Mapped Adjoint Control Transformation for Low-Thrust Space Mission Design

by

Praveen Jawaharlal Ayyanathan

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 5, 2023

Keywords: Adjoint Control Transformation, Optimal Control, Trajectory Optimization

Copyright 2023 by Praveen Jawaharlal Ayyanathan

Approved by

Ehsan Taheri, Chair, Assistant Professor of Aerospace Engineering
Davide Guzzetti, Assistant Professor of Aerospace Engineering
Nan Li, Assistant Professor of Aerospace Engineering

Abstract

Spacecraft trajectory optimization is an essential task in space mission design. The propulsion system of the spacecraft can affect the type of trajectories that can be realized by a spacecraft. In the past few decades, electric propulsion systems (with their characteristic high specific impulse values, but low thrust magnitudes) have revolutionized space trajectories. Low-thrust trajectory design can be converted into boundary-value problems, which are typically challenging to solve because of a small domain of convergence and lack of knowledge about the initial costates when indirect formalism of optimal control is adopted. Estimating missing values of the non-intuitive costates is an important step in solving the resulting boundary-value problems. In this thesis, the initial costates are obtained using two methods: 1) random initialization, and 2) when costate initial values are constrained to lie on a unit 8-dimensional hypersphere. Minimum-fuel trajectories are designed for a heliocentric maneuver from Earth to comet 67P/Churyumov–Gerasimenko. The two costate initialization methods are compared against each other in terms of the percent of convergence and accuracy of the results of the associated boundary-value problems. After this analysis, the Adjoint Control Transformation costate initialization method is considered. By leveraging *costate vector mapping* theorem, the method of Adjoint Control Transformation (ACT) is extended to alternative sets of coordinates/elements for solving low-thrust trajectory optimization problems, called Mapped Adjoint Control Transformation (MACT) . The development of MACT is the main contribution of this thesis. In particular, this extension is applied to the set of modified equinoctial elements and an orbital element set based on the specific angular momentum and eccentricity vectors (h-e). The computational and robustness efficiency of the MACT method is compared against the traditionally used random initialization of costates by solving 1) interplanetary rendezvous maneuvers, 2) an Earth-centered, orbit-raising problem with and without the inclusion of J_2 perturbation, and 3) an Earth-centered, orbit-raising problem with a relatively large number of revolutions around the central body. For the considered problems, numerical results indicate two to three times improvement in the percent of convergence of the resulting boundary-value problems when the MACT method is used compared to the random initialization method. Results also indicate that the h-e set is a contender and suitable choice for solving low-thrust trajectory optimization problems.

Acknowledgments

"Never will I leave you, never will I forsake you" (Hebrews 13:5 NIV).

First I would like to dedicate this work to my parents, Jawaharlal Ayyanathan and Shanmuga Sundari, whose unwavering love, support, and encouragement have been the driving force behind my academic journey. Their belief in my abilities has been a constant source of inspiration, and I am eternally grateful for their sacrifices.

I would also like to thank my advisor, Dr. Ehsan Taheri, for his invaluable guidance. His constructive feedback and continuous support throughout my research have been instrumental in shaping this thesis. I am fortunate to have had the opportunity to work under his mentorship. I would also like to extend my gratitude to my exceptional lab mates, Jack, Keziban, and Nick. Their willingness to help even during the most challenging times has made our lab a vibrant and supportive environment.

I extend my sincere thanks to my committee members, Dr. Davide Guzzetti and Dr. Nan Li, for their valuable insights and constructive suggestions that greatly enhanced the quality of this thesis.

Lastly, I would like to acknowledge the Aerospace Department at Auburn University for providing me with a stimulating and nurturing academic environment. The resources, facilities, and opportunities provided by the department have played a crucial role in my intellectual growth and overall learning experience.

To all those who have supported me, directly or indirectly, throughout this arduous yet fulfilling endeavor, please accept my heartfelt appreciation. Your unwavering belief in my abilities has been a constant source of motivation, and I am truly grateful for your contributions to my academic success.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACT	Adjoint Control Transformation
COE	Classical Orbital Element
DS1	Deep Space 1
GEO	Geostationary Earth Orbit
HTS	Hyperbolic Tangent Smoothing
LVLH	Local-Vertical Local-Horizontal
MACT	Mapped Adjoint Control Transformation
MEE	Modified Equinoctial Elements
NAIF	Navigation and Ancillary Information Facility
NMP	New Millenium Program
OCP	Optimal Control Problem
PMP	Pontryagin Minimum Principle
SEP	Solar Electric Propulsion
SPICE	Spacecraft, Planet, Instrument, Camera-matrix, Events
TPBVP	Two-Point Boundary-Value Problem

Chapter 1

Introduction

NASA launched its New Millenium Program (NMP) in 1994 to test and validate new technologies that aimed to develop cost-effective approaches to design space missions. The NMP was responsible for several notable technology demonstrations. Deep Space 1 (DS1) mission was the first and one of the most significant missions of the NMP. Launched in 1998, DS1 tested Solar Electric Propulsion (SEP) which was used to propel the DS1 spacecraft to encounter an asteroid 3352 McAuliffe [1]. Since then, several missions like SMART-1 [2], Dawn [3] and Hayabusa [4] have successfully demonstrated the use of electric propulsion systems. In addition, Hall-effect thrusters are considered for one of NASA's missions to asteroid (16) Psyche [5]. SEP is now used by nearly half of the satellites in Geostationary Earth Orbit (GEO) for North-South station keeping [6]. SEP systems have considerable economic advantages including reduced launch mass, increased payload mass, and increased operational lifetime. For instance, fully fueled Dawn spacecraft had a total wet mass of about 1200 kilograms with 450 kilograms of xenon propellant. A conventional propellant chemical system would only take a few hours to provide an equivalent total impulse but would require over 66,000 kilograms of propellant [6]. Electric propulsion systems have higher specific impulse values than chemical propulsion systems, making them appealing from a fuel-efficiency point of view. However, to provide the required change in the energy of the spacecraft, thruster(s) has (have) to operate for extended continuous time intervals (on the order of days or even a few months). Moreover, long coasting arcs may separate the thrusting arcs and the sequence and number of thrusting and coasting arcs are not known in advance. This thrust-coast-thrust control profile is a characteristic of minimum-fuel low-thrust trajectories. Thus, generating minimum-fuel, low-thrust trajectories can become quite challenging [7]. Also unlike chemical propulsion systems, spacecraft trajectories with electric propulsion systems are affected by the launch vehicle, solar array power, number and type of thrusters, total mission duration, and the sequence of main events (e.g, number and sequence of the gravity-assist maneuvers in more complex mission scenarios) [8, 9, 10, 11]. The selection of the electric propulsion system will also affect the overall optimality of the resulting trajectories [9, 10, 12, 11].

A trajectory optimization problem can be set up while using a low-thrust propulsion system that aims at maximizing or minimizing a particular cost functional (i.e., to minimize the duration of flight time or maximize the final mass or net delivered payload mass). Multidisciplinary design optimization problems [13, 14] and problems that take into account certain scientific merits [15] are also considered by researchers. Multi- and single-objective optimization of low-thrust trajectories with gravity-assist maneuvers are also considered [16, 17, 18, 19, 20, 21, 22, 23]. In Ref. [24], a minimum-fuel trajectory for orbital hopping missions is considered and the optimization problem is formulated as multiple-phase nonlinear Optimal Control Problems (OCPs) . The problem formulation in [24] contains multiple exo-atmospheric and atmospheric flight phases, which require consideration of two sets of flight dynamics.

In general, OCPs can be solved using *direct* or *indirect* optimization methods [25] or hybrid methods [26, 27]. Direct methods solve for the states and control inputs approximately through the discretization of continuous-time equations of motion into their discrete counterparts to form a non-linear programming (NLP) problem [28]. The main disadvantage of the direct method is that the solution is not guaranteed to be optimal due to the approximations involved within its formulation of OCPs. Ref. [29] provides a review on the state-of-the-art development in numerical multi-objective trajectory optimization algorithms and spacecraft stochastic trajectory planning techniques.

A significant portion of this thesis has already been published and can be found in [30], [31] and [32].

1.1 Indirect Method Overview

The indirect method (typically) forms the set of first-order necessary conditions of optimality using the principles of the calculus of variation and optimal control theory [33]. In essence, the original (dynamic) optimization problem is converted into determining the value or time history of the so-called costates (or Lagrange multipliers) through which the trajectory optimization problem is converted into a Hamiltonian boundary-value problem (i.e., a multi-point boundary-value problem or a two-point boundary-value problem (TPBVP)) [34].

Since first-order necessary conditions of optimality are used to form the Hamiltonian boundary-value problems, the solutions to the Hamiltonian boundary-value problems are extremal/stationary solutions. In the remainder of the thesis, the words "extremal" and "stationary" are used interchangeably. The resulting extremal solutions are of high resolution (in both states and controls vs. time) that are instrumental for not only certain aerospace applications but also provide valuable databases for methods that use supervised learning methods for trajectory optimization [35]. By "high resolution", we mean that we obtain a very accurate (temporal) profile of control inputs and their corresponding states. While trajectory optimization is a prevalent step in mission design, it is also possible to perform co-optimization (i.e., simultaneous optimization) of spacecraft power subsystem parameters (e.g., the solar array size), optimal modes of a multi-mode thruster and trajectory for maximizing the net delivered payload [36]. Although direct optimization methods are easier to implement and do not require derivations using calculus of variations, the accuracy of the solution is compromised by considering approximations and there is no guarantee that the resulting solutions are extremal [37]. The indirect methods, on the other hand, can become quite difficult to apply to complex problems and the resulting Hamiltonian boundary-value problems exhibit extremely sensitive to the value of unknown variables (i.e., have a small domain of convergence). The numerical solutions to the Hamiltonian boundary-value problems are difficult to obtain due to several reasons:

1. In most of the TPBVPs, the costates at the initial time are unknown and an efficient strategy has to be devised to guess their values. However, it is typically the case that

some (or all) states are known at both initial time (initial constraints) and final time (final constraints) in a rendezvous-type trajectory optimization problem. As part of the solution scheme, an integration algorithm is required to propagate the state and costate dynamics using the initial state values and the guessed costate values to check for the closeness of the final propagated state values to the final constraints.

2. The solution of the resulting TPBVPs is highly sensitive to the value of the initial costates and therefore the domain of convergence of the TPBVP is small. Even though costates carry sensitivity information of the cost functional relative to the change in states, they are not physically meaningful [38], which makes their estimation to be a difficult task.

3. In minimum-fuel low-thrust trajectory optimization problems, the optimal thrust profile can exhibit multiple switches between maximum and zero values. The number of switches as well as the time duration of the thrust and coast arcs are not known a priori. The presence of these unknown numbers of non-smooth switches in control reduces the domain of convergence of the resulting TPBVPs substantially.

4. Addition of state-only and mixed state-control equality and inequality constraints requires the derivation of additional necessary conditions (and potential tangency conditions through the addition of additional constant Lagrange multipliers). For the particular case of inequality constraints, activation of the constraints, duration of the activation, and the possibility of having multiple activation and inactivation periods, make the solution of TPBVPs extremely difficult.

Briefly, indirect optimization methods face notable challenges such as the requirement of an initial guess for the costates (typically at the initial time), convergence issues due to discontinuous events (e.g., switches in control profiles or the presence of eclipse entry and exit events [39]), small convergence radius, and more importantly, difficulty in handling state path constraints. However, if appropriate regularization and numerical continuation techniques are leveraged, indirect optimization methods possess attractive numerical features [40, 41, 12] (e.g., rapid convergence and substantially fewer number of design variables compared to the direct method). These features become useful for spacecraft trajectory design, since the motion of the spacecraft is predominantly governed by two-body dynamics during heliocentric phases of space flights [42, 43].

1.2 Improving the Convergence of Indirect Methods

In general, a number of factors impact the convergence performance of the algorithms used for solving TPBVPs, including, 1) the choice of coordinates (e.g., Cartesian versus other sets of elements), 2) regularization of non-smoothness events (e.g., throttle switches and eclipse entry and existing conditions), 3) proper scaling of states, and 4) leveraging homotopy/continuation methods. All the aforementioned items can play significant roles in obtaining extremal solutions to challenging OCPs.

To overcome the difficulties associated with solving OCPs using indirect methods, homotopy or numerical continuation techniques are frequently used [44, 40, 45, 46, 47, 48, 49]. In fact, it is not exaggerated if we state that for solving practical OCPs, it is extremely difficult (if not impossible) to find the solution of OCPs without resorting to numerical continuation/homotopy methods [25, 50]. The principal idea behind numerical continuation and/or homotopy techniques is to embed the original OCP into a one- or multiple-parameter family of neighboring OCPs and to approach the solution of the original OCP in an iterative manner by solving a series of (potentially neighboring) TPBVPs. This multi-parameter family of neighboring OCPs can be constructed through several mechanisms, for instance, by altering the cost functional, dynamics, boundary conditions, and natural parameters of the system.

For instance, let $\boldsymbol{r} = [x, y, z]^\top$ denote the position of the center of mass of a spacecraft relative to the central body (e.g., Earth), equations of motion of the spacecraft can be written in terms of the Cartesian coordinates as

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{\Delta} \rightarrow \ddot{\boldsymbol{r}} = -\kappa_1\frac{\mu}{r^3}\boldsymbol{r} + \kappa_2\boldsymbol{\Delta}_c + \kappa_3\boldsymbol{\Delta}_p, \tag{1.1}$$

where $r = ||\boldsymbol{r}||$, μ denotes the gravitational parameter of the central body and $\boldsymbol{\Delta} = \boldsymbol{\Delta}_c + \boldsymbol{\Delta}_p$ denotes the sum of accelerations due to the propulsion system, $\boldsymbol{\Delta}_c$, and all other perturbing accelerations, $\mathbf{\Delta}_p$ (e.g., non-spherical gravity model, solar radiation pressure and third-body perturbations). At least, for motion "close" to the central body and during preliminary phases of the trajectory design, the predominant nonlinearity is due to the μ/r^3 term. Thus, a reasonable step (to attenuate the impact of this nonlinearity) is to consider smaller values for the gravitational parameter, $\kappa_1 \in [\kappa_{1,\text{min}}, 1]$, and solve spacecraft trajectory optimization problems starting from $\kappa_1 = \kappa_{1,\text{min}}$ and its associated TPBVP. Once an extremal solution is obtained, it is possible to increase the value of κ_1 and use the converged solution of the previous TPBVP for a new TPBVP (with slightly different values for the value of κ_1). The successive solution of neighboring TPBVPs can be extended to other terms that may impact the result substantially. For instance, the control authority due to the propulsion system can be modulated as well as the influence of perturbations through $\kappa_2 \in [0, 1]$ and $\kappa_3 \in [0, 1]$, respectively.

Adopting a similar methodology, the cost functional of an OCP can also be perturbed/modified to perform numerical continuation. Let $\mathbf{\Delta}_c = \frac{T_m}{c}\delta\hat{\boldsymbol{\alpha}}$ denote the thrust produced by the propulsion system, in which $\delta \in [0, 1]$ denotes the throttle input and $\hat{\boldsymbol{\alpha}}$ denotes the thrust steering unit vector (i.e., $||\hat{\boldsymbol{\alpha}}|| = 1$). When the direction of the thrust vector is not constrained, all the complexities of the control profile are attributed to the switches in the profile of δ. But, it is possible to overcome the challenges due to non-smoothness of throttle. For instance, Bertrand and Epenoy [44] modified the cost functional, J, of a minimum-fuel OCP as

$$J = \frac{T_m}{c} \int_{t_0}^{t_f} [\delta + F(\epsilon, \delta)] dt, \tag{1.2}$$

where T_m denotes the maximum thrust produced by the propulsion system, c is the specific impulse, δ is the throttle control, and $F(\epsilon, \delta)$ represents a perturbing function to regularize the control, where ϵ denotes the continuation parameter. Three different forms of functions introduced by Bertrand and Epenoy are,

- Quadratic: $F(\epsilon, \delta) = -\epsilon\delta(1 - \delta)$
- Logarithmic: $F(\epsilon, \delta) = -\epsilon[\log(\delta) - \log(1 - \delta)]$
- Extended Logarithmic: $F(\epsilon, \delta) = -\epsilon[-\delta \log(\delta) - (1 - \delta) \log(1 - \delta)]$

Since the control does not appear linearly in the Hamiltonian, we can use the strong form of optimality while using the above functions. Note also that Bertrand and Epenoy have also introduced the concept of probability density functions to smooth the control by convolution in 2002 for the first time [51].

Among the numerical continuation methods, a linear approximation of the gravitational force and its utility for solving orbit transfer problems using indirect optimization methods is investigated in Ref. [52]. Nevertheless, the embedding process and the numerical solution of the resulting TPBVPs may still consist notable challenges and specialized numerical methods have to be used to find extremal solutions, as is explained in Ref. [53, 54]. The idea of smoothing enlarges the domain of convergence of the resulting TPBVPs. If smoothing is not adopted, detailed event-detection algorithms (for zero-crossing of the switch function or detection of other events) have to be developed and incorporated within the integration routines [18, 55].

Adopting specialized costate initialization strategies can also have substantial ramifications on the class of problems that can be solved. For instance, Thorne and Hall [56] obtained approximate initial costate values for a coplanar, circle-to-circle transfer by initially considering a continuous-thrust spacecraft and by neglecting the gravitational term in the equations of motion (i.e, by setting μ to 0 in Eq. (1.1)). Similar ideas, assuming flat models of the Earth's gravity, are also investigated in Ref. [57]. The ACT method developed by Dixon and Biggs [58] is an effective costate initialization method. The ACT method determines part of the non-physical initial costates by mapping them to a set of more intuitive physical variables. In essence, the initial costates are linked with physical variables (thrust vector angles and their time rate of change) in a spacecraft-centered frame. Application of the ACT for spacecraft trajectory optimization is demonstrated in Refs. [59, 60, 61]. Lee and Bang [62] presented a new guess structure for the initial costates for solving optimal spiral trajectories by using the initial costate characteristics of known minimum-fuel Earth escape trajectories. In Ref. [63], a method for costate estimation for the indirect optimal control formulation is developed by using the solution of the same problem formulated using a direct method. Yan and Wu [64] demonstrated that approximate initial costates can be obtained by solving equations in the neighborhood of the initial time by expressing the costates as first-order approximate value of the Taylor series. In Ref. [65], the initial costates of minimum-time rendezvous problems are estimated using finite Fourier series shape-based methods [66] and it is shown that this method outperforms the ACT method for the class of considered many-orbital-revolution problems.

Despite using numerical continuation methods and various techniques to guess the initial costates, solving low-thrust trajectory optimization problems can still pose many challenges. One notable solution strategy has been to enlarge the domain of convergence of the resulting TPBVPs by using alternate sets of orbital elements [40, 67, 68]. In Ref. [69], a vectorial-form dynamic model is developed based on orbital elements for solving trajectory optimization problems. In Ref. [70], the authors develop a modified version of the equinoctial elements based on the works originally developed in Ref. [71]. By using the homotopy approach, alternate set of orbital elements, and different costate initialization methods, the complexity of a low-thrust trajectory optimization problem is greatly reduced from a computational point of view.

Chapter 2

Coordinate and Element Sets

It is well-known that trajectory optimization problems can be formulated using different sets of coordinates or elements. This section presents an overview of the three main coordinates and elements used to describe the dynamics of spacecraft in this thesis. In addition, incorporation of the perturbation due to the Earth's second zonal harmonic, J_2, is presented since it will be used to investigate the solution of trajectory optimization problems in the presence of J_2 perturbation for Earth-centered problems.

2.1 Cartesian Coordinates

The Cartesian coordinates are easy to visualize and the resulting equations of motion and control expressions take their simplest forms compared to the other choice of coordinates/elements [68]. Using Eq. (1.1), the time rate of change of Cartesian position and velocity vectors can be written as

$$\dot{\boldsymbol{r}} = \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{\Delta}_{\text{c}} + \boldsymbol{\Delta}_{\text{J2}}, \tag{2.1}$$

where $\boldsymbol{r} = [x, y, z]^\top$ and $\boldsymbol{v} = [\dot{x}, \dot{y}, \dot{z}]^\top$ denote position and velocity vectors of the spacecraft, respectively, μ is the gravitational parameter of the central body, $r = ||\boldsymbol{r}||$ denotes the Euclidean norm of the position vector, $\boldsymbol{\Delta}_{\text{c}}$ is the (control) acceleration due to the propulsion system, and $\boldsymbol{\Delta}_{\text{J2}}$ is the perturbing acceleration due to Earth's second zonal harmonic. Perturbing accelerations due to third-body perturbation ($\boldsymbol{\Delta}_{\text{3body}}$) and solar radiation pressure ($\boldsymbol{\Delta}_{\text{SRP}}$) are ignored.

The position-velocity and control vectors for a Cartesian formulation can be written as

$$\boldsymbol{x}_{\text{cart}} = [\boldsymbol{r}^\top, \boldsymbol{v}^\top]^\top, \qquad \boldsymbol{\Delta}_{\text{c}} = [u_x, u_y, u_z]^\top = \frac{T}{m}\delta\hat{\boldsymbol{u}}, \tag{2.2}$$

where T denotes the maximum thrust value, m denotes spacecraft mass, $\delta \in [0, 1]$ is the engine throttling input, and $\hat{\boldsymbol{u}}$ is the thrust steering unit vector. When $\delta = 0$, no propellant is consumed and when $\delta = 1$, engine is operating at its maximum thrust. Acceleration due to Earth's second zonal harmonic can be derived as

$$\boldsymbol{\Delta}_{\text{J2}} = -\frac{\partial U}{\partial \boldsymbol{r}}. \tag{2.3}$$

The potential function 'U' can be written as

$$U = \frac{\mu}{r}\left[J_2\left(\frac{R_E}{r}\right)^2\left(\frac{3}{2}\sin^2(\phi) - \frac{1}{2}\right)\right], \qquad r = \sqrt{x^2 + y^2 + z^2}, \qquad \sin(\phi) = \frac{z}{r}, \tag{2.4}$$

where ϕ is the geocentric latitude. Eq. (2.1) can be written in a control-affine form as

$$\dot{\boldsymbol{x}}_{\text{cart}} = \boldsymbol{A}_{\text{cart}}(\boldsymbol{x}, t) + \mathbb{B}_{\text{cart}}\left[\boldsymbol{\Delta}_c + \boldsymbol{\Delta}_{\text{J2}}\right], \tag{2.5}$$

where $\boldsymbol{A}_{\text{cart}}(\boldsymbol{x}, t)$ and $\mathbb{B}_{\text{cart}}(\boldsymbol{x}, t)$ are defined in Eq. (2.6), respectively, as

$$\boldsymbol{A}_{\text{cart}} = \begin{bmatrix} \boldsymbol{v} \\ -\frac{\mu}{r^3}\boldsymbol{r} \end{bmatrix}, \quad \mathbb{B}_{\text{cart}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} \\ \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{2.6}$$

and $\boldsymbol{I}$ represents an identity matrix and $\boldsymbol{0}$ represents a zero matrix.

