NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

LOW-THRUST REACHABILITY ANALYSIS BY ANALYTICAL MEANS

by

Daniel A. Gregorio

June 2023

Thesis Advisor: Jennifer Hudson
Co-Advisor: Gary Thomason

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 2023	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE LOW-THRUST REACHABILITY ANALYSIS BY ANALYTICAL MEANS	5. FUNDING NUMBERS
6. AUTHOR(S) Daniel A. Gregorio	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

This thesis presents a novel algorithm supporting orbit determination of noncooperative, low-thrust maneuvering spacecraft. The algorithm, as a generic procedure, guides an analyst through a defined process and returns data to enable further analytic understanding. The algorithm consists of a basic guidance scheme for maneuvering the spacecraft from one orbit to another and estimates the likely costs in terms of time or ΔV. Pontryagin's maximum principle is employed to test optimality and Gauss's variational equations of motion are employed in a restricted two-body dynamics model. This work examines methods to extend the applicability of historical efforts through the use of scaling factors to reduce error across the sample space. A novel approximation approach is developed to universalize computational strategies for estimating the costs of changing the semimajor axis of a satellite. As a result, this procedure can be implemented through any available programming language and does not require the use of advanced mathematical libraries, maximizing interoperability with existing mission-management tools and processes. As an example, this algorithmic process can be used to support a reachability assessment of a noncooperative spacecraft in cislunar space.

14. SUBJECT TERMS cislunar, cislunar, orbit determination, space domain awareness, trajectory optimization, low thrust, continuous thrust	15. NUMBER OF PAGES 201
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

LOW-THRUST REACHABILITY ANALYSIS BY ANALYTICAL MEANS

Daniel A. Gregorio
Major, United States Army
BS, United States Military Academy, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
June 2023

Approved by: Jennifer Hudson
Advisor

Gary Thomason
Co-Advisor

James H. Newman
Chair, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis presents a novel algorithm supporting orbit determination of noncooperative, low-thrust maneuvering spacecraft. The algorithm, as a generic procedure, guides an analyst through a defined process and returns data to enable further analytic understanding. The algorithm consists of a basic guidance scheme for maneuvering the spacecraft from one orbit to another and estimates the likely costs in terms of time or ΔV. Pontryagin's maximum principle is employed to test optimality and Gauss's variational equations of motion are employed in a restricted two-body dynamics model. This work examines methods to extend the applicability of historical efforts through the use of scaling factors to reduce error across the sample space. A novel approximation approach is developed to universalize computational strategies for estimating the costs of changing the semimajor axis of a satellite. As a result, this procedure can be implemented through any available programming language and does not require the use of advanced mathematical libraries, maximizing interoperability with existing mission-management tools and processes. As an example, this algorithmic process can be used to support a reachability assessment of a noncooperative spacecraft in cislunar space.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction and Purpose **1**
1.1 Motivation 3
1.2 Overview and Scope 5
1.3 Organization 11
1.4 Orientation 13

2 Methodology **17**
2.1 Background 18
2.2 System Bounding and Assumptions 20
2.3 Model Construction and Assessment 27

3 Optimality of Core Analytical Solutions **47**
3.1 Background 47
3.2 Introduction to Numerical Methods 48
3.3 Optimal Low-Thrust Lunar Orbit Transfers 50
3.4 Problem Family 1 – In-Plane, Low-Lunar Transfers 51
3.5 Problem Family 2 – Out-of-Plane, Low-Lunar Transfers – General Problem Formulation 70
3.6 Analysis of Optimal Control Results 82

4 Analytical Estimates and Regional Accuracy **83**
4.1 Background 83
4.2 Primary Estimates 84
4.3 Out-of-plane Maneuver Estimates 91
4.4 Additional Estimates 95
4.5 A Novel Elementary Implementation for the Estimation of $\Delta\overline{a}$ 96

5 A Generalized Algorithm to Estimate Low-Thrust Maneuvers **99**
5.1 Low-Energy Moon-Earth Transfer Reachability 99

6 Conclusion **113**
6.1 Continued Research Recommendations 113
6.2 Hindsight is 20/20 115
6.3 Generative AI and Its Use in this Research 116

Appendix A Full Size Figure 1.2 **117**

Appendix B Supporting Elements for Chapter 2 **119**
B.1 ODE Solver Evaluation Files 119

Appendix C Supporting Elements for Chapter 3 **121**
C.1 Necessary Equations 121

Appendix D Supporting Elements for Chapter 4 **125**
D.1 Accuracy and Cost Plots for $\Delta\overline{a}_{MAX}$ Estimates 125
D.2 Accuracy and Cost Plots for $\Delta\overline{e}$ Estimates 126
D.3 Accuracy and Cost Plots for $\Delta\overline{i}$ Estimates 128

Appendix E MATLAB Code to Replicate Findings **131**
E.1 Analytical Estimates 137
E.2 Integration Functions 141
E.3 Analysis Scripts 163
E.4 Major Evaluations of Optimality 174

List of References **175**

Initial Distribution List **181**

List of Figures

Figure 1.1 The Low-Thrust Low-Energy Moon-Earth Transfer of Liu et al. Source: [1]. 2

Figure 1.2 Highest Order Problem System. Adapted from [2]. 7

Figure 1.3 Highest Order Problem System Spacecraft Operator Perspective. Adapted from [2]. 9

Figure 1.4 Highest Order Problem System Space Domain Awareness Analyst Perspective. 10

Figure 2.1 Cislunar Distances. Source: [3]. 18

Figure 2.2 Acceleration Vector Description. Source: [4]. 21

Figure 2.3 Elliptical Restricted Four Body Problem Model. Source: [5]. . . . 22

Figure 2.4 Initial Problem System 25

Figure 2.5 Poor Ordinary Differential Equation Solutions for Elliptical Cases and ode45 with Default Settings 42

Figure 2.6 Extreme Tolerance Settings with ode45 to Force a Smooth Path . 43

Figure 2.7 Acceptable Solutions from a Purpose Built Solver, ode15s 44

Figure 2.8 Acceptable Solutions from a Purpose Built Solver, ode23s 45

Figure 2.9 A Good Control History Simulated through a Coarse Solver . . . 46

Figure 3.1 Stalking Horse Path and Control for Family 1 54

Figure 3.2 Unaltered State and Control Histories from DIDO's Problem 1.A. Candidate Solution 59

Figure 3.3 Feasibility Test Results from DIDO's Problem 1.A. Candidate Solution 60

Figure 3.4 Interpolated Results from DIDO's Problem 1.A. Candidate Solution and the Stalking Horse Solution 61

Figure 3.5 Adherence to Minimized Hamiltonian Behavior for Problem 1.A. 61

Figure 3.6 Adherence to Karush-Kuhn-Tucker Conditions for Problem 1.A. . 62

Figure 3.7 Feasibility of DIDO's Solution for Problem 1.B. 64

Figure 3.8 Minimized Hamiltonian of DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions 64

Figure 3.9 Karush-Kuhn-Tucker Conditions Obeyed in DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions 65

Figure 3.10 Confirmation of Costate λ_a's Performance at t_f in DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions 65

Figure 3.11 DIDO's Constrained Solution for Problem 1.C with Basic Interpolation 67

Figure 3.12 Feasibility of DIDO's Solution for Problem 1.C. 68

Figure 3.13 Karush-Kuhn-Tucker Conditions Obeyed in DIDO's Solution for Problem 1.C. Demonstrating Necessary Conditions 68

Figure 3.14 Terminal Transversality of the Endpoint Lagrangian from DIDO's Solution for Problem 1.C. 69

Figure 3.15 Minimized Hamiltonian from DIDO's Solution for Problem 1.C. 69

Figure 3.16 Stalking Horse Path and Control for Family 2 70

Figure 3.17 Comparison of α from the Maximum Δa Strategy and β from the Maximum Δi Strategy 71

Figure 3.18 Infeasible and Non-optimal Low-node (40 Nodes) Result from DIDO for Maneuver 2.A 75

Figure 3.19 Poor Scaling and Sensitivity Yield a Feasible but Non-optimal Moderate-node (66 Nodes) Result from DIDO for Maneuver 2.A 76

Figure 3.20 Adherence to Karush-Kuhn-Tucker Conditions for Sub-optimal but Feasible solution for Maneuver 2.A 76

Figure 3.21 Feasibl but Non-optimal Low-node (40 node) Result from DIDO for Maneuver 2.B 79

Figure 3.22 Karush-Kuhn-Tucker Verification for Maneuver 2.B 79

Figure 3.23 Feasibility of the Overconstrained Controls for Maneuver 2.2 . . 81

Figure 3.24 Feasibility of the Overconstrained Controls for Maneuver 2.2 . . 82

Figure 4.1 Sample State [km] and Control History for the $\Delta\overline{a}_{MAX}$ Strategy . 85

Figure 4.2 Changes to Classical Orbital Elements over Time for a Sample $\Delta\overline{a}_{MAX}$ strategy. 86

Figure 4.3 Sample $\delta\Delta V$ and δ in Computational Time Plots for the $\Delta\overline{a}_{MAX}$ Strategy 88

Figure 4.4 Sample State [km] and Control History for the $\Delta\overline{e}$ Strategy . . . 88

Figure 4.5 Changes in Classical Orbital Elements over Time for a Sample $\Delta\overline{e}$ Strategy. 90

Figure 4.6 Sample $\delta\Delta V$ plots for the $\Delta\overline{e}$ strategy 91

Figure 4.7 Sample State [km] and Control History for the $\Delta\overline{i}$ Strategy . . . 92

Figure 4.8 Changes in Classical Orbital Elements over Time for a Sample $\Delta\overline{i}$ Strategy. 93

Figure 4.9 Critical $\delta\Delta V$ Plots for Understanding the Limitations of the $\Delta\overline{i}$ Strategy 94

Figure 4.10 $\delta\Delta V$ Plots Demonstrating the Interdependence of the $\Delta\overline{i}$ Strategy and ω 94

Figure 4.11 $f(\overline{e})$ of a 8th Order Rational Taylor Series and Elliptic Integrals with $e_0 = 0.3$ 97

Figure 4.12 Taylor Series as a Replacement for Elliptic Integrals 98

Figure 5.1 State Changes for ΔV_1 the $\Delta\overline{a}$. 102

Figure 5.2 State Changes for ΔV_2 the $\Delta\overline{i}$. 103

Figure 5.3 State Changes for ΔV_3 the $\Delta\overline{e}$. 104

Figure 5.4 State Changes for the One-step Transfer. 105

Figure 5.5 State History for a Direct, Single-step Transfer. 106

Figure 5.6 State History for a Direct, Single-step Transfer Aligned with Target Orbit. 107

Figure 5.7 State History for a Two-step Transfer, Aligned with Target Orbit. 108

Figure 5.8 State Changes for the Two-step Transfer. 110

Figure D.1 Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates . . . 125

Figure D.2 Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates . . . 125

Figure D.3 Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates . . . 126

Figure D.4 Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates . . . 126

Figure D.5 Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates . . . 127

Figure D.6 Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates . . . 127

Figure D.7 Sample Accuracy and Computational Costs of $\Delta\overline{i}$ Estimates . . . 128

Figure D.8 Sample Accuracy and Computational Costs of $\Delta\overline{i}$ Estimates . . . 128

Figure D.9 Sample Accuracy and Computational Costs of $\Delta\overline{i}$ Estimates . . . 129

List of Tables

Table 2.1 Dynamics Parameters 31

Table 2.2 Known Maximum Paths 32

Table 2.3 Various Solver Solution Results for Test 1 36

Table 2.4 $ode45_d$ Derived Control Simulation Results for Test 1 37

Table 2.5 $ode45_{ht}$ Derived Control Simulation Results for Test 1 37

Table 2.6 $ode15s$ Derived Control Simulation Results for Test 1 37

Table 2.7 $ode23s$ Derived Control Simulation Results for Test 1 38

Table 2.8 Various Solver Solution Results Time for Test 2 39

Table 2.9 $ode45_d$ Derived Control Simulation Results for Test 2 40

Table 2.10 $ode45_{ht}$ Derived Control Simulation Results for Test 2 40

Table 2.11 $ode15s$ Derived Control Simulation Results for Test 2 41

Table 2.12 $ode23s$ Derived Control Simulation Results for Test 2 41

Table 4.1 Known Maximum Paths 84

Table 5.1 ΔV & ToF for the Direct Single-Step Transfer 106

Table 5.2 ΔV & ToF for the Direct Two-Step Transfer 109

Table 6.1 Equinoctial Elements and Sources 115

Table 6.2 Modified Equinoctial Elements and Sources 116

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

Symbol	Meaning	Units	Chapter Introduced
ΔV	Change in Velocity	km/s^2	Chapter 1
ΔV_{AN}	Change in Velocity Analytically Calculated	km/s^2	Chapter 1
ΔV_{NUM}	Change in Velocity Numerically Calculated	km/s^2	Chapter 1
ToF	Time of Flight	s	Chapter 1
a	Semi-major Axis	km	Chapter 1
a_0 a^0	Goal/Measured Initial Semi-major Axis	km	Chapter 1
a_f a^f	Goal/Measured Final Semi-major Axis	km	Chapter 1
e	Eccentricity		Chapter 1
i	Inclination	rad	Chapter 1
Ω	Right Ascension of the Ascending Node	rad	Chapter 1
ω	Argument of Periapsis	rad	Chapter 1
θ	True Anomaly	rad	Chapter 1
$\mu_{☾}$	Gravitational Parameter, Moon	km^3/s^2	Chapter 1
ϵ	Magnitude of Acceleration	km/s^2	Chapter 2
γ	Flight Path Angle	rad	Chapter 2
coe	Classical Orbital Element	Various	Chapter 2
$\overline{a}$	Single Orbit Averaged Semi-major Axis	km	Chapter 2
$R_{☾}$	Radius of central body	km	Chapter 2
α	In-Plane Acceleration Pointing Angle	rad	Chapter 2
β	Out-of–Plane Acceleration Pointing Angle	rad	Chapter 2
$J_{2☾}$	Second Zonal Harmonic		Chapter 2
$\mathbf{r}$	Radius Vector	$\mathbb{R}^3$ km	Chapter 2
r	Magnitude of the Radius Vector	$\mathbb{R}^1$ km	Chapter 2
h	Specific Angular Momentum	km/s^2	Chapter 2
p	Semi-Parameter	km	Chapter 2
ν	Argument of Latitude	rad	Chapter 2
m	Mass	kg	Chapter2
δ	Normalized Variation		Chapter 2

Symbol	Meaning	Units	Chapter Introduced
m in $F_1\left(\frac{\pi}{2}\middle\vert m\right)$	Parameter or Square of the Modulus		Chapter 3
$\lambda_i,\ \mu_i,\ \nu_i$	Lagrange Multipliers	Anti-State Units	Chapter 3
$\mathbb{X}$	State Domain		Chapter 3
$\mathbb{U}$	Control Domain		Chapter 3
$\mathbb{R}$	Real Numbers		Chapter 3
$\mathbf{J}$	Cost Functional		Chapter 3
$\mathbf{x}$	States	State Units	Chapter 3
$\mathbf{u}$	Controls	Control Units	Chapter 3
H	Hamiltonian	Hamiltonian Units	Chapter 3
$\overline{H}$	Lagrangian of the Hamiltonian	Hamiltonian Units	Chapter 3
$\mathcal{H}$	Lowered Hamiltonian	Hamiltonian Units	Chapter 3

Acknowledgments

I would like to thank my advisors, Dr. Jennifer Hudson and LtCol(RET) Gary 'Grinch' Thomason for their immense support. Most importantly, my endless thanks to my wife and two amazing daughters.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction and Purpose

Expanding and increasing global interest in lunar and cislunar orbits is a consequence of both nation-state prestige missions and the potential for commercial gains. Science and commercial missions with interests in asteroid and lunar resource extraction have rapidly increased the number of current and proposed robotic and manned space missions in cislunar space. Planned national and private commercial robotic and human spaceflight missions are projected to grow significantly in both frequency and operational duration over the next several decades. Simultaneously, the space industry is experiencing a rapid maturation of low-thrust, high-Isp propulsion systems that are poised to take advantage of the unique dynamics present in cislunar space. These two elements offer a profound challenge to the United States' efforts to develop and implement simultaneous space traffic management and space domain awareness capabilities in regions beyond geosynchronous orbit.

The objective of this work is to develop a novel algorithm to support orbit determination for a noncooperative, low-thrust maneuvering spacecraft within cislunar space by assessing reachability. The ability to model finite maneuvers in cislunar space is often considered beyond the reach of junior students, engineers, and analysts in the space operations community. The tools needed to aid a practitioner in understanding the dynamics and to model the feasibility of such missions are often placed beyond the grasp of the beginner both in cost and simplicity of use. The purpose of this research is to make the findings of those methods available to a much broader pool of analysts and not require a dedicated study of these methods or advanced (and expensive) modeling tools.

One particularly exciting application at the intersection of these new capabilities and lunar interest is the low-thrust, low-energy moon-Earth return. Low-thrust propulsion spacecraft can, at extremely low cost in terms of ΔV, transfer from a selenocentric orbit to a geocentric orbit. As investigated by Liu, these types of transfer offer significant efficiency, particularly with high-mass robotic spacecraft that can afford prolonged cislunar transfer in terms of time and radiation exposure [1]. As lunar exploration, habitation, and exploitation begin over the next several decades, situational awareness of spacecraft that utilize these transfers

will become critical to ensure safety of flight and coordination between multiple operating entities.

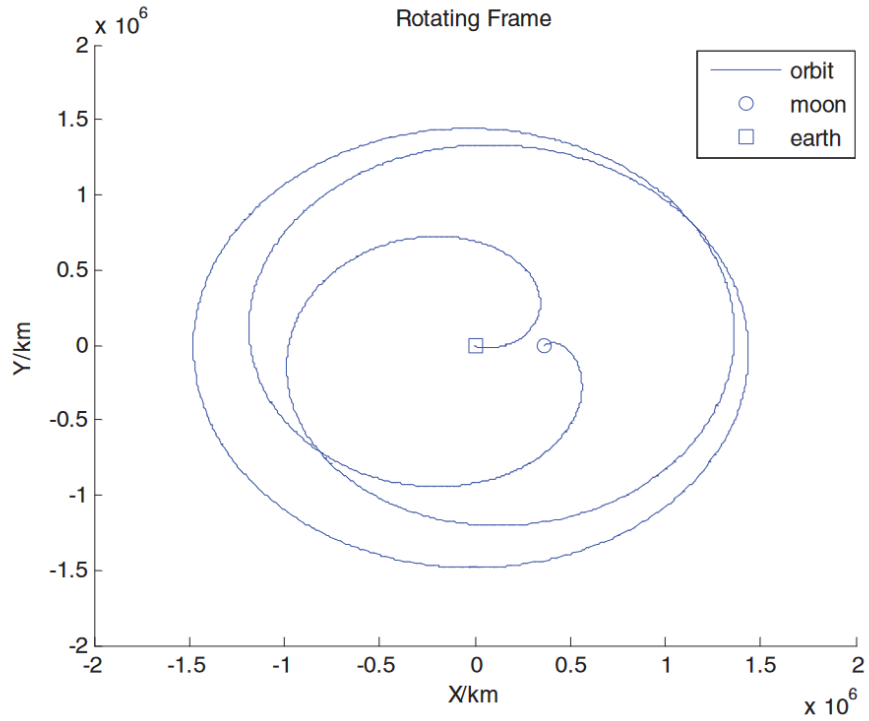


Figure 1.1. The Low-Thrust Low-Energy Moon-Earth Transfer of Liu et al. Source: [1].

This research uses a low-energy low-thrust transfer just like the one depicted in Figure 1.1 from a low-lunar orbit to a geocentric orbit and is used as the culminating case study. Figure 1.1 shows the low-thrust low-energy moon-Earth transfer of Liu et al. [1] in the Earth-moon rotating frame and demonstrates the extreme distances associated with low-energy Earth-moon transfers. Several elements of this transfer make it both advantageous for operators and extremely challenging for actors outside of the mission. The ability of an external observer to anticipate, track, and propagate noncooperative missions using this transfer is extremely limited.

1.1 Motivation

Given the increasing interest in low-thrust capabilities within cislunar space and the inherent complexity of continuous low-thrust dynamics, there will be a strong near-future need for simple and computationally inexpensive algorithms to assess the relative value of further expensive and complex analysis for noncooperative missions. This research achieves forward progress on the simplification of some portions of these problems. It provides computationally cheap and mathematically simple estimates to develop an initial feel for the dynamics of low-thrust maneuvers. Although the overall goal is to make the results of this research as accessible as possible, advanced methods will be required to suitably argue the feasibility of the proposed approach. The reader is encouraged to use Section 1.3 to guide their examination of this research.

Although nation states were once the primary operators of spacecraft, especially for missions capable of operating at lunar distances, continued commercialization of space suggests a future with numerous commercial, non-state, state, and potentially even privately owned spacecraft operating simultaneously in cislunar space. Coordination and collaboration currently seen between and among geocentric spacecraft operators suggest that there will be some minimum level of mission information shared with other operators for situational awareness and to facilitate safety of flight [6], [7]; however, the increasing competition for profit and prestige and the strategic advantage for commercial and national missions of maintaining uncertainty in capability and intent indicate that many spacecraft will be viewed as "noncooperative" missions. As such, analysts may only have the most basic estimates of other spacecraft's capabilities and a general concept of the operators' intent.

Calculating the updated state of a satellite is a central element of space mission management. This work assumes a basic understanding of astrodynamics. Specifically, it assumes an understanding of Newtonian dynamics, a minimal comprehension of calculus, and some exposure to space operations concepts. The author's goal is that the resulting algorithm presented in Chapter 5 and the associated estimates (Equations 4.2 leveraging 4.11, 4.5, 4.8, 4.9, 4.10 may be used by a motivated advanced-placement high school student. Although full consumption of this research requires some experience with specialty optimization theory, familiarity with the main topics in *Understanding Space* [8] or any other early undergraduate text [9]–[12] is all that is necessary to understand and apply the findings of

this work as demonstrated in Algorithms 2 and 3 in Chapter 5.

Almost all introductory instruction in space operations includes the rudimentary calculation of impulsive satellite maneuvers. Impulsive satellite maneuvers assume an instantaneous force impulse on a satellite that immediately changes its orbit. These calculations require only basic algebraic operations to calculate the change in velocity, or ΔV required to move from one orbit to another. Modeling a finite-thrust maneuver is more challenging and is typically introduced only at the graduate level of astrodynamics studies. Because the force required to move the satellite from one orbit to another occurs over a finite period of time, the satellite's orbit changes continuously throughout the thrusting time as it moves from its initial orbit to its final orbit. Traditionally, understanding and modeling these transfer techniques requires the application of calculus and the use of numerical methods. This sets finite maneuvers apart from impulsive satellite maneuvers. Continuous thrust maneuvers are a subset of finite maneuvers that typically involve long periods of low-thrust, often over many revolutions about the satellite's central body.

Although significant efforts have been made to explain the methods involved in more advanced topics that are used to validate the model, such as the application of the calculus of variations, optimal control theory, and other optimization approaches and software, this work is no substitute for dedicated study of these topics. Indeed, the entire purpose of this effort is to make the findings of those methods available to a much broader pool of analysts and not require dedicated study of these methods. For that reason, significant portions of the optimal control section have been sequestered to the appendices. However, all of the code required for the optimal control analysis for both Matrix Laboratory (MATLAB) and Systems Tool Kit is also included as appendices. While complete replication of this research requires access to specialist software such as DIDO for MATLAB and Systems Tool Kit's Astrogator, those applications are primarily limited to verification and validation of the optimality of the calculation schemes. Great effort was placed on enabling any user with a basic MATLAB license to recreate the basic dynamics for those willing to accept the reigning optimality of the analytical components.

1.2 Overview and Scope

The culminating contribution of this research is an algorithm that supports a space mission analyst's understanding. An analyst should be able to use the algorithm developed in this work to decide whether additional high-fidelity modeling is required. The end result is a generic procedure that guides an analyst through a deliberate process and returns an enhanced analytic understanding of noncooperative spacecraft actions. This algorithm is intended to be generic enough so that it may be implemented in nearly any programming language and take advantage of organizationally or individually preferred numeric propagators and solvers. This approach is designed to extend interoperability with existing software systems and human-machine workflows without creating a new requirement for syntax, integration, or epoch management. It is entirely agnostic of the central body of an orbiting spacecraft and is transferable to any restricted two-body dynamics model.

Although the final algorithm relies entirely on linearized estimates to enable extremely low computational costs, validation of the algorithm requires comparative analysis of several dynamical models that describe the motion of a satellite under continuous thrust with much greater fidelity. This analysis includes a detailed discussion of the degrees of accuracy in modeling these dynamics currently available to the analyst. Great emphasis is placed on demonstrating the continuous reduction in model accuracy to achieve simplified model dynamics that can be used for comparison and acceptance of the algorithm. The basis of this effort is that the required ΔV of a continuous, low-thrust transfer can be directly translated into a time of flight cost, or ToF.

Although a specific low-energy moon-Earth transfer orbit will be examined in this study, given a minimum amount of assumed initial conditions and parameters, application of the underlying framework is an appropriate approach for interpreting and anticipating the actions of any noncooperative mission. While subject to and requiring the knowledge of advanced dynamics and other perturbative models, this research makes liberal use of patching concepts to reduce large-scale problems to many manageable smaller-scale ones. As an example, though the case study presented is of a complex low-thrust, low-energy return, analysis is scoped to the lunar exit portion allowing dramatic simplification of localized dynamics and reduction of necessary terms to find implied outcomes. This scoping is initially addressed in this chapter and is revisited in the dynamics section of the model

construction and assessment section within the optimal control chapter. Within the context of this research, cislunar space will focus on the region surrounding the Earth between geosynchronous orbit radius and extending to the far edge of the range of primary Earth-moon gravitational influence. This will generally be considered between one and thirteen times the geostationary radius [3].

Accurate development of an Earth-moon low-energy transfer requries a significant understanding of low-thrust spacecraft maneuver, multibody gravitational modeling with perturbations, and orbit determination; each of which encompasses entire areas of study within academia and engineering industry. This research combines them through a scope-limiting application to allow for the production of a tailored and practical tool. Since this research focuses exclusively on noncooperative spacecraft, capturing and recording uncertainty is a key concept throughout the process. Informed primarily by the logic of optimal control theory, the final output of the completed algorithm is only as good as the uncertainty measurement and foundational simplifying assumptions. In this manner, the algorithm is a "garbage in, garbage out" system.

At the highest level of this system, a spacecraft's ephemeris and planned maneuver information are not publicly or directly shared. This mission is "noncooperative" with space domain awareness elements. Within the lens of optimal control, the mission is assumed to operate by execution of the optimal trajectory. Within the scope of this effort, only a focused element restricted to the system of the robotic space mission in question will be considered. The system consists of a spacecraft, the spacecraft's mission "owners," and the environment in which the spacecraft is operating. The focused system includes the robotic space mission's owners which comprises both the physical and virtual infrastructure and resources which are required to plan and operate a dynamic robotic space mission. The subordinate elements of this system and their generalized relationships are depicted in Figure 1.2.

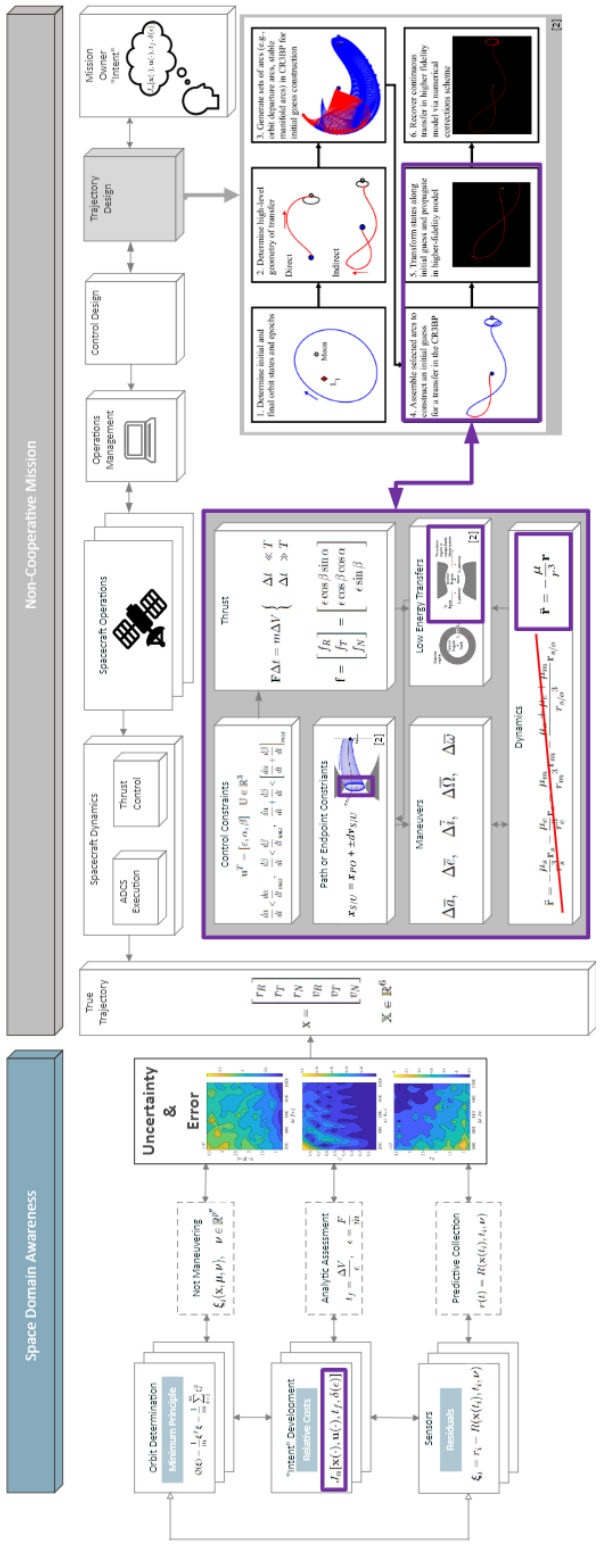


Figure 1.2. Highest Order Problem System. Adapted from [2].

Figure 1.2 contains a containerized view of the broad system related to the management and understanding of a low-thrust mission. On the right side, identified by the gray box at the top, is a conceptual process map required for the operation of the satellite. This side represents the actions of the noncooperative spacecraft mission owners. On the left side, identified in blue is a conceptual process map that is required for space domain awareness. This side represents the actions of an analyst attempting to understand the noncooperative spacecraft's actions and determine the likelihood of future positioning. The relationship between these two sides is compartmentalized. The only common knowledge between the two is the current state of the satellite in question.

On the right side of Figure 1.2 (shown also in Figure 1.3), the mission owner must execute a number of actions. These specific elements range from prioritizing basic mission objectives to executing synchronized subsystem controls to affect a desired maneuver. In an effort to sufficiently generalize this research and to allow for the greatest possible application, this research is mainly concerned with the trajectory design stage. Moving through the model of trajectory design, a dynamics model is selected and a basic guidance scheme is developed to maneuver the satellite from one orbit to another. This guidance scheme is then translated into control actions based on the specific spacecraft, transmitted as a command sequence, and executed within the broader mission context. Together, this string of actions at the top of the process map represents a generic mission cycle. Throughout the lifetime of a cislunar mission, various specific dynamics models are required to achieve sufficient accuracy to execute the mission cycle. At some times, mission designers may need to use complex multibody dynamics. At other times, they may only need to use simplified two-body dynamics. These varied dynamics environments are shown below the mission cycle both mathematically and pictorially.

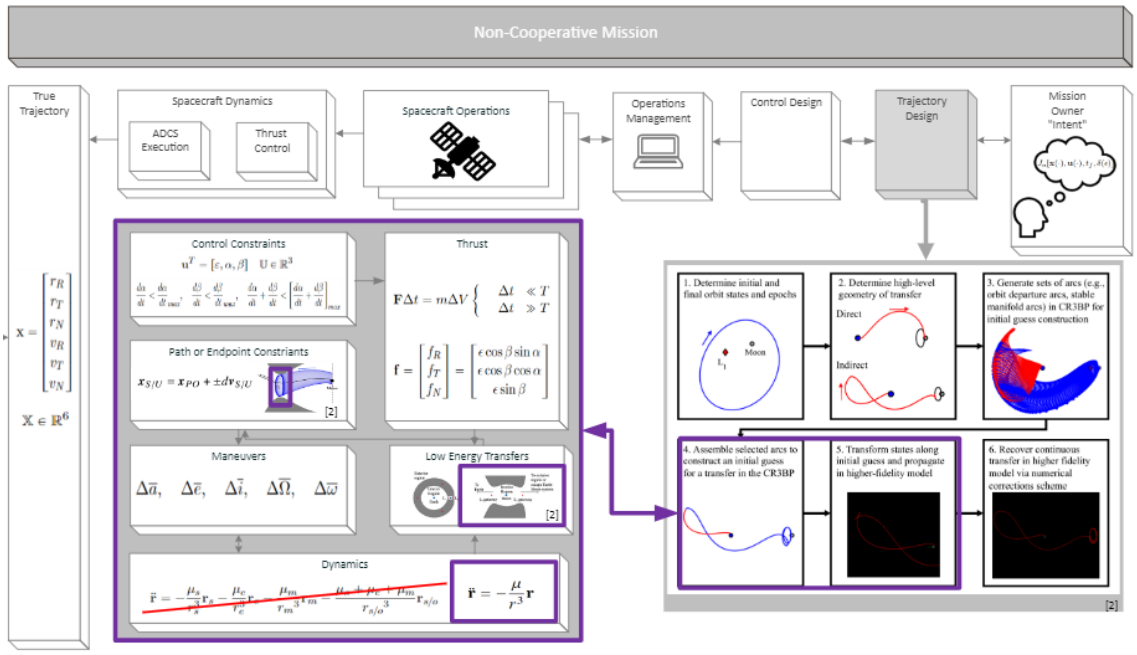


Figure 1.3. Highest Order Problem System Spacecraft Operator Perspective. Adapted from [2].

On the left side of 1.2 (shown also in Figure 1.4) is the Space Domain Awareness actor. The mission cycle for this actor typically consists of data collection through sensors and orbit determination through measurements. These processes are depicted at the top and bottom of the left sequence. The synthesizing element of these two actions is the development of intent. In this stage, the analyst must make a judgment that will affect future sensor tasking without full insight of the noncooperative mission. Although this judgment can be highly predictable for well-observed and easily modeled orbits like geostationary orbits, because the satellite is under continuous thrust and located in the cislunar regime, the analyst must now turn to other methods to predict the future position of the spacecraft. These methods are executed in an environment characterized by uncertainty and error. The analyst has uncertainty in their understanding of the missions capabilities. They do not know the overall intent of the mission, and the mission owner's priorities for the spacecraft. The error enters the analyst's system through inaccuracies in the measurement of the observed state of the spacecraft and errors in the analysis model. Clearly, the noncooperative relationship between the mission owner and Space Domain Awareness analyst sets the analysts uncertainty to an initial

level. Analysis based on assumptions and knowledge of the dynamics of the environment can reduce that uncertainty. Uncertainty and error are graphically depicted between the spacecraft's true state and the actions of the Space Domain Awareness analysts mission cycle actions.

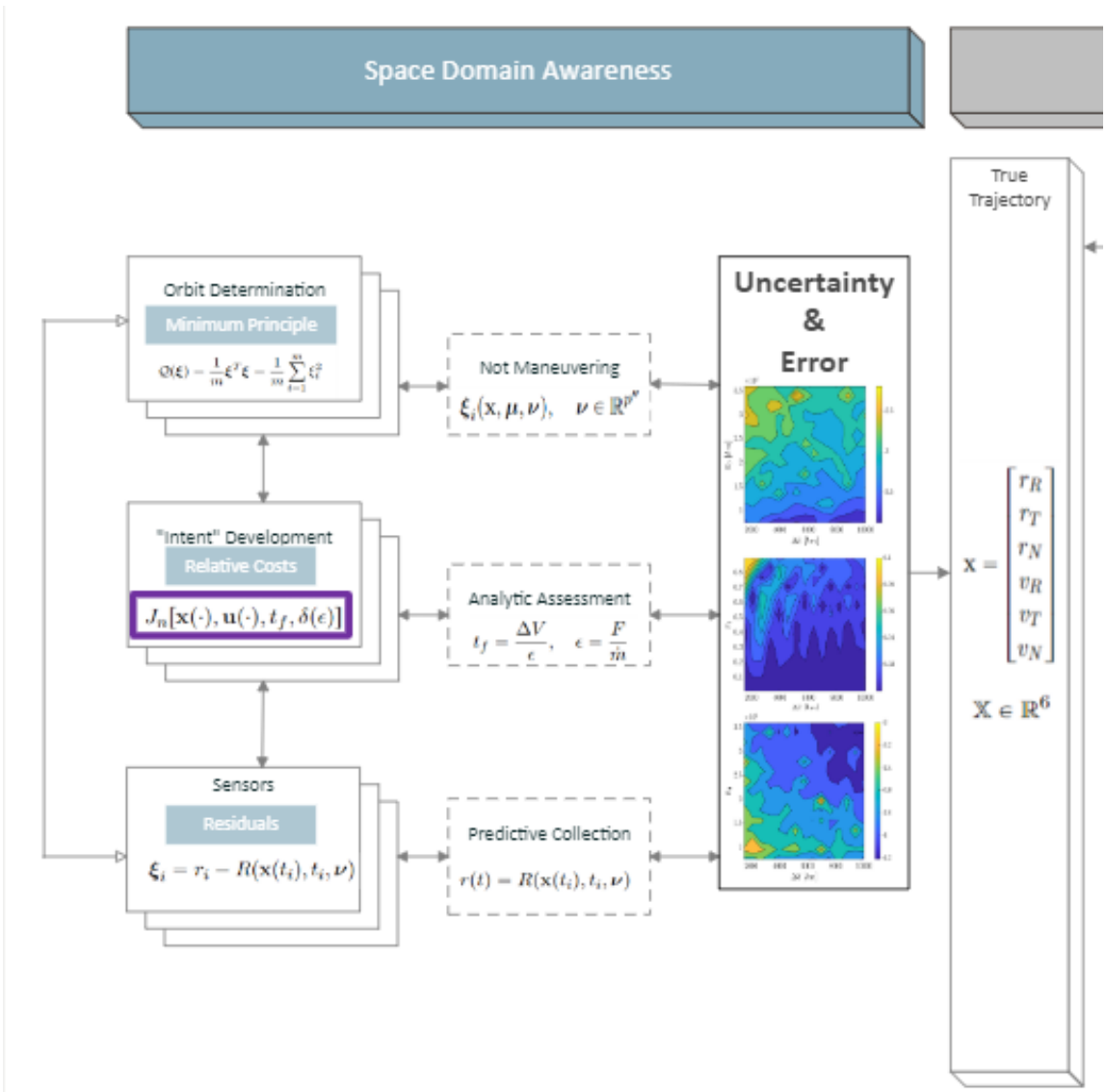


Figure 1.4. Highest Order Problem System Space Domain Awareness Analyst Perspective.

