Low-Thrust Trajectory Optimization to Earth Mini-Moons

A Case Study of 2006 RH120

Enrique Fernández Martín

July 27, 2023

Technische Universiteit Delft

TU Delft

Low-Thrust Trajectory Optimization to Earth Mini-Moons

A Case Study of 2006 RH120

by

Enrique Fernández Martín

in partial fulfillment to obtain the degree of Master of Science

at Delft University of Technology

Student number:	5048591
Project duration:	April, 2022 – June, 2023
	Apr. 2022 to Oct. 2022 full-time
	Mar. 2022 to Jun. 2022 part-time
Thesis supervisor:	Ir. R. Noomen, TU Delft supervisor
	Dr. D. Scheeres, CU Boulder supervisor

Cover image: artist's impression of Near Earth Object passing Earth. Image credit: ESA

TU Delft

A mi madre y mi hermana.

Preface

Completing this thesis work marks the final step of my master's program at TU Delft. Reaching this moment has been a long and rewarding journey, and I am proud it has been the outcome of my most genuine passion for space, exploring, and learning. This report gathers a thesis project that has been a roller-coaster of emotions, learnings, and life lessons, and I am honored to present it.

First of all, I want to express my heartfelt gratitude to my thesis supervisor at TU Delft, Ron Noomen. His positivity and enthusiasm have been a constant source of motivation throughout our countless meetings. His guidance and feedback have been fundamental in this project's success, and having had him as my mentor has been an absolute pleasure. I must also express my warmest thanks to Dan Scheeres, my supervisor and host at CU Boulder. I am grateful he has guided my research by constantly asking the right questions and having insightful answers to mine. Working with him has been an inspiration and a privilege. I am also indebted to Shota Takahashi from CU Boulder for introducing me to the Temporarily Captured Orbiters and allowing being frequently interrupted to address my numerous questions.

A sincere thank you goes to all my colleagues at the CSML lab in Boulder. Your helpful feedback during Astroseminars, the great daily lunch conversations, and the fantastic welcoming atmosphere you created greatly enriched my experience.

I am immensely grateful to the Europe-Colorado program for enabling and sponsoring my visit to CU Boulder. My special thanks go to Britta for selecting me for the program and always being kind and helpful. I also want to express my gratitude to the various forms of financial support that made my stay in Boulder possible: Dan Scheeres for sponsoring my program extension, The Holland Scholarship, the Van der Maas fund, and the Vrijvrouwe van Renswoude fund.

I want to take the opportunity to thank the people who have supported me personally. First, big thanks to Mireia and Marc for being the best possible companions in Delft and being a constant in my life for several years now. Thanks to Burak for being almost my daily life advisor on every possible matter. Special thanks to my Boulder family, Erica, Arnau, Jaume, and Jules. You have made my stay in Boulder an unforgettable experience. And last but not least, thanks to my friends Marc, Joan, and Cris, who traveled across the world to visit me in Colorado. I would certainly not get by without a little help from my friends.

Lastly, my deepest thanks go to my family, who always support and care for me, no matter where I am. Especially thanks to my mom for always believing in me and supporting me in every possible aspect.

Enrique Fernández Martín
Valencia, July 2023

Abstract

This research is concerned with the topic of low-thrust trajectory design to Temporarily Captured Orbiters (TCOs). TCOs are meter-sized objects temporarily captured in the Earth-Moon system before continuing their heliocentric path. They originate from fragments of Near-Earth Asteroids on Earth-like trajectories that present the precise entry conditions to be captured. Their small size and high speed make them difficult to identify from ground measurements. Only two have been confirmed at the time of writing: 2006 RH120 and, more recently, 2020 CD3. However, studies suggest a constant presence of at least one TCO at any point in time. The construction of more advanced asteroid survey systems can potentially increase the rate at which these objects are identified. Hence, TCOs are becoming desirable targets for future exploration missions.

A handful of studies have been published that compute transfer trajectories targeting TCOs, but the topic is far from mature. The studies mainly focus on high-impulse transfer designs departing from hibernating orbits near the Earth-Moon Lagrangian points. The only targets studied are the capture phase of 2006 RH120, which occurred between July 2006 and June 2007, and a database of virtual TCOs generated for lack of physical data.

The present study aims at conducting a preliminary analysis of the transfer possibilities to rendezvous with TCOs for a spacecraft powered by a low-thrust electric propulsion engine departing from a periodic orbit near the Sun-Earth L1 point. This region is a good departure candidate since L1 is considered a gateway for TCOs into the Earth's system. At the same time, mission concepts have been proposed to place a telescope that surveys TCOs at this location. The subject of this study is TCO 2006 RH120, but the conclusions and tools developed are general and are meant to be used to analyze other TCOs as they are identified. The reference spacecraft used for the study has a mass of 500 kg and an ion propulsion engine with a specific impulse of 3000 s and a maximum thrust of 0.1 N.

To conduct this analysis, a fast and robust optimization algorithm is developed. The Hill three-body problem models the dynamics, and the continuous thrust is modeled as a discrete set of delta-v impulses applied at different trajectory points. The algorithm solves a delta-v minimal optimization problem with a direct method transcribed as a single-shooting problem, which is solved using *fmincon*, a gradient-based interior-point solver from MATLAB. The most innovative aspect of the tool that makes it ideal for the present study is the analytical computation of the gradient of the cost and non-linear constraint functions, which very significantly increases the speed of the trajectory optimization process.

The developed tool has proven successful in analyzing 200 departure orbits drawn from families of Halo and Lyapunov periodic orbits. For each orbit, the algorithm optimizes the departure location, the rendezvous location in the TCO's orbit, the magnitude and direction of delta-v impulses, and the transfer time. Several low delta-v transfers have been found for different departure orbits, with values in the 200 m/s range. The resulting transfer trajectories are analyzed in this report, and conclusions are drawn with implications that can be extended to other TCOs.

Nomenclature

Abbreviations

AU	Astronomical Unit
CR3BP	Circular Restricted Three-Body Problem
DC	Differential Correction
ESA	European Space Agency
ECLIPJ2000	Earth Mean Orbit of J2000
EME2000	Earth Mean Equator and Equinox of J2000
EMO2000	Earth Mean Orbit of J2000
GA	Genetic Algorithm
GCRF	Geocentric Celestial Reference Frame
GNC	Guidance, Navigation, and Control
IAU	International Astronomical Union
IC	Initial Condition
ICRF	International Celestial Reference Frame
IVP	Initial Value Problem
JD	Julian Date
JPL	Jet Propulsion Laboratory
LSST	Large Synoptic Survey Telescope
NLP	Nonlinear Programming
ODE	Ordinary Differential Equation
RAAN	Right Ascension of the Ascending Node
SQP	Sequential Quadratic Programming
SRP	Solar Radiation Pressure
SSB	Solar-System Barycenter
STM	State Transition Matrix
TCF	Temporarily Captured Fly-by
TCO	Temporarily Captured Orbiter
TPVBP	Two-Point Boundary Value Problem
Tudat	TU Delft Astrodynamics Toolbox
NEO	Near-Earth Object
UTC	Universal Time Coordinated

Latin Symbols

A	Jacobian of linearized system
a	Semi-major axis [km]
$\mathbf{a}$	Acceleration vector [$\mathrm{km\,s^{-2}}$]
C	Jacobi Constant [-]
c_{eff}	Effective exhaust velocity of propellant [m/s]
e	Eccentricity [-]
F	Engine's Thrust [N]
G	Universal gravitational constant [$\mathrm{m^3\,kg^{-1}\,s^{-2}}$]
g_0	Gravitational acceleration at Earth's surface [$\mathrm{m\,s^{-2}}$]
$\mathbf{g}_{\text{eq}}$	Equality constraint vector
$\mathbf{g}_{\text{ineq}}$	Inequality constraint vector
I	Identity matrix
I_{sp}	Specific impulse [s]
i	Inclination [rad]
J	Cost function [km/s]
m	Mass of the spacecraft [kg]
m_0	Initial wet mass of spacecraft [kg]
m_{f}	Final dry mass of the spacecraft [kg]
$\dot{m}$	Engine's mass flow [kg/s]
N	Angular velocity magnitude of synodic frame [rad/s]
n	Number of delta-v impulses [-]
P	Period of an orbit [days]
P_{source}	Power supplied by spacecraft's power source [J/s]
$\mathbf{r}$	Position vector [km]
r	Magnitude of position vector [km]
r_{a}	Radius of apoapsis [m]
r_{p}	Radius of periapsis [m]
$\mathbf{s}$	Position-velocity state vector
T	Transfer time [days]
T_0	Initialization transfer time [days]
U	Potential function of CR3BP [-]
V	Spacecraft's velocity in the rotating frame [-]
$\mathbf{v}$	Velocity vector [km/s]
v_x, v_y, v_z	Cartesian components of velocity vector [km/s]
W	Potential function of Hill model [-]
$\mathbf{X}$	Decision variable vector
x, y, z	Cartesian components of position vector [km]

Greek Symbols

β	Rotation angle between the rotating and inertial frame [rad]

Γ	Jacobi-like constant in the Hill model [-]
ϵ	Power source conversion efficiency [-]
ε	Departure epoch [days]
ε_0	Initialization departure epoch [days]
θ	True anomaly [rad]
$\lambda_{A,B}$	Scale factor between frames A and B
μ	Mass normalization parameter in CR3BP [-]
μ_m	Gravitational parameter [$\mathrm{m}^3\ \mathrm{s}^{-2}$]
τ	Target epoch [days]
τ_0	Initialization target epoch [days]
$\Phi(t, t_0)$	State Transition Matrix between epoch t_0 and t
Ω	Right ascension of the ascending node [rad]
ω	Argument of periapsis [rad]
$\boldsymbol{\omega}_{S,O}$	Angular velocity vector between synodic and inertial frame

Notation

f'	First derivative of function f
f''	Second derivative of function f
m_i	Mass of i-th body
$R_{A,B}$	Rotation matrix from frame A to B
$\mathbf{r}_i$	Position vector of i-th body
$\mathbf{r}_{ij}$	Position vector of body j relative to body i
Δt_j	Duration of the j-th trajectory arc [days]
ΔV	Total transfer velocity change [km/s]
Δv	Trajectory segment velocity change [km/s]
Δv_i	i-th velocity impulse [km/s]
Δv_0	Departure delta-v [km/s]
Δv_f	Arrival delta-v [km/s]
$X^\top$	Transpose of matrix X
X^{-1}	Inverse of matrix X
$\mathbf{X}_{A,B}$	Origin of frame A relative to frame B [km]
$X_{a\times b}$	Matrix X of size $a \times b$
$\mathbf{x}^{(A)}$	Vector $\mathbf{x}$ expressed in frame A
x^*	Relative to an estimated quantity
$\dot{x}$	Time derivative of x
$\ddot{x}$	Second time derivative of x
δx	Small deviation of x
$\mathbf{x}_{0,j}$	Vector $\mathbf{x}$ at departure of the j-th trajectory arc
$\mathbf{x}_\mathrm{arr}$	Vector $\mathbf{x}$ at the end of a propagation
$\mathbf{x}_\mathrm{dep}$	Vector $\mathbf{x}$ at departure
$\mathbf{x}_{\mathrm{f},j}$	Vector $\mathbf{x}$ at the end of the j-th trajectory arc
$\mathbf{x}_\mathrm{tgt}$	Vector $\mathbf{x}$ at target epoch

$\mathbf{x}^-$	Vector $\mathbf{x}$ before delta-v impulse
$\mathbf{x}^+$	Vector $\mathbf{x}$ after delta-v impulse
$\tilde{\mathbf{x}}$	Skew-symmetric matrix of vector $\mathbf{x}$

Contents

1

Introduction

The exploration of celestial bodies has become increasingly important in recent years. As human presence in space gains relevance and space becomes more accessible, an inevitable spotlight is placed on those foreign objects that are still unexplored. Access to them is becoming attractive from different fronts: as a means of planetary defense against potential impact threats, from the standpoint of gaining scientific knowledge, or from their practical purpose of obtaining material resources that are scarce on Earth. For either of these reasons, accessing asteroids is becoming attractive to both space agencies and commercial companies. They are addressed in the third pillar of the European Space Agency's (ESA) Agenda for 2025, which sets the development of space for safety and security as a European priority (Aschbacher, 2021).

Among the list of many celestial objects that could be explored, this thesis focuses on a specific category called Temporarily Captured Orbiters (TCOs), also known colloquially as mini-moons. TCOs are a relatively minor type of Near Earth Objects (NEOs) (order of a few meters) that become loosely trapped in Earth's gravity field, performing at least one complete revolution until they eventually leave the system and continue their heliocentric course. These objects are attractive to the scientific community as their size, proximity, and accessibility make them affordable candidates for exploration missions. These missions present the opportunity to test current planetary defense technologies, test close-proximity procedures, and assess the feasibility of commercial mining missions (Jedicke et al., 2018). However, due to their small size and their limited time in Earth's environment, they are not easily identified by current ground-based observations – modern surveys have an annual probability of 10^{-7} of detecting a meter-size object within one lunar distance (Granvik et al., 2013). To this date, only two TCOs have been identified. The first was 2006 RH120 (Kwiatkowski et al., 2009), which stayed bound to Earth for a year, completing four revolutions. The second TCO is 2020 CD3, identified before leaving the system (Bolin et al., 2020) and had an excursion duration of 2.7 years (Fedorets et al., 2020a). Recently, a third similar object has been identified called 2022 NX1. However, as it did not complete a full revolution, it falls under the category of Temporarily Captured Fly-bys (TCFs). It is expected that 2020 NX1 will become a TCO in 2051 (de la Fuente Marcos & de la Fuente Marcos, 2022).

Despite the limited observations of TCOs, a study by Granvik et al., 2012 concluded that there is a constant presence of at least one meter-sized TCO at any given time. They tend to enter through Sun-Earth L1-L2 points and remain energetically bound within three Hill sphere radius. They are captured on average for 286 days and make almost three revolutions around the Earth. A study conducted by Fedorets et al., 2020b predicts that more TCOs will be discovered at least once per year using the upcoming Large Synoptic Survey Telescope (LSST). Furthermore, commercial companies such as KARMAN+ have proposed creating dedicated space telescopes to observe potentially accessible NEOs, including TCOs.

Few studies have been conducted that place the focus on transfer trajectories targeting TCOs. Chyba et al., 2014 studied low-thrust transfers targeting six virtual TCOs from the Granvik et al., 2012 database. They optimized trajectories to reduce the total transfer time, departing from Earth's Geostationary orbit and targeting TCO passages near the Earth-Moon L1 point. They obtained a handful of successful trajectories whose transfer time ranges from 55 to 88 days using a maximum thrust of 0.2 N. A later study by Brelsford et al., 2016 specifically searched for transfer trajectories targeting TCO 2006 RH120 from a hibernating halo orbit around the Earth-Moon L2 point. The propulsion selected was chemical, with a maximum of three impulses for the transfer. The best rendezvous transfer detected had a duration of 255 days and a ΔV of 797 m/s (see Figure 1.1). Another study by Chyba and Patterson, 2019 searched for high-thrust fuel-minimum transfers targeting a large population of 3000 virtual TCOs departing from a halo orbit around the Earth-Moon L2 point. They obtained feasible transfer trajectories to every TCO, with a mean ΔV of 724.5 m/s.

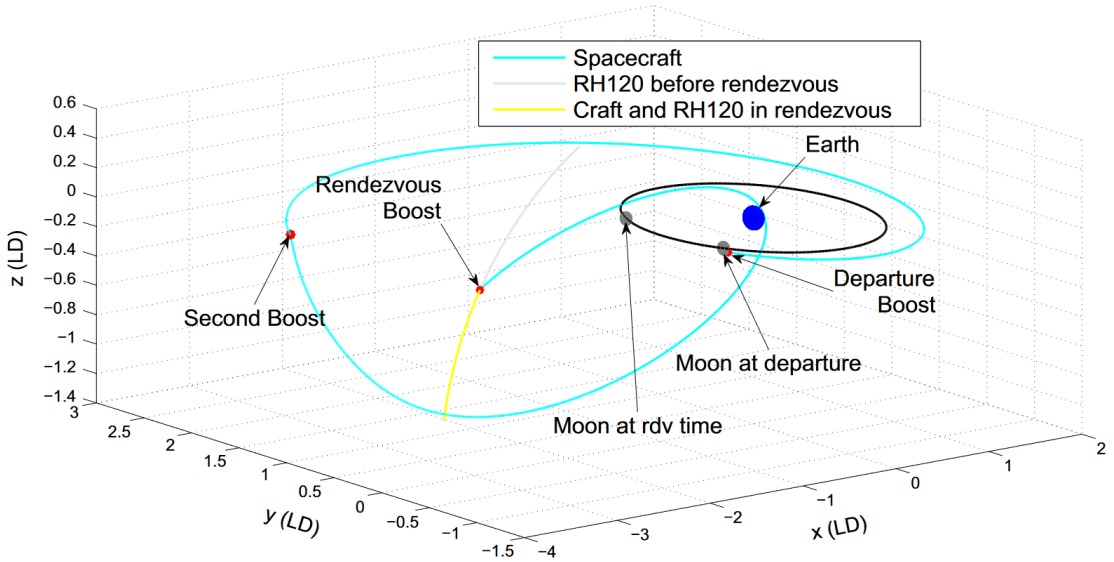


Figure 1.1: Three-impulse rendezvous transfer to 2006 RH120 from Halo orbit around Earth-Moon L2 from Brelsford et al., 2016.

Other studies have focused on other mission-related aspects of exploration missions to 2006 RH120, such as Chyba et al., 2016, which studied the cost of the overall round trip from an Earth-Moon L2 halo orbit. Others like Takahashi et al., 2022 focused on developing close-proximity strategies to rendezvous given their particular irregular gravity. The study by Urrutxua et al., 2015 quantified the cost of acting on 2006 RH120 to extend the capture duration, concluding that a small ΔV of 32 m/s would have sufficed to extend the capture for five years.

This thesis focuses on the study of low-thrust transfer trajectories to rendezvous with TCOs minimizing total transfer ΔV and departing from orbits around Sun-Earth L1. The study uses as a test bench the TCO 2006 RH120 to assess the feasibility of such transfers and serve as a reference for future identified TCOs. As explained, previous research has studied the problem of transferring to TCOs. However, they were mainly concerned with high-thrust solutions departing from Earth-Moon libration point orbits, or the low-thrust studies are focused on minimizing the time of flight instead of the ΔV. The departure location is another fundamental element that differentiates the present study from the literature. Previous studies departed from either Geostationary orbit or Earth-Moon L2 Halo orbits. To the writer's knowledge, no analysis has been done departing from Sun-Earth libration point orbits, which may provide promising rendezvous opportunities for being the gateway of these objects into Earth capture.

1.1. Research Questions

Based on the considerations mentioned above, the objective of this thesis research is to analyze the transfer options for an electrically propelled spacecraft to rendezvous with TCOs from different departure locations in the vicinity of Sun-Earth's L1 point, which is the gateway of TCOs into the Earth's environment. One of the few confirmed TCOs – 2006 RH120 – is used as a target. By achieving this research objective, the following research question will be answered:

Research question: How can transfers originating near the Sun-Earth L1 be leveraged to target Temporarily Captured Orbiters such as 2006 RH120 in a spacecraft configuration that uses electric propulsion?

More specific sub-questions that will be answered are:

- How can the optimization problem be formulated to conduct a preliminary analysis on transfers departing from orbits around L1?
- How do design parameters such as departure orbit or rendezvous location affect the delta-v cost of the transfers?

1.2. Report Outline

This section presents the thesis work structure to meet the research objectives and answer the research questions presented in the previous section. The first part of the thesis in Chapter 2 is focused on building the thesis methodology from the literature heritage regarding asteroid missions, electric propulsion, and low-thrust trajectory optimization methods. The thesis continues in Chapter 3 with a description of the fundamental elements of flight dynamics used in this research. Chapter 4 discusses the different numerical tools that allow propagating and optimizing the trajectories. Chapter 5 starts implementing the trajectory optimization algorithm by considering two-impulse transfers and building up the complexity of the tool. Then, in Chapter 6, the low-thrust optimization algorithm is designed and tested at its total capacity. Along with verification and validation, the code structure is presented in Chapter 7. Once the algorithm is verified and validated, it is used to generate the results in Chapter 8, where different transfers are run from different departure orbit around the Sun-Earth L1 point, and their

results are presented and analyzed. Finally, Chapter 9 gathers the main conclusions from this work and provides recommendations for future work.

2

Heritage

This chapter aims to gather the relevant heritage to tackle the problem introduced in the previous chapter (see Section 1.1). The chapter begins by introducing the different types of celestial objects, focusing on TCOs and relevant missions to NEOs. Then, the topic of electric propulsion for spacecraft is introduced, giving some relevant figures for common engines. Subsequently, the trajectory optimization problem is presented, and the different methods to formulate and solve it are reviewed. Finally, the heritage presented is used to describe the mission scenario of the present research, including the environment and the vehicle selected, and then the methodology to accomplish the research objectives is presented.

2.1. Background on celestial bodies and TCOs

Aside from the eight planets in the Solar System, other smaller bodies orbiting the Sun have been objects of study. Among these, the ones that present a coma or tail are known as *comets*, whereas those that do not possess these features are called *minor planets*. Minor planets and comets can be found in different regions: the asteroid belt, in between Mars and Jupiter's orbit; the Kuiper Belt and Oort Cloud beyond the orbit of Neptune; and near the Lagrangian points of Jupiter, known as Trojans (Lissauer & de Pater, 2019).

The term asteroid refers to minor rocky planets found at distances up to slightly past Jupiter's orbit, mainly coming from the asteroid belt. The largest asteroid is Ceres, which has a size of 470 km in radius. The objects that come close to the Earth are called Near Earth Objects (NEOs). The primary source of NEOs is the gaps found in the asteroid belt that are known as Kirkwood gaps. They are located in regions of resonance with Jupiter. These resonances affect the eccentricities of their orbits so that they can reach the orbits of Mars and the Earth (Lissauer & de Pater, 2019).

The population of NEOs can be subdivided into four categories according to their orbits, the most predominant being the Apollo asteroids, whose perihelia is smaller than 1.017 Astronomical Units (AU)

and semi-major axis larger than 1 AU. The dynamical lifetime of NEOs is of 10^7 years, as they end up being ejected to interstellar space, destructed by the Sun, or colliding with other planets and Moons. Further information on minor planets and comets can be found in Lissauer and de Pater, 2019.

A small fraction of the NEOs with low inclination orbits approach Earth and are removed from their resonance with their giant planet into orbits resembling Earth's orbit. Among these, a tiny fraction of their orbits pass through Earth's Hill sphere at the precise speed, direction, and distance such that they become loosely captured in the Earth-Moon system, becoming TCOs. The small probability of meeting these conditions suggests that TCOs have small sizes of one to two meters in diameter, as their population reduces exponentially with increasing size (Granvik et al., 2012). Figure 2.1 shows the estimated number of objects given their diameter for different types of minor planets. It can be observed how the central belt feeds the NEO population, which in turn feeds the population of objects with Earth-like orbits (called ARM in the figure for being targets of NASA's Asteroid Retrieval Mission). Finally, they feed the population of minimoons as they reduce in size and number.

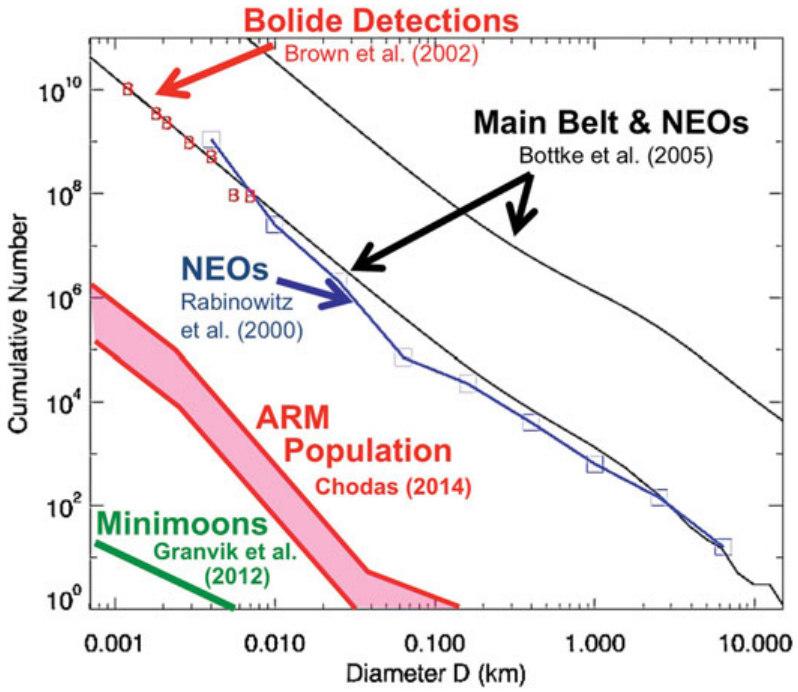


Figure 2.1: Frequency distribution of small objects in the inner Solar System (Jedicke et al., 2016).

Another potential origin for TCOs is that they are the product of the collision of asteroids with the Moon. However, the percentage of TCOs that fall into either category (asteroid belt or lunar ejecta) is unknown (Jedicke et al., 2016).

Although no missions have been flown targeting TCOs, which is the object of the present study, several missions have been sent to rendezvous with NEOs. The main characteristics of these missions relevant to this thesis are gathered in Table 2.1.

Table 2.1: List of missions to NEOs and their characteristics.

Mission	Country	Launch	Target	Type	Propulsion	Launch Mass [kg]
NEAR	USA	1996	433 EROS	Amor	Chemical	805
Hayabusa	Japan	2003	Itokawa	Apollo	Ion	510
Hayabusa 2	Japan	2014	Ryugu	Apollo	Ion	609
OSIRIS-REx	USA	2016	Bennu	Apollo	Chemical	2,110
DART	USA	2021	Didymos	Apollo	Chemical/Ion	610

2.2. Electric Propulsion

The most extended means of propelling a spacecraft are chemical engines, whose high thrust allows to change the spacecraft's velocity in a short time. For trajectory design purposes, these impulses can be modeled as instantaneous events. Although these engines are well developed and reliable, they do require a large amount of propellant to generate the required thrust.

An alternative means of propelling a spacecraft is through continuous thrusting, which involves more mass-efficient systems that allow allotting a higher portion of the spacecraft's total mass to the payload. They achieve so at the expense of providing significantly reduced amounts of thrust compared to impulsive systems, hence commonly known as low-thrust propulsion systems. Although several types of propulsive systems generate low thrust, such as solar sails, we will use the term low-thrust propulsion to refer to electric propulsion systems.

As opposed to chemical propulsion systems, where both the energy source and the exhaust medium come from the propellant, in electric propulsion systems, the energy needed to accelerate the propellant comes from an external source, such as batteries, solar panels, or a nuclear reactor. These systems allow for much higher exhaust velocities than conventional propulsion systems while keeping the propellant consumption at a minimum (Wakker, 2015).

The efficiency of these engines is commonly measured by a metric called specific impulse (I_{sp}), which is related to the exhaust velocity of the particles. High I_{sp} engines require less propellant to achieve the same amount of delta-v. However, as the I_{sp} increases, so is the power requirement from the spacecraft, which adds to the total mass of the spacecraft.

The thrust requirement on low-thrust engines for interplanetary travel, Solar System exploration, and orbit-rising is of the order of 0.01–10 N. Based on current technology, operational electric ion engines meet those thrust constraints with a specific impulse of roughly 3,000 s. These engines have an operational lifetime of more than 30,000 hours and can provide thrust ranging from 0.015 to 0.235 N (Sutton & Biblarz, 2017).

Although the choice of electric propulsion engines can significantly reduce the mass of a spacecraft devoted to the propulsion system, they complicate the trajectory design process, as the thrust impulses cannot be modeled as instantaneous events. In turn, they should be modeled as a continuous event, where the amount of thrust and direction are designed in most cases by solving an optimal control problem.

2.3. Trajectory Optimization

In this section, the topic of trajectory optimization is introduced with application to designing transfers using electric propulsion systems with continuous low thrust.

Spacecraft trajectory optimization can be simplified to solve the following problem: to determine the trajectory of a spacecraft connecting a departure with a target while minimizing a certain quantity of importance (Conway, 2010). The problem becomes particularly complex when dealing with non-linear dynamics, discontinuities in the state variables (caused by delta-v's), and initial and final conditions being unknown (can be optimization variables). To understand and tackle this problem, Shirazi et al., 2018 have divided the problem into four steps. These are subsequently explained with application to low-thrust trajectory design.

Models – The first step in solving an optimal control problem is to select the variables that represent the system's state and introduce the differential equations that shape the evolution of the states.

Objectives – The objective of the optimization problem is defined based on the mission requirements. Common objectives in trajectory design are minimization of time of flight or propellant mass.

Approaches – The next step is to develop an approach for finding the optimal trajectory, which can be either analytical or numerical. Analytical approaches are only available for specific simplified cases and are not the subject of this thesis. Numerical approaches are generally divided into two methods: indirect and direct.

- Indirect methods solve the optimal control problem by deriving necessary analytical conditions from the calculus of variations, which requires expanding the state with parameters known as co-states. Although these methods are generally very reliable, the difficulty of providing good-enough initial estimates for these states makes these methods complex to use as a starting point.
- Direct methods: They transcribe the optimal control problem into a nonlinear programming (NLP) problem where the states are approximated by a piecewise constant parameterization. Direct methods are generally easier to implement, at the expense of being more challenging to ensure the optimality of the solution obtained (Shirazi et al., 2018).

Both direct and indirect methods use standard numerical *techniques* to impose the dynamics and constraints in the solution. The most common are differential inclusion, single shooting, multiple shooting, and collocation techniques. Research has shown that studies targeting low-thrust trajectories in multi-body systems are fairly split between direct and indirect methods (Morante et al., 2021).

Solutions – The last step is to solve the optimization problem after selecting the approach. Several computational techniques and algorithms are available. For both direct and indirect formulations, the solution methods can be generally classified into gradient-based and heuristic techniques.

- *Gradient-based* (deterministic) techniques use the information on the gradient of the objectives and constraints to drive the algorithm to a solution, which benefits from fast convergence and accurate results. Gradient-based methods are popular in trajectory optimization problems. A large number of solvers is readily available.

- *Heuristic* methods are alternative means of solving optimization problems. Instead of using gradient information, they rely on stochastic processes to search. Genetic Algorithms (GA) and Particle Swarm Optimization are among the most common techniques. Especially GAs are popular given their large availability and ease of use. However, heuristic techniques are used more for impulsive thrust trajectories and are less common for continuous thrusting (Morante et al., 2021).

Several algorithms implement gradient-based methods to solve the NLP problem. They can be generally classified into Sequential Quadratic Programming (SQP), such as SNOPT and NPSOL, and interior point methods, like IPOPT and KNITRO (Morante et al., 2021).

2.4. Mission Scenario

After a careful review of previous literature regarding asteroid exploration, transfers to TCOs, and low-thrust trajectory optimization, the following mission scenario has been selected in this thesis.

The thesis is focused on the preliminary study of the potential trajectories for a mission to rendezvous with TCO 2006 RH120. The spacecraft left the Earth-Moon influence in July 2007. Still, given the scarce source of confirmed TCOs and the limitations to identifying them, it is used as a feasibility test to gain insight into the problem and apply it to other potential TCOs identified shortly.

The study focuses on a mission segment where the spacecraft departs from a periodic orbit around Sun-Earth L1, which is the entry region of TCO 2006 RH120 into the Earth's system. This study aims to analyze the different departure options and their impact on the trajectory transfer time and delta-v. Transfers that target early epochs of the TCO excursion into Earth's influence are preferred since they give more margin for contingencies before the TCO leaves the system. However, the overall trajectory of the TCO within the Hill sphere of Earth is studied as a potential target region.

A relatively lightweight spacecraft of 500 kg of wet mass is selected for the mission. Lighter spacecraft could be considered, but this is a maximum-case scenario for a lightweight mission considering the higher mass of other spacecraft targeting NEOs presented in Table 2.1. The amount of propellant is sought to be minimized so that the spacecraft can allot more mass for the payload. The spacecraft's propulsion system is an electric propulsion engine with a specific impulse selected of 3,000 s. Higher I_{sp} engines are being tested for missions, such as the NEXT-C Xenon engine with up to 4,000 s, but, on average, 3,000 s has been identified as a reasonable engine impulse. The maximum thrust provided by the engine is set to 0.1 N. Again, engines have been found with higher thrusts of 0.2 N, but 0.1 N is considered a reasonable force for the preliminary study where the spacecraft systems have not been designed. For these engine specifications, the maximum power needed is not considered an issue for the mission. It is assumed that a reasonably-sized solar panel unit is assumed to feed the electric propulsion system.

2.5. Methodology

The methodology proposed to meet the objectives of the thesis is now presented.

The model selected for the optimization problem is the Hill Three-Body Problem or simply the Hill model, and its selection is motivated in Section 3.4. Overall, it has been selected as a good trade-off between including the most important features that describe the dynamics environment while presenting some valuable features that simplify the study, such as the constant of motion, the origin of the reference frame at Earth, and the simplicity to codify.

The optimization is posed as a single-objective problem focused on minimizing the total transfer delta-v. The time of flight is an essential parameter in the design given that time is limited to rendezvous with them before they leave the Earth-Moon system and their difficulty to be identified before capture. However, given the uncertainty regarding the performance of future asteroid survey systems, the transfer time will be monitored and discussed. Still, it is not considered a mission enabler such as the delta-v. Therefore, the term mission cost will refer to the delta-v cost of the total transfer in this document.

The direct optimization method is proposed for solving the optimization problem. The problem is discretized into a series of parameters called decision variables that are then solved as parameter optimization problems by a numerical solver. A single shooting technique is selected to transcribe the problem. The motivation for why this was selected and its description is introduced in Section 6.2. Overall, it resulted in a more intuitive and straightforward implementation than collocation techniques, and it simplified the computation of analytical gradients compared to the multiple-shooting technique.

Regarding the solution methods, we have opted for a gradient-based technique with an interior-point method, as it has been identified to provide good results for low-thrust trajectory optimization in the many-body problem (Morante et al., 2021). At the same time, it has been found to have good accessibility from the software selected, which is subsequently explained.

Finally, the software selected for the implementation of the optimization problem is MATLAB. It has been traded against other scripting languages like Python or Julia. However, MATLAB provided several advantages: the software was the most familiar of the three, having previous experience coding dynamical equations for propagation and differential correction algorithms. At the same time, MATLAB offers intuitive optimization capabilities through *fmincon* function, which has been used in similar studies for low-thrust trajectory optimization (Ozimek and Howell, 2010, Trofimov et al., 2018).

3

Flight Dynamics

This chapter introduces the flight dynamics elements that are relevant to the research. First, the time systems are introduced. Then, the different reference frames used in the thesis and their transformations are presented. Next, the state model and the dynamical models are presented and discussed. They are fundamental to describing the motion of the propagated bodies of this research.

3.1. Time Systems

It is essential for this thesis to define methods to express time and time intervals, which will be used both to identify epochs from celestial bodies and to quantify time in the simulations. Two systems are used and are subsequently described: Julian Date and Seconds Since J2000.