2.2 Modified Equinoctial Elements

The set of modified equinoctial elements (MEE) possess desirable features for solving orbital mechanics problems. There are five elements, (p, f, g, h, k), that vary slowly as a function of perturbations and there is one fast variable, the true longitude, l. Unlike the Classical Orbital Elements (COEs) , MEEs do not exhibit any singularities at zero eccentricity and inclination

[70]. Another desirable feature of these elements is that the number of orbital revolutions around the central body can be enforced to be a specific integer value. This is an important feature for a systematic study of fixed-time, minimum-fuel rendezvous-type maneuvers [72]. The control vector in an MEE formulation is expressed in a Local-Vertical Local-Horizontal (LVLH) reference frame attached to the spacecraft [70]. The elements and control vectors for the set of MEEs can be written as

$$\boldsymbol{x}_{\text{MEE}} = [p, f, g, h, k, l]^{\top}, \qquad \boldsymbol{\Delta}_{\text{c}} = [u_r, u_t, u_n]^{\top}. \tag{2.7}$$

The MEEs can be expressed in terms of the COEs as $p = a(1-e^2)$, $f = e\cos(\omega+\Omega)$, $g = e\sin(\omega+\Omega)$, $h = \tan(i/2)\cos\Omega$, $k = \tan(i/2)\sin\Omega$, and $l = \Omega+\omega+\theta$, where the COEs are a, e, i, ω, Ω, and θ. Here, a, e, i, ω, Ω, and θ denote semi-major axis, eccentricity, inclination, argument of periapsis, right-ascension of the ascending node, and true anomaly, respectively. A representation of the Equinoctial coordinate frame is shown in Figure (2.1) where [73]

$$\underline{\boldsymbol{W}} = \frac{1}{1+h^2+k^2}\begin{pmatrix} 2k \\ 2h \\ 1-k^2-h^2 \end{pmatrix}, \tag{2.8}$$

$$\underline{\boldsymbol{F}} = \frac{1}{1+h^2+k^2}\begin{pmatrix} 1-k^2-h^2 \\ 2hk \\ -2k \end{pmatrix}, \tag{2.9}$$

$$\underline{\boldsymbol{G}} = \frac{1}{1+h^2+k^2}\begin{pmatrix} 2hk \\ 1+k^2-h^2 \\ 2h \end{pmatrix}. \tag{2.10}$$

$$\tag{2.11}$$

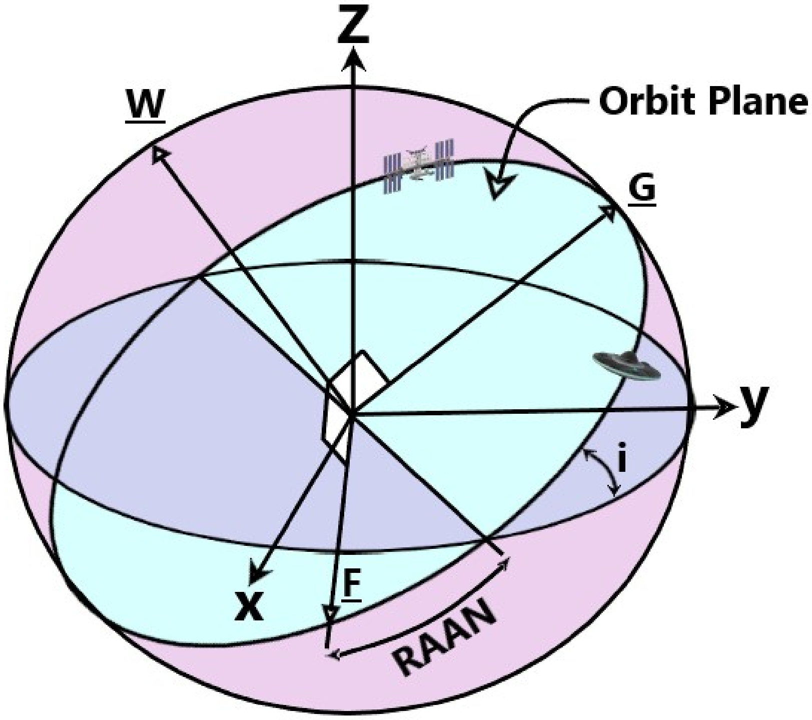


Figure 2.1: Definition of Cartesian and equinoctial coordinate frames.

Derivation of the Gauss variational equations for the set of MEEs is given in Appendix B. The time rate of change of the MEEs can be written in a compact form as

$$\dot{\boldsymbol{x}}_{\mathrm{MEE}} = \boldsymbol{A}_{\mathrm{MEE}} + \mathbb{B}_{\mathrm{MEE}} \left[\boldsymbol{\Delta}_{\mathrm{c}} + \boldsymbol{\Delta}_{\mathrm{J2}} \right]. \tag{2.12}$$

The expressions for $\boldsymbol{A}_{\mathrm{MEE}}$ and $\mathbb{B}_{\mathrm{MEE}}$ are

$$\mathbb{B}_{\mathrm{MEE}} = \begin{bmatrix} 0 & \frac{2p}{w}\sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}}\sin(l) & \sqrt{\frac{p}{\mu}}\frac{1}{w}\left[(w+1)\cos(l)+f\right] & -\sqrt{\frac{p}{\mu}}\frac{g}{w}K \\ -\sqrt{\frac{p}{\mu}}\cos(l) & \sqrt{\frac{p}{\mu}}\frac{1}{w}\left[(w+1)\sin(l)+g\right] & \sqrt{\frac{p}{\mu}}\frac{f}{w}K \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\cos(l)}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\sin(l)}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{1}{w}K \end{bmatrix}, \quad \boldsymbol{A}_{\mathrm{MEE}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{\mu p}\left(\frac{w}{p}\right)^2 \end{bmatrix}, \tag{2.13}$$

where $w = 1 + f\cos(l) + g\sin(l)$, $s^2 = 1 + h^2 + k^2$, $\kappa = h\sin(l) - k\cos(l)$, and μ is the gravitational parameter of the central body. The perturbing acceleration due to the Earth's second zonal harmonic for the set of MEEs is written as [74]

$$\boldsymbol{\Delta}_{\text{J2}} = [a_r, a_t, a_n]^\top, \tag{2.14}$$

with

$$a_r = -\frac{3J_2}{2r^4}\left[1 - \frac{12\kappa^2}{s^2}\right], a_t = -\frac{12J_2}{r^4}\left[\frac{\kappa(h\cos(l) + k\sin(l))}{s^2}\right], a_n = -\frac{6J_2}{r^4}\left[\frac{\kappa(1 - k^2 - h^2)}{s^2}\right], \tag{2.15}$$

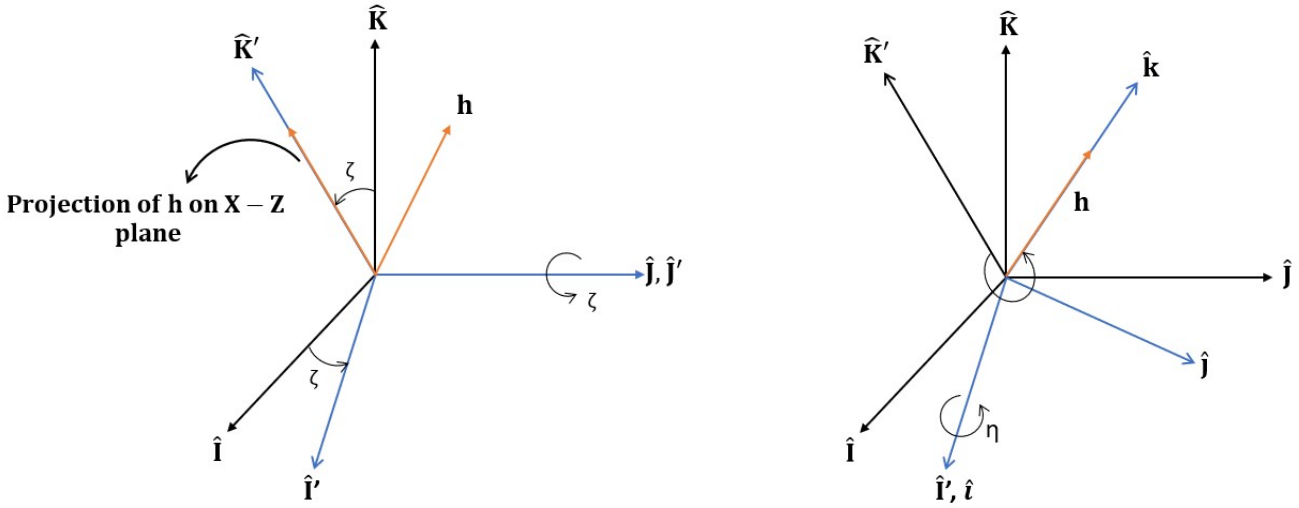
where J_2 denotes the second zonal harmonic and $r = p/w$.

2.3 h-e Orbital Elements

This set of elements is introduced in Ref. [75] with five slow elements (h, h_x, h_y, e_x, e_y) and one fast element (ϕ) that represents a true-anomaly like angle in a Keplerian motion. These elements are non-singular at equatorial orbits, but have singularity for polar orbits when right ascension of the ascending node becomes 0 or π. This set of orbital elements are obtained by rotating the inertial frame (like the Earth-centered inertial reference frame $(\hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}, \hat{\boldsymbol{K}})$) by a 2-1-3 Euler sequence of rotations through angles ζ, η, and ϕ, respectively. The rotations can be summarized as

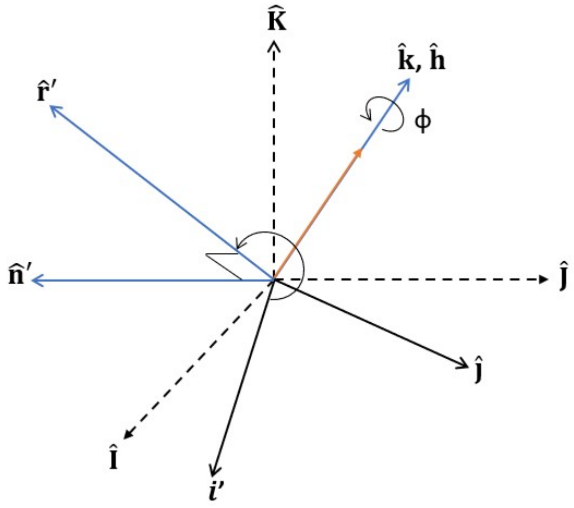
$$\mathcal{I}(\hat{\boldsymbol{I}}, \hat{\boldsymbol{J}}, \hat{\boldsymbol{K}}) \xrightarrow[\hat{J}=\hat{J}']{\zeta} \mathcal{I}'(\hat{\boldsymbol{I}}', \hat{\boldsymbol{J}}', \hat{\boldsymbol{K}}') \xrightarrow[\hat{I}'=\hat{i}']{\eta} \mathcal{O}(\hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}, \hat{\boldsymbol{k}}) \xrightarrow[\hat{k}=\hat{h}]{\phi} \mathcal{R}(\hat{\boldsymbol{r}}, \hat{\boldsymbol{n}}, \hat{\boldsymbol{h}}). \tag{2.16}$$

Figure (2.2) gives a schematic for the rotations. The elements that are used to describe the dynamics are the magnitude of specific angular momentum h, the components h_X and h_Y of specific angular momentum in the $X - Y$ plane of the inertial reference frame $\mathcal{I}$, the components e_x and e_y of the eccentricity vector of the non-inertial reference frame $\mathcal{O}$ and the true anomaly like angle ϕ.



(a) First rotation.

(b) Second rotation.



(c) Third rotation.

Figure 2.2: Sequence of $2-1-3$ Euler rotations for the h-e Orbital Elements.

The derivation details are given in Ref. [75]. Let $\boldsymbol{x}_{\text{h-e}} = [h, h_x, h_y, e_x, e_y, \phi]^\top$ and $\boldsymbol{u}_{\text{h-e}} = [a_r, a_t, a_n]^\top$ denote the h-e element and control vectors, respectively. The time rate of change of the h-e elements can be written as

$$\dot{\boldsymbol{x}}_{\text{h-e}} = \boldsymbol{A}_{\text{h-e}} + \mathbb{B}_{\text{h-e}} \boldsymbol{u}_{\text{h-e}}, \tag{2.17}$$

where $\boldsymbol{A}_{\text{h-e}}$ and $\mathbb{B}_{\text{h-e}}$ are

$$\mathbb{B}_{\text{h-e}} = \begin{bmatrix} 0 & \frac{h^2}{\mu B} & 0 \\ 0 & \frac{h h_x}{\mu B} & \frac{h^2 \sqrt{h^2 - h_x^2 - h_y^2}}{\mu B \sqrt{h^2 - h_y^2}} s_\phi + \frac{h h_x h_y}{\mu B \sqrt{h^2 - h_y^2}} c_\phi \\ 0 & \frac{h h_y}{\mu B} & -\frac{h \sqrt{h^2 - h_y^2}}{\mu B} c_\phi \\ \frac{h}{\mu} s_\phi & \frac{h}{\mu}(2c_\phi + \frac{A}{B} s_\phi) & \frac{h h_y e_y}{\mu B \sqrt{h^2 - h_y^2}} s_\phi \\ -\frac{h}{\mu} c_\phi & \frac{h}{\mu}(2s_\phi - \frac{A}{B} c_\phi) & -\frac{h h_y e_x}{\mu B \sqrt{h^2 - h_y^2}} s_\phi \\ 0 & 0 & -\frac{h h_y}{\mu B \sqrt{h^2 - h_y^2}} s_\phi \end{bmatrix}, \quad \boldsymbol{A}_{\text{h-e}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{(\mu B)^2}{h^3} \end{bmatrix}, \tag{2.18}$$

where $s_\phi = \sin(\phi)$, $c_\phi = \cos(\phi)$, $A = e_x s_\phi - e_y c_\phi$, and $B = 1 + e_x c_\phi + e_y s_\phi$. The perturbing acceleration due to the Earth's second zonal harmonic is also given in Ref. [75] as

$$\boldsymbol{\Delta}_{\text{J2}} = -\mu J_2 R_E^2 \left[\frac{3z}{r^5} \hat{\boldsymbol{K}} + \left(\frac{3}{2r^4} - \frac{15z^2}{2r^6} \right) \hat{\boldsymbol{r}} \right], \tag{2.19}$$

where

$$r = \frac{h^2}{\mu(1 + e_x c_\phi + e_y s_\phi)}, \quad z = -r c_\phi \frac{h_x}{\sqrt{h_x^2 + h_z^2}} - r s_\phi \frac{h_y h_z}{h \sqrt{h^2 - h_y^2}}. \tag{2.20}$$

Chapter 3

Indirect Formulation of Minimum-Fuel Trajectory Optimization Problems

For this thesis, we have chosen minimum-fuel trajectory optimization problems because of the complexity and difficulties associated with solving their associated TPBVPs. Specifically, solving minimum-fuel trajectory optimization problems is harder than solving minimum-time trajectory optimization ones because in the case of the latter, it is known that the thrust is always going to be switched on along the trajectory (i.e., $\delta(t) = 1$) excluding the eclipses in planet-centric phases of flight. It is therefore important to understand the performance of the costate initialization methods while solving minimum-fuel optimization problems. This ensures that the costate initialization techniques are tested and compared under more difficult trajectory optimization scenarios.

A Mayer form is considered for the performance index, which is written as

$$J = -m(t_f). \tag{3.1}$$

As discussed in Chapter 2, the equations of motion for different coordinate/element sets (Cartesian coordinates, MEE or the h-e element sets) can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x}, t) + \mathbb{B}(\boldsymbol{x}, t)\left[\boldsymbol{\Delta}_c + \boldsymbol{\Delta}_{\text{J2}}\right], \qquad \dot{m} = -\frac{T}{c}\delta, \tag{3.2}$$

where $\boldsymbol{x} \in \mathbb{R}^6$ denotes the coordinate/element vector associated with Cartesian coordinates, MEEs or the h-e set of elements, m denotes the spacecraft mass, and $c = I_{\text{sp}} g_0$ denotes the effective constant exhaust velocity. Here, I_{sp} is the specific impulse and g_0 is the acceleration due to gravity at sea level. Dropping the argument list of the $\boldsymbol{A}$ and $\mathbb{B}$ matrices, the (variational) Hamiltonian can be written as

$$H = \boldsymbol{\lambda}_{\boldsymbol{x}}^{\top}\left[\boldsymbol{A} + \mathbb{B}\boldsymbol{\Delta}_c + \mathbb{B}\boldsymbol{\Delta}_{\text{J2}}\right] - \lambda_m \frac{T}{c}\delta, \tag{3.3}$$

where $\boldsymbol{\lambda}_{\boldsymbol{x}} \in \mathbb{R}^6$ represents the costates associated with the vector $\boldsymbol{x}$ and λ_m represents the costate associated with mass. The costate dynamics are obtained using the Euler-Lagrange equation

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\left[\frac{\partial H}{\partial \boldsymbol{x}}\right]^{\top}, \qquad \dot{\lambda}_m = -\frac{\partial H}{\partial m}. \tag{3.4}$$

Eq. (3.3) shows that the Hamiltonian does not explicitly depend on time, and therefore, the Hamiltonian will remain constant throughout any extremal solution. However, the Hamiltonian is a bi-linear function of both thrust steering unit vector, $\hat{\boldsymbol{u}}$, and the throttle magnitude, δ. Therefore, the strong form of optimality conditions $\partial H/\partial \hat{\boldsymbol{u}} = 0$ and $\partial H/\partial \delta = 0$ cannot be used. In order to obtain the optimal thrust steering unit vector, Pontryagin's Minimum Principle (PMP) has to be used. According to PMP, the optimal control vector is the one that minimizes the Hamiltonian

$$\hat{\boldsymbol{u}}^* = \arg\min_{||\hat{\boldsymbol{u}}||=1} H(\boldsymbol{x}^*, m^*, \boldsymbol{\lambda}_{\boldsymbol{x}}^*, \lambda_m^*, \delta^*, \hat{\boldsymbol{u}}). \tag{3.5}$$

The Hamiltonian can be minimized by defining the primer vector as $\boldsymbol{p} = -\mathbb{B}^{\top}\boldsymbol{\lambda}_{\boldsymbol{x}}$. The optimal (denoted by '*' as a superscript) thrust steering unit vector can then be written as

$$\hat{\boldsymbol{u}}^* = \frac{\boldsymbol{p}}{||\boldsymbol{p}||} = -\frac{\mathbb{B}^{\top}\boldsymbol{\lambda}_{\boldsymbol{x}}}{||\mathbb{B}^{\top}\boldsymbol{\lambda}_{\boldsymbol{x}}||}, \tag{3.6}$$

where $||.||$ denotes Euclidean norm. Equation. (3.6) characterizes the vector part of the thrust vector; however, the throttling component has to be characterized too. Upon substituting Eq. (3.6) into Eq. (3.3), the Hamiltonian can be rearranged, simplified and written as

$$H = H_0 - S(\boldsymbol{x}, m, \boldsymbol{\lambda}_x, \lambda_m)\delta, \tag{3.7}$$

where $H_0 \in \mathbb{R}$ denotes the collection of terms that does not depend on δ and the thrust switching function is defined as

$$S = \frac{c||\mathbb{B}^\top \boldsymbol{\lambda}_{\boldsymbol{x}}||}{m} + \lambda_m. \tag{3.8}$$

The other component of the control vector, δ, appears linearly in the Hamiltonian in Eq. (3.7). Therefore, according to PMP, δ^* must minimize the Hamiltonian

$$\delta^* = \arg\min_{\delta \in [0,1]} H(\boldsymbol{x}^*, m^*, \boldsymbol{\lambda}_{\boldsymbol{x}}^*, \lambda_m^*, \hat{\boldsymbol{u}}^*, \delta) \rightarrow \delta^* = \begin{cases} 1, & \text{if } S > 0, \\ \in [0,1], & \text{if } S = 0, \\ 0, & \text{if } S < 0. \end{cases} \tag{3.9}$$

It is possible for the thrust switching function to become zero over a finite time interval (corresponding to the middle relation in Eq. (3.9)) and this condition corresponds to singular arcs [49]. In this thesis, it is assumed that singular arcs are not present along the optimal trajectory. Thus, the time history of the switching function determines the structure of the throttling input (i.e., the sequence of thrusting and coasting arcs). In particular, the sign of the switching function plays a key role, as is characterized in Eq. (3.9).

3.1 Hyperbolic Tangent Smoothing

To determine the throttle profile, precise zero-detection algorithms have to be developed to identify possible multiple zero-crossings of the switching function [76] and to make appropriate changes to the value of δ. However, a substantially easier-to-implement approach is to embed the original non-smooth control into a one-parameter family of smooth curves such that the original non-smooth bang-off-bang profile is approached in an asymptotic manner. For this purpose, the Hyperbolic Tangent Smoothing (HTS) method [46] is leveraged in which a smooth representation of optimal throttle input, δ^*, can be written as

$$\delta^*(S) \approx \delta^*(S;\rho) = \frac{1}{2}\left[1 + \tanh\left(\frac{S}{\rho}\right)\right], \tag{3.10}$$

where $\rho \in (0, \infty]$ is the smoothing parameter that will be used to regularize the non-smooth throttle control. Figure (3.1) illustrates the smoothing process of the throttle profile by varying the smoothing parameter. As the value of the smoothing parameter is decreased, the control profile approaches the theoretical optimal bang-bang profile given in Eq. (3.9). Please note that Figure (3.1) depicts one zero-crossing of the switching function, but the method is general in handling multiple zero-crossings.

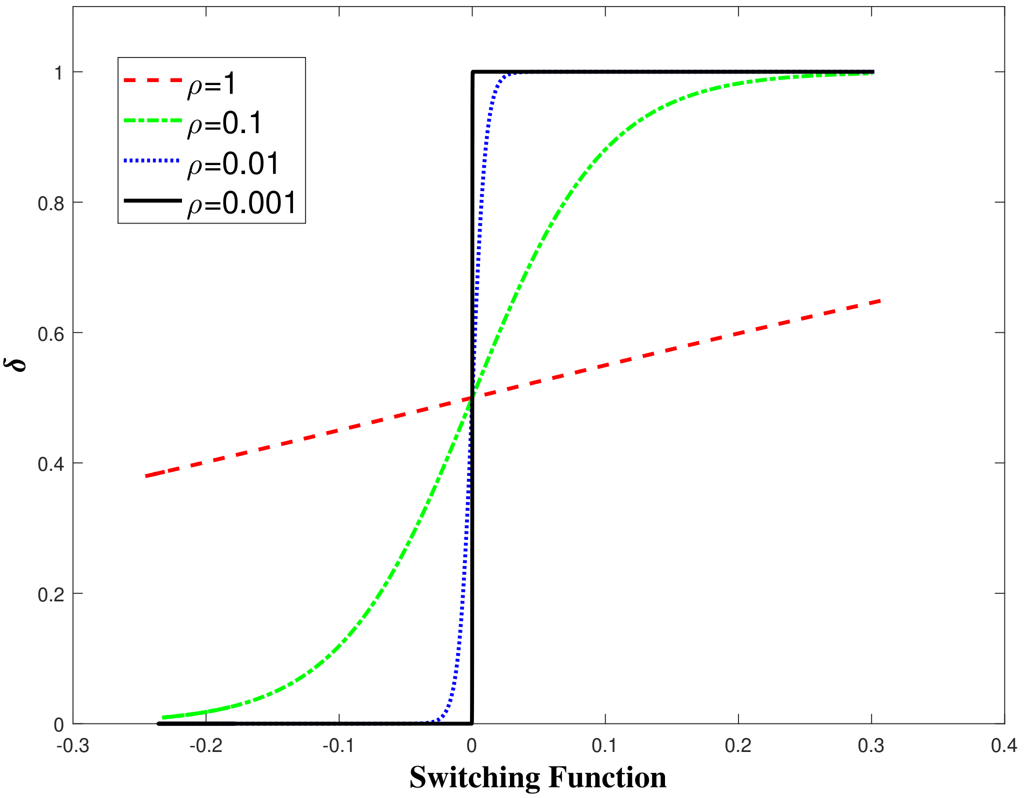


Figure 3.1: Smoothing of control using the HTS method with four values for ρ.