1.3 Organization

The remainder of this work is divided into four main chapters that focus on incrementally developing, assessing, and applying the algorithm. I have attempted to standardize the use of variables to the greatest extent possible and have included a definitions page that outlines the use of all variables, the chapter they are introduced in, and their corresponding units (if applicable). All of these files used in this research are contained in the appendices and are available through the Naval Postgraduate School library as a .zip file.

Chapter 2 and its corresponding appendices capture the methodology used to develop, test, and accurize the algorithm as well as build the foundational assumptions that make the algorithm applicable. Low-thrust dynamics, control, and optimization are used to validate and assess the accuracy of the algorithm's component equations here. Within this chapter, the subordinate sections include a background section, dynamics development, validation of optimality for supporting equations, and several examples which validate the underlying model.

Chapter 3 and its corresponding appendix contain the numerical methods assessment of optimality for the proposed control strategies that build the analytical equations used in the algorithm. This section contains some advanced elements related to optimal control. The supporting references are listed throughout the chapter, but Dr. Ross's *A Primer on Pontryagin's Principle in Optimal Control* is immediately recommended for any interested reader [13]. This chapter develops the model and system that validate the analytical techniques in Chapter 3. Specialist software is required to run the associated MATLAB code and implement the findings in Systems Tool Kit.

Chapter 4 addresses the accuracy of the equations used to provide analytical estimates of ΔV within the algorithm. It relies on the models created in Chapter 2 and the findings espoused in Chapter 3, but knowledge of how these models and methods work is not required to understand and interpret the results. A basic understanding of orbital mechanics and some limited calculus is all that is required to fully digest this section. Although many of the concepts that Vasile and Di Carlo employed to capture, simplify and record these equations rely on moderately advanced mathematical concepts, the intent is to present the information in a manner that does not require deep familiarity with Battin's excellent *An Introduction to the Mathematics and Methods of Astrodynamics* [4] [14]. This chapter highlights and

develops the utility of the specific analytical techniques necessary to make the algorithm work. A basic MATLAB license is all that is required to run the associated code for this chapter.

Chapter 5 contains the algorithm, user explanations, and several demonstrations. The bulk of this chapter is the author's original application of the tools developed by others. A basic MATLAB license is all that is required to run the associated code files for this chapter. The algorithm itself can be executed in any programming language capable of executing elliptic integrals or, as explained in the chapter, accepting increased error while only requiring basic algebraic computations.

Finally, Chapter 6 contains summative reflections on the extended applicability of this algorithm and similar analytical efforts, the author's thoughts on supporting software used in this analysis, as well as recommendations for further associated research. This section also covers the author's use of Generative Artificial Intelligence tools known as Generative Predictive Transformers for the development of supporting computer code and post-writing document revisions.

Appendix A is a full-size reproduction of Figure 1.2.

Appendix B contains the associated elements that are necessary to support Chapter 2, but would detract from the readability of the chapter. Many long derivations of Hamiltonian identities, states, costates, and associated elements are captured here.

Appendix C similarly contains some reproductions of lengthy derivations that support the work in Chapter 3. Only where discrepancies exist between sources or original sources that are particularly difficult to adapt to a recognizable nomenclature are existing derivations repeated.

Appendix D contains supporting results from example demonstrations in Chapter 4 sufficient to examine the findings without running the code. These consist mainly of additional plots.

Finally, Appendix E contains all the MATLAB code necessary to completely recreate this effort. There is a complete index of the included files, and the files that follow are listed alphabetically.

1.4 Orientation

Within the context of this research, an algorithm is a documented series of steps performed by a human or a computer in a specific order. It requires inputs and produces outputs. Between input and output, a series of actions are performed to transform the input data into output data. The principal human tasks in the algorithm occur at the beginning and end of the algorithm. Human intuition and a basic understanding of the underlying dynamics of the system are a prerequisite to ensure that the input data are sufficient and accurate. Human understanding and contextual synthesis again demands human interaction at the end of the algorithm to interpret the computed outputs and apply them to the situation appropriately.

Although this middle portion of the algorithm can rely heavily on automated computer processing, specific implementation is incredibly situation-dependent, and users would be well advised to incorporate intermediate human validation actions before continuing through the algorithm. Automated programming can be used to perform the necessary computational steps between the inputs and outputs of the algorithm. The intent of this research is to produce the most generic computational actions to enable implementation through any available programming language for maximum efficiency and interoperability with existing mission management tools and processes.

This hybrid analyst-software approach serves two purposes. First, as will be discussed in greater detail, the dynamics which govern low-thrust orbits transfers across the cislunar space regime are incredibly computationally expensive. This means that completion of the algorithm can be time consuming, even with perfectly formed inputs and ideally specified conditions. Mid-algorithm human validation gives the analyst the opportunity to reflect upon the validity of the initial inputs at various stages of completion and to re-execute computational steps with updated inputs rather than continuing with time-consuming complications on inaccurate or inefficient input descriptions of the applicable situation. Second, mid-algorithm computations can help provide useful context for enhanced problem understanding. It is not only possible that the analyst's understanding of the nuances of an examined situation may change during the course of algorithm completion, it is likely!

As a sample and format orientation, an example algorithm is shown in Algorithm 1. This format will be used throughout the work to demonstrate the steps necessary to repeat the findings of this research. The final novel deliverable algorithm is also listed in this format in

Chapter 4. This example is listed in an effort to demonstrate the simplicity of the algorithm. The following section offers a roadmap of the remaining work and the direct linkages to the algorithm for those wishing to jump to application.

In the context of the sample mission, starting from a low-lunar orbit and entering a low-energy return trajectory, a simplified example of this algorithmic process can be described to demonstrate the usability of such an algorithm. It will also serve as an orientation to the annotation used throughout this effort. Each demonstration will begin with an orientation as outlined below. Ultimately, these efforts will support the development of applicable algorithms as shown in Algorithm 1.

$$\textit{Orientation}\begin{cases} \text{Dynamics} & \text{the dynamics of the system} \\ \text{Endpoints} & \text{describe the conditions at the end of the maneuver} \\ \text{Parameters} & \text{constants or other elements used in the model} \\ \text{Controls} & \text{the inputs that describe the thrust forces in the maneuver} \end{cases}$$

Algorithm 1 Generic Estimate of Reachability

procedure COST CALCULATION (ΔV)

Record Initial Conditions, $\mathbf{x}_0 = [a_0\, e_0\, i_0\, \Omega_0\, \omega_0\, \theta_0]^T$

Record Necessary Final Conditions, $\mathbf{x}_f = [a_f\, e_f\, i_f\, \Omega_f\, \omega_f\, \theta_f]^T$

Record Initial and Required Final Times, $[t_0\, t_f]$

Record Estimate of Maximum Acceleration, ϵ

Compute the cost in ΔV to move each element of $\mathbf{x}0$ to $\mathbf{x}f$:

$$\begin{aligned}
\Delta V(a_0, a_f) &= \Delta V_1 \\
\Delta V(e_0, e_f) &= \Delta V_2 \\
\Delta V(i_0, i_f) &= \Delta V_3 \\
\Delta V(\Omega_0, \Omega_f) &= \Delta V_4 \\
\Delta V(\omega_0, \omega_f) &= \Delta V_5 \\
\Delta V(\theta_0, \theta_f) &= \Delta V_6
\end{aligned}$$

Compute the total cost in terms of ΔV:

$$\Delta V_{total} = \sum_{i=1}^{6} \Delta V_i$$

Compute the total time of flight for the transfer, ToF

$$ToF = \frac{\Delta V_{total}}{\epsilon}$$

Analyze maneuver against the known t_f

if $t_0 + ToF \leqslant t_f$ **then** maneuver is possible

if $t_0 + ToF > t_f$ **then** maneuver might not be possible*

if $t_0 + ToF \gg t_f$ **then** maneuver unlikely

end procedure

*an examination of accuracy is required

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2: Methodology

This chapter and its corresponding appendix captures the methodology used to develop, test, and accurize the algorithm. It also builds the foundational assumptions that make the algorithm applicable. While this chapter is critical to the development of the algorithm by establishing the model by which simulations can test and evaluate various approaches, it is not a necessary component for users of the algorithm. This chapter focuses on the development of differential equations of motion and the evaluation of the accuracy and computational cost of modeling maneuvers in the cislunar domain. The baseline control schemas used to validate the model are assumed to be optimal here and a numerical methods comparison in will be executed in Chapter 3 to test that assumption. Beginning in this section and throughout this research, the use of classical orbital elements with the $\overline{overline}$ structure ($\overline{a}\,\overline{e}\,\overline{i}\,\overline{\Omega}\,\overline{\omega}\,\theta$) denote the *averaged* value of that element over the course of an orbit as opposed to the instantaneous value of that element ($a\,e\,i\,\Omega\,\omega\,\theta$) at a specific time. This concept was developed by Petropolous and is described in [4], [15], such that

$$\Delta\overline{coe}_{2\pi} = \int_0^{2\pi} \frac{dcoe}{d\theta}\, d\theta$$

where coe represents any element $(a, e, i, \Omega, \omega)$.

The foundational elements of low-thrust dynamics, control, and optimization used to validate and assess the accuracy of algorithm component equations are the major components of this chapter. Within this chapter, the subordinate sections include a background section, dynamics development, and several tests which validate the underlying model. This section contains some advanced elements related to the use of ordinary differential equations. Supporting references are listed throughout the chapter, but Battin and Vallado's texts are immediately recommended for any interested reader [14], [16]. This chapter develops the model and system that Chapter 3 relies on to validate optimality. A basic MATLAB license is all that is required to recreate the findings of this chapter.

In the context of this effort, although cislunar space consists of the full 4π-steradian sphere

with a radius of 13 times that of geostationary orbit ($14 \times 37,000$km) [17], the utility space of the decision support algorithm is necessarily constrained to the region dominated by the gravitational force of the moon. Relative distances within the cislunar regime are demonstrated in Figure 2.1 from Holzinger's primer [3].

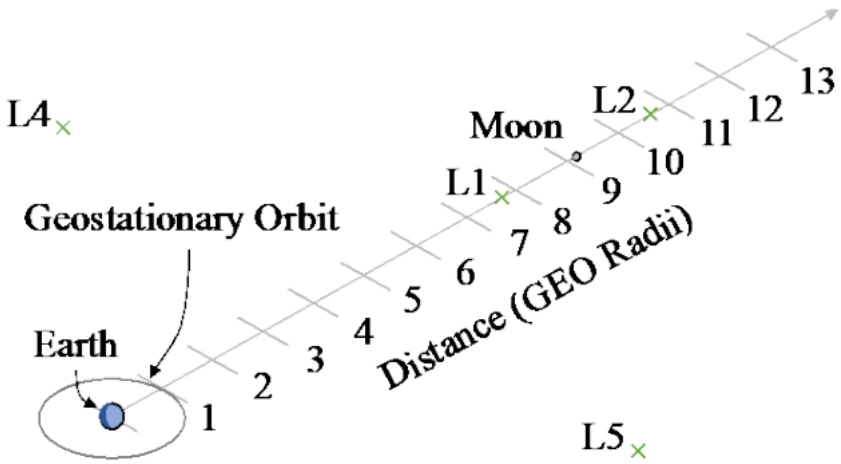


Figure 2.1. Cislunar Distances. Source: [3].

In scope, this research is intended only as a vignette selected from a myriad of possible scenarios to highlight the challenges of space traffic management and space domain awareness in cislunar space. It is the author's hope that this effort highlights the need to further consider the complementarity of analytical estimates for low-thrust maneuver in conjunction with numerical methods associated with select boundary condition problems.

2.1 Background

Optimal trajectory development of continuous thrust spacecraft has been the subject of significant research efforts since the early 1960s. D.F. Lawden demonstrated the value of expanding the cost functional beyond strictly minimum fuel consumption or minimum time transfers [8]. While recognizing Pontryagin's emerging techniques of optimization, Lawden derived much of his findings from the classical calculus of variation and basic Mayer cost functionals. Like many others at the time, Lawden derived non-optimal shape-based curves that assist in the initial development of the optimization problem and can still be used

today to identify extremal case outcomes and ensure sufficient feasibility in initial problem development.

In more recent times, Hudson and Scheeres developed a novel method of optimization for low-thrust maneuvers that models the thrust acceleration components as Fourier series in eccentric anomaly. Relying on Gauss's averaged variational equations, their method provides a general case method for solving optimal transfers with significantly fewer coefficients for spiral trajectories initially developed in [18] and further improved upon in [19]. Without obligation to impose an artificial constraint of constant-magnitude or constant-direction thrust, this approach offers significant advantages to complex low-thrust mission design.

The late 1950s and 1960s produced an entire series of extended analytical, semi-analytical and nascent numerical methods optimization approaches for finite and continuous thrust orbital trajectory design. Canonical texts by Nelson and Loft [20], Deutsch [21], and several key studies [22] and academic publications [23] by Marec provide both a breadth and depth of analytical and semi-analytical estimates for various types of space vehicle maneuvers. Unfortunately, these texts are all written in a now dated mathematical notation and also assume a high degree of competency with advanced Hamiltonian and Lagrangian mechanics. Additionally, many of these findings are restricted only to the most trivial orbit types, mainly circular orbits without inclination. Though they contain highly applicable "shortcuts" that remain useful in trajectory design, their general inaccessibility keeps these findings from most students.

Similarly, Brumberg's *Analytical Techniques of Celestial Mechanics* uses intimidating notation, but does clearly articulate the use of elliptic functions in application to the modeling of averaged osculating elements [24]. These techniques ultimately enable modern authors to devise highly accurate estimates for elliptical orbits. This expansion beyond the triviality of circular orbits greatly expands the applicability of the findings for modern space missions.

Finally, Di Carlo and Vasile produced a refined and consolidated version of Battin's analytical solutions to Gauss's variational parameters with several novel derivations [4], [14]. The most complex of these estimates relies on the use of complete elliptic integrals of the first kind, but still remain several orders of magnitude less computationally expensive than traditional integration methods even for short duration transfers. By leveraging a large history of previous analytical work done by Burt [25], Battin [14], Petropoulos [15], and

Pollard [26],Di Carlo and Vasile are able to produce a number of analytical control schema and analytical estimates of the transfer cost in ΔV required to implement them. The accuracy of many of these estimates is observed to be within 1% of propagated findings for geocentric orbits out to the edge of the Earth's sphere of influence and has been observed to show sufficient accuracy within the lunar sphere of influence as well.

2.2 System Bounding and Assumptions

The robotic spacecraft itself is a part of space domain awareness and noncooperative mission actors, as are the spacecraft dynamics that are caused as a consequence of commanded action in Figure 1.2. Spacecraft actions in Figure are constrained by a physical environment defined by governing differential equations of motion. While each of these system elements are interactive, these "mission owner" elements can be further segmented into basic mission roles including Intent, Trajectory Design, Control Design, and Operations Management. These elements are outlined from right to left in a pseudo-sequential order that represents the segmented process of robotic spacecraft operations from intent development to execution. In reality, there are a number of shared elements that connect and interweave the arbitrarily sterile and segmented sections described previously. Relational exchanges between elements are often iterative and multi-looping; however, for the purpose of simplicity, only a very simplified relational system is illustrated here. Mission owner "intent" represents higher-order, mission-level direction.

Trajectory development and dynamic modeling happens between trajectory design and control design, while low-level commanding occurs at the operations management level. Low-level actuator control occurs within the physical spacecraft between the attitude determination and control system and propulsion (thrust and thrust vector control) system. For the purposes of our model, the spacecraft is a point mass subject to a single propulsive force characterized by the thrust vector described in the R,T,N coordinate system by angles α and β as shown in Figure 2.2. The spacecraft is at the origin of this coordinate system. The R (Radial) axis aligned with the position vector from the central body to the spacecraft. The T (Transverse) axis is formed by right angle to the R axis and in the direction of satellite travel. The plane formed by R and T is aligned with the orbital plane. The N (Normal) axis is formed by the right hand rule of R into T and is perpendicular to the plane formed by R and T. This means that the pointing angles α and β which are used to describe the direction

of the thrust acceleration forces are not necessarily the same as the pointing angles of an engine that would produce these forces on the spacecraft. A control program based on a guidance and navigation plan formed with these angles requires additional conversion to a separate, specific spacecraft coordinate system.

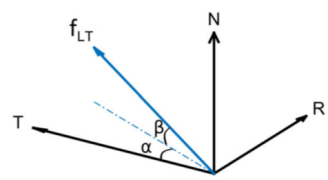


Figure 2.2. Acceleration Vector Description. Source: [4].

The spacecraft itself operates subject to both the internal and external dynamical environments within the overall system. Externally, the spacecraft is subject to a myriad of forces, though the principle force is that of the gravitational potential of other masses on the spacecraft. Internally, the spacecraft has the ability to impart a force through the use of propulsive thrusters. Finally, there are a number of perturbation forces that are a function of internal, external, or both environments. These may include solar radiation pressure, the imparted force of micrometeorite impacts, or the drag imparted to a cross-sectional area of the spacecraft by the atmosphere of a planet. Indeed, even the uneven distribution and concentration of mass inside gravitational bodies can cause zonal harmonics that can be modeled as perturbative forces [14]. As the sum total forces acting on a spacecraft increases, the force of a few gravitational potentials often dominate all others. When a spacecraft is operating near specific celestial bodies, or when one of the celestial bodies is particularly massive in comparison to the others, some limited combination of the closest and most massive of gravitational potential sources suffice to represent the primary force on the spacecraft.

In this research, the maximum accuracy of gravitational potential laws considered will be limited to the elliptical restricted four-body problem. This reduces the number of gravitational forces to those of the sun, Earth and moon and assumes a satellite with negligible mass compared to the planetary and solar bodies.

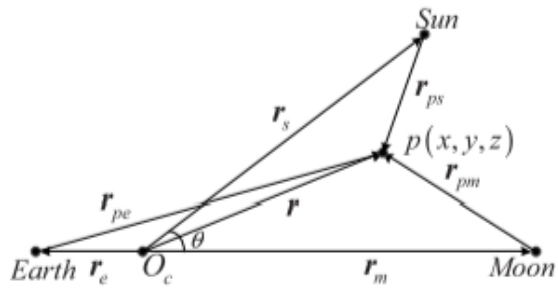


Figure 2.3. Elliptical Restricted Four Body Problem Model. Source: [5].

While the circular restricted three-body problem is often used for cislunar dynamics, however, this model fails to account for the eccentricity of the moon's orbit and this eliminates the opportunity to exploit known low-energy orbits [5], [27], [28]. While time-dependent based on the relative location of the Earth, moon, and sun, these low-energy transfers have been shown to reduce the required ΔV by half compared to direct transfers. Within the context of the circular restricted four body problem, low-energy transfer orbits take advantage of the libration point solutions (weak stability boundary theory) to efficiently ride the forces of many local gravitational potentials to reduce the need for additional thrust energy input to maintain or increase disturbing accelerations on the spacecraft [5].

$$\ddot{\mathbf{r}} = -\frac{\mu_s}{r_s^3}\mathbf{r}_s - \frac{\mu_e}{r_e^3}\mathbf{r}_e - \frac{\mu_m}{{r_m}^3}\mathbf{r}_m - \frac{\mu_s + \mu_e + \mu_m}{{r_{s/o}}^3}\mathbf{r}_{s/o} \tag{2.1}$$

There exist equivalently-structured circular restricted multi-body problems (e.g. circular three-body problem), and because the moon's orbit is only mildly eccentric, these dynamic structures can offer sufficient accuracy for direct transfer methods for both impulsive and finite thrust problems [3], [5], [28]–[31]. However, within the context of this research, the approximation offered by the elliptical restricted four-body problem elliptical restricted four-body problem is necessary to model *low-energy* returns. When the spacecraft is sufficiently close to a specific celestial body such as the Earth or moon, we can further simplify the dynamics to a restricted two body problem where only the primary (closest) gravitational mass is considered. These dynamics are shown in equation 2 as the circular restricted two-body problem.

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.2}$$

And with the inclusion of the low-thrust force modeled as the the magnitude of acceleration of the thrust (ϵ), we find:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \frac{\mathbf{f_{LT}}}{|\mathbf{f_{LT}}|}\epsilon \tag{2.3}$$

For problem areas where circular restricted two-body dynamics can be applied, the equations of motion can equivalently be derived using Gauss's form of Lagrange's planetary equations (variational equations). The following equations outline the time variation with respect to the osculating elements as outlined by Battin and further developed by DiCarlo and Vasile. [4], [14].

$$\frac{da}{dt} = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \tag{2.4}$$

$$\frac{de}{dt} = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \tag{2.5}$$

$$\frac{di}{dt} = \epsilon\frac{r}{h}\cos\nu\sin\beta \tag{2.6}$$

$$\frac{d\Omega}{dt} = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \tag{2.7}$$

$$\frac{d\omega}{dt} = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \tag{2.8}$$

$$\frac{d\theta}{dt} = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \tag{2.9}$$

These equations also require the identity $\nu = \omega + \theta$ for the argument of latitude and $p = a(1-e^2)$ for the semi-parameter.

The advantage of translating the dynamics from state vector to these differential equations based on Keplerian elements is twofold. First, it allows intuitive interpretation of raw-time history values of the elements. This offers significant advantages during debugging and analysis and provides more information directly from the raw histories than state vector data. Even for circular orbits about an ideal point-mass, state vector position and velocity must each be normalized to simple magnitude to provide any human-interpretable data. Secondly, it allows for the use of existing closed form analytical estimates to validate and verify findings and to prevent the imposition of purely infeasible problems during initial condition development. In the subsequent demonstrations in Chapter 5 the final position is modeled as the transfer orbit necessary at the appropriate epoch to enter the low-thrust transfer. This essentially makes the final position of the restricted two-body problem the

patch point for further analysis using elliptical restricted four-body dynamics.

2.2.1 Baseline Assumptions and Construct

As a unified construct within the overlap of the three main elements, the completed algorithm occupies only a small area of existing and future research opportunities. Considered within a simplified paradigm of two major actors, an actor conducting space domain awareness as described in current U.S. executive branch coordinated roles [3] and a noncooperative actor commanding a robotic space mission. There several further simplifying assumptions that establish a common backdrop for this research. First, it is assumed that the two actors are not cooperating; that is, there is not a shared ephemeris for the maneuvering satellite involved in the robotic space mission.

Second, it is assumed that the space domain awareness actor knows some basic characteristics of the robotic mission including an *estimate* of spacecraft mass, propulsion system capability envelopes, and the identity of the mission owner. There are a number of historical and current efforts to address maneuver detection and orbit determination of satellites maneuvering under impulsive force transfers [32], [33], however, this work is focused exclusively on continuous thrust transfers. Within the context of this research, differentiation shall be made based on the duration of the transfer force relative to the initial orbit's period.

$$\mathbf{F}\Delta t = m\Delta V \begin{cases} \Delta t & \ll T_0 \text{ for impulsive maneuvers} \\ \Delta t & \gg T_0 \text{ for continuous thrust maneuvers} \end{cases}$$

In this relationship, $\mathbf{F}$ represents the disturbing force of the maneuvering satellite's engine or propulsion device, Δt is the time duration of the propulsive force, m is the spacecraft mass, ΔV is the change in velocity, and T_0 is the period of the satellite's initial orbit. This will generally restrict application to spiral transfer orbits, though there are examples where orbital period is sufficiently long to allow single-arc transfers. In the case of this research, those single arc transfers are extended to continuous sections with constant magnitude. In some cases, these finite arcs may extend for longer durations and curl around as multi-revolution maneuvers.

2.2.2 Initial Problem Bounding

Although section two will more thoroughly outline the dynamics that will be used to formulate and evaluate candidate optimal solutions, a basic commitment to restricted two-body problem with zonal harmonics to order 4 and the low-thrust acceleration is required to describe in some detail the basic problem constraints. As shown graphically in Figure 2.4, reducing the number of gravitational bodies that influence the path of the spacecraft allows for a much more human-interpretable system of equations to be used or at least referenced (consider state vector to classical orbital element conversions). While some references to the larger multibody system are required to discuss and determine problem constraints, only the space between the lunar surface and extremal of lunar sphere of influence will be considered in analysis and dynamics.

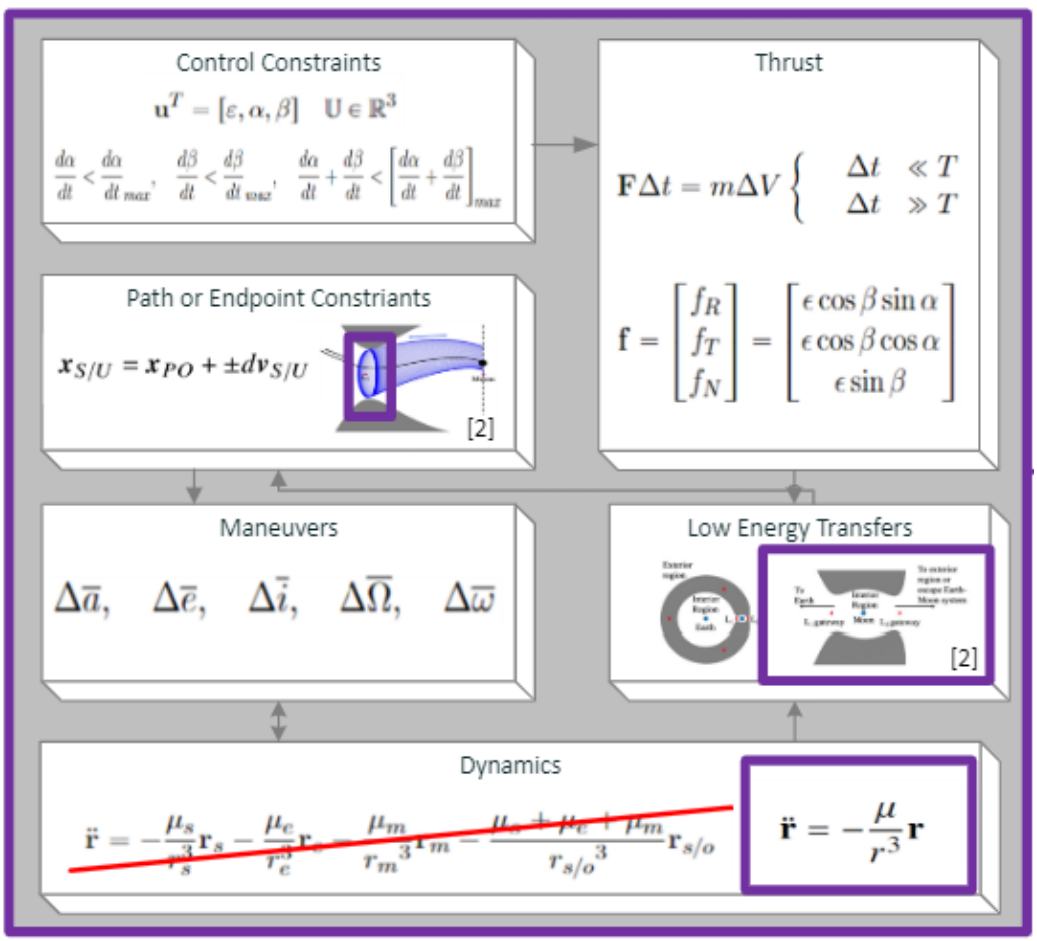


Figure 2.4. Initial Problem System

Because of the time-dependent nature of higher-order dynamics, namely the necessary relative alignment of the Earth, moon, and sun to facilitate a low-energy transfer, the optimal trajectory problem is initially defined as a t_f fixed. At this specific point in time, $\mathbf{x}(t_f)$, the final total state of the spacecraft, must be sufficiently equal to that of the initial state of the next patching manifold. In the continuing cislunar example, the state necessary to make a low-energy transfer in elliptical restricted four-body dynamics is the designated $\mathbf{x}(t_f)$. The precision to which that patch point at t_f is made depends on external elements of the system, including the ability to affect mid-course correction during low-energy Earth-sun transfer, and is beyond the basic scope of this effort. However, this analysis indicates that there may be either a point or area of points which defines the end-point constraint.

A basic necessary path constraint is that the spacecraft trajectory may not pass through the lunar body. This exclusion zone can be described as a sphere of radius equal to the highest surface feature of the moon centered at the origin of the selenocentric inertial coordinate system. This also creates a forbidden area from which the initial state, $\mathbf{x}(0)$ may not begin. Although not a constraint in the classical sense, it is a logical restriction that must be obeyed to achieve convergence.

Without imposing control constraints that necessitate a non-generalized spacecraft model, thrust limits can be established so that $0 \leqslant T \leqslant T_{max}$ and the angle restrictions of $0 \leqslant \alpha \leqslant 2\pi$ and $0 \leqslant \beta \leqslant 2\pi$ apply as assumed control constraints. However, modification of the thrust constraint to $T \leqslant T_{max}$ or even $T = T_{max}$ replicates an element of low-thrust mission design where limited throttle levels and a high reliability rating once engaged but a low reliability of initial ignition may change the motivators of mission designers. Consideration of these modifications might greatly change the problem by eliminating thrust-coast arc trajectories (the low-thrust version of "bang-bang" control) and demand a more rapid resolution as the continuous acceleration from the thruster will push the spacecraft to exit velocity faster than if the candidate trajectory included coasting arcs where $T = 0$. Furthermore, a Lagrangian cost term that captured and penalized the number of on-off cycles could be included in the cost functional. Balancing the impacts of these control constraints and control cost penalties provides a more nuanced metric to assess the envelope of feasible outcomes and makes an initial effort to quantify some of the "softer" elements of mission classification like "complexity."

In this approach, the problem ultimately revolves around the determination of some candidate optimal t_0 that represents the minimum feasible initial state. The conceptual approach is simple; solve the problem backwards. In practice this may require some tricks and translations to match the numerical equations that can be ingested by a solver. The simplest method of addressing this issue is to estimate the cost of transfer from the initial position to the transfer position. Because the problem is intentionally scoped within a generalized trajectory, system specifics regarding external and internal hardware is less critical than an understanding of the physics of the problem and the ability to define and impose hard and soft restrictions during problem development.

2.3 Model Construction and Assessment

This section describes the dynamical model of the system. As previously discussed, while the broader system includes multi-body dynamics, the optimization model will consist solely of restricted two-body dynamics and the impact of the broader multi-body dynamics will be linked to the problem through the specification of endpoint conditions. In this specific case we will capture the impacts of this difference by demanding a certain final state that meets patch-point accuracy for the broader model to describe the multi-body transfer section. Furthermore, this effort assumes a negligible change in spacecraft mass. As mentioned in Section 1, Involvement of the thrust in dynamics has been expanded as the magnitude of the resulting acceleration of the force. As previously discussed, the hypothetical results of propagation of a state vector are identical to those from the integration of Gauss's equations. Differences can be expected from the buildup of rounding errors over long integration times. Because we are dealing with long integration times for the continuous, low-thrust problem, we will conduct integration of Gauss's equations rather than propagating the state vector and then converting to classical orbital elements. Consistency here will save error and computational time throughout the assessment. The one exception to this is in the generation of "mechanic plots" designed to offer the human mechanic a quick visualization of the orbit in three dimensions. Because accuracy in reading this information is restricted to that of the mechanic's eyeball, path histories may be converted from Gauss's equations to state vector for ease of plotting only [28], [34].

In the general sense, the system dynamics are shown in the simplification in Figure 2.4 to the second order ordinary differential equation $\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r}$. This relationship is then expressed

in Gauss's equations as six, first-order, nonlinear, inseparable, autonomous ordinary differential equations. Theoretically these dynamics produce identical results compared to the propagation of a state vector, though rounding errors can cause variation.

2.3.1 Finalized Dynamical Model

While an ideal environment might allow the use of a complete multibody dynamics with dozens of local perturbations resulting from variation in the gravitational potential, atmospheric drag, solar radiation pressure, and other sources, it is unrealistic to expect an optimizer (whether the "engineer" or algorithm) to deal with such computationally expensive and conceptually complex equations. The key to achieving meaningful outcomes from optimization will be identifying "good enough" dynamics sets for use in optimization applications, and an accepted "truth" dynamics set to assess usability of candidate trajectories. As outlined in the results, this effort includes a high accuracy truth model, a medium accuracy J_2 or "Perturbed" model, and a lower accuracy "point mass" model. These models allow a mechanic to dial up or down the accuracy of the system to stress the candidate control schema, and allow a higher probability of early convergence in candidate control search in early analysis. An advantage to using Systems Tool Kit in this case includes the ability to also produce path and control histories for lower complexity models. Proper initial bounding in model development will allow intermediate troubleshooting to confirm that numeric results generated from MATLAB are confirmed in Systems Tool Kit and that there are no "translation" errors.

"Truth" model: The Systems Tool Kit High-Precision Orbit Propagator (High Precision Orbit Propagator) represents the highest accuracy model available to most academic groups and is marketed by the creators to be useful beyond lunar orbit using advanced modeling of complex gravitational zonal harmonics (21x21) of the central body, solar radiation pressure with complex shadow occlusion modeling, multi-body effects, and atmospheric drag. While it is possible to derive simplified expressions of zonal harmonic perturbations [35] such efforts are significantly computationally demanding and, because of the complexity of the expressions, extremely prone to error in code development. This model is also trusted as accurate for high precision simulations by National Aeronautics and Space Administration [19]. Because an existing commercial model is both available and reliable, it will be used as the truth model. Truth model settings for High Precision Orbit Propagator in this analysis

will use all perturbative forces, with Runge-Kutta Fehlberg integration of order 7/8 of Gauss-Jackson equations of motion, and rely on the variation of parameters and regularized time. Specific matching Systems Tool Kit settings for this model can be found online [36]. No optimization will be done in this model, however, candidate solutions can be evaluated in this model for suitability analysis.

J_2 or "Perturbed" model: Recognized variations of Gauss's equations exist to include the second harmonic of the gravitational potential of the central body. Although research indicates that low-lunar orbits should be modeled with perturbations out to the fourth harmonic or J_4 term [2], options to include them in the equations of motion involve continued expansion of a modified Hamiltonian, the development of a disturbing function, elimination of short-period perturbations, calculation of long-period terms, and potential coupling effects for each expansion of the Legendre polynomial. [37] This process is both complex and lengthy, and while the result is accuracy nearing that of High Precision Orbit Propagator, for an intermediate case it is not advised. Additionally, the precise measurement of parameters like mass, surface area, and relative orientation required for the calculation of higher order perturbations make such calculations difficult when dealing with a generalized point mass spacecraft.

$$\frac{da}{dt} = \frac{2a^2}{h}\left(\epsilon\cos\beta\cos\alpha - J_2\mu\frac{R^2}{a^4}3\sin^2 i\sin\nu\cos nu\right) \tag{2.10}$$

$$\frac{di}{dt} = \frac{a}{h}\cos\nu\left(\epsilon\sin\beta - J_2\mu\frac{R^2}{a^4}3\sin i\cos i\sin\nu\right) \tag{2.11}$$

$$\frac{d\omega}{dt} = \frac{a\sin\nu}{h\sin i}\cos\nu\left(\epsilon\sin\beta - J_2\mu\frac{R^2}{a^4}3\sin i\cos i\sin\nu\right) \tag{2.12}$$

Equations 2.10, 2.11, and 2.12 *sufficiently* reflect the perturbed dynamics of a restricted two-body environment with zonal harmonic perturbation effects in simulation without significant complication to the equations of motion or imposition of unacceptable computational costs. Derived from Gauss's equations by DiCarlo and Vasile, the equations for $\frac{da}{dt}$, $\frac{di}{dt}$, and $\frac{d\Omega}{dt}$ allow for orbital procession over time [4].

These modified equations now involve the product of a measured constant, J_2, the impact

of the gravitational potential of the central body and the geometry of the spacecraft to the specific relative face of the central body. This model moves beyond the point-mass approach and allows for a nonspherical, nonuniform central body. These equations can be particularly useful once convergence on simple models has been achieved, evaluating time requirements for maneuvers that involve $\Delta\Omega$. They also serve to support intermediate trouble shooting should significant discrepancies emerge between the lower accuracy engineering model and High Precision Orbit Propagator results.