3.1.1. Julian Date

Julian Date (JD) is a system commonly used to determine time intervals in catalogs of celestial bodies. It represents a more straightforward way of measuring the time between two epochs than traditional calendar dates, which present length variations depending on the months involved and leap year. The Julian Date system is based on the concept of Julian Day, which corresponds to 24 hours of 3,600 s each. A Julian Year is composed of 365.25 Julian Days, eliminating the need for a leap year. An epoch is selected as a reference of the Date Zero, January 1st of the year 4,713 BC at Noon in Alexandria (Wakker, 2015). A more common implementation to avoid large numbers is the Modified Julian Date, which is obtained by subtracting 2,400,000.5 days from the Julian Date. However, this thesis uses Julian Dates due to conventions from reference material.

This thesis uses Julian Dates to retrieve epochs from the Earth, Sun, Moon, and TCOs from Jet Propulsion Laboratory's (JPL) SPICE Toolkit and JPL's Small-Body Database. Conversions to calendar dates are used in this report for interpretation and readability but are not used in numerical simulations. The conversions are performed using the TU Delft Astrodynamics Toolbox (Tudat), which uses Python's

Datetime library.

3.1.2. Seconds Since J2000

Another reference time system is used in simulations to identify the epochs, which is *Seconds Since J2000*. Seconds Since J2000 has the second as the fundamental unit of time and uses as a reference origin January 1st of 2000 at UTC0. Seconds Since 2000 are also used as the reference time system for every propagation performed in the Tudat software and is therefore convenient to compare simulations.

Computing the transformation between the two time systems used in the thesis is useful. The transformation is computed by identifying the Julian Date corresponding to 2000 January 1st, which is JD $2,451,545.0$. Therefore, the conversion between the two systems is done using the following formula (Tudat Space Documentation, 2022).

$$\text{Seconds since J2000} = (\,\text{Julian Date} - 2,451,545.0\,) \times 86,400.0 \tag{3.1}$$

It is noted that for practical reasons, the simulation is started at time zero. However, by providing the simulation's initial epoch, every subsequent epoch in the propagation can be identified.

3.2. Reference Frames and Transformations

A reference frame serves to locate and orientate the coordinate system used to define the motion of bodies in space. Inertial reference frames are fixed in space both in position and orientation and do not undergo any acceleration. This means that a physical object described in such a frame keeps a constant velocity if no forces are acting on it, and the dynamics can be defined using Newton's second law. On the contrary, the motion of the bodies described in non-inertial reference frames should account for the accelerations the frame undergoes. In engineering, a frame is considered inertial when the acceleration of the frame can be neglected in the motion of the dynamical system. For more information, see Curtis, 2020.

Given that this thesis deals with the motion of TCOs that are loosely trapped in the Earth-Moon system, it is convenient to place the origin of the reference frames in the Earth center for simplicity. In this section, we explore a selection of reference frames relevant to this work.

3.2.1. Geocentric Celestial Reference Frame

It is common for astrodynamics applications to use the International Celestial Reference Frame (ICRF) or its geocentric equivalent, the Geocentric Celestial Reference Frame (GCRF), adopted by the International Astronomical Union (IAU). Both frames share the orientation of the axes, which is considered inertial for being fixed with respect to distant objects (*i.e.* quasars and galactic nuclei) and has no date associated with its location. The center of the ICRF is located at the Solar-System Barycenter (SSB), which is considered inertial for describing the motion inside the Solar System. The center of the GCRF is placed at the center of the Earth, which can only be considered inertial for specific applications near the Earth. For more information on ICRF and GCRF frames, see Petit and Luzum, 2010 and Kaplan,

2005.

To describe the motion of celestial objects in ephemeris models, other reference systems are selected, which are defined using the ecliptic and equatorial planes. These are the J2000 and ECLIP2000, which are subsequently described.

3.2.2. Geocentric Earth Mean Equator and Equinox at J2000

The Earth's Mean Equator and Equinox of J2000 (EME2000 or simply "J2000") is a commonly used reference frame with an inertial orientation aligned with the Earth's equator (Parker & Anderson, 2014). The direction of the axes is aligned as follows (see Figure 3.1):

Origin: Earth's center.
z-axis: Pole vector of the Earth's equator at J2000.
x-axis: Vernal equinox at J2000. *i.e.* the cross product of equatorial and ecliptic pole vectors.
y-axis: Completes the right-hand system.

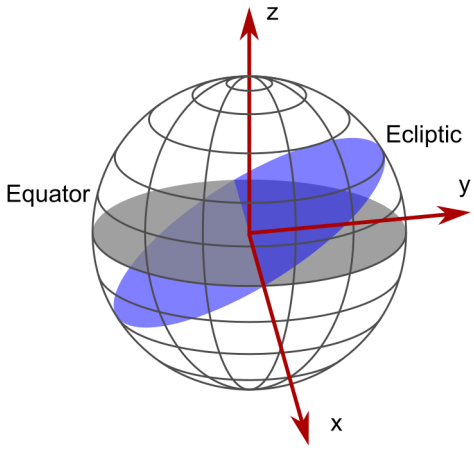


Figure 3.1: Graphical representation of J2000 reference frame.

The equatorial plane pole and vernal equinox directions are selected at a reference epoch: the beginning of the year 2000. This selection makes the axes of the frame practically considered identical to those of the GCRF frame, presenting only a slight frame bias (Kaplan, 2005). The frame's origin can be either the SSB or the Earth's center. Throughout this thesis, the latter is used for interpretation purposes. Note that although the orientation of the frame is inertial, the acceleration of the Earth's center has an influence on the motion of the bodies described in this frame.

The EME2000 or J2000 frame is commonly found in Emphemeris models and, in this thesis, will be used to retrieve the motion of celestial objects from SPICE and when propagating trajectories using the Tudat software.

3.2.3. Geocentric Earth Mean Orbit of J2000

Another reference frame similar to EME2000 is the Earth Mean Orbit of J2000 (EMO2000 or simply "ECLIPJ2000"), which is aligned with the ecliptic instead of with the equator (Parker & Anderson, 2014). The direction of the axes is the following (see Figure 3.2):

Origin: Earth's center.
z-axis: Ecliptic pole at J2000.
x-axis: Vernal equinox at J2000.
y-axis: Completes the right-hand system.

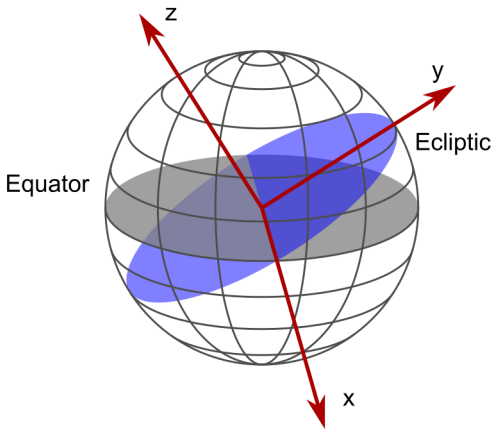


Figure 3.2: Graphical representation of ECLIPJ2000 reference frame.

Similarly to the EME2000 frame, the axes' directions are selected at the J2000 epoch, and the origin is selected at the Earth's center for this thesis. This frame is also used in ephemeris models and Tudat. This frame is more practical than EME2000 for this thesis since it already aligns the Earth and Sun into the same plane, which is convenient for transformations to the Sun-Earth synodic frame subsequently presented. It is also the reference frame used in other studies targeting 2006 RH120, such as Urrutxua et al., 2015.

3.2.4. Synodic Frame

A useful frame when working with three-body dynamics models is the co-rotating or synodic reference frame (Wakker, 2015). This frame allows expressing the equations of motion in a simplified and insightful way, as will be explained in Section 3.4. Synodic frames rotate at the same rate as the primaries, which makes their orientation non-inertial. If the orbits of the primaries are assumed to be circular, the frame rotates at a constant rate, while it oscillates periodically along the *x-axis* if they are eccentric. The orientation of the axes in the frame is as follows (see Figure 3.3):

Origin: Earth's center.
z-axis: Angular momentum vector direction of the system.
x-axis: Connects the center of the two bodies, directed opposite to the most massive body.

y-axis: Completes the right-hand system.

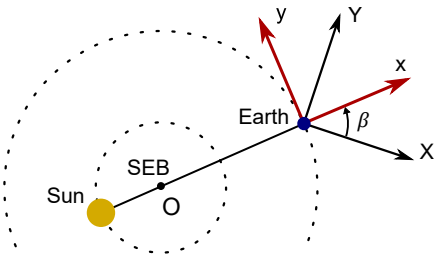


Figure 3.3: Graphical representation of synodic reference frame.

The frame origin is commonly placed at the barycenter of the system for being an inertial point. However, for the sake of this thesis, the center will be placed at the location of the Earth, which is the less massive body, while the other massive body included in the model is the Sun.

3.2.5. Frame Transformations

It is important for this thesis to be able to compute transformations between reference frames. These transformations are needed when data is obtained from sources that use different reference frames or to change the frame to another that provides better visualization of results. The frame transformation that is used in this thesis is between EMOJ2000 data from ephemeris models and the synodic frame used for designing transfer trajectories. This transformation is subsequently derived.

The formula to express the coordinates of a point in a reference frame $\mathbf{X}^{(A)}$ in another reference frame $\mathbf{X}^{(B)}$ is:

$$\mathbf{X}^{(B)} = \mathbf{X}_{A,B}^{(B)} + \lambda_{A,B}\, R_{A,B}\, \mathbf{X}^{(A)} \tag{3.2}$$

where $\mathbf{X}_{A,B}$ is the relative position of the origin of frame A with respect to the origin of frame B, $\lambda_{A,B}$ is the scale factor, and $R_{A,B}$ is the rotation matrix from frame A to frame B (ECSS-E-ST-10-09C, 2008).

When the transformation is between position-velocity states (*i.e.* $\mathbf{x} = [\mathbf{r}; \mathbf{v}]$), and assuming no scale factor, the formula adopts the following shape (Dirkx, 2022)

$$\mathbf{x}^{(B)} = \mathbf{x}_{A,B}^{(B)} + \begin{bmatrix} R_{A,B} & 0_{3\times3} \\ \dot{R}_{A,B} & R_{A,B} \end{bmatrix} \mathbf{x}^{(A)} \tag{3.3}$$

where $\dot{R}_{A,B}$ is the time derivative of the rotation matrix $R_{A,B}$.

The EMOJ2000 and the synodic frames share the same origin at the Earth's center. Regarding their orientation, EMOJ2000 has an inertial orientation, whereas the synodic frame is a non-inertial rotating frame. Since the angular momentum vector of the synodic frame points in the direction of the Ecliptic, both frames are aligned on the z-axis. Therefore, the transformation between them only involves a rotation with respect to this axis. Elements expressed in the synodic frame are marked with the superscript S, while the elements expressed in EMOJ2000 are marked with the superscript O. The equation that transforms these two frames from Equation (3.2) adopts the following form:

$$\mathbf{x}^{(O)} = \begin{bmatrix} R_{S,O} & 0_{3\times3} \\ \dot{R}_{S,O} & R_{S,O} \end{bmatrix} \mathbf{x}^{(S)} \tag{3.4}$$

where the rotation matrix $R_{S,O}$ is the following direction cosine matrix

$$R_{S,O}(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

being β the rotation angle between the rotating and inertial frame (see Figure 3.3). The time derivative of the rotation matrix is computed from kinematic relations such as (Zhao, 2016):

$$\dot{R}_{S,O} = R_{S,O}\,\tilde{\boldsymbol{\omega}}_{S,O}^{(S)} \tag{3.6}$$

where $\tilde{\boldsymbol{\omega}}_{S,O}^{(S)}$ is the skew-symmetric matrix of the vector $\boldsymbol{\omega}_{S,O}^{(S)}$, which is angular velocity vector between the synodic and the inertial frame expressed in the synodic frame. This vector is

$$\boldsymbol{\omega}_{S,O}^{(S)} = \begin{bmatrix} 0 \\ 0 \\ N \end{bmatrix} \tag{3.7}$$

where N is the angular velocity of the synodic system. And the corresponding skew-symmetric matrix is

$$\tilde{\boldsymbol{\omega}}_{S,O}^{(S)} = \begin{bmatrix} 0 & -N & 0 \\ N & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

To compute the inverse transformation, *i.e.* from the inertial to the synodic frame, the rotation matrix and its derivative are the following

$$R_{O,S} = R_{S,O}^{\mathsf{T}} \tag{3.9}$$

$$\dot{R}_{O,S} = -\tilde{\boldsymbol{\omega}}_{S,O}^{(S)}\,R_{O,S} \tag{3.10}$$

The rotation angle β can be found from insight into the transformation. The x-axis of the synodic frame points away from the Sun, so angle β results from computing the arctangent of the -x and -y coordinates of the Sun's location in the EMO2000 frame. MATLAB's function *atan2* is used to resolve the quadrant.

3.3. State Model

The previous section has dealt with the choice of a reference where to describe the motion of an object in space. In this section, the different options for describing the motion within a frame are explored.

In astrodynamics, the means to describe the motion of a body in a reference frame is through a state model. This state model describes the motion of an object within the equations of motion of the system. The two most simple and extended forms of state models are the Cartesian position-velocity state and the Keplerian state. The appropriate selection of a state model has a great impact on the results of numerical propagation and can have a strong effect on the accuracy of the computed solution. Although there are complex representations tailored to specific cases, the two most appropriate state models for this thesis are the Keplerian (for describing the heliocentric position of TCOs before entering Earth's Hill sphere) and Cartesian (for all simulations run in this thesis). Both are subsequently introduced.

3.3.1. Keplerian state

A Keplerian state representation comprises a set of six parameters that completely define the orbit of an object under a two-body trajectory with respect to a central body in a non-rotating reference frame centered at the primary body (Curtis, 2020).

The six parameters used are known as orbit elements and are :

a: Semi-major axis
e: Eccentricity
i: Inclination
Ω: Right Ascension of the Ascending Node (RAAN)
ω: Argument of periapsis
θ: True anomaly

The semi-major axis a is defined as half the distance from the periapsis r_p (point of closest approach) to the apoapsis r_a (furthest orbital point). The eccentricity is defined as $e = (r_\mathrm{a} - r_\mathrm{p})/(r_\mathrm{a} + r_\mathrm{p})$. The inclination i is the angle between the central body's reference plane and the orbit's plane. The intersection between both planes defines the line of nodes, with the main direction labeled as the vernal point or the vernal direction. The RAAN Ω is the angle measured along the reference plane of the vernal point and the ascending node. The argument of periapsis is the angle measured along the orbital plane between the ascending node and the periapsis. Finally, the true anomaly θ is the angle measured along the orbital plane between the periapsis and the body's position. An illustration of the orbital elements is presented in Figure 3.4.

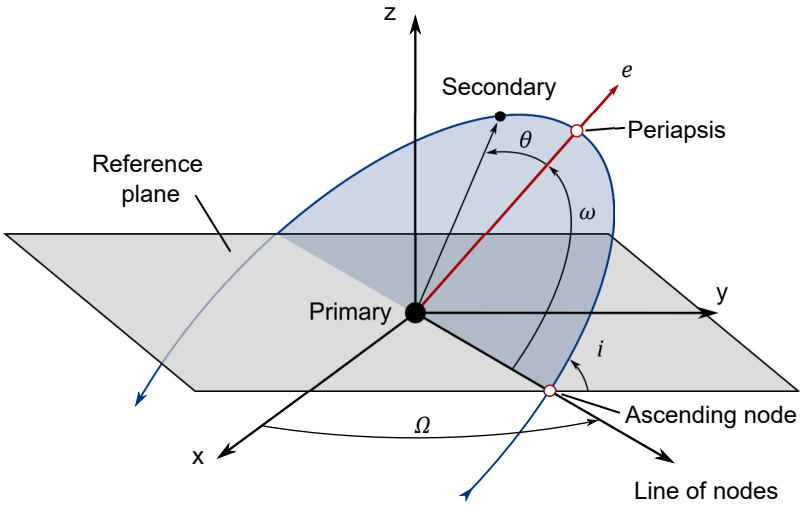


Figure 3.4: Graphical representation of a number of orbital elements used for the Keplerian state representation.

As previously mentioned, this representation comes into play in this thesis work when retrieving information about TCOs in their heliocentric trajectories. However, this representation is not used in the propagation of the equations of motion since it is only suitable for applications when only the central gravity from one massive object is considered.

3.3.2. Cartesian state

The most common state representation, and the one that is used in every numerical propagation in this thesis, is the position-velocity Cartesian state representation.

In this representation, the state of the system that is propagated is a column six-dimensional vector $\mathbf{s}$, which contains the Cartesian position and velocity of the propagated body at a given epoch in time. That is

$$\mathbf{s} = \left[\mathbf{r}, \mathbf{v}\right]^T = \left[x, y, z, v_x, v_y, v_z\right]^T \tag{3.11}$$

An illustration of the Cartesian position-vector state is presented in Figure 3.5.

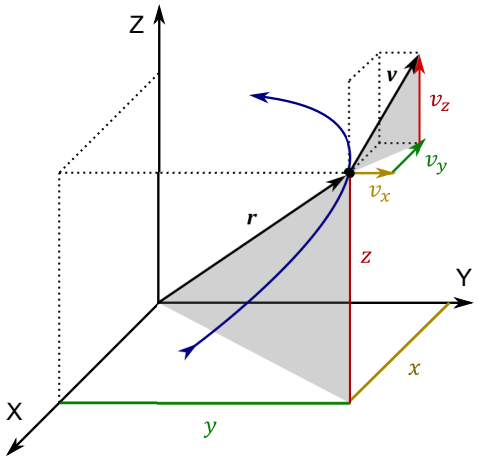


Figure 3.5: Graphical representation of the position-velocity Cartesian state vector.

3.4. Dynamics Models

Dynamics models are used to describe the motion of a spacecraft in space, and they are selected such that they resemble as much as desired the force environment to which it is exposed. It is the object of this section to survey candidate dynamics models and select those that describe the physics of the problem up to the desired accuracy while simplifying the simulation process as much as possible.

3.4.1. Gravity Models

The first dynamics models that are discussed are those that model the gravity that the bodies undergo since they are the cause of the primary acceleration that defines the trajectory of objects in space. The most simple model used in astrodynamics is the two-body problem, where the motion of a spacecraft is defined by the gravitational acceleration of a primary body. This model is convenient for mission design, as the equations of motion can be analytically solved for a set of initial conditions. However, the assumption of only a central-body acceleration term can only be considered in very specific cases where other accelerations are neglected. This can be the case for orbits in close proximity to the Earth.

The motion of 2006 RH120 within Earth's Hill sphere follows a seemingly chaotic motion influenced by the gravity of many bodies, such as the Sun, the Earth, and the Moon. Once a third body is introduced in the dynamics model, the differential equation of the system does not have an analytical solution

anymore, and the trajectory needs to be propagated using numerical methods.

This subsection explores a set of gravity model formulations that have the potential to be useful for the present research. These models are subsequently introduced.

Many-body problem

The many-body problem is a gravity field model that allows including the gravitational accelerations of as many bodies as desired in the equations of motion.

The equation that describes the motion of a massless body subject to the attraction of n bodies is the following.

$$\ddot{\mathbf{r}}_i = \sum_{j \neq i}^{n} G \frac{m_j}{r_{ij}^3} \mathbf{r}_{ij} \tag{3.12}$$

This vectorial differential equation describes the motion of body i in a reference frame with inertial origin and orientation. Here, $\mathbf{r}_i$ is the position of body i and $\mathbf{r}_{ij}$ is the position of body j relative to body i.

When the motion of body i wants to be computed relative to another body k (such as the Earth), the expression adopts the following shape (Wakker, 2015).

$$\ddot{\mathbf{r}}_{ki} = -G \frac{m_i + m_k}{r_{ki}^3} \mathbf{r}_{ki} + G \sum_{j \neq i,k}^{n} m_j \left(\frac{\mathbf{r}_{ij}}{r_{ij}^3} - \frac{\mathbf{r}_{kj}}{r_{kj}^3} \right) \tag{3.13}$$

This model is mainly used in this thesis when propagating orbits in a high-fidelity model using the Tudat software, TU Delft's in-house astrodynamics toolbox for orbit propagation. The locations of each one of the bodies included are obtained from precise ephemeris libraries retrieved from JPL's SPICE toolbox.

CR3BP

The Circular Restricted Three-Body Problem (CR3BP) is a dynamics model used to describe the motion of a massless particle in the presence of two massive bodies, such as that of a spacecraft or TCO in the Sun-Earth system. In this model, both primaries move according to their mutual gravitational interaction, revolving about the barycenter of the system following co-planar circular orbits (Wakker, 2015).

The equations of motion of the spacecraft are described in the synodic frame of the primaries. It is convenient to express the equations in non-dimensional form by normalizing the units of the system. For such, the constant μ is introduced as

$$\mu = \frac{m_2}{m_1 + m_2} \tag{3.14}$$

where m_1 and m_2 represent the masses of the larger and smaller primary, respectively. The units are normalized by the following measures

Unit of mass: Sum of the mass of the primaries $m_1 + m_2$

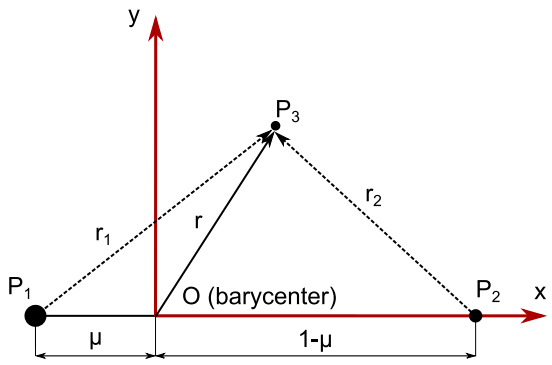


Figure 3.6: Graphical representation of the Circular-Restricted Three-Body Problem in dimensionless form.

Unit of length: The distance between the primaries r_{12}
Unit of time: Inverse of the rotation rate of the system $1/N$

Using this dimensionless representation, the angular velocity ω is equal to one. A graphical representation of the system with the relevant distances in dimensionless form is presented in Figure 3.6.

The equations of motions of the CR3BP in dimensionless units and synodic reference frames are given by (Wakker, 2015)

$$\ddot{x} = 2\dot{y} + x - (1-\mu)\frac{x+\mu}{r_1^3} - \mu\frac{x-1+\mu}{r_2^3} \tag{3.15a}$$

$$\ddot{y} = -2\dot{x} + y - (1-\mu)\frac{y}{r_1^3} - \mu\frac{y}{r_2^3} \tag{3.15b}$$

$$\ddot{z} = -(1-\mu)\frac{z}{r_1^3} - \mu\frac{z}{r_2^3} \tag{3.15c}$$

where the distance of the third body to the primaries is defined by

$$r_1^2 = (x+\mu)^2 + y^2 + z^2 \tag{3.16a}$$

$$r_2^2 = (x-1+\mu)^2 + y^2 + z^2 \tag{3.16b}$$

At this point, the potential function U is introduced as

$$U = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{3.17}$$

which allows to rewrite the equations of motion in a more compact form as

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x} \tag{3.18a}$$

$$\ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y} \tag{3.18b}$$

$$\ddot{z} = \frac{\partial U}{\partial z} \tag{3.18c}$$

It should be noted that U is a potential function that includes gravitational and centrifugal terms, resulting in a field that is non-central, *i.e.* it does not point towards the center of the planetary bodies. Furthermore, the force field is conservative, as it does not explicitly depend on time due to the fixed positions of the primaries in the rotating frame.

Furthermore, the equations of motion of the spacecraft described above only depend on the constant μ, which is also called the three-body constant. A complete derivation of the equations of motion can be found in Wakker, 2015.

One useful property of the CR3BP is that, although the motion of a body within the model does not have an analytical solution, it allows for an integral of motion, known as Jacobi's integral C and defined as

$$C = 2\,U - V^2 \tag{3.19}$$

where V represents the velocity of the spacecraft in the rotating frame and C is Jacobi's constant. This expression gives an analytical relation between the velocity and position of a spacecraft in the CR3BP, which is useful for trajectory design purposes. A spacecraft with a particular Jacobi constant is bounded to move within a specific region, limited by the zero-velocity curves found by setting the velocity of the spacecraft equal to zero in Equation 3.19.

Hill Model

The Hill model is an approximation of the CR3BP where the center of coordinates is shifted to the lighter body(m_2), and the ratio between the masses of lighter and heavier bodies is equal to zero. The resulting equations of motion are the following (Villac & Scheeres, 2003)

$$\ddot{x} = 2N\dot{y} + 3N^2x - (\mu_m/r^3)x \tag{3.20a}$$

$$\ddot{y} = -2N\dot{x} - (\mu_m/r^3)y \tag{3.20b}$$

$$\ddot{z} = -N^2z - (\mu_m/r^3)z \tag{3.20c}$$

where N is the angular motion, r is $\sqrt{x^2 + y^2 + z^2}$, and μ_m is the gravitational parameter of m_2, *i.e.* $\mu_m = G\,m_2$.

The corresponding libration points L_1 and L_2 are located at

$$x_{L_{1/2}} = \pm\left(\frac{\mu_m}{3N^2}\right)^{1/3} \tag{3.21}$$

and $y = z = \dot{x} = \dot{y} = \dot{z} = 0$.

The system can be normalized by taking the unit length as $l = (\mu_m/N^2)^{1/3}$ and the unit time as $\tau = 1/N$. This results in the following non-dimensional equations

$$\ddot{x} = 2\dot{y} + 3x - x/r^3 \tag{3.22a}$$

$$\ddot{y} = -2\dot{x} - y/r^3 \tag{3.22b}$$

$$\ddot{z} = -z - z/r^3 \tag{3.22c}$$

which formally is the result of taking $N = 1$ and $\mu_m = 1$.

These equations of motion can also be expressed in terms of a gradient of a potential function W (Szebehely, 1967) defined as

$$W = \frac{1}{2}\left(3x^2 - z^2\right) + 1/r \tag{3.23}$$

resulting in

$$\ddot{x} - 2\dot{y} = \frac{\partial W}{\partial x} \tag{3.24a}$$

$$\ddot{y} + 2\dot{x} = \frac{\partial W}{\partial y} \tag{3.24b}$$

$$\ddot{z} = \frac{\partial W}{\partial z} \tag{3.24c}$$

The first coordinate of the first and second libration points results in

$$\hat{x}_{L_{1/2}} = \pm\left(\frac{1}{3}\right)^{1/3} \tag{3.25}$$

Similarly to the CR3BP, the Hill model has an integral or Jacobi-like constant Γ, which is computed as

$$\Gamma = 2W - V^2 \tag{3.26}$$

where $V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$.

The Hill model is the dynamics model used in this thesis to propagate the trajectories of the spacecraft and the TCOs within Earth's Hill Sphere. Urrutxua et al., 2015 states that the trajectory of the TCO within the Hill sphere is extremely sensitive to entry conditions from the Heliocentric phase. However, Figure 3.7 shows that, once inside the Hill sphere, the Hill model reproduces the trajectory of the TCO with enough reliability for this preliminary analysis. The model has a certain deviation from the ephemeris data, but the propagation qualitatively represents the trajectory of the TCO.

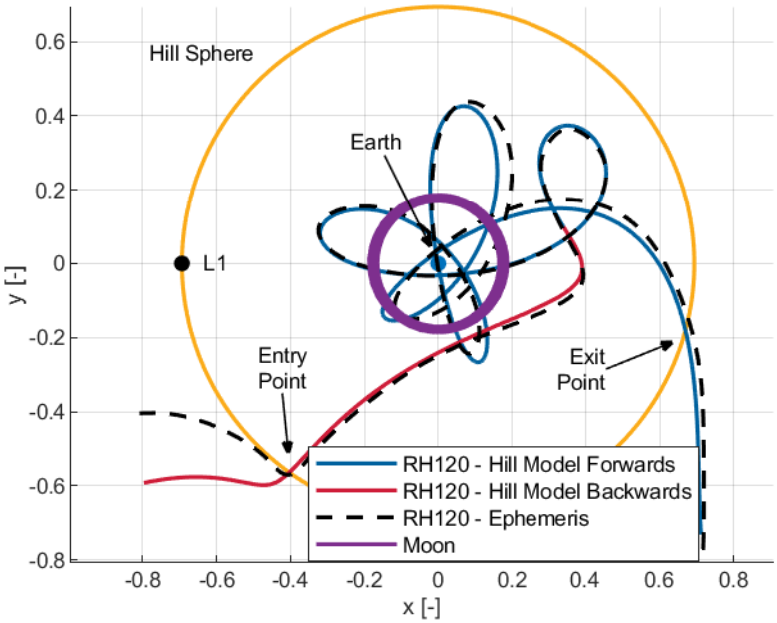


Figure 3.7: Comparison between ephemeris of TCO 2006 RH120 and propagation using Hill model projected into the x-y axis.

3.4.2. Perturbations

Objects that travel in space can experience accelerations other than the central gravity of the nearby bodies. These accelerations are regarded as perturbations to the orbits and can further complicate the equations of motion of propagated objects. Several perturbations can be found in the environment comprised inside Earth's Hill sphere. The most common are subsequently analyzed.

- **Irregular gravity**: The shape and internal mass distribution of the objects exerting gravity is not perfectly spherical and homogeneous. Therefore, for some applications, the point-mass gravity model can be inaccurate. The irregularities in a body's gravity field can be modeled through spherical harmonics. They allow expressing the gravitational potential of the body as a sum of spherical terms that, depending on the number of coefficients selected and their values, result in a better approximation of the gravity field generated by the irregular shape of the planet. The irregular shape of gravitational objects becomes important in close-proximity applications; for example, in two-body trajectories around the Earth where the oblateness of the Earth makes Keplerian orbits drift or orbits around small asteroids where the motion is mainly described by their irregular gravity. In this thesis, however, the trajectories are distant enough from gravitational bodies (the Earth and Sun) that these effects can be neglected.

- **Solar radiation pressure**: The Solar Radiation Pressure (SRP) is the force that results from the exchange of momentum between electromagnetic waves coming from the Sun and the different faces of a spacecraft or object. The force is therefore directed in the opposite direction to the Sun. Modeling this force accurately is complex, as it depends on the shape of the object and its reflectivity, but a first-order approximation can be found in Wakker, 2015. Apart from the aforementioned shape characteristics of the object undergoing the SRP, the magnitude of the acceleration also depends on the energy flux at the position with respect to the Sun (the further from the Sun, the lower) and the mass of the object. Previous studies that focus on designing transfer trajectories to TCOs have ignored the effect of SRP. Urrutxua et al., 2015 mentions that the effect of SRP is strong enough to perceptibly perturb the motion of 2006 RH120, while the effect is still left out of the analysis. Takahashi et al., 2022 includes the effect of SRP in their model to rendezvous with 2006 RH120 autonomously. Wakker, 2015 indicates that the SRP acceleration in the Earth-Moon environment can reach an order of magnitude of $0.1\,\mathrm{mm\,s^{-2}}$ for large light satellites. This value has a similar order of magnitude as the thrust generated by the electric propulsion engine and may have an impact on both the transfer shape and the cost. However, SRP is neglected in the present study on preliminary transfers to 2006 RH120 for simplicity. However, this limitation in the model is noted and left as a recommendation to assess its impact in future work where more accurate models are used.

- **Atmospheric Drag**: The third common perturbing acceleration found in the objects that move in the Earth-Moon vicinity is the drag originating from interaction with the Earth's atmosphere. According to Wakker, 2015, the effect of atmospheric drag decreases rapidly with altitude, and it can be neglected in most applications for altitudes above 1000 km. Therefore, given that the transfers studied do not originate or pass by low Earth orbit, the effect of atmospheric drag is neglected in the present study.

3.4.3. Thrust Model

Another force that is commonly modeled in astrodynamics is that of the thrust of the engine. For conventional chemical engines, the thrust can be modeled as independent discrete thrusting events that do not influence the coasting motion of the spacecraft in space. However, low-thrust engines exert continuous thrust over large arcs of the spacecraft's trajectory and must be modeled.

To model the continuous thrust of a low-thrust engine, the approach presented by Sims and Flanagan, 1999 for the preliminary design of low-thrust interplanetary missions will be used. The continuous thrust is modeled as a series of delta-v impulses. The trajectory is discretized into several segments, with an impulsive delta-v applied at the midpoint of the segment. The propagation between impulses follows the dynamics model selected. The model becomes a good representation of the continuous-thrust nature when a sufficient number of arcs is considered, as will be discussed in Section 6.1.

The delta-v impulses can be related to propellant consumption using Tsiolkovsky's rocket equation

$$\Delta V = c_{\text{eff}} \log \frac{m_0}{m_{\text{f}}} \tag{3.27}$$

where m_0 is the initial wet mass of the spacecraft and m_{f} is the final mass after the propellant has been consumed. g_0 is the gravitational acceleration at Earth's surface, and c_{eff} is the effective exhaust velocity of the propellant. c_{eff} is computed as the ratio between the engine's thrust (F) and the mass flow ($\dot{m}$)

$$c_{\text{eff}} = \frac{F}{\dot{m}} = I_{\text{sp}}\, g_0 \tag{3.28}$$

The parameter I_{sp} represents the engine's specific impulse, and it is commonly used to measure the efficiency of a propulsion system.

The exhaust velocity in an electric propulsion system is limited by the available power supplied by the power source P_{source}, and they are related via the following relationship (Wakker, 2015):

$$\epsilon\, P_{\text{source}} = \frac{1}{2}\dot{m}\, c_{\text{eff}}^2 = \frac{1}{2}\, F\, I_{\text{sp}}\, g_0 \tag{3.29}$$

where ϵ is the power conversion efficiency.

I_{sp} tends to be a specification of the engine and will be assumed constant for the simulations conducted in this thesis. The thrust will, however, be assumed variable, and the means of practically acting on the thrust level is varying the mass flow.

4

Numerical Tools

This chapter presents the numerical tools that are used in the present research. First, root-finding and numerical differentiation techniques are introduced. Then, a large part of the chapter is devoted to the selection of the numerical integration tool used to propagate the trajectories, placing the focus on meeting the accuracy requirement also introduced. Similarly, a numerical interpolator is selected to obtain states at intermediate epochs of propagated trajectories. Then, the state transition matrix is introduced, which plays an important role in the present research. Next, the differential-corrector algorithm is presented, which allows for obtaining the first estimates of transfers that meet boundary conditions. Finally, the numerical optimization tool used for the thesis is presented and explained.

4.1. Root Finding

Numerical root-finding methods are essential in the intermediate steps of this thesis to find the minimum of a function by equating to zero the partial derivative of a function with respect to the variable under study.

The root-finding method used in this thesis is the Newton-Raphson method, which finds the root of a function $f(x)$ iteratively. The algorithm is started with an initial guess for the root x_0, and this guess is updated according to the following expression (Sanchez Ruiz & Lengua Fernandez, 2008):

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \tag{4.1}$$

In this thesis, this method is generally used to find the minimum of a cost function J with respect to the decision variables. The accuracy of the method can be tuned by selecting a tolerance. Once the updated solution is smaller than a certain tolerance, the iteration process is stopped.

4.2. Numerical Differentiation

Numerical methods for differentiation are used throughout this thesis to compute the derivatives of functions that are otherwise too intricate to compute analytically.