3.2 Single-Shooting Method

The TPBVPs considered in this thesis correspond to fixed-time, rendezvous-type maneuvers in which the initial and final states are known. Since terminal states are known, it is not possible to obtain any information regarding initial and final costates by using the transversality conditions. The only exception is for the free final mass, whose costate can be determined using the transversality conditions. There are seven final constraints that must be satisfied and are summarized as

$$\boldsymbol{\Psi}(\boldsymbol{\Lambda};\rho) = \left[[\boldsymbol{x}(t_f) - \boldsymbol{x}_T]^\top, \lambda_m(t_f) + 1\right]^\top = \boldsymbol{0}, \tag{3.11}$$

where $\boldsymbol{x}_T$ denotes the target states and t_f denotes the final time. Thus, a one-parameter family of neighboring TPBVPs is set up with the unknown design variables collectively denoted as $\boldsymbol{\Gamma} = [\boldsymbol{\lambda}_{\boldsymbol{x}}^\top(t_0), \lambda_m(t_0)]^\top$ at initial time. The resulting TPBVPs are solved by following a standard numerical continuation method, in which a large value for ρ (e.g., 1 or 0.5) is initially considered. Upon finding a solution to the first TPBVP, its solution is used for finding the solution of a new TPBVP with a lower value of ρ. The only difference between the two neighboring TPBVPs is the value of the continuation parameter, ρ. The solution to the original minimum-fuel problem can be approached by reducing the continuation parameter to "relatively" small values (e.g., $\rho < 1.0 \times 10^{-3}$) below which the cost value does not show significant change.

As depicted in Figure 3.2, a single-shooting method treats a TPBVP as an initial-value problem by guessing values for the unknown decision variables. There are several methods using which the initial costates can be guessed, with each method having its own advantages. A detailed review of some of the costate initialization methods is given in the next chapter. Using the initial guesses, the set of state-costate differential equations are propagated forward in time and the final values of states and costates are obtained. These values are then compared with the final boundary conditions. The error in the final constraint is used to iteratively update the initial costates to satisfy the final constraints within a prescribed user-defined tolerance. In this thesis, MATLAB's *ode45* is used to integrate the state and costate differential equations, Eqs. (3.2) and (3.4), respectively. Using the MATLAB *ode45* can be slow. Hence, a *mex* version of the *ode45* function is used. This step makes the code run significantly faster, since the numerical integration of the set of state-costate is the computationally expensive part of solving TPBVPs. Figure 3.2 represents the forward-in-time propagation of the set of state-costate differential equations, collectively denoted as $\dot{\boldsymbol{z}}$, along with the extremal control expressions. Here, a general form of dynamics is considered and the expressions are updated according to the set of coordinates/elements. MATLAB's built-in non-linear solver, *fsolve*, is used as the solver using which unknown values of the initial costates are iteratively improved using its in-built trust-region dog-leg algorithm until a user-defined tolerance, ϵ, on the residual vector, $\boldsymbol{\psi}$, is satisfied, i.e., $Tol = \sqrt{\boldsymbol{\psi}^T\boldsymbol{\psi}} < \epsilon$.

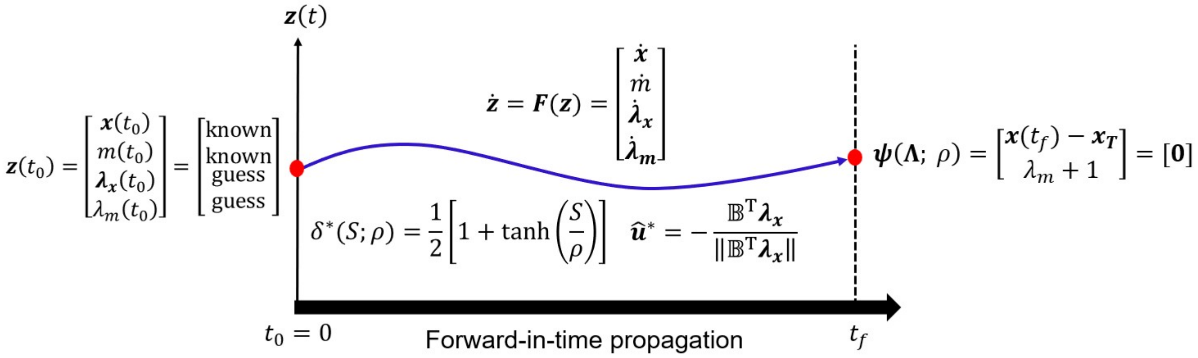


Figure 3.2: A schematic of the single-shooting solution procedure.

Chapter 4

Costate Initialization Techniques

4.1 Random Guessing

Random guessing of the initial costate values is the simplest approach to implement for solving a TPBVP. However, this method requires experience to choose the correct range for the initial costates. An educated guess for some of the initial costates is possible by analyzing their differential equations and final conditions (if it is known *a priori* from the transversality conditions). For instance, if the cost functional of a minimum-fuel trajectory optimization problem is written in Mayer form (i.e., $J = -m(t_f)$), the final mass costate has a value of $\lambda_m(t_f) = -1$ (due to transversality conditions under Minimum Principle assumptions). Since the mass costate has a negative time derivative after substituting for the optimal thrust steering vector in the mass costate differential equation (i.e., $\dot{\lambda}_m = -T||\boldsymbol{\lambda}_v||\delta/m^2 \leq 0$), extensive numerical simulations show that initial mass costate values close to zero lead to better convergence. However, such insights are not available for the costates associated with the other states and guessing a good initial value relies heavily on experience and also on the adopted units for scaling the problem. A pseudocode for solving a TPBVP using random initialization of costates is given in Algorithm (1).

Algorithm 1 Algorithm to solve a TPBVP using random initialization of costates.

```
Initialize $\rho = 1$
define a tolerance $\epsilon$
randomly guess $[\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}, \lambda_m]$
for 1 : 5 do
    solve the TPBVP by following the single-shooting method
    if $\sqrt{\boldsymbol{\psi}^T\boldsymbol{\psi}} \leq \epsilon$ then
        use the converged normalized initial costates as the guess for the next iteration
        update the continuation parameter, $\rho = \rho \times 0.1$
    else
        break and choose better initial costates
    end
end
```

It is also known that the mass costate estimation can be entirely removed from the problem formulation by multiplying the final mass by a constant positive factor (say $k > 0$) whose value is not known in advance before solving the problem. However, multiplying the cost functional with a constant positive factor does not affect the optimal solution [60]. Thus, by setting $\lambda_m(t_0) = 0$ (and removing it from the set of unknown initial costates) and after solving the resulting TPBVP, the final value of costate associated with mass is known. Assume that the final value of the costate associated with mass to be $\lambda_m(t_f) = -k$, which is precisely the condition that has to be satisfied upon applying the transversality condition on the new cost functional. For example, after solving the TPBVP and if we have $\lambda_m(t_f) = -1.1$ (while the other constraints in $\boldsymbol{\psi} = \mathbf{0}$ are satisfied), we can claim that we were minimizing $J = -1.1m(t_f)$ without affecting the optimal solution.

4.2 Costate Hypersphere Mapping

It is well-known that by multiplying the Lagrange cost functional (associated with the minimum-fuel problem) with a positive constant coefficient, λ_0, the solution of the OCP is not affected due to the homogeneity of the costate differential equations [77]. Motivated by this idea, the costate values can be mapped to the surface of a hypersphere, as introduced by Jiang et al. [18]. By doing so, the cost functional (taken in Lagrange form) and the Hamiltonian will take the following forms,

$$J = \lambda_0 \int_{t_0}^{t_f} \frac{T}{c}\delta, \tag{4.1}$$

$$H = \lambda_0 \frac{T}{c}\delta + \boldsymbol{\lambda}_{\boldsymbol{x}}^\top \left[\boldsymbol{A} + \mathbb{B}\frac{T}{m}\delta\hat{\boldsymbol{\alpha}} \right] - \lambda_m \frac{T}{c}\delta. \tag{4.2}$$

It can be shown that the switching function associated with the new cost functional takes the following form

$$S = \frac{c||\mathbb{B}^\top \boldsymbol{\lambda}_x||}{\lambda_0 m} + \frac{\lambda_m}{\lambda_0} - 1. \tag{4.3}$$

Now let the positive factor λ_0 together with the initial costates be considered as the new set of costates as

$$\boldsymbol{\lambda} = [\lambda_0, \boldsymbol{\lambda}_{\boldsymbol{x}}^\top, \lambda_m]^\top. \tag{4.4}$$

Multiplying the new set of costates with a positive constant does not change the value of the switching function (see Eq. (4.3)). Also, costate differential equations are homogeneous with respect to the cosates. Since we are free in selecting the positive constant coefficient, it is possible to choose it to be the inverse of the Euclidean norm of the costate vector at the initial time [18]. Thus, a normalized set of costates, $\tilde{\boldsymbol{\lambda}}$, is formed as

$$\tilde{\boldsymbol{\lambda}} = \frac{1}{||\boldsymbol{\lambda}(t_0)||}\boldsymbol{\lambda} = \frac{\boldsymbol{\lambda}}{||\boldsymbol{\lambda}(t_0)||}, \tag{4.5}$$

By definition, at the initial time, the resulting normalized vector is a unit vector in the (N + 1)-dimensional space, given in Eq. (4.6) as

$$||\tilde{\boldsymbol{\lambda}}(t_0)|| = 1, \tag{4.6}$$

and its components are bounded within $[-1, 1]$. Through this costate normalization procedure, the normalized costate values are restricted to lie on a unit eight-dimensional hypersphere. Thus, Eq. (3.11) has to be updated to take into account this additional constraint and is written as

$$\boldsymbol{\Psi}(\tilde{\boldsymbol{\lambda}}(t_0);\rho) = \left[\tilde{\lambda}_0, [\boldsymbol{x}(t_f) - \boldsymbol{x}_T]^\top, \tilde{\lambda}_m(t_f), ||\tilde{\boldsymbol{\lambda}}(t_0)|| - 1\right]^\top = \boldsymbol{0}. \tag{4.7}$$

Reference [18] further proposes a procedure to express the normalized costate variables in terms of 7 angles ($\beta_1, \cdots, \beta_7$). For instance, if a Cartesian coordinate system is considered to represent the spacecraft motion, then the normalized costate variables, $\tilde{\boldsymbol{\lambda}} = [\tilde{\lambda}_0, \tilde{\boldsymbol{\lambda}}_{\boldsymbol{r}}^\top, \tilde{\boldsymbol{\lambda}}_{\boldsymbol{v}}^\top, \tilde{\lambda}_m]^\top$, can be expressed in terms of the angles as

$$\tilde{\lambda}_0 = \sin(\beta_1), \tag{4.8}$$

$$\tilde{\boldsymbol{\lambda}}_{\boldsymbol{r}}(t_0) = \cos(\beta_1)\cos(\beta_2)\cos(\beta_3)[\cos(\beta_4)\cos(\beta_6), \cos(\beta_4)\sin(\beta_6), \sin(\beta_4)]^\top, \tag{4.9}$$

$$\tilde{\boldsymbol{\lambda}}_{\boldsymbol{v}}(t_0) = \cos(\beta_1)\cos(\beta_2)\cos(\beta_3)[\cos(\beta_5)\cos(\beta_7), \cos(\beta_5)\sin(\beta_7), \sin(\beta_5)]^\top, \tag{4.10}$$

$$\tilde{\lambda}_m(t_0) = \cos(\beta_1)\sin(\beta_2), \tag{4.11}$$

with 'β_i' defined as follows,

$$\beta_{1,2,3} = \frac{\pi}{2}\boldsymbol{X}_{1,2,3} \in \left[0, \frac{\pi}{2}\right], \tag{4.12}$$

$$\beta_{4,5} = \pi\left(\boldsymbol{X}_{4,5} - \frac{1}{2}\right) \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right], \tag{4.13}$$

$$\beta_{6,7} = 2\pi\boldsymbol{X}_{6,7} \in [0, 2\pi], \tag{4.14}$$

where,

$$\boldsymbol{X}_1 \in (0, 1], \qquad \boldsymbol{X}_{2,3,4,5,6,7} \in [0, 1]. \tag{4.15}$$

Therefore, the new design variables are $\boldsymbol{X}_i$ ($i = 1, \cdots, 7$) and these values have to be iterated over to find a solution. A noticeable advantage of the above mapping is that it offers a mechanism to search over the design variables within the [0,1] range. In addition, the scaled costates are automatically guaranteed to lie on the hypersphere, which allows removing $||\tilde{\boldsymbol{\lambda}}(t_0)|| - 1 = 0$ constraint from the set of constraints in Eq. (4.7). A pseudocode for solving a TPBVP using the costate hypersphere mapping method is shown in Algorithm (2).

Algorithm 2 Algorithm to solve a TPBVP using the hypersphere mapping of costates.

```
Initialize $\rho = 1$
define a tolerance $\epsilon$
randomly guess $\boldsymbol{X}_i$ $(i = 1, \cdots, 7)$
find the normalized costate variables, $\tilde{\boldsymbol{\lambda}}$ using Eqs. (4.8)-(4.11)
for 1 : 5 do
    solve the TPBVP using $\tilde{\boldsymbol{\lambda}}$ by following the single-shooting method
    if $\sqrt{\boldsymbol{\psi}^T \boldsymbol{\psi}} \leq \epsilon$ then
        use the converged normalized initial costates as the guess for the next iteration
        update the continuation parameter, $\rho = \rho \times 0.1$
    else
        break and choose better $\boldsymbol{X}_i$ $(i = 1, \cdots, 7)$ and repeat the algorithm
    end
end
```

4.3 Adjoint Control Transformation

The ACT method maps the non-physical costates to physical variables in a spacecraft-centered frame. A spacecraft-centered frame is taken with one axis in the direction of osculating spacecraft velocity ($\hat{\boldsymbol{V}}$), another axis parallel to the osculating angular momentum direction ($\hat{\boldsymbol{h}}$), and the third axis ($\hat{\boldsymbol{b}}$) completes the right-handed coordinate system as

$$\hat{\boldsymbol{V}} = \frac{\boldsymbol{v}}{||\boldsymbol{v}||}, \quad \hat{\boldsymbol{h}} = \frac{\boldsymbol{r} \times \boldsymbol{v}}{||\boldsymbol{r} \times \boldsymbol{v}||}, \quad \hat{\boldsymbol{b}} = \hat{\boldsymbol{h}} \times \hat{\boldsymbol{V}}. \tag{4.16}$$

A representative diagram for the spacecraft-centered frame is shown in Figure (4.1). The thrust steering unit vector ($\hat{\boldsymbol{u}}$) can be obtained in the spacecraft-centered frame by defining two angles, α and β with α denoting an in-plane ($\hat{\boldsymbol{V}} - \hat{\boldsymbol{b}}$) angle measured positively away from $\hat{\boldsymbol{V}}$ and β denotes an out-of-plane angle above or below the $\hat{\boldsymbol{V}} - \hat{\boldsymbol{b}}$ plane. The thrust steering unit vector and its derivative (in the spacecraft-centered frame denoted by the prime) can be written

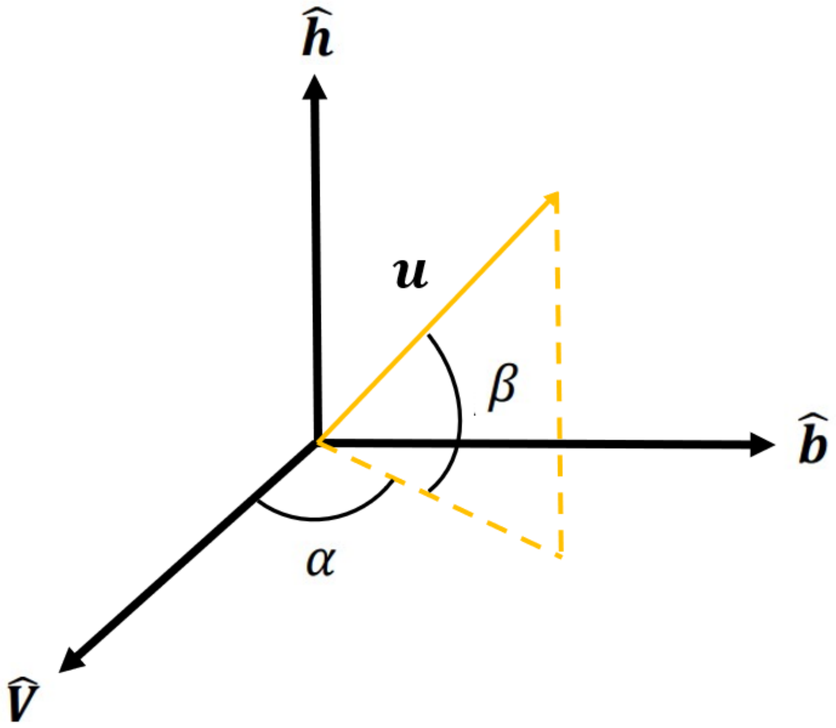


Figure 4.1: A schematic of the spacecraft-centered frame.

as

$$\hat{\boldsymbol{u}}' = \begin{bmatrix} \cos\alpha\cos\beta \\ \sin\alpha\cos\beta \\ \sin\beta \end{bmatrix}, \quad \dot{\hat{\boldsymbol{u}}}' = \begin{bmatrix} -\dot{\alpha}\sin\alpha\cos\beta - \dot{\beta}\cos\alpha\sin\beta \\ \dot{\alpha}\cos\alpha\cos\beta - \dot{\beta}\sin\alpha\sin\beta \\ \dot{\beta}\cos\beta \end{bmatrix}. \tag{4.17}$$

The thrust steering unit vector, $\hat{\boldsymbol{u}}'$, and its derivative, $\dot{\hat{\boldsymbol{u}}}'$, can then be transformed into the Cartesian frame by following

$$\hat{\boldsymbol{u}} = R\hat{\boldsymbol{u}}', \quad \dot{\hat{\boldsymbol{u}}} = R\dot{\hat{\boldsymbol{u}}}' + \dot{R}\hat{\boldsymbol{u}}', \tag{4.18}$$

where R is the rotation matrix from the spacecraft-centered frame to the Cartesian frame and $\dot{R}$ is its first-time derivative, i.e., $\dot{R} = dR/dt$. The costate associated with the velocity vector can be obtained by using the definition of the primer vector, see Eq. (3.6), and can be written as

$$\hat{\boldsymbol{u}} = \frac{\boldsymbol{\lambda_v}}{\lambda_v}, \rightarrow \boldsymbol{\lambda_v} = \lambda_v\hat{\boldsymbol{u}}, \tag{4.19}$$

where $\lambda_v = ||\boldsymbol{\lambda_v}||$. The costate associated with the position vector can be obtained by using the velocity costate differential equation in a Cartesian formulation. Using Eqs. (2.6) and (3.4) we get

$$\boldsymbol{\lambda_r} = \dot{\boldsymbol{\lambda}}_{\boldsymbol{v}} \rightarrow \boldsymbol{\lambda_r} = \dot{\lambda}_v \hat{\boldsymbol{u}} + \lambda_v \dot{\hat{\boldsymbol{u}}}. \tag{4.20}$$

By using an educated guess for λ_m, it is now possible to evaluate Eq. (3.8). By evaluating and rearranging Eq. (3.8) we get

$$\lambda_v = \frac{m}{c}(S - \lambda_m). \tag{4.21}$$

As discussed in [60] and by taking the derivative of Eq. (4.21), it is possible to estimate $\dot{\lambda}_v$ as

$$\dot{\lambda}_v = \frac{\dot{m}}{c}(S - \lambda_m) + \frac{m}{c}(\dot{S} - \dot{\lambda}_m), \tag{4.22}$$

where $\dot{m}$ and $\dot{\lambda}_m$ can be obtained using Eqs. (3.2) and (3.4) and $\dot{S}$ is guessed. Therefore, the new initial variables are $[\lambda_v, \dot{S}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \lambda_m] \rightarrow [\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}, \lambda_m]$. The advantage of using these new variables is that a larger number of them have a physical meaning. A pseudocode for solving a TPBVP using the ACT costate initialization method is shown in Algorithm (2).

Algorithm 3 Algorithm to solve a TPBVP using the ACT method.

```
Initialize $\rho = 1$
define a tolerance $\epsilon$
make an educated guess for $[\lambda_v, \dot{S}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \lambda_m]$
make the matrices $\hat{\boldsymbol{u}}'$ and $\dot{\hat{\boldsymbol{u}}}'$ using Eq. (4.17)
form $R$ to transform $\hat{\boldsymbol{u}}'$ and $\dot{\hat{\boldsymbol{u}}}'$ to Cartesian frame using Eq. (4.18)
find $\boldsymbol{\lambda_v}$ and $\boldsymbol{\lambda_r}$ using Eqs. (4.19) and (4.20) respectively
for 1 : 5 do
    solve the TPBVP using $[\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}, \lambda_m]$ by following the single-shooting method
    if $\sqrt{\boldsymbol{\psi}^T \boldsymbol{\psi}} \leq \epsilon$ then
        use the converged initial costates $[\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}, \lambda_m]$ as the guess for the next iteration
        update the continuation parameter, $\rho = \rho \times 0.1$
    else
        break and choose better $[\lambda_v, \dot{S}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \lambda_m]$ and repeat the algorithm
    end
end
```

Chapter 5

Costate Hypersphere Mapping Numerical Results

First, the Costate Hypersphere Mapping costate initialization method is compared against the random initialization method. As a test case, a fixed-time, rendezvous maneuver from Earth to comet 67P/Churyumov-Gerasimenko is considered. Comet 67P/Churyumov-Gerasimenko was the target for the European Space Agency Rosetta mission [78]. The inclination, eccentricity and semi-major axis values of the orbits of this comet are (taken from JPL small body database and given in heliocentric ecliptic J2000 reference frame) given below as

$$\begin{aligned} \text{inclination} &= 7.043449510613212 \text{ (deg)}, \\ \text{eccentricity} &= 0.641180538092906, \\ \text{semi-major axis} &= 3.463763991035476 \text{ (AU)}. \end{aligned}$$

Any low-thrust trajectory should be able to achieve the required changes in the inclination, eccentricity and semi-major axis values between the Earth and comet 67P. Such rendezvous maneuvers are achievable only through multiple orbital revolutions around the Sun and over multiple years for spacecraft equipped with solar-powered electric propulsion systems. As an example, this test case is already solved when a variable thrust engine [79] or multiple engines [11] are used. Optimal trajectories to this comet are also investigated in Ref [80].

Canonical units are used to normalize the states to guarantee a better convergence. In canonical scaling, one distance unit (DU) is taken as one astronomical unit (AU) with a value of 1.496×10^6 km and one time unit (TU) is set to 5.022×10^6 seconds. The gravitational parameter of the Sun is set to $\mu = 1$ DU^3/TU^2. The following values are considered for the spacecraft and its low-thrust propulsion system: initial mass, $m_0 = 3000$ kg, maximum thrust, $T_{\text{max}} = 0.6$ Newtons and specific impulse, $I_{\text{sp}} = 3000$ seconds. Maximum thrust and specific impulse do not change along the trajectory. All computations were performed on an Intel Core i5-6200U Laptop with a 2.30 GHz processor and 8 GB of RAM. The maximum number of function evaluations and iterations for *fsolve* are set to 10,000. The absolute and relative tolerances for *ode45* are set to 1.0×10^{-10}.