Point Mass model: Finally, the simplest version of the dynamics, and the one used extensively in this study integrates Equations 2.10 through 2.12. However, as previously discussed, for long, low-thrust influenced orbits, rounding errors in conversion between the two forms may result in minor variance. This model does not allow for precession or any natural Δi as observed in cases where $i_0 \neq 0$. All of the more advanced models are based on these equations, though some require advanced mathematical handling to include the extended perturbative forces.

Control Vector: As described in the dynamics introduction and depicted in Figure 1.2, the control components of the low-thrust force are described in the context of the thrust emanating from the spacecraft. In this system, the control variables include the relative direction and magnitude of the thrust. These were previously labeled as angles α and β, and the magnitude of acceleration ϵ. In the real world, these might be more complex and composed of parameters that represent the specific impulse of an engine, the mass flow, attitude of the spacecraft, or even a thrust vector control pointing system in a more advanced system. In the simplified engineering model, we assume a generic spacecraft of negligible mass. The decomposition of the force into components is shown in Equation 13.

$$\mathbf{f} = \begin{bmatrix} f_R \\ f_T \\ f_N \end{bmatrix} = \begin{bmatrix} \epsilon \cos\beta \sin\alpha \\ \epsilon \cos\beta \cos\alpha \\ \epsilon \sin\beta \end{bmatrix} \tag{2.13}$$

Which provides a control vector $\mathbf{u}$ where $\mathbb{U} \in \mathbb{R}^3$ of:

$$\mathbf{u} = [\varepsilon, \alpha, \beta]^T \tag{2.14}$$

However, because the engine is either on or off only, ϵ can be further defined by:

$$\epsilon = \begin{cases} 0 & \text{if propulsion OFF} \\ \epsilon_{const} & \text{if propulsion ON} \end{cases} \tag{2.15}$$

State Vector: Though classical dynamics would represent the state vector $\mathbf{x}$ where $\mathbb{X} \in \mathbb{R}^6$ as derived from Figure 3:

$$\mathbf{x} = \begin{bmatrix} r_R & r_T & r_N & v_R & v_T & v_N \end{bmatrix}^T \tag{2.16}$$

Our use of Gauss's equations converts the state to the following while maintaining $\mathbb{X} \in \mathbb{R}^6$:

$$\mathbf{x} = \begin{bmatrix} a & e & i & \Omega & \omega & \theta \end{bmatrix}^T \tag{2.17}$$

Parameter Vector: Within the scope of the discussed and transparent dynamics, that is, for models *as complex as* the J_2 perturbation model, the parameters include the gravitational parameter,μ of the central body, the radius of the central body, R and the value of the harmonic coefficient J_2 [8]. All of these are considered constants. Finally, the magnitude of the acceleration from the thrust, ϵ, is set for each low-thrust engine. This effort will not consider variable thrust engines; it is either on or off with a fixed thrust. The fixed parameters for this effort are outlined in Table 2.1, though as mentioned they could include final state information and a particular ϵ for a specific trajectory.

Table 2.1. Dynamics Parameters

Parameter	Value	Units	Description
$\mu_{☾}$	4902.799108	km^3/s^2	Gravitational parameter
$R_{☾}$	1738.0	km	Radius of central body
$J_{2☾}$	$0.909010949481 \times 10^{-04}$	km^5/s^2	Second Zonal Harmonic
ϵ_{const}	User Defined	km/s^2	Magnitude of thrust acceleration

2.3.2 Model Validation

There exist three accepted analytic control solutions to achieve maximum change of select orbital elements. These equations for the maximum $\Delta\overline{a}$, $\Delta\bar{i}$, and $\Delta\overline{\Omega}$ with $\overline{e} = 0$. The control angles for these cases are outlined as well [4]. These analytic control equations are fully developed and assessed in Chapter 4.

Table 2.2. Known Maximum Paths

ΔCOE_{MAX}	Control Angles
$\Delta\overline{a}_{MAX}$	$\alpha = \arctan\left(\frac{e\sin\theta}{1+e\cos\theta}\right) = \gamma, \beta = 0$
$\Delta\bar{i}_{MAX}$	$\beta = sgn(\cos\nu)sgn\left(\bar{i}_f - \bar{i}_0\right)\frac{\pi}{2}$
$\Delta\overline{\Omega}_{MAX}$	$\beta = sgn\left(\overline{\Omega}_f - \overline{\Omega}_0\right))sgn\left(\sin\nu\right)\frac{\pi}{2}$

Because these extreme cases have been derived from analytical solutions, estimates also exist that can closely approximate the cost of a maneuver in terms of ΔV.

$$\Delta V = \epsilon ToF = \frac{2\pi\left|\sqrt{\frac{\mu}{\overline{a}_0}} - \sqrt{\frac{\mu}{\overline{a}_f}}\right|}{(1-\overline{e}^2)f(\overline{e})} \tag{2.18}$$

$$\Delta V = \frac{\pi}{2}\sqrt{\frac{\mu}{\overline{a}}}\Delta\bar{i} \tag{2.19}$$

$$\Delta V = \frac{\pi}{2}\sqrt{\frac{\mu}{\overline{a}}}\Delta\overline{\Omega}\sin\bar{i} \tag{2.20}$$

where the term $f(e)$ represents:

$$f(\overline{e}) = \frac{2}{1-\overline{e}}E_{Ic}\left(\frac{4\overline{e}}{(1+\overline{e})^2}\right) + \frac{2}{1+\overline{e}}F_{Ic}\left(\frac{4\overline{e}}{(1+\overline{e})^2}\right) \tag{2.21}$$

where F_{Ic} and E_{Ic} are the complete first and second elliptic integrals.

$$E_{Ic}(m) = E_1(\frac{\pi}{2}, m) \tag{2.22}$$

$$F_{Ic}(m) = F_1(\frac{\pi}{2}, m) \tag{2.23}$$

While these estimates are valuable for validating the numeric integration methods used by a solver, the solver tool itself can play a role in the accuracy of simulation for these dynamics. Because the optimal trajectory will eventually be computed in MATLAB, the pre-installed solvers represent an efficient and reliable way to solve Gauss's equations through integration. Because Gauss's equations are considered "stiff" differential equations, results from "general" or "all-purpose" ordinary differential equation solvers can become less accurate as the stiffness of the ordinary differential equation increases. For Gauss's equations, the stiffness is the result of fast and slow developing terms within the equations. Because these dynamics are for elliptical orbits, velocity related elements will undergo significant variation in the rates of change at apoapsis and periapsis. As the eccentricity of the studied orbit increases, so too does the stiffness. At the same time, there are fast changing dynamics like the change of semi-major axis during the maximum $\Delta\overline{a}$ maneuver as described previously and any slow-evolution elements like Ω as a result of J_2 perturbations. General purpose solvers will have increasing difficulty selecting appropriate time intervals that balance computational accuracy with efficiency. As such, various ordinary differential equation solvers will be tested with the dynamics in maximum variation experiments. An identical control theory will be applied to various models and the results will be compared. Where MATLAB was used to solve the ordinary differential equations, code functions are provided in Appendix 1. Systems Tool Kit data was generated using Astrogator, and state histories exported through the reports manager tool. These histories were reformatted as Arrays in MATLAB and imported for analysis. Control histories can also be provided to Systems Tool Kit by converting a MATLAB control history as a stored array to a Systems Tool Kit control history file. This can either be done by converting the angles α and β in the RTN/RIC/RSW frame to quaternions, or by defining the angles for the engine pointing dynamics in Systems Tool Kit. Both options should be considered viable patching methods for optimal control study (not to be confused with trajectory patch points as used in the method of patched conics).

Model Experimentation: The model is validated in MATLAB using two variations of a known analytic solution for the maximum $\Delta\overline{a}$. One of these is conducted with low eccentricity ($e_0 \leqslant 0.1$) and high eccentricity ($e_0 \leqslant 0.7$). Parameter setting for ϵ_{const} is arbitrary but selected to minimize the transfer time (and memory costs to store and evaluate position and control histories) without accelerating the satellite to or beyond lunar exit

velocity. Range of a_0 and a_f are likewise arbitrary but coupled to the choice of ϵ_{const} with the additional conditions that $a_0 \geqslant 100km$ and $a_f \leqslant$ sphere of influence$_{☾}$. Where necessary for insight, multiple tests of the same variation are conducted with different tools required to solve the system of equations. All variations are compared across analytic, various numeric, and "truth" models. Both the solving of a problem (generating a control) and the simulation of a candidate control are tested for both variations of the known maximum transfer case. Additional future validation could include variational testing of the known cases for $\Delta\overline{\Omega}$ with $\overline{e} = 0$. Particularly because of the singularity produced by the restriction of $\overline{e} = 0$ in the initial condition of known solution, further testing and analysis is advisable. Where these situations are demanded, it is advisable to manually "offset" the point-mass simulation to some initial eccentricity that is low but mathematically $\gg 0$ based on the conditions of the coupled states. This is acceptable since candidate optimal solutions are already being chosen in an "engineering" fidelity model and are only intended to seed initial guesses for higher fidelity solving or provide candidate control histories that can be modified later to remove the manual shift of eccentricity.

The dynamics are described in a MATLAB Function that use Gauss's six, first-order ordinary differential equations with some ordinary differential equation solver. It also requires the passing of four additional elements, $\alpha, \beta, \mu, \epsilon_{const}$. Because only μ and ϵ_{const} are parameters, and because known maximal cases require a varying control, it is practical to build the step-by-step calculation of α and β within the ordinary differential equation function in these cases (the core ordinary differential equation function file is included in the appendix). When validating a control history, it is convenient to pass the history as a parameter, though this requires an interpolation schema to match the control value with that of the solver if the tolerance settings do not perfectly match. Because the constant, low-thrust orbit maneuver often takes several orbits to complete, control histories of the angles α and β often demonstrate sinusoidal histories with the potential for high frequency oscillation near the central body and lower frequency oscillation at higher altitudes. As such, linear interpolation schemes at time steps near relative maxima and minima of the control history can fail to provide sufficiently accurate results. It is also important to note that when Gauss's equations describe orbits with high eccentricity, they demonstrate sufficient stiffness as ordinary differential equations that general-purpose ordinary differential equation solvers may be inadequate.

MATLAB solvers ode45, ode15s, and ode23s are used to generate a control history for the known case, then these controls are propagated through all four solvers to assess relative performance across combined use cases. Linear, spline, and quadratic interpolation will be attempted to assess interpolation performance requirements across solvers.

Test 1 Description:

$$
\begin{cases}
\text{Dynamics} & \frac{da}{dt} = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
& \frac{de}{dt} = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + \left[(p+r)\cos\theta + re\right]\cos\alpha\right) \\
& \frac{di}{dt} = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
& \frac{d\Omega}{dt} = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
& \frac{d\omega}{dt} = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
& \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
\text{Endpoints} & a_0 = 2000 \quad [km] \\
& e_0 = 0.1 \\
& i_0 = 0.1 \quad [rad] \\
& \Omega = 0.1 \quad [rad] \\
& \omega = 0.1 \quad [rad] \\
& \theta = \pi \quad [rad] \\
& a_f = 4300 \quad [km] \\
\text{Parameters} & \mu = \mu_{☾} \\
& \epsilon = 10 \times 10^{-7} \quad [km/s^2] \\
\text{Controls} & \alpha = \arctan\left(\frac{e\sin\theta}{1+e\cos\theta}\right) = \gamma \quad [rad] \\
& \beta = 0 \quad [rad]
\end{cases}
$$

The following tables demonstrate the results of a study intended to capture relative performance data across models for a sample selenocentric orbit as described by the following control strategy and endpoints. At this stage, without a defined cost functional, qualitative assessment of the simulation is centered on $\$DeltaV$ as a measure of performance. Accuracy will be judged to the "truth" model system, meaning that point-mass models will be compared against each other for precision but against the Systems Tool Kit point-mass value for accuracy. Comparison sources are drawn from the integration of Gauss's Variational Equations through ODE45 with relative and absolute tolerances set to 10^{-12}, ODE15s with

default variable tolerance, and System Tool Kit using "point mass" dynamics.

Table 2.3 displays the findings of various solvers to solve the ordinary differential equations and produce control histories. ΔV is measured in km/s^2, a_f in km, $\delta\Delta V$ by the result of $\frac{|\Delta V_{SystemsToolKit} - \Delta V_{solver}|}{\Delta V_{SystemsToolKit}}$, datapoints as the number of double-precision floating point rows in the solution array for the solver, and computational time in seconds. $ode45_d$ represents the default settings on $ode45_{ht}$ represents a RelTol and AbsTol setting of 10×10^{-12}. $ode15s$ and $ode23s$ were executed at default tolerance settings only as they are designed to function as variable-step, variable-order solvers for stiff ordinary differential equations. Computational time has been normalized for all of these studies. Items of particular value for this study include the variation in ΔV from Systems Tool Kit as the truth model, number of necessary datapoints, and computational time.

Table 2.3. Various Solver Solution Results for Test 1

Solver	ΔV	e^f	a^f	$\delta\Delta V$	Datapoints	Comp. Time
Truth	0.49853	0.075353	4300	0	0	0
Analytic	0.49901	0.1	4300	0.00096577	1	0.00119
$ode45_d$	0.49891	0.060960	4300	0.00077162	469	0.02491
$ode45_{ht}$	0.49890	0.060868	4300	0.00075898	8889	0.08267
$ode15s$	0.49889	0.061124	4300	0.00073527	697	0.07658
$ode23s$	0.49892	0.060530	4300	0.00078126	693	0.10646

Tables 2.4 through 2.7 show the results of a specific ordinary differential equation solver's control history solution propagated back through various ordinary differential equation solvers as a simulation. Datapoint analysis was eliminated as a performance metric as the author was unable to achieve reliable interpolation with an ordinary differential equation solver derived time steps. As such, all interpolation results shown are for linearly interpolated controls with a $\Delta t = 0.1sec$. Because the control history dictates the t_f of the solution, $\delta\Delta V$ is replaced by $\delta\overline{a}_f$ by the result of $\left|\overline{a}^f_{SystemsToolKit} - \overline{a}^f_{solver}\right|$ in km. Computational time was assessed for the analytic solution along with various control source assessments to account for the variation of computer resources during each iteration. This normalizes the computational times across solvers within the same control simulation analysis.

Table 2.4. $ode45_d$ Derived Control Simulation Results for Test 1

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.49853	0.075353	4300.0000	0.000000	n/a
Analytic	0.49901	0.100000	4300.0000	0.000000	0.00365
$ode45_d$	0.49891	0.060503	4299.9561	0.043886	21.2548
$ode45_{ht}$	0.49891	0.060579	4300.0788	0.078787	16.0383
$ode15s$	0.49891	0.060848	4300.1578	0.157830	25.6362
$ode23s$	0.49891	0.060239	4299.9828	0.017160	10.4979

Table 2.5. $ode45_{ht}$ Derived Control Simulation Results for Test 1

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.49853	0.075353	4300	0	n/a
Analytic	0.49901	0.1	4300	0	0.00687
$ode45_d$	0.4989	0.060903	4299.9289	0.071061	18.4854
$ode45_{ht}$	0.4989	0.060867	4299.9995	0.000512	28.8630
$ode15s$	0.4989	0.060794	4299.9905	0.009488	23.9638
$ode23s$	0.4989	0.060560	4299.9154	0.084616	11.1960

Table 2.6. $ode15s$ Derived Control Simulation Results for Test 1

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.49853	0.075353	4300	0	n/a
Analytic	0.49901	0.1	4300	0	0.0039578
$ode45_d$	0.49889	0.060792	4299.8621	0.13785	21.2825
$ode45_{ht}$	0.49889	0.060775	4299.9158	0.084174	16.0574
$ode15s$	0.49889	0.061172	4300.0717	0.071729	30.0521
$ode23s$	0.49889	0.06043	4299.8154	0.1846	14.4322

Table 2.7. $ode23s$ Derived Control Simulation Results for Test 1

Solver	ΔV	e_f	a_f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.49853	0.075353	4300	0	n/a
Analytic	0.49901	0.1	4300	0	0.0051538
$ode45_d$	0.49892	0.060743	4300.0584	0.058368	18.7613
$ode45_{ht}$	0.49892	0.060752	4300.1047	0.10473	17.0582
$ode15s$	0.49892	0.060682	4300.0664	0.066445	29.1171
$ode23s$	0.49892	0.060414	4300.0239	0.02393	11.2042

Test 2 Description:

$$
\begin{cases}
\text{Dynamics} & \frac{da}{dt} = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
& \frac{de}{dt} = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
& \frac{di}{dt} = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
& \frac{d\Omega}{dt} = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
& \frac{d\omega}{dt} = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
& \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
\text{Endpoints} & a_0 = 50000 \quad [km] \\
& e_0 = 0.7 \\
& i_0 = 0.1 \quad [rad] \\
& \Omega = 0.1 \quad [rad] \\
& \omega = 0.1 \quad [rad] \\
& \theta = \pi \quad [rad] \\
& a_f = 66000 \quad [km] \\
\text{Parameters} & \mu = \mu_{☾} \\
& \epsilon = 10 \times 10^{-10} \quad [km/s^2] \\
\text{Controls} & \alpha = \arctan\left(\frac{e\sin\theta}{1+e\cos\theta}\right) = \gamma \quad [rad] \\
& \beta = 0 \quad [rad]
\end{cases}
$$

Test 2 stresses the model with a high eccentricity orbit. As previously discussed, an $e_0 \geqslant 0.5$ increases the stiffness of Gauss's equations, as such solver selection and settings become even

more important than with Test 1. This test begins in a moderately elliptical medium-altitude lunar orbit and ends with the semi-major axis equal to the radius of lunar sphere of influence. While this test exceeds lunar sphere of influence, it is useful to demonstrate the sensitivity of the model to variations beyond the most stressing case anticipated. Additionally, the low force of the moon's gravity at ranges out to Lunar sphere of influence $(66,000km)$ make low-thrust transfers with $\epsilon \leqslant= 10 \times 10^{-7}[km/s^2]$ do not provide a long enough transfer window to suitably stress the model. The complete endpoint conditions and parameters for Test 2 are also outlined. Test 2 also used spline interpolation as a default to assess if there were any issues increasing the complexity of computation at each step.

Table 2.8 displays the findings of various solvers to solve the ordinary differential equation's and produce control histories. ΔV is measured in km/s^2, a_f in km, $\delta\Delta V$ by the result of $\frac{|\Delta V_{SystemsToolKit} - \Delta V_{solver}|}{\Delta V_{SystemsToolKit}}$, datapoints as the number of double-precision floating point rows in the solution array for the solver, and computational time in seconds. $ode45_d$ represents the default settings on $ode45_{ht}$ represents a RelTol and AbsTol setting of 10×10^{-12} interpolation of the control propagated through the solvers in Tables 2.9 through 2.12 with a one-second interpolation scheme as $0.1sec$ histories exceed MATLAB's automatic restriction of 15 GB for a single array due to the extreme transfer time.

Table 2.8. Various Solver Solution Results Time for Test 2

Solver	ΔV	e^f	a^f	$\delta\Delta V$ from Truth	Datapoints	Comp Time
Truth	0.046307	0.62709	66000	0	0	0
Analytic	0.047014	0.7	66000	0.015265	1	0.0007021
$ode45_d$	0.044066	0.63518	66000	0.048396	853	0.011636
$ode45_{ht}$	0.046011	0.62304	66000	0.0063962	19245	0.13168
$ode15s$	0.046153	0.62044	66000	0.0033275	768	0.057121
$ode23s$	0.046	0.6238	66000	0.006637	422	0.10755

Tables 2.9 through 2.12 show the results of a specific ordinary differential equation solver's control history solution propagated back through various ordinary differential equation solvers as a simulation. Datapoint analysis was eliminated as a performance metric as the author was unable to achieve reliable interpolation with ordinary differential equation

solver derived time steps. As such, all interpolation results shown are for linearly interpolated controls with a $\Delta t = 0.1sec$. Because the control history dictates the t_f of the solution, $\delta\Delta V$ is replaced by $\delta\overline{a}_f$ by the result of $\left|\overline{a}^f_{SystemsToolKit} - \overline{a}^f_{solver}\right|$ in km. Computational time was assessed for the analytic solution along with various control source assessments to account for the variation of computer resources during each iteration. This normalizes the computational times across solvers within the same control simulation analysis.

Table 2.9. $ode45_d$ Derived Control Simulation Results for Test 2

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp Time
Truth	0.046307	0.62709	66000	0	n/a
Analytic	0.047014	0.7	66000	0	0.0022495
$ode45_d$	0.0432	0.62758	61375.0381	4624.9619	147.0527
$ode45_{ht}$	0.0432	0.6083	60635.8163	5364.1837	128.8457
$ode15s$	0.0432	0.61159	60978.9526	5021.0474	207.0396
$ode23s$	0.0432	0.614	61716.2628	4283.7372	114.4991

Table 2.10. $ode45_{ht}$ Derived Control Simulation Results for Test 2

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.046307	0.62709	66000	0	n/a
Analytic	0.047014	0.7	66000	0	0.0067063
$ode45_d$	0.0432	0.63084	62173.0312	3826.9688	135.0193
$ode45_{ht}$	0.0432	0.62768	64773.8549	1226.1451	156.5232
$ode15s$	0.0432	0.61554	62707.0647	3292.9353	211.8152
$ode23s$	0.0432	0.62483	64098.388	1901.612	92.8065

Table 2.11. $ode15s$ Derived Control Simulation Results for Test 2

Solver	ΔV	e^f	a^f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.046307	0.62709	66000	0	n/a
Analytic	0.047014	0.7	66000	0	0.00757
$ode45_d$	0.0432	0.63084	61125.4395	4874.5605	142.5307
$ode45_{ht}$	0.0432	0.62358	64139.2655	1860.7345	134.5524
$ode15s$	0.0432	0.61004	61196.8214	4803.1786	244.3319
$ode23s$	0.0432	0.62021	63307.7643	2692.2357	110.8187

Table 2.12. $ode23s$ Derived Control Simulation Results for Test 2

Solver	ΔV	e_f	a_f	$\delta\overline{a}^f$ from Truth	Comp. Time
Truth	0.046307	0.62709	66000	0	n/a
Analytic	0.047014	0.7	66000	65999.953	0.0076955
$ode45_d$	0.0432	0.62229	61963.9744	4036.0256	157.0687
$ode45_{ht}$	0.0432	0.61143	61404.7624	4595.2376	134.2783
$ode15s$	0.0432	0.61263	62054.4669	3945.5331	248.733
$ode23s$	0.0432	0.62869	64838.8613	1161.1387	110.008

2.3.3 Analysis of Test Results

First order inspection of the test results suggests a strong correlation between ordinary differential equation stiffness and initial eccentricity as suggested by mathematical theory. For most orbits in this problem set, ode23s will likely provide the best mix of reliability, memory consumption, accuracy, and computational expense. Gauss's equations are inherently stiff, and this stiffness only increases at higher eccentricity. This is shown in the relative accuracy in $\overline{a}^f$ of the stiff solvers over general purpose solvers in test 2. The best accuracy in final $\overline{a}^f$ is seen when using ode45's default settings to derive a control history and using ode23s with default settings as the solver for the simulation engine. As previously mentioned, interpolation method also plays a significant role in accuracy of a propagated control. With the

use of a spline method of interpolation, no observable improvements are seen in the best case. When applied in the worst performance case, where ode15s' controls are propagated through ode23s, spline interpolation results in a four-second computational time increase, with a further deviation from the truth case of $0.0094km$.These patterns may be expected to continue as initial altitude is lowered, as the controls will display increasing frequency that will lose accuracy near the local minima and maxima. It appears linear interpolation is sufficient for sample problems, lower orbit cases, or cases where lower computational times are necessary should apply non-linear interpolation in simulations.

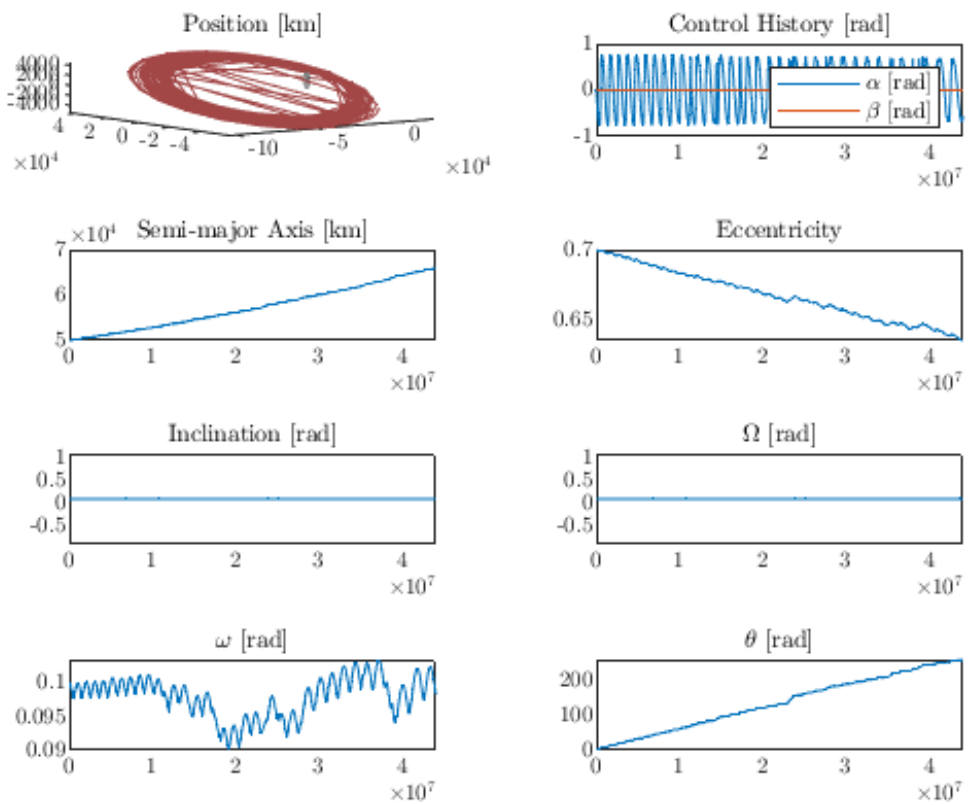


Figure 2.5. Poor Ordinary Differential Equation Solutions for Elliptical Cases and ode45 with Default Settings

As shown in Figure 2.5, ode45 under default settings produces poor path and control histories, resulting in observably non-periodic behavior in the time histories of both the

semi-major axis and eccentricity. While one solution may be to manually increase the fidelity of the model by decreasing the allowed tolerances as shown in Figure 7 to produce a smooth, spiraling transfer history, a potentially more responsible approach is using a solver designed for stiff odes. While both ode15s and ode23s show "choppier" histories when linearly interpolated for plotting the satellite's path, the control and path histories of the two observed COEs more closely match the truth model and produce more reliable results when propagated.

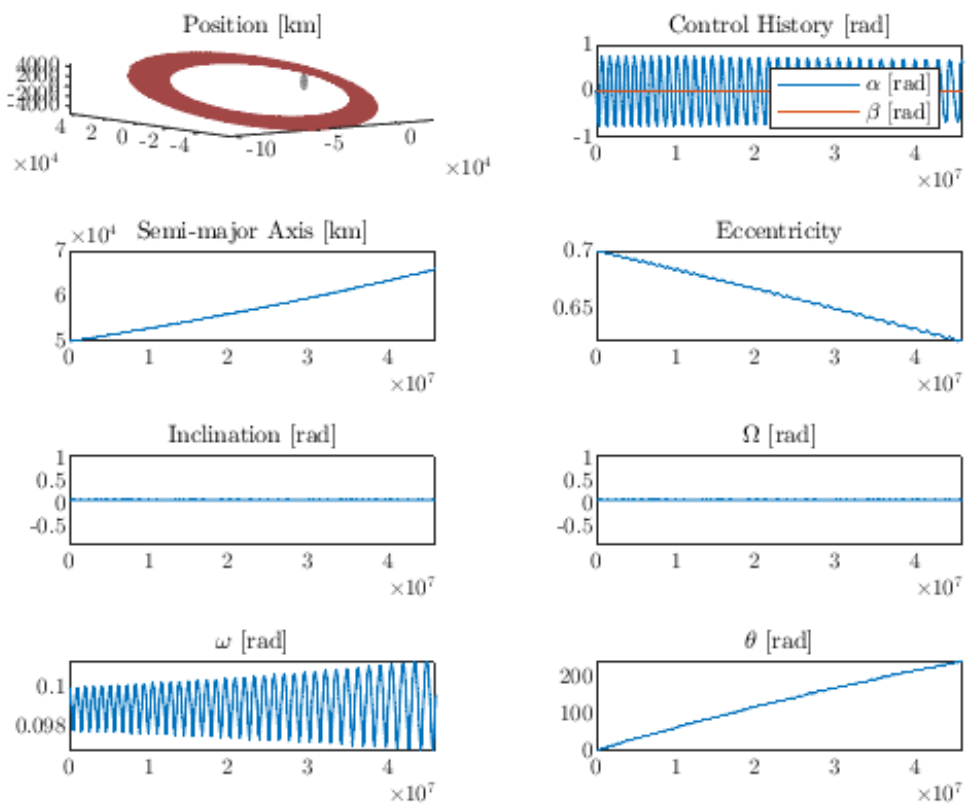


Figure 2.6. Extreme Tolerance Settings with ode45 to Force a Smooth Path

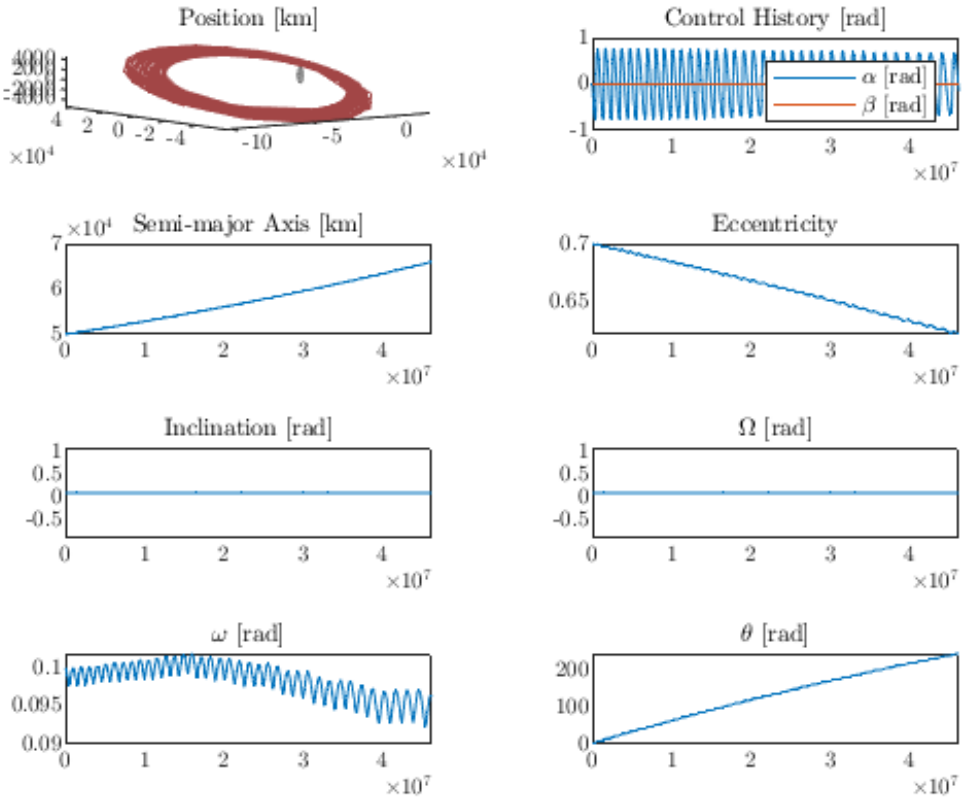


Figure 2.7. Acceptable Solutions from a Purpose Built Solver, ode15s

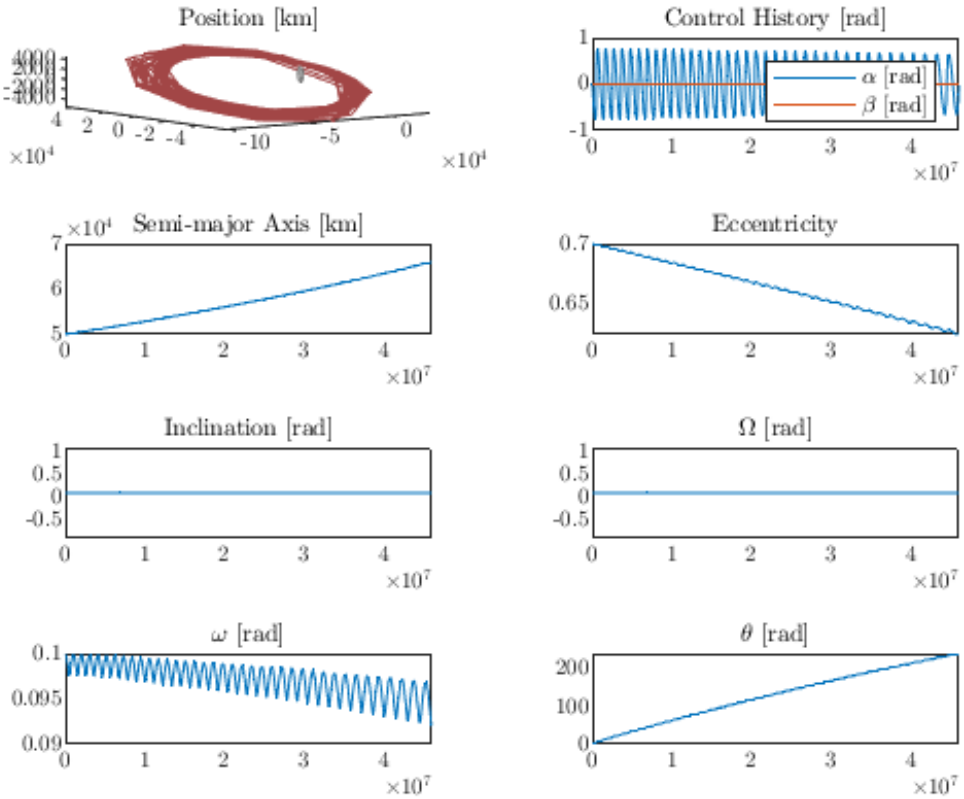


Figure 2.8. Acceptable Solutions from a Purpose Built Solver, ode23s

From the perspective of the "mechanic," the elements provided in Figures 2.6 through 2.9 show the two competing concepts at work. Firstly, the mechanic is seeking a path history of both the state and controls that are accurate enough to hold up to application either in simulation or as intended by control engineers. At the same time, computational efficiency and memory costs must strongly be considered, particularly for cases seeking an eventual distributed or on-orbit capability. Shudders in path histories of classical orbital elements (the true states as outlined by the dynamics) are cause for concern in a solver, while visual disturbances in the orbit's "picture" on the top left of these figures is purely the result of algorithmic step sizing and not of true position. Plotting the position history is purely meant as a sanity check for the operator. The key idea is finding the best way to get an accurate signal. A simulation report was compiled by using ode45 as a simulator under default settings and a *good* control history from ode15s in solving Test 2.

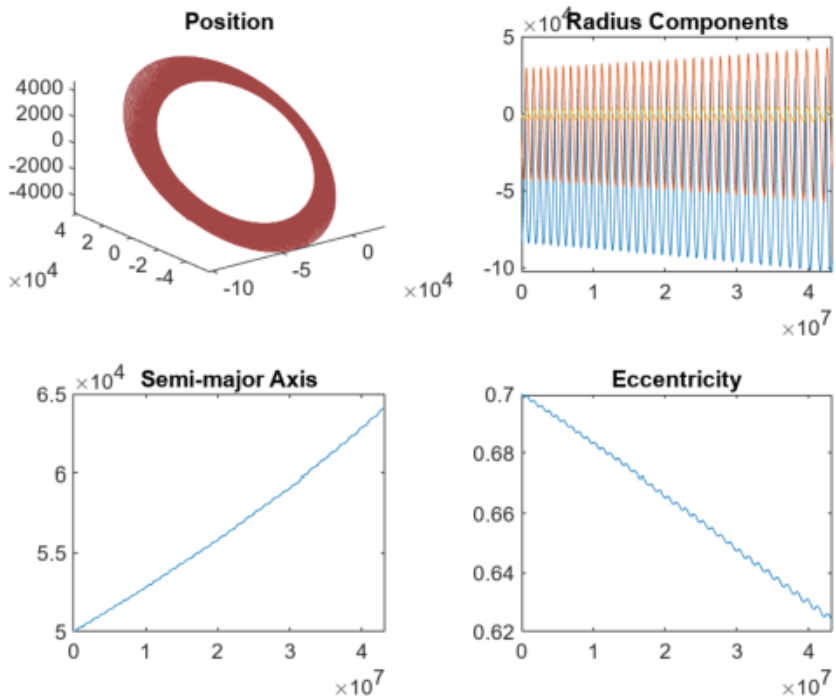


Figure 2.9. A Good Control History Simulated through a Coarse Solver

Early Development of Constraints: Though full constraints will be further developed in the following sections, it is useful to point out some identified constraints uncovered by this section.