The method used to compute these derivatives numerically is the so-called *finite-differences* method. Specifically, the *centered-difference* method will be employed. In this method, the derivative of a function $f(x)$ with respect to x is computed numerically as (Lewis et al., 2022):

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\,\Delta x} \tag{4.2}$$

Similarly, the method can be used to obtain the second derivative of the function, such as

$$f''(x) = \frac{f(x + \Delta x) - 2\,f(x) + f(x - \Delta x)}{\Delta x^2} \tag{4.3}$$

These are called second-order methods as they have an error proportional to Δx^2 or $O(\Delta x^2)$.

In this thesis, numerical differentiation through finite differences is extensively used. First, its usage is fundamental in the early stages of the optimization algorithm development to compute the gradient of the cost function and constraints. The numerical differentiation quickly proved to be very demanding in terms of computing power. Still, its usage is fundamental to running preliminary tests to ensure the problem was properly formulated before attempting to find the analytical solution. At the same time, they were used to verify that the gradients obtained were correct. Finite differences are also used in the verification process of the state transition matrix.

4.3. Numerical Integration

Numerical integration is a fundamental part of the trajectory-propagation process. It is required for solving the differential equations that govern the motion of the bodies that are propagated. Since these equations of motion are modeled by the Hill model, they do not have an analytical solution available and have to be numerically integrated.

There are many numerical integration algorithms to choose from that can be used for solving the equations needed for this thesis. Among the list of algorithms that are available in MATLAB, a selection is tested to check their computing performance and numerical accuracy for the problem at hand.

The first step in the process of selecting an integrator is to define the numerical accuracy that is considered acceptable for the problem. It is assumed that, for this preliminary analysis, we are interested in the qualitative results of the study. Therefore, the requirements for the accuracy of the solution are not stringent. A numerical accuracy of the order of 10 m is considered reasonable for the present study. It is the order of magnitude of the size of the TCOs, and it is assumed that for more accurate rendezvous, the spacecraft would be equipped with Guidance, Navigation, and Control (GNC) equipment. Regarding the minimum accuracy in velocity, it is set to 1 mm/s since it is six orders of magnitude smaller than the delta-v cost of a very efficient transfer of 100 m/s (Brelsford et al., 2016).

Another important aspect of the integrator selection is that it should generate trajectories in a reasonable computation time. There is no hard value for the computation time required for the integrator,

but runtime will be checked during the problem simulations in later steps to check if it is reasonable to perform the necessary analysis to complete the research objective.

Once the desired performance of the integrator has been set, different available algorithms and settings are studied to match these requirements. MATLAB has several built-in functions to solve Ordinary Differential Equations (ODEs) based on different numerical-integration algorithms. Among them, those with medium to high accuracy have been selected. Their characteristics are summarized in Table 4.1.

Table 4.1: List of MATLAB ODE solvers and their characteristics (The MathWorks Inc., 2023a).

Function	Accuracy	Algorithm	Remarks
ode45	Medium	Variable step size explicit Runge-Kutta (4,5) Dormand Prince Pair.	MATLAB documentation recommends it as the first solver to try.
ode113	Low to High	Variable step variable order integrator of the Adams-Bashford-Moulton family with order ranging from one to 13.	May be more efficient than *ode45* at problems with stringent tolerances and costly function evaluations.
ode78	High	Variable step-size explicit Runge-Kutta 8(7) pair advanced with the 8th-order result with continuous 7th order Verner's extension.	More efficient than *ode45* at problems with high accuracy requirements.
ode98	High	Variable step-size explicit Runge-Kutta 9(8) pair advanced with the 9th-order result with continuous 8th order Verner's extension.	More efficient than *ode78* integrating long time intervals with stringent tolerances.

To compare the different integrators, a reference trajectory is selected as representative of the future simulations of this thesis. For such, the trajectory of the reference TCO RH120 has been chosen. The initial epoch used is 2006, October 1, 0:00:00 Universal Time Coordinated (UTC), and its state was obtained in the Hill model from Takahashi et al., 2022 and summarized Table 4.2. The propagation time is set to 300 days. This reference trajectory has been selected since the environment of TCOs trajectory resembles that of the transfer trajectory that will be explored later on, and 300 days are long enough to cover the range of transfer times that are analyzed in further sections.

To measure the numerical error of the algorithm, a benchmark solution is generated with an integrator of high order and stringent tolerance. For this benchmark, the function *ode89* with a tolerance of 10^{-14} is used. The performance of the remaining algorithms from Table 4.1 are measured against this benchmark solution for different tolerance values. The results of this analysis for the position error

Table 4.2: Initial conditions for reference trajectory in Hill frame from Takahashi et al., 2022.

Epoch	2006 October 1 | JD 2,454,009.5
x	738504.1118393708 km
y	213454.0880487227 km
z	-955911.2794879418 km
v_x	-0.1342568737871429 km/s
v_y	0.21860028536472628 km/s
v_z	-0.46910677002491197 km/s

are shown in Figure 4.1.

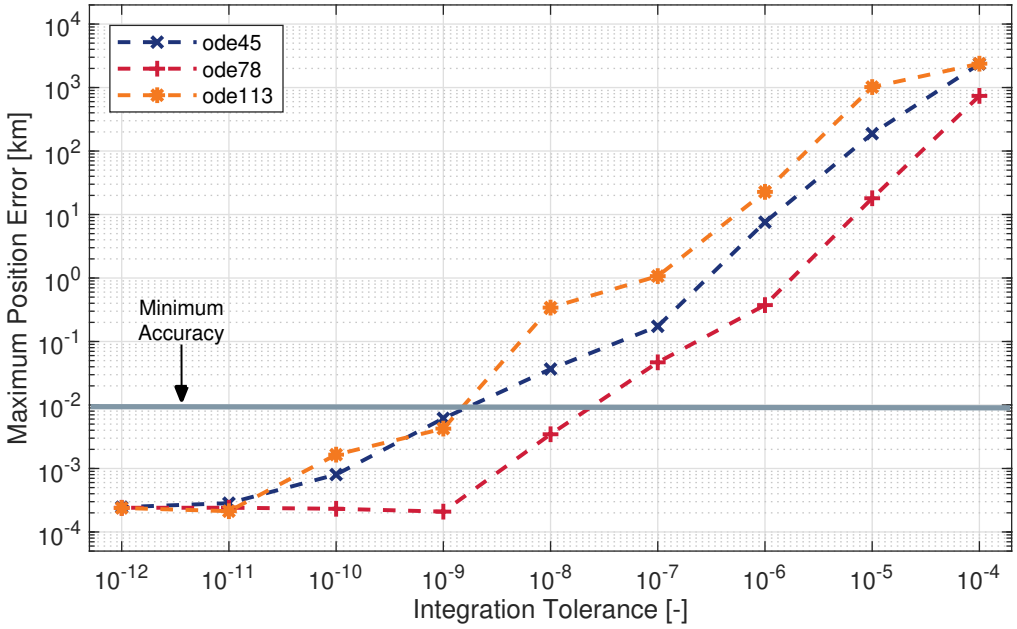


Figure 4.1: Position error of different integrator configurations.

Figure 4.1 shows that any integrator with a tolerance of 10^{-9} or better achieves a numerical accuracy for the position below the minimum established of 10 meters. The most accurate algorithm is *ode78* for every tolerance selected. However, based on these results, any of the three integrators are suitable for this thesis work. It can be seen from the graph that for tolerances below 10^{-11}, the error of the benchmark may have been reached at a value of 0.2 meters, which is settled for reduced tolerances not shown in the graph. It should be noted that the velocity error also meets the required accuracy of 1 mm/s with every integrator for tolerances below 10^{-9}, reaching a minimum of 10^{-5} m/s for the most stringent tolerances.

To assess the computational cost of the different algorithms, the number of function evaluations has been recorded and plotted in Figure 4.2. It is observed that the *ode113* function computes the solution with the least number of function evaluations. The slowest integrator is *ode45*, with a steep exponential increase in runtime as the tolerance is reduced compared to the other integrators.

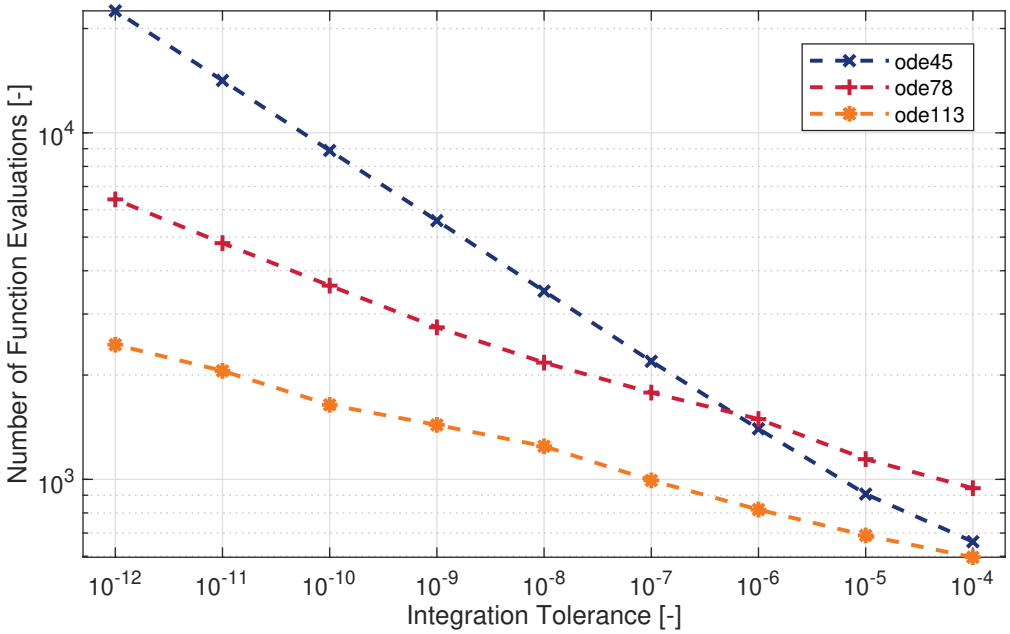


Figure 4.2: Function evaluations of different integrator configurations.

The integrator with the best overall performance for the reference trajectory is the *ode78*, which has a promising trade-off between accuracy and speed. It will be used for the main results of this thesis, and in case the speed is a problem, *ode113* will be tested. For some preliminary simulations where speed is not a concern, *ode45* is used, as it meets the accuracy requirements (see Figure 4.1) and was used before performing this analysis. Finally, the tolerances are initially set to 10^{-12} to ensure accuracy is below requirements. In case speed becomes a problem, it can be reduced up to 10^{-9} while still complying with the requirements.

4.4. Interpolation

To be able to compare trajectories that overlap in time in the same discrete time epochs, a numerical interpolation method is used. This thesis uses the MATLAB built-in function *interp1*. This function takes as input a vector of samples and their corresponding values and another vector of samples of which the corresponding interpolated values are returned. Within MATLAB's *interp1* function, several different interpolation methods are available. Three have been tested to check their accuracy for the problem at hand. These are summarised in Table 4.3.

The benchmark solution from the previous section has been used to compare and evaluate the performance of the interpolation methods. This benchmark solution has $100,000$ intermediate time epochs in the propagation. The same trajectory has been integrated using *ode78*, resulting in a trajectory with $3,000$ epochs. The last trajectory has been interpolated and evaluated at the sample times of the benchmark solution. The results are compared and reflected in Figure 4.3.

It is observed from Figure 4.3 that the only interpolation method that meets the accuracy requirements of 10 meters is the *spline* method. It is therefore selected for the interpolations carried out in this

Table 4.3: List of MATLAB interpolation methods tested (The MathWorks Inc., 2023b).

Method	Description
linear	Linearly interpolates the values at the neighboring points. At least two points are required.
makima	Based on Modified Akima cubic Hermite interpolation. Interpolation is performed through the piece-wise function of polynomials with a degree of at most three. At least two points are required.
spline	Based on Spline interpolation using not-a-knot end conditions. Based on a cubic interpolation neighboring points. At least four points are required.

thesis.

4.5. State Transition Matrix

The State Transition Matrix (STM) is an insightful tool that maps variations in the initial state of a system to state variations at a later time. The STM provides information that is fundamental for the present thesis. Its usage is two-fold: first, it is used within a differential-correction algorithm explained in Section 4.6 to find the departure state that allows achieving a target state in trajectory propagation. The second useful application of the STM for this thesis is in calculating the gradient of the cost function and constraints used to find the optimum trajectories. It will be explained in Section 6.3 how the components of the STM are related to the analytical gradient of these functions.

A general definition for the state transition matrix is the following (Heiligers, 2020):

$$\Phi(t, t_0) = \frac{\partial \mathbf{s}(t)}{\partial \mathbf{s}(t_0)} \tag{4.4}$$

For a six-element position-velocity Cartesian state, the STM has a 6×6 dimension and results in the following expression

$$\Phi(t, t_0) = \begin{bmatrix} \Phi_{\mathbf{rr}_0} & \Phi_{\mathbf{rv}_0} \\ \Phi_{\mathbf{vr}_0} & \Phi_{\mathbf{vv}_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{r}}{\partial \mathbf{v}_0} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{r}_0} & \frac{\partial \mathbf{v}}{\partial \mathbf{v}_0} \end{bmatrix} \tag{4.5}$$

where the partial derivatives are evaluated at time t for a given set of initial conditions.

The STM of a non-linear system is computed numerically by propagating the following expression alongside the system's equations of motion.

$$\dot{\Phi}(t, t_0) = A(t)\, \Phi(t, t_0) \tag{4.6}$$

where the initial conditions for the STM $\Phi(t_0, t_0)$, equal the identity matrix. The matrix $A(t)$ is the Jacobian of the linearized system and is computed from

$$\delta \dot{\mathbf{s}}(t) = A(t)\, \mathbf{s}(t) \tag{4.7}$$

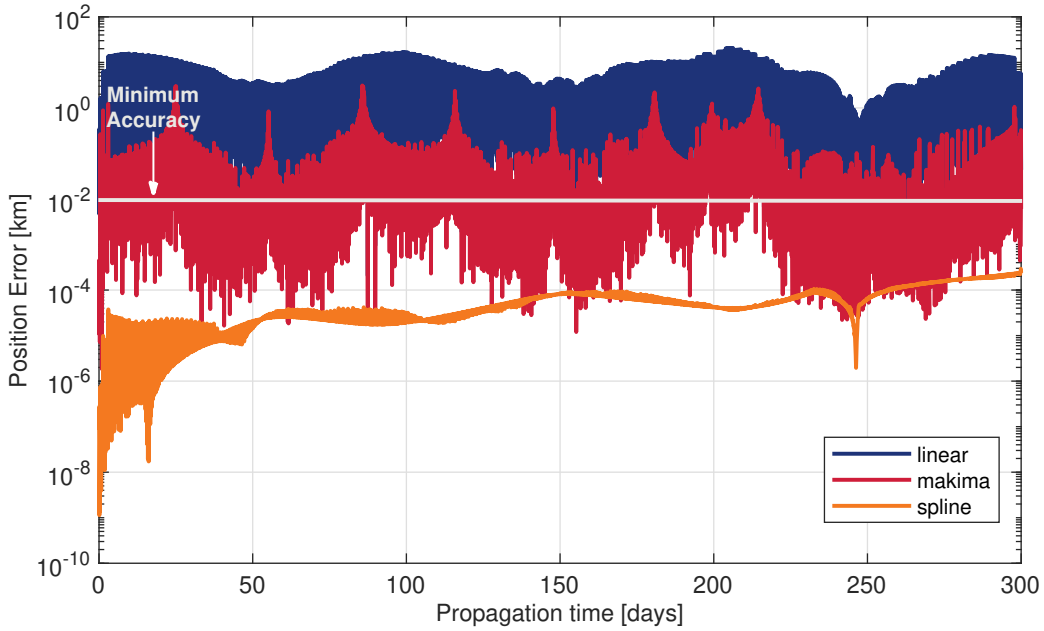


Figure 4.3: Interpolator methods comparison.

For the Hill three-body problem, this Jacobian is found by taking the analytical partials of the equations of motion of the model (Equation 3.22), resulting in

$$A = \frac{\partial \dot{\mathbf{s}}}{\partial \mathbf{s}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial y} & \frac{\partial \ddot{x}}{\partial z} & 0 & 2 & 0 \\ \frac{\partial \ddot{y}}{\partial x} & \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial z} & -2 & 0 & 0 \\ \frac{\partial \ddot{z}}{\partial x} & \frac{\partial \ddot{z}}{\partial y} & \frac{\partial \ddot{z}}{\partial z} & 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

where the partials are evaluated alongside the trajectory and are

$$\frac{\partial \ddot{x}}{\partial x} = 3 - \frac{1}{r^3} + \frac{3x^2}{r^5} \tag{4.9a}$$

$$\frac{\partial \ddot{x}}{\partial y} = \frac{3xy}{r^5} \tag{4.9b}$$

$$\frac{\partial \ddot{x}}{\partial z} = \frac{3xz}{r^5} \tag{4.9c}$$

$$\frac{\partial \ddot{y}}{\partial x} = \frac{3xy}{r^5} \tag{4.9d}$$

$$\frac{\partial \ddot{y}}{\partial y} = -\frac{1}{r^3} + \frac{3y^2}{r^5} \tag{4.9e}$$

$$\frac{\partial \ddot{y}}{\partial z} = \frac{3yz}{r^5} \tag{4.9f}$$

$$\frac{\partial \ddot{z}}{\partial x} = \frac{3xz}{r^5} \tag{4.9g}$$

$$\frac{\partial \ddot{z}}{\partial y} = \frac{3yz}{r^5} \tag{4.9h}$$

$$\frac{\partial \ddot{z}}{\partial z} = -1 - \frac{1}{r^3} + \frac{3z^2}{r^5} \tag{4.9i}$$

4.6. Differential Correction

Differential Correction (DC) is a numerical algorithm used in this thesis to generate a desired trajectory that meets certain boundary constraints, e.g. finding the departure conditions that allow a spacecraft to reach a specific target.

At the core of the DC algorithm is the correction of the departure velocity of a given propagation. The DC algorithm is initialized with given initial conditions (departure location and estimated departure velocity) and propagated for a specified transfer time T. The final position after the propagation is stored and compared with the desired target position. The STM is then used to compute the required change in departure velocity such that the target is reached.

The formula for the initial velocity correction that is the backbone of the DC algorithm is the following (Heiligers, 2020):

$$\delta\mathbf{v} = [\Phi_{\mathbf{r},\mathbf{v}_0}(T,\, t_0)]^{-1}(\mathbf{r}_{\text{tgt}} - \mathbf{r}^*_{\text{arr}}) \tag{4.10}$$

where $\Phi_{\mathbf{r},\mathbf{v}_0}(T,\, t_0)$ is a portion of the system's STM (see Equation 4.5). $\mathbf{r}_{\text{tgt}}$ is the desired target position, and $\mathbf{r}^*_{\text{arr}}$ is the final arrival position of the spacecraft after propagation time T, from a departure point $\mathbf{r}_{\text{dep}}$ and estimated initial velocity $\mathbf{v}^*_{\text{dep}}$. More information on the derivation of the algorithm is presented in Appendix A.

Since the DC algorithm relies on an approximation by Taylor series expansion, the correction process must be iterated until the target position error falls below a specified tolerance, set to 10 meters according to the specifications presented in Section 4.3. In some cases, specifically when a particular iteration makes the propagation pass very close to the Earth, the algorithm does not converge to the target.

In the present work, the DC algorithm is used to generate two-impulse transfers connecting a departure location with a target position. These trajectories are used in Chapter 8 first to analyze potential

rendezvous locations in the trajectory of 2006 RH120, and later on, it is used to generate initialization trajectories for the low-thrust optimization algorithm consistently.

4.7. Numerical Optimization

As explained in Section 2.3, there are many numerical optimization methods used to solve a trajectory optimization problem. This thesis follows a direct approach, which is solved with a gradient-based method. The tool that implements this optimization algorithm is MATLAB's built-in function *fmincon*, which allows solving single-objective non-linear constrained optimization with an interior-point algorithm.

As part of the direct optimization approach, the problem is transcribed so that a finite set of decision variables fully defines a low-thrust transfer. The transcription of the problem is explained in Section 6.2. The cost function defined for the problem is a function of these decision variables. At the same time, the constraints are also a function of the decision variables, and the trajectory optimization problem consists of finding the set of decision variables that minimize the cost function while meeting the constraints.

The function *fmincon* takes two main inputs: the two MATLAB functions that output the cost value and the non-linear constraints value for a set of decision variables input. The non-linear constraints are formulated as equality and inequality constraints, such as

$$\mathbf{g}_{\text{eq}} = \mathbf{0} \tag{4.11}$$

$$\mathbf{g}_{\text{ineq}} \leq \mathbf{0} \tag{4.12}$$

More information on the contents of the cost and constraint functions is provided in Section 7.1.2.

The function *fmincon* uses, by default, the finite-differences method to compute the gradient of the cost function as constraints. However, the cost and constraint functions can be extended such that, on top of evaluating the functions, they also return their gradient with respect to the decision variables. Computation of these analytical derivatives is one of the main advantages introduced in this thesis that allows for fast computation of low-thrust transfers and is explained in Section 6.3.

As optional inputs, *fmincon* also is fed with specific options for the algorithm, such as the pair value `'PlotFcn', @optimplotfval`, which automatically generates a plot of the optimization history, or `'DerivativeCheck', 'on'`, which allows checking that the analytical gradient computed is correct. Other settings can be tuned, such as the maximum number of function evaluations or iterations.

5

Two-impulse Transfer Design

To start the development of a low-thrust optimization algorithm, we begin the process by constructing simpler two-impulse optimal trajectories. The algorithm's complexity is built until the low-thrust approximation is introduced in Chapter 6. These first trajectories will allow us to gain insight into the types of transfers that are available connecting L1 with the target TCO and will provide the order of magnitude of the transfer delta-v. Developing this simpler two-impulse algorithm will allow us to gain experience with the software tools and test the performance of different problem formulations before getting into more complex simulations.

The chapter starts with connecting a departure and a target point with the DC method explained in Section 4.6. Then, a cost function is introduced into the problem, and the number of decision variables progressively increases. Finally, some conclusions and learnings regarding the trajectory optimization process are discussed, which will feed the development of the algorithm in the next chapter.

A common transfer configuration is selected to test the algorithms developed in this section and for illustration purposes. The transfer configuration consists of a departure from L1 and targets the TCO on 2007 February 11, 220 days after entry into Earth's Hill sphere. The simulation settings are summarized in Table 5.1.

Table 5.1: Transfer configuration for the two-impulse simulations.

Departure	Sun-Earth L1
Target epoch	JD 2,454,143.2 | 2007 February 11
Transfer Time	50 days

5.1. Differential Corrector

The DC algorithm introduced in Section 4.6 is used here to generate the first transfers that connect L1 with the TCO 2006 RH120. The algorithm treats the trajectory generation problem as a single-shooting scheme, which is illustrated alongside the used notation in Figure 5.1.

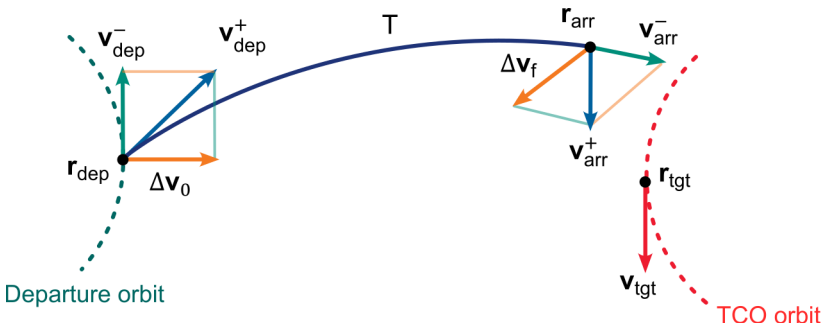


Figure 5.1: Illustration of single-shooting scheme for two-impulse trajectory generation with relevant variables.

The DC algorithm has been run for the transfer configuration presented in Table 5.1. As an initial velocity estimate, a vector pointing from departure to target is selected. The magnitude of the initialization velocity is computed as the quotient between the departure-target distance and the transfer time. The resulting generated trajectory is shown in Figure 5.2.

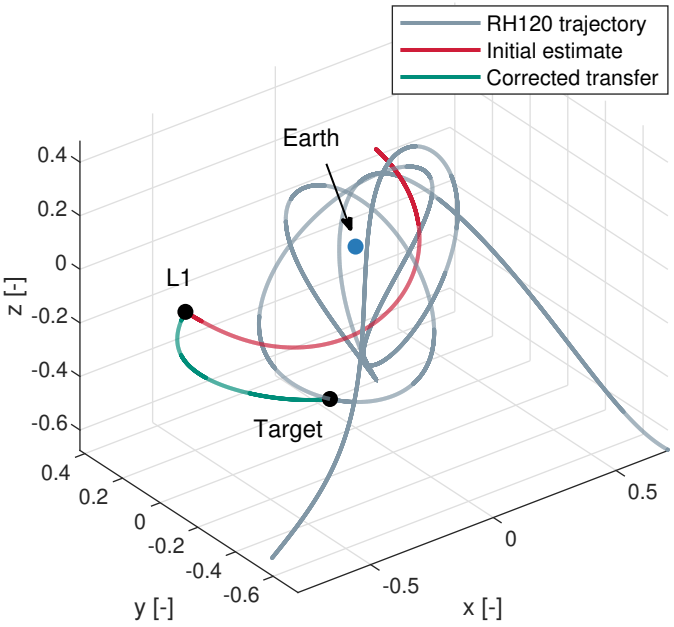


Figure 5.2: Two-impulse trajectory generated with the differential corrector.

It is observed in Figure 5.2 that the correction algorithm modifies the transfer trajectory to reach the target. The resulting total delta-v transfer cost is 1.0058 km/s. It is observed from Figure 5.2 that the final burn to rendezvous with the TCO results in an abrupt change in velocity direction. However, a

smoother transition will likely reduce the total delta-v. These smoother solutions will be explored as the optimization algorithm is introduced in the following sections.

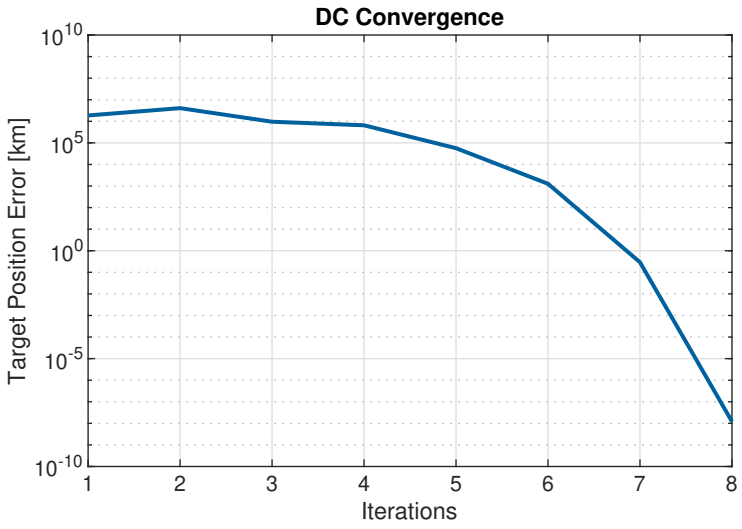


Figure 5.3: DC algorithm target position convergence.

The algorithm's convergence is shown in Figure 5.3. It is observed that the algorithm converges into the final solution in eight iterations until the position error falls below the tolerance of 10^{-2} km. It should be noted that, up to this point, the DC algorithm only converges into a trajectory that meets the constraints without introducing any cost function to minimize.

5.2. Introducing Cost Function

The algorithm in Section 5.1 can link a departure point to a target destination, but the trajectory generated is not optimized. The first approach to optimizing the trajectory for minimum delta-v is introducing a cost function to the two-impulse trajectory. MATLAB's *fmincon* function is used for such a task.

The problem is formulated using the single-shooting approach illustrated in Figure 5.1. As a first optimization simulation, the only decision variables are the three components of the initial velocity vector. Other design variables, such as the transfer time and target epoch, are fixed at this point. The cost function is the sum of the initial and final delta-v's, and the only constraint of the problem is the terminal constraint, *i.e.*, to arrive at the selected target point. This constraint is satisfied internally by *fmincon*, and the DC algorithm only initializes the optimization. For this light simulation, the gradient of the cost function and constraints with respect to the decision variables is computed using finite differences. The optimization details are summarized in Table 5.2.

To test the optimization algorithm and illustrate the transfer, a test simulation is run, with the departure, target, and transfer time used as a reference during this chapter in Table 5.1.

As already mentioned, the simulation is initialized using the DC algorithm developed in the previous section. To avoid the optimizer getting stuck at the initialization, a deviation of 10^{-3} non-dimensional units (4.3×10^{-3} km/s) is introduced in every component of $\Delta\mathbf{v}_0$. The results of the optimization simulation are plotted in Figure 5.4.

Table 5.2: Optimization problem formulation for two-impulse simulation with departure velocity as a decision variable. *Practically, the constraint is set to be smaller than a specified tolerance.

Decision Variables	$\Delta\mathbf{v}_0$
Cost function	ΔV
Constraints	$\lvert\mathbf{r}_{\text{arr}} - \mathbf{r}_{\text{tgt}}\rvert = 0^*$
Partials	Finite differences

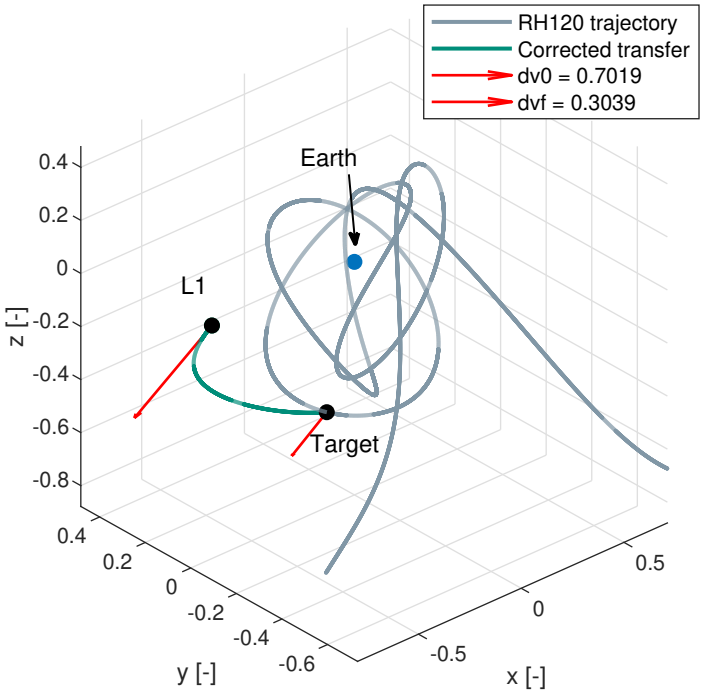


Figure 5.4: Trajectory optimization simulation results optimizing departure velocity.

It can be observed from Figure 5.4 that the resulting trajectory matches that of the one computed by the DC algorithm in the previous section (see Figure 5.2). The total ΔV for this transfer is 1.0058 km/s, which matches the result of the DC algorithm. At the same time, the distribution between the initial and final Δv's also matches.

Figure 5.5 shows the optimization history of the algorithm. It can be observed that the algorithm starts searching for transfers with a very low cost function value. However, as the algorithm iterates to match the constraints, the solution searches for higher delta-v values until the algorithm converges.

This first simulation shows that the optimization works to find a connecting trajectory that meets the constraints. However, the limited number of decision variables prevents the algorithm from finding an optimal transfer trajectory other than the one already found by the DC algorithm. In the next section, the decision variable vector is augmented to give more room for the algorithm to search for optimal transfers.

This section has served to gain familiarity with the *fmincon* optimizer and test the problem transcrip-

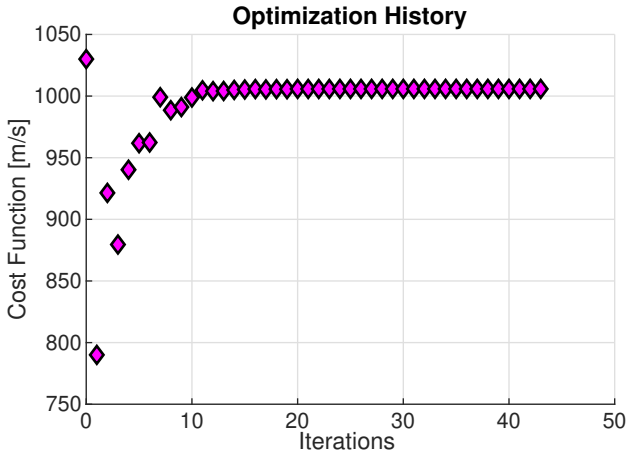


Figure 5.5: Optimization history for simulation optimizing departure velocity.

tion. It should be noted that a few other problem formulations were tested, such as an unconstrained optimization where the terminal constraint was enforced inside the cost function evaluation through the DC algorithm. The results from these other formulations were not efficient or promising.

5.3. Augmenting Decision Variables

As shown in the previous section, the decision variable vector can be expanded to allow the search algorithm to find optimal solutions that not only meet the constraints. This task is tackled by introducing the transfer time as a decision variable. Other design variables, such as the target epoch, remain fixed at this point. As such, the optimization problem from Table 5.2 is re-defined in Table 5.3.

Table 5.3: Optimization problem formulation for two-impulse transfer optimizing departure velocity and transfer time. *Practically, the constraint is set to be smaller than a specified tolerance.

Decision Variables	$\Delta\mathbf{v}_0$, T
Cost function	Δv_{total}
Constraints	$\lvert\mathbf{r}_{\text{arr}} - \mathbf{r}_{\text{tgt}}\rvert = 0^*$
Partials	Finite differences

With these settings, the simulation is run for the same target location as in Table 5.1 and initializing the decision variables from the results of the DC deviated 10^{-3} non-dimensional units (4.3×10^{-4} km/s). The resulting trajectory is plotted in Figure 5.6.

It is observed that, with this new problem formulation, the algorithm has found a solution at a lower cost than the previous simulations. The total ΔV has decreased from 1.0058 km/s to 0.9375 km/s. The algorithm has decreased the transfer time from 50 days to 41.0481 days. It is noted that the initial delta-v has slightly increased to reduce the target injection Δv.

Figure 5.7 shows the optimization history for this simulation. It can be observed that the algorithm converges in 60 iterations, which is higher than the 40 iterations for the previous simulation. However, the settlement value is lower than optimizing just the Δv_0.