The spacecraft is set to depart from Earth on 17 June 2024, and the total time of flight of the mission is 1776 days. Spacecraft's departure and arrival position and velocity data were obtained using NASA's Navigation and Ancillary Information Facility's (NAIF) Spacecraft, Planet, Instrument, Camera-matrix, Events (SPICE) packages for MATLAB (MICE). The position and velocity data of the Earth and comet are obtained using MICE's 'cspice_spkezr.m' function. The departure and arrival times in UTC are taken as '2024 JUN 17 00:00:00.000' and '2029 APR 28 00:00:00.000', respectively. The spacecraft's departure position and velocity vectors from the Earth are

$$\boldsymbol{r}(t_0) = [-10687809.15, -151602518.3, 8676.494013]^\top \text{km},$$
$$\boldsymbol{v}(t_0) = [29.22497601, -2.197707221, 0.000972199]^\top \text{km/s}.$$

The position and velocity vectors of the spacecraft upon arrival at the comet 67P/Churyumov-Gerasimenko (the target states denoted by subscript 'T') are

$$\boldsymbol{r}_T = [-536251927.7, -126576922.3, 14541016.26]^\top \text{ km},$$
$$\boldsymbol{v}_T = [-6.858900316, -13.35248149, -0.453167946]^\top \text{km/s}.$$

Spacecraft's set of MEEs at departure (from Earth) and upon rendezvous are

$$\boldsymbol{x}_i = [0.99924, -0.00343, 0.01668, -2.96314 \times 10^{-5}, -1.44047 \times 10^{-5}, 10.92519]^\top,$$
$$\boldsymbol{x}_T = [2.00321, 0.33835, 0.55443, 0.02720, 0.01996, 28.50531]^\top.$$

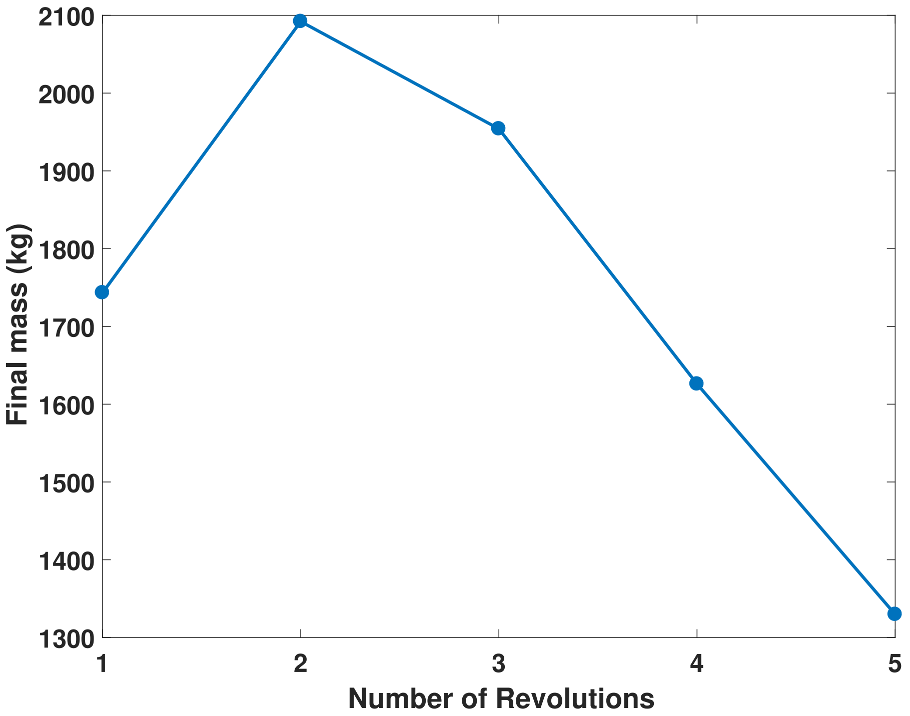


Figure 5.1: Minimum fuel Earth-Comet67P problem: Final mass for different number of revolutions around the Sun.

The value of maximum thrust, T, considered in this thesis is different from the one used in Ref [11]. The minimum-fuel trajectory consists of two revolutions around the Sun when variable thrust and multiple engines are used [11]. Since the number of revolutions can be enforced when a TPBVP is formulated using MEEs, it is possible to perform a systematic study on the number of revolutions, N_{rev}, to find the best minimum-fuel trajectory.

Figure 5.1 shows how final mass changes for different number of revolutions around the Sun, N_{rev}. It was found that the maximum final mass corresponds to the trajectory with $N_{\text{rev}} = 2$. Hence, all the simulations performed using the MEEs were achieved by fixing the number of revolutions at $N_{\text{rev}} = 2$. However, this is not the case for Cartesian coordinates, as the number of revolutions cannot be fixed in the Cartesian formulation. Therefore, some of the simulations performed using Cartesian coordinates produced sub-optimal solutions and the final mass obtained was less than the known optimal final mass.

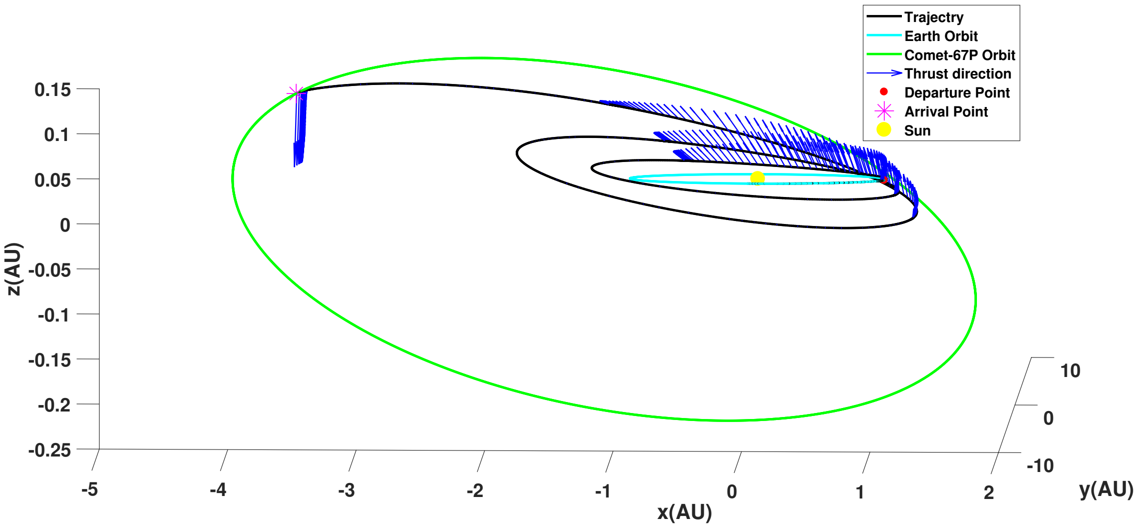


Figure 5.2: Minimum fuel Earth-Comet67P problem: Three-dimensional view of the optimal trajectory with $N_{\text{rev}} = 2$.

Figure 5.2 shows the three-dimensional minimum-fuel trajectory and Figure 5.3 shows the top view of the minimum-fuel trajectory from the Earth to comet 67P/Churyumov-Gerasimenko. The time histories of optimal control and mass are shown in Figures 5.4 and 5.5, respectively. There are seven switching events for this trajectory, and this can be seen clearly from the optimal control profile. From Figure 5.4, it can be seen that the spacecraft leaves the Earth's orbit later than the initial time for this minimum-fuel trajectory. In other words, the best strategy is to just coast along with the Earth in its orbit. The initial coast is part of a *late-departure* boundary curve introduced in Ref [81].

Numerical results indicate that the set of MEEs led to better convergence than the Cartesian coordinates, and random initialization of costates produced better convergence than when the initial costates were mapped to a unit hypersphere. Table 5.1 summarizes the statistical results (all values correspond to mean values) that were obtained for $\rho = 1$ and by running the simulations using 100 sets of random initial conditions for the different methods (where $\Delta r_f = ||\boldsymbol{r}(t_f) - \boldsymbol{r}_T||$ and $\Delta v_f = ||\boldsymbol{v}(t_f) - \boldsymbol{v}_T||$). It should be noted that the mass shown in Table 5.1 corresponds to the average mass obtained from the 100 simulations and not the best final mass. Cartesian coordinate best final mass will be equal to the case when the number of revolutions is two. But it is not guaranteed that the Cartesian coordinates will always give a

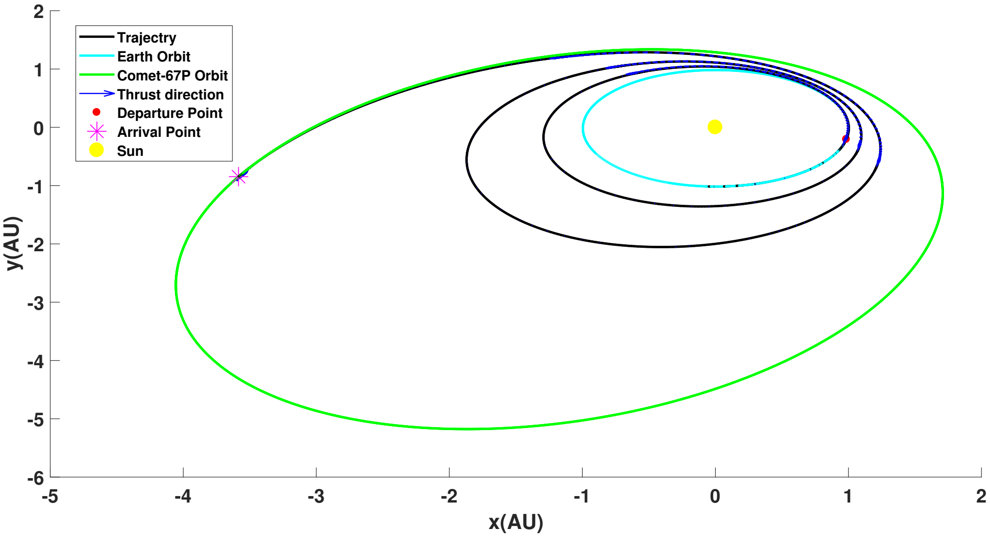


Figure 5.3: Minimum fuel Earth-Comet67P problem: Projection of the optimal trajectory with $N_{\text{rev}} = 2$ onto the ecliptic.

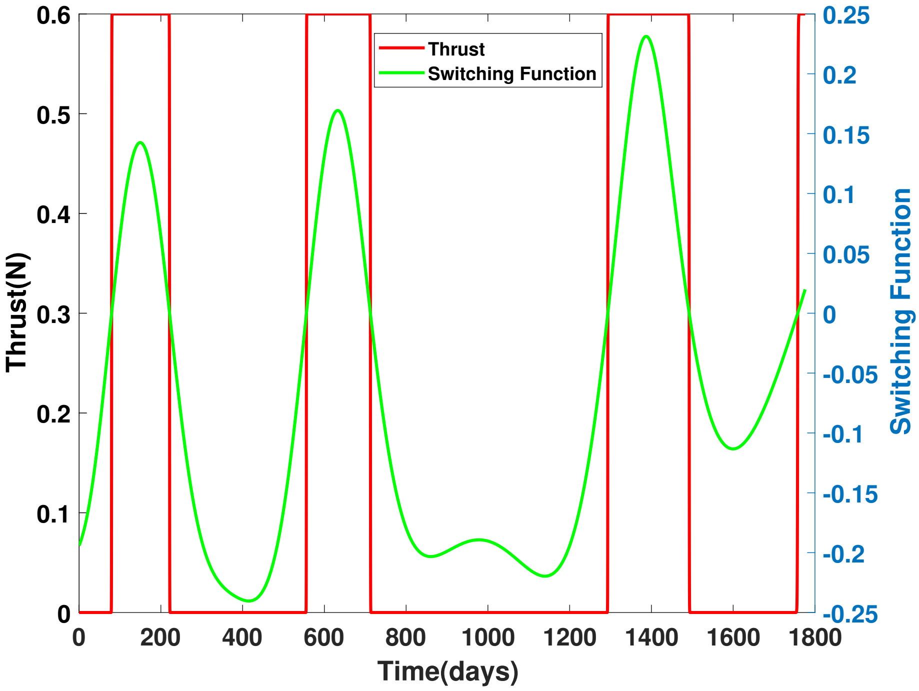


Figure 5.4: Minimum fuel Earth-Comet67P problem: Thrust history for the optimal trajectory with $N_{\text{rev}} = 2$.

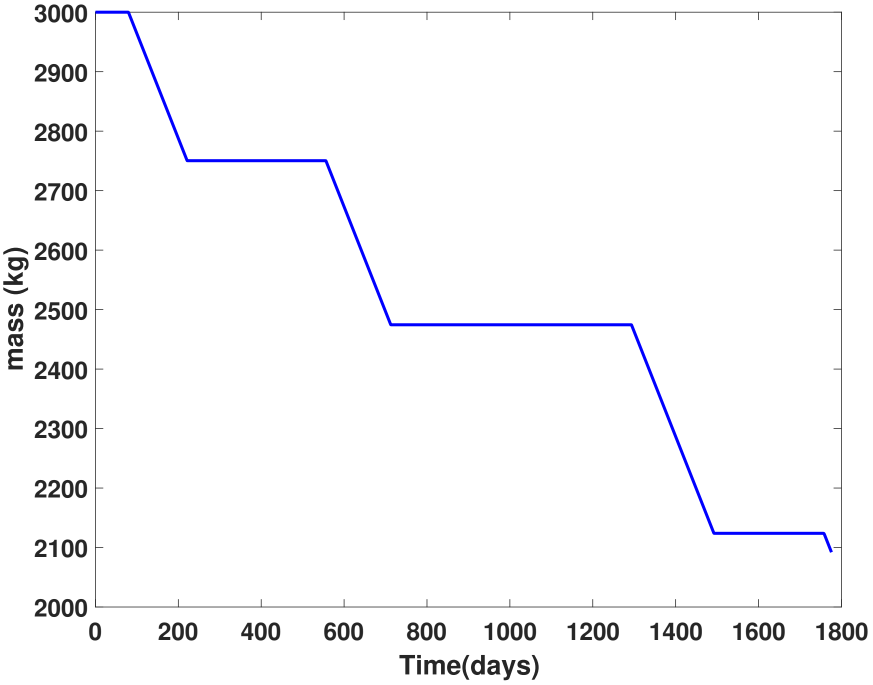


Figure 5.5: Minimum fuel Earth-Comet67P problem: Mass variation for the minimum-fuel trajectory with $N_{\text{rev}} = 2$.

solution corresponding to two revolutions. Therefore, its average final mass is less than that of the final mass corresponding to two revolutions. It can be seen that the rendezvous problem when formulated using the set of MEEs with random initialization has a convergence of 99%. This is followed by MEE with angle mapping that has a convergence of 67% and Cartesian coordinates with random initialization that has a convergence of 30%. When the set of Cartesian coordinates are used along with angle mapping of the initial costates, only 16 cases converged.

Table 5.2 summarizes the statistical results (all values correspond to mean values) that were obtained for $\rho = 1.0 \times 10^{-3}$ and by running the simulations using 100 sets of random initial conditions for the different methods. The ρ value is reduced from its initial value of 1 to its final value of 1.0×10^{-3} by following the numerical continuation approach. The ρ value is reduced by a factor of 0.1 after each iteration. Thus, four sub-problems are only solved. The rendezvous problem with MEE formulation resulted in the same convergence as that of the case when the final ρ value is 1. Cartesian formulation with random initialization resulted in 16% convergence, although it had 30% convergence when the numerical continuation method was not used and the final value of ρ was fixed to 1. Similarly, Cartesian formulation with angle mapping initialization resulted in zero convergence, although it had 16% convergence in the previous case. This suggests that a finer numerical continuation is needed. Therefore, the reduction of ρ value by a factor of 0.1 is not enough when the costates are mapped to a unit hypersphere and more sub-problems need to be solved to get an optimal solution. From Tables 5.1 and 5.2, it can also be observed that the error in final position and final velocity is, at least, an order less when the TPBVP is formulated using MEE than when the problem is formulated using the Cartesian coordinates. Also, the MEE formulation using random initialization of costates has a lower error in the final position and final velocity when compared to the MEE formulation with angle mapping of the initial costates. The final mass reported in Table 5.2 is higher compared to the final mass in Table 5.1 since a smaller value of ρ is used for generating the results in Table 5.2. The results indicate the superiority and utility of the set of MEEs for solving spacecraft low-thrust trajectory optimization problems.

Table 5.1: Statistical comparison of random initialization method with the costate hypersphere mapping method for $\boldsymbol{\rho} = 1$. (init. - initialization, func. - function, PST - Per Simulation Time)

Element set	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
Cartesian (random init.)	30	1437.8	121	923	6.2	4.9	31.4
Cartesian (angle map)	16	1796.8	416	3160	1.4	0.33	45.5
MEE (random init.)	99	1821.8	82	697	0.015	0.0061	4.6
MEE (angle map)	67	1821.8	97	688	0.25	0.051	3.3

Usually, there are no bounds on the costates and hence it is highly improbable to randomly guess a good set of initial costates that are close to the true initial costates. It is therefore intuitive to assume that the hypersphere mapping will result in a better convergence of the TPBVPs, since the costates are bounded between 0 and 1. But numerical simulations suggest otherwise. It could be that the hypersphere mapping increased the sensitivity of the problem to the value of the initial costates. While the 8D hypersphere approach is valid, it is possible that the introduction of highly nonlinear transformations, (Eqs. (4.8)-(4.11)) have a negative impact on finding the appropriate set of costates. This could be the reason that evolutionary algorithms (e.g., particle swarm optimization) are typically used to zero in on the set of costates as is discussed in [18]. Typically, a two-step solution methodology is used with numerical continuation [18]. However, in this thesis, a second optimization algorithm is not used, and instead, MATLAB's *fsolve* is used, which uses a trust-region optimization method, i.e., a Quasi-Newton method.

Table 5.2: Statistical comparison of random initialization method with the costate hypersphere mapping method for $\boldsymbol{\rho = 1.0 \times 10^{-3}}$. (init. - initialization, func. - function, PST - Per Simulation Time)

Element set	Total cases converged	m_f (kg)	Total iterations	Total func calls	Δr_f (km) $\times 10^{-3}$	Δv_f (km/s) $\times 10^{-10}$	PST (s)
Cartesian (random init.)	16	1440.5	113	879	27	0.91	22.5
Cartesian (angle map)	0	–	–	–	–	–	–
MEE (random init.)	99	2092.1	116	822	0.26	0.16	8.6
MEE (angle map)	67	2092.1	147	1012	297	132	6.1

Chapter 6

Mapped Adjoint Control Transformation

The ACT method has not been implemented for coordinate systems or element sets other than the set of Cartesian coordinates for which the original ACT method has been developed. Extension of the ACT method to other coordinates constitutes one of the important contributions of this thesis. Specifically, the idea is to extend the ACT method to alternate sets of coordinates/elements in a straightforward manner that offers implementation simplicity and can be incorporated into existing tools (e.g., Copernicus [82]) that already use the ACT method.

In a recent paper [83], a compact relation has been proposed to map the costates between two sets of coordinates/elements. For instance, the costates associated with Cartesian coordinates can be mapped to the costates associated with the set of modified equinoctial elements (MEEs). By leveraging *costate vector mapping* theorem, the ACT method is extended to alternative sets of coordinates/elements, called the MACT method, for solving low-thrust trajectory optimization problems. Thus, it is possible to combine the advantages of the ACT method and the use of alternative element sets when solving trajectory optimization problems.

In this thesis, the initial costates are still generated in a Cartesian formulation using the ACT method, and by following [83], these costates are mapped to the initial costates of different sets of orbital elements considered. Thus, codes that are developed based on the ACT method require minimal modifications to incorporate the proposed method. The utility of the method is demonstrated when two different sets of elements are used for solving low-thrust trajectory optimization problems: the set of MEEs and an elemental set based on the specific angular momentum and eccentricity vector, denoted as the h-e set of elements [75].

6.1 Derivation of Costate Vector Mapping Theorem

Detailed analyses of the utility of the costate vector mapping is presented in Ref. [83]. Consider the set of state and control vectors denoted by $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$, respectively. The cost functional can be expressed in Bolza form as

$$\underset{\boldsymbol{x}_1 \in \boldsymbol{X}_1, \boldsymbol{u}_1 \in \boldsymbol{U}_1}{\text{minimize}} \; J_1 = \Phi(\boldsymbol{x}_1(t_0), t_0, \boldsymbol{x}_1(t_f), t_f) + \int_{t_0}^{t_f} L(\boldsymbol{x}_1(t), \boldsymbol{u}_1(t), t) dt, \tag{6.1}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$ represent the admissible values for the state and control vectors, respectively, t_0 and t_f denote the initial and final times, respectively. The Hamiltonian can be written as

$$H_1 = L(\boldsymbol{x}_1(t), \boldsymbol{u_1}(t), t) + \boldsymbol{\lambda}_{\boldsymbol{x1}}^\top \, \boldsymbol{f}(\boldsymbol{x_1}(t), \boldsymbol{u_1}(t), t), \tag{6.2}$$

where $\boldsymbol{\lambda}_{\boldsymbol{x1}}$ denotes the costate vector associated with the states and $\boldsymbol{f}(\boldsymbol{x_1}(t), \boldsymbol{u_1}(t), t)$ represents the state dynamics given as

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} = \boldsymbol{f}(\boldsymbol{x_1}(t), \boldsymbol{u_1}(t), t). \tag{6.3}$$

Next consider another coordinate system with the set of states and control vectors given by $\boldsymbol{x_2}$ and $\boldsymbol{u_2}$, respectively. The Hamiltonian for this second coordinate system can be written as

$$H_2 = L(\boldsymbol{x_2}(t), \boldsymbol{u_2}(t), t) + \boldsymbol{\lambda}_{\boldsymbol{x2}}^\top \, \boldsymbol{g}(\boldsymbol{x_2}(t), \boldsymbol{u_2}(t), t), \tag{6.4}$$

where $\boldsymbol{\lambda}_{\boldsymbol{x2}}$ denotes the costate vector associated with the states and $\boldsymbol{g}(\boldsymbol{x_2}(t), \boldsymbol{u_2}(t), t)$ represent the state dynamics given as

$$\dot{\boldsymbol{x}}_{\boldsymbol{2}} = \boldsymbol{g}(\boldsymbol{x_2}(t), \boldsymbol{u_2}(t), t). \tag{6.5}$$

For a local/global extremal trajectory, the Hamiltonian (as a scalar quantity) should remain invariant under any coordinate transformation. By equating the Hamiltonian for the two coordinate systems we get

$$\boldsymbol{\lambda}_{\boldsymbol{x1}}^{\top}\, \boldsymbol{f}(\boldsymbol{x_1}(t), \boldsymbol{u_1}(t), t) = \boldsymbol{\lambda}_{\boldsymbol{x2}}^{\top}\, \boldsymbol{g}(\boldsymbol{x_2}(t), \boldsymbol{u_2}(t), t). \tag{6.6}$$

Let $\boldsymbol{\varrho}$ represent a non-linear one-to-one continuous and invertible mapping between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ such that

$$\boldsymbol{x_1}(t) = \boldsymbol{\varrho}(\boldsymbol{x_2}(t)). \tag{6.7}$$

If $\boldsymbol{\varrho}$ is not an explicit function of time, then the time derivative of the above equation can be written as

$$\dot{\boldsymbol{x}}_{\boldsymbol{1}} = \left[\frac{\partial \boldsymbol{\varrho}(\boldsymbol{x_2}(t))}{\partial \boldsymbol{x_2}(t)}\right] \dot{\boldsymbol{x}}_{\boldsymbol{2}} \rightarrow \boldsymbol{f} = \left[\frac{\partial \boldsymbol{\varrho}(\boldsymbol{x_2}(t))}{\partial \boldsymbol{x_2}(t)}\right] \boldsymbol{g} \tag{6.8}$$

Substituting Eq. (6.8) in the left-hand side of Eq. (6.6), we obtain

$$\boldsymbol{\lambda}_{\boldsymbol{x1}}(t)^{\top} \left[\frac{\partial \boldsymbol{\varrho}}{\partial \boldsymbol{x_2}}\right]\Bigg|_{t} = \boldsymbol{\lambda}_{\boldsymbol{x2}}(t)^{\top}. \tag{6.9}$$

Equation. (6.9) represents a mapping of costates between two different coordinate systems at any time instant along local extremal trajectories. This mapping provides an important step in extending the utility of the ACT method to any other sets of coordinates/elements.