Scoped by the broader context of these dynamics representing only a portion of an overall trajectory, the final time, t_f, and all six elements of the state, $a, e, i, \Omega, \omega, \theta$ represent endpoint constraints. While other generic optimal control problems with the same dynamics may allow for free states, because the overall trajectory demanded by third-body dynamics requires complete specificity. Also, the satellite must not crash into the moon. This can be modeled as a path constraint consisting of a sphere with radius equal to some margin plus the radius of the moon. Practically, it will be wise to set this margin $\geqslant 100km$ as orbits lower than this require perturbation models of the require models of at least a 50×50 field term to *sufficiently* account for the actual impacts of the lunar gravity field at this altitude [38].

CHAPTER 3:
Optimality of Core Analytical Solutions

3.1 Background

Chapter 2 introduced a literature of work related to analytically derived control schemas for maneuvers and the development and validation of a model that simulates these control schemas. Chapter 3 focuses now on the use of numerical methods to discern the optimality of these control schemas. Why is this important? The final algorithm assumes that the noncooperative actor is seeking the most optimal transfer in an effort to minimize fuel and time costs. Specialist software is required to run the associated MATLAB code and to implement the findings in Systems Tool Kit.

Within the scope of this work, feasibility is taken as the first reigning optimal solution. It serves as a validation of optimality for supporting equations and several examples which validate the underlying model. This section contains some advanced elements related to optimal control. Supporting references are listed throughout the chapter, but Ross's *A Primer on Pontryagin's Principle in Optimal Control* is immediately recommended for any interested reader [13]. Why is this important? The final algorithm assumes that the noncooperative actor is seeking the most optimal transfer in an effort to minimize fuel and time costs. Within the scope of this work, feasibility is taken as the first reigning optimal solution.

The initial guesses used to generate the problem families and the specific cases discussed in this chapter come from mathematically generated analytical solutions. Although for trivial cases, closed-form solutions can be shown to analytically prove global optimality, the introduction of small changes such as removing the restriction of circular orbits immediately dissolves the confidence that the analytical solution is globally optimal and forces the mission designer or analyst into a search for regional optimality. In this world, the reigning candidate solution is assumed to be optimal as long as it can demonstrate the necessary conditions for optimality. This chapter relies on the use of Pontryagin's principle to form and analyze the necessary, though not sufficient conditions for optimality.

As introduced in Chapter 2, the use of classical orbital elements with the $\overline{overline}$ structure ($\overline{a}\,\overline{e}\,\overline{i}\,\overline{\Omega}\,\overline{\omega}\,\theta$) denote the *averaged* value of that element over the course of an orbit as opposed to the instantaneous value of that element ($a\ e\ i\ \Omega\ \omega\ \theta$) at a specific time [15].

3.2 Introduction to Numerical Methods

The scope of this chapter is narrow and deep. It presupposes a requisite degree of understanding of challenging mathematical theories. Readers familiar with working with real problems in convex optimization described broadly by Peypouquet [39], can skim through as if grading a homework assignment. Students of Pontryagin's principle and the associated numerical methods solvers used to show necessary conditions of optimality should be able to understand the included demonstrations. Students of optimal control theory with only a basic understanding of the calculus of variations can wade through and will recognize landmarks along the way, but may not fully understand the context. This chapter holds no necessary elements for the *employment* of the algorithm. It merely contains the supporting demonstrations that verify the soundness of the logical underpinning of the algorithm.

This chapter relies on a very specific combination of concepts and tools to assess the optimality of candidate maneuvers. This analysis applies Pontryagin's maximum principle, as described by I.M. Ross [13], [40]. It uses DIDO for MATLAB to validate the necessary conditions and applies Dr. Ross's approach to evaluate the results and assess optimality. Readers interested in replicating elements of this chapter are encouraged to contact the Naval Postgraduate School Aerospace Engineering Department for additional information and resources on DIDO [41], [42]. Various artifacts necessary to recreate these findings, such as derived Hamiltonians, Lagrangians of the Hamiltonian, and costates used in developing the necessary conditions of optimality, are also provided in the associated appendix for users without access to DIDO or other pseudospectral solvers that might seek to compare results with other direct or indirect methods available from MATLAB's optimization toolkit or made as a home-brew solver.

Although this analysis used DIDO and Dr. Ross's interpretation of Pontryagin's maximum principle, many other numerical methods are available to the guidance, navigation, and control engineers or mission managers. There are two excellent reviews that outline the state of the industry at the time of this research for trajectory optimization and low-thrust

optimization, and interested readers are referred to them as sources [43], [44]. Indeed, any reader beginning an optimization study for any spacecraft mission should start with Shirazi et al. who painstakingly compiled a comprehensive compilation of the various models and approaches that can be combined based on the specifics of the sample problem. Well referenced and thorough, their work offers an exceptional review of the nearly 100 years of efforts invested in this topic. Low-thrust mission analysts will find the survey work by Morante et al. a fantastic guide to the subject; however, a general understanding of the calculus of variations and some familiarity with the terms used in this field are prerequisites.

One of the major challenges to the application of optimal control is the rather high bar of subject-specific knowledge mastery required to work with even trivial problems. At the heart of this challenge is the fact that the differential equations associated with the optimization of these continuous low-thrust trajectories are highly non-linear and natively stiff. Several emerging tools are being developed that attempt to demystify the art of optimal control design and serve as a translator between the astrodynamicist or mission analyst and the mathematical philosophy of Bellman [45]. The System Tool Kit Astrogator offers several methods for journeyman analysts to leverage powerful numerical solvers, including a home-brew differential solver, the Interior Point Optimizer Program, and even a Sparse Nonlinear Optimizer Profile [46]–[48].

This chapter addresses the application of optimal control of a robotic spacecraft under low-thrust acceleration for maneuver in cislunar space. It principally serves as to develop the relative cost to transfer from some initial condition to some final condition. For comparison to the results of analytic estimate calculations, the specific initial and final states are arbitrary. For the relative comparison to other possible orbit transfers, the optimal path between the common initial position and the several final positions assumes optimality as a design goal. Although it is posed as a classical trajectory optimization analysis which assumes a cooperative relationship between the researcher and the hypothetical robotic space mission, the ultimate use is to validate the development of a decision support tool that enables Space Domain Awareness of noncooperative spacecraft. Towards this ultimate end, emphasis is placed on the impacts of capturing constraining situational elements as endpoint constraints, path constraints, and elements of the final cost functional.

This chapter consists of four sections that cover an orientation to the application of opti-

mal control theory on trajectory optimization, model construction and assessment, initial problem formulation and candidate solutions, advanced problem development and solutions including specific constraints.

Many of these tools are also accessible through other, potentially free, open-source methods. System Tool Kit's proprietary Astrogator software allows an extremely user-friendly way to leverage these tools [49]. Perhaps one of the greatest strengths of these integrated tools is their built-in scaling and balancing methods. Compared to the complicated methods required for use with DIDO, the ability of an analyst to simply click a button significantly lowers the bar for accessing numerical optimization [50]. Indeed, the original authors of the sample maneuver used in Chapter 5 for the low-thrust,low-energy, Earth-moon return system leveraged backward propagation with a targeting sequence. Although not specified, it is presumed that they used one of these tools to enable solution convergence [1]. Several efforts have also been made using hybrid efforts (not to be confused with hybrid optimal control approaches) that combine classical optimization approaches such as multiple shooting methods and analytical derivatives [34], [51], [52]. These methods attempt to avoid the classical problems associated with gradient-based approaches that invite roundoff and truncation errors in the calculation of the Jacobian by finitie difference methods. Finally, Di Carlo and Vasile have published many descriptions of a highly specialized optimization toolbox called CAMELOT, though the software was not used in this research [53].

3.3 Optimal Low-Thrust Lunar Orbit Transfers

This section examines the optimality of several families of selenocentric low-thrust transfer orbits. Sample transfers are grouped into two major families: in-plane transfers and out-of-plane transfers. In-plane transfers include both low ($e = 0.1$) and high ($e = 0.7$) eccentricity cases over both few (around one) and many orbital periods. Given the three known maximum control strategies presented earlier in Table 3, we have two out-of-plane trajectory transfers ($\Delta\overline{\Omega}_{MAX}$, $\Delta\bar{i}_{MAX}$) and a single in-plane transfer ($\Delta\overline{a}_{MAX}$). With the subtle assumption of constant-only thrust consideration, we can add a second in-plane control strategy for $\Delta\overline{e}$ by applying the control strategy $\alpha = \theta$, $\beta = 0$. With proper scaling, DIDO can solve for multi-revolution in-plane transfers of moderate duration. Future sections will address out-of-plane and combined in-plane and out-of plane maneuvers.

DIDO's pseudospectral method will be used to validate the necessary conditions of the candidate trajectories. Pseudospectral methods are powerful tools to determine optimality; however, while they are able to escape from Kalman's curse of sensitivity, they remain bound by the limitations of numerical methods [13]. As such, pseudospectral methods still struggle to produce qualitative results on problems of long duration, heavily varying dynamics often associated with "stiff" ODE's, and problem formations that show a high variance in the magnitudes of the states. Gauss' equations display all of these attributes. While Cartesian or polar dynamics may also be used to simplify computation, Gauss' equations will be used as they simultaneously allow immediate human interpretation of results and provide adequate accuracy to truth when considering selenocentric orbits within the range required for this application [34], [52].

This chapter consists primarily of an explanation of minimum-node solutions for a close family of in-plane example problems are examined for feasibility and optimality. Optimality is demonstrated by inspection of the necessary conditions based on Pontryagin's principle as well as implementation of an artificial constraint on α, the in-plane pointing angle of the thruster for comparison to the unconstrained solution. Finally, Where feasible low-node optimal solutions are found to meet the necessary conditions of Pontryagin's principle, Bellman tests between solution nodes are conducted to demonstrate the generation of a composite high-fidelity control signal.

3.4 Problem Family 1 – In-Plane, Low-Lunar Transfers

3.4.1 Problem Formulation

Maneuver one's problem family seeks optimal control strategies for coplanar, low-lunar transfers described mainly by variation of the semi-major axis. General elements of problem formulation such as scaling, parameterization, endpoint conditions selection are described up front. Specific problem formulation elements such as cost functional composition, varying constraints, and scaling management are addressed in each specific problem sibling.

Problem Family One's dynamics remain unchanged across siblings. Specifically as discussed

in the previous sections where the states and controls are described by

$$\mathbb{X} \in \mathbb{R}^6 \qquad\qquad \mathbb{U} \in \mathbb{R}^3$$
$$\mathbf{x} = \begin{bmatrix} a & e & i & \Omega & \omega & \theta \end{bmatrix}^T \qquad\qquad \mathbf{u} = \begin{bmatrix} \epsilon & \alpha & \beta \end{bmatrix}^T$$

Given these states and controls, the dynamics restricted two-body motion about a central point mass:

$$\begin{aligned}
\frac{da}{dt} &= \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} &= \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} &= \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} &= \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} &= \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} &= \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right]
\end{aligned}$$

Canonical scaling is employed to improve numeric efficiency and encourage solution convergence. The state and time units and generalized magnitudes of Gauss' equations for this problem family of in-plane maneuvers are:

a	$[DU]$	: semi-major Axis [Distance Unit] in km	$\|a\| \approx 10^4$
e	$[\quad]$	: Eccentricity [ratio] a unitless ratio	$10^{-1} \leqslant \|e\| \leqslant 1$
i	$[AU]$	: Inclination [Angle Unit] in rad	$10^{-1} \leqslant \|i\| \leqslant 10^1$
Ω	$[AU]$	: Longitude of the Ascending Node [Angle Unit] in rad	$10^{-1} \leqslant \|\Omega\|$
ω	$[AU]$	: Argument of Periapsis [Angle Unit] in rad	$\|\omega\| \leqslant 10^1$
θ	$[AU]$	: True Anomaly [Angle Unit] in rad	$10^1 \leqslant \|\theta\| \leqslant 10^2$
t	$[TU]$	: Time [Time Unit] in sec	$t \leqslant 10^4$
ϵ	$[AccU]$	: Magnitude of Acceleration [Acceleration Unit] in $\frac{km}{sec^2}$	$\|\epsilon\| = 10^{-5}$
α	$[AU]$	: Pointing Angle [Angle Unit] in rad	$\|\alpha\| \leqslant \pi$
β	$[AU]$	: Pointing [Angle Unit] in rad	$\beta = 0$

Examination of these natural (engineering) units suggests the use of scaling to normalize the magnitude of semi-major axis, a, argument of periapsis, ω, and time, t. The semi-major axis, a, may be scaled by a_f such that $\tilde{a} = \frac{a}{R}$ where $R = a_f$ so that $\tilde{a}$ now has a normalized magnitude of $0 < \|\tilde{a}\| \leqslant 1$. Similarly, time may be parameterized so that $\tilde{t} \in [0, 1]$ by the relationship $\tilde{a} = \frac{a}{\tau}$ where $\tau = t_f$. While these scaling efforts have now normalized all the states to within two orders of magnitude, the true anomaly, θ, is now the outlier. Additional scaling of this angle unit would require the use of designer units and modification of the dynamics to avoid the cat and mouse game produced by the further reduction in magnitude of i, Ω, and ω, should a scaled angle unit be selected. Fortunately, DIDO's robustness allows convergence *if* self-limiting actions are taken in specific problem formulation to reduce the duration and range of the problem and minor massaging of the scaling factors. Calculations are conducted within DIDO exclusively in engineering units and are scaled following calculation to avoid issues arising from the calculation of dynamics with artificially sized terms.

Throughout Problem Family One, two control constraints are invariant. The magnitude of acceleration, ϵ is set to a non-zero constant for the duration of the maneuver so that, while generally $\epsilon_{MIN} \leqslant \epsilon \leqslant \epsilon_{MAX}$, in the following cases $\epsilon_{MIN} = \epsilon_{MAX}$. This simulates a single

low-thrust transfer maneuver, and while Section I outlines full scale maneuvers which may employ multiple low-thrust arcs, these problems model only the localized continuous thrust. Similarly, β is allowed a range such that $\beta_{MIN} \leqslant \beta \leqslant \beta_{MAX}$ globally, within this problem set, $\beta_{MIN} = \beta_{MAX} = 0$.

The final problem characteristics shared by all sibling problems in Problem Family One are the endpoints. This family contains low-lunar transfers as described by the endpoint conditions outlined below in engineering units.

$$(a_0, e_0, i_0, \Omega_0, \omega_0, \theta_0) = (2000, 0.1, 0.1, 0.1, 0.1, \pi)$$
$$(a_f, e_f, i_f, \Omega_f, \omega_f, \theta_f) = (5000, 0.0151, 0.1, 0.3756, 0.1, 54.3893)$$

These final conditions were generated using an analytical control strategy that not only serves as a stalking horse candidate optimal control solution but also verifies feasibility of the problem prior to DIDO employment. It also produces an initial value for τ to support initial scaling. As discussed earlier, this analytical control strategy sets $\alpha = \gamma$ throughout the transfer and is designed to produce the maximum change in the semi-major axis, Δa_{MAX}. Because of the constraints placed on ϵ in this family of problems, these configurations enables unique assessment of various cost functionals between siblings. The path and control histories of this analytical strategy are shown in Figure 3.1.

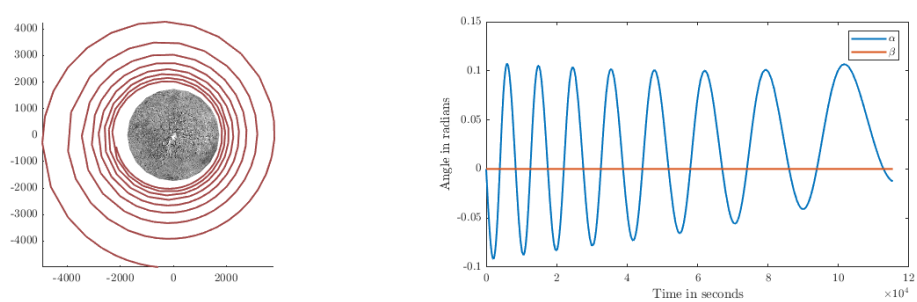


Figure 3.1. Stalking Horse Path and Control for Family 1

3.4.2 Maneuver 1.A. Minimum Time Transfer

$$\mathbf{x} := \begin{bmatrix} a & e & i & \Omega & \omega & \theta \end{bmatrix}^T \in \mathbb{R}^6$$
$$\mathbf{u} := \begin{bmatrix} \epsilon & \alpha & \beta \end{bmatrix}^T \in \mathbb{U}$$

$$\begin{cases}
\text{Minimize} & J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\
\text{Subject to} & \frac{da}{dt} = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
& \frac{de}{dt} = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
& \frac{di}{dt} = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
& \frac{d\Omega}{dt} = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
& \frac{d\omega}{dt} = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
& \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
& h_1 := 5\times10^{-6} \leqslant \epsilon \leqslant 5\times10^{-6} \\
& h_2 := -\pi \leqslant \alpha \leqslant \pi \\
& h_3 := 0 \leqslant \beta \leqslant 0\} \\
& \mathbf{x}_0 = (2000, 0.1, 0.1, 0.1, 0.1, \pi) \\
& \mathbf{x}_f = (5000, 0.0151, 0.1, 0.3756, 0.1, 54.3893) \\
& t_0 = 0 \\
& p = a(1-e^2) \\
& h = \sqrt{\mu p} \\
& r = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
& \nu = \omega + \theta
\end{cases}$$

Maneuver 1.A. seeks the time optimal transfer from a fully defined $\mathbf{x}_0$ to a fully defined $\mathbf{x}_f$ subject to constraints on controls. Given this problem definition, the development of the necessary conditions for optimality based on Pontryagin's principle are outlined below. We first form the Hamiltonian from the problem information. Since there is a control constraint,

we must form the **Lagrangian of the Hamiltonian**, $\bar{\mathbf{H}}$ as well.

$$
\begin{aligned}
H :=& F + \lambda^T f \\
H =& \lambda_1 \frac{2\, a^2\, \epsilon\, \cos(\beta)\, (\cos(\alpha)\, (e\, \cos(\theta) + 1) + e\, \sin(\alpha)\, \sin(\theta))}{\sqrt{-a\, \mu\, (e^2 - 1)}} \\
& - \lambda_2 \frac{\epsilon\, \cos(\alpha)\, \cos(\beta)\, \left(\cos(\theta)\, \left(a\, \left(e^2 - 1\right) + \frac{a\left(e^2-1\right)}{e\, \cos(\theta)+1}\right) + a\, \sin(\alpha)\, \sin(\theta)\, \left(e^2 - 1\right)\right)}{\sqrt{-a\, \mu\, (e^2 - 1)}} \\
& + \lambda_2 \frac{a\, \sin(\alpha)\, \sin(\theta)\, \left(e^2 - 1\right) + \frac{a\, e\left(e^2-1\right)}{e\, \cos(\theta)+1}}{\sqrt{-a\, \mu\, (e^2 - 1)}} \\
& - \lambda_3 \frac{a\, \epsilon\, \cos(\omega + \theta)\, \sin(\beta)\, \left(e^2 - 1\right)}{(e\, \cos(\theta) + 1)\, \sqrt{-a\, \mu\, (e^2 - 1)}} \\
& - \lambda_4 \frac{a\, \epsilon\, \sin(\omega + \theta)\, \sin(\beta)\, \left(e^2 - 1\right)}{\sin(i)\, (e\, \cos(\theta) + 1)\, \sqrt{-a\, \mu\, (e^2 - 1)}} \\
& - \lambda_5 \frac{\epsilon\, \cos(\beta)\, \left(\cos(\alpha)\, \sin(\theta)\, \left(a\, \left(e^2 - 1\right) + \frac{a\left(e^2-1\right)}{e\, \cos(\theta)+1}\right) - a\, \sin(\alpha)\, \cos(\theta)\, \left(e^2 - 1\right)\right)}{e\, \sqrt{-a\, \mu\, (e^2 - 1)}} \\
& - \lambda_5 \frac{a\, \epsilon\, \sin(\omega + \theta)\, \cos(i)\, \sin(\beta)\, \left(e^2 - 1\right)}{\sin(i)\, (e\, \cos(\theta) + 1)\, \sqrt{-a\, \mu\, (e^2 - 1)}} \\
& + \lambda_6 \left(\frac{(e\, \cos(\theta) + 1)^2\, \sqrt{-a\, \mu\, (e^2 - 1)}}{a^2\, (e^2 - 1)^2}\right. \\
& \left. + \frac{\epsilon\, \cos(\beta)\, \left(\cos(\alpha)\, \sin(\theta)\, \left(a\, \left(e^2 - 1\right) + \frac{a\left(e^2-1\right)}{e\, \cos(\theta)+1}\right) - a\, \sin(\alpha)\, \cos(\theta)\, \left(e^2 - 1\right)\right)}{e\, \sqrt{-a\, \mu\, (e^2 - 1)}}\right)
\end{aligned}
$$

$$
\begin{aligned}
\bar{H} :=& H + \mu^T u \\
\bar{H} =& H + \mu_1\, \epsilon + \mu_2\, \alpha + \mu_1\, \beta
\end{aligned}
$$

Because of the constrained controls, the Karush-Kuhn-Tucker conditions corresponding to the costates of the controls are:

$$\mu_1 \epsilon$$

$$
\begin{aligned}
\mu_1 &\leqslant 0 \qquad if \qquad \epsilon = 5 \times 10^{-5} \\
\mu_1 &= 0 \qquad if \qquad 5 \times 10^{-5} \leqslant \epsilon \leqslant 5 \times 10^{-5} \\
\mu_1 &\geqslant 0 \qquad if \qquad \epsilon = 5 \times 10^{-5}
\end{aligned}
$$

Clearly, because $\epsilon_{MIN} = \epsilon_{MAX}$, μ_1 is unrestricted and study of its complementarity conditions will not be useful in verification and validation.

$$\mu_2 \alpha$$

$$
\begin{aligned}
\mu_2 &\leqslant 0 \qquad if \qquad \alpha = -\pi \\
\mu_2 &= 0 \qquad if \qquad -\pi \leqslant \alpha \leqslant \pi \\
\mu_2 &\geqslant 0 \qquad if \qquad \alpha = \pi
\end{aligned}
$$

Here we see that any sign changes observed in μ_2's history from any candidate solution should obey the complimentary conditions noted above.

$$\mu_3 \beta$$

$$
\begin{aligned}
\mu_3 &\leqslant 0 \qquad if \qquad \beta = 0 \\
\mu_3 &= 0 \qquad if \qquad 0 \leqslant \beta \leqslant 0 \\
\mu_3 &\geqslant 0 \qquad if \qquad \beta = 0
\end{aligned}
$$

As seen with the complementarity of ϵ's costate, because $\beta_{MIN} = \beta_{MAX}$, μ_3 is unrestricted and study of its complementarity conditions will not be useful in verification and validation.

Adjoint equations are formed next from the identity $-\dot{\lambda} := \frac{\partial \bar{H}}{\partial \mathbf{x}}$. These are particularly ugly because of the state involvement in the seemingly simplifying p, h, and r. Full identities for the adjoint equations are provided in the appendix.

$$-\dot{\lambda}_a = \frac{\partial \bar{H}}{\partial a}, \quad -\dot{\lambda}_e = \frac{\partial \bar{H}}{\partial e}, \quad -\dot{\lambda}_i = \frac{\partial \bar{H}}{\partial i}, \quad -\dot{\lambda}_\Omega = \frac{\partial \bar{H}}{\partial \Omega}, \quad -\dot{\lambda}_\omega = \frac{\partial \bar{H}}{\partial \omega}, \quad -\dot{\lambda}_\theta = \frac{\partial \bar{H}}{\partial \theta}$$

Minimization of the Hamiltonian with respect to u produces the lower Hamiltonian described by

$$\mathcal{H} = \frac{\partial \bar{H}}{\partial \mathbf{u}} = \frac{\partial H}{\partial \mathbf{u}} + \frac{\partial h}{\partial \mathbf{u}}^T \boldsymbol{\mu} = 0$$

with the inclusion of the complementarity conditions developed previously. The full minimization is also included in the appendix.

The **Endpoint Lagrangian**, $\bar{E}$, is formed by the identity $\bar{E} := E + \mathbf{e}^T \boldsymbol{\nu}$ yielding $\bar{E} = t_f + \boldsymbol{e}\nu$ for this specific problem, and allows the development of the Hamiltonian value condition.

The **Hamiltonian Value Condition**, $\mathcal{H}[@t_f] = -\frac{\partial \bar{E}}{\partial t_f} = -1$, which is expected for a minimum time problem. This is a verifiable necessary condition for any candidate solution.

Examination of the **Hamiltonian Evolution Equation** yields a second verifiable necessary condition $\frac{d\mathcal{H}}{dt} = -\frac{\partial \bar{H}}{\partial t}$. While the Hamiltonian for this problem is lengthy, it is mercifully time-invariant, indicating that the lowered Hamiltonian should be stationary at -1 throughout the maneuver.

Finally, the **Terminal Transversality of the Endpoint Lagrangian** defined by $\dot{\boldsymbol{\lambda}}(t_f) = \frac{\partial \bar{E}}{\partial \mathbf{x}_f}$, such that

$$\lambda_a(t_f) = \frac{\partial \bar{E}}{\partial a_f} \quad \lambda_e(t_f) = \frac{\partial \bar{E}}{\partial e_f} \quad \lambda_i(t_f) = \frac{\partial \bar{E}}{\partial i_f}$$
$$\lambda_\Omega(t_f) = \frac{\partial \bar{E}}{\partial \Omega_f} \quad \lambda_\omega(t_f) = \frac{\partial \bar{E}}{\partial \omega_f} \quad \lambda_\theta(t_f) = \frac{\partial \bar{E}}{\partial \theta_f}$$

Armed with expectations of the necessary conditions of optimality and reachable final position (the problem is confirmed not to be solvable) the problem is loaded into DIDO and the results are analyzed (DIDO files are included in the appendix). Thankfully, DIDO's resilience to the rudimentary scaling produces a feasible extremal solution to problem 1.A. with only 99 nodes. Skeptics are invited to review current literature on the use of DIDO for high-fidelity trajectory development of low-thrust missions [54] where the combined use of both antialiasing techniques and designer units result in a sparse-node trajectory on a two-dimensional circular orbit. This method enables Bellman segment propagation to patch multiple bite-sized optimal control histories into a singular solution without violating the underlying principles of dynamic programming.

In the case of problem 1.A., however, DIDO's 99-node solution provides sufficient accuracy to verify feasibility with basic control interpolation and was processed in 261 seconds. Figure 3.2 displays the unaltered state and control histories produced by DIDO's candidate solution. While

the DIDO-produced states are both unreliable and infeasible for validation, the unaltered control history indicates a highly usable control solution was found with which to conduct propagation and feasibility testing. The resulting cost score of 2 provides the first indication that our stalking horse analytic trajectory is indeed optimal for this problem. Because time was scaled to $\tau = \frac{t_{fAnalytic}}{2}$ the resulting cost functional score, as well as the close fit of control histories suggest that the necessary conditions for optimality derived from examination of DIDO's results also hold for the analytic model.

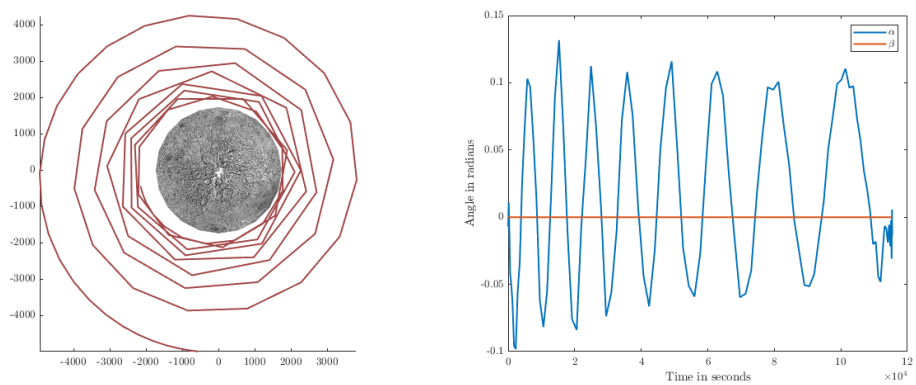


Figure 3.2. Unaltered State and Control Histories from DIDO's Problem 1.A. Candidate Solution

Feasibility is evaluated first. Given a good enough initial solution, spline interpolation of the control history of α allows for test by propagation through the original dynamics using ode15s in MATLAB. The results are then compared to the demanded final state. Parallel plotting of the stalking horse trajectory also indicates strong correlation between the two control strategies. The graphic results are displayed in Figure 3.3. Numerically, maximum variation in final state for this result was 5% variation for e^f as measured by the methodology used in Section II. The next highest variance from the final position was 0.05% deviation for θ^f with the other variations reporting beyond the magnitude of 10^{-5}%. Given the observable "jiggle" near the end of the control history, this outcome is readily validated as feasible.

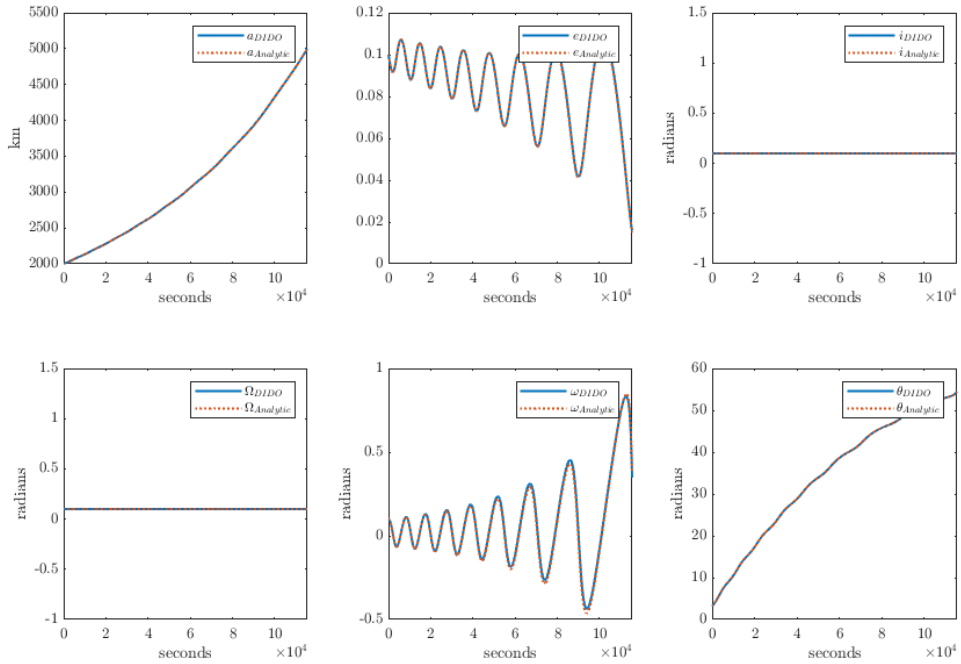


Figure 3.3. Feasibility Test Results from DIDO's Problem 1.A. Candidate Solution

The interpolated control and position are shown with the stalking horse analytical results for comparison in Figure 3.4.

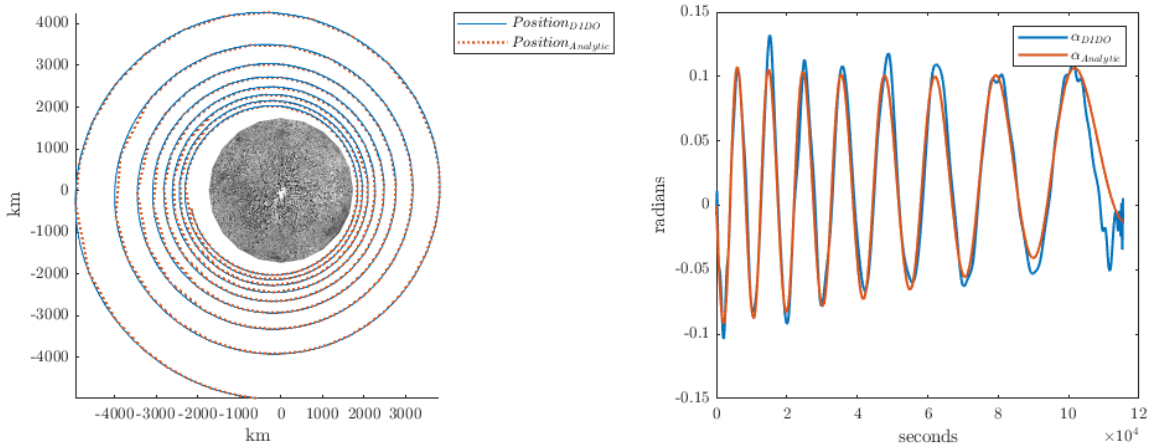


Figure 3.4. Interpolated Results from DIDO's Problem 1.A. Candidate Solution and the Stalking Horse Solution

Verification of the **Necessary Conditions** as developed earlier in this problem examination indicate that the necessary conditions for optimality are met based on analysis of the minimized Hamiltonian shown in Figure 3.5.

Figure 3.5. Adherence to Minimized Hamiltonian Behavior for Problem 1.A.

Inspection of the control, α, with its corresponding costate, μ_2, confirm that throughout the maneuver, α never reaches its upper or lower bounds (π and $-\pi$ respectively) and that μ_2 obeys the Karush-Kuhn-Tucker conditions that dictate it remain at zero throughout the maneuver. These results are demonstrated in Figure 3.6.

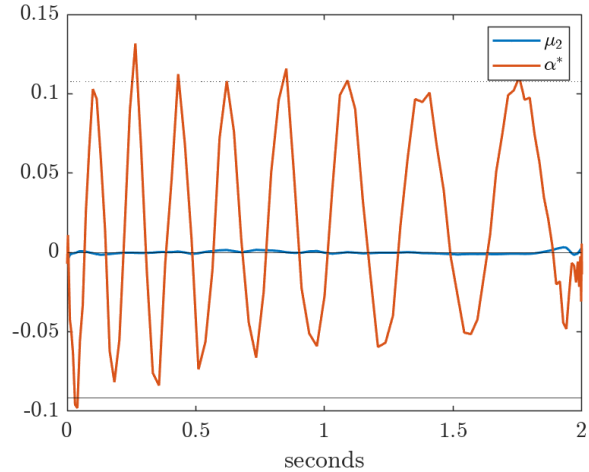


Figure 3.6. Adherence to Karush-Kuhn-Tucker Conditions for Problem 1.A.

3.4.3 Maneuver 1.B. Maximum Change to semi-major Axis, Fixed Time Transfer

Maneuver 1.B. seeks the fixed time solution from a fully defined set $\mathbf{x}_0$ over a set time that achieves the maximum a_f. The only change from problem 1.A. is the final time cost of the cost functional, J. Control constraints are identical. The complete problem statement is outlined below, followed by changes to the derivation of the necessary conditions required to transfer the previous analysis to problem 1.B.

$$\mathbf{x} := [a \quad e \quad i \quad \Omega \quad \omega \quad \theta]^T \in \mathbb{R}^6$$
$$\mathbf{u} := [\epsilon \quad \alpha \quad \beta]^T \in \mathbb{U}$$

$$
\begin{cases}
\text{Minimize } J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = -a_f \\
\\
\text{Subject to } \frac{da}{dt} & = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} & = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} & = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} & = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} & = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} & = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
h_1 & := 5\times10^{-6} \leqslant \epsilon \leqslant 5\times10^{-6} \\
h_2 & := -\pi \leqslant \alpha \leqslant \pi \\
h_3 & := 0 \leqslant \beta \leqslant 0\} \\
(a_0, e_0, i_0, \ \Omega_0, \omega_0, \theta_0) & = (2000, 0.1, 0.1, 0.1, 0.1, \pi) \\
t_0 & = 0 \\
t_0 & = 1.1543\times10^5 \\
p & = a(1-e^2) \\
h & = \sqrt{\mu p} \\
r & = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
\nu & = \omega + \theta
\end{cases}
$$

The Hamiltonian and Lagrangian of the Hamiltonian are identical to those formed for Problem 1.A. The **Endpoint Lagrangian**, $\bar{E}$, is transformed to $\bar{E} = -a_f + \boldsymbol{e}\nu$ for this specific problem, and allows the development of the Hamiltonian value condition. Identical Karush-Kuhn-Tucker conditions are expected. The **Hamiltonian Value Condition**, $\mathcal{H}[@t_f] = -\frac{\partial\bar{E}}{\partial t_f} = 0$, which is expected for a minimum time problem. This is a verifiable necessary condition for any candidate solution. When combined with the **Hamiltonian Evolution Equation** a second verifiable necessary condition is again found in $\frac{d\mathcal{H}}{dt} = -\frac{\partial\bar{H}}{\partial t} = 0$. It is now necessary for candidate solutions to demonstrate a minimized Hamiltonian that is stationary at zero. Finally, the **Terminal Transversality of the Endpoint Lagrangian**, defined by $\dot{\boldsymbol{\lambda}}(t_f) = \frac{\partial\bar{E}}{\partial\mathbf{x}_f}$, now yields an additional necessary condition, $\lambda_a(t_f) = \frac{\partial\bar{E}}{\partial a_f} = -1$

DIDO again produces a perfectly suitable candidate solution, this time in 340 seconds, with the expected cost score of -1. Nearly identical control and path histories are produced, and interpolation and test through propagation again supports feasibility for this DIDO solution, as shown in Figure 3.7.