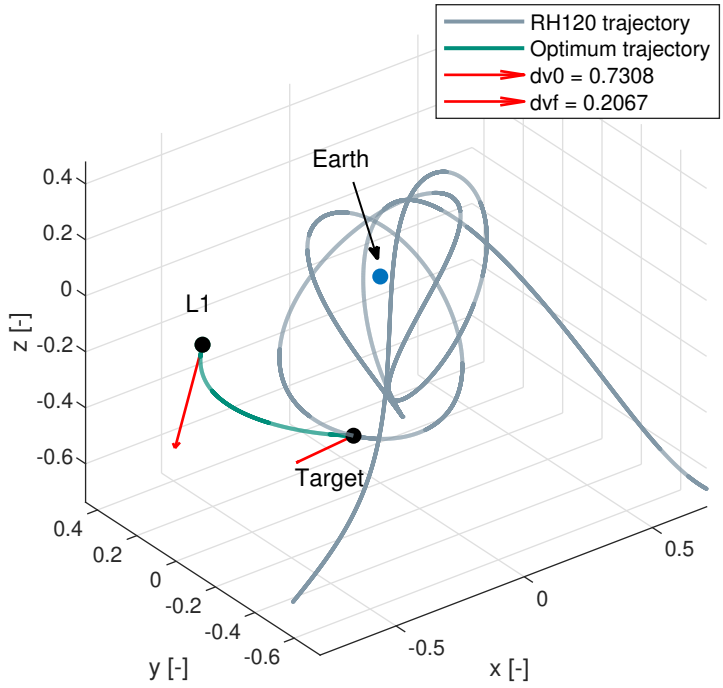


Figure 5.6: Trajectory optimization simulation results optimizing departure velocity and transfer time.

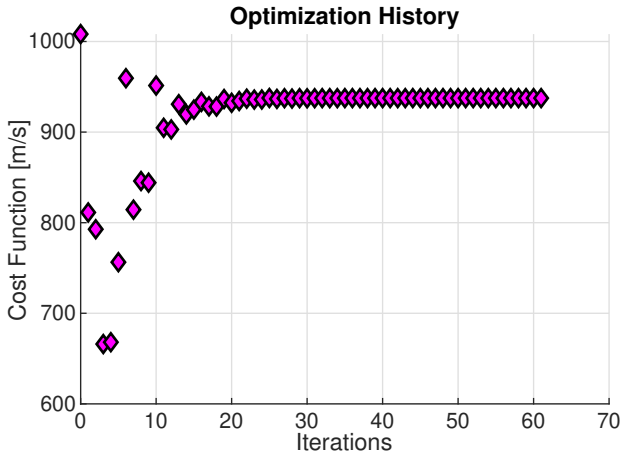


Figure 5.7: Optimization history for simulation optimizing departure velocity and transfer time.

Another variable that limits the algorithm's search space is the target point in the TCO trajectory, which has been fixed until now. Therefore, the next step is to include it in the list of decision variables. This position is parameterized by a variable τ that measures the epoch in seconds since 2006 RH120 enters the Hill sphere of the Earth.

The formulation of the new optimization problem, including the arrival epoch τ as a decision variable, is gathered in Table 5.4.

Figure 5.8 shows the optimization results for this simulation.

Table 5.4: Optimization problem formulation two-impulse simulation optimizing departure velocity, transfer time, and arrival epoch.

Decision Variables	$\Delta\mathbf{v}_0$, T, τ
Cost function	Δv_{total}
Constraints	$\lvert\mathbf{r}_{\text{arr}} - \mathbf{r}_{\text{tgt}}\rvert$
Partials	Finite differences

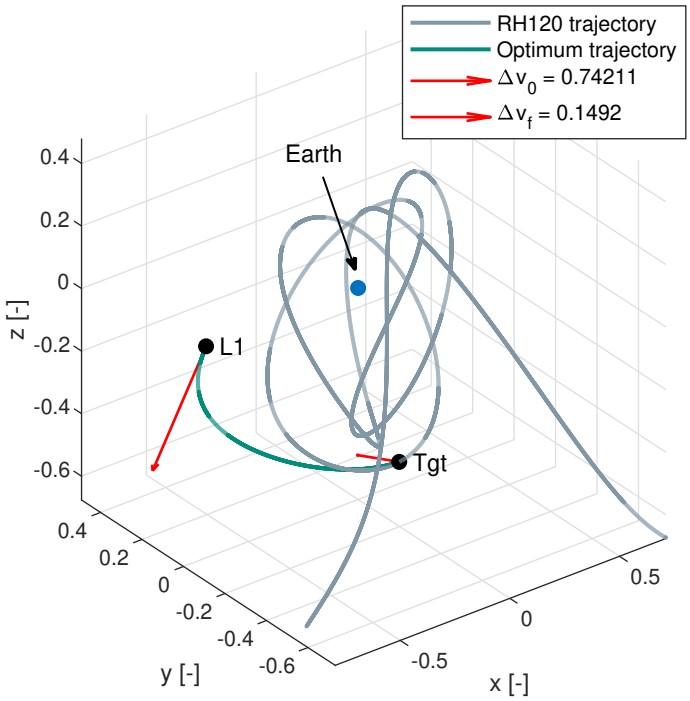


Figure 5.8: Trajectory optimization simulation results optimizing departure velocity, transfer time, and arrival epoch.

It can be observed from the figure that the arrival epoch has been pushed to an area where the arrival is more aligned with the trajectory of the TCO. The optimized rendezvous location is 235 days since Hill entry, as opposed to the previously fixed 220 days. The algorithm converges to a transfer time of 55 days, slightly longer than the previous optimum found at 41 days. A lower transfer cost has been found for this trajectory, sitting at a total delta-v of 0.8913 km/s.

Finally, the optimization history is shown in Figure 5.9. It can be observed that the optimizer takes around 90 iterations to converge, as compared to 70 in the previous simulation. This is still a reasonable number, and the optimizer starts converging toward the final value early in the optimization history. It should be noted that each iteration's runtime has increased compared to previous simulations. This simulation took 140 seconds to converge, while the previous one, which did not optimize the arrival epoch, converged in only 11 seconds. The runtime is not an issue with this two-impulse transfer formulation since the simulation was completed in under three minutes. However, this behavior indicates that it may be necessary to reduce the runtime once the simulations get more complex in the next section,

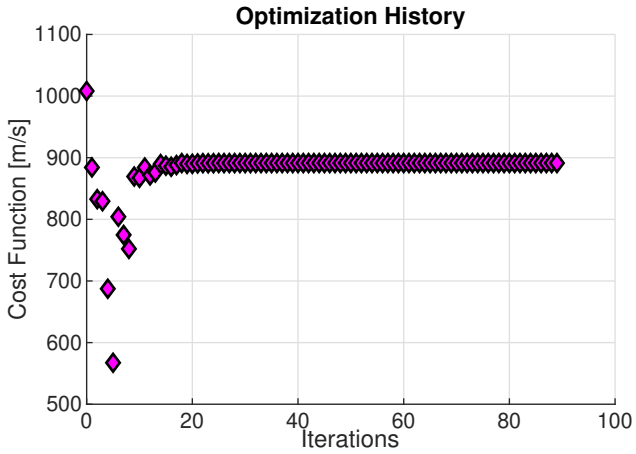


Figure 5.9: Optimization history for simulation optimizing departure velocity, transfer time, and arrival epoch.

such as passing the analytical gradient of the cost function and constraints to the optimizer.

5.4. Two-Impulse Optimization Remarks

This chapter has explored the methods for optimizing two-impulse transfers targeting TCOs. To illustrate the methods, a reference scenario has been selected with details gathered in Table 5.1. The variables that played a role in these types of trajectories were the departing Δv, the transfer time, and the arrival epoch in the trajectory of the TCO. The optimization has progressively increased complexity, and the results are summarized in Table 5.5.

Table 5.5: Two-impulse algorithm optimization simulation results. * Fixed values.

	Δv_0 [km/s]	Δv_f [km/s]	ΔV [km/s]	T [days]	τ [days]	Runtime [s]
Differential Corrector	0.7019	0.3039	1.0058	50*	220*	0.1
Optimizing Δv_0	0.7019	0.3039	1.0058	50*	220*	7
Optimizing Δv_0, T	0.7308	0.2067	0.9375	41.04	220*	11
Optimizing Δv_0, T, τ	0.7421	0.1492	0.8913	54.85	235.2	140

Several remarks on the optimization process are subsequently discussed.

5.4.1. Partial Derivatives

For this two-impulse problem, the partial derivatives of the cost function and constraints have been computed in the back-end of *fmincon* function through finite differences. The finite differences evaluate the cost function multiple times and can significantly increase the computation time of the simulation. For this light-weight two-impulse simulation, the computation time has not been an issue, and therefore finite differences lead to good results. However, attempts have been made to compute the partial derivatives analytically to gain familiarity with heavier simulations.

5.4.2. Decision Variables

Several trials have been made to make the final selection of the decision variables for this two-impulse problem.

Regarding the Δv, it should be noted that only the departure Δv has been selected. The arrival impulse is computed as the difference between the arrival velocity and the target. Since this second impulse is included in the cost function formulation, it is indirectly optimized. However, it has been observed that not including it as an independent variable complicates the formulations when deriving the analytical partial derivatives. Therefore, for future heavier simulations, the rendezvous delta-v impulse will be included in the decision variable formulation to simplify the analytical derivations.

Regarding the arrival epoch τ, it was initially introduced in the transcription of the problem by propagating the trajectory of the TCO until the target epoch and storing the final state. The computational cost of this method has been reduced by following an alternative approach: the discrete trajectory of the TCO is loaded into the cost function, and the trajectory is only propagated from the closest epoch found in the trajectory discretization to the target epoch τ.

Finally, it is observed that most of the Δv from the two-impulse simulation comes from the initial impulse. This is because the departing condition has been chosen as the Lagrangian point L1 for simplicity. However, the spacecraft's velocity at this location is zero, which requires a large amount of energy to intercept the TCO coming from the engines. The selection of a convenient departure orbit will be studied in future stages of this thesis.

5.4.3. Variable Scaling

It has been observed that the scaling of the variables greatly impacts the performance of the optimization algorithm. The non-dimensionalization of the problem has helped scale all the variables to a range better managed by the algorithm. For future simulations, the range of values that the decision variables can take should also be monitored to ensure they remain similar.

5.4.4. Optimality of Solution

Ensuring the optimizer finds the absolute optimal solution for a problem is complex. In this case, the simulations run in this section had the purpose of illustrating the optimization procedure and gaining familiarity with the algorithm. However, it has been observed that the outcome solution of the simulations depends on the initial target selected and simulation settings. To avoid this, it is recommended to augment the constraints of the problem to limit the range that the decision variables can take, allowing the division of the simulation into different ranges where local optima can be found and compared to each other.

6

Low-Thrust Transfer Design

To complete the objective of this thesis of computing optimum low-thrust trajectories to TCOs, it is required to transition from two-impulse trajectories to a model that resembles the continuous nature of a low-thrust trajectory. This chapter presents this algorithm upgrade. First, the low-thrust formulation is presented, followed by the selected problem transcription. Then, the gradient of the cost and constraint functions with respect to the decision variables is presented, which is one of the significant innovations introduced in this research. Then, the initialization process is explained, and a simulation to test the algorithm's performance and illustrate its behavior is presented. Next, the algorithm is further upgraded to optimize the departure location from a pre-selected periodic orbit, and a test simulation is again presented. The chapter concludes with a selection of remarks that apply to the low-thrust optimization process introduced.

6.1. Low-Thrust Formulation

As explained in Section 3.4.3, the means of discretizing this continuous trajectory for preliminary analysis is by dividing the transfer to a series of trajectory segments with delta-v impulses applied at the mid-points (Sims & Flanagan, 1999). At the same time, this simplifies the engine's acceleration over a trajectory arc to a constant, resulting in a total delta-v at the end of the segment.

This simplification can be assumed while the trajectory arcs are kept short enough such that the change in velocity comes from the acceleration of the low-thrust engine, which exerts a continuous acceleration over the center of mass of the spacecraft. Therefore, the following relation holds for the i-th arc

$$F = m \frac{\Delta v_i}{\Delta t_i} \tag{6.1}$$

The mass m of the spacecraft is a design parameter considered fixed throughout the trajectory for simplicity. Although in a real-case scenario, the mass would decrease as the propellant is being consumed, an electric propulsion spacecraft is assumed to have only a small fraction of the total mass

of the spacecraft devoted to the propellant. A preliminary calculation using the rocket equation in Equation 3.27 considering a conservative transfer delta-v of 1 km/s results in a propellant requirement of 16 kg for the selected spacecraft (wet mass of 500 kg and I_{sp} of 3,000 s). This translates into a 3.3% of the spacecraft mass devoted to the propellant. Therefore, the assumption of constant mass is taken for the preliminary design.

The thrust F is allowed to vary throughout the trajectory while being kept constant for the duration of a trajectory arc. However, this thrust is limited by a maximum available value, chosen as 0.1 N in Section 2.4.

For Equation 6.1 to hold, the acceleration exerted on the center of mass of the spacecraft is capped by two aspects: the maximum thrust available by the engine and an arc length small enough that replicates the continuous thrust behavior over the whole trajectory.

The problem is formulated by considering both the Δv_i's and the Δt_i's as decision variables for the solver to optimize. The solution's desired behavior is tuned by constraints imposed on these variables. The strategy to adjust these variables is to select a reasonable maximum ΔV for the overall trajectory and divide it by the number of impulses selected. This sets a maximum value for the Δv exerted at each arc.

$$\Delta v_{\text{max}} = \frac{\Delta V_{\text{max}}}{n} \tag{6.2}$$

Then, to comply with Equation 6.1, the minimum Δt for each trajectory arc is computed using the maximum thrust available. This prevents the optimizer from providing arc lengths not balanced throughout the trajectory while allowing for some margin.

$$\Delta t_{\text{min}} = m\frac{\Delta v_{\text{max}}}{F_{\text{max}}} \tag{6.3}$$

Once a solution is generated, the arc lengths are checked so that they are not excessively large. It has been found in literature that a reasonable arc length is comprised between half a day and eight days (Herman, 2012) for a comparable three-body trajectory in the Sun-Earth system with a similar total transfer time duration.

6.2. Optimization Problem Transcription

The type of transcription consists of a similar method as the one used in the two-impulse trajectory formulation but with the addition of the intermediate Δv's. This means that the scheme is still that of a single shooting, in which the position constraint is met at the target point. A visual representation of the used transcription is shown in Figure 6.1.

It can be observed from the illustration that n Δv impulses are introduced, resulting in $n-1$ trajectory arcs. Each trajectory arc has a transfer time Δt_i, which is allowed to vary to control the total transfer time of the simulation.

The cost function of the problem is the sum of all the Δv's performed

$$J = \sum_{i=1}^{n} |\Delta \mathbf{v}_i| \tag{6.4}$$

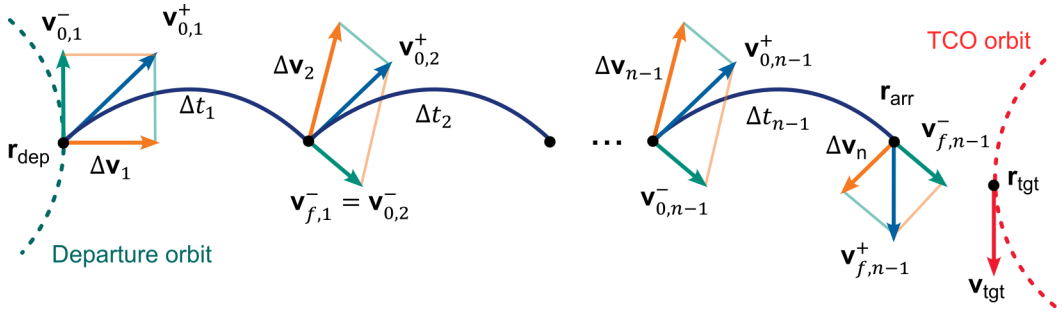


Figure 6.1: Illustration of low-thrust transfer problem transcription with relevant variables.

The decision variables are the same as in the two-impulse problem, which includes the Δv's, arc transfer times Δt_i's, and TCO arrival epoch τ. That is

$$\mathbf{X} = \left[\Delta\mathbf{v}_1, \quad \cdots, \quad \Delta\mathbf{v}_n, \quad \Delta t_1, \quad \cdots, \quad \Delta t_{n-1}, \quad \tau\right] \tag{6.5}$$

Two equality constraints have been selected for this problem transcription. The first constraint is the arrival at the target position by the end of the trajectory arcs, similar to the two-impulse formulation. In addition, the final Δv of the last leg has been included in the list of decision variables. Then a constraint is added such that the final velocity after the last Δv matches the velocity of the TCO at the target point. These constraints are defined as the following six-element column vector using the notation introduced. They are formulated such that all terms have to be zero (smaller than a specified numerical tolerance).

$$\mathbf{g}_{\text{eq}} = \begin{bmatrix} \mathbf{r}_{\text{tgt}} - \mathbf{r}_{\text{arr}} \\ \mathbf{v}_{\text{tgt}} - \mathbf{v}_{\text{arr}} \end{bmatrix} \tag{6.6}$$

It should be noted that this equality constraint vector in Equation 6.6 is passed into `fmincon` as a row vector, so it has to be transposed.

In addition to the equality constraints, certain inequality constraints are added to limit the values the decision variables can take. These constraints are the following

- Maximum value for Δv's – it is set not to exceed the thrust available by the engine and to avoid the simulation converging to a two-impulse type where the largest Δv's are applied close to the departure and arrival points.
- Maximum value for total transfer time T – it allows having control over the length of the entire trajectory to prevent excessively large simulations that are unfeasible in terms of mission design.
- Maximum and minimum values for Δt's – they prevent the simulation from creating very short arcs where practically an engine would not be able to switch the acceleration profile in time or very long arcs that dominate the total transfer time.

These inequality constraints are defined as follows

$$\mathbf{g}_{\text{ineq}} = \left[\Delta v_i - \Delta v_{\max}, \quad T - T_{\max}, \quad -\Delta t_j + \Delta t_{\min}, \quad \tau - \tau_{\max}, \quad -\tau + \tau_{\min}\right] \tag{6.7}$$

for all i in [1, n] and all j in [1, n-1]. They are formulated such that all terms have to be negative as required by `fmincon`.

6.3. Partial Derivatives

To generate a multi-impulse trajectory with an increased number of impulses, the decision variable vector (Equation 6.5) becomes large enough such that optimizing using finite differences consumes excessive computing resources. As previously explained, finite differences have been used up until now to compute the partial derivatives of the cost and constraint functions with respect to the decision variables vector. To solve this issue, the combination of problem transcription and the dynamics model selected has been specifically designed to allow computing the partial derivatives analytically. These analytical partial derivatives are one of the most innovative aspects of this thesis and are explained in this section.

6.3.1. Gradient of Cost Function

The gradient of the cost function (Equation 6.4) with respect to the decision variables (Equation 6.5) results in the following matrix.

$$\frac{\partial J}{\partial \mathbf{X}} = \left[\begin{array}{ccc} \frac{\partial J}{\partial \Delta \mathbf{v}_i}, & \frac{\partial J}{\partial \Delta t_j}, & \frac{\partial J}{\partial \tau} \end{array} \right]_{1\times 4n} \quad \forall\, i \in [1, n] \text{ and } j \in [1, n-1] \tag{6.8}$$

The gradient of the cost function with respect to the i-th impulse is computed as

$$\frac{\partial J}{\partial \Delta \mathbf{v}_i} = \frac{\Delta \mathbf{v}_i^{\mathsf{T}}}{|\Delta \mathbf{v}_i|}, \quad \forall\, i \in [1, n] \tag{6.9}$$

Since the cost function is formed by adding all the impulses, which are independent variables, the partials with respect to the remaining independent variables are zero.

6.3.2. Gradient of Equality Constraints

The position-velocity state at a given epoch is expressed by the six-element column vector $\mathbf{s}$, which therefore allows rewriting the constraints defined in Equation 6.6 as

$$\mathbf{g}_{\text{eq}} = \mathbf{s}_{\text{tgt}} - \mathbf{s}_{\text{arr}} \tag{6.10}$$

The gradient of the equality constraints with respect to the problem decision variables (Equation 6.5) is the following matrix, which is then transposed to pass it as an argument to `fmincon`.

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \mathbf{X}} = \left[\begin{array}{ccc} \frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta \mathbf{v}_i}, & \frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta t_j}, & \frac{\partial \mathbf{g}_{\text{eq}}}{\partial \tau} \end{array} \right]_{6\times 4n} \quad \forall\, i \in [1, n], \text{ and } j \in [1, n-1] \tag{6.11}$$

Several analytical expressions have been derived and introduced to populate this matrix.

The first element of the expression, which represents the partials of the constraint with respect to the velocity impulses, is computed as

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta \mathbf{v}_i} = \begin{cases} -\prod_{l=n-1}^{i} \Phi_l \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} & i < n \\ -\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} & i = n \end{cases}, \quad \forall\, i \in [1, n] \tag{6.12}$$

The derivation of this expression is explained subsequently.

Derivation of Equation 6.12:

By recalling the notation used in Figure 6.1, the equality constraints vector in Equation 6.10 can be rewritten as

$$\mathbf{g}_{\text{eq}} = \mathbf{s}_{\text{tgt}} - \mathbf{s}^{+}_{\text{f},n-1} \tag{6.13}$$

where $\mathbf{s}^{+}_{\text{f},n-1}$ represents the position-velocity state at the end of the propagation of the final arc of the transfer after the last Δv_n has been applied. Or equivalently,

$$\mathbf{s}^{+}_{\text{f},n-1} = \mathbf{s}^{-}_{\text{f},n-1} + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \Delta \mathbf{v}_n \end{bmatrix} \tag{6.14}$$

which allows to rewrite Equation 6.13 as

$$\mathbf{g}_{\text{eq}} = \mathbf{s}_{\text{tgt}} - \mathbf{s}^{-}_{\text{f},n-1} - \begin{bmatrix} \mathbf{0}_{3\times1} \\ \Delta \mathbf{v}_n \end{bmatrix} \tag{6.15}$$

When taking the partial derivative of the equality constraint in Equation 6.15 with respect to the i-th Δv, the partials of $\mathbf{s}_{\text{tgt}}$ and $\Delta \mathbf{v}_n$ are zero for every i-th impulse except for the last one ($i = n$), resulting in

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta \mathbf{v}_i} = -\frac{\partial \mathbf{s}^{-}_{\text{f},n-1}}{\partial \Delta \mathbf{v}_i} \tag{6.16}$$

The right-hand side of Equation 6.16 can be rewritten in terms of known quantities as follows

$$\frac{\partial \mathbf{s}^{-}_{\text{f},n-1}}{\partial \Delta \mathbf{v}_i} = \frac{\partial \mathbf{s}^{-}_{\text{f},n-1}}{\partial \mathbf{s}^{+}_{0,n-1}} \frac{\partial \mathbf{s}^{+}_{0,n-1}}{\partial \Delta \mathbf{v}_i} = \Phi_{n-1} \frac{\partial \mathbf{s}^{+}_{0,n-1}}{\partial \Delta \mathbf{v}_i} \tag{6.17}$$

The vector $\mathbf{s}^{+}_{0,n-1}$ is obtained after propagating the previous leg of the transfer and can be equivalently written as

$$\mathbf{s}^{+}_{0,n-1} = \mathbf{s}^{-}_{\text{f},n-2} + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \Delta \mathbf{v}_{n-1} \end{bmatrix} \tag{6.18}$$

And therefore, by referring to the l-th trajectory arc, while $l > i$, the following expression holds

$$\frac{\partial \mathbf{s}^{+}_{0,l}}{\partial \Delta \mathbf{v}_i} = \frac{\partial \mathbf{s}^{-}_{\text{f},l-1}}{\partial \Delta \mathbf{v}_i} \tag{6.19}$$

which allows to rewrite Equation 6.17 as

$$\frac{\partial \mathbf{s}^{-}_{\text{f},n-1}}{\partial \Delta \mathbf{v}_i} = \Phi_{n-1} \frac{\partial \mathbf{s}^{-}_{\text{f},n-2}}{\partial \Delta \mathbf{v}_i} \tag{6.20}$$

This equation represents that the partial of the final state of a given leg equals the product of the STM of such leg and the partial of the final state of the previous leg. This result can be extrapolated to previous legs following the same reasoning as to derive Equation 6.20 until the leg count matches that of the i-th impulse, resulting in

$$\frac{\partial \mathbf{s}_{\mathrm{f},n-1}^{-}}{\partial \Delta \mathbf{v}_i} = \prod_{l=n-1}^{i} \Phi_l \, \frac{\partial \mathbf{s}_{0,i}^{+}}{\partial \Delta \mathbf{v}_i} \tag{6.21}$$

The element $\frac{\partial \mathbf{s}_{0,i}^{+}}{\partial \Delta \mathbf{v}_i}$ can be rewritten by using the same transformation as in Equation 6.18 into

$$\frac{\partial \mathbf{s}_{0,i}^{+}}{\partial \Delta \mathbf{v}_i} = \frac{\partial \mathbf{s}_{0,i}^{-}}{\partial \Delta \mathbf{v}_i} + \begin{bmatrix} \mathbf{0}_{3\times3} \\ \frac{\partial \Delta \mathbf{v}_i}{\partial \Delta \mathbf{v}_i} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} \tag{6.22}$$

Finally, Equations 6.21 and 6.22 are substituted into Equation 6.16 to give

$$\frac{\partial \mathbf{g}_{\mathrm{eq}}}{\partial \Delta \mathbf{v}_i} = -\prod_{l=n-1}^{i} \Phi_l \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{I}_{3\times3} \end{bmatrix} \tag{6.23}$$

This result is valid for every i-th arc except the last one. For the partial of the equality constraint with respect to the last arc, $i = n$, it is observed from Equation 6.13 that the only non-zero element comes from Equation 6.22, which results in the expression for the gradient that was sought that equals Equation 6.12

$$\frac{\partial \mathbf{g}_{\mathrm{eq}}}{\partial \Delta \mathbf{v}_i} = \begin{cases} -\prod_{l=n-1}^{i} \Phi_l \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} & i < n \\ -\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix} & i = n \end{cases}, \ \forall i \in [1, n] \tag{6.24}$$

This result implies that to obtain the analytical expression for the gradient of the equality constraints with respect to the arc impulses, the only information needed is the STM that maps the arrival with the departure condition of each arc, which is a known matrix.

The second element of Equation 6.11, which represents the partials of the gradient with respect to the transfer time of each segment, is computed as

$$\frac{\partial \mathbf{g}_{\mathrm{eq}}}{\partial \Delta t_j} = \begin{cases} -\begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} & j = n-1 \\ -\prod_{l=n-1}^{j+1} \Phi_l \begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} & j < n-1 \end{cases}, \ \forall j \in [1, n-1] \tag{6.25}$$

The derivation of this expression is explained subsequently.

Derivation of Equation 6.25:

By recalling Equation 6.15, the gradient of the equality constraints with respect to the j-th arc length is given by

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta t_j} = -\frac{\partial \mathbf{s}_{\text{f},n-1}^{-}}{\partial \Delta t_j} \tag{6.26}$$

which, similarly to Equations 6.17 and 6.18, can be rewritten in terms of known quantities as

$$\frac{\partial \mathbf{s}_{\text{f},n-1}^{-}}{\partial \Delta t_j} = \frac{\partial \mathbf{s}_{\text{f},n-1}^{-}}{\partial \mathbf{s}_{0,n-1}^{+}} \frac{\partial \mathbf{s}_{0,n-1}^{+}}{\partial \Delta t_j} = \Phi_{n-1} \frac{\partial \mathbf{s}_{0,n-1}^{-}}{\partial \Delta t_j} \tag{6.27}$$

and therefore, referring to the l-th trajectory arc, the following expression holds

$$\frac{\partial \mathbf{s}_{0,l}^{+}}{\partial \Delta t_j} = \frac{\partial \mathbf{s}_{\text{f},l-1}^{-}}{\partial \Delta t_j} \tag{6.28}$$

which allows to rewrite Equation 6.27 as

$$\frac{\partial \mathbf{s}_{\text{f},n-1}^{-}}{\partial \Delta t_j} = \Phi_{n-1} \frac{\partial \mathbf{s}_{\text{f},n-2}^{-}}{\partial \Delta t_j} \tag{6.29}$$

This equation represents that the partial of the final state of a given leg equals the product of the STM of such leg and the partial of the final state of the previous leg. This result can be extrapolated to previous legs until the leg count matches that of the $j+1$-th impulse, resulting in

$$\frac{\partial \mathbf{s}_{\text{f},n-1}^{-}}{\partial \Delta t_j} = \prod_{l=n-1}^{j+1} \Phi_l \frac{\partial \mathbf{s}_{\text{f},j}^{-}}{\partial \Delta t_j} \tag{6.30}$$

It is identified that the element $\frac{\partial \mathbf{s}_{\text{f},j}^{-}}{\partial \Delta t_j}$ represents the velocity and acceleration at the end of the j-th arc, *i.e.*

$$\frac{\partial \mathbf{s}_{\text{f},j}^{-}}{\partial \Delta t_j} = \begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} \tag{6.31}$$

Finally, Equations 6.30 and 6.31 are substituted into Equation 6.26 to give

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta t_j} = -\prod_{l=n-1}^{j+1} \Phi_l \begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} \tag{6.32}$$

It is observed that when $j = n-1$, the only transformation to occur is that of Equation 6.31, which results in the expression for the gradient that was sought that equals Equation 6.12

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \Delta t_j} = \begin{cases} -\begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} & j = n-1 \\ -\prod_{l=n-1}^{j+1} \Phi_l \begin{bmatrix} \mathbf{v}_{f,j}^{-} \\ \mathbf{a}_{f,j} \end{bmatrix} & j < n-1 \end{cases}, \ \forall j \in [1, n-1] \tag{6.33}$$

This solution implies that the gradient of the equality constraints with respect to the trajectory arc's lengths is only a function of known quantities, *i.e.* the STM of each arc, and the velocity and acceleration vectors at the end of each arc.

Finally, the partials of the equality constraints with respect to the target epoch τ are simply composed by the velocity and acceleration of the target point at epoch τ

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \tau} = \begin{bmatrix} \mathbf{v}_{\text{tgt}} \\ \mathbf{a}_{\text{tgt}} \end{bmatrix} \tag{6.34}$$

This result completes the expression for the partials of the equality constraints with respect to the decision variables in Equation 6.11.

6.3.3. Gradient of Inequality Constraints

The gradient of the inequality constraints (Equation 6.7) with respect to the problem decision variables (Equation 6.5) is the following matrix:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \Delta t_l} & \frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \tau} \\ \frac{\partial(T - T_{\max})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(T - T_{\max})}{\partial \Delta t_l} & \frac{\partial(T - T_{\max})}{\partial \tau} \\ \frac{\partial(-\Delta t_j + \Delta t_{\min})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(-\Delta t_j + \Delta t_{\min})}{\partial \Delta t_l} & \frac{\partial(-\Delta t_j + \Delta t_{\min})}{\partial \tau} \\ \frac{\partial(\tau - \tau_{\max})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(\tau - \tau_{\max})}{\partial \Delta t_l} & \frac{\partial(\tau - \tau_{\max})}{\partial \tau} \\ \frac{\partial(-\tau + \tau_{\min})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(-\tau + \tau_{\min})}{\partial \Delta t_l} & \frac{\partial(-\tau + \tau_{\min})}{\partial \tau} \end{bmatrix}_{2(n+1)\times 4n} \tag{6.35}$$

for all i and k in [1, n], and j and l in [1,n-1].

After identifying the independent variables, the matrix is rewritten as

$$\frac{\partial \mathbf{g}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \Delta \mathbf{v}_k} & \mathbf{0}_{n\times n-1} & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{1\times 3n} & \mathbf{1}_{1\times n-1} & 0 \\ \mathbf{0}_{n-1\times 3n} & \mathbf{I}_{n-1\times n-1} & \mathbf{0}_{n-1\times 1} \\ \mathbf{0}_{1\times 3n} & \mathbf{0}_{1\times n-1} & 1 \\ \mathbf{0}_{1\times 3n} & \mathbf{0}_{1\times n-1} & -1 \end{bmatrix}_{4n\times 2(n+1)} \tag{6.36}$$

for all i and k in [1, n]. The first matrix equals

$$\frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \Delta \mathbf{v}_k} = \begin{cases} \mathbf{0}_{1\times 3} & i \neq j \\ \frac{\Delta \mathbf{v}_i^{\mathrm{T}}}{|\Delta \mathbf{v}_i|} & i = k \end{cases}, \quad \forall\, i \text{ and } k \in [1, n] \tag{6.37}$$

6.4. Algorithm Initialization

In order to run the optimization algorithm, the decision variable vector in Equation 6.5 must be initialized. For such a task, an initial target epoch of the TCO trajectory and an initial transfer time is selected. Then, a two-impulse trajectory is generated using the DC algorithm. Next, the transfer time is divided into the desired number of equally spaced segments, and the delta-v vectors for such epochs are retrieved from the TCO states. Then, a small deviation of 10^{-4} non-dimensional units is added to every element of the velocity vectors. This small value is added to avoid the algorithm getting stuck at the initialization.

6.5. Optimization Simulation

Similarly to the two-impulse case, to illustrate the works of the algorithm, we have prepared a simulation. The simulation settings are gathered in Table 6.1. They represent the input parameters to the simulation, which can be modified to tune the solution. Since the purpose of this simulation is illustrative, the parameters selected are deemed reasonable, but they have not been properly tuned yet. This will be done in Section 8.1.

Table 6.1: Low-thrust simulation settings for departure from L1.

Parameter	Value	Units	Description
Algorithm			
n	50	[-]	Number of impulses
Initialization			
T_0	50	[days]	Initial transfer time
τ_0	220	[days]	Initial target epoch
Boundaries			
T_{max}	75	[days]	Maximum transfer time
ΔV_{max}	1.2	[km/s]	Maximum total ΔV
τ_{range}	$[200, 240]$	[days]	Range of target epoch

The simulation is run, and the resulting optimal trajectory is plotted in Figure 6.2. It can be observed from the figure that the impulses align at the beginning and end of the transfer to create a smooth transition from departure to rendezvous. This transfer strategy differs from the solutions in Chapter 5, where the transitions were abruptly performed by two high-impulse maneuvers (see Figure 5.8).

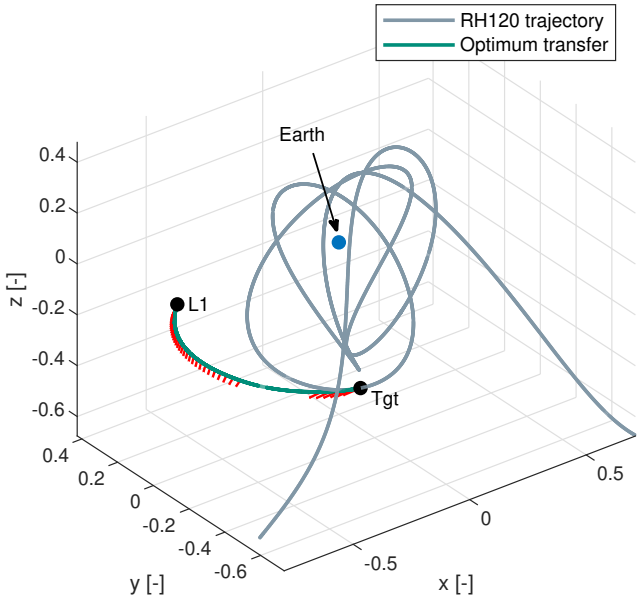


Figure 6.2: Optimized low-thrust transfer trajectory departing from L1.