6.2 Mapped Adjoint Control Transformation

Using Eq. (6.9), costates between two different sets of coordinates can be mapped. With this mapping, it is now possible to extend the ACT method to different coordinate/element sets. First, the costates are constructed using a Cartesian formulation. Next, the resulting costates are mapped to their counterparts in a different coordinate system (i.e., MEE or h-e element sets). The steps are given in Eq. (6.10)

$$[\lambda_v, \dot{S}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \lambda_m] \rightarrow [\boldsymbol{\lambda_r}, \boldsymbol{\lambda_v}] \xrightarrow{\text{Eq. (6.9)}} \boldsymbol{\lambda_{MEE}} \text{ or } \boldsymbol{\lambda_{he}}. \tag{6.10}$$

Please note that λ_m is removed from the costate mapping step since its value is invariant under any coordinate transformation. In addition, the proper non-linear mapping from Cartesian coordinates to MEE or h-e element sets has to be used when using Eq. (6.9).

Chapter 7

MACT Numerical Results

As discussed in Chapter 3, a numerical continuation approach based on the HTS method [46] is followed. The TPBVPs are initially solved by setting the smoothing parameter ρ to 1 per Eq. (3.10). To evaluate the performance of the MACT method against the random initialization method, several simulations are performed using both approaches for $\rho = 1$. Finding a first solution (with a smooth control profile) to the resulting TPBVPs is an important solution step. Thus, the main focus is on comparing the costate initialization method's performance for solving the first member of the one-parameter family of the neighboring TPBVPs. While using the MACT method, the initial values for $\alpha, \dot{\alpha}, \beta, \dot{\beta}$ are set to zero, so that the initial thrust acceleration is along the velocity direction. This is a common practice for generating initial guesses for orbit-raising maneuvers using direct optimization methods. Similarly, the initial thrust vector will be aligned against the velocity vector for orbit maneuvers to lower energetic orbits relative to the initial orbit. The value of $\dot{S}$ is also fixed and set to zero, since the history of $\dot{S}$ is not known a priori. The value for λ_m is guessed using the *rand* function in MATLAB as $0.5\times$*rand* for both the MACT and random initialization methods. In the case of the MACT method, the only other variable that needs to be guessed is λ_v. However, in the case of random initialization of costates it is not possible to approximately fix the initial costates to a particular value and all the costates associated with the states have to be guessed.

Therefore, the design variables (other than λ_m) that are to be guessed are λ_v in the case of the MACT method and the costates associated with states in the case of the random initialization method). The costates ($[\lambda_v(t_0), \dot{S}(t_0), \alpha(t_0), \dot{\alpha}(t_0), \beta(t_0), \dot{\beta}(t_0), \lambda_m(t_0)]$) are initialized in the following way for the MACT method:

- $[-1000\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-1), $[-750\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-2),
- $[-500\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-3), $[-250\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-4),
- $[-100\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-5), $[-10\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-6),
- $[-1 \times \textit{rand}, 0, 0, 0, 0, 0, 0.5 \times \textit{rand}]^\top$(CS-7), $[0.1 \times \textit{rand}, 0, 0, 0, 0, 0, 0.5 \times \textit{rand}]^\top$(CS-8),
- $[1 \times \textit{rand}, 0, 0, 0, 0, 0, 0.5 \times \textit{rand}]^\top$(CS-9), $[10 \times \textit{rand}, 0, 0, 0, 0, 0, 0.5 \times \textit{rand}]^\top$(CS-10),
- $[100\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-11), $[250\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-12),
- $[500\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-13), $[750\times \textit{rand}, 0, 0, 0, 0, 0, 0.5\times \textit{rand}]^\top$(CS-14),
- $[1000 \times \textit{rand}, 0, 0, 0, 0, 0, 0.5 \times \textit{rand}]^\top$(CS-15).

Here, 'CS' followed by a number is used to distinguish the range of design variables in the MACT method from the range of design variables in the random initialization method. In practice, it is hard to have an estimate of the range of the design variables. This step relies heavily on the practitioner's experience and also on the units used for scaling. Thus, the fifteen ranges of design variables are considered with the goal of considering a specific structure and range for variables in order to make the entire simulations repeatable by other researchers. In addition, the performance of the costate initialization methods is better determined when the percent of convergence is compared over all ranges. For each of these initial design vectors, 50 simulations are performed and verified for convergence to a solution when ρ is fixed to a value of 1. A pseudo-code for the MACT method is given in Algorithm 4. The value of ϵ is set to 1.0×10^{-8}. Please note that the proposed algorithm is less forgiving compared to the algorithms used in References [67, 68] in which the initial guesses were randomly generated within a "while" loop until a first solution was found. Here, each simulation corresponds to one set of randomly generated set of design variables. If no convergence is achieved, that simulation is immediately flagged as infeasible. Thus, more definitive and reliable conclusions can be drawn on the utility of the considered costate initialization methods.

Algorithm 4 Algorithm for the MACT method ($\rho = 1$).

```
for 1 : 50 do
    costate initialization using the ACT method
    map the costate values from Cartesian to MEE/h-e set
    solve the TPBVP using the single-shooting method
    if $\sqrt{\psi^T \psi} \leq \epsilon$ then
        store the necessary data (costates, iterations, function calls, etc)
    end
end
```

The initial costate vector for random initialization is generated as follows:

- $[-1000 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-a), $[-750 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-b),
- $[-500 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-c), $[-250 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-d),
- $[-100 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-e), $[-10.0 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-f),
- $[-1.00 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-g), $[0.10 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-h),
- $[1.00 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-i), $[10.0 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-j),
- $[100 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-k), $[250 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-l),
- $[500 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-m), $[750 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-n),
- $[1000 \times$ *rand(6,1)*$, 0.5 \times$ *rand*$]^\top$(CS-o).

In the case of the random guessing of initial costates, 50 simulations are performed and checked for convergence with $\rho = 1$. Pseudo code for the random initialization method is given in Algorithm 5.

Algorithm 5 Algorithm for the random initialization method ($\rho = 1$).

```
for 1 : 50 do
    randomly initiate the costates
    solve the TPBVP using fsolve
    if $\sqrt{\psi^T \psi} \leq \epsilon$ then
        store the necessary data (costates, iterations, function calls, and so on)
    end
end
```

Once execution of *fsolve* leads to convergence, the following parameters are stored for comparison: final mass, number of iterations, number of function evaluations, simulation time, error in final position ($||\boldsymbol{r}(t_f) - \boldsymbol{r}_T||$), and error in final velocity ($||\boldsymbol{v}(t_f) - \boldsymbol{v}_T||$) (the target states are denoted by subscript 'T').

After the simulations for MACT and random initialization are completed, the best initial costate vector is chosen for both the MACT and random initialization methods. The best costate vector range is the one with a higher convergence percentage, lower number of iterations, function evaluations, simulation time and error in final velocity and position. Once the best costate vector is chosen, 50 simulations are performed for a ρ value of 0.001 and checked for convergence. This is equivalent to finding the solution to the test case problems without following a continuation approach and is therefore a harder-to-solve problem. This is the reason why the simulations for $\rho = 0.001$ are done by only using the initial costate vector that gives the best convergence during the simulations done using $\rho = 1$.

Performance of the MACT and random initialization methods are tested and compared on four different multiple orbital revolution problems: 1) an interplanetary rendezvous maneuver from Earth to the asteroid Dionysus, 2) transfer from a GTO to the GEO, 3) transfer from a GTO to the GEO with J_2 perturbation, and 4) transfer from a GTO to the GEO with a higher number of revolutions. All computations are performed on an Intel Core i5-6200U DELL Inspiron 5559 Laptop with a 2.30 GHz processor and 8 GB of RAM. The maximum number of function evaluations and iterations for MATLAB's `fsolve` is set to 500 and 200, respectively.

For numerical integration of the differential equations, a MEX version of MATLAB's `ode45` is used with the absolute and relative tolerances set to 1.0×10^{-10}.

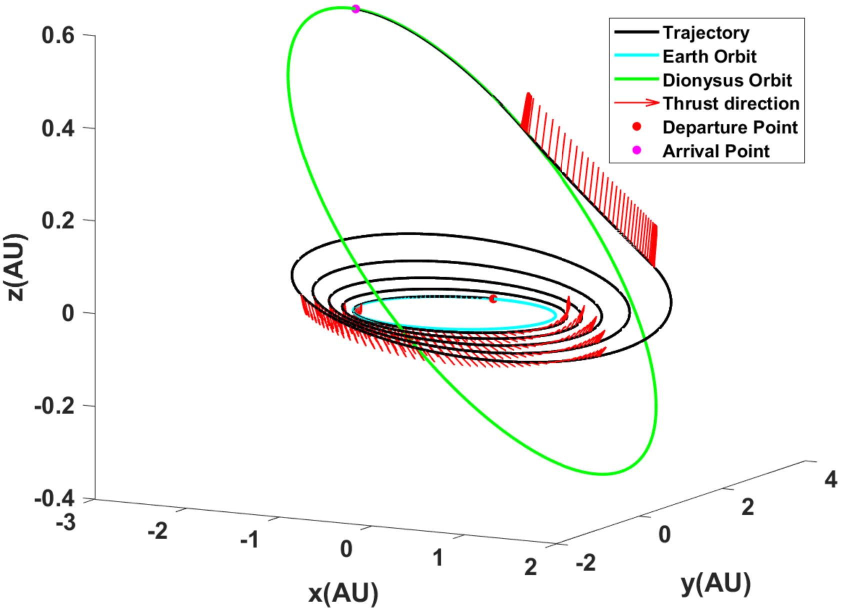
7.1 Earth-Dionysus Rendezvous Problem

This problem is already solved in Reference [67] and the boundary conditions that are reported in that paper are used. For an epoch date of 53400 (Modified Julian Date) MJD, the Keplerian orbital elements of the asteroid Dionysus are: $a = 2.2$ AU, $e = 0.542$, $i = 13.6$ degrees, $\Omega = 82.2$ degrees, $\omega = 204.2$ degrees, and mean anomaly $M = 114.4232$ degrees. Asteroid Dionysus is in a high eccentricity and inclination orbit, which makes the Earth-Dionysus trajectory optimization problem difficult. In addition, the optimal solution to this problem is known, which provides a basis for comparison and validation of the results obtained. The spacecraft's initial position vector is $\boldsymbol{r}(t_0) = [-3637871.081, 147099798.784, -2261.441]^\top$ km and the initial velocity vector is $\boldsymbol{v}(t_0) = [-30.265097, -0.8486854, 0.0000505]^\top$ km/s. The spacecraft's target position vector is $\boldsymbol{r}_T = [-302452014.884247, 316097179.632028, 82872290.075518]^\top$ km and velocity vector $\boldsymbol{v}_T = [-4.53347379984029, -13.1103098008475, 0.65616382601745]^\top$ km/s. All vectors are given in the Sun-centered Inertial frame. The state values are converted into the MEE and the h-e element sets (see Table 3 in Ref [68]) in order to use them as boundary conditions for the MEE and h-e minimum-fuel formulations. The mission's total flight time is 3534 days. In the dynamics, $\boldsymbol{\Delta}_{\text{J2}}$ is set to $\boldsymbol{0}$ for this problem. Initial mass is fixed at $m_0 = 4000$ kg. The spacecraft's low-thrust propulsion system has the following parameters: maximum thrust, $T = 0.32$ Newtons and specific impulse, $I_{\text{sp}} = 3000$ seconds. Maximum thrust value, T, and specific impulse, I_{sp}, are constant along the trajectory. The convergence of a TPBVP can be improved by adopting proper scaling strategies. This is implemented by using Canonical units, in which one distance unit (DU) is taken as one astronomical unit (AU) and one time unit (TU) is taken as 5.022×10^6 seconds. Sun's gravitational parameter is $\mu = 1$ DU^3/TU^2.

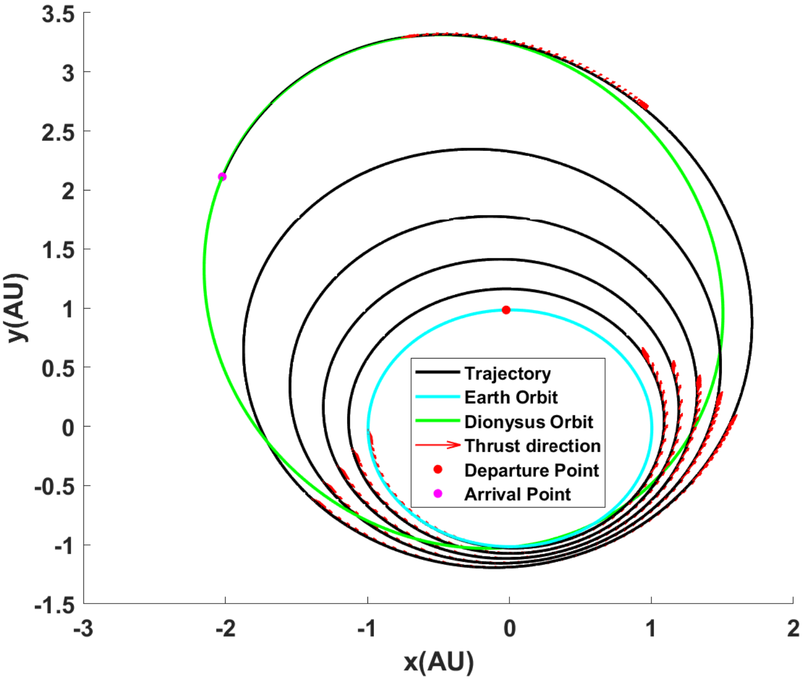
The global minimum-fuel trajectory for this problem is known with a final mass of 2718.16 kg [67], where the spacecraft makes five orbital revolutions around the Sun. The solution that is reported in this thesis (with $\rho = 1.0 \times 10^{-5}$) matches with the expected solution and the spacecraft trajectory along with the orbits of the Earth and asteroid Dionysus are plotted in Figures 7.1a and 7.1b. It is assumed that the spacecraft is at the edge of the sphere of influence of the Earth with zero excess velocity. The spacecraft initially coasts along with the Earth on its initial orbit. Even though the problem was posed as a fixed-time rendezvous problem, the spacecraft has a long coast arc at the end of the maneuver. Thus, the actual optimal solution consists of both *late-departure* and *early-arrival* phases that are introduced and studied extensively in [72]. The thrust and switching function time histories associated with the minimum-fuel trajectory are also plotted in Figure (7.2).

As explained in the previous section, convergence for $\rho = 1$ is checked for the different initial costate vectors mentioned for both the random initialization and MACT methods. The simulation results for the MEE formulation are summarized in Table 7.1 and for the h-e set formulation are summarized in Table 7.2. From Table 7.1. It can be seen that random initialization of costates has higher convergence for first nine costate vectors and the MACT method has higher convergence from CS-10 to CS-15. But while using the first nine costate vectors, the difference in the number of converged cases between random initialization and MACT methods is less than five for all the costate vectors except for the CS-5 and CS-8 cases. However, from CS-10 to CS-15 it can be seen that the MACT method has higher convergence percentage than the random initialization method and the difference in the convergence percentage is also very high. Overall, of the 750 simulations performed using the set of MEEs, the MACT method converged in 505 simulations and the random initialization method converged in 270 simulations. The simulation time is also small for the MACT method compared to when the costates are randomly initialized for most of the ranges (e.g., CS-4, CS-7, CS-9 to CS-15). The error in final position and velocity are in the same order for both the methods.

Similar results for the h-e formulation can be seen from Table 7.2. The random initialization method performed better than the MACT method only for the costate vectors CS-b, CS-c, CS-d, CS-e and CS-f. The MACT method is better over all other ranges considered. Of the 750 simulations performed using h-e formulation, the MACT method converged for 512 simulations and random initialization method converged for 251 simulations. The simulation time using the MACT method is also better than the random initialization method for most of the



(a) Earth-Dionysus problem: 3D view of the optimal trajectory.



(b) Projection of optimal trajectory onto the $x - y$ plane.

Figure 7.1: Earth-Dionysus problem: Optimal trajectory with $\rho = 1.0 \times 10^{-5}$.

Table 7.1: Statistical comparison of the MACT and random initialization methods for the MEE formulation of the Earth-Dionysus problem with $\boldsymbol{\rho}$ = 1 (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	3	2452.87	62	262	0.054	0.024	3.21
CS-a	0	-	-	-	-	-	-
CS-2	1	2452.87	45	170	0.072	0.030	2.14
CS-b	2	2452.87	46	178	1.399	0.667	1.55
CS-3	2	2452.87	42	165	0.048	0.020	2.24
CS-c	4	2452.87	50	194	0.868	0.400	2.05
CS-4	8	2452.87	47	177	0.026	0.012	2.49
CS-d	12	2452.87	74	272	0.908	0.404	3.37
CS-5	15	2452.87	49	185	0.015	0.074	2.64
CS-e	42	2452.87	46	180	0.427	0.194	2.00
CS-6	46	2452.87	35	137	0.651	0.258	1.83
CS-f	50	2452.87	43	180	0.246	0.105	1.69
CS-7	47	2452.87	25	101	0.713	0.315	1.99
CS-g	50	2452.87	57	235	0.499	0.021	2.48
CS-8	37	2452.87	25	101	0.231	0.106	2.09
CS-h	49	2452.87	48	199	0.764	0.325	1.67
CS-9	46	2452.87	22	90	0.377	0.154	1.71
CS-i	49	2452.87	71	277	0.524	0.2488	3.24
CS-10	50	2452.87	20	81	0.652	0.277	1.03
CS-j	11	2452.87	73	290	0.133	0.054	3.53
CS-11	50	2452.87	16	67	1.268	0.556	0.93
CS-k	1	2452.87	120	418	0.011	0.004	6.50
CS-12	50	2452.87	15	65	0.845	0.383	0.83
CS-l	0	-	-	-	-	-	-
CS-13	50	2452.87	15	65	1.014	0.464	1.41
CS-m	0	-	-	-	-	-	-
CS-14	50	2452.87	15	64	0.516	0.290	1.33
CS-n	0	-	-	-	-	-	-
CS-15	50	2452.87	15	64	0.283	0.208	1.35
CS-o	0	-	-	-	-	-	-

Table 7.2: Statistical comparison of the MACT and random initialization methods for the h-e set formulation of the Earth-Dionysus problem with $\boldsymbol{\rho}$ = 1 (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	2	2452.87	57	408	0.158	0.069	2.82
CS-a	2	2452.87	26	195	0.139	0.061	1.51
CS-2	0	-	-	-	-	-	
CS-b	4	2452.87	24	186	0.121	0.053	6.82
CS-3	5	2452.87	55	406	0.516	0.023	3.00
CS-c	7	2452.87	29	211	0.103	0.045	1.60
CS-4	2	2452.87	44	325	0.008	0.003	2.43
CS-d	15	2452.87	24	184	0.816	0.356	4.81
CS-5	13	2452.87	51	379	0.287	0.122	4.31
CS-e	42	2452.87	26	195	0.289	0.125	1.61
CS-6	44	2452.87	31	248	0.158	0.069	2.02
CS-f	50	2452.87	25	208	0.074	0.032	1.43
CS-7	47	2452.87	29	225	0.244	0.105	2.09
CS-g	46	2452.87	24	193	0.001	0.005	1.15
CS-8	49	2452.87	27	212	0.130	0.056	1.84
CS-h	48	2452.87	26	210	0.281	0.119	2.32
CS-9	50	2452.87	23	188	0.086	0.037	1.17
CS-i	35	2452.87	38	298	0.181	0.079	2.13
CS-10	50	2452.87	20	170	0.211	0.100	1.20
CS-j	2	2452.87	28	228	0.309	0.134	1.66
CS-11	50	2452.87	18	149	0.088	0.046	1.03
CS-k	0	2452.87	-	-	-	-	-
CS-12	50	2452.87	18	146	0.088	0.047	1.03
CS-l	0	2452.87	-	-	-	-	-
CS-13	50	2452.87	17	141	0.668	0.296	1.00
CS-m	0	2452.87	-	-	-	-	-
CS-14	50	2452.87	16	137	0.628	0.198	0.97
CS-n	0	2452.87	-	-	-	-	-
CS-15	50	2452.87	16	134	0.234	0.104	0.97
CS-o	0	2452.87	-	-	-	-	-

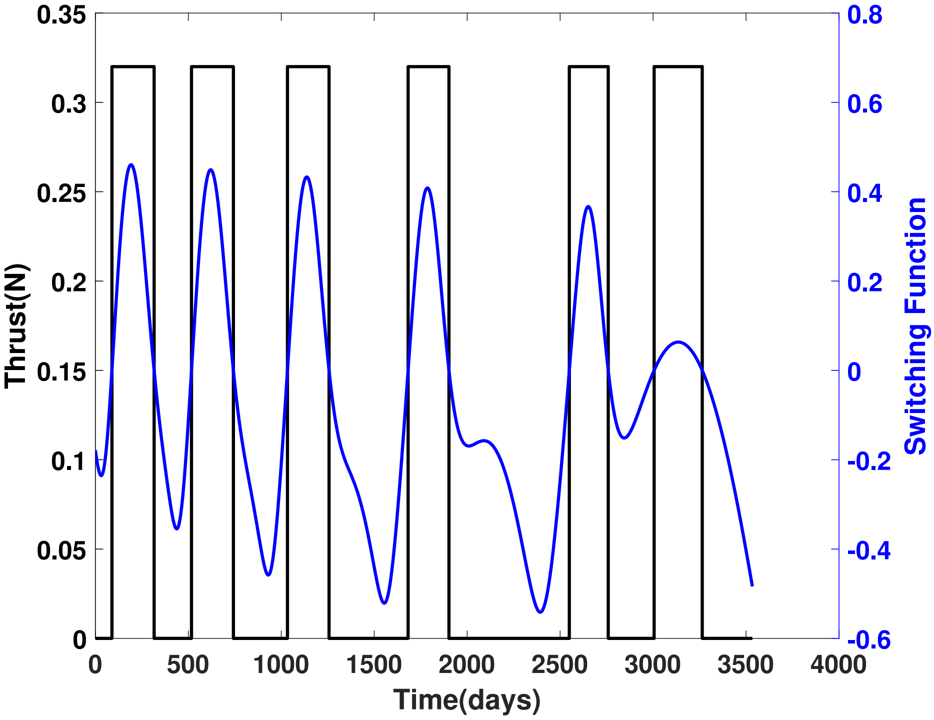


Figure 7.2: Earth-Dionysus problem: Thrust and switching function time histories for $\rho = 1.0 \times 10^{-5}$.

ranges considered (e.g., CS-4, CS-8 to CS-15). The error in final position and velocity for both the initialization methods using h-e formulation are in the same order.

From Table 7.1 the best initial costate vector for MEE formulation using the MACT method is CS-12 and using random initialization method is CS-f. These two initial costate vectors are then used to perform 50 simulations for a ρ value of 0.001 without using the continuation approach. It was observed that both the initialization methods were not able to converge to a solution and hence require some continuation approach (like the HTS method) to be followed. From Table 7.2 the best initial costate range for the h-e formulation using the MACT method is CS-15 and using the random initialization method is CS-f. Similar to the MEE formulation these two costate vectors are then used to perform 50 simulations for a ρ value of 0.001. The results are summarized in Table 7.3. Both the initialization methods have the same convergence performance. This also indicates that the h-e set of elements has a better performance than the set of MEEs since the latter has zero convergence when used without any continuation.

Table 7.3: Statistical comparison for h-e using MACT and random initialization methods with ρ = 0.001 for the Earth-Dionysus problem. (func. - function, PST - Per Simulation Time).

