solution accounting for the uncertainties in the modeling parameters.
3.5.1	Using the same optimizer and constraint-handling strategy as in 2.7, the optimization is re-tuned to find a Pareto front of trajectories without manifold phases.
3.5.2	The Pareto front without manifold theory is obtained, and the results are compared to those of 2.7.
4.1.1	The problem analysis chapter, which includes the integrator analysis and the design space exploration, is finalized.
4.1.2	The results chapter, which includes the optimization tuning and the optimizations with manifolds, is finalized.
4.2.1	The verification and validation section is finalized, including the unit tests.
4.2.2	The trajectory refinement with ASTOS section is finalized.
4.2.3	The sections on the sensitivity analysis and the optimizations without manifolds are finalized.
4.3	The original literature review chapters are updated with the decisions made throughout the research.
4.4	The conclusions and recommendations chapter is written.
4.5	The abstract and preface chapters are written.
4.6	The report is finalized by applying the feedback from the Green-Light review.
5.1	The final presentation structure and storyline are developed.
5.2	The presentation slides are made according to the previous work package.
5.3	The presentation is improved with the generation of explanatory diagrams and videos
5.4	The presentation is rehearsed for the final defense.

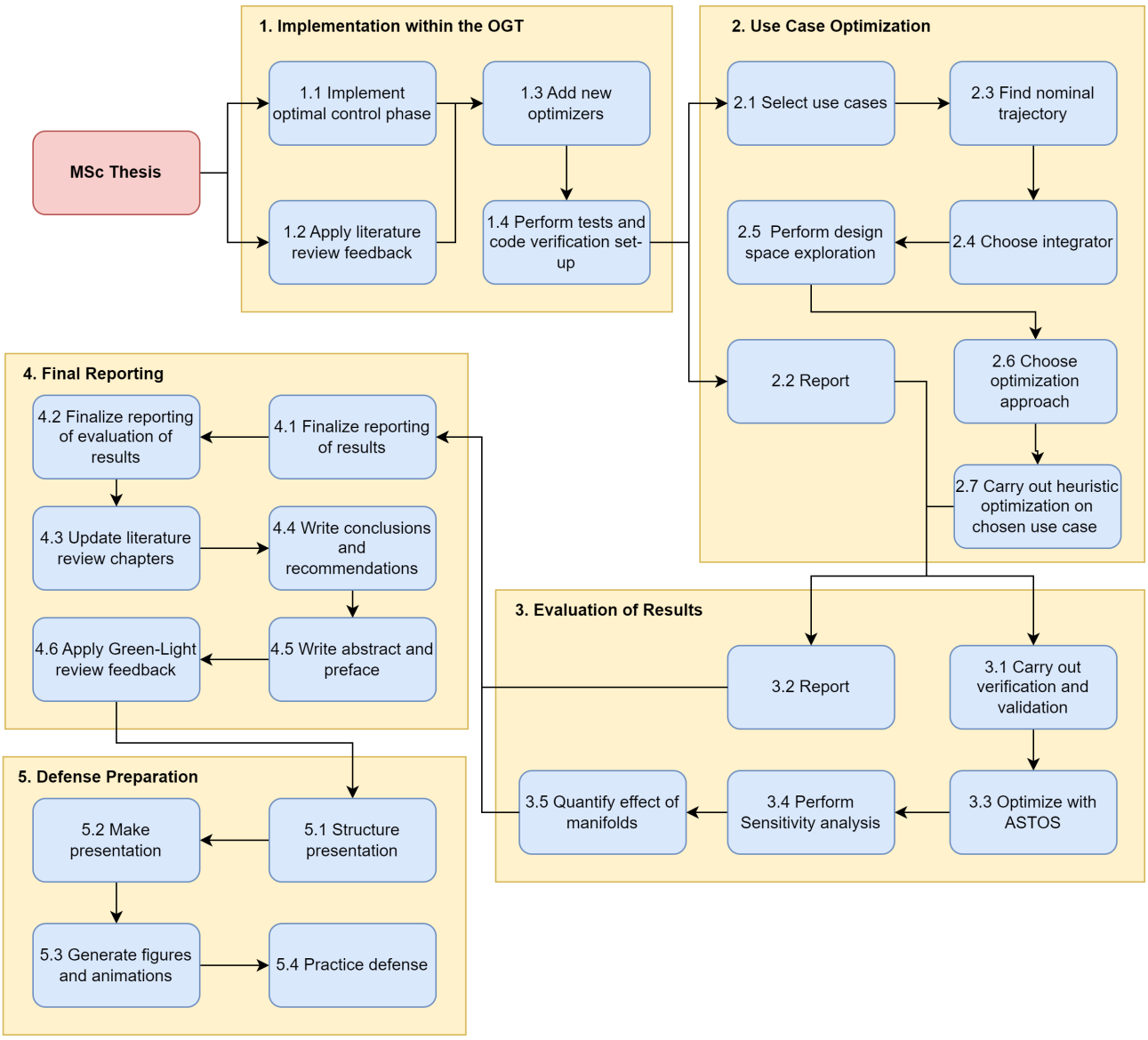

Figure A.1: Work Flow Diagram of the research.