To understand the evolution of the impulses, the resulting thrust profile has been plotted in Figure 6.3. The figure shows the thrust magnitude as a percentage of the maximum available thrust by the engine over the transfer time. It is observed that the highest thrust is applied during the first half of the transfer, where the engines are kept at roughly 95% of their maximum thrust. Then, the thrust is brought down to almost zero for trajectory arcs where no impulse is required. This means that the optimizer converged to introduce some coasting into the transfer trajectory. Finally, the thrust is brought up again to about 90% for rendezvous.

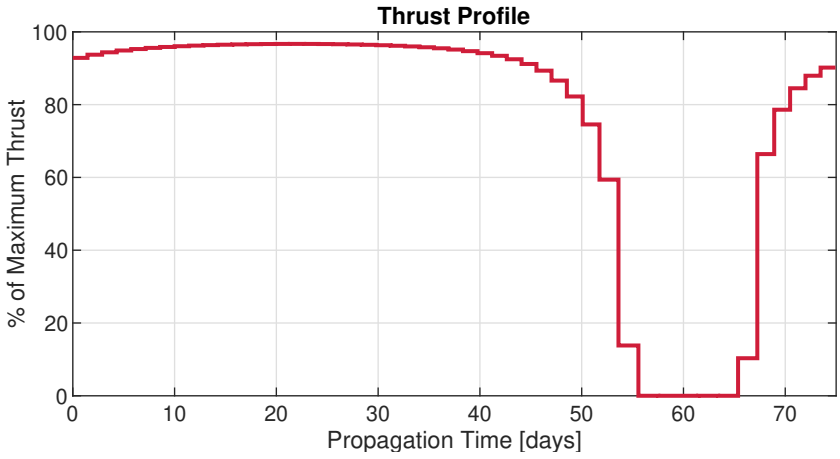


Figure 6.3: Thrust profile for low-thrust optimum transfer departing from L1.

To give more insight into the simulation results, some parameters have been gathered in Table 6.2. A few things can be observed from these results.

Table 6.2: Low-thrust simulation results for departure from L1.

Parameter	Value	Description
ΔV	0.9979 km/s	Total transfer delta-v
T	74.93 days	Total transfer time
τ	230.58 days	Arrival epoch in days since entering Hill
runtime	72 s	Simulation runtime

First, it is noted that the total ΔV of the transfer has slightly increased with respect to the two-impulse results obtained in Section 5.4, Table 5.5, from 0.8913 to 0.9979 km/s. This is expected since concentrating the impulses at fewer points tends to be more efficient than spreading the impulses over the transfer due to the Oberth effect (see Wakker, 2015). However, the gain obtained from using multiple impulses comes from the significant reduction in the propellant mass when using high-I_{sp} engines as compared to low-I_{sp} chemical propulsion systems.

Second, it is noted that the total transfer time optimized has increased from 50 days to almost 75, which is very close to the constraint that has been set for the simulation. This means that other, more optimal solutions may be found if the algorithm is allowed to run for longer transfer times.

Third, it is observed that the arrival epoch has increased from 220 days to 230 days since entering the Hill sphere. This value is, however, kept well within the constraint boundaries. It is noted that the combination of arrival epoch and transfer time sets the required departure epoch for the spacecraft at 155 days since the TCO enters the Hill sphere.

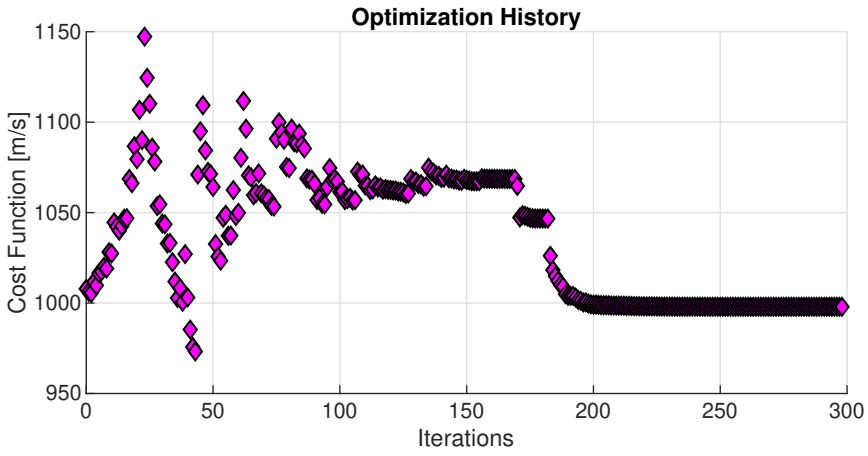


Figure 6.4: Optimization history of low-thrust trajectory simulation departing from L1.

To illustrate the search process of the algorithm, the optimization history is plotted in Figure 6.4. It can be observed that the search algorithm does not directly converge to the solution as happened in previous simulations. Several optima are explored in this case until the algorithm converges to the solution. Given that the number of variables has significantly increased, it is reasonable that the algorithm takes more iterations to converge, requiring 300 compared to less than 100 for the two-impulse simulations (see Figure 5.9). However, it should be noted that although the number of iterations has increased, passing the analytical partials has significantly increased the algorithm's speed compared to

finite differences used for two-impulse simulations, hence reducing the optimization time. In this case, the total runtime, as observed in Table 6.2, is slightly longer than one minute, around half of the runtime of the latest two-impulse simulation in Table 5.5.

Finally, the simulation has been rerun using finite differences instead of the analytical solution, both for verification purposes and to compare the computational cost. The results using finite differences converged to the same optimum (with minor deviation) of $\Delta V = 0.9983$ km/s, $T = 74.93$ days, and $\tau = 230.54$ days, but the runtime was indeed significantly larger at 105 minutes, compared to 72 s for the analytical derivatives. Therefore, the reduction in runtime achieved by using the analytical method is 99.86%. This confirms the great advantage of passing the analytical gradients of the cost and constraint functions. However, this advantage should be traded against the time spent developing, if possible, the analytical expressions. For our problem, which deals with a preliminary design, the formulation could be simplified enough to find the analytical expressions. But this may not be the case for higher-fidelity formulations.

6.6. Adding Departure Conditions

It has been shown in previous sections that an algorithm has been derived that computes an optimal low-thrust transfer trajectory from L1 to a point on the trajectory of a target TCO. At this point, it is convenient to introduce a departure condition different from the Lagrangian point. This is the case as the Lagrangian point L1 has zero velocity in the Hill formulation, and therefore all the energy to match the orbit of the TCO has to come from the spacecraft. It is the purpose of this section to introduce into the algorithm a departure condition from a periodic orbit around L1. This departure alternative has the potential to benefit from an initial velocity vector that reduces the dependency on onboard energy.

The method to include the departure conditions in the algorithm is to parameterize the departure point in the orbit similarly as done with the target within the TCO trajectory. A time variable ε is introduced, which quantifies the epoch of the departure point within the period P of the orbit, $\varepsilon = 0$ being at the Initial Conditions (ICs) of the periodic orbit. Therefore, the decision variable vector from Equation 6.5 is extended to

$$\mathbf{X} = \left[\Delta\mathbf{v}_1, \quad \cdots, \quad \Delta\mathbf{v}_n, \quad \Delta t_1, \quad \cdots, \quad \Delta t_{n-1}, \quad \tau, \quad \varepsilon\right] \tag{6.38}$$

Note that the variable ε is non-dimensionalized by the time parameter to scale the variable, similar to τ.

To keep control over the values that this decision variable may take, two inequality constraints are added, which allow the user to control the range of values the variable can take. Usually, it takes values between zero and one period. However, these can be modified to target a specific region within the departure orbit. Therefore, the inequality constraint vector from Equation 6.7 is augmented to the following expression.

$$\mathbf{g} = \left[\Delta v_i - \Delta v_{\max}, \quad T - T_{\max}, \quad -\Delta t_j + \Delta t_{\min}, \quad \tau - \tau_{\max}, \quad -\tau + \tau_{\min}, \quad \varepsilon - \varepsilon_{\max}, \quad -\varepsilon + \varepsilon_{\min}\right] \tag{6.39}$$

After adding this decision variable, the partial derivatives of the cost function and constraints have

to be augmented to accommodate this new dependency.

The partial derivative of the cost function in Equation 6.8 is augmented by an extra column element taking the value

$$\frac{\partial J}{\partial \varepsilon} = 0 \tag{6.40}$$

since the cost function is formed by adding the Δv's, which are independent variables.

The partial of the equality constraints as presented in Equation 6.11 is augmented with a column on the right with

$$\frac{\partial \mathbf{g}_{\text{eq}}}{\partial \varepsilon} = -\prod_{l=n-1}^{i} \Phi_l \begin{bmatrix} \mathbf{v}_{\text{dep}} \\ \mathbf{a}_{\text{dep}} \end{bmatrix} \tag{6.41}$$

whose derivation is similar to that of Equation 6.25 and is omitted to keep this section concise.

Finally, the matrix containing the partials of the inequality constraints with respect to the decision variables (Equation 6.35) is augmented by two bottom rows and one right-most column with the following elements, respectively

$$\begin{bmatrix} \frac{\partial(\varepsilon - \varepsilon_{\max})}{\partial \mathbf{X}} \\ \frac{\partial(-\varepsilon + \varepsilon_{\min})}{\partial \mathbf{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial(\varepsilon - \varepsilon_{\max})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(\varepsilon - \varepsilon_{\max})}{\partial \Delta t_l} & \frac{\partial(\varepsilon - \varepsilon_{\max})}{\partial \tau} \\ \frac{\partial(-\varepsilon + \varepsilon_{\min})}{\partial \Delta \mathbf{v}_k} & \frac{\partial(-\varepsilon + \varepsilon_{\min})}{\partial \Delta t_l} & \frac{\partial(-\varepsilon + \varepsilon_{\min})}{\partial \tau} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{1\times 3n} & \mathbf{0}_{1\times n-1} & 0 \\ \mathbf{0}_{1\times 3n} & \mathbf{0}_{1\times n-1} & 0 \end{bmatrix} \tag{6.42}$$

$$\frac{\partial \mathbf{g}}{\partial \varepsilon} = \begin{bmatrix} \frac{\partial(\Delta v_i - \Delta v_{\max})}{\partial \varepsilon} \\ \frac{\partial(T - T_{\max})}{\partial \varepsilon} \\ \frac{\partial(-\Delta t_j + \Delta t_{\min})}{\partial \varepsilon} \\ \frac{\partial(\tau - \tau_{\max})}{\partial \varepsilon} \\ \frac{\partial(-\tau + \tau_{\min})}{\partial \varepsilon} \\ \frac{\partial(\varepsilon - \varepsilon_{\max})}{\partial \varepsilon} \\ \frac{\partial(-\varepsilon + \varepsilon_{\min})}{\partial \varepsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ 0 \\ \mathbf{0}_{n-1\times 1} \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \tag{6.43}$$

Once the new decision variable has been accommodated into the algorithm, a simulation is run to test its performance and illustrate the method. A periodic departure orbit around the L1 point has been selected for such a simulation. Based on intuition, a northern halo orbit with the same Jacobi constant as the TCO has been selected for this test. The parameter settings used for the simulation are gathered in Table 6.3. The maximum transfer time has been increased to 250 days since previous results were close to the upper boundary. The remaining parameters have again been selected as reasonable for illustration purposes but will be tuned appropriately in Section 8.1.

The simulation has been run, and the resulting trajectory is plotted in Figure 6.5. It can be observed that the optimizer has selected a departure location from the Halo orbit that points the velocity in a

Table 6.3: Low-thrust Simulation settings for departure from northern halo orbit.

Parameter	Value	Units	Description
Algorithm			
n	50	[-]	Number of impulses
Initialization			
T_0	50	[days]	Initial transfer time
τ_0	220	[days]	Initial target epoch
ε_0	0	[days]	Initial departure epoch
Boundaries			
T_{max}	250	[days]	Maximum transfer time
τ_{range}	$[200, 240]$	[days]	Range of target epoch
$\varepsilon_{\text{range}}$	$[0, 1]\,P$	[days]	Range of departure epoch

favorable direction to reach the target. It will be analyzed in Section 8.1 whether this is the absolute optimum or other locations within the departure orbit could lead to more favorable ICs. It is also observed from the figure that the resulting trajectory has again the Δv's aligned in a way that creates a smooth transition between departure and target.

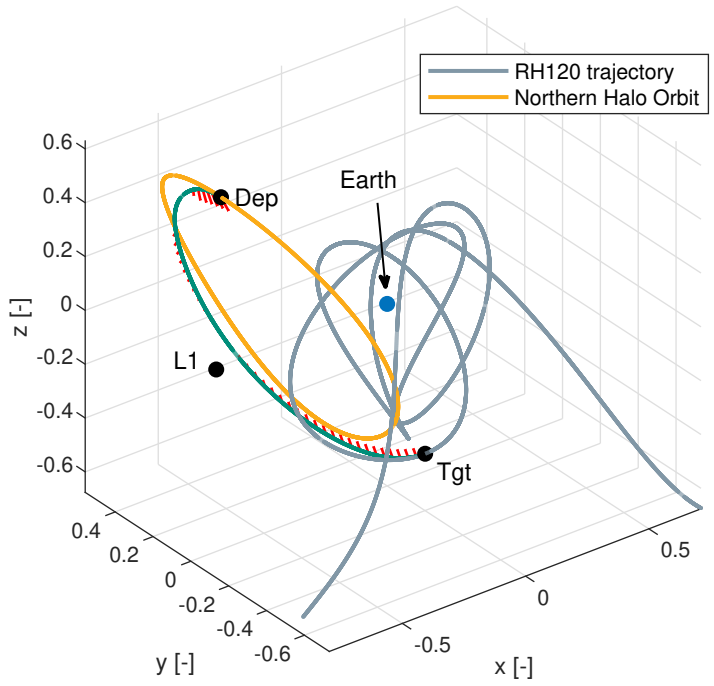


Figure 6.5: Simulated Low-Thrust optimum trajectory from Northern Halo Orbit,

The thrust profile is plotted in Figure 6.6, where it can be observed that the engine reaches a maximum thrust of 85% of the maximum thrust available in two segments. The remaining trajectory is a stepped but relatively smooth progression that includes a coasting phase close to departure.

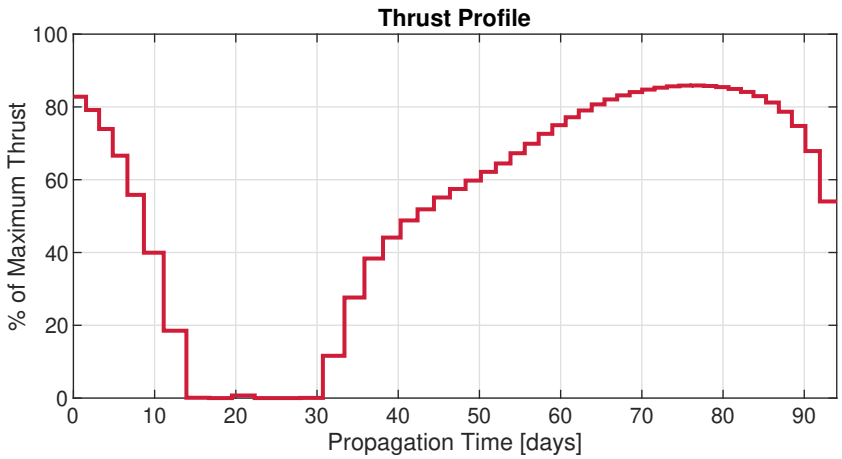


Figure 6.6: Thrust profile low-thrust trajectory from Northern Halo orbit.

The numerical results of the simulation are gathered in Table 6.4. Something that stands out is the total transfer time, which has been given a larger upper boundary of 250 days and has converged to 93 days, which is now well within the limits. The arrival epoch sits at 234 days for a maximum limit of 240 days. It is also observed that the transfer's total ΔV has decreased from 997.9 m/s in previous simulations (see Table 6.2) to 874.5 m/s, which is a reduction of 12%. However, it is still believed that this value can be further reduced by conducting a departure orbit analysis (Chapter 8). It is also notable that the runtime of the simulation has only marginally increased with respect to the optimization that did not include the departure epoch.

Table 6.4: Simulation results of low-thrust transfer departing from northern halo orbit.

Parameter	Optimum	Units	Description
ΔV	874.5	[m/s]	Total transfer ΔV
T	94.02	[days]	Total transfer time
τ	234.11	[days]	Arrival epoch
ε	0.0058	[-]	Departure epoch
t	80	[s]	Simulation runtime

Figure 6.7 shows the optimization history for the simulation. It can be observed that the algorithm reaches the convergence area with only a few more iterations than in the previous simulation (see Figure 6.4). This may indicate that it has converged to a local minimum, and the search could improve by dividing the search space further. A suggestion is to split the departure orbit into four regions and run four independent simulations. Finally, the number of iterations behaves as expected, reaching a slightly higher number than in the previous simulation but still kept within the same order of magnitude.

6.7. Low-thrust Optimization Remarks

In this chapter, we have finalized the design of the low-thrust trajectory optimization algorithm that will be used to find optimum transfer trajectories to 20006 RH120. The problem has been formulated using

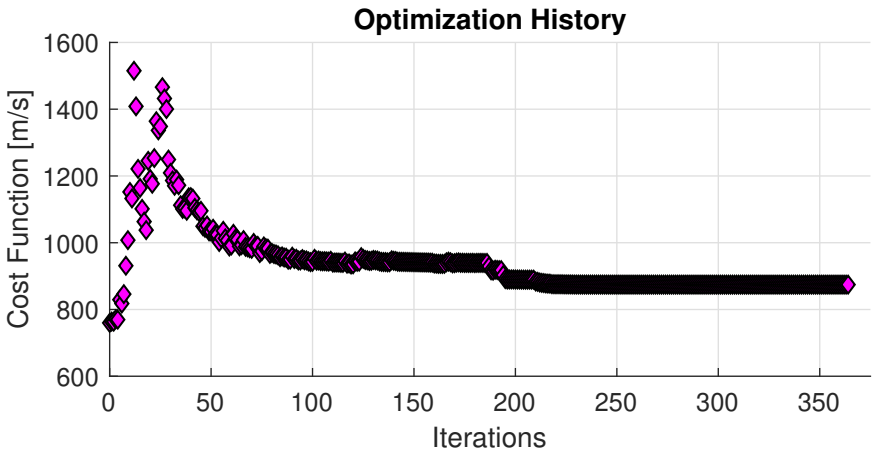


Figure 6.7: Optimization history for low-thrust trajectory from Northern Halo Orbit.

a single-shooting approach, in which the trajectory is divided into segments where impulses are applied. For the formulation to work within a reasonable computation time, the analytical partials for the cost and constraint functions have been derived and presented in this chapter. Initially, the transfers departed from a fixed L1 point, and later, the algorithm was augmented to optimize the departure location within a given periodic orbit. The algorithm has been tested, first departing from L1 and then selecting a departure orbit that matches the Jacobi constant of 2006 RH120. The algorithm has proven to work correctly and converge in a very low runtime of the order of one minute. The impulsive delta-v's obtained follow a smooth transition that resembles the low-thrust engine's continuous nature.

Several essential aspects of the trajectory optimization problem have been learned in this section. The first one is that progressively increasing the simulations' complexity has been very advantageous. Initially, arriving at a problem formulation with good convergence was challenging. Different options could be tested by starting from more accessible simulations, and the most promising were followed. At the same time, this approach has allowed checking that the analytical partials work correctly before increasing the complexity of the derivations. At first, getting the analytical gradient of cost and constraint functions was difficult. Many trials and errors were required. As familiarity with the problem was gaining, the complexity was increased, and by the time the most complex version of the algorithm was reached, increasing the number of variables to accommodate the departure orbit resulted in a relatively easy task.

Although the test simulations in this section resulted in feasible trajectories, some work is still left to verify and validate the algorithm and, most importantly, tune it, which will be performed in later chapters (Section 8.1).

Finally, it should be noted that much effort and time was devoted to other transcription methods. Specifically, at first, many resources were devoted to developing a multiple-shooting scheme, where constraints were satisfied at every intermediate point. The simulations run with this formulation had low convergence rates, and the resulting impulses were not smooth. The simulations became extremely difficult to test as the number of intermediate points was more than three, which resulted in an inadequate representation of the continuous nature of the low-thrust engine.

The current optimization approach that was developed has been presented, with all ups and downs,

tested, and discussed in this chapter.

7

Code, Verification and Validation

In this chapter, the code of the low-thrust optimization algorithm is explained and verified, and the problem is validated. The code structure is first presented, which is divided into three modules labeled as propagation module, optimization module, and Tudat module. Then, the code presented is verified using a list of unit tests. Finally, the optimization problem is validated.

7.1. Code Structure

The code implemented throughout this thesis is explained in this section. In particular, the overall architecture is presented, and the main functions and data flow are explained. Attention is paid to the final version of the algorithm, which includes the complete version of the problem formulations presented in Chapter 6. It is noted that earlier development versions of the code also fit within the same structure but with reduced capabilities.

As said already, the code can be structured in three main modules: a propagation module, an optimization module, and an external Tudat module. The former involves all the functions that allow propagating a body in the dynamics model selected for a specified time starting from a set of ICs. The second one is the thesis's most innovative and complex module, which deals with all the code related to optimizing a low-thrust trajectory from a given departure orbit and targeting a specific TCO. The external Tudat module retrieves ephemeris data of celestial objects and propagates trajectories for benchmarking.

7.1.1. Propagation Module

The propagation module is a fundamental block of the algorithm developed for this thesis. It is part of the code responsible for propagating a trajectory numerically for given departure conditions and transfer time. The block generates a discrete history of epochs, their corresponding states (position-velocity), and the STM.

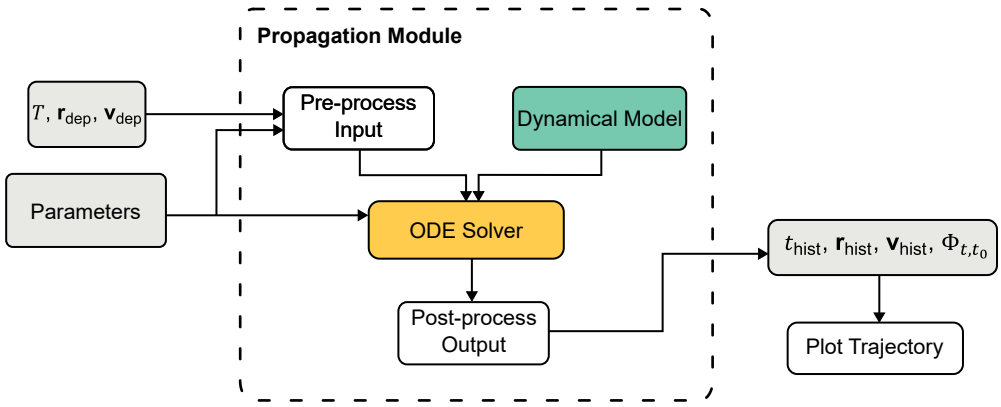


Figure 7.1: Flow-chart of software propagation module.

Figure 7.1 provides a flow chart of the module, where the inputs, outputs, and internal flow are depicted. The block framed under the dashed line is responsible for the propagation and is represented in the code by a MATLAB function called `propagateDynamics.m`. As can be observed from the chart, the propagation module makes use of a MATLAB built-in ODE solver to propagate the trajectory. This solver is fed with the integration time and initial conditions and with the differential equation to be solved, which is represented by the *Dynamics Model* block in the chart.

The dynamics model is a function that takes as input a state vector at a given epoch and returns the derivative of such state. The function implements the analytical expression for the derivative of the dynamics equations. The dynamics model used for the simulations is the Hill Three-body problem as given by Equation 3.22 in non-dimensional units. At the same time, the position-velocity state is augmented by the elements of the STM, which are integrated according to Equation 4.6. The function that implements these dynamics is called `Hill3BP.m` and has an input-output structure that suits the ODE solver.

The ODE solver used is either `ode45` or `ode78`, as already discussed and selected in Section 4.3. Together with the initial state and transfer time, the solver takes as input the integration settings, such as the relative and absolute tolerances, which are gathered in the script `parameters.m`.

The trajectory can be plotted, although this step is not always performed. For such, a MATLAB script called `plotTrajectory.m` has been written, which takes as input the variables from `parameters.m` and the output of the propagation module and generates a 3-D figure useful for visualization.

The script `parameters.m` is fundamental to most of the code and includes relevant variables stored under a MATLAB structure array data type, which can then be passed into any function as a single argument. The main elements that this array includes are:

- Information of periodic orbits, including ICs and periods for several orbit families and the selected departure orbit for the simulation.
- Integrator and optimizer settings, including tolerances and the maximum number of iterations.
- Useful orbit dynamics constants, such as gravitation parameters and distances.
- Spacecraft parameters, such as the mass or maximum thrust.

- Variables relevant to the Hill model, which include the non-dimensionalization factors, among others.
- Relevant points of interest, such as position and velocity of L1 and position vectors of Earth and Sun.
- All 2006 RH120 relevant data, such as epochs of interest, ephemeris model, and propagated trajectory in the Hill model.
- Ephemeneris of other celestial bodies during at the epoch ranges of the TCO, such as the Sun and the Moon.
- Settings for graphs and plots, such as font size and line thickness.

7.1.2. Optimization Module

The optimization module is the second fundamental block in the code for this thesis. It encompasses the functions and scripts that generate an optimal low-thrust trajectory. To illustrate this module, a flow-chart is presented in Figure 7.2.

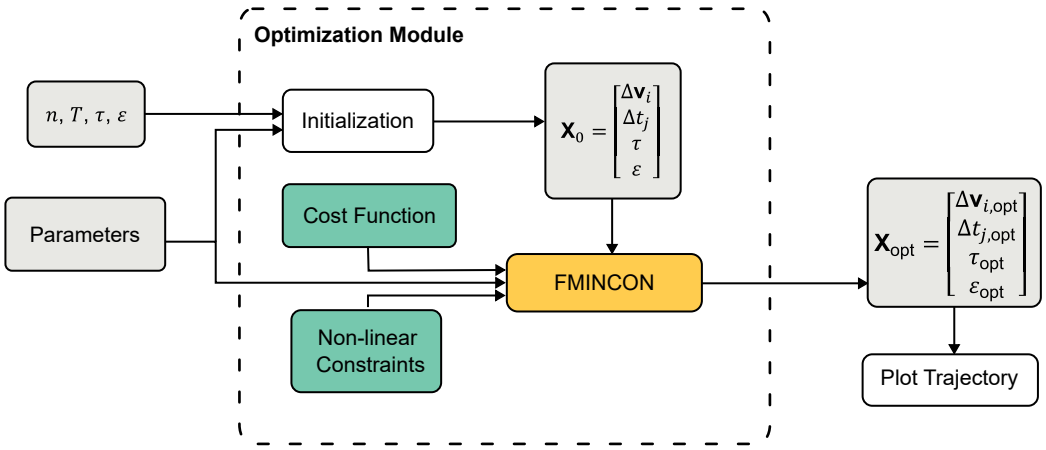


Figure 7.2: Flow chart of optimization module.

The block framed under the dashed line is where the optimization occurs, and it is activated by running the MATLAB script `optimize.m`. At the beginning of the script, the simulation settings are selected, such as the number of trajectory segments n, the initialization departure and target epochs, τ_0 and ε_0 respectively, and the initialization transfer time T_0. At the same time, the script `parameters.m` is run, which generates a structure array with relevant constants and variables to be used in the simulation.

Subsequently, the input is processed to generate the decision variables vector for initializing the algorithm, $\mathbf{X}_0$. These decision variables are input to the optimization solver, the MATLAB built-in function `fmincon`. The other inputs that `fmincon` takes are two functions that evaluate the cost function and constraint functions and their gradients at the decision variables introduced.

The function `fmincon` iterates the decision variables until it converges to an optimum that minimizes the cost function while meeting the constraints. The iteration process is internal and obscure to the user, but it is a gradient-based search that uses the gradient introduced to update the decision variables.

Next, the initialization function, the cost function, and the non-linear constraint functions are explained.

It is noted that the propagation module is accessed on several occasions within the optimization module every time a trajectory needs to be propagated. This occurs within the initialization block and within the non-linear constraints block, as it will be subsequently explained.

Initialization

The initialization is performed by a function called `initialize.m`, which takes as input the initialization departure and arrival epochs, ε_0 and τ_0 respectively, the transfer time T_0, the number of delta-v impulses n, and the structure array from `parameters.m`. The function populates the initialization decision variable vector $\mathbf{X}_0$ (Equation 6.38) with the missing values for Δv_0's and Δt_0's. The steps to generate $\mathbf{X}_0$ from the input parameters are the following:

1. Find departure state by using `propagateDynamics` to propagate the departure orbit initial conditions until initialization epoch ε_0.
2. Find target state by using `propagateDynamics` to propagate the TCO orbit initial conditions until initialization target epoch τ_0.
3. Generate two-impulse trajectory from departure to target in transfer time T_0
   (a) Find departure velocity using `differentialCorrector` function.
   (b) Propagate trajectory using `propagateDynamics`.
4. Find the intermediate initialization epochs by diving the two-impulse trajectory into $n-1$ segments.
5. Find the initialization departure velocity vectors
   (a) Save the direction of the velocity vector at the intermediate points.
   (b) Set the magnitude of the velocity vectors to a small value.

Cost function

The cost function is called `costFcn.m`, and takes as input a decision variable vector and computes the total ΔV. At the same time, it computes the gradient of the cost function with respect to the decision variable vector. The cost function and gradient have been defined in Equations 6.4 and 6.9, respectively. Since a sum of independent variables forms the cost function, the cost function and the gradient are simple calculations. To drive the cost to lower values, it suffices to select small values for the Δv's. However, the complexity of the problem arises when these Δv vectors have to result in a trajectory that meets the problem constraints defined in the following function.

Non-linear Constraints

The non-linear constraints passed to the `fmincon` function are gathered under a function named `nlConst.m`. The function works similarly as `costFcn.m`. It takes as input the vector of decision variables $\mathbf{X}$ (Equation 6.38) and evaluates the equality and inequality constraints and their corresponding gradients. These expressions were provided in Section 6.2:

- Equality constraints: Equation 6.6.

- Equality constraints gradient: Equations 6.11, 6.12, 6.25, 6.34, and 6.41.
- Inequality constraints: Equation 6.39.
- Inequality constraints gradient: Equations 6.35 to 6.37, 6.42 and 6.43.

The function `nlConst.m` starts by computing all the necessary elements to populate the equations above. These elements come from the problem transcription, represented in Figure 6.1. The list of steps that the function takes to compute these parameters subsequently is the following:

1. Find departure state by using `propagateDynamics` to propagate the departure orbit initial conditions until epoch ε.
2. Find target state by using `propagateDynamics` to propagate the TCO orbit initial conditions until epoch τ.
3. Propagate the trajectory arcs from the departure.
   (a) Compute departure velocity from initial orbit conditions at ε and the first Δv from the decision variable vector.
   (b) Use `propagateDynamics` to propagate the trajectory arc and STM for a propagation time set by the first Δt from the decision variable vector.
   (c) Use the state vector of the last epoch as the initial conditions for the next trajectory arc.
   (d) Loop through all the remaining arcs with the Δv's and Δt's from the decision variables vector.
4. With all the transfer information stored conveniently in variables, evaluate the constraints and gradients.
   (a) Evaluate the terminal constraint by comparing the transfer's last and target states.
   (b) Evaluate the gradient of the terminal constraint for the optimization algorithm to drive the decision variables toward meeting the terminal constraint.
   (c) Evaluate the inequality constraints by checking that the Δv's, Δt's, total transfer time, departure, and arrival epoch are within the established boundaries.
   (d) Evaluate the gradient of the boundary constraints for the optimization algorithm to drive the decision variables toward meeting the boundary constraints.

7.1.3. Tudat Module

The third module is that of the external software Tudat, which is internally developed by the Faculty of Aerospace Engineering of Delft University of Technology. The software provides an Astrodynamics toolbox which includes propagation and optimization modules that can be accessed via Python.

Two are the main capabilities of Tudat that are used in this thesis are hereby explained:

- The SPICE module allows access to JPL's SPICE database of celestial body ephemeris while also allowing to import data of TCOs retrieved from NASA's Small-Body Database through the function `load_kernel`. At the same time, the functions `get_body_cartesian_state_at_epoch` allow obtaining the state of a celestial body in the reference frame specified – normally the EMOJ2000 – and relative to the Earth. This is useful for retrieving states of the TCO 2006 RH120

while also the Sun's position relative to the Earth to compute the transformation between inertial and synodic frames.

- the `numerical_simulator` function, which allows propagating bodies with specific settings, such as reference frame EMOJ2000 and relative to the Earth. This simulation is used to propagate the initial conditions of the TCO 2006 RH120 with a simplified dynamical model including Earth, Sun, and Moon's point-mass gravity fields and assess the magnitude of perturbing accelerations such as SRP or spherical harmonics.

All the generated data can be transferred to MATLAB by means of Python's package `scipy.io` with the function `savemat`.

7.2. Code Verification

The algorithm verification process is important to ensure that the code written works as intended. For such a task, all the functions are tested independently through unit tests. A list of all the functions and scripts with their dependencies is gathered in Table 7.1.

Unit tests are performed to check that every code unit performs as expected. For such, fundamental calculations are tested. Each function has a verification method tailored to them. The unit test verification is subsequently explained and discussed for each function.

Dynamics Model

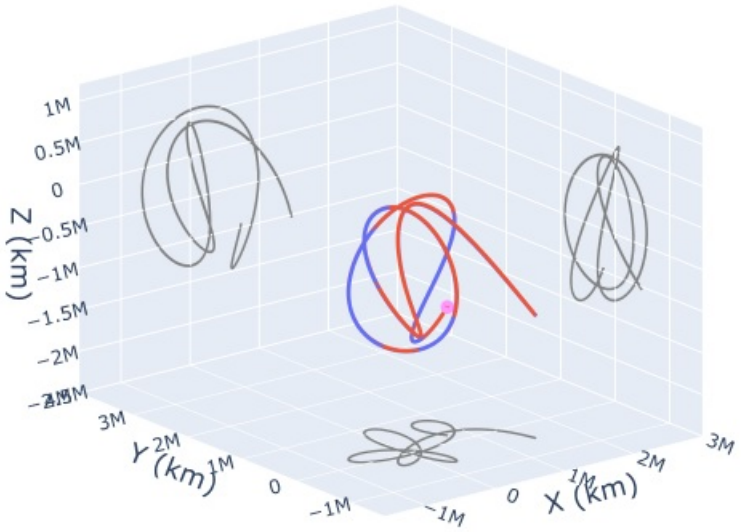
The function `Hill3BP` has two main units that must be verified. These are the model equations and the STM calculation.