Range Chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-15	9	2718.33	44	305	51.69	2.14	4.53
CS-f	9	2718.33	44	301	15.83	6.59	3.57

7.2 GTO-GEO Problem without J_2 Perturbation

This is an orbit-rising problem from a Geostationary Transfer Orbit (GTO) to the Geostationary Earth Orbit (GEO) and is solved in [68]. The orbital elements for the initial and final orbit are given in Table 7.4. The initial true anomaly of the spacecraft in the GTO is taken as zero and the final true anomaly on the GEO is taken as 180 degrees. The spacecraft's initial and target position and velocity vectors in the Earth-centered frame are

$$\boldsymbol{r}(t_0) = [6378.9, 0.0, 0.0]^\top \text{km}, \qquad \boldsymbol{r}_T = [-42165, 0.0, 0.0]^\top \text{ km},$$

$$\boldsymbol{v}(t_0) = [0.0, 10.0258, 1.231]^\top \text{km/s}, \qquad \boldsymbol{v}_T = [0.0, -3.0746, 0.0]^\top \text{km/s}.$$

Table 7.4: Classic orbital elements for the GTO and GEO orbits.

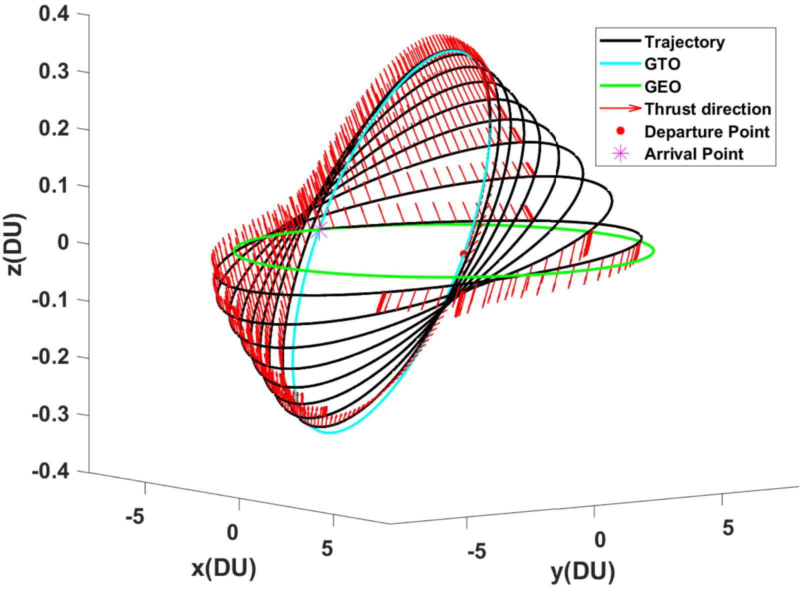
Orbit	a (km)	e	i (deg)	Ω (deg)	ω (deg)
GTO	24505	0.725	7.0	0.0	0.0
GEO	42165	0.0	0.0	–	–

The state values are converted into the MEE and the h-e element sets (see Table 8 in Ref [68]) in order to use them as boundary conditions for the MEE and h-e minimum-fuel formulations. The orbit-rising maneuver time is fixed to be 6 days with an initial fuel mass of $m_0 = 100$ kg. The spacecraft's low-thrust propulsion system has the following parameters: maximum thrust, $T_{\text{max}} = 0.5$ Newtons and specific impulse, $I_{\text{sp}} = 3100$ seconds. It is possible to consider an engine with variable specific impulse value and thrust value as is studied in [12], but, for simplicity, in this work, thrust and specific impulse values are constant along the trajectory. Similar to the interplanetary test case, Canonical units are used to scale the states and boundary conditions. For this problem, one distance unit (DU) is taken as one Earth radius

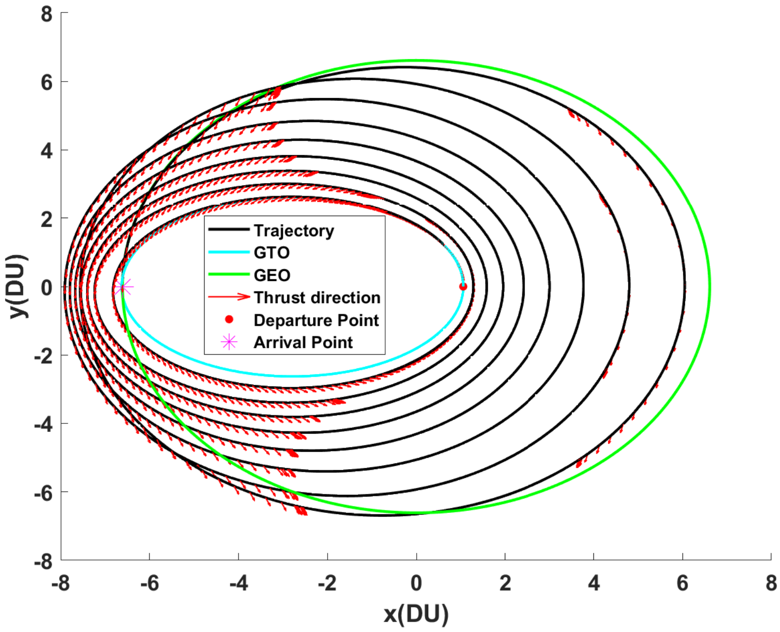
(6378 km) and one time unit (TU) is taken such that the gravitational parameter of the Earth becomes one (i.e., $\mu = 1$ DU^3/TU^2). The minimum-fuel solution for this problem corresponds to the trajectory that makes 8 orbital revolutions around the Earth and the optimal final mass is 94.155 kg (for $\rho = 0.03125$) as reported in [68]. Different views of the optimal trajectory are plotted in Figures 7.3a and 7.3b and the associated thrust and switching function time histories are plotted in Figure 7.4.

Table 7.5: Statistical comparison of the MACT and random initialization methods for the MEE formulation of GTO-GEO transfer with $\boldsymbol{\rho}$ = 1 (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-7}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	18	93.59	31	222	0.126	0.009	4.66
CS-a	1	93.59	53	334	0.029	0.002	11.24
CS-2	24	93.59	30	212	0.118	0.008	4.48
CS-b	5	93.59	67	405	0.092	0.007	16.19
CS-3	34	93.59	29	207	0.110	0.008	6.47
CS-c	6	93.59	51	325	0.041	0.003	7.19
CS-4	50	93.59	27	192	0.097	0.007	5.90
CS-d	8	93.59	39	250	58.30	4.100	7.33
CS-5	50	93.59	30	223	0.112	0.008	6.22
CS-e	18	93.59	37	241	0.149	0.011	6.06
CS-6	50	93.59	16	120	0.089	0.007	1.87
CS-f	22	93.59	32	223	0.095	0.007	5.01
CS-7	47	93.59	14	110	0.099	0.007	1.74
CS-g	8	93.59	29	199	0.078	0.006	4.38
CS-8	47	93.59	21	151	0.227	0.017	3.85
CS-h	4	93.59	35	234	0.108	0.008	4.17
CS-9	20	93.59	37	261	0.121	0.011	7.11
CS-i	17	93.59	28	195	0.119	0.009	8.59
CS-10	3	93.59	28	192	0.124	0.009	4.21
CS-j	9	93.59	37	259	0.188	0.0142	14.06
CS-11	0	-	-	-	-	-	-
CS-k	2	93.59	36	237	0.063	62.98	13.04
CS-12	0	-	-	-	-	-	-
CS-l	3	93.59	58	367	0.180	0.013	12.98
CS-13	0	-	-	-	-	-	-
CS-m	2	93.59	36	230	0.140	0.010	14.59
CS-14	0	-	-	-	-	-	-
CS-n	2	93.59	46	285	0.218	0.016	14.35
CS-15	0	-	-	-	-	-	-
CS-o	1	93.59	76	483	0.080	0.005	21.06



(a) Three-dimensional view of the optimal trajectory.



(b) Projection of the optimal trajectory onto the $x - y$ plane.

Figure 7.3: Minimum-fuel GTO-GEO problem: optimal trajectory of the transfer problem with $\rho = 1.0 \times 10^{-5}$.

Table 7.6: Statistical comparison of MACT and random initialization methods for the h-e formulation of GTO-GEO problem with $\boldsymbol{\rho}$ = 1 (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-7}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	0	-	-	-	-	-	-
CS-a	3	93.59	69	427	0.137	0.009	11.13
CS-2	0	-	-	-	-	-	-
CS-b	5	93.59	55	340	0.072	0.005	12.46
CS-3	0	-	-	-	-	-	-
CS-c	6	93.59	54	335	0.134	0.009	7.92
CS-4	0	-	-	-	-	-	-
CS-d	2	93.59	44	273	0.057	0.004	5.32
CS-5	5	93.59	30	210	0.161	0.011	5.38
CS-e	18	93.59	43	273	0.165	0.012	5.38
CS-6	2	93.59	38	260	0.162	0.018	4.89
CS-f	47	93.59	32	207	0.177	0.013	3.87
CS-7	15	93.59	35	240	0.143	0.010	6.45
CS-g	50	93.59	31	203	0.166	0.022	3.54
CS-8	49	93.59	15	113	0.014	0.009	1.83
CS-h	40	93.59	35	236	0.133	0.009	5.55
CS-9	50	93.59	16	119	0.139	0.010	1.79
CS-i	36	93.59	36	231	0.020	0.015	6.24
CS-10	50	93.59	18	128	0.127	0.009	2.26
CS-j	16	93.59	40	264	0.170	0.017	10.28
CS-11	50	93.59	32	218	0.161	0.012	4.54
CS-k	13	93.59	41	271	0.278	0.021	13.49
CS-12	50	93.59	28	191	0.121	0.008	5.79
CS-l	3	93.59	60	373	0.160	0.015	19.31
CS-13	50	93.59	29	202	0.171	0.012	7.36
CS-m	1	93.59	60	369	0.189	0.019	37.72
CS-14	41	93.59	33	232	0.260	0.018	8.25
CS-n	2	93.59	58	349	0.012	0.009	18.15
CS-15	31	93.59	33	230	0.149	0.011	8.48
CS-o	5	93.59	57	358	0.156	0.011	17.54

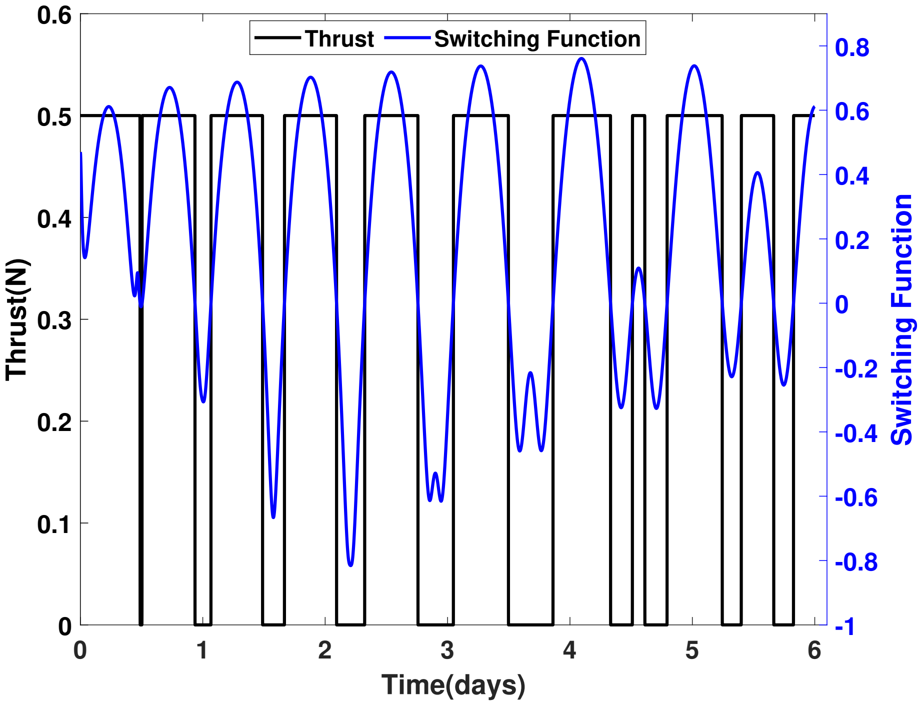


Figure 7.4: Minimum-fuel GTO-GEO problem: thrust and switching function time histories for $\rho = 1.0 \times 10^{-5}$.

The simulation results for the MEE formulation are summarized in Table 7.5 and the simulation results for the h-e formulation are summarized in Table 7.6. It can be seen from Table 7.5 that the MACT method completely outmatches the random initialization method for the MEE formulation by having a higher convergence percentage. Of the 750 simulations performed, random initialization method converged for only 108 simulations, but the MACT method converged for 325 simulations. The MACT method has three times more convergence percentage than the random initialization method. The simulation time, total iterations and function calls are also lower for the MACT method for most of the ranges for design variables. The error in final position and velocity is almost in the same order for both the costate initialization methods. However, random initialization has higher error for some costate vectors like CS-d and CS-k.

The MACT method also proved to be effective in the h-e formulation for the GTO-GEO transfer problem, and this is evident from the results summarized in Table 7.6. Of the 750 simulations performed random initialization method converged for 247 simulations but the MACT method converged for 393 simulations. The simulation time, number of iterations and function calls is much better for MACT method from CS-8 to CS-15. Using the h-e elements, the error in final position and velocity is almost in the same order for both the costate initialization methods.

Results summarized in Table 7.5 indicate that the best initial costate vector for the MACT method is CS-6 and for the random initialization method is CS-f. 50 simulations are then performed by taking the value of ρ directly as 0.001 and without following the continuation approach. It was observed that none of the simulations converged for both initialization methods.

The best initial costate vector for both methods is then chosen for the h-e formulation, and 50 simulations are performed for a ρ value of 0.001 without following the continuation approach. Table 7.7 summarizes the simulation results. The indirect ACT method did not converge to a solution, but the random initialization method converged for one simulation. Both the costate initialization method seems to have poor performance when the continuation approach is not used. However, it looks like random initialization has an edge over the MACT method since it converged for one simulation. It was observed that both of the costate initialization methods were not able to converge to a solution. Thus, it is important to start the continuation process using a relatively large value of ρ (e.g., $\rho = 1$ or $\rho = 0.5$).

Table 7.7: Statistical comparison for h-e using the MACT and random initialization methods with $\boldsymbol{\rho}$ = 0.001 for GTO-GEO transfer (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-7}$	PST (s)
CS-9	0	-	-	-	-	-	-
CS-g	1	94.15	78	485	0.991	0.724	9.07

7.3 GTO-GEO Problem with J_2 Perturbation

The GTO-GEO trajectories are also generated by considering the J_2 perturbation. The value for the J_2 coefficient is considered as 1082.63×10^{-6} in this thesis. In this case, the optimal final mass is 94.145 kg. The optimal trajectory profile and thrust time history are visually identical to the case when the J_2 perturbation is not included and hence they are not shown here. The simulation results for the MEE formulation are given in Table 7.8 and for the h-e formulation are given in Table 7.9. Table 7.8 strongly suggests that the MACT method performed better than the random initialization method when the value of ρ is taken as 1. Random initialization of costates performed better than the MACT method only for two initial costate vectors: CS-j and CS-m. Of the 750 simulations performed MACT method converged for 337 simulations and random initialization method converged for 108 simulations. The MACT method has an overall better performance in terms of the simulation time, number of iterations and function calls.

Table 7.8: Statistical comparison of the MACT and random initialization methods for the MEE formulation of the GTO-GEO problem with $\boldsymbol{\rho} = 1$ and J_2 perturbation considered (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-7}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	12	93.58	34	244	0.866	0.006	8.31
CS-a	2	93.58	50	317	0.120	0.008	11.76
CS-2	21	93.58	31	227	0.120	0.008	6.53
CS-b	5	93.58	57	361	0.061	0.004	16.39
CS-3	35	93.58	30	216	2.800	0.385	5.92
CS-c	8	93.58	47	305	0.095	0.005	9.63
CS-4	50	93.58	27	191	0.124	0.009	6.41
CS-d	11	93.58	44	284	0.111	0.008	12.12
CS-5	50	93.58	29	212	0.130	0.009	5.46
CS-e	20	93.58	45	282	0.122	0.008	11.68
CS-6	49	93.58	17	127	0.085	0.006	2.67
CS-f	20	93.58	29	200	0.094	0.006	5.17
CS-7	49	93.58	15	118	0.117	0.015	2.49
CS-g	9	93.58	25	171	0.068	0.005	4.83
CS-8	48	93.58	21	150	0.099	0.007	3.35
CS-h	2	93.58	45	316	0.147	0.010	8.80
CS-9	23	93.58	31	224	0.073	0.005	8.09
CS-i	19	93.58	31	213	0.119	0.008	9.46
CS-10	0	-	-	-	-	-	-
CS-j	9	93.58	48	323	0.151	0.011	23.24
CS-13	0	-	-	-	-	-	-
CS-m	2	93.58	46	292	0.789	0.005	33.49

Next, the best initial costate vector using random initialization and MACT method is determined from Table 7.8 and 50 simulations are performed for a ρ value of 0.001 without using the continuation approach. The simulation results are summarized in Table 7.10. For the MACT

Table 7.9: Statistical comparison of the MACT and random initialization methods for the h-e formulation of the GTO-GEO problem with $\boldsymbol{\rho}$ = 1 and J_2 perturbation considered (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-7}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-10}$	PST (s)
CS-1	0	-	-	-	-	-	-
CS-a	1	93.58	71	436	0.191	0.014	30.47
CS-2	0	-	-	-	-	-	-
CS-b	5	93.58	48	302	0.102	0.007	12.53
CS-3	0	-	-	-	-	-	-
CS-c	4	93.58	47	293	0.228	0.016	20.82
CS-4	0	-	-	-	-	-	-
CS-d	8	93.58	44	277	0.187	0.013	14.35
CS-5	0	-	-	-	-	-	-
CS-e	19	93.58	52	324	0.105	0.008	20.06
CS-6	3	93.58	44	309	0.079	0.006	13.54
CS-f	47	93.58	33	210	0.143	0.010	8.26
CS-7	21	93.58	39	263	0.099	0.007	12.62
CS-g	50	93.58	27	178	0.156	0.011	6.93
CS-8	50	93.58	15	115	0.136	0.009	3.29
CS-h	33	93.58	37	243	0.136	0.009	17.06
CS-9	50	93.58	14	109	0.118	0.008	3.10
CS-i	24	93.58	44	293	1.37	0.116	39.97
CS-10	50	93.58	17	126	0.171	0.012	3.76
CS-j	8	93.58	48	313	0.195	0.021	32.25
CS-11	50	93.58	35	234	0.128	0.009	13.18
CS-k	2	93.58	33	227	0.156	0.011	26.39
CS-12	50	93.58	30	209	0.132	0.009	29.86
CS-l	4	93.58	55	341	0.220	0.015	46.30
CS-13	49	93.58	30	213	0.180	0.0130	10.29
CS-m	1	93.58	53	327	0.254	0.018	36.38
CS-14	45	93.58	34	240	0.130	0.009	11.95
CS-n	1	93.58	79	500	0.329	0.023	49.39
CS-15	33	93.58	35	247	0.222	0.016	11.69
CS-o	1	93.58	46	313	0.063	0.005	116.99

method, CS-7 is chosen as the best initial costate vector, although it does not have 100% convergence (like CS-4 and CS-5) because it has a far lower number of iterations and function calls. The best range for the random initialization method is taken as CS-f on the basis of the results given in Table 7.8. Table 7.10 shows that the random initialization method has a better convergence (however not a large difference in convergence) for a ρ value of 0.001. The MACT method resulted in zero convergence, and this implies that the MACT method for the MEE formulation needs a continuation approach to be followed to obtain a solution for lower values of ρ.

Table 7.10: Statistical comparison for MEE using MACT and random initialization methods with $\boldsymbol{\rho}$ = 0.001 for the GTO-GEO problem with J2 perturbation considered. (func. - function, PST - Per Simulation Time).

Costate vector chosen	Total cases converged	m_f (kg)	Total iterations	Total func. calls	$\Delta \boldsymbol{r}_f$ (km) $\times 10^{-3}$	$\Delta \boldsymbol{v}_f$ (km/s) $\times 10^{-7}$	PST (s)
CS-7	0	-	-	-	-	-	-
CS-f	2	94.14	45	294	0.130	0.095	11.85

The simulation results for the h-e formulation are summarized in Table 7.9, which indicate that random initialization performed better than the MACT method from CS-1 to CS-7 and beyond CS-7, the MACT method performed well. Of the 750 simulations, the MACT method converged for 401 simulations and the random initialization method converged for 208 cases. Similar to previous simulation results, the MACT method requires less simulation time, number of iterations and function calls when compared to the random initialization method. As done before, the best initial Costate vectors are chosen as CS-9 and CS-g for the MACT and random initialization method respectively.

A summary of the results for all the simulations is given in Table 7.11.

Table 7.11: Summary of convergence percentage of the problems simulated using the MACT and random initialization methods (conv. - convergence, RI - Random Initialization).

Test Problem	MEE set		h-e set	
	MACT conv. (%)	RI conv. (%)	MACT conv. (%)	RI conv. (%)
Earth-Dionysus	67.33	36.00	68.67	33.47
GTO-GEO w/o J_2	44.93	14.40	53.47	27.73
GTO-GEO w/J_2	43.33	14.44	43.33	14.40

7.4 GTO-GEO Problem with higher number of revolutions

The total number of orbital revolutions in the test problems of the previous sections is less than ten. In order to compare the utility of the MACT method with the random initialization method for more complex problems, the GTO-GEO Problem with J_2 perturbation is modified by increasing the total time of flight of the maneuver. The maximum thrust T_{max} is reduced to 0.45 Newtons. The maximum number of function evaluations and iterations for MATLAB's `fsolve` is set to 750 and 300 respectively. All other spacecraft parameters, boundary conditions and the manner in which different initial costates are found are identical to that of the problem considered in Section 7.3. Since the time of flight is increased, the spacecraft will make more revolutions around the Earth. The transfer times that are considered include 10, 12, 15, 17 and 20 days and the corresponding optimal revolutions that must be made around Earth are 15, 18, 23, 26 and 30, respectively. The projection of the optimal trajectory onto the $x - y$ plane when the number of revolutions is taken as 30 in Figure 7.5. The convergence summary for the MACT and random initialization methods for the MEE formulation is given in Table 7.12.

Table 7.12: Summary of convergence of the MACT and random initialization methods for the MEE formulation and for different number of revolutions around the Earth and different times of flight.

Nrev	t_f (days)	MACT Convergence (%)	Rand init. Convergence (%)
15	10	25.87	3.47
18	12	11.6	0.93
23	15	4.27	0
26	17	2.53	0
30	20	0.53	0

Numerical results indicate an improvement in the percentage of convergence of the resulting BVPs by a factor of 2 or 3 for the considered test cases. Results also indicate that the MACT method is suitable for solving low-thrust minimum-fuel trajectory optimization problems that involve a higher number (up to 30 in case of the considered GTO-GEO problem) even without considering any homotopy approach on the number of revolutions. The poor performance of the random initialization method is compensated by the ease of its implementation. However,

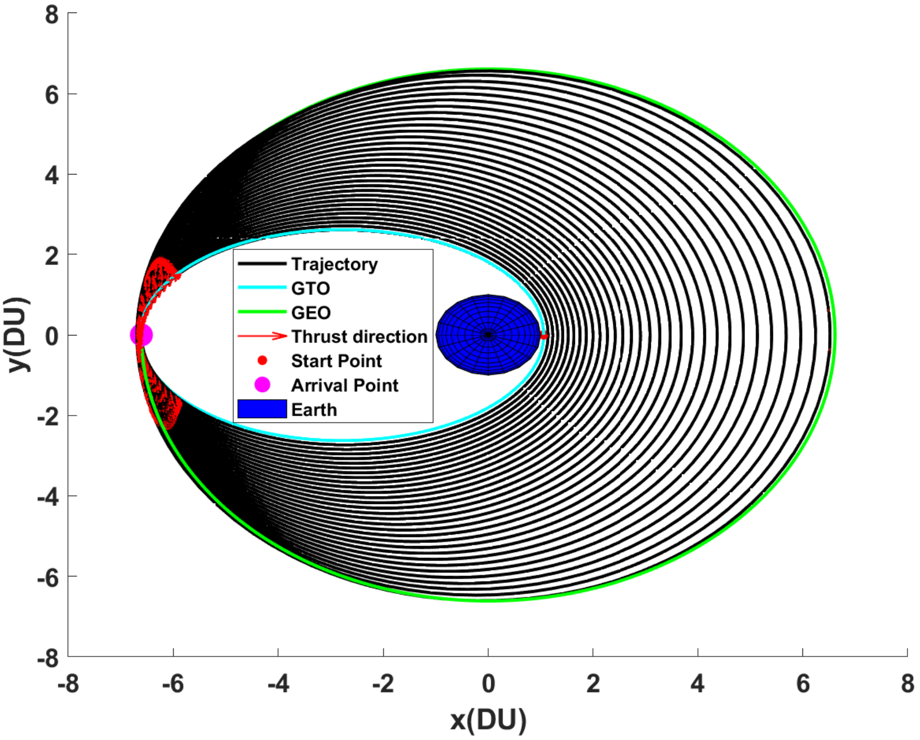


Figure 7.5: Projection of the Optimal trajectory of the minimum-fuel GTO-GEO transfer problem with 30 revolutions onto the $x - y$ plane.

the results indicate that as the complexity of the problem increases either due to longer time of flight and more number of orbital revolutions, a proper strategy for costates initialization becomes more distinguished.