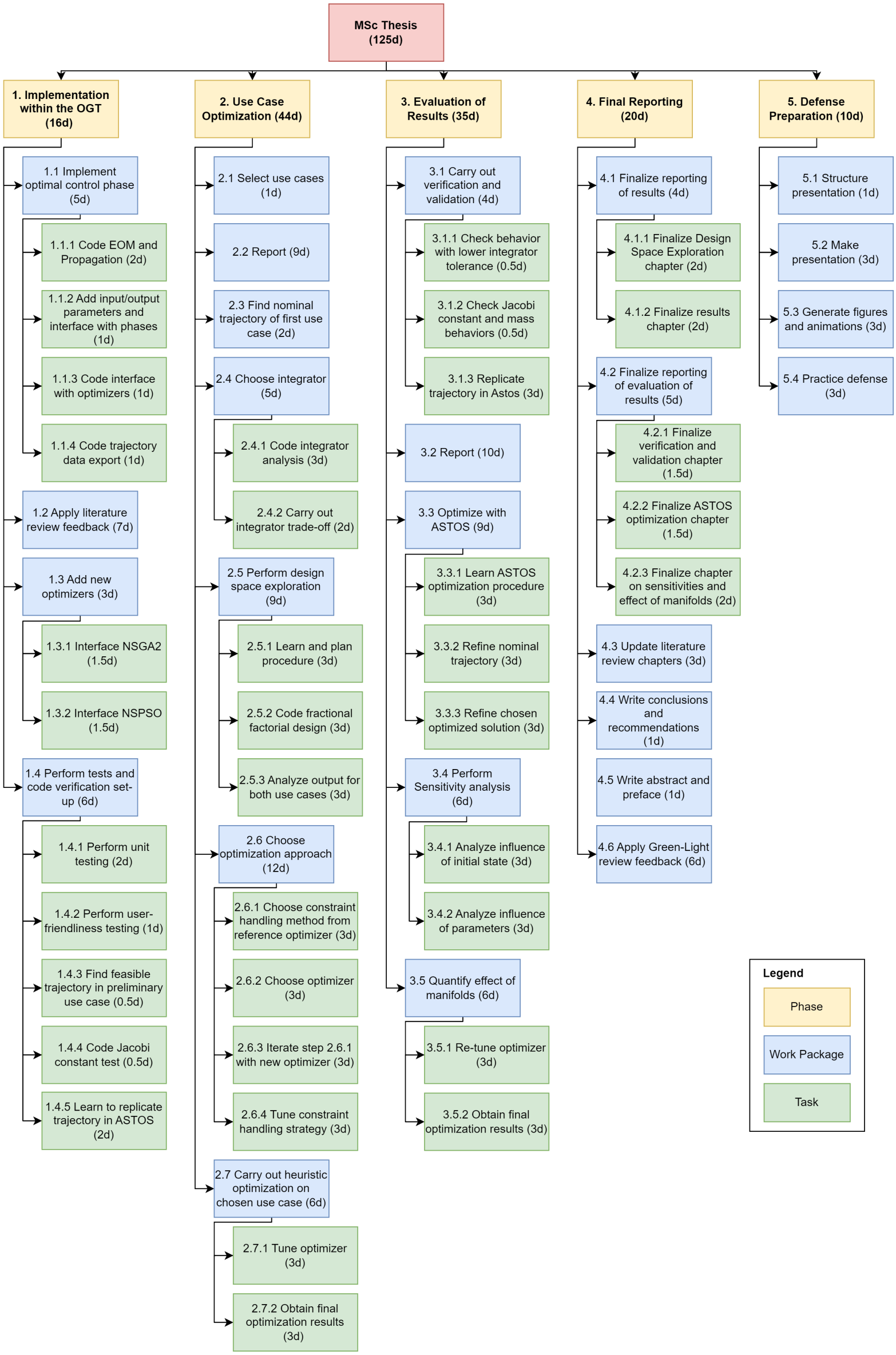

Figure A.2: Work Breakdown Structure Diagram of the research.

A.3 Gantt Chart

The Gantt Chart is displayed in Figure A.3 and Figure A.4. Some tasks, such as reporting, are done in parallel with other tasks, however, it is difficult to predict what percentage of the time will be dedicated to each task. As such, for the sake of readability and to simplify the analysis, the tasks are structured sequentially. "Day #" at the beginning of a task refers to the day of the week in which the task ought to be started.

		05/02/24								05/04/24			
		Research Start								Midterm			
Task Name	**Work (days)**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
1. Implementation within the OGT	16												
1.1 Implement opt. control phase	5												
1.1.1 Code EOM and Propagation	2												
1.1.2 Add input/output parameters and interface with phases	1	Day 3											
1.1.3 Add interface with optimizers	1	Day 4											
1.1.4 Code trajectory data export	1	Day 5											
1.2 Apply literature review feedback	2												
1.3 Add new optimizers	3		Day 3										
1.3.1 Interface NSGA2	1.5		Day 3										
1.3.1 Interface NSPSO	1.5		Day 4										
1.4 Perform tests and code verification set-up	6												
1.4.1 Perform unit testing	2												