To verify the dynamics model, the ideal test would consist of replicating a benchmark solution and measuring the tolerance. However, due to a lack of benchmark data in the Hill three-body problem, the following alternative tests have been run to verify the model:

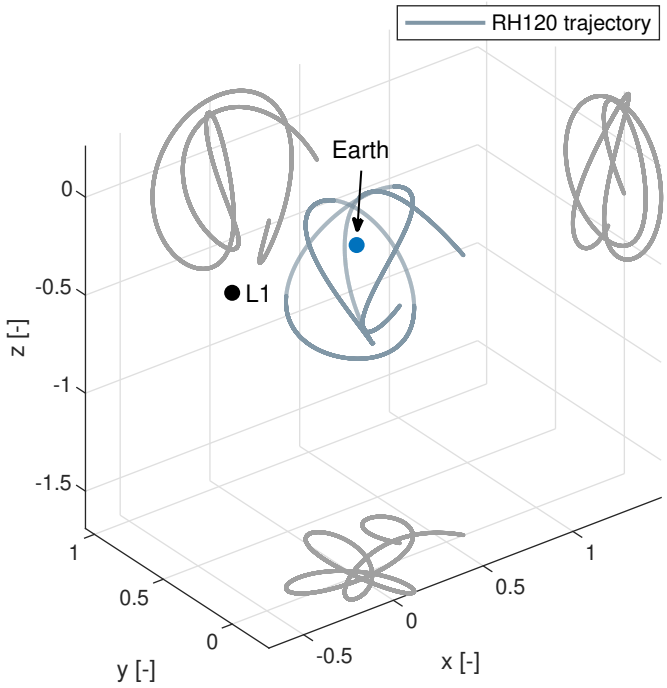
1. The trajectory of the TCO has been propagated in the dynamical model from initial conditions at a given epoch, and the Jacobi Constant has been computed for every propagated epoch. The test shows that the Jacobi constant remains indeed constant for every propagated point, resulting in a successful verification test as shown in Table B.1 in Appendix B.
2. The trajectory of the TCO has been propagated in the Hill model with the same initial conditions as Takahashi et al., 2022. Although numerical data is not provided, the resulting trajectory can be plotted and compared with the plots in Takahashi et al., 2022. Upon visual inspection of the figures, both trajectories look the same, which is considered a successful pass of the test. The figures are shown in Figure 7.3.

The verification tests for the dynamics model are more qualitative than numerical and do not provide values for the numerical tolerance of the model. However, they give enough reason to believe that the model has been coded appropriately, and therefore it is assumed that the model is verified.

To test the STM, a nominal trajectory is run from a certain initial state, and the STM is computed. Then, three tests are performed to ensure that the result obtained is correct:



(a) 2006 RH120 trajectory in the Hill model from Takahashi et al., 2022.



(b) 2006 RH120 trajectory in the Hill model computed for verification.

Figure 7.3: Verification of dynamical model by comparing the propagated trajectory of 2006 RH120 to Takahashi et al., 2022.

Table 7.1: List of MATLAB functions and scripts.

Function	Description	Dependencies
	Main Functions	
`Hill3BP`	Implements dynamics Hill model equations of motion and STM to propagate using ODE solver	
`propagateDynamics`	Propagates initial position-velocity state in the Hill model for a specified transfer time	`Hill3BP`, `ode45`, `ode78`
`differentialCorrector`	Computes departure velocity to arrive at a given target	`propagateDynamics`
`costFcn`	Evaluates the cost function and its gradient from decision variable vector	
`nlConst`	Evaluates trajectory constraints and their gradient from decision variable vector	`propagateDynamics`, `Hill3BP`
`initialize`	Generates initialization trajectory decision variables from simulation settings	`propagateDynamics`, `differentialCorrector`
`optimize.m`	Optimizes the decision variables that define a trajectory given a list of simulation settings	`findOrbit`, `fingICs`, `propagateDynamics`, `fmincon`, `plotTransfer`
	Auximiary Functions	
`emo2synodic`	Transforms vectors in EMOJ2000 to Synodic rotating frame	
`ssJ2JD`	Transforms time in Seconds since J2000 to Julian Date	
`jacobiConstant`	Computes the Jacobi constant given a position-velocity state	
	Plots	
`plotRelevantPoints`	Plots TCO trajectory, Earth and L1	
`plotTrajectory`	Plots a trajectory as given by state history	
`plotPeriodicOrbit`	Plots periodic orbit specified by orbital period	
`plotVelocities`	Plots the ΔV impulse vectors of a transfer trajectory	

1. The determinant of the STM has to be equal to unity.
2. A second propagation with a small deviation in the initial state is run. It is checked that the deviation in the final state equals the product of the STM and the deviation in the departure state, *i.e.* $\delta\mathbf{s}_f = \Phi\,\delta\mathbf{s}_0$. Results of this test are gathered in Table B.2
3. The STM is computed using numerical integration using finite differences, and the result is compared. Results of this test are gathered in Table B.3.

By performing tests 2 and 3, a typo was found in the computation of the STM in `Hill3BP`. The typo was corrected, and afterward, the unit tests passed, resulting in a numerical accuracy of the STM of O(10^{-6}) m for position and O(10^{-12}) m/s for velocity, which are well below the numerical requirements

established.

Differential Corrector

The `differentialCorrector` function can be verified by inspecting its outcome. Once the STM and dynamics model is verified, the outcome of the DC function is a departure velocity that has the spacecraft arrive at a specified target point with given tolerances. Therefore, the function is verified by simply inspecting that these characteristics are met.

During the many simulation runs, it was noted that the algorithm does not always converge and is dependent on the initial conditions but is also sensitive to the arrival point. Once the target is behind the Earth, the resulting trajectory is greatly perturbed by its gravitation, making it very difficult to converge. In such cases, increasing the maximum number of iterations for the algorithm helps converge to the right solution, but it is not guaranteed. A flag is triggered when the algorithm does not converge in such cases.

It should also be noted that the STM is an essential aspect of the algorithm. Before verifying this matrix, an error was made, and the DC had difficulty converging. However, even with errors in the STM, the algorithm managed to converge in most of the cases.

Auxiliary Functions

The three auxiliary functions from Table 7.1 are verified as follows:

- `emo2synodic` – The transformation from an EMOJ2000 frame to a synodic frame is verified using a test with benchmark data. The test replicates a position-velocity state of RH120 at a specific epoch in the Hill model from its state in the EMOJ2000 frame. The epoch initializing the RH120 orbit is used for such a test. The state in the frame EMOJ2000 at such epoch is obtained from SPICE and accessed through Tudat's interface. After running the test, the results are gathered in Table B.4, showing that the computed synodic state matches the benchmark to 40 km accuracy, while the velocity matches 5 mm/s. Both position and velocity accuracy are above the numerical tolerance requirements selected for the thesis, which are set to 10 m and 1 mm/s, respectively. However, the precise accuracy of the benchmark solution is not known, as it may be possible that our transformation is more accurate since it uses ephemeris data for the position of the Sun. Nonetheless, this potential error must be assumed for all the calculations involving this transformation. The numbers are, however, close enough to consider the transformation verified.

- `ssJ2JD` – This function that transforms Seconds since J2000 to Julian Date is verified by the external software Tudat, which already has verified built-in functions for computing such transformation. The same input has been given to both software, and the same output has been obtained, so the function has passed the verification test, and the results are gathered in Table B.6.

- `jacobiConstant` – A test to verify the function has been implemented, consisting of running the function selecting as input a position-velocity state vector of which the Jacobi Constant is already known. The benchmark data for this test are obtained from Kalantonis, 2020. The function has passed the verification test, and the results are gathered in Table B.5.

Cost Function and Constraints

Different tests are run to verify their proper working under this category, encompassing the functions `costFcn` and `nlConst` from Table 7.1. First, evaluating the cost function and constraints for a given set of decision variables is tested. For a known set of Δv's, the total cost function can be checked by adding all the values and comparing them with the function's output. At the same time, for a list of decision variables, the expected outcome of the constraint function is known beforehand since constraints represent the physical limitations of the problem that has been chosen. The part of these two functions that do not have an evident physical interpretation is the gradient of the functions. For such, `fmincon` provides a handy tool that checks whether the input functions' gradients are correct. It evaluates them internally by performing numerically deriving them using finite differences. It should be noted that the gradient checks initially helped identify several mistakes in the analytical derivatives until they were adjusted and passed the tests, verifying the method.

Optimization

Under this category, two functions are verified: the script `optimize.m` and the function `initialize`.

Verification of the latter is conducted similarly to the cost function and constraints in that the results have a physical interpretation that can be observed. In this case, it should be observed that the initialization trajectory converges to the target, that the Δts are equal and add up to the total transfer time, and that the Δv's point into the velocity direction, *i.e.* are tangent to the trajectory. The function has been tested for different inputs and has been verified.

Given the lack of benchmark data, the verification of the `optimize.m` script is also performed by inspecting the results. Two things are checked: first, the trajectory generated meets all the constraints. Second, the search history has a profile that converges to a solution that reduces the Δv. On top of this, measures have to be taken to ensure that the algorithm does not converge to a local minimum. This is done case-by-case and consists of modifying the inputs and searching restricted regions for different solutions.

7.3. Model Validation

Validation of the code is fundamental to ensure that the correct problem is being solved and that the simulations properly represent the behavior in physical reality. In astrodynamics, comparing results with reality tends to be difficult since innovative aspects take decades to be incorporated into missions. Therefore, it is common to validate results by comparing them with the outcome of other studies available in literature to ensure that reasonable assumptions have been made and that the results obtained are comparable.

In this particular study, it is not possible to replicate other studies' results with the developed algorithm since it uses a combination of a dynamics model and an optimization method that has not been found in the literature. However, it is still possible to roughly compare the cost of transfers with other studies targeting TCOs in a similar dynamical environment to ensure that the results are reasonable and that the transfers have a similar shape.

The assumptions taken regarding mass, maximum thrust, and trajectory constraints have already

been validated when selecting them in Section 2.4. In Chapter 8, it will be observed that the family of reproduced periodic orbits around Lagrangian points matches the structure of families found in the literature. At the same time, all the continuous low-thrust trajectories are validated in two ways: first, the shape of the converged transfers seems to leave the periodic orbits following the shape of manifold trajectories with small thrust input, and the overall structure of the transfer seems to navigate such manifolds while still drifting off due to the application of the continuous thrust. Secondly, the converged trajectories present a smooth thrust profile, with the vectors' orientation also presenting smooth transitions that adapt to the transfer's velocity vectors. These converged transfers are similar to the continuous low-thrust transfers generated in other studies, such as Chikine, 2021. And third, the overall ΔV for the transfer trajectory is of the same order of magnitude as other studies targeting TCOs, Brelsford et al., 2016 (797 m/s) or Chyba and Patterson, 2019, 725 m/s. At the same time, the approach and assumptions are fundamentally different. The solutions obtained in this thesis range from 200 m/s to 900 m/s.

These arguments support the premise that the results obtained in this thesis appropriately replicate actual behavior and are therefore considered validated.

8

Results

This chapter gathers the results of applying the low-thrust optimization algorithm designed to find optimal transfers between L1 periodic orbits and TCO 2006 RH120. The chapter begins with a fundamental step: tuning the algorithm to solve the specific problem at hand. Then, the results of the departure orbit analysis are presented. Three families of periodic orbits are scouted to characterize the optimum transfers departing from them.

8.1. Algorithm Tuning

The tuning of the algorithm is crucial to obtain an optimal solution. This section looks at three aspects that affect a simulation's outcome: the number of impulses selected, the initialization parameters, and the search space boundaries. This analysis helps us understand how robust the algorithm is to different inputs and will allow searching for optimal transfers in an efficient way.

This analysis starts by pre-selecting the simulation settings to generate a baseline solution. The baseline solution will be used to compare the algorithm's performance when tuning the settings. The settings from this baseline simulation are gathered in Table 8.1 and have been selected based on insight gained while developing and testing the algorithm. The baseline simulation will have 30 impulses, a maximum transfer time of 100 days (short transfer), and a target epoch range between 200 and 240 since the TCO enters the Hill sphere. The epoch range was selected after identifying a potential low-cost region when conducting a preliminary search with the DC algorithm.

The baseline transfer with these settings is plotted in Figure 8.1, and its main results are summarized in Table 8.2. The main characteristics of the baseline transfer obtained are the following: a delta-v cost of 789 m/s, a transfer time of 99.4 days, close to the upper limit from Table 8.1, and a target epoch of 227 days since the TCO enters the Hill sphere.

Table 8.1: Simulation settings for baseline trajectory used for algorithm tuning.

Parameter	Value	Units	Description
Algorithm			
n	30	[-]	Number of impulses
Initialization			
T_0	50	[days]	Initial transfer time
τ_0	220	[days]	Initial target epoch
ε_0	0	[days]	Initial departure epoch
Boundaries			
T_{max}	100	[days]	Maximum transfer time
τ_{range}	$[200,\ 240]$	[days]	Range of target epoch
$\varepsilon_{\text{range}}$	$[-0.5,\ 0.5]\,P$	[days]	Range of departure epoch

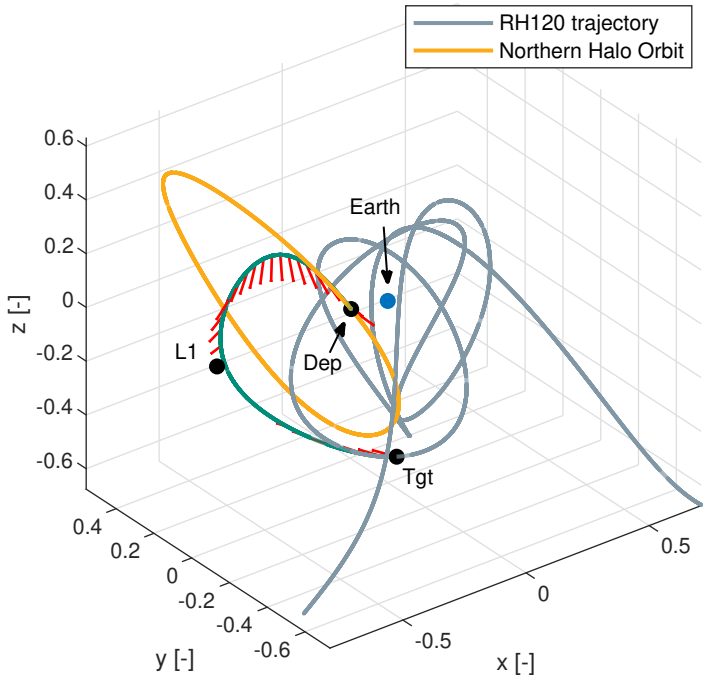


Figure 8.1: Baseline transfer trajectory used for algorithm tuning.

Table 8.2: Optimization results for baseline transfer used for algorithm tuning.

Parameter	Optimum	Units	Description
ΔV	789.2400	[m/s]	Total transfer ΔV
T	99.4016	[days]	Total transfer time
τ	227.4611	[days]	Arrival epoch
ε	-0.2024P	[days]	Departure epoch
Δt_{mean}	3.4276	[days]	Mean arc length

Next, the tuning analysis is presented, covering the three different settings categories from Table 8.1: the number of trajectory impulses, the algorithm initialization parameters, and the search space boundaries.

8.1.1. Number of Impulses

The number of delta-v impulses n measures the discretization selected to describe the continuous thrust trajectory. It is equivalently described by the number of trajectory arcs since it equals $n-1$. This number is chosen to make the discretization as realistic as possible while computationally viable. On one side of the spectrum, this number is high enough such that a smooth line represents the thrust profile. On the other side of the spectrum, we find a trajectory composed of a few impulses, which is an inadequate representation of the behavior of a continuous thruster.

By increasing the number of delta-v impulses, the decision variable vector and constraints matrices increase in size, complicating the convergence, slowing the optimization process, and increasing the memory used for a single simulation. Therefore, it is essential to find an appropriate number of impulses that provides a good trade-off between the validity of the model and the computational requirements.

Several trajectories have been run with different numbers of impulses, and the results are gathered in Table 8.3.

Table 8.3: Algorithm performance parameters for a different number of delta-v impulses (n).

n	function evaluations	iterations	ΔV [m/s]	runtime [s]	Δt_{min} [days]	Δt_{max} [days]	Δt_{avg} [days]	T_{opt} [days]
5	208	77	774.10	7	20.00	29.58	24.36	97.30
10	694	205	781.28	24	8.73	14.56	10.91	98.55
25	622	168	787.58	24	3.49	5.01	4.13	99.29
35	788	198	789.27	33	2.92	4.14	3.42	99.40
50	1674	388	784.38	90	1.61	2.68	2.03	99.77
75	2448	555	789.21	185	1.11	1.68	1.34	99.80
100	2907	697	788.57	316	0.84	1.25	1.00	99.85

Several conclusions can be extracted from these results. First, a low number of impulses corresponds to a low ΔV transfer cost. This was expected since a low number of impulses corresponds to more efficient high-impulse transfers (see Section 6.5) that do not accurately represent the continuous nature of the electric propulsion system. However, it is observed that from 25 points onwards, the ΔV stagnates. We can also observe that for 25 and more impulses, the average arc length is well within the established limits of half and eight days (see section 6.1). However, this number highly depends on the total transfer time of the obtained solution.

In light of these results, a rule of thumb for selecting the number of impulses is to start with a number equal to half the initialization transfer time (*i.e.* 25 impulses for a 50 days initialization transfer) and then adjust it if the resulting optimum arc lengths exceed the boundary set in Section 6.1 (eight days). This allows considering the results to be a good representation of the actual behavior of the continuous thrust and, therefore, the resulting ΔV to be reliable.

8.1.2. Initialization

To measure how robust the algorithm is to changes in the initialization conditions, the initialization parameters are varied from those of the baseline trajectory, and the results are inspected. The initialization parameters varied are the departure epoch ε_0, the transfer time T_0, and the target epoch τ_0. The results of the simulations are gathered in Table 8.4. Note that the orbit period P has been dropped from the definition of ε for simplicity.

Table 8.4: Tuning results for initialization parameters. * indicates transfer not converged.

Parameter varied	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]
$\varepsilon_0 = -0.1$	789.2648	99.4019	227.4557	-0.2023	3.4277
$\varepsilon_0 = -0.5$	789.2731	99.4028	227.4470	-0.2024	3.4277
$\varepsilon_0 = 0.1$	789.2705	99.4028	227.4474	-0.2024	3.4277
$\varepsilon_0 = 0.5$	789.2717	99.4028	227.4474	-0.2024	3.4277
$\tau_0 = 200$	789.2715	99.4027	227.4475	-0.2024	3.4277
$\tau_0 = 210$	789.2705	99.4027	227.4476	-0.2024	3.4277
$\tau_0 = 230$	789.2703	99.4027	227.4474	-0.2024	3.4277
$\tau_0 = 240$	789.2707	99.4027	227.4476	-0.2024	3.4277
$T_0 = 30$	789.2705	99.4027	227.4476	-0.2024	3.4277
$T_0 = 40$	789.2705	99.4027	227.4476	-0.2024	3.4277
$T_0 = 60$	789.2701	99.4027	227.4478	-0.2024	3.4277
$T_0 = 70$*	1126.0	100.000	239.6815	-0.4882	3.4483

Overall, results in Table 8.4 show that the simulation is not sensitive to initialization parameters. The initialization departure epoch ε_0 has been varied from zero to $\pm P/2$, and the algorithm has managed to output the same optimum at $\varepsilon_0 = -0.2023P$. Similarly, the algorithm has been proven to reach the same optimum independent of the target epoch selected. It should be noted that the target epoch was varied within the search range, and it is left to check whether this is still the case when the range is further extended to other regions of the TCO with different characteristics. Regarding the transfer time, except for $T = 70$, which is a non-converged solution, the algorithm converges to the same solution irrespective of the initialization transfer time selected.

8.1.3. Boundaries

It is important to understand the algorithm's behavior when certain search space boundaries are modified. For such, the parameters bounding the algorithm's search space have been modified, and the results are gathered in Table 8.5. The parameters varied are the maximum transfer time, the maximum target epoch, and the range of departure epochs.

The maximum transfer time is a crucial parameter with important effects on the solution obtained by the algorithm. It was observed from the previous studies that the solution converged to a transfer time of 99.4 days, very close to the limit set at 100 days. This hinted that increasing the maximum allowed

Table 8.5: Tuning results for search space boundaries. * Indicated trajectory not converged.

Parameter varied	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]
$T_{\text{max}} = 30$	846.4310	35.1190	200.9090	-0.0128	1.2110
$T_{\text{max}} = 70$	899.2362	69.9442	215.5416	-0.1147	2.4119
$T_{\text{max}} = 80$	829.0552	79.8248	220.2520	-0.1451	2.7526
$T_{\text{max}} = 150$	740.3624	149.2478	229.2004	-0.3361	5.1465
$T_{\text{max}} = 200$	676.3838	194.4887	235.8390	-0.4852	6.7065
$T_{\text{max}} = 250$	672.1220	199.2304	237.1613	-0.4883	6.8700
$\varepsilon_{\text{range}} = [0, 0.25]$	869.5966	90.8997	231.0389	8.1460e-04	3.1345
$\varepsilon_{\text{range}} = [0.25, 0.5]$	885.2174	69.7962	239.8787	0.2507	2.4068
$\varepsilon_{\text{range}} = [0.5, 0.75]$	798.6623	99.7043	224.0324	0.7479	3.4381
$\varepsilon_{\text{range}} = [0.75, 1]$	789.4131	99.3893	227.5184	0.7991	3.427
$\tau_{\text{range}} = [0, 250]$	789.2662	99.4030	227.4581	-0.2024	3.4277
$\tau_{\text{range}} = [0, 100]$	242.1163	98.8957	76.3128	-0.4817	3.4102
$\tau_{\text{range}} = [100, 200]$*	1216.4	100.0087	140.1878	0.1202	3.4486
$\tau_{\text{range}} = [200, 300]$	789.1691	99.4088	227.6648	-0.2020	3.4279

transfer time would result in more optimal solutions. This is indeed the case, as can be observed in Table 8.5 for $T_{\text{max}} > 150$ days, where the overall ΔV is reduced. However, the transfer time boundary is a design parameter that may depend on mission constraints. In this case, we are less interested in long transfer time solutions since TCOs are hard to identify before entering the Hill sphere and stay captured for a short period. Therefore, shorter transfer times are preferred. At the same time, when looking at the trajectory generated by longer transfer time, it is observed that they fall into a category of different orbits which revolve around the Earth before arriving at the target (see Figure 8.2). These solutions are not preferred for this algorithm since the single-shooting nature is not the most reliable when revolutions are performed. When running preliminary simulations, these solutions were found very difficult to converge and were very sensitive to initial conditions.

Regarding the tests that vary the range for departure epochs, an orbit has been divided into four segments to check whether the algorithm is missing any optimum when the whole orbit is selected as the range. It is observed that no other optimal departure locations are found when searching specifically in other segments of the orbit. This is not the case when varying the range of target epochs.

Several tests have been run varying the range of target epochs. First, the minimum target epoch is set to zero, and the algorithm searches for a solution within a large segment of the TCO orbit for up to 250 days. It is observed that, for the initialization time of 220 days, the solver still manages to converge to the baseline solution. To check that the solution is not converging to a local minimum, the target epoch range has been subdivided into smaller segments of 100 days, similar to what was done in the tests with the departure range. It is observed that the solver was indeed missing a very significant minimum found in the range of zero to 100 days with an initialization epoch of 50 days. The transfer cost found is 242 m/s, which translates into a delta-v reduction of almost 70% with respect to

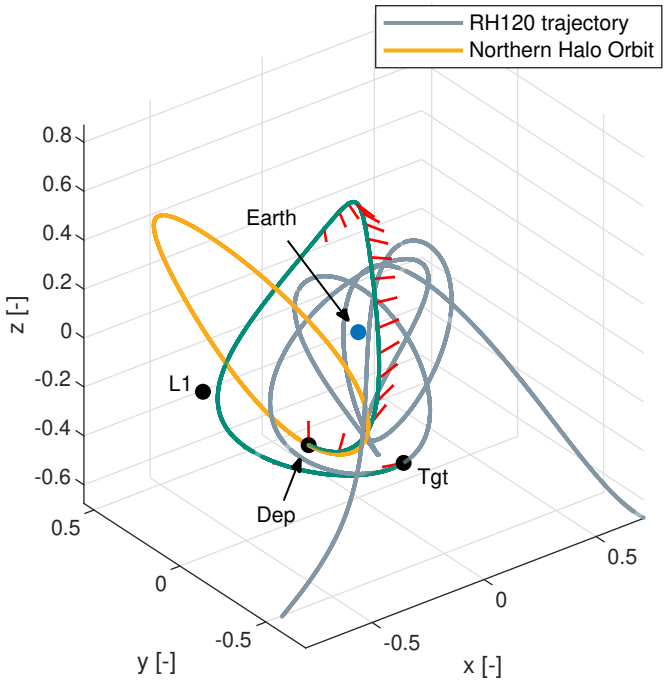


Figure 8.2: Simulation for $T_{\text{max}} = 250$ days with one revolution around the Earth.

the baseline solution. For the range of 100 to 200 days, the solver did not converge, and for the range of 200 to 300 days, the solver converged to the baseline solution. It is then concluded that subdividing the target range into segments of 100 days is fundamental to finding optimal trajectories.

It should be noted that for some simulations, the algorithm does not converge. Two reasons have been identified for this behavior: the solver has reached the maximum number of iterations and needs more to converge, or the settings are too strict for the constraints to be met.

Inspecting the solution obtained in the target epoch range of 0 to 100 days, it is noted that the transfer structure differs significantly from the baseline result. The trajectory is plotted in Figure 8.3. It is observed that the transfer targets an entirely different region of the TCO, and the departure epoch sets the spacecraft in a different direction as compared with the baseline transfer in Figure 8.1. This type of transfer will be further discussed in Section 8.2.

8.1.4. Tunning Conclusions

The tuning test has resulted in beneficial recommendations for the algorithm to find optimal transfers to 2006 RH120. First, regarding the discretization in trajectory arcs, it is concluded that any number above 25 leads to a reasonable approximation of the continuous-thrust behavior for the transfers studied with a transfer time of roughly 100 days. The initialization conditions have shown that the algorithm is reasonably robust to initialization when kept within the boundaries selected for the problem. Finally, regarding the search space boundaries, it is concluded that the maximum transfer time T_{max} significantly

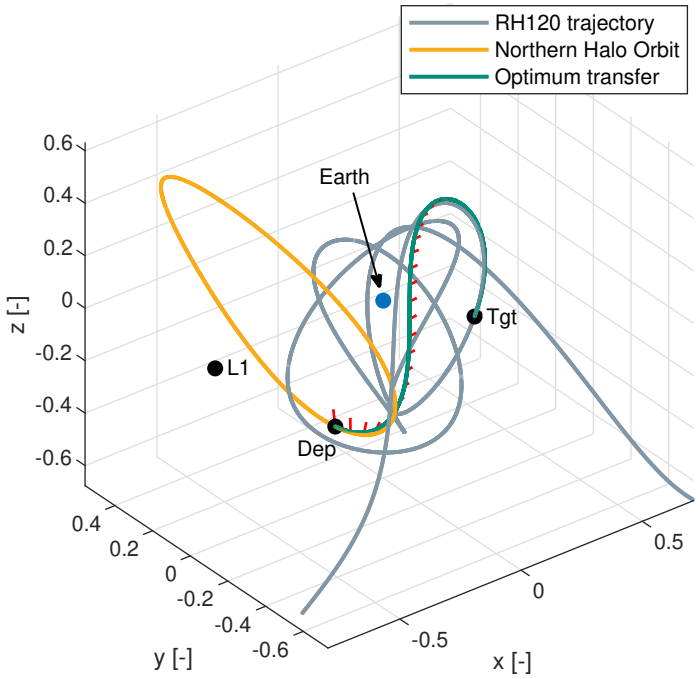


Figure 8.3: Optimum transfer trajectory targeting epoch range of 0 to 100 days.

impacts the optimal trajectory found and should be tuned according to mission design specifications. The study of the departure epoch range $\varepsilon_{\text{range}}$ has concluded that the algorithm does a good job of finding the optimal departure epoch without the need to split the departure into smaller search regions. On the contrary, the study of the arrival epoch range τ_{range} has shown that it is convenient to limit the search space to avoid missing potential optimal solutions. It is recommended to split the search space into several independent simulations covering a range smaller than 100 days.

8.2. Departure Orbit Analysis

In light of the results obtained in the tuning section, simulations are prepared to characterize the set of potential transfer trajectories from orbits around L1 to rendezvous with 2006 RH120. This study aims to develop recommendations on the most optimal departure orbits while providing information on the cost in terms of ΔV and transfer time when departing from a range of available orbits. This study shows the potential of the algorithm developed to perform a preliminary study of the transfer opportunities to a TCO that can be easily extended to other future observed TCOs as they are found.

Three families of trajectories are analyzed to study the attractive departure orbits that may be used in a mission: the northern and southern halo orbits and the vertical Lyapunov family. An analysis of these three families and several different orbits within each family covers a broad range of departure possibilities. The orbits are received as a list of position-velocity initial states and the corresponding orbit periods, which are propagated in the Hill dynamical model using the developed software. The results are subsequently analyzed.

8.2.1. Northern Halo Orbits

Since the families of orbits obtained include a high number of orbits obtained through continuation, a subset of orbits has been selected to represent the different types within the family. For such, ten are selected from the 60 northern halo orbits, with an even spreading over the family. The same is done with the southern halo orbits. The selected sub-set of orbits is plotted in Figure 8.4, and their characteristics are gathered in Appendix C, Table C.1.

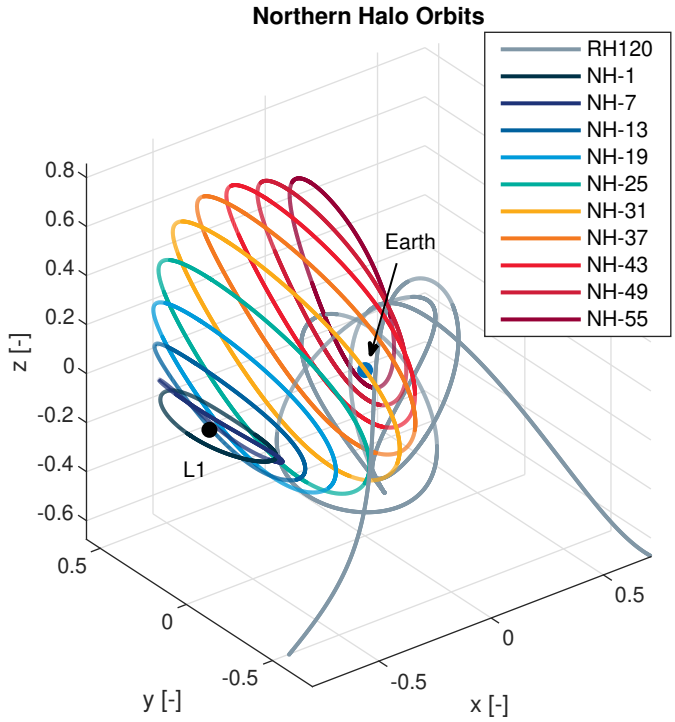


Figure 8.4: Set of northern halo orbits selected for simulations.

The low-thrust trajectory optimization algorithm has been run for a sub-set of orbits from each family targeting any point in the TCO trajectory up until $\tau = 300$ days. As concluded in Section 8.1, the most reliable means of scouting the whole target trajectory is to subdivide it into three different target ranges: from zero to 100, from 100 to 200, and from 200 to 300 days since Hill entry. From the three ranges, the trajectories that lead to the lowest ΔV from each departure orbit have been gathered in Table 8.6.

From the results, the first thing noticed is that some simulations led to surprisingly low ΔV costs for the transfer, ranging from 200 m/s in the best case to 700 m/s in the worst. It is observed that the transfers that have the lowest ΔV depart from orbits comprised between NH-19 and NH-31, which have Jacobi constants ranging between 2.9 and 1.5. The lowest ΔV found departs from NH-25, which has the Jacobi constant closest to that of the TCO 2006 RH120 (2.22 versus 2.15). Another aspect noticed is that, when looking at the target epoch, the algorithm converged to a solution close to entry in the Hill sphere, ranging from 40 to 85 days. The solutions found when specifically searching in ranges from 100 to 300 days led to transfer trajectories with higher ΔV. At the same time, it is observed that for these trajectories, the converged transfer time is always close to the upper limit of 100 days. This

Table 8.6: Simulation results for transfers departing from northern halo orbits with short transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
NH-1	4.0053	700.0292	98.4526	43.8175	-0.9027	4.1022	Figure 8.5a
NH-7	3.9057	580.8656	99.2471	60.7606	-0.9017	4.1353	Figure 8.5b
NH-13	3.5081	407.2876	98.8278	75.1418	-0.8766	4.1178	
NH-19	2.9172	268.6776	98.5853	72.5024	-0.8771	4.1077	
NH-25	2.2193	214.1788	98.6243	76.2834	-0.8588	4.1093	Figure 8.5c
NH-31	1.5570	298.6782	99.0607	77.8442	-0.7773	4.1275	
NH-37	1.1556	366.6617	98.9729	81.6914	-0.6926	4.1239	
NH-43	1.0718	455.5223	98.7140	85.3124	-0.6215	4.1131	Figure 8.5d
NH-49	1.1800	554.8885	98.6809	85.7602	-0.5424	4.1117	
NH-55	1.3819	623.2124	98.7516	84.1166	-0.4701	4.1147	

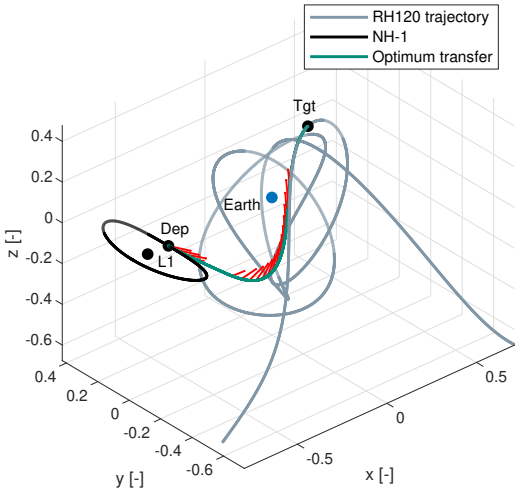
issue was already identified when tuning the algorithm in Section 8.1 and will be addressed further in this section.

The simulation results have converged to different types of transfers depending on the departure halo orbit. A selection has been plotted in Figure 8.5 to understand the results better.

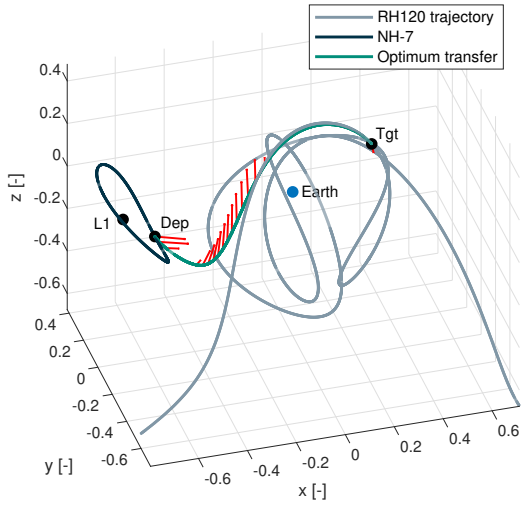
Figure 8.5a displays the transfer to the earliest rendezvous epoch in the TCO trajectory, at 43 days since entry in the Hill sphere of the Earth, departing from the smallest northern halo orbit. The transfer cost in terms of ΔV is 700 m/s, which is considered feasible for the spacecraft, although far from the best that can be found. The transfer time is close to 100 days, meaning the TCO must be identified before entering the Hill sphere for the transfer to be feasible for a real-time mission. Figure 8.5b shows the transfer from a slightly larger orbit with a similar period, which results in a similar result as obtained in Figure 8.5a. The rendezvous location is found at a later epoch of 60 days, which is still considered relatively early in the path of the TCO within the Hill sphere. The transfer cost found is 17% smaller than departing from NH-1. As the departure orbit increases, the transfer with the lowest cost is found for departing from NH-25. It can be observed in Figure 8.5c how the departure conditions already set the spacecraft in a trajectory that almost matches that of the target TCO, requiring a small input from the low-thrust engine to rendezvous. Figure 8.5d shows the converged transfer departing from a larger halo orbit far enough from L1 that almost resembles a vertical orbit around the Earth. The transfer cost is double that of NH-25 while still being significantly low, at 455 m/s.