Chapter 8

Concluding Remarks

The results from Chapter 5 indicate that when the costates are normalized to lie on a unit hypersphere, poor convergence is achieved if a Quasi-Newton based solver is used to find the ensuing OCPs. Also, the set of MEEs demonstrated a far superior convergence compared to the case where the set of Cartesian coordinates was used to formulate the optimal control problems. For the test case considered, it was found that MEE with random initialization of costates has the best performance, followed by MEE with the costates mapped to a hypersphere. The worst convergence is obtained by Cartesian coordinates with costates mapping.

In Ref. Chapter 6, the ACT method originally developed for solving trajectory optimization problems using Cartesian coordinates is extended to different sets of coordinates or elements with the help of costate vector mapping theorem. In Ref. Chapter 7, the percent of convergence of the MACT is compared against the standard random initialization method over a wide domain of initial design variables to remove any possible bias towards the MACT or the random initialization methods. Application of the MACT method is demonstrated by solving two classes of problems, i.e., interplanetary and planet-centric low-thrust minimum-fuel trajectory optimization problems. Numerical results indicate an improvement in the percentage of convergence in solving the resulting boundary-value problems by a factor of 2 or 3 for the considered test cases when the MACT method is used. This extension is advantageous since it is possible to solve trajectory optimization problems while leveraging the features offered by the ACT method.

Results support the usability of the MACT method compared to the random initialization method for more complex planet-centric maneuvers involving many orbital revolutions.

Results also indicate that the set that consists of the elements of the specific angular momentum and eccentricity (h-e) and the set of modified equinoctial elements (MEEs) have similar convergence. But the h-e element set performed better than the MEE set for the GTO-GEO transfer problem (without J_2 perturbation). The results indicate that the h-e element set can also be considered as a contender while designing low-thrust trajectories. Our future work will investigate the utility of the MACT method for solving more complex trajectory optimization problems from orbits around the Earth to orbits around the Moon and multiphase interplanetary trajectories.

Chapter 9

Future Work

A potential future work is to investigate the utility of the MACT method for solving more complex trajectory optimization problems. For instance, the convergence of a TPBVP problem becomes difficult when a three-body dynamical model is considered [84, 19]. The interest in transfers between Earth and Low Lunar Orbits has increased because of NASA's Gateway program. It will be worth finding such transfers using the indirect method, as they offer finer thrust (and potentially more optimal) solutions. The main challenge in finding optimal trajectories for the Gateway program is due to the low acceleration associated with the Gateway main propulsion module. The maximum acceleration the Gateway is capable of providing is $2.904 \times 10^{-5} m/s^2$ [85], considering its mass and thrust capabilities.

It will also be interesting to investigate the application of the MACT method to spiral trajectories, as they are essential for transfers like Low Earth Orbit (LEO) to Low Mars Orbit (LMO) [86]. Also, problems involving round-trip transfers between Earth-Mars and Earth-Jupiter can be considered [17].

References

[1] Marc D Rayman and Steven N Williams. Design of the first interplanetary solar electric propulsion mission. *Journal of Spacecraft and Rockets*, 39(4):589–595, 2002.

[2] Joakim Kugelberg, Per Bodin, Staffan Persson, and Peter Rathsman. Accommodating electric propulsion on smart-1. *Acta Astronautica*, 55(2):121–130, 2004.

[3] Marc D Rayman, Thomas C Fraschetti, Carol A Raymond, and Christopher T Russell. Dawn: A mission in development for exploration of main belt asteroids vesta and ceres. *Acta Astronautica*, 58(11):605–616, 2006.

[4] Yuichi Tsuda, Makoto Yoshikawa, Masanao Abe, Hiroyuki Minamino, and Satoru Nakazawa. System design of the hayabusa 2—asteroid sample return mission to 1999 JU3. *Acta Astronautica*, 91:356–362, 2013.

[5] John S Snyder, Vernon H Chaplin, Dan M Goebel, Richard R Hofer, Alejandro Lopez Ortega, Ioannis G Mikellides, Taylor Kerl, Giovanni Lenguito, Faraz Aghazadeh, and Ian Johnson. Electric propulsion for the psyche mission: Development activities and status. In *AIAA Propulsion and Energy 2020 Forum*, page 3607, 2020.

[6] David H Manzella and Kurt Hack. High-power solar electric propulsion for future nasa missions. In *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, page 3718, 2014.

[7] Tara Polsgrove, Larry Kos, Randall Hopkins, and Tracie Crane. Comparison of performance predictions for new low-thrust trajectory tools. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6742, 2006.

[8] Monika Auweter-Kurtz and Helmut Kurtz. Optimization of electric thrusters for primary propulsion based on the rocket equation. *Journal of propulsion and power*, 19(3):413–423, 2003.

[9] Damon Landau, James Chase, Thomas Randolph, Paul Timmerman, and David Oh. Electric propulsion system selection process for interplanetary missions. *Journal of Spacecraft and Rockets*, 48(3):467–476, 2011.

[10] Donald H Ellison, Bruce A Conway, Jacob A Englander, and Martin T Ozimek. Application and analysis of bounded-impulse trajectory models with analytic gradients. *Journal of Guidance, Control, and Dynamics*, 41(8):1700–1714, 2018.

[11] Ehsan Taheri, John L Junkins, Ilya Kolmanovsky, and Anouck Girard. A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 2. *Acta Astronautica*, 172:166–179, 2020.

[12] Ehsan Taheri, John L Junkins, Ilya Kolmanovsky, and Anouck Girard. A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 1. *Acta Astronautica*, 172:151–165, 2020.

[13] VG Petukhov, AV Ivanyukhin, and Woo Sang Wook. Joint optimization of control and main trajectory and design parameters of an interplanetary spacecraft with an electric propulsion system. *Cosmic Research*, 57(3):188–203, 2019.

[14] Jasmine Rimani, Christopher Andrea Paissoni, Nicole Viola, Giorgio Saccoccia, and Jose Gonzalez del Amo. Multidisciplinary mission and system design tool for a reusable electric propulsion space tug. *Acta Astronautica*, 175:387–395, 2020.

[15] Jacob A Englander, Matthew A Vavrina, Lucy F Lim, Lucy A McFadden, Alyssa R Rhoden, and Keith S Noll. Trajectory optimization for missions to small bodies with a focus on scientific merit. *Computing in science & engineering*, 19(4):18–28, 2017.

[16] Lorenzo Casalino and Guido Colasurdo. Optimization of variable-specific-impulse interplanetary trajectories. *Journal of guidance, control, and dynamics*, 27(4):678–684, 2004.

[17] Chris L Ranieri and Cesar A Ocampo. Optimization of roundtrip, time-constrained, finite burn trajectories via an indirect method. *Journal of Guidance, Control, and Dynamics*, 28(2):306–314, 2005.

[18] Fanghua Jiang, Hexi Baoyin, and Junfeng Li. Practical techniques for low-thrust trajectory optimization with homotopic approach. *Journal of Guidance, Control, and Dynamics*, 35(1):245–258, 2012.

[19] Hong-Xin Shen, Lorenzo Casalino, and Hai-Yang Li. Adjoints estimation methods for impulsive moon-to-earth trajectories in the restricted three-body problem. *Optimal Control Applications and Methods*, 36(4):463–474, 2015.

[20] David Morante, Manuel Sanjurjo Rivo, and Manuel Soler. Multi-objective low-thrust interplanetary trajectory optimization based on generalized logarithmic spirals. *Journal of Guidance, Control, and Dynamics*, 42(3):476–490, 2019.

[21] Sean W Napier, Jay W McMahon, and Jacob A Englander. A multi-objective, multi-agent transcription for the global optimization of interplanetary trajectories. *The Journal of the Astronautical Sciences*, 67(4):1271–1299, 2020.

[22] Nicholas Nurre and Ehsan Taheri. Multiple gravity-assist low-thrust trajectory design using finite Fourier series. In *AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 20–671*, 2020.

[23] Nicholas Nurre and Ehsan Taheri. Hybrid evolutionary-indirect multiple-gravity-assist low-thrust interplanetary trajectory design. In *31st AAS/AIAA Space Flight Mechanics Meeting, Virtual, February 1-February 3, AAS Paper 21–388*, 2021.

[24] Runqi Chai, Al Savvaris, Antonios Tsourdos, Senchun Chai, and Yuanqing Xia. Optimal fuel consumption finite-thrust orbital hopping of aeroassisted spacecraft. *Aerospace Science and Technology*, 75:172–182, 2018.

[25] Emmanuel Trélat. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154(3):713–758, 2012.

[26] Craig A Kluever and B Conway. Low-thrust trajectory optimization using orbital averaging and control parameterization. *Spacecraft Trajectory Optimization*, pages 112–138, 2010.

[27] David Jimenez-Lluva and Bart Root. Hybrid optimization of low-thrust many-revolution trajectories with coasting arcs and longitude targeting for propellant minimization. *Acta Astronautica*, 177:232–245, 2020.

[28] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[29] Runqi Chai, Al Savvaris, Antonios Tsourdos, Senchun Chai, and Yuanqing Xia. A review of optimization techniques in spacecraft flight trajectory design. *Progress in aerospace sciences*, 109:100543, 2019.

[30] Praveen Jawaharlal Ayyanathan and Ehsan Taheri. Mapped adjoint control transformation method for low-thrust trajectory design. *Acta Astronautica*, 193:418–431, 2022.

[31] Praveen Jawaharlal Ayyanathan and Ehsan Taheri. Adjoint control transformation for modified equinoctial elements: An indirect approach. In *2021 AAS/AIAA Astrodynamics Specialist Conference, August 9-August 11, 2021, AAS Paper 21–706*, 2021.

[32] Praveen Jawaharlal Ayyanathan and Ehsan Taheri. Comparison of costate initialization techniques for fuel-optimal low-thrust trajectory. In *31st AAS/AIAA Space Flight Mechanics Meeting, Virtual, February 1-February 3, AAS Paper 21–385*, 2021.

[33] James M Longuski, José J Guzmán, and John E Prussing. *Optimal control with aerospace applications*. Springer, 2014.

[34] Kshitij Mall, Michael J Grant, and Ehsan Taheri. Uniform trigonometrization method for optimal control problems with control and state constraints. *Journal of Spacecraft and Rockets*, 57(5):995–1007, 2020.

[35] Dario Izzo and Ekin Öztürk. Real-time guidance for low-thrust transfers using deep neural networks. *Journal of Guidance, Control, and Dynamics*, 44(2):315–327, 2021.

[36] Vishala Arya, Ehsan Taheri, and John Junkins. Electric thruster mode-pruning strategies for trajectory-propulsion co-optimization. *Aerospace Science and Technology*, 116:106828, 2021.

[37] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[38] Arthur Bryson and Yu-Chi Ho. Applied optimal control: Optimization, estimation, and control (revised edition). *Levittown, Pennsylvania: Taylor & Francis*, 1975.

[39] Sandeep Singh, John Junkins, Brian Anderson, and Ehsan Taheri. Eclipse-conscious transfer to lunar gateway using ephemeris-driven terminal coast arcs. *Journal of Guidance, Control, and Dynamics*, pages 1–17, 2021.

[40] Thomas Haberkorn, Pierre Martinon, and Joseph Gergaud. Low thrust minimum-fuel orbital transfer: a homotopic approach. *Journal of Guidance, Control, and Dynamics*, 27(6):1046–1060, 2004.

[41] Binfeng Pan, Xun Pan, and Ping Lu. Finding best solution in low-thrust trajectory optimization by two-phase homotopy. *Journal of Spacecraft and Rockets*, 56(1):283–291, 2019.

[42] Vyacheslav G. Petukhov. Optimization of multi-orbit transfers between noncoplanar elliptic orbits. *Cosmic Research*, 42(3):250–268, 2004.

[43] Vishala Arya, Ehsan Taheri, and John L Junkins. A composite framework for co-optimization of spacecraft trajectory and propulsion system. *Acta Astronautica*, 178:773–782, 2020.

[44] Régis Bertrand and Richard Epenoy. New smoothing techniques for solving bang–bang optimal control problems—numerical results and statistical interpretation. *Optimal Control Applications and Methods*, 23(4):171–197, 2002.

[45] Mirco Rasotto, Roberto Armellin, and Pierluigi Di Lizia. Multi-step optimization strategy for fuel-optimal orbital transfer of low-thrust spacecraft. *Engineering Optimization*, 48(3):519–542, 2016.

[46] Ehsan Taheri and John L Junkins. Generic smoothing for optimal bang-off-bang spacecraft maneuvers. *Journal of Guidance, Control, and Dynamics*, 41(11):2470–2475, 2018.

[47] Taibo Li, Zhaokui Wang, and Yulin Zhang. Double-homotopy technique for fuel optimization of power-limited interplanetary trajectories. *Astrophysics and Space Science*, 364(9):1–12, 2019.

[48] Xun Pan and Binfeng Pan. Practical homotopy methods for finding the best minimum-fuel transfer in the circular restricted three-body problem. *IEEE Access*, 8:47845–47862, 2020.

[49] Kshitij Mall and Ehsan Taheri. Unified trigonometrization method for solving optimal control problems in atmospheric flight mechanics. In *AIAA Scitech 2020 Forum*, page 0022, 2020.

[50] Sandeep K Singh, Brian D Anderson, Ehsan Taheri, and John L Junkins. Exploiting manifolds of l1 halo orbits for end-to-end earth–moon low-thrust trajectory design. *Acta Astronautica*, 183:255–272, 2021.

[51] R. Bertrand and R. Epenoy. New smoothing techniques for solving bang-bang optimal control problems - numerical results and statistical interpretation. *Optimal Control Applications and Methods*, 23(4):171–197, July 2002.

[52] Binfeng Pan, Xun Pan, and Yangyang Ma. A quadratic homotopy method for fuel-optimal low-thrust trajectory design. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(5):1741–1757, 2019.

[53] Xun Pan, Binfeng Pan, and Zenghao Li. Bounding homotopy method for minimum-time low-thrust transfer in the circular restricted three-body problem. *The Journal of the Astronautical Sciences*, 67:1220–1248, 2020.

[54] Jingrui Zhang, Qian Xiao, and Lincheng Li. Solution space exploration of low-thrust minimum-time trajectory optimization by combining two homotopies. *Automatica*, 148:110798, 2023.

[55] Chen Zhang, Francesco Topputo, Franco Bernelli-Zazzera, and Yu-Shan Zhao. Low-thrust minimum-fuel optimization in the circular restricted three-body problem. *Journal of Guidance, Control, and Dynamics*, 38(8):1501–1510, 2015.

[56] James D Thorne and Christopher D Hall. Approximate initial lagrange costates for continuous-thrust spacecraft. *Journal of Guidance, Control, and Dynamics*, 19(2):283–288, 1996.

[57] Max Cerf, Thomas Haberkorn, and Emmanuel Trélat. Continuation from a flat to a round earth model in the coplanar orbit transfer problem. *Optimal Control Applications and Methods*, 33(6):654–675, 2012.

[58] LCW Dixon and Michael C Bartholomew-Biggs. Adjoint—control transformations for solving practical optimal control problems. *Optimal Control Applications and Methods*, 2(4):365–381, 1981.

[59] Juan Senent, Cesar Ocampo, and Antonio Capella. Low-thrust variable-specific-impulse transfers and guidance to unstable periodic orbits. *Journal of Guidance, Control, and Dynamics*, 28(2):280–290, 2005.

[60] Ryan P Russell. Primer vector theory applied to global low-thrust trade studies. *Journal of Guidance, Control, and Dynamics*, 30(2):460–472, 2007.

[61] MT Ozimek and KC Howell. Low-thrust transfers in the earth-moon system, including applications to libration point orbits. *Journal of Guidance, Control, and Dynamics*, 33(2):533–549, 2010.

[62] Donghun Lee and Hyochoong Bang. Efficient initial costates estimation for optimal spiral orbit transfer trajectories design. *Journal of Guidance, Control, and Dynamics*, 32(6):1943–1947, 2009.

[63] Hans Seywald and Renjith R Kumar. Method for automatic costate calculation. *Journal of Guidance, Control, and Dynamics*, 19(6):1252–1261, 1996.

[64] Hui Yan and Hongxin Wu. Initial adjoint-variable guess technique and its application in optimal orbital transfer. *Journal of guidance, control, and dynamics*, 22(3):490–492, 1999.

[65] Ehsan Taheri, Nan I Li, and Ilya Kolmanovsky. Co-state initialization for the minimum-time low-thrust trajectory optimization. *Advances in Space Research*, 59(9):2360–2373, 2017.

[66] Ehsan Taheri and Ossama Abdelkhalik. Initial three-dimensional low-thrust trajectory design. *Advances in Space Research*, 57(3):889–903, 2016.

[67] Ehsan Taheri, Ilya Kolmanovsky, and Ella Atkins. Enhanced smoothing technique for indirect optimization of minimum-fuel low-thrust trajectories. *Journal of Guidance, Control, and Dynamics*, 39(11):2500–2511, 2016.

[68] John L Junkins and Ehsan Taheri. Exploration of alternative state vector choices for low-thrust trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 42(1):47–64, 2019.

[69] Yu Wang, Chao Han, and Xiucong Sun. Optimization of low-thrust earth-orbit transfers using the vectorial orbital elements. *Aerospace Science and Technology*, 112:106614, 2021.

[70] MJH Walker, B Ireland, and Joyce Owens. A set modified equinoctial orbit elements. *Celestial mechanics*, 36(4):409–419, 1985.

[71] Roger A Broucke and Paul J Cefola. On the equinoctial orbit elements. *Celestial mechanics*, 5(3):303–310, 1972.

[72] Ehsan Taheri and John L Junkins. How many impulses redux. *The Journal of the Astronautical Sciences*, pages 1–78, 2019.

[73] Paul Cefola. Equinoctial orbit elements-application to artificial satellite orbits. In *Astrodynamics Conference*, page 937, 1972.

[74] Jean Albert Kechichian. The treatment of the earth oblateness effect in trajectory optimization in equinoctial coordinates. *Acta astronautica*, 40(1):69–82, 1997.

[75] Suwat Sreesawet and Atri Dutta. Fast and robust computation of low-thrust orbit-raising trajectories. *Journal of Guidance, Control, and Dynamics*, 41(9):1888–1905, 2018.

[76] Pierre Martinon and Joseph Gergaud. Using switching detection and variational equations for the shooting method. *Optimal Control Applications and Methods*, 28(2):95–116, 2007.

[77] Bruce A Conway. *Spacecraft trajectory optimization*, volume 29. Cambridge University Press, 2010.

[78] Karl-Heinz Glassmeier, Hermann Boehnhardt, Detlef Koschny, Ekkehard Kührt, and Ingo Richter. The rosetta mission: flying towards the origin of the solar system. *Space Science Reviews*, 128(1-4):1–21, 2007.

[79] Ehsan Taheri. Composite smooth control method for low-thrust trajectory design: Variable specific impulse engine. In *AIAA Scitech 2020 Forum*, page 2184, 2020.

[80] Donald H Ellison, Bruce A Conway, Jacob A Englander, and Martin T Ozimek. Analytic gradient computation for bounded-impulse trajectory models using two-sided shooting. *Journal of Guidance, Control, and Dynamics*, 41(7):1449–1462, 2018.

[81] Ehsan Taheri and John L Junkins. How many impulses redux. *The Journal of the Astronautical Sciences*, pages 1–78, 2019.

[82] Jacob Williams, Juan S Senent, Cesar Ocampo, Ravi Mathur, and Elizabeth C Davis. Overview and software architecture of the copernicus trajectory design and optimization system. In *4th International Conference on Astrodynamics Tools and Techniques, Madrid, Spain*, 2010.

[83] Ehsan Taheri, Vishala Arya, and John L Junkins. Costate mapping for indirect trajectory optimization. *Astrodynamics (2021).*

[84] Hong-Xin Shen and Lorenzo Casalino. Indirect optimization of three-dimensional multiple-impulse moon-to-earth transfers. *The Journal of the Astronautical Sciences*, 61:255–274, 2014.

[85] Robert E Pritchett. *Strategies for low-thrust transfer design based on direct collocation techniques*. PhD thesis, Purdue University Graduate School, 2020.

[86] Christopher L Ranieri and Cesar A Ocampo. Indirect optimization of spiral trajectories. *Journal of Guidance, Control, and Dynamics*, 29(6):1360–1366, 2006.

[87] David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.

Appendices

Appendix A

Costate mapping from Cartesian coordinates to the MEE set

This section provides an example for mapping the costates from Cartesian coordinates to the set of MEEs. Let $\boldsymbol{x}_{\text{MEE}} = [p, f, g, h, k, l]^\top$ and $\boldsymbol{x}_{\text{Cart}} = [\boldsymbol{r}^\top, \boldsymbol{v}^\top]^\top$ denote the MEE and Cartesian state vectors, respectively. Let $\alpha = h^2 - k^2$, $w = 1 + f\cos(l) + g\sin(l)$, $s^2 = 1 + h^2 + k^2$ and $M = \frac{1}{s^2\sqrt{p}}$. The Cartesian coordinates can be written in terms of MEEs as

$$\boldsymbol{x}_{\text{Cart}} = \begin{bmatrix} \frac{p}{ws^2}\{\cos(l) + \alpha\cos(l) + 2hk\sin(l)\} \\ \frac{p}{ws^2}\{\sin(l) - \alpha\sin(l) + 2hk\cos(l)\} \\ 2\frac{p}{ws^2}(h\sin(l) - k\cos(l)) \\ -M\{\sin(l) + \alpha\sin(l) - 2hk\cos(l) + g - 2fhk + \alpha g\} \\ -M\{-\cos(l) + \alpha\cos(l) + 2hk\sin(l) - f + 2ghk + \alpha f\} \\ 2M\{h\cos(l) + k\sin(l) + fh + gk\} \end{bmatrix} = \boldsymbol{\varrho}(\boldsymbol{x}_{\text{MEE}}).$$

By following Eq. (6.9), the costates from the Cartesian coordinates ($\boldsymbol{\lambda}_{\text{cart}}$) can be mapped to the costates of the set of MEEs ($\boldsymbol{\lambda}_{\text{MEE}}$) as

$$\boldsymbol{\lambda}_{\text{Cart}}(t)^\top \left[\frac{\partial\boldsymbol{\varrho}(\boldsymbol{x}_{\text{MEE}})}{\partial\boldsymbol{x}_{\text{MEE}}}\right]\Bigg|_t = \boldsymbol{\lambda}_{\text{MEE}}(t)^\top. \tag{A.1}$$

The required Jacobian matrix, $(\partial\boldsymbol{\varrho}/\partial\boldsymbol{x}_{\text{MEE}})$, is calculated symbolically. As an example, for the interplanetary test problem and using the Canonical units, the optimal costate vectors (of a converged case) for the Cartesian coordinates ($\boldsymbol{\lambda}_{\text{Cart}}$) for $\rho = 1$ is given in the first column of Table (A.1). By applying the costate vector theorem, the initial costates (at $t = 0$) in the

Cartesian coordinates can be mapped to the costates of the MEE set, and the mapped costates ($\boldsymbol{\lambda}_{\text{MEE}}^{\text{mapped}}$) are given in the second column of Table A.1.