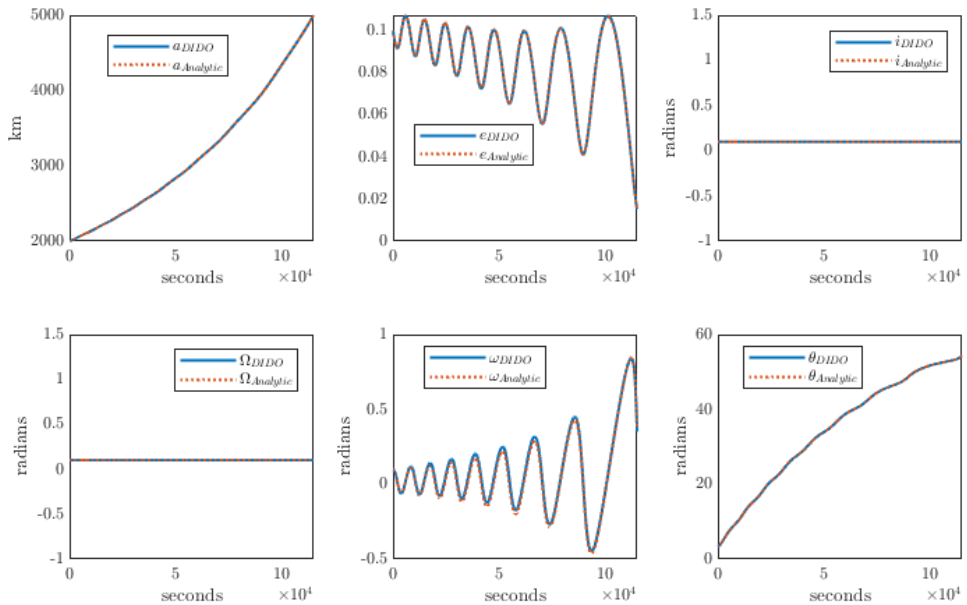


Figure 3.7. Feasibility of DIDO's Solution for Problem 1.B.

Inspection of the minimized Hamiltonian, μ_2 Karush-Kuhn-Tucker conditions, and λ_a position at the final time all support an adherence to the necessary conditions of optimality. These findings are captured in Figures 3.8 through 3.10.

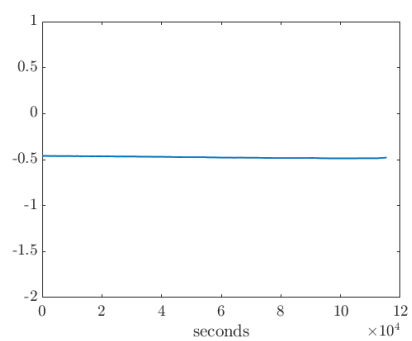


Figure 3.8. Minimized Hamiltonian of DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions

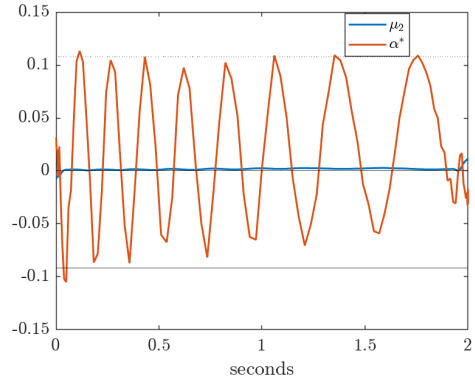


Figure 3.9. Karush-Kuhn-Tucker Conditions Obeyed in DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions

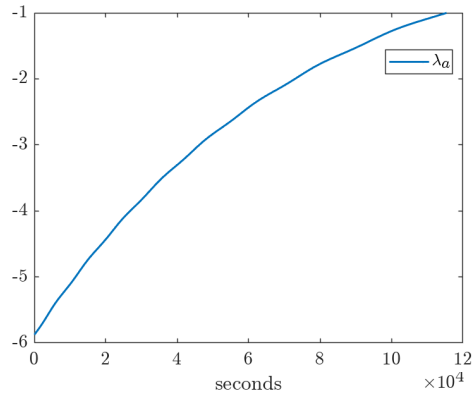


Figure 3.10. Confirmation of Costate λ_a's Performance at t_f in DIDO's Solution for Problem 1.B. Demonstrating Necessary Conditions

The findings support confirmation of optimality for DIDO's solution and by extension the path and control histories created by the analytical control schema in this case, but there is another test that can support an assessment of optimality for this trajectory control. Problem 1.C outlined below is identical to problem 1.B except for an artificial constraint on α that now demands $\|\alpha\| \leqslant 0.4$. Since problem 1.B Karush-Kuhn-Tucker condition testing indicates that the candidate optimal control history requires α to be free to oscillate between at least ± 0.5, this new constraint should impact the cost score of the optimal control produced by solving problem 1.C. Furthermore, Karush-Kuhn-Tucker testing should confirm the impact of this constraint on the control while validating the solution to problem 1.C.

3.4.4 Maneuver 1.C. Maximum Change to Semi-major Axis, Fixed Time, Active Control Constraint on α

$$\mathbf{x} := [a \quad e \quad i \quad \Omega \quad \omega \quad \theta]^T \in \mathbb{R}^6$$
$$\mathbf{u} := [\epsilon \quad \alpha \quad \beta]^T \in \mathbb{U}$$

$$\begin{cases}
\text{Minimize } J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = -a_f \\
\\
\text{Subject to } \frac{da}{dt} & = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} & = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} & = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} & = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} & = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} & = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
h_1 & := 5\times10^{-6} \leqslant \epsilon \leqslant 5\times10^{-6} \\
h_2 & := -0.4 \leqslant \alpha \leqslant 0.4 \\
h_3 & := 0 \leqslant \beta \leqslant 0\} \\
(a_0, e_0, i_0, \ \Omega_0, \omega_0, \theta_0) & = (2000, 0.1, 0.1, 0.1, 0.1, \pi) \\
t_0 & = 0 \\
t_0 & = 1.1543\times10^5 \\
p & = a(1-e^2) \\
h & = \sqrt{\mu p} \\
r & = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
\nu & = \omega + \theta
\end{cases}$$

To accurately reflect and acknowledge the change to the Lagrangian of the Hamiltonian from problems 1.A and 1.B, updated Karush-Kuhn-Tucker conditions

$$\mu_2\alpha \text{ updated for problem 1.C.}$$

$$
\begin{aligned}
\mu_2 &\leqslant 0 \qquad if \qquad \alpha = -0.4 \\
\mu_2 &= 0 \qquad if \qquad -0.4 \leqslant \alpha \leqslant 0.4 \\
\mu_2 &\geqslant 0 \qquad if \qquad \alpha = 0.4
\end{aligned}
$$

There are no additional changes to the necessary conditions for optimality, though a more bespoke evaluation of the numerical results will be necessary in verification and validation. The goal is to declare the solution sufficiently optimal to serve as a benchmark of the impact of restricting α below the performance levels demanded by the relatively unconstrained problem developed in 1.B.

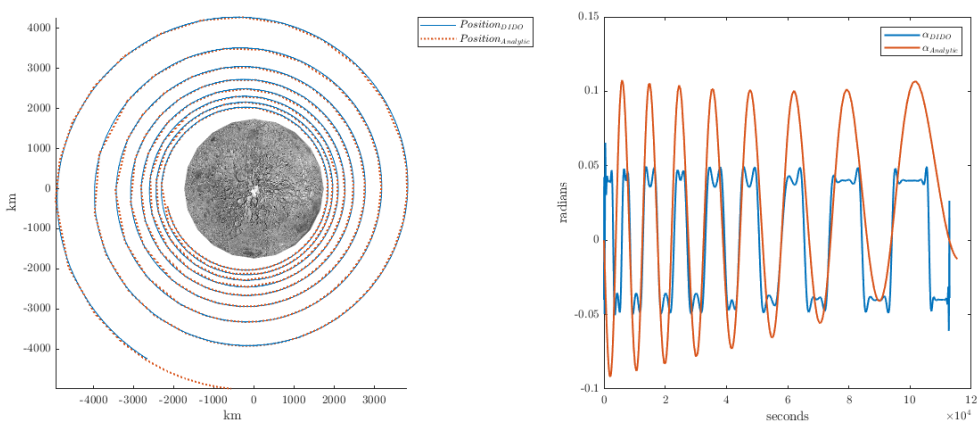


Figure 3.11. DIDO's Constrained Solution for Problem 1.C with Basic Interpolation

Optimality DIDO produces a candidate solution to problem 1.C. in just 31 seconds at a node density of 99 nodes. As expected, the resulting cost is higher (The cost is a measure of the negative of the final semi-major axis at the final time. A lower score is better. The solution to problem 1.B. scored a -1.) at -0.9629. *Feasibility* is interpreted by visual adherence to the expected behavior of such a control in the model and comparison to other *similar* solutions to sibling problems like 1.B and 1.C.

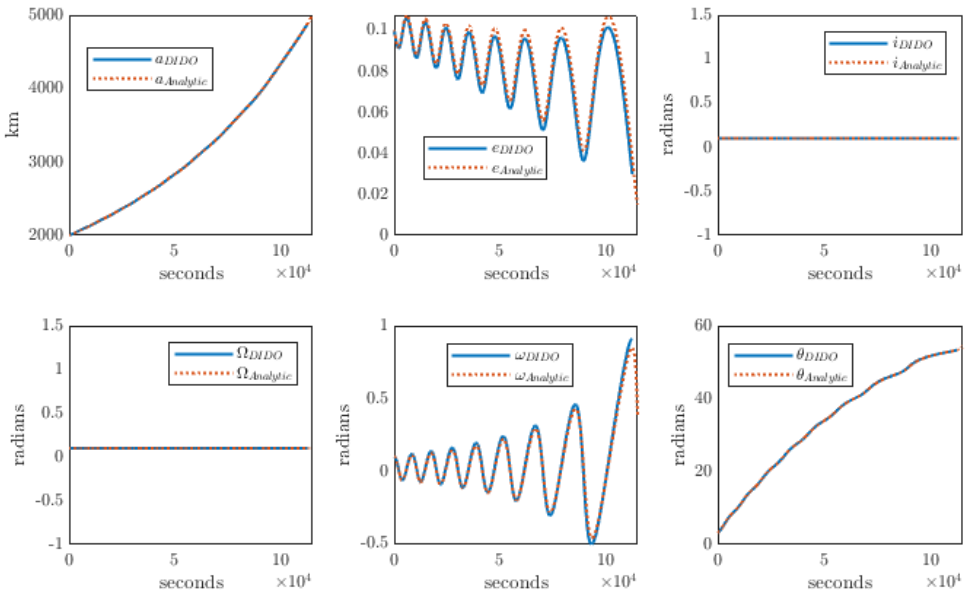


Figure 3.12. Feasibility of DIDO's Solution for Problem 1.C.

label1.cfeas

Inspection of μ_2 and α meets the derived Karush-Kuhn-Tucker conditions. Figure **??** displays both by scaling mu_2 by a factor of 5×10^{-11}. The artificial constraint clearly demonstrates an impact on the presumed optimal solution formed by 1.B. Note also that this plot shows DIDO's native values without interpolation and the associated smoothing seen in Figure 3.13.

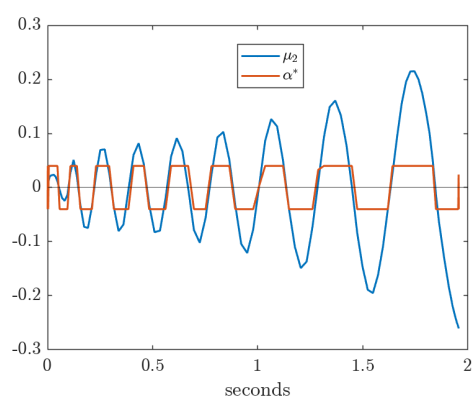


Figure 3.13. Karush-Kuhn-Tucker Conditions Obeyed in DIDO's Solution for Problem 1.C. Demonstrating Necessary Conditions

Optimality is further assessed through inspection of the minimized Hamiltonian and costate λ_a at t_f demonstrate that this is a far from optimal solution, however, as the purpose is to compare two set-piece problems 1.B and 1.C, it is accepted as a suboptimal but valid solution given the impact of the control constraint.

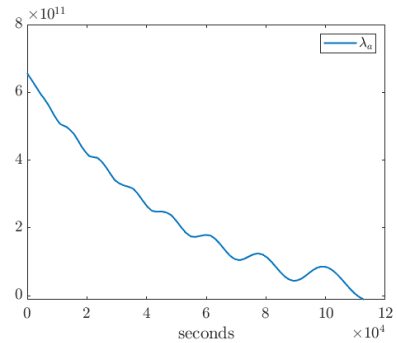


Figure 3.14. Terminal Transversality of the Endpoint Lagrangian from DIDO's Solution for Problem 1.C.

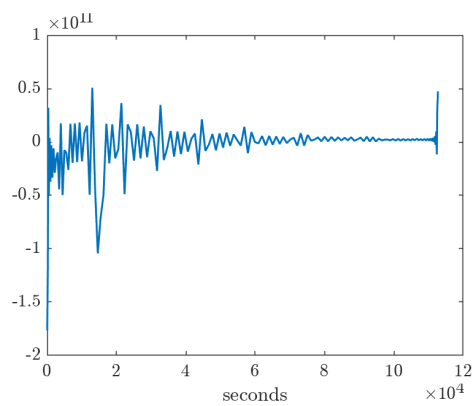


Figure 3.15. Minimized Hamiltonian from DIDO's Solution for Problem 1.C.

3.5 Problem Family 2 – Out-of-Plane, Low-Lunar Transfers - General Problem Formulation

3.5.1 Problem Formulation

This family of problems seeks optimal control strategies for non-coplanar, low-lunar transfers described mainly by variation of the inclination. General elements of problem formulation such as scaling, parameterization, endpoint conditions selection are addressed first. Specific problem formulation elements such as cost functional composition, varying constraints, and scaling management are addressed in each specific problem sibling. Again, a stalking horse candidate control history is provided from literature [55] describing the analytically derived maximum variation to the inclination of an orbit. Derived as $|\beta| = \frac{\pi}{2}$ and specifically employed as $\beta = sign(\cos(\omega + \nu)\frac{\pi}{2}$ for implementation in programming. An exemplar application of this control scheme is shown in Figure 3.16.

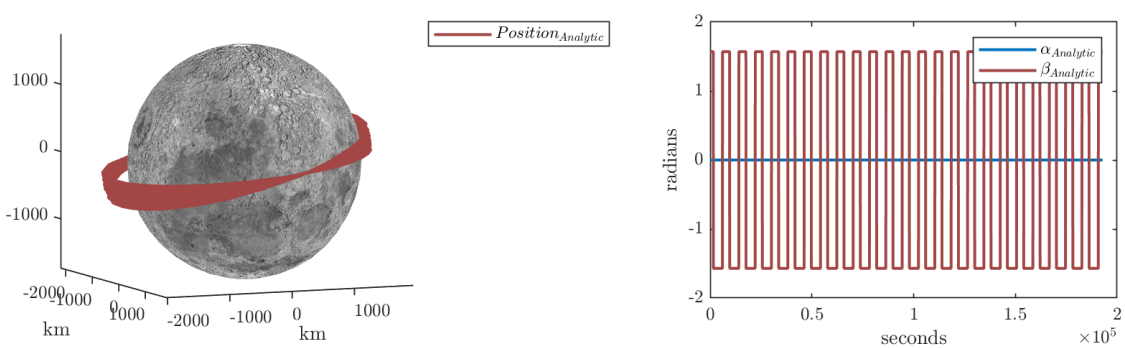


Figure 3.16. Stalking Horse Path and Control for Family 2

Simple inspection of the candidate control history indicates that the "bang-bang" nature of β and the multi-revolution time-period will be challenging for DIDO to compete with. Additionally, this family of transfer also suffers a distinct challenge of non-additive forces that was not present in Problem Family 1. Recall that the acceleration in the first family is continuously aligned with γ, the flight path angle, with a significant portion of the acceleration continuously adding to the velocity of the spacecraft. This additive acceleration causes the increasing period and decreasing amplitude observed in the optimal control history for Problem Family 1. In this case, variations of inclination rely solely on out-of-plane forces and can be confirmed by observing that neither $\frac{di}{dt} = \epsilon \frac{r}{h} \cos\nu \sin\beta$ nor any of its component identities rely on α. By definition none of the initial velocity is in the normal

direction, therefore, unlike in Problem Family 1, there is no existing or natural velocity in the direction required to facilitate the maneuver.

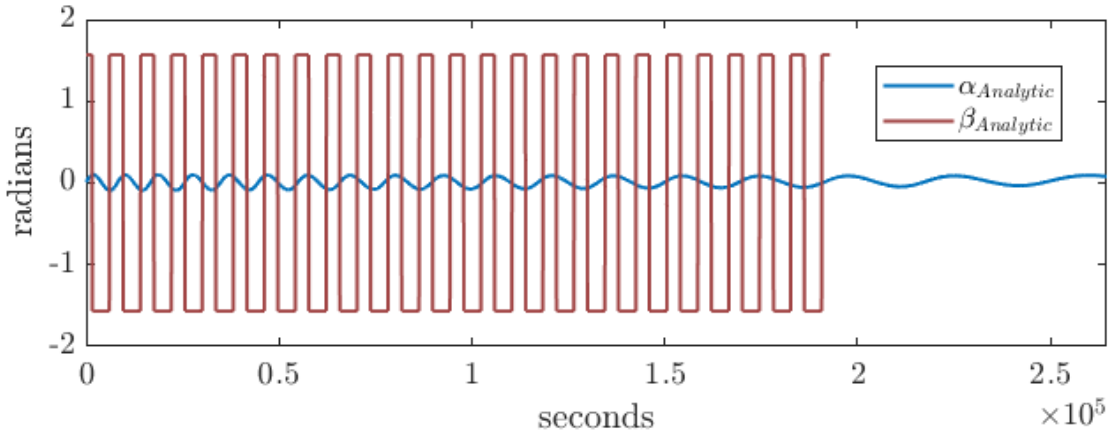


Figure 3.17. Comparison of α from the Maximum Δa Strategy and β from the Maximum Δi Strategy

Figure 3.17 displays the α control history for the maximum Δa strategy and β from the maximum Δi Strategy. The α angle control history corresponds to a change of semi-major axis from $2000km$ to $6000km$, and the β angle control history corresponds to a change of inclination from $0.1rad$ to $0.3rad$ with a semi-major axis of $2000km$. For each of these strategies, the complimentary control angle is zero. Additionally, because the thrust direction must instantaneously change signs as the spacecraft crosses the line of nodes for the out-of-plane motion, any gathered velocity in the normal plan is immediately abandoned. While this suggests that a number of candidate control strategies might be employed depending on the specific problem, the imposition of continuous thrust throughout the maneuver and demand for instantaneous switching of sign of the control angle β make the analytically optimal control strategy unachievable with machine precision in calculation. However, as will be shown in and validated in the analysis of maneuvers 2.B and 2.C, the *method* of pseudospectral optimal control theory and the application of Pontryagin's principle can be used to support arguments for the reigning optimality of the analytic solutions even when problems demonstrate overwhelming non-convexity and prevent definitive gathering of the necessary conditions.

All of these elements conspire to make the clear demonstration of the necessary conditions of optimality a tall order with a general optimal control solver with the "human friendly" variational

equations. Problem Family Two's dynamics remain unchanged across siblings and are identical to family one. Gauss' equations are again employed as the dynamics, with the same basic controls describing the applied force of the low-thrust engine. Here again, the nonlinear aspect of the dynamics requires scaling of the states to localize the behavior of the costates in an effort to capture the necessary conditions for optimality. Canonical scaling is abandoned for designer units in an effort to tame the magnitude of the costates and resulting Hamiltonian. Though a variety of strategies were attempted, scaling parameters that provided the least-bad costate and Hamiltonian magnitudes are presented with the analysis of each maneuver.

3.5.2 Maneuver 2.A. Minimum Time Transfer

Maneuver 1.A. seeks the time optimal transfer from a fully defined $\mathbf{x}_0$ to a fully defined, $\mathbf{x}_f$ subject to constraints on controls. The Hamiltonian of this problem is identical to those used in Problem Family One; however, the Lagrangian of the Hamiltonian, $\bar{\mathbf{H}}$, is altered to account for the change from in-plane to out-of-plane controls for α and β as well.

$$\mathbf{x} := \begin{bmatrix} a & e & i & \Omega & \omega & \theta \end{bmatrix}^T \in \mathbb{R}^6$$
$$\mathbf{u} := \begin{bmatrix} \epsilon & \alpha & \beta \end{bmatrix}^T \in \mathbb{U}$$

$$
\begin{cases}
\text{Minimize } J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = t_f \\
\\
\text{Subject to } \frac{da}{dt} & = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} & = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} & = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} & = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} & = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} & = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
h_1 & := 2.5\times10^{-6} \leqslant \epsilon \leqslant 2.5\times10^{-6} \\
h_2 & := 0 \leqslant \alpha \leqslant 0 \\
h_3 & := -\pi \leqslant \beta \leqslant +\pi \\
\mathbf{x}_0 & = (10000, 0.1, 0.1, 0.1, 0.1, \pi) \\
\mathbf{x}_f & = (10000, 0.3, 0.1, 0.4023, -0.1985, 6.3598) \\
t_0 & = 0 \\
p & = a(1-e^2) \\
h & = \sqrt{\mu p} \\
r & = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
\nu & = \omega + \theta
\end{cases}
$$

Because of the constrained controls, the Karush-Kuhn-Tucker conditions corresponding to the costates of the controls are:

$$\mu_1\epsilon \tag{3.1}$$

$$
\begin{aligned}
\mu_1 &\leqslant 0 \quad && if \quad && \epsilon = 5\times10^{-5} \\
\mu_1 &= 0 \quad && if \quad && 5\times10^{-5} \leqslant \epsilon \leqslant 5\times10^{-5} \\
\mu_1 &\geqslant 0 \quad && if \quad && \epsilon = 5\times10^{-5}
\end{aligned}
$$

Again, because $\epsilon^L = \epsilon^U$, μ_1 is unrestricted and study of its complementarity conditions will not be useful in verification and validation.

$$\mu_2\alpha$$

$$\begin{aligned} \mu_2 &\leqslant 0 \qquad if \qquad \alpha = 0 \\ \mu_2 &= 0 \qquad if \qquad 0 \leqslant \alpha \leqslant 0 \\ \mu_2 &\geqslant 0 \qquad if \qquad \alpha = 0 \end{aligned}$$

As with the **Karush-Kuhn-Tucker conditions** of ϵ, because $\alpha^L = \alpha^U$, μ_2 is also unrestricted and is not useful in verification and validation.

$$\mu_3 \beta$$

$$\begin{aligned} \mu_3 &\leqslant 0 \qquad if \qquad \beta = -\pi \\ \mu_3 &= 0 \qquad if \qquad -\pi \leqslant \beta \leqslant \pi \\ \mu_3 &\geqslant 0 \qquad if \qquad \beta = \pi \end{aligned}$$

Now it is seen that any sign changes observed in mu_3's history from any candidate solution should obey the complimentary conditions noted above. This will end up being the primary condition that is easily accessible using general solving methods and imperfect balancing.

Again we observe that the form of the **adjoint equations** is unchanged from problem 1 except for the updated Karush-Kuhn-Tucker conditions on the constraints on the controls in the Lagrangian of the Hamiltonian which forms the adjoint equations in the identity $-\dot{\lambda} := \frac{\partial \bar{H}}{\partial \mathbf{x}}$.

Minimization of the Hamiltonian with respect to u produces the lower Hamiltonian described by

$$\mathcal{H} = \frac{\partial \bar{H}}{\partial \mathbf{u}} = \frac{\partial H}{\partial \mathbf{u}} + \frac{\partial h}{\partial \mathbf{u}}^T \boldsymbol{\mu} = 0$$

As discussed previously, because this family of problems is entirely dependent on the out-of-plane force imparted by the value of β, full examination of $\frac{\partial \bar{H}}{\partial \beta} = 0$ is worth clearly developing here since it will play a significant role in the development of the lowered Hamiltonian's magnitude. It can be shown that

$$\frac{\partial \bar{H}}{\partial \beta} = 0 \tag{3.2}$$

Inspection of $\frac{\partial \bar{H}}{\partial \beta} = 0$ as a full solution in terms of the states and costates (available in Appendix B)

suggests that the predominance of the magnitude of deviation from analytically optimal is related to the costates and states, except for Ω. This will inform some search criteria for successful designer unit scaling and balancing.

With regard to the **Endpoint Lagrangian**, $\overline{E}$'s inclusion of t_f from the cost functional and the leading to a verifiable necessary condition **Hamiltonian Value Condition**, $\mathcal{H}[@t_f] = -\frac{\partial \bar{E}}{\partial t_f} = -1$ and due to the by the **Hamiltonian Evolution Equation** yields a another verifiable necessary condition $\frac{d\mathcal{H}}{dt} = -\frac{\partial \bar{H}}{\partial t} = 0$ indicate that an optimal lowered Hamiltonian should be stationary at -1 throughout the maneuver. Finally, the **Terminal Transversality of the Endpoint Lagrangian** defined by $\dot{\boldsymbol{\lambda}}(t_f) = \frac{\partial \bar{E}}{\partial \mathbf{x}_f}$, is further modified by the endpoints described in the problem formulation.

Low-node solutions from DIDO, while neither optimal nor feasible, show both early commitment to the instantaneous nature of the switching of β as seen in pre- and post-switch impulse "jitter" and general commitment to the maximum and minimum values of β developed in the stalking horse optimal control strategy formed analytically.

Initial experimentation indicates that, given this problem, native state magnitudes remain unchanged from family 1, however, costate values for the unscaled problem are on the order of 10^{11}. Through the use of designer units, it is possible to achieve costate values on the order of 10^4. However, given $\frac{\partial \bar{H}}{\partial \beta}$ and the bang-bang nature of the stalking horse strategy, the sensitivity of both the lowered Hamiltonian and terminal transversality of the endpoint Lagrangian rapidly exceeds the abilities of a general solver to demonstrate the necessary conditions.

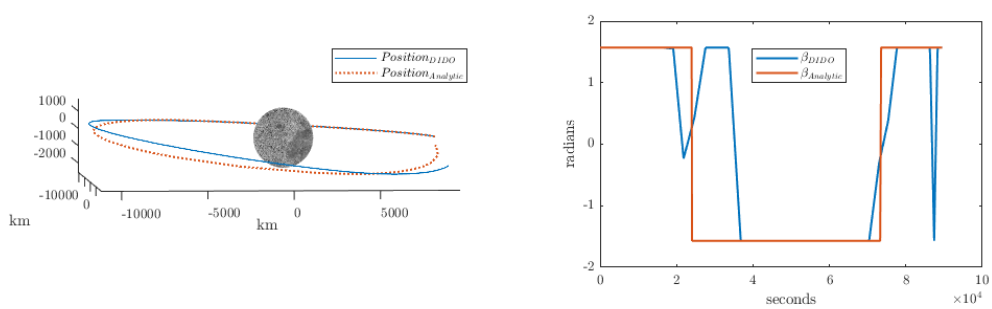


Figure 3.18. Infeasible and Non-optimal Low-node (40 Nodes) Result from DIDO for Maneuver 2.A

One indication of this sensitivity is the persistence of "artifacts" (spikes, etc) across a broad range of nodal evaluations. These impacts are seen in the low-node feasibility failure in Figure 3.18. Further

refinement of the node selection yields much better, though still infeasible and non-optimal, solutions like those in Figure 3.19. Unfortunately, the nature of the problem means that scaling the states to result in excellent control histories causes costate growth to enormous levels. Though Figure 3.19 is readily accepted as feasible, because of the scaling units needed to achieve these results, the Hamiltonian and costates are beyond 10^{20} and unsuitable for validating the necessary conditions.

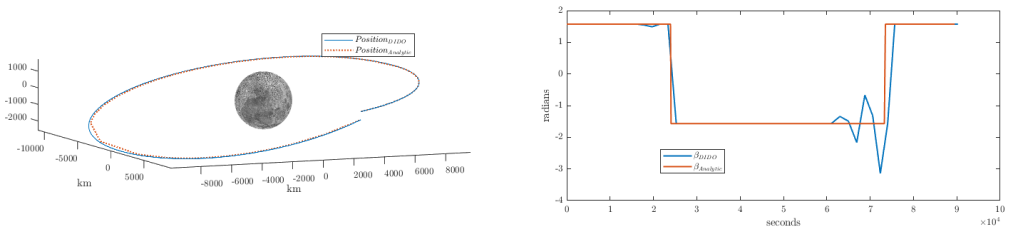


Figure 3.19. Poor Scaling and Sensitivity Yield a Feasible but Non-optimal Moderate-node (66 Nodes) Result from DIDO for Maneuver 2.A

Observe that while the spike around 2×10^4 seconds remains present, even with the significant node increase, a second spike that appears to now extend all the way to the lower control limit of β. Further analysis of this behavior, with mutual context provided from the associated Karush-Kuhn-Tucker condition, μ_3 does suggest that this behavior conforms to the expected conditions as shown in Figure 3.20. Note that mu_3 was scaled by a factor of 10^{-7} for the ease of scaling the graph. Again, the sensitivity caused by the imbalanced scaling has driven the costates of the controls into high orders of magnitude.

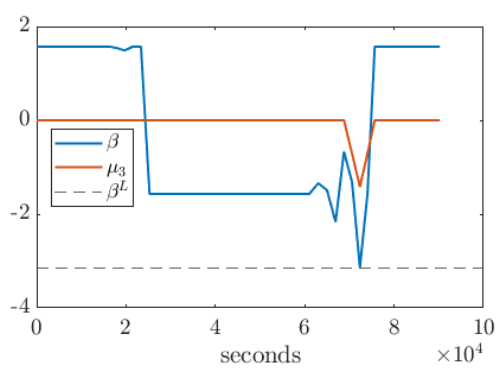


Figure 3.20. Adherence to Karush-Kuhn-Tucker Conditions for Sub-optimal but Feasible solution for Maneuver 2.A

Although the sensitivity of the problem demands balancing and scaling far superior to the author's abilities, no indication of optimality derived for DIDO's solution can be made. As such, while there is no optimality from the solver to apply to the stalking horse optimal control history, it is clearly observed that the stalking horse control produced superior cost performance (had the fastest time) of all solutions produced by DIDO.

3.5.3 Maneuver 2.B. Unconstrained Maximum Variation of Inclination

Maneuver 2.B. seeks the maximum variation of inclination from a fully defined $\mathbf{x}_0$ to any final state, $\mathbf{x}_f$ free, subject to constraints on controls over a fixed time period. The Hamiltonian of this problem is identical to those used in problem 2.A.

$$\mathbf{x} := \begin{bmatrix} a & e & i & \Omega & \omega & \theta \end{bmatrix}^T \in \mathbb{R}^6$$
$$\mathbf{u} := \begin{bmatrix} \epsilon & \alpha & \beta \end{bmatrix}^T \in \mathbb{U}$$

$$
\begin{cases}
\text{Minimize } J[\mathbf{x}(\cdot),\mathbf{u}(\cdot),t_f] = -i_f \\
\\
\text{Subject to } \frac{da}{dt} & = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} & = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} & = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} & = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} & = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} & = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
h_1 & := 2.5\times10^{-6} \leqslant \epsilon \leqslant 2.5\times10^{-6} \\
h_2 & := 0 \leqslant \alpha \leqslant 0 \\
h_3 & := -\pi \leqslant \beta \leqslant +\pi \\
(a_0, e_0, i_0, \Omega_0, \omega_0, \theta_0) & = (10000, 0.1, 0.1, 0.1, 0.1, \pi) \\
t_0 & = 0 \\
t_f & = 8.9494\times10^4 \\
p & = a(1-e^2) \\
h & = \sqrt{\mu p} \\
r & = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
\nu & = \omega + \theta
\end{cases}
$$

Because of the constrained controls, the Karush-Kuhn-Tucker conditions corresponding to the costates of the controls are identical to those of problem 2.A as are the adjoint equations, and minimized (lower) Hamiltonian.

With regard to the **Endpoint Lagrangian**, $\overline{E}$'s inclusion of i_f in the place of t_f from the cost functional and the leading to a verifiable necessary condition of**Hamiltonian Value Condition**, $\mathcal{H}[@t_f] = -\frac{\partial\bar{E}}{\partial t_f} = 0$ and due to the by the **Hamiltonian Evolution Equation** yields a another verifiable necessary condition $\frac{d\mathcal{H}}{dt} = -\frac{\partial\bar{H}}{\partial t} = 0$ indicate that an optimal lowered Hamiltonian should be stationary at 0 throughout the maneuver, although the same sensitivity and resulting impacts on observation of necessary conditions apply from problem 2.B. Finally, the **Terminal Transversality of the Endpoint Lagrangian** defined by $\dot{\boldsymbol{\lambda}}(t_f) = \frac{\partial\bar{E}}{\partial\mathbf{x}_f}$, provides another necessary condition of $\lambda_3(t_f) = -1$.

Again, DIDO's Low-node and High-node solutions, while still suboptimal, show both early commitment to the instantaneous nature of the switching of β as seen in pre- and post-switch impulse "jitter" and general commitment to the maximum and minimum values of β developed in the stalking horse optimal control strategy formed analytically.

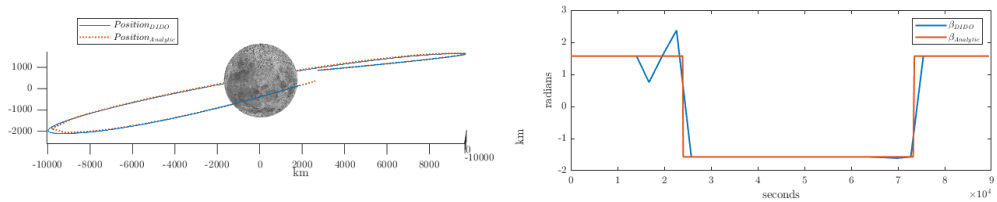


Figure 3.21. Feasibl but Non-optimal Low-node (40 node) Result from DIDO for Maneuver 2.B

Close inspection of the DIDO provided control history for β demonstrates the continued presence of both "spikes" observed in maneuver 1.A. Because of the closeness of maneuvers 2.A and 2.B, and predominantly due to the imposition of continuous thrust, similar sensitivity is observed here. Recalling the control constraints on β for this maneuver, we can verify that the Lagrange multiplier for β is inactivated throughout the maneuver, and conforms properly to the Karush-Kuhn-Tucker condition as shown in Figure 3.22.

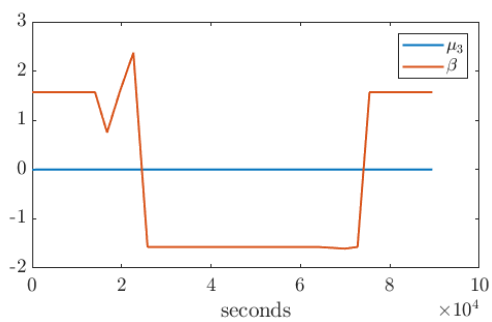


Figure 3.22. Karush-Kuhn-Tucker Verification for Maneuver 2.B

Without the ability to assess a clear adherence to the necessary conditions of the performance of the minimized Hamiltonian throughout the maneuver and at the final time, and a similar inability to replicate the necessary conditions demanded by the behavior of $\lambda_3(t_f)$ from the terminal transversality of the endpoint Lagrangian optimality cannot be determined by Pontryagin's principle. However, the stalking horse control scheme outperforms DIDO. This reigning optimality is the only claim that can be clearly stated through this method. As we will see in the final

3.5.4 Maneuver 2.C. Constrained Maximum Variation of Inclination

Maneuver 2.C. seeks the maximum variation of inclination from a fully defined $\mathbf{x}_0$ to any final state, $\mathbf{x}_f$ free, subject to **strong** constraints on the primary control over a fixed time period. The complete problem description is below.

$$\mathbf{x} := [a \quad e \quad i \quad \Omega \quad \omega \quad \theta]^T \in \mathbb{R}^6$$
$$\mathbf{u} := [\epsilon \quad \alpha \quad \beta]^T \in \mathbb{U}$$

$$\begin{cases}
\text{Minimize } J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = -i_f \\
\\
\text{Subject to } \frac{da}{dt} & = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
\frac{de}{dt} & = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
\frac{di}{dt} & = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
\frac{d\Omega}{dt} & = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
\frac{d\omega}{dt} & = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
\frac{d\theta}{dt} & = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
h_1 & := 2.5\times10^{-6} \leqslant \epsilon \leqslant 2.5\times10^{-6} \\
h_2 & := 0 \leqslant \alpha \leqslant 0 \\
h_3 & := -1.5688 \leqslant \beta \leqslant 1.5688 \\
(a_0, e_0, i_0, \Omega_0, \omega_0, \theta_0) & = (10000, 0.1, 0.1, 0.1, 0.1, \pi) \\
t_0 & = 0 \\
t_f & = 8.9494\times10^4 \\
p & = a(1-e^2) \\
h & = \sqrt{\mu p} \\
r & = \frac{\frac{h^2}{\mu}}{1+e\cos(\theta)} \\
\nu & = \omega + \theta
\end{cases}$$

The Hamiltonian of this problem is identical to those used in problem 2.A. The Lagrangian of the Hamiltonian must be restated with the modification of $\beta^L_{DIDO} = MIN(\beta_{Analytical}) + 0.002$ and $\beta^U_{DIDO} = MAX(\beta_{Analytical}) - 0.002$ with the intent of seeking the change in μ_3 when the candidate optimal solution is restricted from the solver. For the remaining controls, the Karush-Kuhn-Tucker conditions corresponding to the costates of the controls are identical to those of problem 2.B as are the adjoint equations, and minimized (lower) Hamiltonian. There are no changes to the findings

based on the endpoint Lagrangian, nor any direct changes to the Hamiltonian value or evolution equations from problem 2.B.