Shifting the analysis to the transfer time, it was already discussed in Section 8.1 the issue of having trajectories converge to the upper limit of 100 days. However, the topic is further analyzed since the issue is common to every generated trajectory for this family of orbits. For such, the simulations departing from northern halo orbits have been repeated, allowing longer transfer times of up to 250 days to check whether significantly lower cost transfers are found. The results are gathered in Table 8.7. A comparison between the delta-v cost of extending the transfer time and the short transfers are depicted in Figure 8.6. Relevant trajectories are plotted in Figure 8.7.

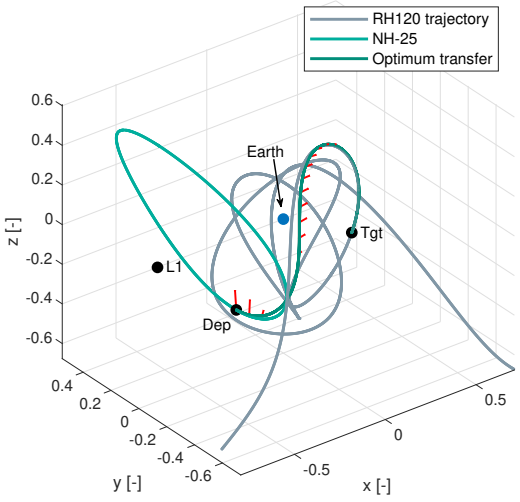
The results show how, indeed, lower-cost transfers are obtained in general. However, the savings



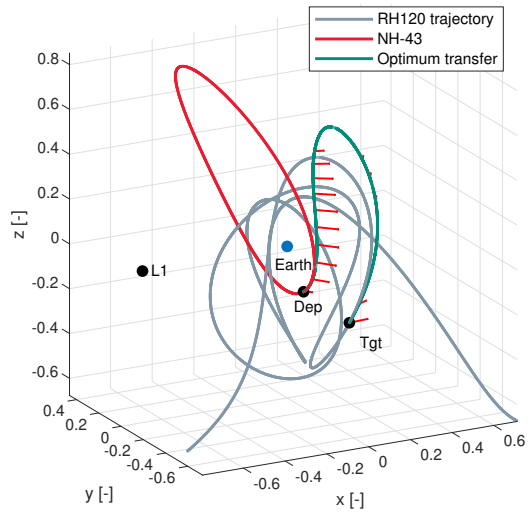
(a) $\Delta V = 700$ m/s, $T = 99$ days, $\tau = 43$ days.



(b) $\Delta V = 580$ m/s, $T = 99$ days, $\tau = 60$ days.



(c) $\Delta V = 214$ m/s, $T = 98$ days, $\tau = 76$ days.



(d) $\Delta V = 455$ m/s, $T = 98$ days, $\tau = 85$ days.

Figure 8.5: Optimal transfers from northern halo orbits with short transfer times.

are not significant. This behavior is illustrated in Figure 8.6, where the lines almost overlap for the first up to NH-25, and the long transfers perform better as the orbit count increases. In the case of departing from NH-25, which corresponds to the lowest cost transfer found from short transfer times, the cost when allowing for longer transfer time is the same, while resulting in a less convenient transfer for being longer and arriving significantly later in the trajectory of the TCO within the Earth's Hill sphere. The difference in the trajectory is shown in Figure 8.7a, where it can be seen that the transfer is very similar, only different in the late rendezvous location. Other orbits only present savings of tenths of meters per second, while the greatest saving is found for NH-55 at 161 m/s. Figure 8.7b shows the transfer from NH-35. It can be observed how, similarly to NH-25, the converged solution for longer transfer time loops around the Earth. This result is not convenient for the algorithm developed since

Table 8.7: Simulation results for transfers departing from northern halo orbits with long transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
NH-1	4.0053	588.9951	202.4398	146.2637	-0.7839	8.4350	
NH-7	3.9057	565.2599	172.4481	142.7116	-0.8827	7.1853	
NH-13	3.5081	426.9549	174.0119	144.1472	-0.8780	7.2505	
NH-19	2.9172	293.6346	175.1371	144.9039	-0.8721	7.2974	
NH-25	2.2193	216.2384	176.7622	145.0118	-0.8550	7.3651	Figure 8.7a
NH-31	1.5570	253.3309	184.1661	145.4349	-0.8109	7.6736	
NH-37	1.1556	348.4713	189.9137	146.2956	-0.7327	7.9131	Figure 8.7b
NH-43	1.0718	381.8877	186.9039	147.4151	-0.6487	7.7877	
NH-49	1.1800	424.6109	186.1202	147.3465	-0.5613	7.7550	
NH-55	1.3819	462.5440	186.2715	147.6074	-0.4604	7.7613	

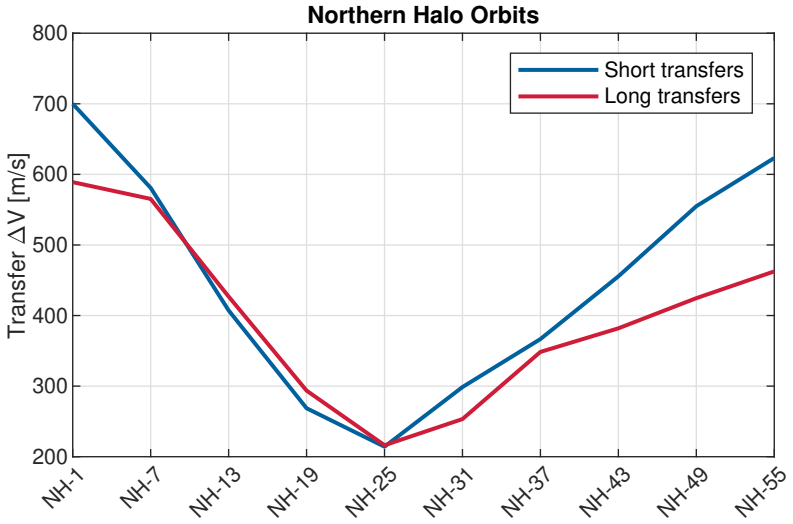


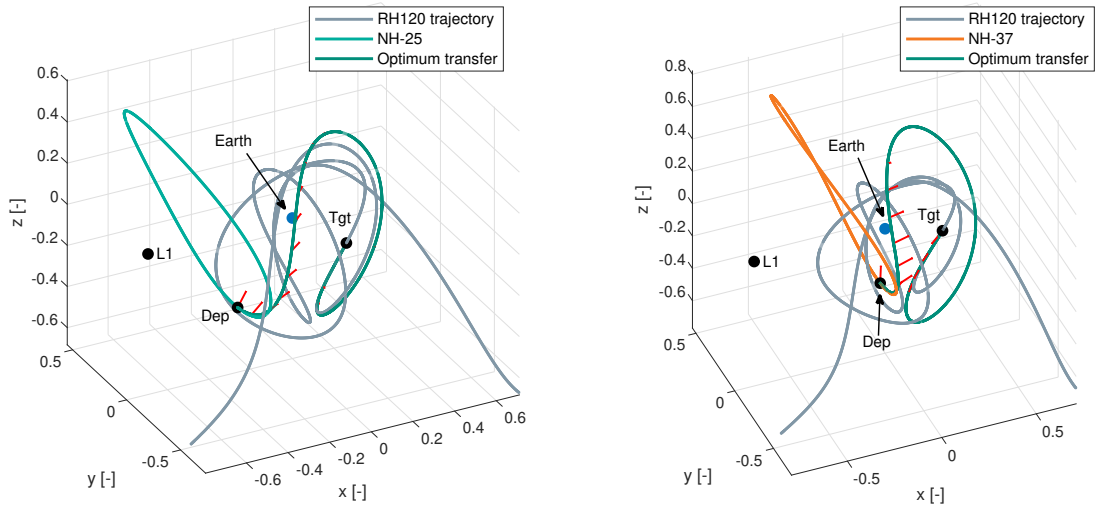
Figure 8.6: Comparison between delta-v cost of short and long transfer times departing from northern halo orbits.

such loops are inadequate for the single-shooting strategy, and the solutions are very sensitive to initial conditions and algorithm settings.

8.2.2. Southern Halo Orbits

The Hill three-body problem's symmetry allows the mirroring of the northern periodic orbits in the x-y plane to generate the southern family of halo orbits; to a long extent, the orbits have the same characteristics as the northern ones. The direction of the velocity vectors and the departure location change, allowing the opportunity to find different transfer orbits. The same search strategy for the northern halo orbits is repeated here for the southern family. The orbits under study are represented in Figure 8.8, and their characteristics are gathered in Appendix C, Table C.2.

Several simulations are rerun, splitting the target epoch range into three parts of 100 days for each



(a) $\Delta V = 216$ m/s, $T = 176$ days, $\tau = 145$ days.

(b) $\Delta V = 348$ m/s, $T = 190$ days, $\tau = 146$ days.

Figure 8.7: Optimal transfers from northern halo orbits with long transfer times.

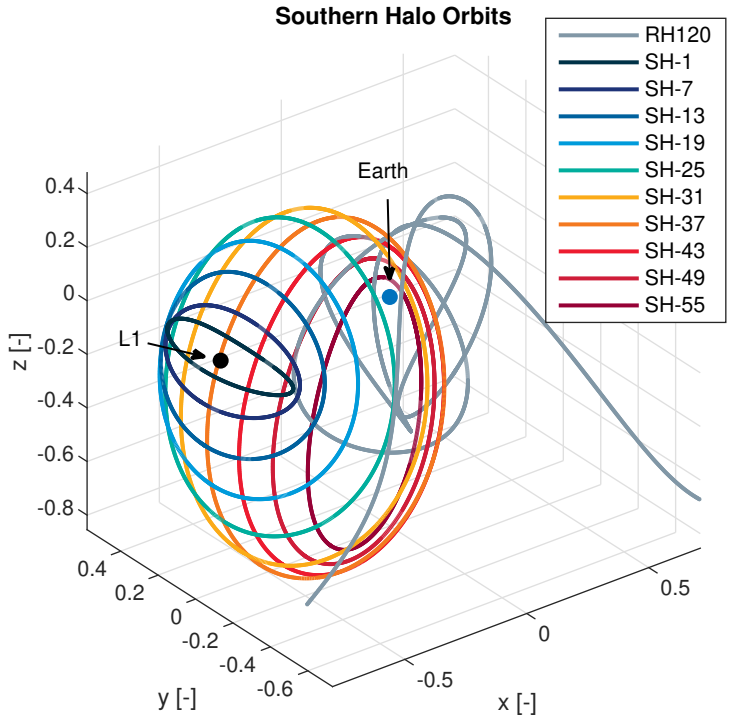


Figure 8.8: Set of southern halo orbits selected for simulations.

orbit, and the transfer with the lowest cost is stored in Table 8.8. Results show that the lowest-cost transfers are not found departing from orbits with the same energy as the TCO, as was the case for the northern halos, but the lowest cost is found for the more giant halos located closer to the Earth. In particular: SH-43 and SH-55. The cost of these transfers is 350 m/s, which is 100 m/s larger than

the lowest-cost transfer found for the northern orbits. Like in the northern halo case, all solutions have converged to a transfer time of close to 100 days, while the range of optimal target epochs has increased, finding solutions at up to 200 days. A number of these transfers are plotted in Figure 8.9.

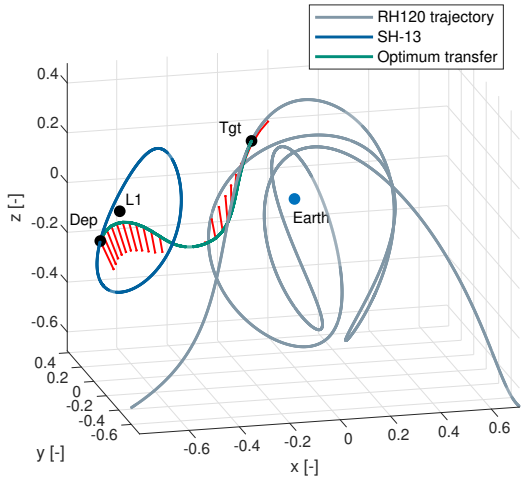
Table 8.8: Simulation results for transfers departing from southern halo orbits with short transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
SH-1	4.0053	727.9499	98.1818	38.3155	-0.8932	4.0909	
SH-7	3.9057	758.7998	99.3856	34.1537	-0.9045	4.1411	
SH-13	3.5081	816.8276	99.4308	24.8341	0.2717	4.1429	Figure 8.9a
SH-19	2.9172	860.4729	99.5441	37.5366	-0.1385	4.1477	
SH-25	2.2193	609.2422	99.3829	93.6257	-0.2643	4.1410	
SH-31	1.5570	594.2568	99.0714	99.0540	-0.1463	4.1280	Figure 8.9b
SH-37	1.1556	478.4649	99.3865	200.9902	0.6999	4.1411	
SH-43	1.0718	341.0866	99.6393	200.4384	0.5586	4.1516	Figure 8.9c
SH-49	1.1800	544.1910	98.9372	168.8252	-0.3080	4.1224	
SH-55	1.3819	354.4766	99.2760	201.7034	-0.4982	4.1365	Figure 8.9d

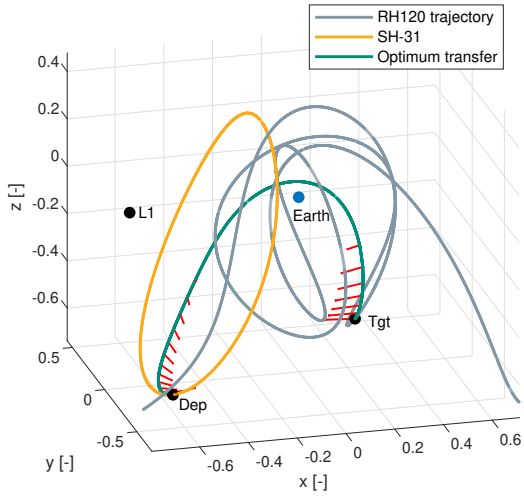
Figure 8.9a shows the transfer that targets the earliest TCO epoch at 24 days after entry into Earth's Hill sphere. The transfer is costly, at 816 m/s, but still feasible for a mission to intercept the asteroid at this epoch. Figure 8.9b shows a transfer from SH-31, which has an energy similar to that of the TCO. The simulation has converged to a target epoch of 99 days at 594 m/s, significantly higher than the cost of transferring from the mirroring northern halo. It is observed that the transfer gets relatively close to Earth, which bends the trajectory significantly before reaching the target. The following two transfers, depicted in Figures 8.9c and 8.9d, correspond to the lowest transfer cost of the family of southern halo orbits and have converged to a similar target epoch at 200 and 201 days respectively. Upon inspection of the transfers, it is observed that they perform a complete revolution around the Earth before reaching the target. Similarly to the lengthy transfer time case studied with the northern halo orbits, these trajectories are not very well suited for the algorithm since they are sensitive to initial conditions and algorithm settings. The transfers, however, show alternatives for more complex transfer options, which can be investigated in future studies.

Similarly to the northern halo case, longer transfer times are studied next since the previous solutions converged close to the upper boundary of the transfer time. Although the resulting transfers in the northern halo case did not necessarily result in lower transfer costs, the study is repeated in case different results are found with the southern family. For such, the maximum transfer time has been increased to 300 days, and the simulations have been run. The results are gathered in Table 8.9, and a comparison between the delta-v cost of short and long transfers is depicted in Figure 8.10.

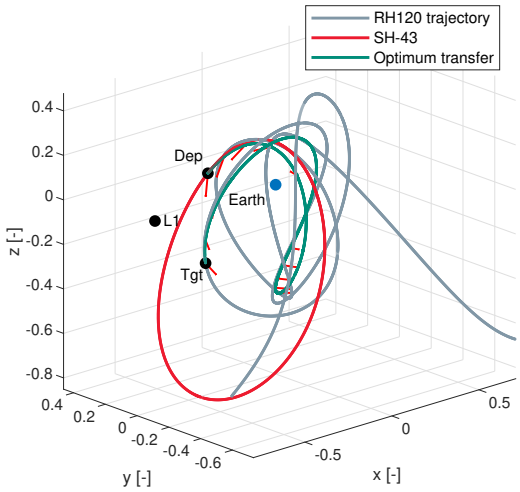
Results show how the overall transfer cost is lower for longer transfers, ranging from 209 to 580 m/s. It is observed in Figure 8.10 that the cost reduction is more significant than for the northern halos (see Figure 8.6). The largest differences are found in orbits up to SH-37. For orbits closer to the Earth, SH-43 to SH-55, the trend is broken, and the cost of longer transfers is comparable to that of short transfers.



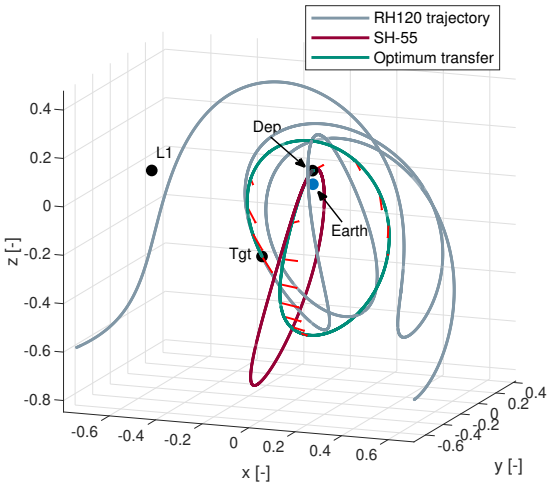
(a) $\Delta V = 816$ m/s, $T = 99$ days, $\tau = 24$ days.



(b) $\Delta V = 594$ m/s, $T = 99$ days, $\tau = 99$ days.



(c) $\Delta V = 341$ m/s, $T = 99$ days, $\tau = 200$ days.



(d) $\Delta V = 354$ m/s, $T = 99$ days, $\tau = 201$ days.

Figure 8.9: Optimal transfers from southern halo orbits with short transfer times.

Solutions for SH-19 and SH-25 are close to the lowest possible transfer cost found up until now. The transfer time that leads to optimum transfers is close to 200 days, reaching up to 245 days for SH-37. The optimal target epoch varies significantly for different transfers. However, it results in almost the same point for the first half of the orbits, from SH-1 until SH-31, at 146 days since the TCO enters the Hill sphere. The simulations that depart from orbits closer to Earth, SH-43 and SH-55, target late segments of the TCO at 289 and 235 days, respectively. Although this target region is not attractive at first for mission design purposes, these solutions can be helpful in case a contingency makes the spacecraft miss the initial departure window and forces a late arrival.

Table 8.9: Simulation results for transfers departing from southern halo orbits with long transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
SH-1	4.0053	580.6190	203.5740	146.3018	-0.7915	8.4822	
SH-7	3.9057	466.9797	204.5499	146.4314	-0.8138	8.5229	
SH-13	3.5081	325.3641	206.6239	146.3916	-0.8441	8.6093	
SH-19	2.9172	209.0804	203.0068	146.8196	-0.8438	8.4586	Figure 8.11a
SH-25	2.2193	242.9883	202.2871	147.0927	-0.8026	8.4286	
SH-31	1.5570	363.8070	198.1095	148.4969	-0.7578	8.2546	
SH-37	1.1556	419.3551	245.9536	110.9705	-0.7356	10.2481	
SH-43	1.0718	345.0870	205.5074	289.9600	-0.6457	8.5628	Figure 8.11b
SH-49	1.1800	524.5705	242.5134	96.9555	-0.4761	10.1047	
SH-55	1.3819	388.6657	240.0086	235.7637	-0.4873	10.0004	

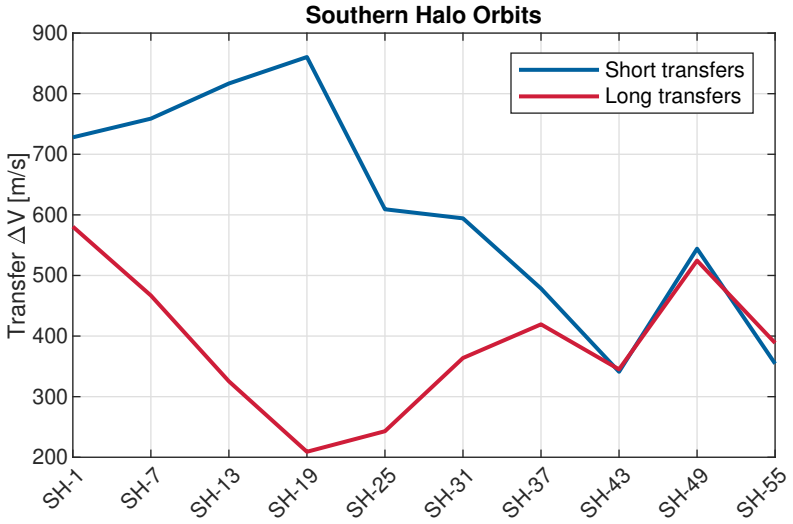
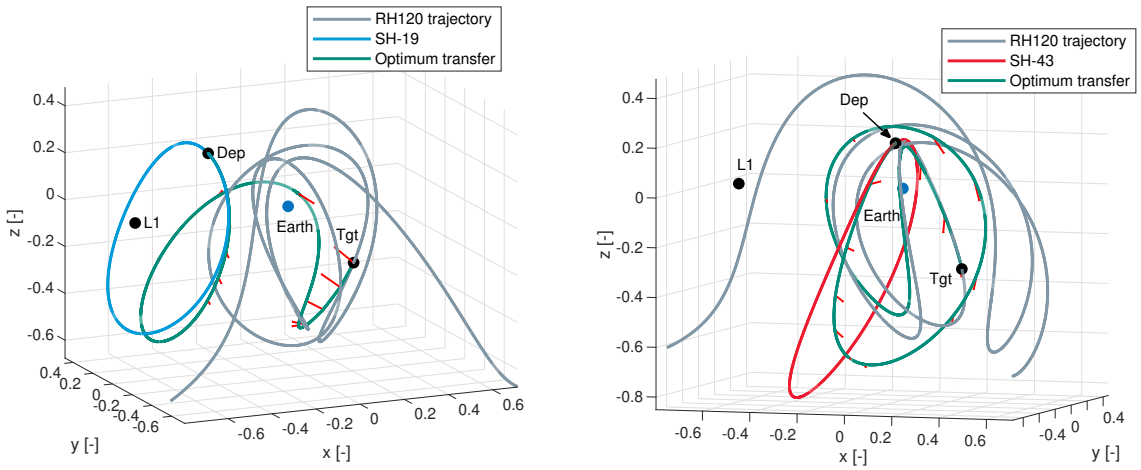


Figure 8.10: Comparison between delta-v cost of short and long transfer times departing from southern halo orbits.

Regarding the mean arc transfer length, it is observed that it is significantly higher than for shorter transfers. This was expected and suggested that the number of arcs should be increased to represent the continuous thrust better. However, the algorithm is sensitive to the number of impulses selected for such long transfer times and converges into different trajectories with significantly different transfer strategies. This was expected, as the algorithm is not optimized for such long transfer times.

Two transfers have been plotted to visualize the shape of the trajectories in Figure 8.11. It can be observed that the trajectories perform a few loops around the Earth before reaching the target. Mainly, Figure 8.11a performs a trajectory that seems to spiral through Earth, crossing it horizontally from the L1 side of Earth until the target is on the other side. The trajectory in Figure 8.11b follows a similar transfer strategy to that of Figure 8.9d, although targeting a later epoch. However, the cost of the maneuver is only ten m/s less costly, which is considered insignificant. This indicates that, for the same transfer strategy, increasing the transfer time does not necessarily decrease the total cost of the transfer.



(a) $\Delta V = 209$ m/s, $T = 203$ days, $\tau = 146$ days.

(b) $\Delta V = 345$ m/s, $T = 205$ days, $\tau = 289$ days.

Figure 8.11: Optimal transfers from southern halo orbits with long transfer times.

8.2.3. Vertical Lyapunov Orbits

The third and last type of orbits considered for the study are the vertical Lyapunov orbits. These orbits are selected since, although not very common operationally, the direction of the velocity vector along the trajectory indicates that there may be good transfers to the TCO.

From the vertical Lyapunov orbit family, a sub-set is selected, comprising ten orbits representing different types of orbits evenly distributed within the family. These are plotted Figure 8.12, and their characteristics are given in Appendix C, Table C.3. It should be noted that the smallest and largest orbits from the family are not studied since the simulations run with them did not converge to feasible solutions.

Similarly to the study with the Halo orbits, a set of simulations is run for different orbits targeting TCO epochs from zero to 300 days since entering the Hill sphere, subdivided into intervals of 100 days for low transfer times below 100 days. The results of the simulations are gathered in Table 8.10. It is observed that transfers from vertical Lyapunov orbits result in a similar cost range to the halo families, ranging from 261 to 954 m/s for the orbits studied, converging at close to 100 days of transfer time. Except for a few cases, it is noticed that most transfers target late epochs above 250 days since Hill entry. It is also noted that the correlation between the lowest transfer cost and the Jacobi constant of the TCO is insignificant, finding the minimum cost for a constant of 1.24, compared to 2.15 for 2006 RH120.

Figure 8.13 displays two transfers from Table 8.10 with different characteristics. Figure 8.13a shows a transfer that targets an early TCO trajectory epoch at 33 days since Hill entry for a transfer cost of 568 m/s. This result represents the lowest cost transfer for such an early target epoch which may be relevant for mission design. A similar transfer solution was obtained for the southern halo orbit with a transfer cost of 200 m/s more costly. At the same time, the transfer is fairly direct from departure,

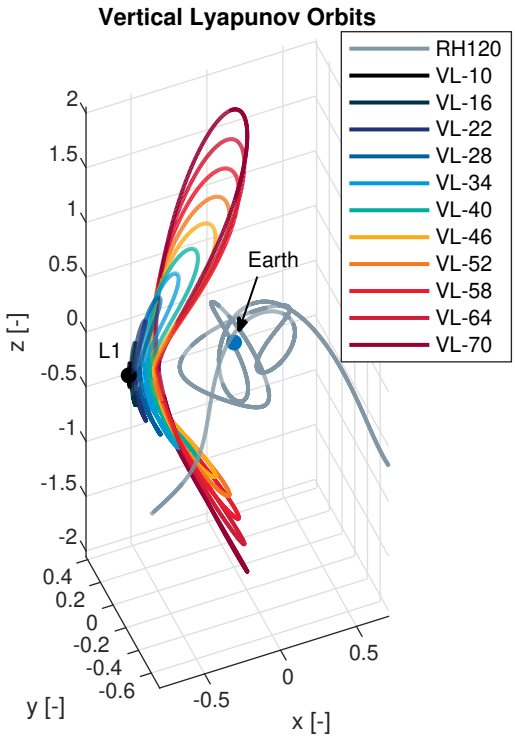
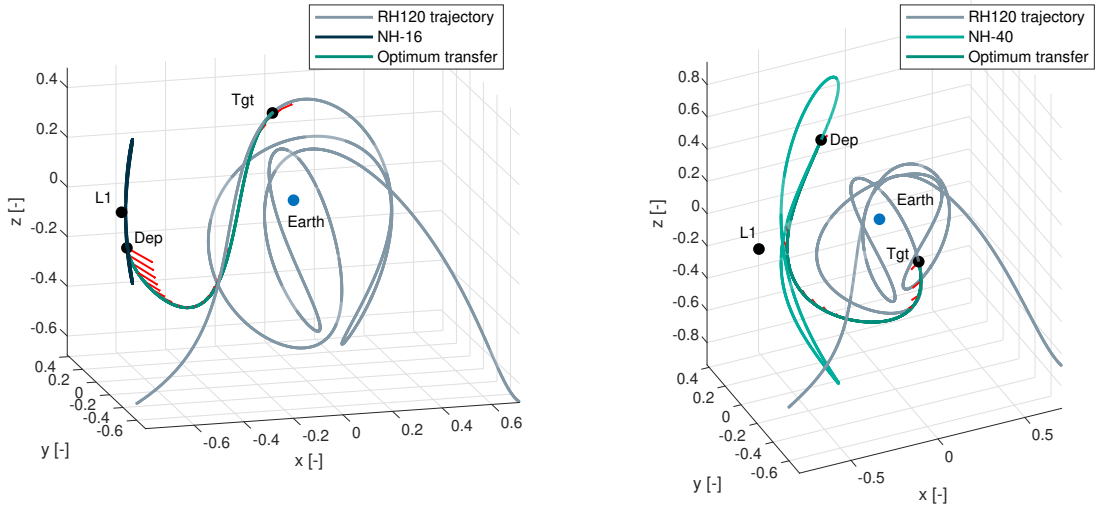


Figure 8.12: Vertical Lyapunov Orbits used for simulations.

Table 8.10: Simulation results for transfers departing from vertical Lyapunov orbits for short transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
VL-10	183.0537	702.2921	99.8571	25.1599	0.0298	4.1607	
VL-16	185.2360	568.9044	99.5597	33.3141	0.0478	4.1483	Figure 8.13a
VL-22	189.7033	510.0306	98.9121	278.6646	-0.0361	4.1213	
VL-28	197.3863	733.3887	99.6611	273.1383	-0.0399	4.1525	
VL-34	209.5737	315.2357	98.7145	254.0525	-0.0840	4.1131	
VL-40	227.1009	261.3732	98.6899	253.7137	-0.0812	4.1121	Figure 8.13b
VL-46	249.0960	279.7345	98.7060	254.1160	-0.0765	4.1127	
VL-52	272.8099	343.6091	98.6393	257.3757	-0.0612	4.1100	
VL-58	294.6740	423.5047	98.6472	258.1600	-0.0552	4.1103	
VL-64	312.1727	954.1650	99.7676	78.7814	-0.2686	4.1570	
VL-70	325.0260	622.9095	98.8145	256.5320	-0.0656	4.1173	

which is more reliable than looping transfers since the model error is smaller. Figure 8.13b shows a different trajectory type that targets an epoch of 250 days. It is more representative of the remaining transfers from the family. The transfer cost is significantly low, close to the lowest found, for a fairly direct transfer. The departure direction sets the spacecraft in a path that aligns with the TCO, allowing such an efficient transfer.



(a) $\Delta V = 568$ m/s, $T = 99$ days, $\tau = 33$ days.

(b) $\Delta V = 261$ m/s, $T = 98$ days, $\tau = 254$ days.

Figure 8.13: Optimal transfers departing from vertical Lyapunov orbits with short transfer times.

The study is repeated for longer transfer times, up to 300 days, to check for lower-cost alternative transfers. The results are gathered in Table 8.11 and compared in Figure 8.14. It is observed in Figure 8.14 that increasing the transfer time significantly reduces the transfer cost for some orbits (VL-16, VL-28, and VL-64), while others have a comparable cost (VL-46, VL-52, VL-58, and VL-70). Table 8.11 shows that an all-time lowest-cost transfer has been found departing from VL-34, costing 190 m/s. Regarding the transfer times, they are, in most cases, above 200 days, and the target epochs are divided into two regions: one at roughly 130 days and another in the region of 260 to 290 days.

Table 8.11: Simulation results for transfers departing from vertical Lyapunov orbits for long transfer times.

Orbit	Γ [-]	ΔV [m/s]	T [days]	τ [days]	ε [-]	Δt_{mean} [days]	Figure
VL-10	183.0537	517.1368	241.9603	131.2710	-0.2123	10.0817	
VL-16	185.2360	346.0850	239.3231	135.1089	-0.1909	9.9718	Figure 8.15a
VL-22	189.7033	641.8813	124.7532	148.0555	-0.2901	5.1980	
VL-28	197.3863	280.8604	211.0734	294.2948	-0.2893	8.7947	
VL-34	209.5737	190.7907	216.7015	292.4988	-0.3039	9.0292	Figure 8.15b
VL-40	227.1009	197.8307	200.9493	261.3999	-0.3380	8.3729	
VL-46	249.0960	273.5348	232.2596	278.3464	-0.3618	9.6775	
VL-52	272.8099	323.7843	225.9700	262.9392	-0.4046	9.4154	
VL-58	294.6740	450.3673	240.1608	264.6954	-0.4178	10.0067	
VL-64	312.1727	525.1014	240.6634	263.2169	-0.3836	10.0276	
VL-70	325.0260	627.2119	243.6004	263.1727	-0.3635	10.1500	

A representative trajectory of each region is plotted in Figure 8.15. Figure 8.15a shows a transfer that targets an epoch of 135 days. Inspecting the transfer, it is observed that the strategy is similar to

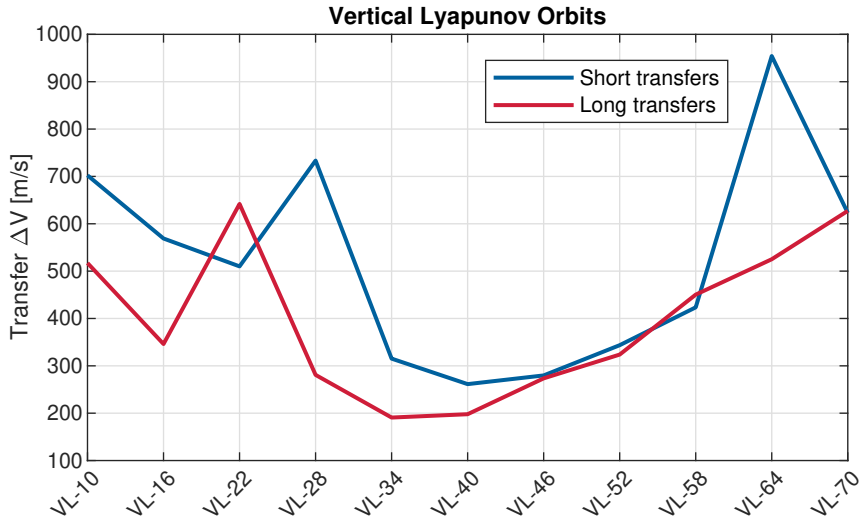
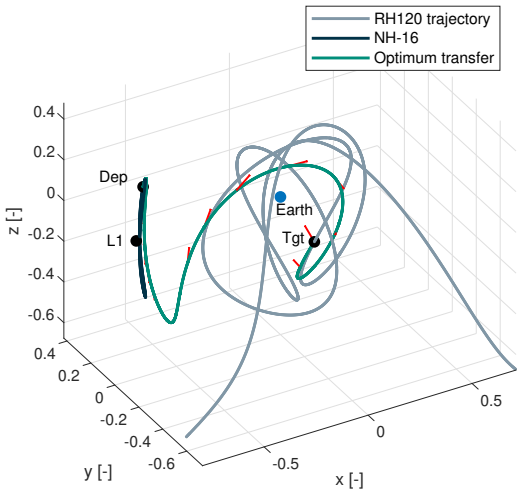
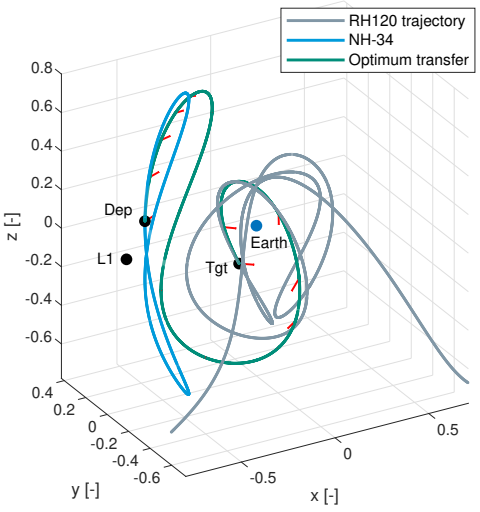


Figure 8.14: Comparison between delta-v cost of short and long transfer times departing from vertical Lyapunov orbits.

that of Figure 8.13a, only allowing the spacecraft to loop further around the Earth to lower the transfer cost by 200 m/s. Figure 8.15b represents the lowest-cost transfer, costing 190 m/s. Upon inspection, the strategy is similar to that of Figure 8.13b, although again, letting the spacecraft loop further around the Earth. The departure orbit sets the spacecraft on a trajectory aligning with the target for a large range until rendezvous 216 days from departure. Although the transfer is less costly, it only reduces the cost with respect to the shorter transfer in Figure 8.13b by 70 m/s. The reduction in propellant mass should be traded against transfer time and rendezvous location in the mission design process.