Table A.1: Comparison of the mapped costate vector with the optimal costate vector for the MEE set of Earth-Dionysus rendezvous problem.

$\boldsymbol{\lambda}_{\text{Cart}}$	$\boldsymbol{\lambda}_{\text{MEE}}^{\text{mapped}}$
$\begin{bmatrix} 178.451232047246 \\ -2064.16011615921 \\ -349.575926331475 \\ 1353.58251261986 \\ -84.6061809237475 \\ -638.174235536129 \end{bmatrix}$	$\begin{bmatrix} -1347.90134742511 \\ -134.085754847598 \\ 646.824779833027 \\ -651.160791168936 \\ -1313.94659559597 \\ -0.491214765547389 \end{bmatrix}$

Appendix B

MEE Differential Equations Derivation

The set of MEEs $[p, f, g, h, k, L]^\top$ can be written in terms of COEs $[a, e, i, \omega, \Omega, \nu]^\top$ in the following manner,

$$p = a(1 - e^2),\ f = e\cos(\omega + \Omega),\ g = e\sin(\omega + \Omega), \tag{B.1}$$

$$h = \tan\left(\frac{i}{2}\right)\cos(\Omega),\ k = \tan\left(\frac{i}{2}\right)\sin(\Omega),\ L = \Omega + \omega + \nu, \tag{B.2}$$

where 'a' is the semi-major axis, 'e' is the eccentricity, 'ω' is the argument of periapsis, 'Ω' is the right ascension of the ascending node, 'i' is the inclination, 'ν' is true anomaly, 'p' is semi-latus rectum and 'L' is the true longitude. Gauss variational equations are given by [87],

$$\dot{a} = \frac{2a^2}{H}\left(e\sin(f)a_r + \frac{p}{r}a_\theta\right),\ \dot{e} = \frac{1}{H}(p\sin(f)a_r + ((p + r)\cos(f) + re)a_\theta), \tag{B.3}$$

$$\dot{i} = \frac{r\cos(\theta)}{H}a_h,\ \dot{\Omega} = \frac{r\sin(\theta)}{H\sin(i)}a_h, \tag{B.4}$$

$$\dot{\omega} = \frac{1}{He}(-p\cos(f)a_r + (p + r)\sin(f)a_\theta) - \frac{r\sin(\theta)\cos(i)}{H\sin(i)}a_h, \tag{B.5}$$

where a_r, a_h and a_θ are the external acceleration along the radial direction, orbit normal direction and perpendicular to both radial and orbit normal direction, H is the specific angular momentum, and

$$\theta = \omega + \nu. \tag{B.6}$$

The COEs can be written in terms of the MEEs in the following manner,

$$e = \sqrt{f^2 + g^2},\ a = \frac{p}{1 - e^2} = \frac{p}{1 - (f^2 + g^2)}, \tag{B.7}$$

$$\tan\left(\frac{i}{2}\right) = \sqrt{h^2 + k^2},\ i = 2\tan^{-1}(\sqrt{h^2 + k^2}), \tag{B.8}$$

$$\cos(\Omega) = \frac{h}{\sqrt{h^2 + k^2}},\ \sin(\Omega) = \frac{k}{\sqrt{h^2 + k^2}}. \tag{B.9}$$

Below are some useful equations that are used throughout the derivations for simplifying the math,

$$s^2 = 1 + h^2 + k^2, \tag{B.10}$$

$$w = \frac{p}{r}, \tag{B.11}$$

$$\frac{w + 1}{w} = \frac{p + r}{p}, \tag{B.12}$$

$$r = \frac{p}{1 + e\cos(\nu)}, \tag{B.13}$$

$$H = \sqrt{p\mu}. \tag{B.14}$$

Eq. (B.8) can be rewritten as,

$$\tan\left(\frac{i}{2}\right) = \sqrt{s^2 - 1}, \tag{B.15}$$

$$\sec^2\left(\frac{i}{2}\right) = 1 + \tan^2\left(\frac{i}{2}\right) = s^2, \tag{B.16}$$

$$\sin(i) = \frac{2\tan\left(\frac{i}{2}\right)}{\sec^2\left(\frac{i}{2}\right)} = \frac{2\sqrt{h^2 + k^2}}{s^2}, \tag{B.17}$$

$$\frac{\tan\left(\frac{i}{2}\right)}{\sin(i)} = \frac{1}{2}\sec^2\left(\frac{i}{2}\right), \tag{B.18}$$

$$\frac{1 - \cos(i)}{\sin(i)} = \tan\left(\frac{i}{2}\right). \tag{B.19}$$

Below are some useful identities that are used throughout the derivations,

$$
\begin{aligned}
\cos(\nu) &= \cos(L-(\omega+\Omega)) \\
&= \cos(L)\cos(\omega+\Omega)+\sin(L)\sin(\omega+\Omega) \\
&= \frac{f}{e}\cos(L)+\frac{g}{e}\sin(L),
\end{aligned} \tag{B.20}
$$

$$
\begin{aligned}
\sin(\nu) &= \sin(L-(\omega+\Omega)) \\
&= \sin(L)\cos(\omega+\Omega)-\cos(L)\sin(\omega+\Omega) \\
&= \frac{f}{e}\sin(L)-\frac{g}{e}\cos(L).
\end{aligned} \tag{B.21}
$$

Eqs. (B.20) and (B.21) use Eq. (B.1) to rewrite $\cos(\omega+\Omega)$ and $\sin(\omega+\Omega)$ as

$$\cos(\theta) = \cos(\omega+\nu) = \cos(\omega)\cos(\nu)-\sin(\omega)\sin(\nu) \tag{B.22}$$

$$= \frac{\cos(\omega)}{e}(f\cos(L)+g\sin(L))-\frac{\sin(\omega)}{e}(f\sin(L)-g\cos(L)), \tag{B.23}$$

$$\sin(\theta) = \sin(\omega+\nu) = \sin(\omega)\cos(\nu)+\cos(\omega)\sin(\nu) \tag{B.24}$$

$$= \frac{\sin(\omega)}{e}(f\cos(L)+g\sin(L))+\frac{\cos(\omega)}{e}(f\sin(L)-g\cos(L)), \tag{B.25}$$

$$\sin(\theta) = \sin(L-\omega) = \sin(L)\cos(\Omega)+\cos(L)\sin(\Omega) \tag{B.26}$$

$$= \frac{h\sin(i)-k\cos(L)}{\tan\left(\frac{i}{2}\right)}. \tag{B.27}$$

Eqs. (B.23) and (B.25) use Eqs. (B.20) and (B.21) to rewrite $\cos(\nu)$ and $\sin(\nu)$.

B.1 Time derivative of p

With the above equations, it is possible to derive the Gauss variational equations for MEEs. Going order-wise, the time derivative of Eq. (B.1) can be written as (by using the chain rule),

$$\dot{p} = \dot{a}(1-e^2)-2ae\dot{e}. \tag{B.28}$$

Substituting for e, a, $\dot{a}$ $\dot{e}$ from Eqs. (B.7) and (B.3) respectively, we get,

$$\dot{p} = \frac{2a^2}{H}(1-e^2)\left(e\sin(\nu)a_r + \frac{p}{r}a_\theta\right) - \frac{2ae}{H}\left(p\sin(\nu)a_r + ((p+r)\cos(\nu) + re)a_\theta\right). \quad \text{(B.29)}$$

Collecting out the terms along a_r direction we get,

$$\frac{2a^2}{H}(1-e^2)e\sin(\nu)a_r - \frac{2ae}{H}p\sin(\nu)a_r. \quad \text{(B.30)}$$

Now upon using Eq. (B.1) we get,

$$\frac{2ae}{H}p\sin(\nu)a_r - \frac{2ae}{H}p\sin(\nu)a_r = 0. \quad \text{(B.31)}$$

Therefore, there are no contributions from the acceleration along the a_r direction. Next, by considering the terms along a_θ direction we can simplify $\dot{p}$ as

$$\dot{p} = \frac{2a^2}{H}(1-e^2)\frac{p}{r}a_\theta - \frac{2ae}{H}((p+r)\cos(\nu) + re)a_\theta, \quad \text{(B.32)}$$

$$\dot{p} = \frac{2ap^2}{Hr}a_\theta - \frac{2ae}{H}((p+r)\cos(\nu) + re)a_\theta, \quad \text{(B.33)}$$

$$\dot{p} = \frac{2a}{H}\left(\frac{p^2}{r} - e\left((p+r)\cos(\nu) + re\right)\right)a_\theta, \quad \text{(B.34)}$$

$$\dot{p} = \frac{2ap}{Hr}\left(p - \left(er\cos(\nu) + \frac{r^2e}{p}\cos(\nu) + \frac{r^2e^2}{p}\right)\right)a_\theta. \quad \text{(B.35)}$$

Using equation (B.13) and evaluating we get,

$$\dot{p} = \frac{2apr}{Hr}\left(\frac{p}{r} - \left(e\cos(\nu) + \frac{e\cos(\nu)}{1+e\cos(\nu)} + \frac{e^2}{1+e\cos(\nu)}\right)\right)a_\theta, \quad \text{(B.36)}$$

$$\dot{p} = \frac{2ap}{H}\left(1 + e\cos(\nu) - \left(\frac{e^2\cos^2(\nu) + 2e\cos(\nu) + e^2}{1+e\cos(\nu)}\right)\right)a_\theta, \quad \text{(B.37)}$$

$$\dot{p} = \frac{2ap}{H}\left(\frac{1 + e^2\cos^2(\nu) + 2e\cos(\nu) - e^2\cos^2(\nu) - 2e\cos(\nu) - e^2}{1+e\cos(\nu)}\right)a_\theta, \quad \text{(B.38)}$$

$$\dot{p} = \frac{2p}{H}\left(\frac{a(1-e^2)}{1+e\cos(\nu)}\right)a_\theta, \quad \text{(B.39)}$$

$$\dot{p} = \frac{2p}{H}\left(\frac{p}{1+e\cos(\nu)}\right)a_\theta. \quad \text{(B.40)}$$

By using Eqs. (B.13), (B.14) and (B.11) we get,

$$\dot{p} = \frac{2pr}{\sqrt{p\mu}} a_\theta, \tag{B.41}$$

$$\dot{p} = \frac{2p}{w}\sqrt{\frac{p}{\mu}} a_\theta. \tag{B.42}$$

B.2 Time derivative of k

The time derivative of k, see Eq. (B.2), can be written as,

$$\dot{k} = \frac{1}{2}\sec^2\left(\frac{i}{2}\right)\dot{i}\ \sin(\Omega) + \tan\left(\frac{i}{2}\right)\cos(\Omega)\dot{\Omega}, \tag{B.43}$$

$$\dot{k} = \frac{1}{2}\sec^2\left(\frac{i}{2}\right)\frac{r\cos(\theta)}{H}\sin(\Omega)a_h + \tan\left(\frac{i}{2}\right)\cos(\Omega)\frac{r\sin(\theta)}{H\sin(i)}a_h. \tag{B.44}$$

Using Eq. (B.18), we can write the above expression as,

$$\dot{k} = \frac{r}{2H}\sec^2\left(\frac{i}{2}\right)(\cos(\theta)\sin(\Omega) + \sin(\theta)\cos(\Omega))a_h, \tag{B.45}$$

Using equations (3.3), (B.11) and (B.16) we can further simplify as

$$\dot{k} = \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\sin(\theta + \Omega)a_h, \tag{B.46}$$

$$\dot{k} = \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\sin(L)a_h. \tag{B.47}$$

B.3 Time derivative of h

The time derivative of h, see Eq. (B.2), can be written as,

$$\dot{h} = \frac{1}{2}\sec^2\left(\frac{i}{2}\right)\dot{i}\cos(\Omega) - \tan\left(\frac{i}{2}\right)\sin(\Omega)\dot{\Omega}, \tag{B.48}$$

$$\dot{h} = \frac{1}{2}\sec^2\left(\frac{i}{2}\right)\frac{r\cos(\theta)}{H}\cos(\Omega)a_h - \tan\left(\frac{i}{2}\right)\sin(\Omega)\frac{r\sin(\theta)}{H\sin(i)}a_h. \tag{B.49}$$

Using equation (B.18), we can write the above expression as

$$\dot{h} = \frac{r}{2H} \sec^2\left(\frac{i}{2}\right) (\cos(\theta)\cos(\Omega) - \sin(\theta)\sin(\Omega))a_h, \tag{B.50}$$

Using equations (B.11), (B.14) and (B.16) we get,

$$\dot{h} = \sqrt{\frac{p}{\mu}} \frac{s^2}{2w} \cos(\theta + \Omega)a_h, \tag{B.51}$$

$$\dot{h} = \sqrt{\frac{p}{\mu}} \frac{s^2}{2w} \cos(L)a_h. \tag{B.52}$$

B.4 Time derivative of g

The time derivative of g, see Eq. (B.1), can be written as,

$$\dot{g} = \dot{e}\sin(\omega + \Omega) + e\cos(\omega + \Omega)(\dot{\omega} + \dot{\Omega}). \tag{B.53}$$

Substituting for $\dot{e}$, $\dot{\omega}$ and $\dot{\Omega}$ we get

$$\begin{aligned} \dot{g} =& \frac{1}{H}(p\sin(\nu)a_r + ((p+r)\cos(\nu) + re)a_\theta)\sin(\omega + \Omega) + \\ & e\cos(\omega + \Omega)\left(\frac{1}{He}(-p\cos(\nu)a_r + (p+r)\sin(\nu)a_\theta) - \right. \\ & \frac{r\sin(\theta)\cos(i)}{H\sin(i)}a_h + \frac{r\sin(\theta)}{H\sin(i)}a_h. \end{aligned} \tag{B.54}$$

By considering the terms along a_r direction, $\dot{g}_{ar}$, we get,

$$\dot{g}_{ar} = -\frac{p}{H}(\cos(\omega + \Omega)\cos(\nu) - \sin(\omega + \Omega)\sin(\nu))a_r, \tag{B.55}$$

$$\dot{g}_{ar} = -\frac{p}{H}\cos(L)a_r, \tag{B.56}$$

$$\dot{g}_{ar} = -\sqrt{\frac{p}{\mu}}\cos(L)a_r. \tag{B.57}$$

Next consider the terms along a_θ direction, $\dot{g}_{a\theta}$,

$$\dot{g}_{a\theta} = \frac{1}{H}((p+r)\cos(\nu) + re)\sin(\omega+\Omega)a_\theta + \frac{1}{H}\cos(\omega+\Omega)(p+r)\sin(\nu)a_\theta. \tag{B.58}$$

By multiplying and expanding the terms we get

$$\begin{aligned}\dot{g}_{a\theta} =& \frac{1}{H}(p(\cos(\nu)\sin(\omega+\Omega) + \cos(\omega+\Omega)\sin(\nu)) + r(\cos(\nu)\sin(\omega+\Omega) + \\ & \cos(\omega+\Omega)\sin(\nu)) + re\sin(\omega+\Omega))a_\theta.\end{aligned} \tag{B.59}$$

By using trigonometric identities, we can further simplify the above equation as

$$\dot{g}_{a\theta} = \frac{1}{H}(p\sin(L) + r\sin(L) + rg)a_\theta, \tag{B.60}$$

$$\dot{g}_{a\theta} = \frac{p}{\sqrt{\mu p}}\left(\frac{\sin(L)(p+r) + rg}{p}\right)a_\theta. \tag{B.61}$$

Finally, using Eqs. (B.11) and (B.12) we get,

$$\dot{g}_{a\theta} = \sqrt{\frac{p}{\mu}}\left(\frac{(w+1)\sin(L) + g}{w}\right)a_\theta. \tag{B.62}$$

Next, consider the terms along the a_h direction, $\dot{g}_{ah}$,

$$\dot{g}_{ah} = e\cos(\omega+\Omega)\left(\frac{r\sin(\theta)\cos(i)}{H\sin(i)} + \frac{r\sin(\theta)}{H\sin(i)}\right)a_h, \tag{B.63}$$

$$\dot{g}_{ah} = \frac{re}{H\sin(i)}\cos(\omega+\Omega)(1-\cos(i))\sin(\theta)a_h. \tag{B.64}$$

By using Eqs. (B.1), (B.19) and (B.25) we get,

$$\dot{g}_{ah} = \frac{re}{H}\frac{f}{e}\tan\left(\frac{i}{2}\right)\frac{h\sin(L) - k\cos(L)}{\tan\left(\frac{i}{2}\right)}a_h, \tag{B.65}$$

$$\dot{g}_{ah} = \frac{rf}{\sqrt{p\mu}}(h\sin(L) - k\cos(L))a_h. \tag{B.66}$$

Rewriting the above expression using Eq. (B.11), we get

$$\dot{g}_{ah} = \sqrt{\frac{p}{\mu}}\left(\frac{f(h\sin(L) - k\cos(L))}{w}\right)a_h. \tag{B.67}$$

By using Eqs. (B.57), (B.62) and (B.67) we can write

$$\begin{aligned}\dot{g} =&\sqrt{\frac{p}{\mu}}\left(-\cos(L)a_r + \left(\frac{(w+1)\sin(L)+g}{w}\right)a_\theta\right) + \\ &\sqrt{\frac{p}{\mu}}\left(\frac{f(h\sin(L) - k\cos(L))}{w}\right)a_h.\end{aligned} \tag{B.68}$$

B.5 Time derivative of f

The time derivative of Eq. (B.1) can be written as

$$\dot{f} = \dot{e}\cos(\omega+\Omega) - e\sin(\omega+\Omega)(\dot{\omega}+\dot{\Omega}). \tag{B.69}$$

Substituting for $\dot{e}$, $\dot{\omega}$ and $\dot{\Omega}$ we get

$$\begin{aligned}\dot{f} =&\frac{1}{H}(p\sin(\nu)a_r + ((p+r)\cos(\nu)+re)a_\theta)\cos(\omega+\Omega) - \\ &e\sin(\omega+\Omega)\left(\frac{1}{He}\left(-p\cos(\nu)a_r + (p+r)\sin(\nu)a_\theta\right) - \right. \\ &\frac{r\sin(\theta)\cos(i)}{H\sin(i)}a_h + \frac{r\sin(\theta)}{H\sin(i)}a_h.\end{aligned} \tag{B.70}$$

By considering the terms along a_r direction, $\dot{f}_{ar}$, we get,

$$\dot{f}_{ar} = \frac{p}{H}(\cos(\omega+\Omega)\sin(\nu) - \sin(\omega+\Omega)\cos(\nu))a_r, \tag{B.71}$$

$$\dot{f}_{ar} = \frac{p}{H}\sin(L)a_r, \tag{B.72}$$

$$\dot{f}_{ar} = \sqrt{\frac{p}{\mu}}\sin(L)a_r. \tag{B.73}$$

By considering the terms along a_θ direction, $\dot{f}_{a\theta}$, we get,

$$\dot{f}_{a\theta} = \frac{1}{H}((p + r)\cos(\nu) + re)\cos(\omega + \Omega)a_\theta - \frac{1}{H}\sin(\omega + \Omega)(p + r)\sin(\nu)a_\theta. \tag{B.74}$$

By multiplying and expanding the terms we get

$$\begin{aligned}\dot{f}_{a\theta} =& \frac{1}{H}(p(\cos(\nu)\cos(\omega + \Omega) - \sin(\omega + \Omega)\sin(\nu))+ \\ & r(\cos(\nu)\cos(\omega + \Omega) - \sin(\omega + \Omega)\sin(\nu)) + re\cos(\omega + \Omega))a_\theta.\end{aligned} \tag{B.75}$$

By using trigonometric identities, we can further simplify the above equation as

$$\dot{f}_{a\theta} = \frac{1}{H}(p\cos(L) + r\cos(L) + rf)a_\theta, \tag{B.76}$$

$$\dot{f}_{a\theta} = \frac{p}{\sqrt{\mu p}}\left(\frac{\cos(L)(p + r) + rf}{p}\right)a_\theta, \tag{B.77}$$

Using Eqs. (B.11) and (B.12), we get,

$$\dot{f}_{a\theta} = \sqrt{\frac{p}{\mu}}\left(\frac{(w + 1)\cos(L) + f}{w}\right)a_\theta. \tag{B.78}$$

By considering the terms along a_h direction, $\dot{f}_{ah}$, we get

$$\dot{f}_{ah} = e\sin(\omega + \Omega)\left(\frac{r\sin(\theta)\cos(i)}{H\sin(i)} - \frac{r\sin(\theta)}{H\sin(i)}\right)a_h, \tag{B.79}$$

$$\dot{f}_{ah} = \frac{re}{H\sin(i)}\cos(\omega + \Omega)(1 - \cos(i))\sin(\theta)a_h, \tag{B.80}$$

by using Eqs. (B.1), (B.19) and (B.25), we simplify the above equation as

$$\dot{f}_{ah} = \frac{re}{H}\frac{g}{e}\tan\left(\frac{i}{2}\right)\frac{h\sin(L) - k\cos(L)}{\tan\left(\frac{i}{2}\right)}a_h, \tag{B.81}$$

$$\dot{f}_{ah} = \frac{rg}{\sqrt{p\mu}}(h\sin(L) - k\cos(L))a_h. \tag{B.82}$$

Rewriting the above expression using Eq. (B.11), we get

$$\dot{f}_{ah} = \sqrt{\frac{p}{\mu}} \left(\frac{g(h\sin(L) - k\cos(L))}{w} \right) a_h. \tag{B.83}$$

Finally by using equations (B.73), (B.78) and (B.83) we get,

$$\begin{aligned} \dot{f} =& \sqrt{\frac{p}{\mu}} \left(\sin(L) a_r + \left(\frac{(w+1)\cos(L) + f}{w} \right) a_\theta \right) + \\ & \sqrt{\frac{p}{\mu}} \left(\frac{g(h\sin(L) - k\cos(L))}{w} \right) a_h. \end{aligned} \tag{B.84}$$

B.6 Time derivative of L

The time derivative of L (see Eq. (B.2)) can be written as

$$\dot{L} = \dot{\omega} + \dot{\Omega} + \dot{\nu}, \tag{B.85}$$

$$\dot{L} = \dot{\omega} + \dot{\Omega} + \frac{\sqrt{\mu p}}{r^2} - (\dot{\omega} + \dot{\Omega}\cos(i)), \tag{B.86}$$

$$\dot{L} = \dot{\Omega}(1 - \cos(i)) + \frac{\sqrt{\mu p}}{r^2}, \tag{B.87}$$

$$\dot{L} = \frac{r\sin(\theta)}{H\sin(i)}(1 - \cos(i)) a_h + \frac{\sqrt{\mu p}}{r^2}. \tag{B.88}$$

Using Eqs. (B.19) and (B.27), we can simplify the above equation as

$$\dot{L} = \frac{r}{H} \frac{h\sin(l) - k\cos(L)}{\tan\left(\frac{i}{2}\right)} \tan\left(\frac{i}{2}\right) a_h + \frac{\sqrt{\mu p}}{r^2}, \tag{B.89}$$

$$\dot{L} = \frac{\sqrt{\mu p}}{r^2} + \frac{r}{H}(h\sin(L) - k\cos(L)) a_h, \tag{B.90}$$

$$\dot{L} = \sqrt{\mu p} \left(\frac{w}{p} \right)^2 + \sqrt{\frac{p}{\mu}} \left(\frac{h\sin(L) - k\cos(L)}{w} \right) a_h. \tag{B.91}$$