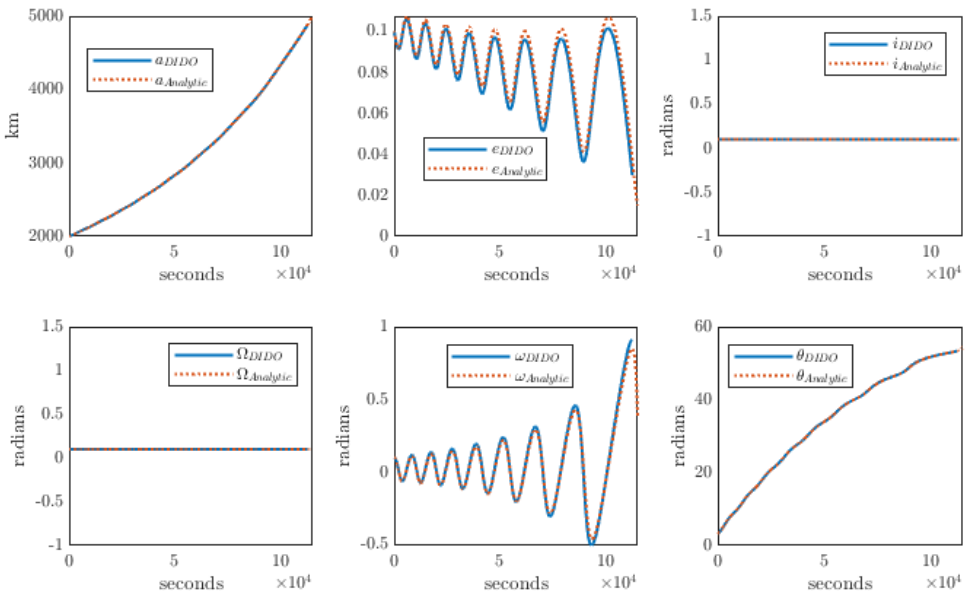


Figure 3.23. Feasibility of the Overconstrained Controls for Maneuver 2.2

Finally, inspection of the control, β, and its associated multiplier, μ_3, show commitment to the Karush-Kuhn-Tucker conditions. Demonstrated in Figure 3.24, there is clear adherence to the updated Karush-Kuhn-Tucker conditions for problem 2.C.

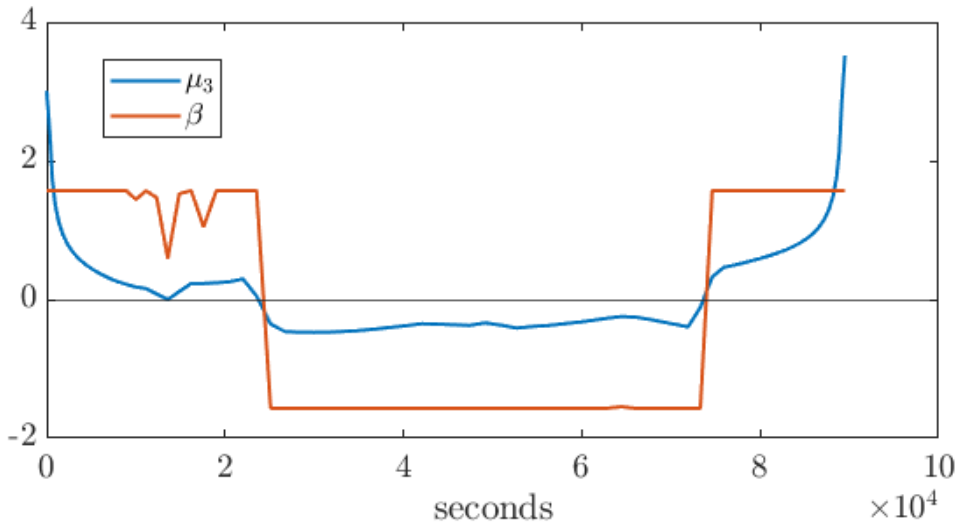


Figure 3.24. Feasibility of the Overconstrained Controls for Maneuver 2.2

3.6 Analysis of Optimal Control Results

As demonstrated in the previous six test problems, stalking horse candidate control algorithms continue to serve as valid tools for considering continuous, low-thrust optimization problems. Particularly when one considers the computational and human costs required to properly formulate, scale, balance, and solve for the infinite potential of trajectory problems related to noncooperative satellite orbit determination and improved space domain awareness, analytic solutions can form the backbone of first guess estimates for reachability. However, it should be stressed that without a clear solution to Bellman's equation, there is no globally optimal solution to these problems.

CHAPTER 4:
Analytical Estimates and Regional Accuracy

This chapter will validate three critical analytical estimates of ΔV. Each of these estimates correlates with a specific control schema that is intended to induce a specific change in the classical orbital elements. Although Battin, Burt, Di Carlo, and Vasile provide additional control schemas that analytically provide specific changes to the elements, only those strategies for $\Delta\overline{a}$, $\Delta\overline{e}$, and $\Delta\overline{i}$, are fully developed in terms of programming implementation and accuracy measurement. These elements were chosen as the focus because they provide a time-invariant description of the shape of the orbit. Ω and ω purely define the orbit orientation. The purpose of this research is to develop an algorithm that serves as an initial decision tool, not a final high-accuracy estimate. Although an understanding of orientation is required for complete mission development or to fully confirm that a satellite could execute a low-energy maneuver, this algorithm informs an analyst if a complete analysis is even necessary. The relative accuracy of each estimate is measured and simple maneuvers are simulated. A novel approach is introduced to further simplify an estimate of cost to change the semi-major axis, and a novel scaling factor is introduced to extend the usability of inclination change cost estimation. Again, the use of classical orbital elements with the $\overline{overline}$ structure ($\overline{a}\,\overline{e}\,\overline{i}\,\overline{\Omega}\,\overline{\omega}\,\theta$), as developed in [15] and introduced in Chapter 2, denotes the *averaged* value of that element over the course of an orbit, as opposed to the instantaneous value of that element ($a\,e\,i\,\Omega\,\omega\,\theta$) at a specific time. Similarly, the continuous acceleration vector is described by the angles α and β as defined by Equation 2.13 reproduced here for convenience.

$$\mathbf{f} = \begin{bmatrix} f_R \\ f_T \\ f_N \end{bmatrix} = \begin{bmatrix} \epsilon \cos\beta \sin\alpha \\ \epsilon \cos\beta \cos\alpha \\ \epsilon \sin\beta \end{bmatrix}$$

4.1 Background

As developed in the background section of Chapter 2, analytical approaches to optimal low-thrust maneuvers have existed conceptually since the first understanding of celestial mechanics. In fact, until the introduction of numerical methods executed by mechanical automata arose with modern computers of the early space age, analytical and semi-analytic methods comprised the main methods of planning optimal finite maneuvers. Both the mission analysis phase of mission management and

the first step of space domain awareness involve a coarse estimate that seeks to assess feasibility. As demonstrated in Chapter 3, the inability to directly solve Bellman's equations for maneuvers of any reasonable complexity leaves any mission designer or space domain awareness analyst stuck at the precipice of regional or reigning optimality, forever unable to determine if they have reached the globally optimal solution.

4.2 Primary Estimates

The maneuvers that represent the majority of the ΔV costs in any transfer, $\Delta\overline{a}\,\Delta\overline{e}\,\Delta\bar{i}$ were selected as the main research focus points. As outlined in Table 4.1, these elements represent half of the necessary descriptive states for the plane match, shape and alignment, and phasing and position matching of another orbit. Additional estimates for a variety of additional maneuvers are included in a separate section for completeness and to enable future research [4], [14], [25], [56]. The full derivations of these strategies are not recreated here; however, an explanation of each strategy and sample transfer maneuver is provided to help explain the equations presented.

Table 4.1. Known Maximum Paths

ΔCOE_{MAX}	Control Angles
$\Delta\overline{a}_{MAX}$	$\alpha = \arctan\left(\frac{e\sin\theta}{1+e\cos\theta}\right) = \gamma, \beta = 0$
$\Delta\bar{i}_{MAX}$	$\beta = sgn(\cos\nu)sgn\left(\bar{i}_f - \bar{i}_0\right)\frac{\pi}{2}$
$\Delta\overline{\Omega}_{MAX}$	$\beta = sgn\left(\overline{\Omega}_f - \overline{\Omega}_0\right))sgn\left(\sin\nu\right)\frac{\pi}{2}$

4.2.1 $\Delta\overline{a}_{MAX}$ - Strategy to change semi-major axis

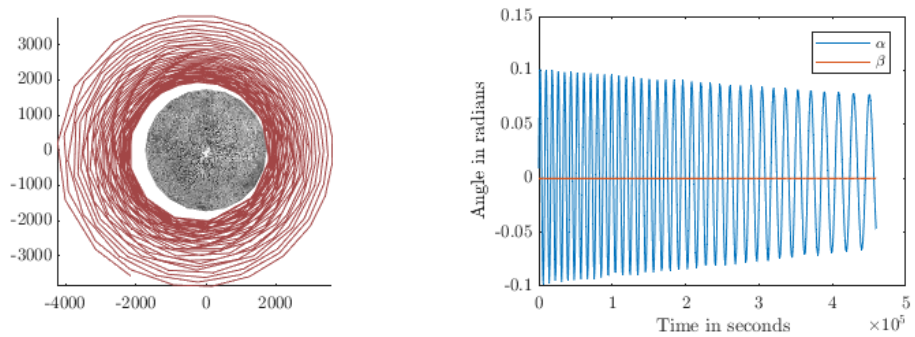


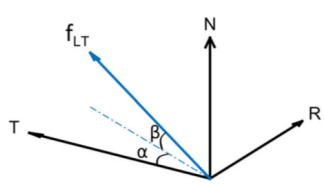
Figure 4.1. Sample State [km] and Control History for the $\Delta\overline{a}_{MAX}$ Strategy

This strategy is derived from a highly simplified analytical control schema for the optimal continuous thrust between circular orbits. The development and analytical proof of this strategy is seen in demonstrations from Lawden, Battin, and Burt from a control perspective, but was simplified and adapted to a simplified cost estimate by Di Carlo and Vasile [4], [14], [25], [53], [56]. Although it was developed as the optimal solution subject to the constraint $e_0 = e_f = 0$, it's use as an initial guess for a solution to a feasible transfer between elliptic orbits was demonstrated in Chapter 2.

The control strategy in terms of the thrust angles α and β for this strategy is [4], [55]:

$$\alpha = \arctan\left(\frac{e\sin(\theta)}{1 + e\cos(\theta)}\right) = \gamma\,,\ \beta = 0 \tag{4.1}$$

where γ is the flight path angle. DiCarlo and Vasile explain that $\alpha = \gamma$ for $a_f > a_o$ and $\alpha = \pi + \gamma$ for cases where $a_0 > a_f$. These control angles provide orientation to the magnitude of the acceleration, ϵ. Inspection of Figure 2.2 reproduced here for convenience,



Acceleration Vector Description [4]

shows that α corresponds to an in-plane angle and β refers to the out-of-plane element as defined by

the orbit plane of the satellite. Logically, it follows that for a strategy that maximizes $\Delta\overline{a}$, $\beta = 0$.

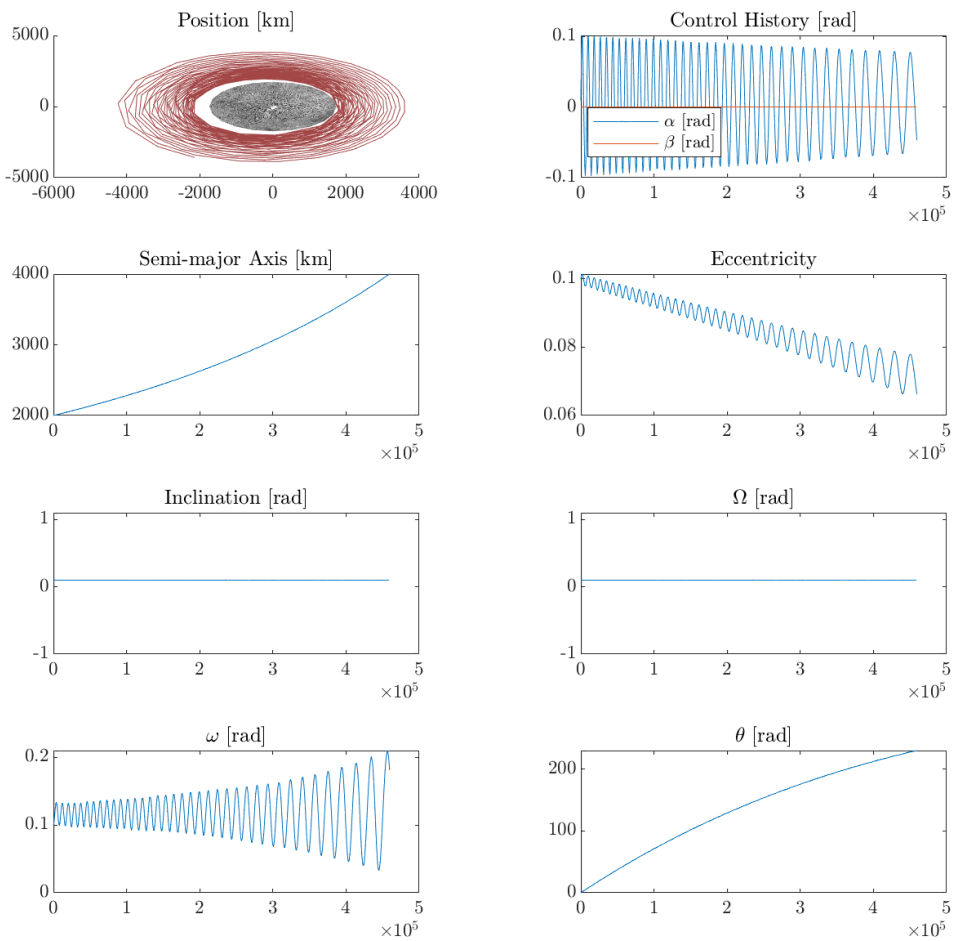


Figure 4.2. Changes to Classical Orbital Elements Over Time for a Sample $\Delta\overline{a}_{MAX}$ strategy.

Di Carlo and Vasile then used an advanced technique from Battin to allow the estimation of elliptical motion [4], [14],

$$f\left(\bar{e}\right) = \frac{2}{1-\bar{e}}\, E_{\text{lc}}\left(\frac{4\bar{e}}{\left(1+\bar{e}\right)^2}\right) + \frac{2}{1+\bar{e}}\, F_{\text{lc}}\left(\frac{4\bar{e}}{\left(1+\bar{e}\right)^2}\right),$$

which relies on the complete elliptic integrals of the first and second kind

$$F_{\mathrm{lc}}(m) = F_1\left(\frac{\pi}{2}, m\right)$$
$$E_{\mathrm{lc}}(m) = E_1\left(\frac{\pi}{2}, m\right).$$

to produce an estimate of the value of $\overline{a}$ over time.

$$\overline{a}(t) = \left[\frac{1}{\overline{a}_0} + \left(\frac{\epsilon(1-\overline{e}^2)f(\overline{e})}{2\pi\sqrt{\mu}}\right)^2 t^2 - sign(\cos(\alpha))\frac{\epsilon(1-\overline{e}^2)f(\overline{e})}{\pi\sqrt{\overline{a}_0\mu}}\right]^{-1}$$

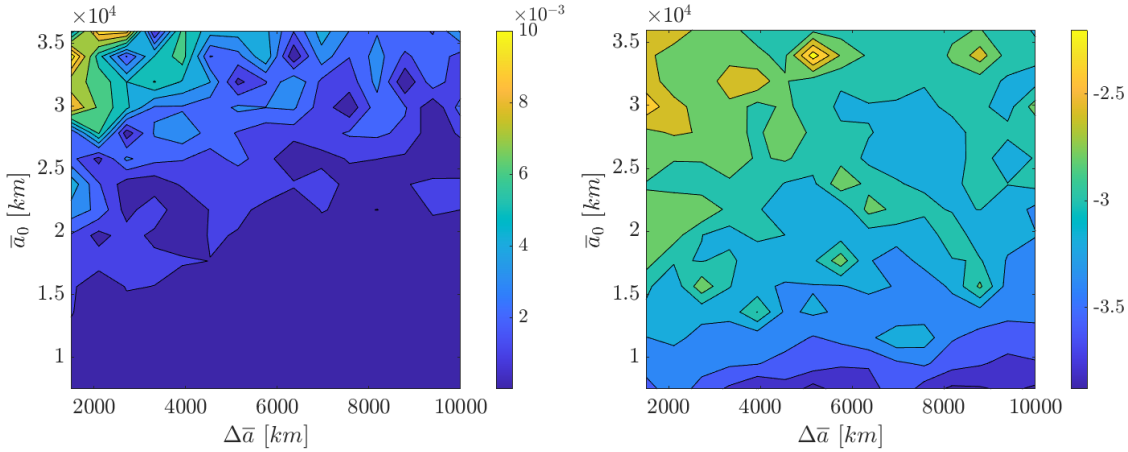
Finally, the time variation of the semi-major axis

$$\frac{d\overline{a}}{dt} = \frac{\Delta\overline{a}_{2\pi}}{T} = sign(\cos\alpha)\frac{\epsilon\overline{a}^{\frac{3}{2}}}{\pi\sqrt{\overline{a}_0\mu}}\left(1-\overline{e}^2\right)f(\overline{e})$$

can be integrated to produce the desired estimate of ΔV

$$\Delta V_{\mathrm{An}} = \epsilon \mathrm{ToF}_{\mathrm{An}} = \frac{2\pi\left|\sqrt{\frac{\mu}{\bar{a}_0}} - \sqrt{\frac{\mu}{\bar{a}_f}}\right|}{(1-\bar{e}^2)\,f(\bar{e})} \tag{4.2}$$

Accuracy and computational efficiency plots for various cases of initial conditions and transfer sizes are provided in Appendix C. In these plots, the variation of ΔV, $\delta\Delta V$ is normalized to the numerical solution provided by the integration of Gauss's equations. For example, if a specific transfer shows approximately 6×10^{-3} variation, then for that case the analytical estimate will compute a ΔV cost 0.6% greater than what Gauss's equations indicate. The variation in computational time is assessed in orders of magnitude as measured by MATLAB's `tic` and `toc` built-in functions. This measures the efficiency in time to compute an estimated solution compared to numerical means. Figure 4.3 shows these products for a sample of transfers with $a_0 = (7500km\, 36000km)$ and $\Delta\overline{a} = (1500km\, 10000km)$.



(a) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$ (b) δ Computation Time, Orders of Magnitude

Figure 4.3. Sample $\delta\Delta V$ and δ Computational Time Plots for the $\Delta\overline{a}_{MAX}$ Strategy

Given these findings, an analytical estimate for the cost of a $\Delta\overline{a}$ maneuver can be used across a broad range of applications. This estimate shows little sensitivity to eccentricity or other initial conditions within a reasonable span.

4.2.2 $\Delta\overline{e}$ - Strategy to change eccentricity

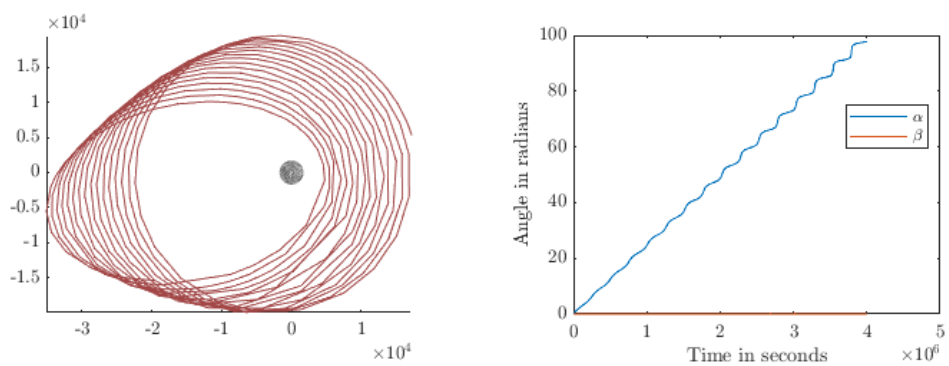


Figure 4.4. Sample State [km] and Control History for the $\Delta\overline{e}$ Strategy

DiCarlo and Vasile's modification of Pollard's control schema for this strategy when $e_f > e_0$ is

$$\alpha = \theta\,,\ \beta = 0. \tag{4.3}$$

When $e_0 > e_f$, the schema is modified to

$$\alpha = \theta + \pi\,,\ \beta = 0. \tag{4.4}$$

DiCarlo and Vasile then substituted this control scheme into a modified set of variational equations that captures the time rate of change with respect to the eccentric anomaly, $\frac{da}{dE}$, $\frac{de}{dE}$, $\frac{d\omega}{dE}$ as shown in 4.4 [4], [26]. This allows the development of an updated expression for the secular rate of change of eccentricity with respect to time. When integrated, this function yields an estimate of the orbit average eccentricity at any time,

$$\overline{e}(t) = \overline{e}_0 + sign(\overline{e}_f - \overline{e}_0)\sin\left(\frac{3}{2}\sqrt{\frac{\overline{a}}{\mu}}\epsilon t\right)$$

which is consistent with the integration findings in Figure 4.5 and allows the development of

$$\Delta V = \epsilon ToF = \frac{2}{3}\sqrt{\frac{\mu}{\overline{a}}}\arcsin(\overline{e}_f - \overline{e}_0) \tag{4.5}$$

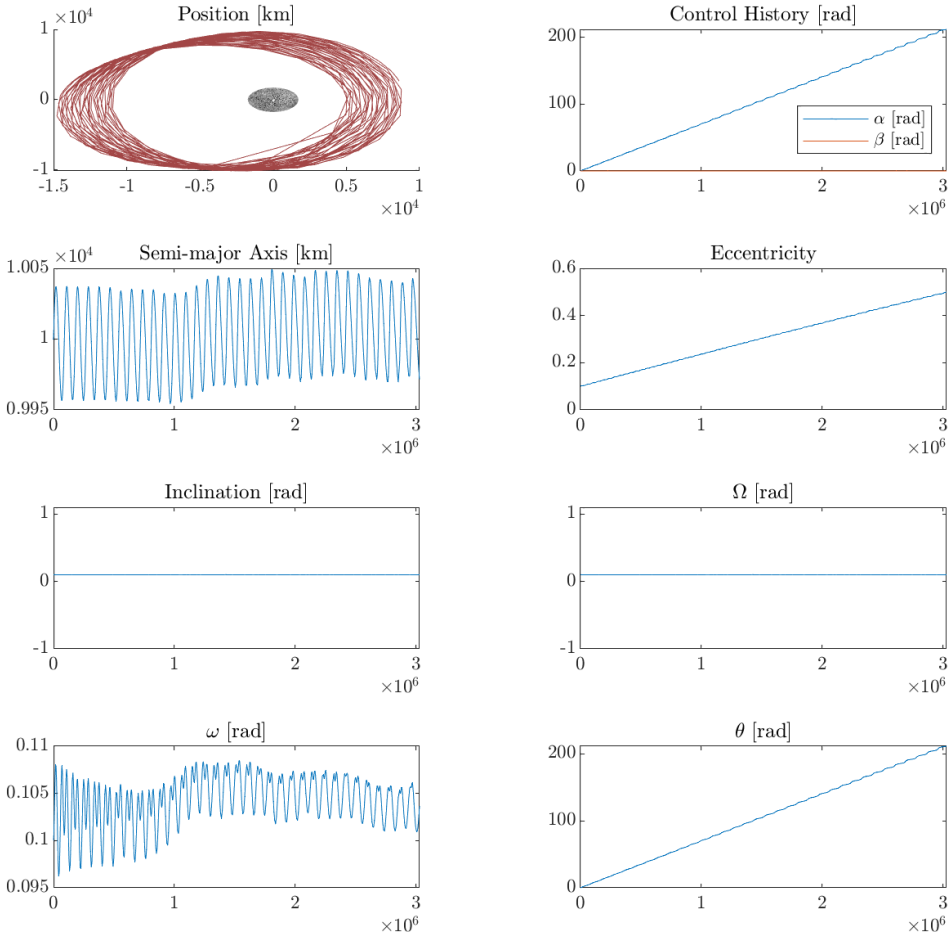
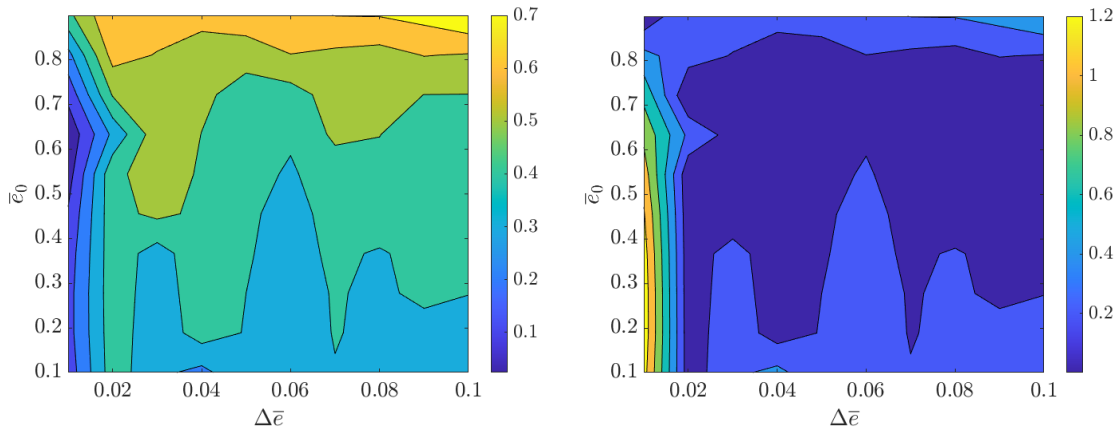


Figure 4.5. Changes in Classical Orbital Elements over Time for a Sample $\Delta\overline{e}$ Strategy.

In this case, a comparison with numerical integration results yields a surprising accuracy space. Figure 4.6 shows this for a sample space built with $a_0 = 60000km$, a wide range of initial eccentricity, $e_0 = (0.01\,,\,0.9)$, and a range of change in eccentricity for each case, $\Delta\overline{e} = (0.01\,,\,0.9)$. Inspection of Figure 4.6(a) indicates a large error of up to 70%. The author applied a scalar factor, annotated as c in Figure 4.6(b), that inverts this error field and transfers the inaccuracies to a small region of extremely small eccentricity changes. For the lunar case, a scalar factor of 2 achieves sufficient regional accuracy for broad applications. Using a scaling factor reduces almost all examined

transferred space to a variation of $< 20\%$ with most of the space at or below 10% variation. These results are shown in Figure 4.6(b). Additional accuracy measurements are provided in Appendix C. Computational efficiency exceeds three orders of magnitude for this estimate compared to the integration of variational equations.



(a) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$ (b) $\delta\Delta V = \frac{\Delta V_{NUM} - c\Delta V_{AN}}{\Delta V_{NUM}}$

Figure 4.6. Sample $\delta\Delta V$ Plots for the $\Delta\overline{e}$ Strategy

Given these findings, a scaled application of the analytical estimate for $\Delta\overline{e}$ can be broadly employed given some initial calibration of the scaling factor for the central body.

4.3 Out-of-plane Maneuver Estimates

This section includes the development and analysis of the final estimate used in the time-invariant description of the orbit and briefly captures the remaining estimates for the out-of-plane maneuvers required for higher-fidelity modeling.

4.3.1 $\Delta\bar{i}$ - Strategy to change inclination

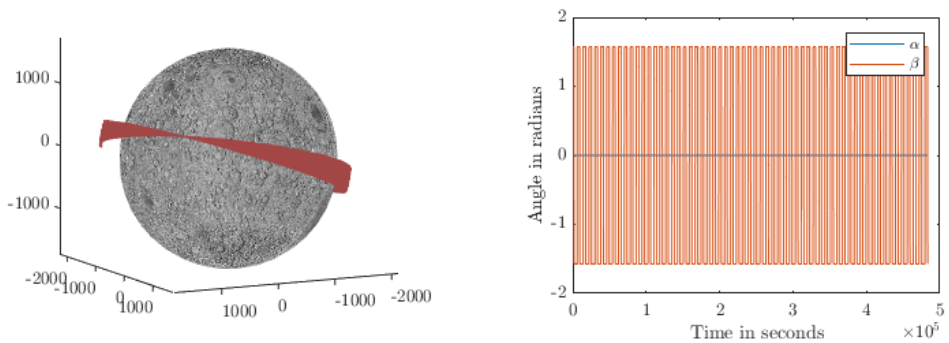


Figure 4.7. Sample State [km] and Control History for the $\Delta\bar{i}$ Strategy

DiCarlo and Vasile's modification of Ruggiero's presented control schema for this strategy is

$$\alpha = 0\,,\ |\beta| = \frac{\pi}{2}. \tag{4.6}$$

In this strategy, the sign of β must switch when crossing the line of nodes [4], [55]. It can also be expressed as

$$\beta = sign(\cos(\nu))sign(\bar{i}_f - \bar{i}_0)\frac{\pi}{2}. \tag{4.7}$$

There is no analytical solution to the secular rate of change of inclination for noncircular orbits that does not affect $\overline{\omega}$ [4]; however, the solution for circular orbits can be used at the expense of accuracy when estimating eccentric transfers and a high variance of performance for various conditions of ω. It can be used to produce:

$$\bar{i}(t) = \bar{i}_0 + sign(\bar{i}_f - \bar{i}_0)\frac{2\epsilon}{\pi}\sqrt{\frac{\overline{a}}{\mu}}t.$$

This linear estimate is again consistent at lower eccentricities with maneuvers like the one shown in Figure 4.8 and yields

$$\Delta V = \epsilon ToF = |\overline{e}_f - \overline{e}_0|\frac{\pi}{2}\sqrt{\frac{\mu}{\overline{a}}} \tag{4.8}$$

as an estimate of ΔV.

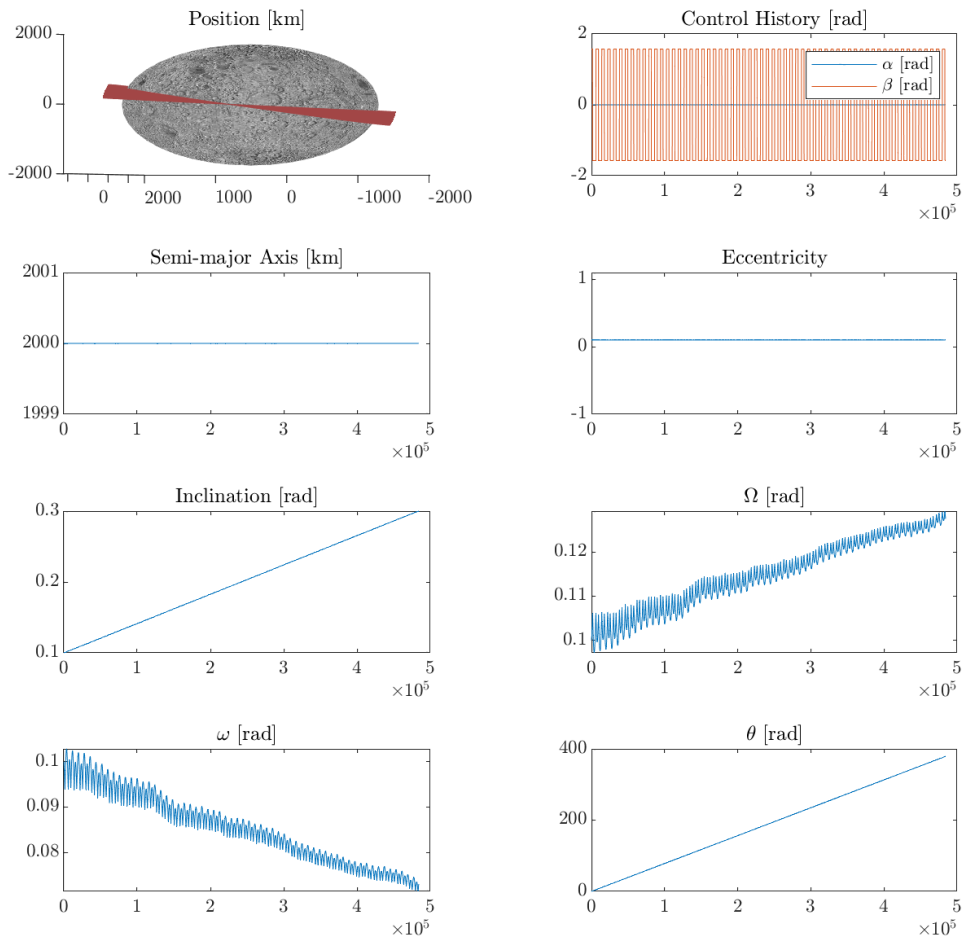
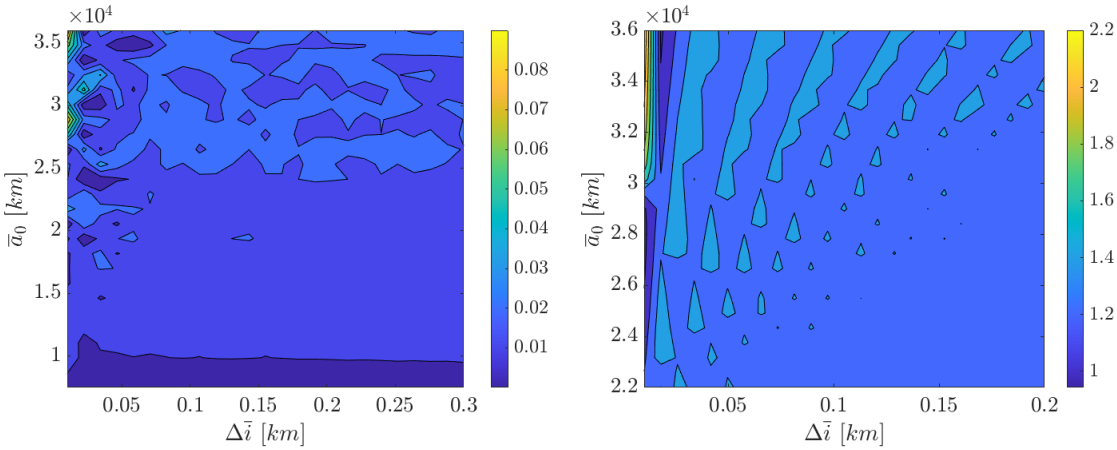


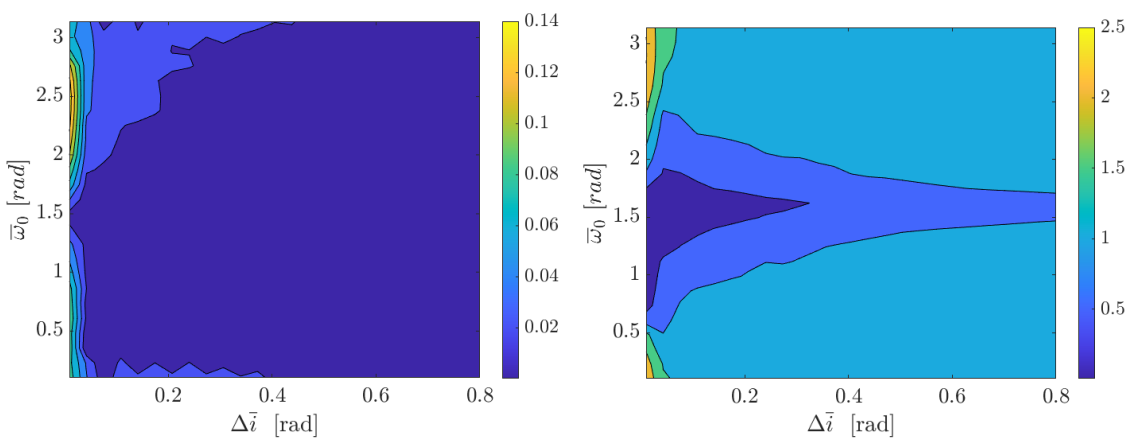
Figure 4.8. Changes in Classical Orbital Elements over Time for a Sample $\Delta\bar{i}$ Strategy.

Modeled accuracy confirms that this estimate is sensitive both to eccentricity and to ω_0 as shown in Figures 4.9 and 4.10.Because this estimate was based on the circular orbit case, variance on the order of 10% is present for small transfers at relatively low eccentricity as shown in Figure 4.9 (a) and extremely problematic results at high eccentricity as shown in Figure 4.9 (b). The relationship is exacerbated by the dependence on ω as shown in Figure 4.10 where the semi-major axis is kept constant and $\overline{\omega}$ is varied.



(a) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$, $e_0 = 0.1$ (b) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$, $e_0 = 0.7$

Figure 4.9. Critical $\delta\Delta V$ Plots for Understanding the Limitations of the $\Delta\bar{i}$ Strategy. $\omega_0 = 0.1$.



(a) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$, $e_0 = 0.1$ (b) $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$, $e_0 = 0.7$

Figure 4.10. $\delta\Delta V$ Plots Demonstrating the Interdependence of the $\Delta\bar{i}$ Strategy and ω. $a_0 = 36000km$.