(a) $\Delta V = 346$ m/s, $T = 239$ days, $\tau = 135$ days.



(b) $\Delta V = 190$ m/s, $T = 216$ days, $\tau = 292$ days.

Figure 8.15: Optimal transfers departing from vertical Lyapunov orbits with long transfer times.

8.2.4. General Remarks on Departure Orbit

Several remarks can be extracted from the departure orbit analysis of the three families of Lagrangian-point orbits studied.

First, regarding the transfer time analysis, it has been observed that most short-transfer trajectories converge to the established upper limit of 100 days for the three orbit families. Letting the algorithm run for longer transfer times than 100 days generally reduces the cost of the transfers. However, this cost reduction is not significant for many of the orbits studied. Generally, longer transfers also converge to late target epochs in the TCO orbit. Late target epochs are less attractive from a mission design perspective since they are close to when the TCO leaves the Earth-Moon system. Another downside of long-transfer trajectories is that they tend to perform more than one revolution around the Earth, which is not the best use case for the developed algorithm. The trajectories obtained with the tool developed for this research are sensitive to initial conditions and algorithm settings. This means that the optimality of long-transfer solutions is less reliable than short-time transfers. However, the results obtained are important in understanding the potential delta-v savings that long transfers offer.

For the analysis of the target epoch and ΔV, Figure 8.16 has been plotted, combining the simulation results of the different orbit families and transfer durations. This chart shows the ΔV cost as a function of the target epoch for the three orbit families studied. The simulations are divided between short transfers, which roughly take 100 days, and long transfers, which take between 180 and 250 days.

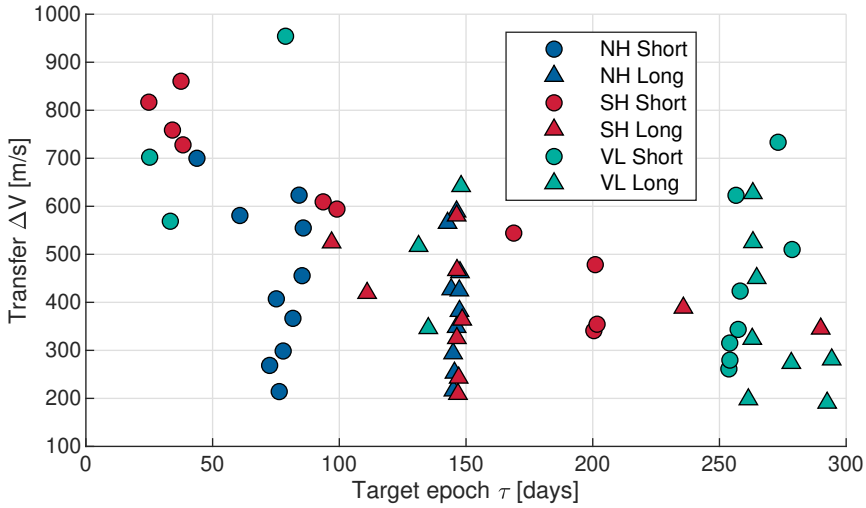


Figure 8.16: ΔV as a function of target epoch for different families of orbits and different transfer times. NH = Norther Halo, SH = Shourthern Halo, and VL = Vertial Lyapunov.

Figure 8.16 it is observed that several optimum transfers have been found spanning over an extensive range of target epochs. Among these, very low-cost transfers of 200 m/s can be obtained targeting three regions of the TCO: 75 days (early target), 150 days (mid-range target), and 250-300 days (late target). Northern halo orbits offer low-cost short-time transfers for the early targets while also offering low-cost longer-time transfers in the target mid-range. For the late targets, the best results are provided by vertical Lyapunov orbits. They also provide low-cost 200 m/s transfers with long transfer times while also allowing for shorter transfers of 250-300 m/s. The best performance of the southern orbits is found in long transfers in the target mid-range. However, it only matches that of the northern halo orbits while

performing significantly worse at the short transfers.

Regarding the energy of the departure orbit, measured through the Jacobi constant Γ, it is a general trend that those closer to the energy of the TCO result in less costly transfers. This can be observed in Figure 8.17, where the lowest cost transfers depart from orbits with the Jacobi constant closer to that of the TCO, and the cost increases as they get further apart. It should be noted, though, that another significant factor influencing the transfer cost is the direction of the velocity vector at departure. Lower-cost transfers were generally found from favorable departure directions that align with the target epoch velocity.

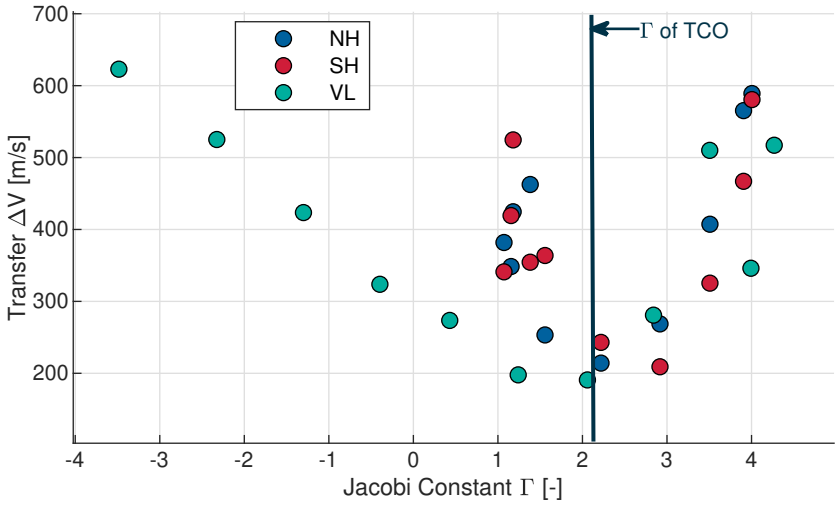


Figure 8.17: Transfer cost as a function of Jacobi constant Γ.

9

Conclusions and Future Work

This final chapter gathers the conclusions of the present research and provides recommendations about potential improvements and extensions of the work.

9.1. Conclusions

This report has dealt with the preliminary design of low-thrust transfer trajectories from periodic orbits around the Sun-Earth L1 point to answer the main research question:

How can transfers originating near the Sun-Earth L1 be leveraged to target Temporarily Captured Orbiters such as 2006 RH120 in a spacecraft configuration that uses electric propulsion?

The research question is answered by dividing it into more specific sub-questions

- Sub-question 1: How can the optimization problem be formulated to conduct a preliminary analysis on transfers departing from orbits around L1?

The optimization problem has been formulated in such a way that the partial derivatives of the cost and constraint functions were derived analytically. This allowed for the algorithm to converge into low-thrust optimization transfers for a fraction of the computation cost of standard optimization methods, where the partial derivatives are computed numerically. These analytical solutions have been possible thanks to reducing the dynamics model to the Hill three-body problem and the continuous thrust model to a discrete set of delta-v impulses.

The algorithm has been designed by progressively increasing the complexity and capabilities of the method. It has resulted in a robust and fast tool that consistently optimizes transfers from different departure orbits near the Sun-Earth L1 point. Some of the optimization variables included in the problem are the departure location within a pre-selected orbit, the rendezvous location within the TCO trajectory inside Earth's Hill sphere, the magnitude of the impulses, and the duration of the transfer.

- Sub-question 2: How do design parameters such as departure orbit or rendezvous location affect the delta-v cost of the transfers?

The developed tool has been used to analyze three families of periodic orbits and develop recommendations for mission designers on the performance of different orbits, transfer times, and rendezvous locations. Most orbits analyzed resulted in feasible optimal transfers to 2006 RH120, with total transfer delta-v's ranging from 190 to 900 m/s. Northern Halo orbits have resulted in the most promising departure family of orbits, with a preference for orbits with the same Jacobi constant as the target TCO. These orbits allow targeting early epochs of the TCO trajectory within the Earth's Hill sphere, in a short transfer time of 100 days, with a low delta-v cost of 200 m/s. In general, southern halo orbits resulted in worse departure conditions, while vertical Lyapunov orbits were an excellent candidate for short-time transfers to later epochs in the TCO trajectory.

The most reliable optimum solutions were obtained for short-time transfers (100 days) that perform less than a revolution around the Earth. However, some solutions have been found with low delta-v costs that perform more than one revolution around the Earth. The developed algorithm has not been designed to be robust to these types of transfers, and a tool specifically designed to study fly-bys may be needed to survey these types of transfers.

The vicinity of the Sun-Earth L1 point has been proven to be a favorable environment for a lightweight spacecraft to hibernate until a TCO is detected. Its proximity to the entry point of TCOs into the Earth-Moon system allows targeting segments of the TCO orbit at the beginning of the excursion, which can be favorable for maximizing the observation time of the TCO before leaving the system. At the same time, optimal transfers have been found for intermediate and late epochs in the trajectory of the TCO, offering a wide departure window versatile for mission-design purposes.

9.2. Recommendations for Future Work

The present research has many potential implementations that did not fit in the time allotted for the master thesis and are recommended for future work. Subsequently, there is a list of some implementations that could serve as the subject of study of future thesis works or research projects.

- The first recommendation for future work is to extend the study to other TCOs, such as the recently discovered 2020 CD3 or the virtual TCOs generated by Granvik et al., 2012. This extension of the work was originally intended to be included in the present research, but the project's time constraints did not allow it.
- Another potential extension of the present work is to modify the algorithm to accommodate fly-bys around the Earth. The single-shooting strategy used in this thesis has resulted in a robust and reliable tool for relatively short direct transfers of up to 100 days. The algorithm has also found a set of longer transfers that perform revolutions around the Earth with low delta-v costs. However, these solutions were very sensitive to initialization settings and could benefit from a dedicated tool. An excellent first candidate for this implementation that could still leverage analytical gradients is the Sims-Flanagan transcription (Sims & Flanagan, 1999). It consists of propagating the spacecraft forward from departure and backward from the target and forcing state continu-

ity at the midpoint. Other schemes that may be suitable for longer revolving trajectories are the multiple-shooting methods. They extend the single-shooting scheme to include multiple intermediate continuity points that can force the solution to perform a specified number of revolutions around the Earth.

- Another reasonable extension of the work is to include a more complex representation of the dynamics model that represents the TCO environment more accurately. To keep the complexity of the model at a level that permits the computation of the gradients analytically, the circular-restricted four-body model could be explored. This model would allow including the effect of the Moon in the simulations, which has been identified as crucial to replicate the capture conditions of the TCO by Urrutxua et al., 2015. As such, the solar radiation pressure could also be included in the model, which introduces an acceleration in the direction opposite to the Sun (fixed in the synodic frame), which may have implications on the shape and cost of the computed transfers.
- A third extension to the present work is to use a more accurate thrust model. The controls of the thrust engine could be added and introduce the effect of the thruster as an acceleration in the dynamical model instead of delta-v impulses. This will allow expression of the optimization objective as minimization of the propellant consumed and integration of the spacecraft's mass over the transfer trajectory.
- A final recommendation on a potential extension of this work is to include mission-related considerations regarding the departure orbit selected. For example, the cost of reaching each specific hibernation orbit could be addressed, as well as performing a sensitivity analysis on the effect of contingencies, such as reduced thrust or the cost of late departure when a nominal transfer has been selected.

A

Differential Correction

The method solves a Two-Point Boundary Value Problem (TPBVP) by reducing it to an Initial Value Problem (IVP). The TPBVP to be solved is the following:

Problem A.0.1 *Find a solution to the differential equation:*

$$\ddot{\mathbf{r}} = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, \mathbf{u}) \tag{A.1}$$

such that the following boundary conditions are met:

$$\mathbf{r}(t = t_0) = \mathbf{r}_{dep} \tag{A.2}$$

$$\mathbf{r}(t = t_f) = \mathbf{r}_{arr} \tag{A.3}$$

The differential equation in Equation A.1 describes the motion of the spacecraft in the dynamical environment and therefore has a different shape for different problems. The problem is reduced to the following IVP:

Problem A.0.2 *Find a solution of the differential Equation A.1 which has the following initial conditions*

$$\mathbf{r}(t = t_0) = \mathbf{r}_{dep} \tag{A.4}$$

$$\dot{\mathbf{r}}(t = t_0) = \mathbf{v}_{dep} \tag{A.5}$$

By fixing the departure location, the initial velocity vector $\mathbf{v}_{\text{dep}}$ can be modified such that the solution of the differential equation at $t = t_f$ meets the boundary condition in Equation A.3.

To adjust the departure velocity, the differential correction method is introduced. The method applied to the case of trajectory generation is subsequently explained (Heiligers, 2020).

The position of a spacecraft at time t is a function of an initial position $\mathbf{r}_0$ and velocity $\mathbf{v}_0$ at $t = t_0$ such as

$$\mathbf{r} = \mathbf{r}(t, \mathbf{r}_0, \mathbf{v}_0) \tag{A.6}$$

The final arrival position of the spacecraft after propagation time T, from a departure point $\mathbf{r}_{\text{dep}}$ and estimated initial velocity $\mathbf{v}^*_{\text{dep}}$ is

$$\mathbf{r}^*_{\text{arr}} = \mathbf{r}(T, \mathbf{r}_{\text{dep}}, \mathbf{v}^*_{\text{dep}}) \tag{A.7}$$

Note that the superscript $[\,]^*$ is introduced to denote that the value is estimated. Such initial velocity estimate will most likely not reach the desired target position $\mathbf{r}_{\text{tgt}}$, so

$$\mathbf{r}^*_{\text{arr}} \neq \mathbf{r}_{\text{tgt}} \tag{A.8}$$

At this point, we introduce a small correction $\delta\mathbf{v}$ in the initial velocity vector such that

$$\delta\mathbf{v} = \mathbf{v}_0 - \mathbf{v}^*_{\text{dep}} \tag{A.9}$$

and linearize the rendezvous position of the spacecraft with respect to the initial velocity estimate $\mathbf{v}^*_{\text{dep}}$ expanding in a Taylor series such as

$$\mathbf{r}(T, \mathbf{r}_{\text{dep}}, \mathbf{v}_0) = \mathbf{r}_{\text{arr}} = \mathbf{r}(T, \mathbf{r}_{\text{dep}}, \mathbf{v}^*_{\text{dep}}) + \left.\frac{\partial \mathbf{r}(T, \mathbf{r}_{\text{dep}}, \mathbf{v}_0)}{\partial \mathbf{v}_0}\right|_{\mathbf{v}^*_{\text{dep}}} (\mathbf{v}_0 - \mathbf{v}^*_{\text{dep}}) \tag{A.10}$$

Imposing that $\mathbf{r}_{\text{arr}} = \mathbf{r}_{\text{tgt}}$, and using definitions from equations A.7 to A.9, the previous equation can be rewritten as

$$\mathbf{r}_{\text{tgt}} = \mathbf{r}^*_{\text{arr}} + \left.\frac{\partial \mathbf{r}(T, \mathbf{r}_{\text{dep}}, \mathbf{v}_0)}{\partial \mathbf{v}_0}\right|_{\mathbf{v}^*_{\text{dep}}} \delta\mathbf{v} \tag{A.11}$$

Recalling the concept of the state transition matrix in Equation 4.5, Equation A.11 can be rewritten as

$$\mathbf{r}_{\text{tgt}} = \mathbf{r}^*_{\text{arr}} + \Phi_{\mathbf{r},\mathbf{v}_0}\delta\mathbf{v} \tag{A.12}$$

so the initial velocity correction becomes

$$\delta\mathbf{v} = [\Phi_{\mathbf{r},\mathbf{v}_0}(T,\, t_0)]^{-1}(\mathbf{r}_{\text{tgt}} - \mathbf{r}^*_{\text{arr}}) \tag{A.13}$$

B

Verification Tables

Table B.1: Verification results for dynamics model checking the Jacobi constant in different epochs of a propagated trajectory.

Epoch [days]	**Jacobi Constant** Γ [-]
0	2.15201897650130
42.59	2.15201897650127
172.23	2.15201897650137
264.38	2.15201897650104

Table B.2: Verification results for STM in non-dimensional units. Deviation of 10^{-11} and transfer time of 24 days.

	IC's	$\Phi\,\delta\mathbf{s}_0$	$\delta\mathbf{s}_f$	Error
x	-0.693361274350635	2.34271475098954e-11	2.34267050203130e-11	4.42489582452000e-16
y	0	4.34629721583274e-12	4.34607905219764e-12	2.18163635097306e-16
z	0	1.36628577853453e-11	1.36622380075835e-11	6.19777761778273e-16
$\dot{x}$	0.441547893556523	4.85249280438305e-11	4.85238516034769e-11	1.07644035353541e-15
$\dot{y}$	0.984290857994792	-4.88303798637160e-11	-4.88303147916369e-11	6.50720791085340e-17
$\dot{z}$	-1.33927778874451	1.86531950307369e-11	1.86510806798879e-11	2.11435084901443e-15

Table B.3: Verification results for STM using finite differences in non-dimensional units. Same initial conditions as in Table B.2 for a propagation duration of 24 hours.

Φ	1.0013	-1.6161e-05	1.5064e-05	0.0172	2.9586e-04	1.3011e-07
	-2.6387e-05	0.9996	-3.1412e-07	-2.9605e-04	0.0172	-2.8078e-09
	1.5072e-05	-1.8510e-07	0.9994	1.3016e-07	-5.6918e-10	0.0172
	0.1564	-0.0028	0.0026	1.0008	0.0344	3.0436e-05
	-0.0046	-0.0524	-7.3442e-05	-0.0344	0.9990	-8.2037e-07
	0.0026	-4.3325e-05	-0.0696	3.0451e-05	-1.6655e-07	0.9994
Φ_{FD}	1.0013	-1.1102e-05	2.2204e-05	0.0172	2.9976e-04	0
	-2.6021e-05	0.9996	-3.4694e-07	-2.9594e-04	0.0172	0
	1.5613e-05	0	0.9994	1.0408e-06	0	0.0172
	0.1564	-0.0028	0.0026	1.0008	0.0344	2.7756e-05
	-0.0046	-0.0524	-7.7716e-05	-0.0344	0.9990	0
	0.0027	0	-0.0695	8.8818e-05	0	0.9994
Error	-2.3996e-06	-5.0584e-06	-7.1409e-06	-1.2289e-06	-3.9003e-06	1.3011e-07
	-3.6590e-07	-2.5116e-07	3.2827e-08	-1.0608e-07	-8.1484e-08	-2.8078e-09
	-5.4094e-07	-1.8510e-07	-7.6483e-07	-9.1068e-07	-5.6918e-10	-6.5688e-07
	8.5416e-06	-5.4987e-06	-3.3895e-06	3.0583e-06	-2.0400e-06	2.6801e-06
	1.4939e-06	-5.2662e-06	4.2740e-06	-3.4735e-06	-1.2595e-05	-8.2037e-07
	-3.9899e-05	-4.3325e-05	-5.3892e-05	-5.8366e-05	-1.6655e-07	-2.3411e-05

Table B.4: Verification results for *emo2synodic*. Benchmark data from Takahashi et al., 2022.

	EMO State	Benchmark Synodic	Computed Synodic	Error
x [km]	703844.715	738504.112	738515.011	10.8991828
y [km]	309110.719	213454.088	213406.649	47.4382781
z [km]	-955888.743	-955911.279	-955888.743	22.5364624
$\dot{x}$ [m/s]	-223.510136	-134.256874	-134.262197	5.32306984e-3
$\dot{y}$ [m/s]	339.059284	218.600285	218.573691	2.65940027e-2
$\dot{z}$ [m/s]	-469.134097	-469.106770	-469.134097	2.73271540e-2

Table B.5: Verification results for *jacobiConstant* in non-dimensional units. Benchmark data from Kalantonis, 2020.

	Orbit 1	Orbit 2	Orbit 3
x	0.58126467	0.09738337	1.28678525
y	0	0	0
z	0	0	-0.19596176
$\dot{x}$	0	0	0
$\dot{y}$	0.67012429	4.35161692	-2.59789645
$\dot{z}$	0	1.3	
$\Gamma_{benchmark}$	4.00531265	-0.06073137	-0.28347259
$\Gamma_{computed}$	4.00531266	-0.06073182	-0.28347258

Table B.6: Verification results for *ssJ2JD*. Benchmark obtained from Tudat software.

seconds since J2000	205477435.0
Julian Date Benchamrk	2453923.2110532406
Julian Date Computed	2453923.211053241

C

Periodic Orbits

Table C.1: Northern Halo orbits characteristics used for transfer simulations.

Orbit	Period [days]	Γ [-]	x_0 [-]	z_0 [-]	v_0 [-]
NH-1	179.1233	4.0053	-0.7747	0.0031	0.6138
NH-7	178.9488	3.9057	-0.7722	0.1285	0.6492
NH-13	178.1097	3.5081	-0.7582	0.2951	0.7666
NH-19	176.2505	2.9172	-0.7281	0.4529	0.8946
NH-25	172.2162	2.2193	-0.6798	0.6052	0.9991
NH-31	163.4529	1.5570	-0.6123	0.7381	1.0529
NH-37	148.2177	1.1556	-0.5281	0.8204	1.0285
NH-43	130.2385	1.0718	-0.4360	0.8507	0.9311
NH-49	113.2587	1.1800	-0.3402	0.8544	0.7821
NH-55	98.7975	1.3819	-0.2397	0.8473	0.5864

Table C.2: Southern Halo orbits characteristics used for transfer simulations.

Orbit	Period [days]	Γ [-]	x_0 [-]	z_0 [-]	v_0 [-]
SH-1	179.1233	4.0053	-0.7747	-0.0031	0.6138
SH-7	178.9488	3.9057	-0.7722	-0.1285	0.6492
SH-13	178.1097	3.5081	-0.7582	-0.2951	0.7666
SH-19	176.2505	2.9172	-0.7281	-0.4529	0.8946
SH-25	172.2162	2.2193	-0.6798	-0.6052	0.9991
SH-31	163.4529	1.5570	-0.6123	-0.7381	1.0529
SH-37	148.2177	1.1556	-0.5281	-0.8204	1.0285
SH-43	130.2385	1.0718	-0.4360	-0.8507	0.9311
SH-49	113.2587	1.1800	-0.3402	-0.8544	0.7821
SH-55	98.7975	1.3819	-0.2397	-0.8473	0.5864

Table C.3: Vertical Lyapunov orbit characteristics used for transfer simulations.

Orbit	Period [days]	Γ [-]	x_0 [-]	v_0 [-]	w_0 [-]
VL-10	183.0537	4.2685	-0.6905	0.0058	-0.2415
VL-16	185.2360	3.9930	-0.6768	0.0346	-0.5788
VL-22	189.7033	3.5058	-0.6528	0.0904	-0.9101
VL-28	197.3863	2.8405	-0.6209	0.1786	-1.2269
VL-34	209.5737	2.0588	-0.5866	0.3022	-1.5139
VL-40	227.1009	1.2400	-0.5575	0.4519	-1.7537
VL-46	249.0960	0.4301	-0.5387	0.6000	-1.9476
VL-52	272.8099	-0.3962	-0.5283	0.7153	-2.1231
VL-58	294.6740	-1.3004	-0.5205	0.7821	-2.3117
VL-64	312.1727	-2.3254	-0.5110	0.8062	-2.5244
VL-70	325.0260	-3.4840	-0.4995	0.8038	-2.7551

Bibliography

Aschbacher, J. (March 31, 2021). *ESA Agenda 2025*. ESA unclassified. European Space Agency. URL: esa.int/About_Us/ESA_Publications/Agenda_2025 (visited on June 5, 2023).

Bolin, B. T. et al. (September 2020). "Characterization of Temporarily Captured Minimoon 2020 CD3 by Keck Time-resolved Spectrophotometry". In: *The Astrophysical Journal Letters* 900.2, p. L45.

Brelsford, S., M. Chyba, T. Haberkorn, and G. Patterson (April 1, 2016). "Rendezvous Missions to Temporarily Captured near Earth Asteroids". In: *Planetary and Space Science*. SI:ACM Interrelated 123, pp. 4–15.

Chikine, S. (2021). "Low-Thrust Trajectory Optimization in the Circular Restricted Three Body Problem". Master's Thesis. University of Colorado Boulder.

Chyba, M., M. Granvik, R. Jedicke, G. Patterson, G. Picot, and J. Vaubaillon (2014). "Time-Minimal Orbital Transfers to Temporarily-Captured Natural Earth Satellites". In: *Optimization and Control Techniques and Applications*. Ed. by H. Xu, K. L. Teo, and Y. Zhang. Springer Proceedings in Mathematics & Statistics. Berlin, Heidelberg: Springer, pp. 213–235.

Chyba, M., T. Haberkorn, and R. Jedicke (2016). "Minimum Fuel Round Trip from a L2 Earth-Moon Halo Orbit to Asteroid 2006 RH120". In: *Recent Advances in Celestial and Space Mechanics*. Ed. by B. Bonnard and M. Chyba. Mathematics for Industry. Cham: Springer International Publishing, pp. 117–142.

Chyba, M. and G. Patterson (2019). "Indirect Methods for Fuel-Minimal Rendezvous with a Large Population of Temporarily Captured Orbiters". In: *Numerical Algebra, Control and Optimization* 9.2, p. 225.

Conway, B. A. (2010). *Spacecraft Trajectory Optimization*. New York, U.S.A.: Cambridge University Press.

Curtis, H. (2020). *Orbital Mechanics for Engineering Students*. 4th ed. Butterworth Heinemann. 780 pp.

De la Fuente Marcos, C. and R. de la Fuente Marcos (August 10, 2022). "How to Become a Mini-moon: Some Hints from 2022 NX1". In: *Research Notes of the AAS* 6.8, p. 160.

Dirkx, D. (January 24, 2022). *Tudat Mathematical Model Definition*. URL: https://docs.tudat.space/ (visited on April 7, 2023).

ECSS-E-ST-10-09C (July 31, 2008). *Reference Coordinate System*. Standard. European Cooperation for Space Standardization. URL: ecss.nl.

Fedorets, G., M. Micheli, R. Jedicke, S. Naidu, D. Farnocchia, M. Granvik, N. Moskovitz, M. Schwamb, R. Weryk, K. Wierzchoś, E. Christensen, T. Pruyne, W. Bottke, Q. Ye, R. Wainscoat, M. Devog le, L. Buchanan, A. Djupvik, D. Faes, D. Föhring, J. Roediger, T. Seccull, and A. Smith (2020a). "Establishing Earth's Minimoon Population through Characterization of Asteroid 2020 CD3". In: *Astronomical Journal* 160.6.

Fedorets, G., M. Granvik, R. L. Jones, M. Jurić, and R. Jedicke (March 1, 2020b). "Discovering Earth's Transient Moons with the Large Synoptic Survey Telescope". In: *Icarus* 338, p. 113517.

Granvik, M., R. Jedicke, B. Bolin, M. Chyba, G. Patterson, and G. Picot (2013). “Earth’s Temporarily-Captured Natural Satellites – The First Step towards Utilization of Asteroid Resources”. In: *Asteroids: Prospective Energy and Material Resources*. Ed. by V. Badescu. Berlin, Heidelberg: Springer, pp. 151–167.

Granvik, M., J. Vaubaillon, and R. Jedicke (March 1, 2012). “The Population of Natural Earth Satellites”. In: *Icarus* 218.1, pp. 262–277.

Heiligers, J. (2020). *Periodic Orbits Around the Colinear Lagrange Points*. Lecture Slides AE4889: Special Topics in Astrodynamics. Delft University of Technology.

Herman, J. (March 29, 2012). “Preliminary Mission Analysis for a Far-Side, Highly Inclined Solar Observatory Using Electric Propulsion”. MSc Thesis. Delft Univeristy of Technology. URL: `repository.tudelft.nl`.

Jedicke, R., B. Bolin, W. Bottke, M. Chyba, G. Fedorets, M. Granvik, and G. Patterson (2016). “Small Asteroids Temporarily Captured in the Earth-Moon System”. In: *Proceedings of the International Astronomical Union* 10, pp. 86–90.

Jedicke, R., B. Bolin, W. Bottke, M. Chyba, G. Fedorets, M. Granvik, L. Jones, and H. Urrutxua (2018). “Earth’s Minimoons: Opportunities for Science and Technology”. In: *Frontiers in Astronomy and Space Sciences* 5.13, pp. 6–24.

Kalantonis, V. S. (June 2020). “Numerical Investigation for Periodic Orbits in the Hill Three-Body Problem”. In: *Universe* 6.6, p. 72.

Kaplan, G. H. (2005). *The IAU Resolutions on Astronomical Reference Systems, Time Scales, and Earth Rotation Models (Naval Observatory Circular No. 179)*. Washington: USNO.

Kwiatkowski, T., A. Kryszczyńska, M. Polińska, D. A. H. Buckley, D. O’Donoghue, P. A. Charles, L. Crause, S. Crawford, Y. Hashimoto, A. Kniazev, N. Loaring, E. Romero Colmenero, R. Sefako, M. Still, and P. Vaisanen (March 2009). “Photometry of 2006 RH $_{120}$: An Asteroid Temporary Captured into a Geocentric Orbit”. In: *Astronomy & Astrophysics* 495.3, pp. 967–974.

Lewis, B. J., E. N. Onder, and A. A. Prudil (January 1, 2022). “Chapter 9 - Numerical Analysis”. In: *Advanced Mathematics for Engineering Students*. Ed. by B. J. Lewis, E. N. Onder, and A. A. Prudil. Butterworth-Heinemann, pp. 265–285.

Lissauer, J. J. and I. de Pater (2019). *Fundamental Planetary Science: Physics, Chemistry and Habitability*. Cambridge: Cambridge University Press.

Morante, D., M. S. Rivo, and M. Soler (2021). “A Survey on Low-Thrust Trajectory Optimization Approaches”. In: *Aerospace* 8.88.

Ozimek, M. T. and K. C. Howell (March 2010). “Low-Thrust Transfers in the Earth-Moon System, Including Applications to Libration Point Orbits”. In: *Journal of Guidance, Control, and Dynamics* 33.2, pp. 533–549.

Parker, J. S. and R. L. Anderson (2014). *Low-Energy Lunar Trajectory Design*. Deep-Space Communications and Navigation Series. Hoboken, New Jersey, U.S.A.: Wiley.

Petit, G. and B. Luzum (2010). *IERS Conventions (Technical Note; No. 36)*. Frankfurt am Main: Verlag des Bundesamts fur Kartographie und Geodasie, p. 179.

Sanchez Ruiz, L. M. and M. P. Lengua Fernandez (2008). *Calculo Matematico con Applicaciones*. 5th ed. Editorial Universitat Politecnica de Valencia. 450 pp.

Shirazi, A., J. Ceberio, and J. A. Lozano (2018). “Spacecraft Trajectory Optimization: A Review of Models, Objectives, Approaches and Solutions”. In: *Progress in Aerospace Sciences* 102, pp. 76–98.

Sims, J. and S. Flanagan (August 1999). “Preliminary Design of Low-Thrust Interplanetary Missions”. In: AAS/AIAA Astrodynamics Specialist Conference. Vol. AAS 99-338. Girdwood, Alaska, U.S.A.

Sutton, G. P. and O. Biblarz (2017). *Rocket Propulsion Elements*. 9th ed. John Wiley & Sons Inc.

Szebehely, V. (January 1, 1967). *Theory of Orbit*. 1st ed. Academic Press. 684 pp.

Takahashi, S., D. Crull, L. Whittle, and D. J. Scheeres (August 10, 2022). “Autonomous Rendezvous with Small Temporarily Captured Orbiters”. Presentation AAS 22-700. 2022 AAS/AIAA Astrodynamics Specialist Conference (Charlotte, NC, U.S.A.).

The MathWorks Inc. (2023a). *Choose an ODE Solver*. MATLAB Documentation. URL: `mathworks.com/help/matlab/math/choose-an-ode-solver.html` (visited on April 7, 2023).

The MathWorks Inc. (2023b). *Interp1 Function Documentation*. MATLAB Documentation. URL: `mathworks.com/help/matlab/ref/interp1.html#btwp6lt-1-method` (visited on April 18, 2023).

Trofimov, S. P., A. A. Tselousova, and M. G. Shirobokov (November 1, 2018). “Two Direct Low Thrust Trajectory Optimization Techniques”. In: *Journal of Computer and Systems Sciences International* 57.6, pp. 989–1000.

Tudat Space Documentation (2022). *Available State Definitions and Conversions*. tudat.space 0.3.1 documentation. URL: `https://docs.tudat.space` (visited on March 2, 2023).

Urrutxua, H., D. J. Scheeres, C. Bombardelli, J. L. Gonzalo, and J. Peláez (November 2015). “Temporarily Captured Asteroids as a Pathway to Affordable Asteroid Retrieval Missions”. In: *Journal of Guidance, Control, and Dynamics* 38.11, pp. 2132–2145.

Villac, B. F. and D. J. Scheeres (March 2003). “Escaping Trajectories in the Hill Three-Body Problem and Applications”. In: *Journal of Guidance, Control, and Dynamics* 26.2, pp. 224–232.

Wakker, K. (2015). *Fundamentals of Astrodynamics*. Institutional Repository. TU Delft Library. URL: `repository.tudelft.nl`.

Zhao, S. (September 20, 2016). “Time Derivative of Rotation Matrices: A Tutorial”. In: *arXiv* (arXiv:1609.06088). URL: `http://arxiv.org/abs/1609.06088` (visited on March 2, 2023).