1.4.2 Perform user-friendliness testing	1			Day 3									
1.4.3 Find feasible trajectory in preliminary use case	0.5			Day 4									
1.4.4 Code Jacobi constant test	0.5			Day 4									
1.4.5 Learn to replicate trajectory in ASTOS	2			Day 5									
2. Use Case Optimization	44				Day 2								
2.1 Select use cases	1				Day 2								
2.2 Report	9											Day 2	
2.3 Find nominal trajectory	2				Day 3								
2.4 Choose integrator	5				Day 5								
2.4.1 Code integrator analysis	3				Day 5								
2.4.2 Carry out integrator analysis with both trajectories	2					Day 3							
2.5 Perform design space exploration	9					Day 5							
2.5.1 Learn and plan procedure	3					Day 5							
2.5.2 Code fractional factorial design	3						Day 3						
2.5.3 Analyze output for both use cases	3												
2.6 Choose optimization approach	12							Day 4					
2.6.1 Choose constraint handling method with reference optimizer	3												
2.6.2 Choose optimizer	3								Day 2				
2.6.3 Iterate step 2.6.1 with new optimizer	3								Day 5				
2.6.4 Tune constraint handling strategy	3									Day 3			
2.7 Carry out heuristic optimization of chosen use case	6												
2.7.1 Tune optimizer	3												
2.7.2 Obtain final optimization results	3										Day 4		

Figure A.3: First part of the Gantt Chart of the research.