Given these findings, estimating maneuvers that change a satellite's inclination using this strategy can be used if due consideration is made for the dependence on ω and the increasing sensitivity to eccentricity.

4.4 Additional Estimates

The following analytical estimates were produced by Di Carlo and Vasile and are presented here to provide completeness of the available techniques to the reader [4]. These strategies relate to the final classical orbital elements that complete the description of the alignment and shape of the orbit. Because this research focused on comparison of the estimates to a point mass model, Ω and ω were ignored as the impacts of harmonic perturbations would have had the largest impact on these elements.

4.4.1 $\Delta\Omega$ - Changing the Right Ascension of the Ascending Node

Finally, a control strategy

$$\beta = sign(\overline{\Omega}_f - \overline{\Omega}_0) sign(\sin(\nu)\frac{\pi}{2}\,,\ \alpha = 0$$

produces the maximum instantaneous change to Ω when $\overline{e} = 0$. As previously seen, this available optimal case is then linearized and an analytical estimate can be employed with varying degrees of accuracy for cases where $\overline{e} > 0$. In this case, Ruggiero has already provided the estimate in [55].

$$\Delta V = \epsilon ToF = \frac{\pi}{2}\sqrt{\frac{\mu}{\overline{a}}}|\overline{\Omega}_f - \overline{\Omega}_0)|\sin(\overline{i}) \tag{4.9}$$

4.4.2 $\Delta\omega$ - Changing the Argument of Periapsis

A control strategy from [25],

$$\text{if } sign\left(\bar{\omega}_f - \bar{\omega}_0\right) > 0 \text{ then } \alpha = \begin{cases} 0 & \text{for } 0 \leqslant E < \pi \\ \pi & \text{for } \pi \leqslant E < 2\pi \end{cases},\ \beta = 0$$

$$\text{if } sign\left(\bar{\omega}_f - \bar{\omega}_0\right) < 0 \text{ then } \alpha = \begin{cases} \pi & \text{for } 0 \leqslant E < \pi \\ 0 & \text{for } \pi \leqslant E < 2\pi \end{cases},\ \beta = 0$$

is employed to produce

$$\Delta V = \epsilon ToF = \frac{pi}{2}\sqrt{\frac{\mu}{\overline{a}}}\frac{\overline{e}}{(2-\overline{e}^2)}|\overline{\omega}_f - \overline{\omega}_0)|\,. \tag{4.10}$$

4.5 A Novel Elementary Implementation for the Estimation of $\Delta\overline{a}$

Given DiCarlo and Vasile's analytical estimate for the ΔV costs of transfers operating under the maximum $\Delta\overline{a}$ strategy (Equation 4.2) [4]

$$\Delta V_{\text{An}} = \epsilon \text{ToF}_{\text{An}} = \frac{2\pi \left| \sqrt{\frac{\mu}{\bar{a}_0}} - \sqrt{\frac{\mu}{\bar{a}_f}} \right|}{\left(1 - \bar{e}^2\right) f\left(\bar{e}\right)}$$

where

$$f\left(\bar{e}\right) = \frac{2}{1-\bar{e}} E_{\text{lc}} \left(\frac{4\bar{e}}{\left(1+\bar{e}\right)^2} \right) + \frac{2}{1+\bar{e}} F_{\text{lc}} \left(\frac{4\bar{e}}{\left(1+\bar{e}\right)^2} \right)$$

requiring the complete elliptic integrals of the first and second kind:

$$F_{\text{lc}}\left(m\right) = F_1\left(\frac{\pi}{2}, m\right)$$
$$E_{\text{lc}}\left(m\right) = E_1\left(\frac{\pi}{2}, m\right)$$

which MATLAB executes as

$$F_1\left(\frac{\pi}{2}|m\right) = \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 - m\ \sin^2\theta}} d\theta$$
$$E_1\left(\frac{\pi}{2}|m\right) = \int_0^{\frac{\pi}{2}} \sqrt{1 - m\ \sin^2\theta}\ \ d\theta.$$

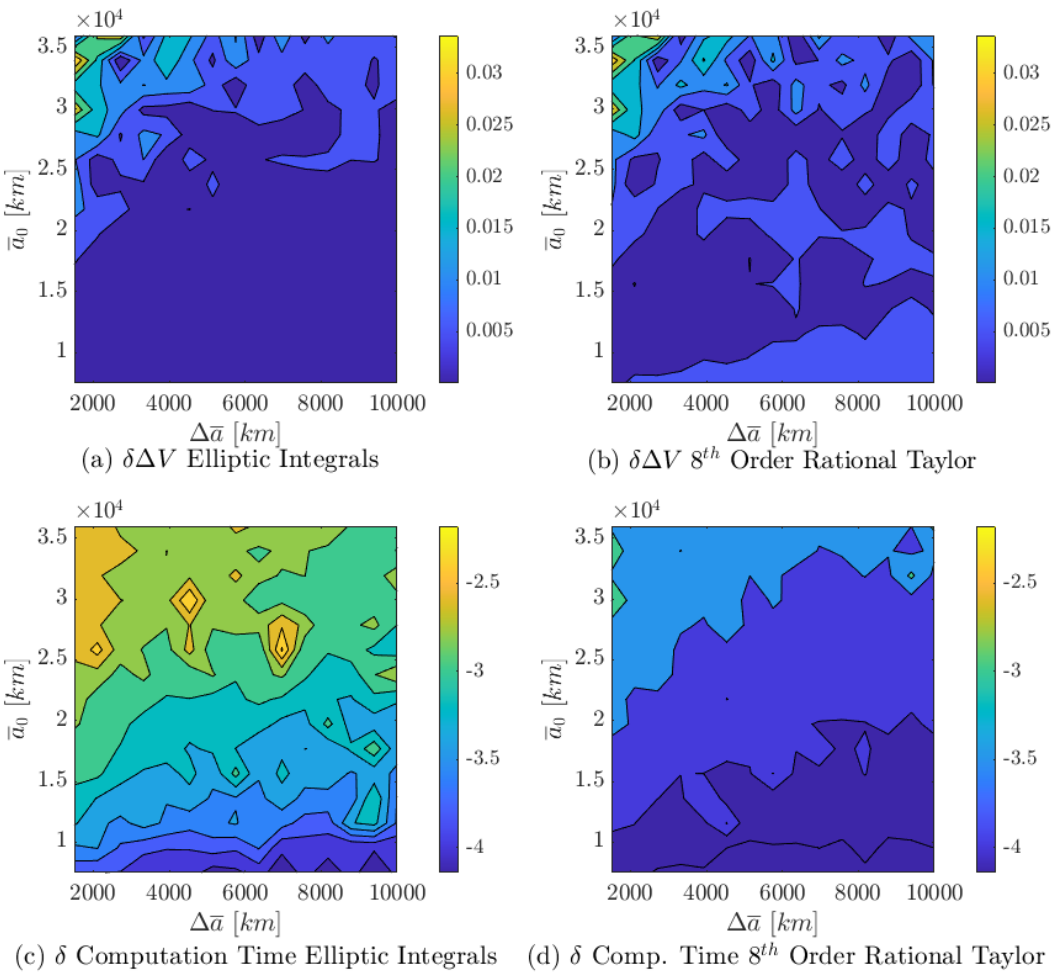
The author proposes approximating the entire function, $f(\overline{e})$, using a series expansion to simplify the calculation and minimize computational costs. Approximated functions are annotated as $f\left(\tilde{\bar{e}}\right)$. A simplified, though still complex, function can be found:

$$f\left(\tilde{\bar{e}}\right) = \frac{45\,\pi\,\bar{e}^4}{32} + \frac{3\,\pi\,\bar{e}^2}{2} + 2\,\pi \tag{4.11}$$

This fifth-order Taylor-MacLaurin expansion is now in an elementary form, but it is still complex. The truncation of π to 3.14 and the expansion to the 8th order rationalize the function and ensure that it matches sufficiently the original elliptic integrals for most uses. This approximation is annotated $f\left(\tilde{\tilde{\bar{e}}}\right)$ to indicate the additional truncation of π.

$$f\left(\tilde{\tilde{\bar{e}}}\right) = \frac{175\ (3.14)\ \bar{e}^6}{128} + \frac{45\ (3.14)\ \bar{e}^4}{32} + \frac{3\ (3.14)\ \bar{e}^2}{2} + 2\,(3.14)$$

The performance of the 8th order rational Taylor expansion can be numerically compared to the elliptic integrals method.



(a) $\delta\Delta V$ Elliptic Integrals (b) $\delta\Delta V$ 8th Order Rational Taylor

(c) δ Computation Time Elliptic Integrals (d) δ Comp. Time 8th Order Rational Taylor

Figure 4.11. $f(\bar{e})$ of a 8th Order Rational Taylor Series and Elliptic Integrals with $e_0 = 0.3$.

Figure 4.11 compares the both $f(\bar{e})$ functions for a transfer with $e_0 = 0.3$ to the performance of MATLAB's ode15s function to integrate Gauss's equations. Results for normalized accuracy and computational advantage are shown for each. $\delta\Delta V = \frac{\Delta V_{NUM} - \Delta V_{AN}}{\Delta V_{NUM}}$ Computation cost is measured in orders of magnitude. If a higher accuracy is required, the series can be further expanded. A comparison between the use of elliptic integrals and rational 8th order Taylor expansion to compute

$f(\overline{e})$ are shown in Figure 4.12.

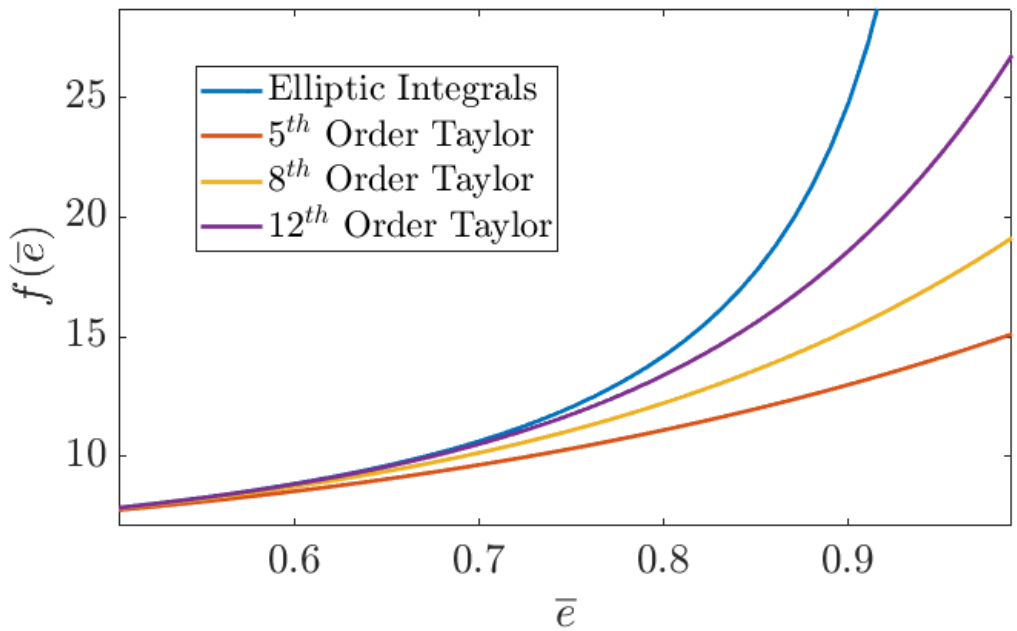


Figure 4.12. Taylor Series as a Replacement for Elliptic Integrals

The major benefit of this approach is not the increase in computational efficiency, but the translation of the equation into one that is both rational and comprised of elementary functions. This enables ΔV calculation using this estimate without the need for a program (or human) with advanced mathematical computational capabilities.

CHAPTER 5: A Generalized Algorithm to Estimate Low-Thrust Maneuvers

This chapter demonstrates the use of analytical estimates of ΔV costs for simple continuous low-thrust maneuvers within the lunar sphere of influence. A generalized algorithm that leverages analytical estimates for changes to the semi-major axis, eccentricity, and inclination is described and executed in several test scenarios. These estimates are compared to numerical methods for complex transfers involving long-duration changes to many classical orbital elements. As a tool for an orbital analyst, these algorithms are limited to estimating the cost of an orbit transfer in ΔV for an orbit within a restricted two-body model. The estimates that underpin this algorithm were derived from Gauss's variational equations. As such, the input to these estimates may be heavily influenced by extremal behavior in other elements because of the strongly coupled, stiff differential equations that represent the time-based rates of change. Many of the estimates were derived from singular cases in which $e_0 = 0$ and can suffer from increasing error as eccentricity increases or for extreme cases. Additional background information on the causes and severity of these limitations is available in Chapters 2 - 4.

The algorithm intentionally assumes a false concept known as "piecewise optimality." This means that the algorithm will solve for the optimal cost of a transfer for each of the necessary classical orbital elements in isolation. While the calculations will account for the impact of coupled elements within a single estimate, this estimate does not propagate the orbit. These equations do not provide a complete state history, and as such, their error grows continuously as each previous estimate makes the initial conditions for the next less accurate. The minimum necessary information describes the shape of the initial and final orbits ($\mathbf{x}_0$ and $\mathbf{x}_f$), the final time condition of the final orbit (t_f), and the continuous acceleration capability of the spacecraft(ϵ). The algorithm will use the initial and final classical orbital elements to calculate the transfer costs in ΔV. If an estimate of the magnitude of acceleration produced by the engine, ϵ, is known, then a duration of the transfer can be estimated as well.

5.1 Low-Energy Moon-Earth Transfer Reachability

This example problem models a noncooperative spacecraft in low-lunar orbit. As developed in Chapters 1 and 2, we return to the perspective of the space domain awareness analyst asking: "can this spacecraft execute a low-energy return?" Practical examples for this first order screening question

have been scoped to the elements of a, e and i. While the complete orbit for a specific epoch still requires Ω, ω, and θ, such an estimate demands significantly more information to accurately model even for low-accuracy answers. Additionally, elements sensitive to singularities have been offset from zero by 0.1 (i_f, Ω_f, ω_f). This algorithm does not confirm or deny if the spacecraft will make the low-energy transfer, or even if it is feasible to do so. This algorithm simply acts as a first level screen that helps an analyst determine if more analysis is necessary. The complete information is presented in the dynamics, endpoints, and parameters of the reachability example. Piecewise optimality is assumed and estimates are applied as such. The corresponding control histories are propagated to demonstrate what this piece-wise optimal transfer would look like.

$$
\begin{cases}
\text{Dynamics} & \frac{da}{dt} = \frac{2a^2}{h}\epsilon\cos\beta\left(e\sin\theta\sin\alpha + \frac{p}{r}\cos\alpha\right) \\
& \frac{de}{dt} = \epsilon\cos\beta\frac{1}{h}\left(p\sin\theta\sin\alpha + [(p+r)\cos\theta + re]\cos\alpha\right) \\
& \frac{di}{dt} = \epsilon\frac{r}{h}\cos\nu\sin\beta \\
& \frac{d\Omega}{dt} = \epsilon\frac{r}{h}\frac{\sin\nu}{\sin i}\sin\beta \\
& \frac{d\omega}{dt} = \frac{1}{he}\epsilon\cos\beta\left[-p\cos\theta\sin\alpha + (p+r)\sin\theta\cos\alpha\right] - \frac{r\sin\nu\cos i}{h\sin i}\epsilon\sin\beta \\
& \frac{d\theta}{dt} = \frac{h}{r^2} + \frac{\epsilon}{eh}\cos\beta\left[p\cos\theta\sin\alpha - (p+r)\sin\theta\cos\alpha\right] \\
\text{Endpoints} & a_0 = 1761,9 \quad [km](\text{altitude at perigee of } 200km) \\
& e_0 = 0.1 \\
& i_0 = 1.0 \quad [rad] \\
& \Omega = 0.1 \quad [rad] \\
& \omega = 0.1 \quad [rad] \\
& \theta = \pi \quad [rad] \\
& a_f = 2.5362 \times 10^4 \quad [km] \\
& e_f = 0.9275 \\
& i_f = 0.1 \quad [rad] \\
\text{Parameters} & \mu = \mu_{☾} \\
& \epsilon = 10 \times 10^{-6} \quad [km/s^2]
\end{cases}
$$

In this scenario, the satellite in question is observed in a low-lunar orbit with an altitude at perigee of $200km$ above the lunar surface. Although it is not known for certain, previous versions of this spacecraft have had low-thrust engines capable of sustaining a constant acceleration of $\epsilon = 10 \times 10^{-6} \quad [km/s^2]$. The analyst knows that the next opportunity for a low thrust transfer is in approximately 30 days. Is the satellite capable of conducting a low-thrust maneuver to reach the transfer state, $\mathbf{x}_f$, in time ($ToF \leqslant 30days$)?

Algorithm 2 Direct Orbit Shape and Inclination Only

1: Record Problem Values:

2: Record $\mathbf{x}_0 = [a_0\, e_0\, i_0]^T$

3: Record $\mathbf{x}_f = [a_f\, e_f\, i_f]^T$

4: Estimate $[t_0\, t_f]$

5: Estimate ϵ

6: Compute the cost in ΔV to move each element of $\mathbf{x}^0$ to $\mathbf{x}^f$:

$$f(\bar{e}) = \frac{2}{1-\bar{e}} E_{\text{lc}}\left(\frac{4\bar{e}}{(1+\bar{e})^2}\right) + \frac{2}{1+\bar{e}} F_{\text{lc}}\left(\frac{4\bar{e}}{(1+\bar{e})^2}\right)$$

$$\text{where } F_{\text{lc}}(m) = F_1\left(\frac{\pi}{2}, m\right), \text{ and } E_{\text{lc}}(m) = E_1\left(\frac{\pi}{2}, m\right)$$

$$\Delta V_1(a_0, a_f) = \frac{2\pi\left|\sqrt{\frac{\mu}{\bar{a}_0}} - \sqrt{\frac{\mu}{\bar{a}_f}}\right|}{(1-\bar{e}^2)\, f(\bar{e})}$$

$$\Delta V_2(e_0, e_f) = \frac{2}{3}\sqrt{\frac{\mu}{a}}\left|\arcsin\left(e_f - e_0\right)\right|$$

$$\Delta V_3(i_0, i_f) = \frac{\pi}{2}\sqrt{\frac{\mu}{a}}\left|\arcsin\left(i_f - i_0\right)\right|$$

7: Compute the total cost in terms of ΔV:

$$\Delta V_{total} = \sum_{i=1}^{3} \Delta V_i$$

8: Compute the total time of flight for the transfer, ToF

$$ToF = \frac{\Delta V_{total}}{\epsilon}$$

9: Analyze maneuver against the known t_f

if $t_0 + ToF \leqslant t_f$ **then** maneuver is possible

if $t_0 + ToF > t_f$ **then** maneuver might not be possible*

if $t_0 + ToF \gg t_f$ **then** maneuver unlikely

*an examiniation of accuracy is required

Following Algorithm 2, endpoint conditions and other known problem elements are provided in the problem description. The maximum control strategy to increase the semimajor axis is then employed until the spacecraft has reached a_f. Analytical estimates anticipate a ΔV cost of $2.1382\left[\frac{km}{s}\right]$, or a burn time of about $24.7[days]$. The propagation results of this strategy yield a ΔV cost of $2.2207\left[\frac{km}{s}\right]$, or a burn time of $25.7[days]$. While we see excellent performance in capturing the burn time for the change to $\overline{a}$, there is now a significant change to the $\overline{e}$ of the orbit at this transfer point.

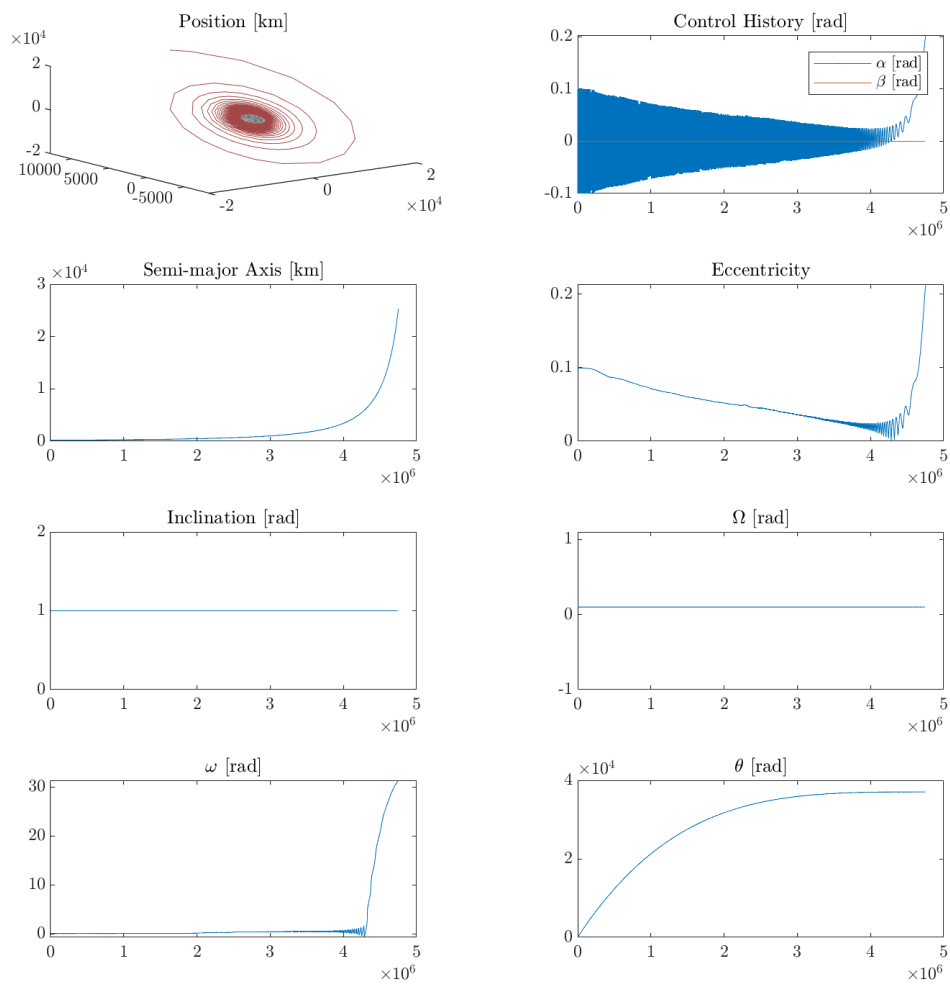


Figure 5.1. State Changes for ΔV_1 the $\Delta\overline{a}$.

Following the $\Delta\overline{a}$ maneuver, a $\Delta\bar{i}$ maneuver is performed. The initial conditions for $\overline{a}$, $\overline{e}$, and $\overline{\omega}$ have changed compared to the first maneuver.

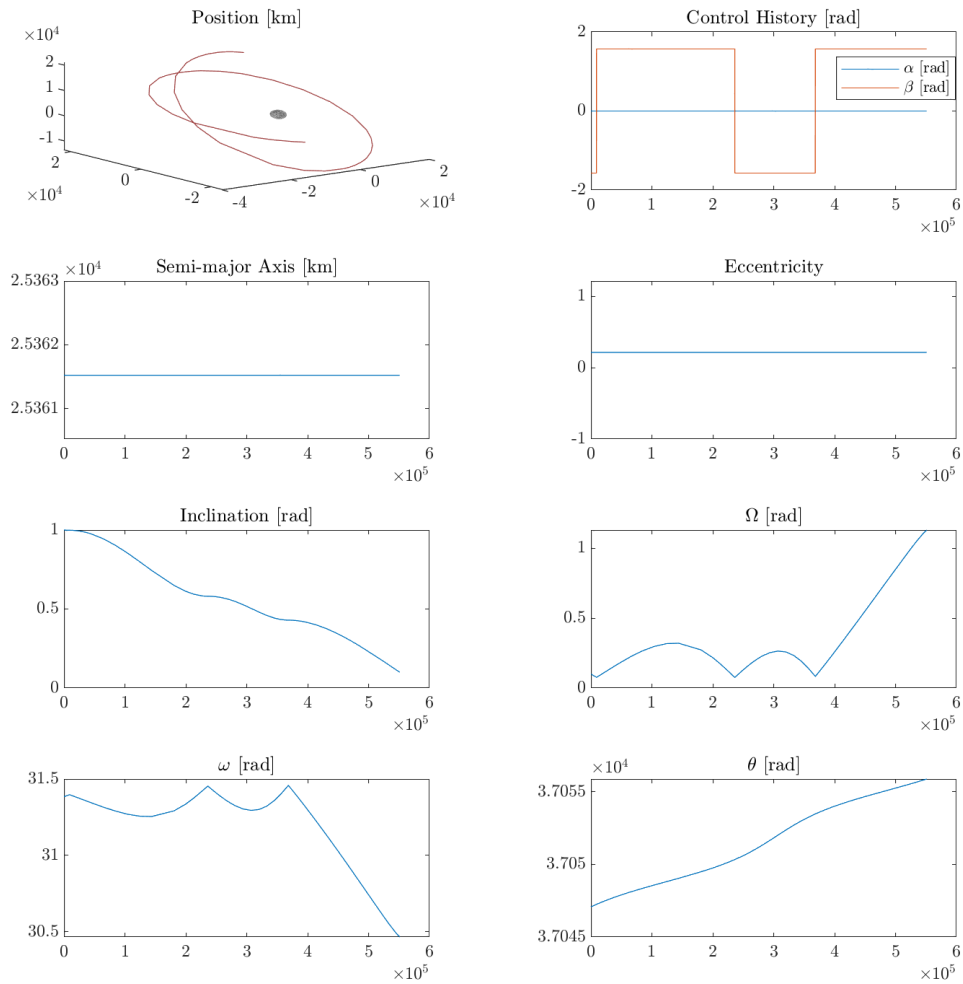


Figure 5.2. State Changes for ΔV_2 the $\Delta\bar{i}$.

Finally, $\Delta\overline{e}$ maneuver is performed. The initial conditions for $\bar{i}$, $\overline{\Omega}$, and $\overline{\omega}$ have changed compared to the second maneuver.

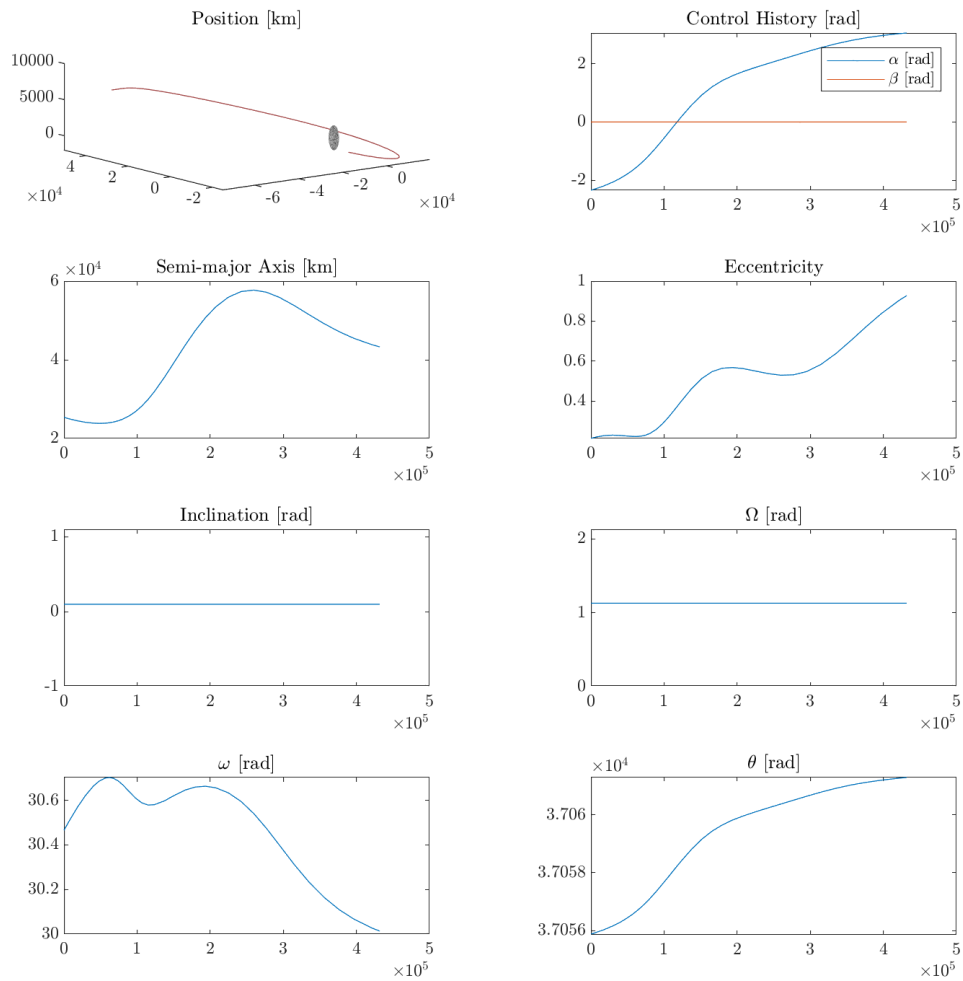


Figure 5.3. State Changes for ΔV_3 the $\Delta\overline{e}$.

Following the final maneuver, the engine is shut off and the spacecraft continues on the new orbit described by

$$\mathbf{x}_{tf} = [\overline{a}_{tf}\overline{e}_{tf}\overline{i}_{tf}\overline{\Omega}_{tf}\overline{\omega}_{tf}\theta]^T$$

where the complete state history for the this transfer is seen in Figure 5.4.

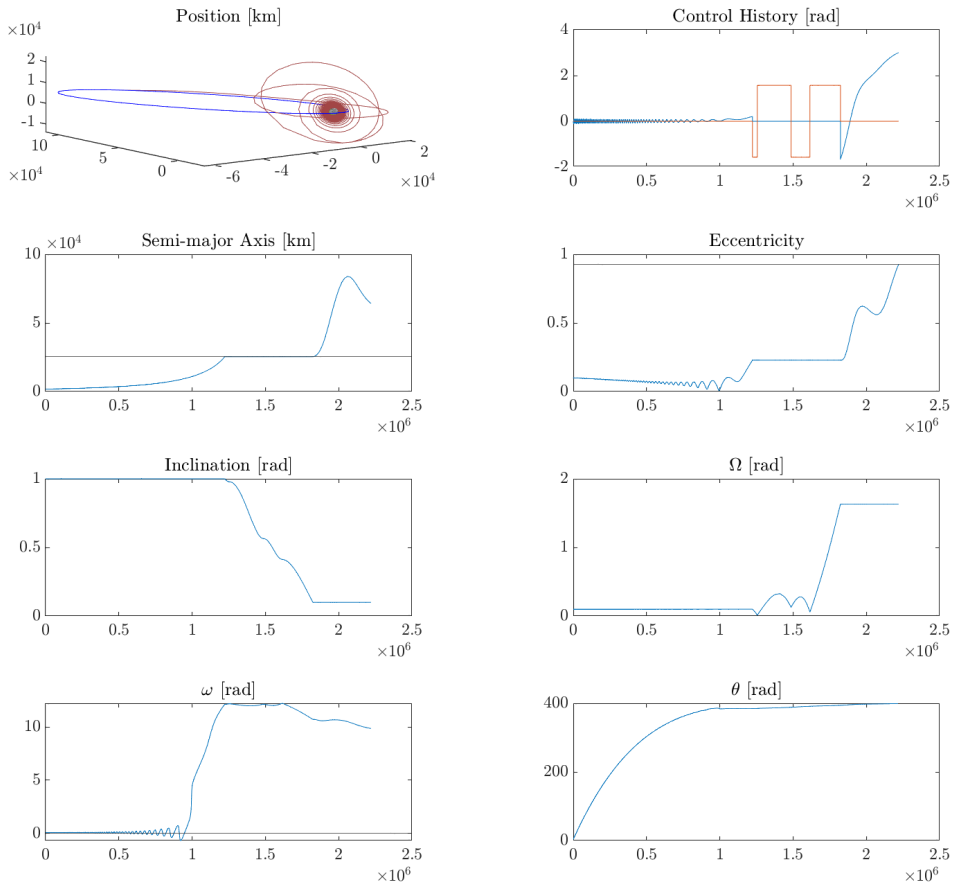


Figure 5.4. State Changes for the One-step Transfer.

Following the $\Delta\overline{a}$ maneuver, a $\Delta\bar{i}$ maneuver is performed. The initial conditions for $\overline{a}$, $\overline{e}$, and $\overline{\omega}$ have changed compared to the first maneuver. This results in a significant "overshoot" of the targeted orbit if one were to directly implement these strategies as described. Figure 5.5 shows the resulting error quite clearly. The target orbit is shown in black dashes while the final orbit produced by this strategy is shown in blue.

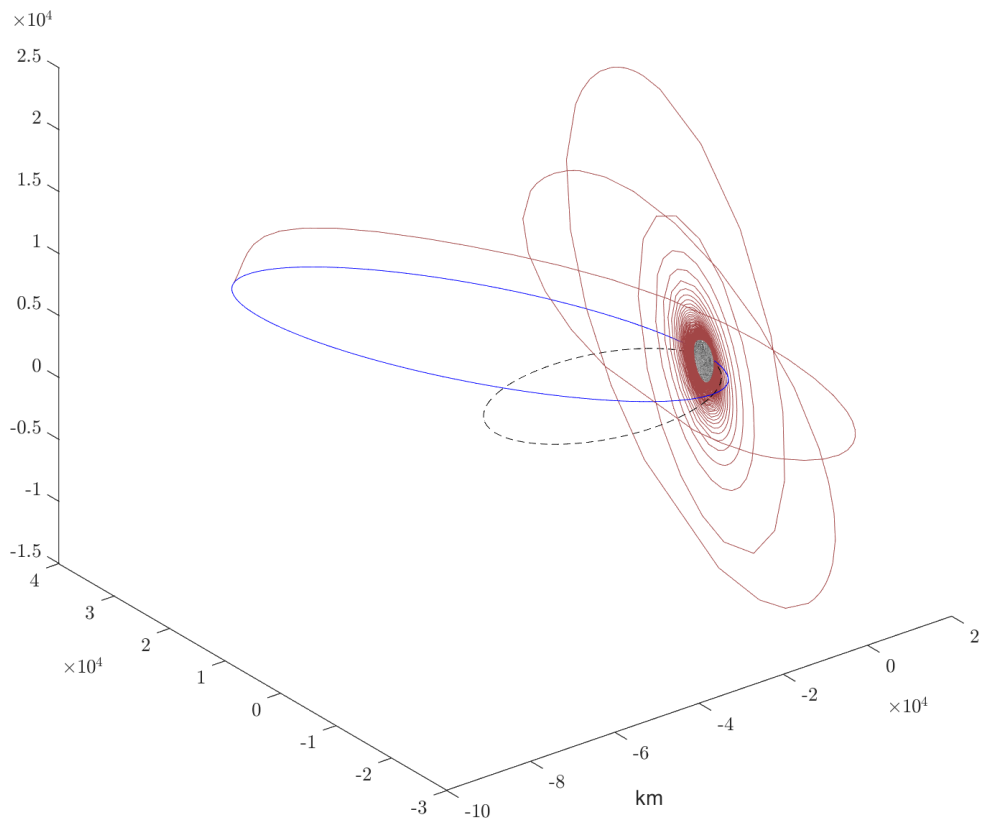


Figure 5.5. State History for a Direct, Single-step Transfer.

Table 5.1. ΔV & ToF for the Direct Single-Step Transfer

Source	ΔV	ToF
Analytic Estimate	$2.1382 \frac{km}{sec^2}$	$24.7478\, days$
Gauss's Equations	$2.2207 \frac{km}{sec^2}$	$25.702\, days$

Table 5.1 shows the computed time of flight and ΔV for both the analytical estimate and the integration of Gauss's equations. Figure 5.6 shows the resulting orbit aligned in Ω and ω for a clearer representation of the scale.

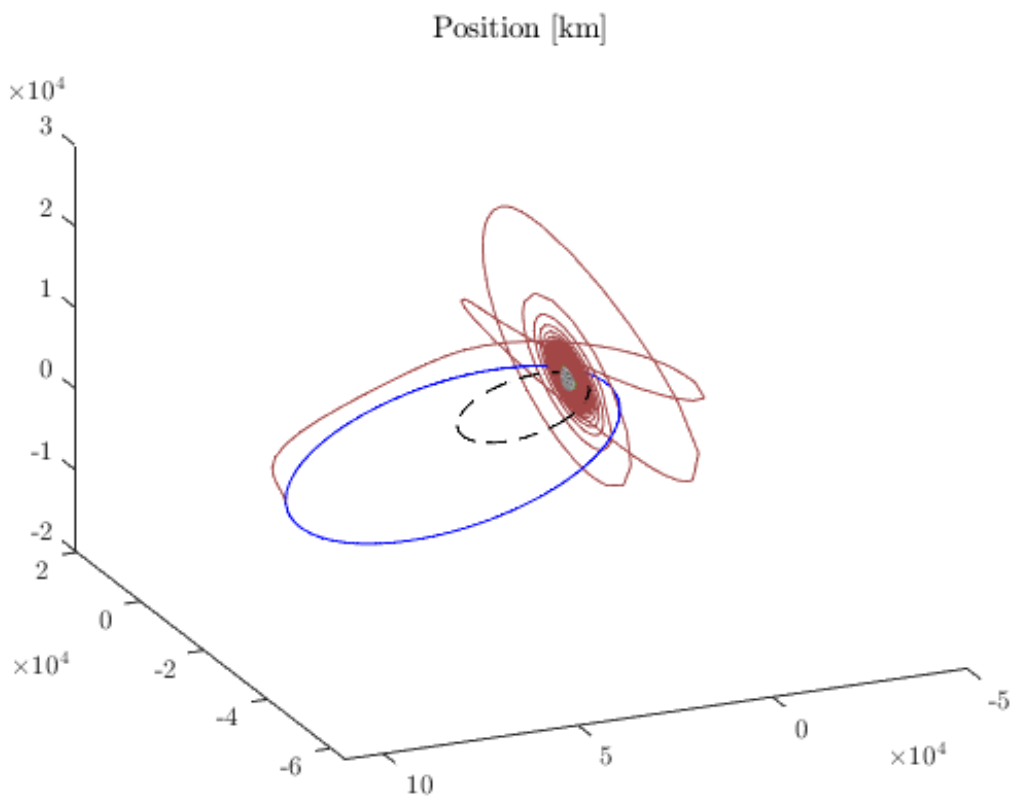


Figure 5.6. State History for a Direct, Single-step Transfer Aligned with Target Orbit.