								14/06/24		28/06/24		12/07/24		26/07/24
								Draft		Green-Light		Submision		Defense
Task Name	**Work (days)**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
3. Evaluation of results	35													
3.1 Carry out verification and validation	4													
3.1.1 Check behavior with lower integrator tolerance	0.5													
3.1.2 Check Jacobi constant and mass behaviors	0.5													
3.1.3 Replicate trajectory in ASTOS	3	Day 2												
3.2 Report	10													
3.3 Optimize with ASTOS	9	Day 5												
3.3.1 Learn ASTOS optimization procedure	3	Day 5												
3.3.2 Refine nominal trajectory	3		Day 3											
3.3.3 Refine chosen optimized solution	3													
3.4 Perform sensitivity analysis	6			Day 4										
3.4.1 Analyze influence of initial state	3			Day 4										
3.4.2 Analyze influence of parameters	3				Day 2									
3.5 Quantify effect of manifolds	6				Day 5									
3.5.1 Re-tune optimizer	3				Day 5									
3.5.2 Obtain final optimization results	3					Day 3								
4. Final Reporting	20													
4.1 Finalize reporting of results	4													
4.1.1 Update design space exploration chapter	2													
4.1.2 Finalize results chapter	2								Day 3					
4.2 Finalize reporting of evaluation of results	5								Day 5					
4.2.1 Finalize verification and validation chapter	1.5													
4.2.2 Finalize ASTOS optimization chapter	1.5													
4.2.3 Finalize chapter on sensitivities and effect of manifolds	2									Day 3				
4.3 Update literature review chapters	3									Day 5				
4.4 Write conclusions and recommendations	1										Day 3			
4.5 Write abstract and preface	1										Day 4			
4.6 Apply Green-Light review feedback	6										Day 5			
5. Defense Preparation	10													
5.1 Structure presentation	1													
5.2 Make presentation	3												Day 2	
5.3 Generate figures and animations	3												Day 5	
5.4 Practice defense	3													Day 3

Figure A.4: Second part of the Gantt Chart of the research.

A.4 Reflection on the Planning

Overall, the Gantt Chart provided an accurate prediction of the tasks to be carried out and helped with keeping track of what was left to be done and how delayed the project was getting over time. However, due to the difficulty of the research project, several work packages took longer than expected, entailing roughly a 3.5-week delay relative to the planning in terms of hand-in date of the final draft. In addition, the OGT optimizations required extensive computation times, forcing the need for parallelization of the tasks. Fortunately, comfortable periods were allocated to reporting, so the delays could be alleviated, and because of the thorough code testing during the implementation process, the project was rarely delayed due to bugs in the program.

For instance, learning about design of experiments and fully understanding the ANOVA method took longer than expected, due to the large number of design variables and the high complexity of the problem (all interactions were estimated). Moreover, learning how to properly set up an optimization problem with direct collocation in ASTOS was more complicated than predicted because of the difficult scaling of the parameters. In addition, after some preliminary optimizations and analyses, the original optimization strategy was majorly restructured with the addition of the closest state search strategy, something not included in the planning that took roughly three days. Also related to coding, ensuring that the manifolds were propagated in the desired direction (interior/exterior) for all cases required the development of a new functionality and a testing tool. Lastly, the transitioning of the trajectories to a different dynamic model that included the Sun and Jupiter was not part of the planning. However, this exercise helped to gain confidence in the usefulness of the provided trajectories and elevated the quality of the analysis.

Nevertheless, the main source of delay was the OGT optimizations, especially finding an appropriate constraint-handling strategy and tuning the optimizer. A vast number of ideas and combinations were explored without promising results, suggesting the necessity to relax the constraints. The key to mitigating the delays due to the long optimization times was to start the third research phase before obtaining the final results. As such, the verification and the refinements with ASTOS were carried out in parallel with the OGT optimizations. This decision permitted the analysis of the severity of the constraints, allowing the constraint relaxation that triggered a major improvement in the results. On a separate note, one week of delay can be attributed to the fact that an additional final review was carried out at Astos Solutions GmbH, so most of phase 5 had already been completed by the time the report was handed in.

In conclusion, the thorough planning was crucial for the completion of this research without major delays. This is due to the unexpected difficulties encountered in several tasks. A key takeaway is the positive influence that parallelizing work packages can have on the outcome of a project, especially when dealing with long and complex optimizations.