Although this result may lead some to question the logic of this approach in estimating complex maneuvers, it is important to recall that the (false) concept of piecewise optimality is a critical assumption for the algorithm. Figure 5.5 demonstrates that the satellite is capable of achieving an orbit that far exceeds the orbital parameters of the target orbit in the time from t_0 to t_f. In the context of initial screening criteria, this indicates that a transfer orbit is possible and that further analysis is definitely warranted.

5.1.1 Multi-step Estimates

Previous efforts demonstrated that employment of the maximum rate of change strategies for one orbital element have carry-over effects on other elements due to the coupled dynamics. While the analytical estimates do not consider these changes as they relate to the calculations, stepwise implementation of these strategies through propagation of Gauss's equations does affect the final state and outcome if these conditions are not accounted for.

To demonstrate this difference, the same maximum deviation strategies are applied, but this time twice each. The general strategy is modified to first attain half of the needed changes to each of the classical orbital elements and then, from that midpoint orbit, to generate the remaining changes with the same control strategies. From an integration perspective, this minimizes the impact of coupled element changes and allows for a more accurate final position in terms of all classical elements.

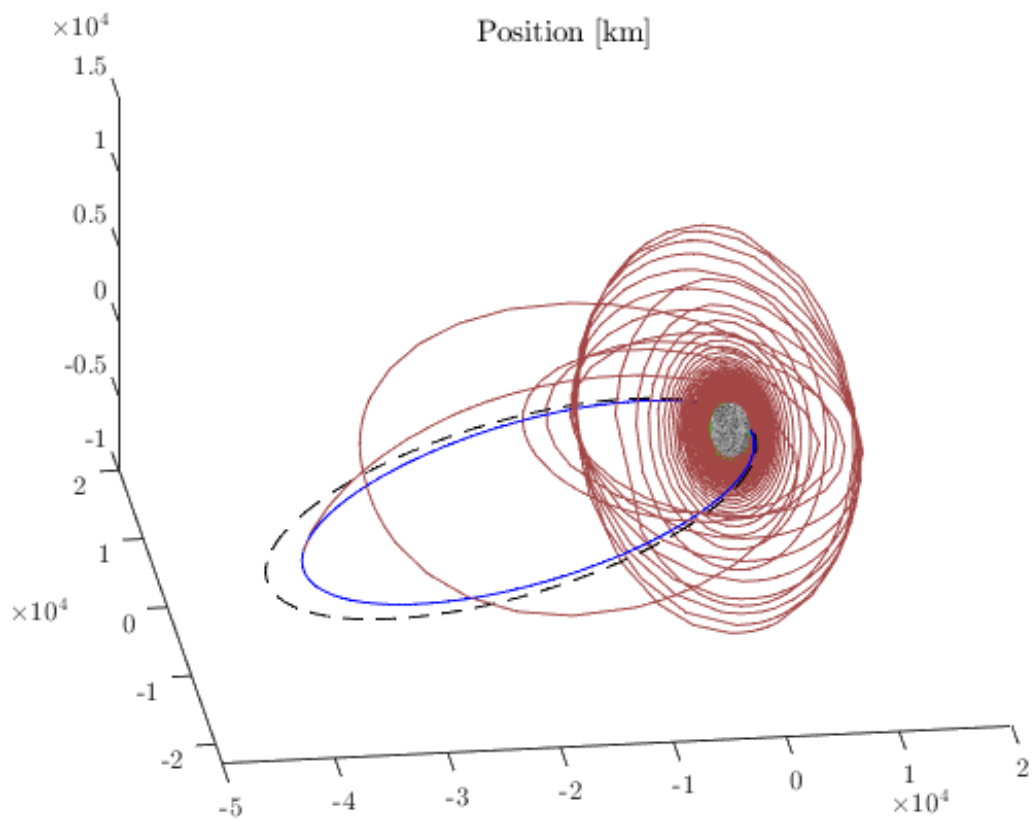


Figure 5.7. State History for a Two-step Transfer, Aligned with Target Orbit.

The complete state histories for this two-step transfer and the state changes by element contain both intuitive and non-intuitive aspects. Inspection of the state changes in Figure 5.8 show a familiar control history that repeats at about two-thirds of the way through the maneuver. This is to be expected as the second run of changes to the classical elements is enacted. The complete state history for this strategy is shown in Figure 5.7. This strategy has no effect on the analytical estimate, as there is no change to $\mathbf{x}_0$ and $\mathbf{x}_f$. Table 5.2 shows the cost results of this transfer strategy. This is not intuitive. In this case a series of shorter, more accurate maneuvers to a final state much closer to the transfer shape cost the mission manager more in terms of ΔV and in time when compared to the simple, direct method described in the previous section. In this case a satellite that employs this control strategy will not be in the proper transfer orbit to execute a low-energy moon-Earth return. The analytical estimate would have, in this case, encouraged the analyst to conduct more thorough and accurate assessments on a mission that would not enter the return trajectory during this cycle.

Table 5.2. ΔV & ToF for the Direct Two-Step Transfer

Source	ΔV	ToF
Analytic Estimate	$2.1382 \frac{km}{sec^2}$	$24.7478\, days$
Gauss's Equations	$2.7322 \frac{km}{sec^2}$	$31.6225\, days$

The algorithm is designed to under-promise and over deliver. By assuming piecewise optimality, it chooses a strategy that very likely can be beat; however, it does so in exchange for guaranteeing feasibility. This will consistently produce underestimates of both ΔV and total flight time. Here an overly optimistic result from the algorithm is preferred when it is used as an initial and early screening tool to identify satellites possibly capable of reaching the transfer orbit.

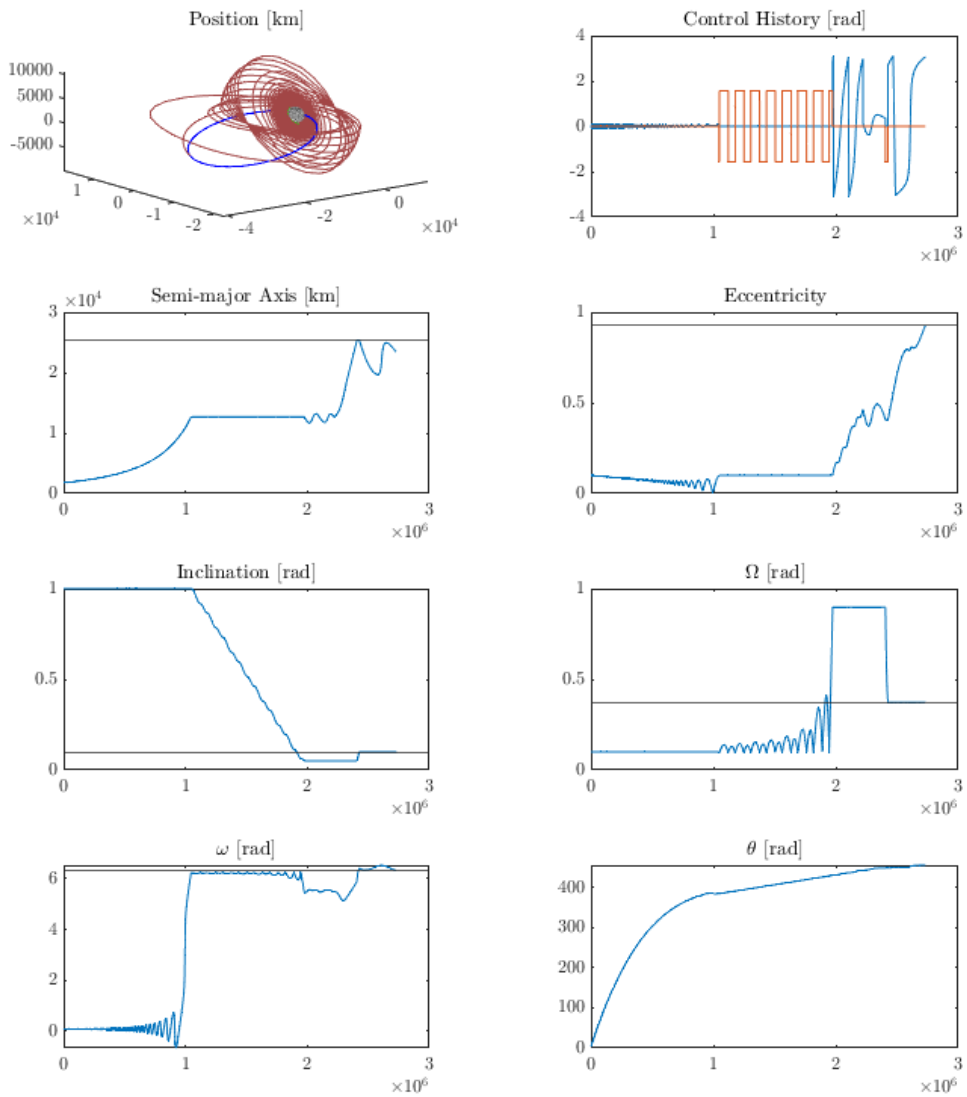


Figure 5.8. State Changes for the Two-step Transfer.

Finally, a modified version of this algorithm, utilizing the simplified estimate for $f(\overline{e}$ developed by the author. This modification is presented as Algorithm 3 and produces an estimated time of flight of 24.7550 days. This is a total error of 3.68% from the numerical solution and represents a 0.03% error improvement from the elliptic integral estimate.

Algorithm 3 Direct Orbit Shape and Inclination - Elementary and Rational Estimate

Record Problem Values:

  Record $\mathbf{x}_0 = [a_0\, e_0\, i_0]^T$

  Record $\mathbf{x}_f = [a_f\, e_f\, i_f]^T$

  Estimate $[t_0\, t_f]$

  Estimate ϵ

Compute the cost in ΔV to move each element of $\mathbf{x}^0$ to $\mathbf{x}^f$:

$$
\begin{aligned}
f\left(\bar{e}\right) &= \frac{175\ (3.14)\ \bar{e}^6}{128} + \frac{45\ (3.14)\ \bar{e}^4}{32} + \frac{3\ (3.14)\ \bar{e}^2}{2} + 2\,(3.14) \\
\Delta V_1(a_0, a_f) &= \frac{2\pi \left|\sqrt{\frac{\mu}{\bar{a}_0}} - \sqrt{\frac{\mu}{\bar{a}_f}}\right|}{(1-\bar{e}^2)\, f\left(\bar{e}\right)} \\
\Delta V_2(e_0, e_f) &= \frac{2}{3}\sqrt{\frac{\mu}{a}}\left|\arcsin\left(e_f - e_0\right)\right| \\
\Delta V_3(i_0, i_f) &= \frac{\pi}{2}\sqrt{\frac{\mu}{a}}\left|\arcsin\left(i_f - i_0\right)\right|
\end{aligned}
$$

Compute the total cost in terms of ΔV:

$$\Delta V_{total} = \sum_{i=1}^{3} \Delta V_i$$

Compute the total time of flight for the transfer, ToF

$$ToF = \frac{\Delta V_{total}}{\epsilon}$$

Analyze maneuver against the known t_f

if $t_0 + ToF \leqslant t_f$ **then** maneuver is possible

if $t_0 + ToF > t_f$ **then** maneuver might not be possible*

if $t_0 + ToF \gg t_f$ **then** maneuver unlikely

*an examiniation of accuracy is required

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6: Conclusion

This chapter concludes the thesis with additional recommended applications for the analytical estimates of ΔV costs for continuous low-thrust orbital maneuvers. Three possible focus areas for follow-on research are proposed. The benefits and challenges of including select elements in lower astrodynamics courses as a way to expose students to continuous low-thrust earlier in their education are assessed. Finally, borne with the benefit of hindsight, reflections on ways that the author would have conducted the research differently are captured.

6.1 Continued Research Recommendations

Future research efforts could continue along any of the three major areas used in this work.

6.1.1 Dynamics Focus

For those wishing to improve the implementation of available analytical estimates, several nonoptimal but feasible control schemas presented by Pollard, Di Carlo and Vasile, and Battin, which can be automated into the algorithm [4], [14], [26]. Expanding the algorithm to fully describe both shape and orientation at a final-time epoch would fully address the reachability of a spacecraft in a bounded-time problem. Additionally, more advanced comparisons could be made that include the implementation of perturbed force models, such as zonal harmonics. Di Carlo and Vasile outline one such example in which they simply calculate the J2 effect over time on an orbit as part of a more comprehensive design strategy. These efforts were not fully pursued by the author due to time constraints. The availability of all supporting code used in this thesis should enable continued advancement with a significantly lower burden of entry for follow-up researchers.

Further research could also develop a linearized estimate for $\Delta\bar{e}$ in conjunction with the maximum variation of $\Delta\bar{a}$ strategy. This effort could also include the development of an automata to generate the scaling function necessary to accurize the $\Delta\bar{i}$ strategy. These efforts would further enable the ability to automate these estimates and to implement them as intermediate elements of a broader and more capable decision support tool.

6.1.2 Optimal Control Focus

For those who wish to continue the advancement of optimal control theory for continuous low-thrust orbits, the use of either the DIDO or the Sparse Nonlinear Optimizer Profile Tool set is strongly recommended. Although DIDO was severely node limited at the time of this research (100s of nodes was extreme), future releases are reported to be capable of computationally efficient runs with orders of magnitude more nodes available. This is likely to have a great impact on the solution space of many multi-revolution, long-duration dynamics problems, and should be seen as a call to reapply these gradient-based methods to the low-thrust mission space. Since Systems Tool Kit v.11, Astrogator has included the ability to use the Sparse Nonlinear Optimizer Profile Tool with a seeded initial guess file. These initial guesses can be provided directly from the included MATLAB scripts that integrate the equations of motion over time. Future researchers can simply run the algorithm for a complex run, export the state and control history from the MATLAB workspace to a .txt file and feed them directly into Systems Tool Kit as an initial guess. A short demonstration of this workflow is included in Appendix C.

6.1.3 Implementation Focus

A major element of this work was to collect, validate, and accurize low-computation cost estimates so that they can be used rapidly in many applications. These algorithms could be implemented on space-flight representative hardware instead of in MATLAB. Similarly, expanded research could focus on development of a concept of operations for incorporating the algorithm into existing Space Domain Awareness or Space Traffic Management procedures.

The increasing importance of continuous low-thrust maneuvers goes hand in hand with the need to create and sustain a new generation of workers who are comfortable thinking about these missions from many different aspects. Gauss's variational equations are typically taught only in follow-on orbital mechanics courses. The need to integrate the equations of motion make continuous-thrust or finite-maneuver problems unsuitable for test-based educational programs that encourage a broad understanding of orbital mechanics. These estimates enable the inclusion of low-thrust elements without the requirement to apply numerical methods. Extended descriptions, explanatory graphics, and practice problems could be developed that would enable the inclusion of these estimates in undergraduate and graduate space courses.

6.2 Hindsight is 20/20

A major element that the author would have changed in the early research is the choice of states for the dynamics model. In retrospect, beginning the research with a thorough understanding of equinoctial or modified equinoctial elements would have greatly expanded the utility of the model and analysis to include situations where integration-halting singularities appear in Gauss's variational equations. Their use would also have encouraged a more general sense of the end point orbit in terms of shape and orientation. Finally, early adoption may have enabled better DIDO optimization findings as both forms limit the state to only one fast element and significantly easier scaling and balancing. One challenge in working with these elements is the nonintuitive relation to the shape and orientation. Another challenge is that no two sources use the same terminology or symbols to represent these equations. Tables 6.1 and 6.2 translate between the major software and texts to help future researchers.

$Equation$	STK	$Battin$	$de\,laco\,Veris$	$Vallado$
$Semi-MajorAxis$	a	a	a	
$e\ \sin(\Omega+\omega)$	h	p_1	p_1	$h_e\ and\ a_g$
$e\ \cos(\Omega+\omega)$	k	p_2	p_2	$k_e\ and\ a_f$
$\tan(\frac{i}{2})\ sin(\Omega)$	p	q_1	q_1	$\mathcal{X}$
$\tan(\frac{i}{2})\ cos(\Omega)$	q	q_2	q_2	Ψ
$M_0+\Omega+\omega$	L	L	ℓ_0	L

Table 6.1. Equinoctial Elements and Sources

$Equation$	STK	$de\,laco\,Veris$
$a(1-e^2)$	p	p
$e\ \cos(\Omega+\omega)$	k	f
$e\ \sin(\Omega+\omega)$	h	g
$\tan(\frac{i}{2})\,cos(\Omega)$	q	h
$\tan(\frac{i}{2})\,sin(\Omega)$	p	k
$\Omega+\omega+\phi_0$	$True\ Longitude$	ℓ_0

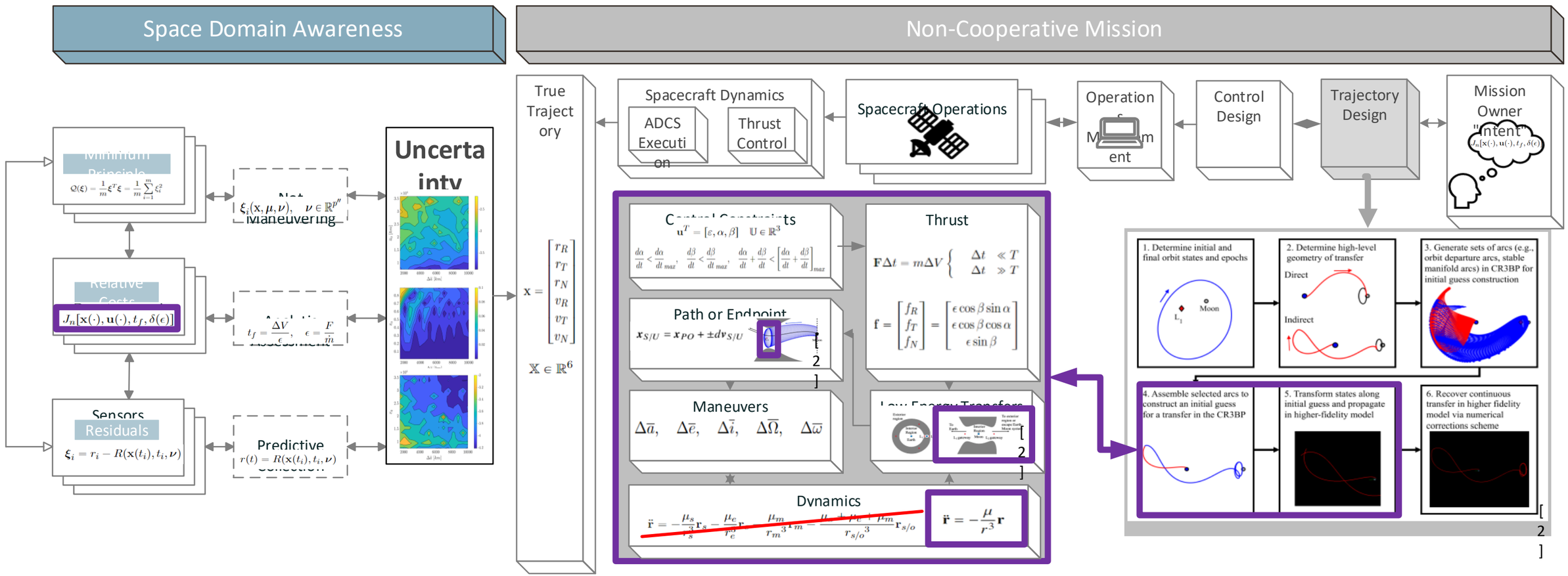
Table 6.2. Modified Equinoctial Elements and Sources

6.3 Generative AI and Its Use in this Research

Generative artificial intelligence tools built on large language models exploded into reality as the author was midway through this research. The author used tools built on these models in nearly every element of this research. Embedded tools scanned every word of this text to help identify (numerous) grammatical, spelling, and formatting errors. Integrated programming tools within and without my programming environments wrote perfect shell logic loops that never ran away. Chatbots never tired of my late night or early morning queries about semi-separable, nonlinear differential equations, or hidden analysis modes of DIDO. Even when these tools were wrong, and all of them were more than once, they succeeded in getting me closer to my objective faster than I could have done on my own. As a demonstration of its absolute inability to tire of the most mundane and dreary work, OpenAI's ChatGPT has provided all of the commentary in the included MATLAB scripts.

APPENDIX A:
Full Size Figure 1.2

Intent Development of Noncooperative, Low-Thrust Maneuvering Spacecraft in Cislunar Space



THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Supporting Elements for Chapter 2

B.1 ODE Solver Evaluation Files

The included code in these sections is extremely fragile and computationally expensive. Only confident MATLAB users should employ this code. It is available upon request from the Dudley Knox Library to enable study replication.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
Supporting Elements for Chapter 3

C.1 Necessary Equations

```
%{
ChatGPT Summary and comments:
This MATLAB code calculates the differential equations for the state
    variables and costates (Lagrange multipliers) of an optimal control
    problem involving spacecraft trajectory optimization. The problem
    considers a satellite orbiting the Moon and aims to optimize its
    trajectory based on the given control inputs.

The code first initializes a series of symbolic variables, including
    state variables (x1, x2, ..., x6), control inputs (u, u1, u2, u3),
    and Lagrange multipliers (lambda1, lambda2, ..., lambda6). The
    variables represent orbital elements and other parameters related to
     the spacecraft trajectory.

Then, it assigns values to some of the variables (E, F, and u3). It
    computes intermediate variables related to the orbital dynamics,
    such as the semilatus rectum (p), specific angular momentum (h),
    radius (r), and argument of latitude (nu). These variables are
    derived from the given state variables and parameters (e.g., mu_moon
    , which is the gravitational parameter of the Moon).

Next, the code calculates the differential equations for the state
    variables (x1_dot, x2_dot, ..., x6_dot) as functions of the control
    inputs (u1, u2, and u3). These equations describe how the state
    variables evolve over time.

Afterward, the code constructs the Hamiltonian (H) for the optimal
    control problem, which is the sum of the cost function (F) and the
    inner product of the Lagrange multipliers (lambda) and the state
    derivatives (x_dot). The Hamiltonian is used to derive the costate (
    Lagrange multiplier) dynamics.
```

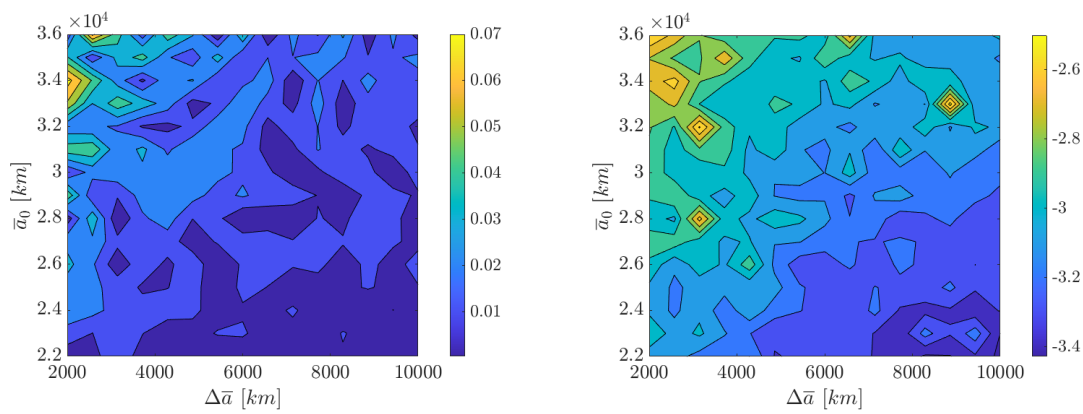
```
Finally, the code computes the derivatives of the Hamiltonian with
    respect to the state variables (lambda1_dot, lambda2_dot, ...,
    lambda6_dot) and the control inputs (diff(H, u1), diff(H, u2), diff(
    H, u3)). These derivatives are used to determine the optimal control
     laws and are essential for solving the optimal control problem.
%}


% Requires symbolic toolkit from MATLAB

     clc, clear all, close all

% Initialize symbolic variables for state variables, control inputs, and
% Lagrange multipliers
syms x xdot...
    x1 x2 x3 x4 x5 x6...
    x1dot x2dot x3dot x4dot x5dot x6dot...
    a e i Omega omega theta...
    adot edot idot Omegadot omegadot thetadot...
    a_0 e_0 i_0 Omega_0 omega_0 theta_0...
    a_f e_f i_f Omega_f omega_f theta_f...
    u...
    u1 u2 u3...
    alpha beta...
    mu_moon epsilon...
    lambda lambda1 lambda2 lambda3 lambda4 lambda5 lambda6...
    lambda_dot lambda1_dot lambda2_dot lambda3_dot lambda4_dot
       lambda5_dot lambda6_dot...
    mu1 mu2...
    t t_0 t_f...
    F f E e...
    H H_bar H_lower EPL EPL_bar...
    J...
    h p r nu

% Assign values to some of the variables
E = -a_f;
F = 0;
u3 = 0;
```

```
% Compute intermediate variables related to the orbital dynamics
p = x1*(1-x2^2); %semilatus rectum
h = sqrt(mu_moon*p); %specific angular momentum
r = h^2/mu_moon/(1 + x2*cos(x6)); %The radius
nu = x5 + x6; %Argument of latitude

% Calculate the differential equations for the state variables
%=====================================================================
x1_dot = ((2*x1^2)/h)*u1*cos(u3)*(x2*sin(x6)*sin(u2)+(p/r)*cos(u2));
x2_dot = u1*cos(u3)/h*((p*sin(x6)*sin(u2)+((p+r)*cos(x6)+r*x2))*cos(u2))
    ;
x3_dot = u1*(r/h)*cos(nu)*sin(u3);
x4_dot = u1*(r/h)*sin(nu)/sin(x3)*sin(u3);
x5_dot = 1/(h*x2)*u1*cos(u3)*(-p*cos(x6)*sin(u2)+(p+r)*sin(x6)*cos(u2))
    -(r*sin(nu)*cos(x2))/(h*sin(x2))*u1*sin(u3);
x6_dot = (h/r^2)+u1/(x2*h)*cos(u3)*(p*cos(x6)*sin(u2)-(p+r)*sin(x6)*cos(
    u2));

% Construct the Hamiltonian for the optimal control problem
x_dot = [x1_dot x2_dot  x6_dot]';
lambda = [lambda1 lambda2  lambda6]';
H = F + dot(lambda',x_dot)

% Compute the derivatives of the Hamiltonian with respect to the state
    variables
lambda1_dot = -diff(H, x1)
lambda2_dot = -diff(H, x2)
lambda3_dot = -diff(H, x3)
lambda4_dot = -diff(H, x4)
lambda5_dot = -diff(H, x5)
lambda6_dot = -diff(H, x6)

% Compute the derivatives of the Hamiltonian with respect to the control
     inputs
diff(H, u1)
diff(H,u2)
diff(H, u3)
```

THIS PAGE INTENTIONALLY LEFT BLANK

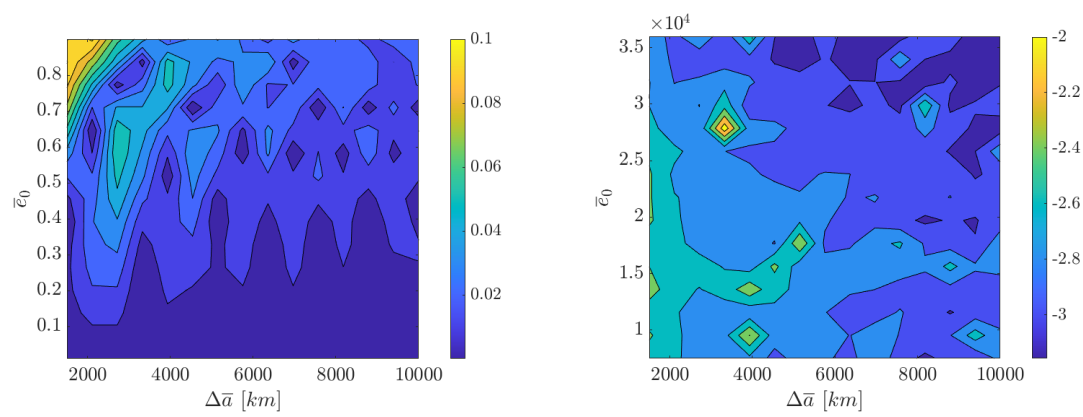
APPENDIX D:
Supporting Elements for Chapter 4

D.1 Accuracy and Cost Plots for $\Delta\overline{a}_{MAX}$ Estimates



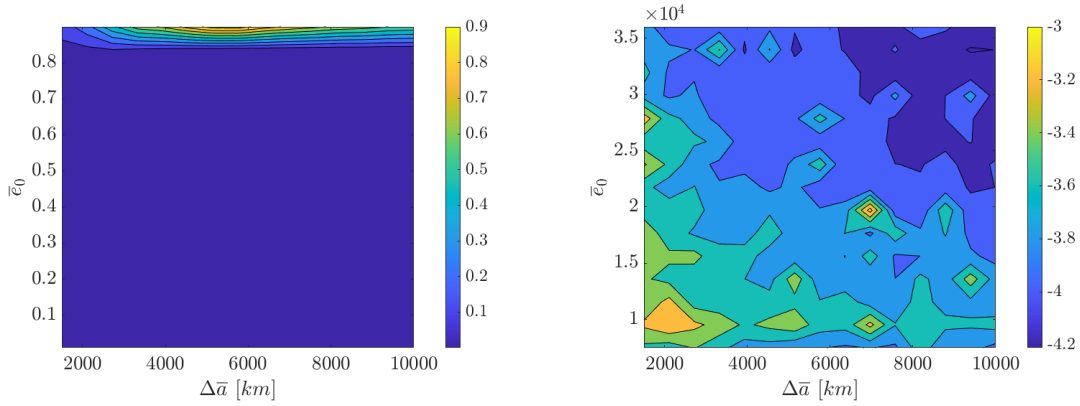
(a) $\delta\Delta V$ by $\overline{a}_0$ (b) δ Computation variance by $\overline{a}_0$

Figure D.1. Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates



(a) $\delta\Delta V$ by $\overline{e}_0$ (b) δ Computation Variance by $\overline{a}_0$

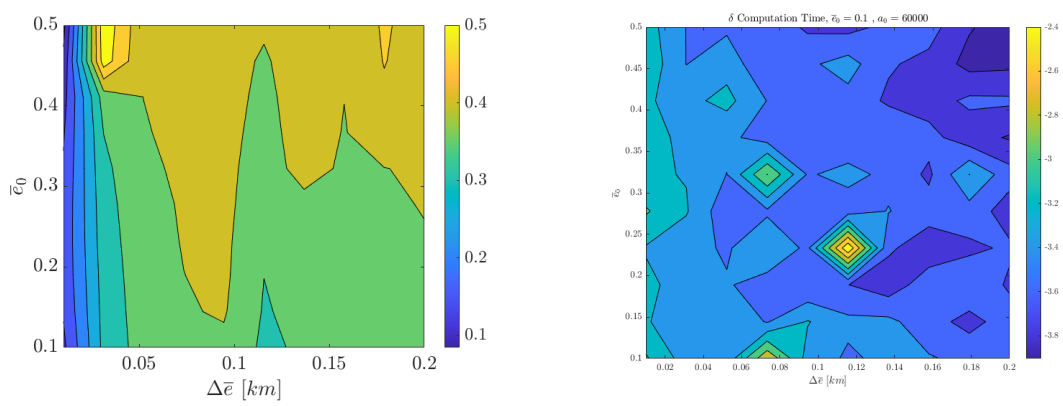
Figure D.2. Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates



(a) $\delta\Delta V$ by $\overline{e}_0$ (b) δ Computation Variance by $\overline{a}_0$

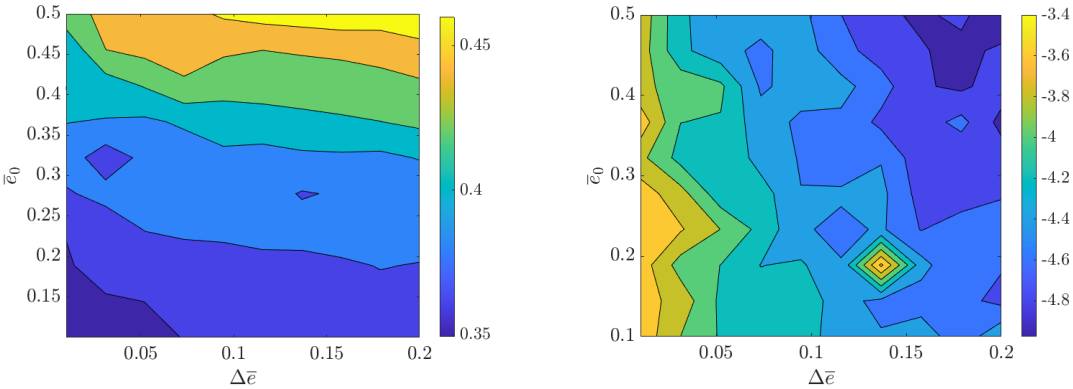
Figure D.3. Sample Accuracy and Computational Costs of $\Delta\overline{a}$ Estimates

D.2 Accuracy and Cost Plots for $\Delta\overline{e}$ Estimates



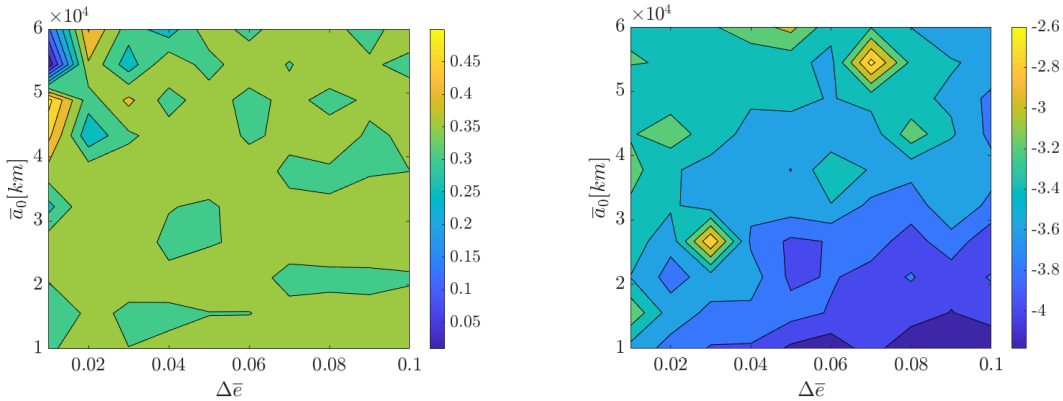
(a) $\delta\Delta V$ by $\overline{e}_0$ (b) δ Computation Variance by $\overline{e}_0$

Figure D.4. Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates



(a) $\delta\Delta V$ by $\overline{e}_0$ (b) δ Computation Variance by $\overline{a}_0$

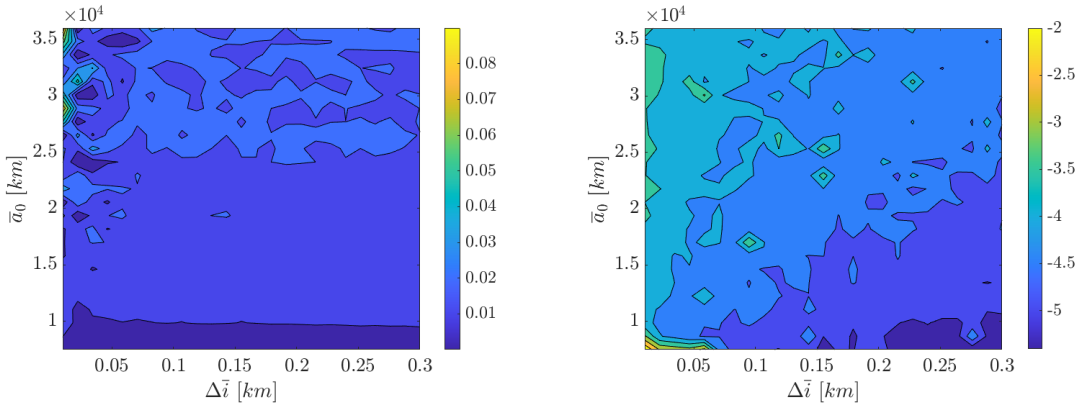
Figure D.5. Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates



(a) $\delta\Delta V$ by $\overline{a}_0$ (b) δ Computation Variance by $\overline{a}_0$

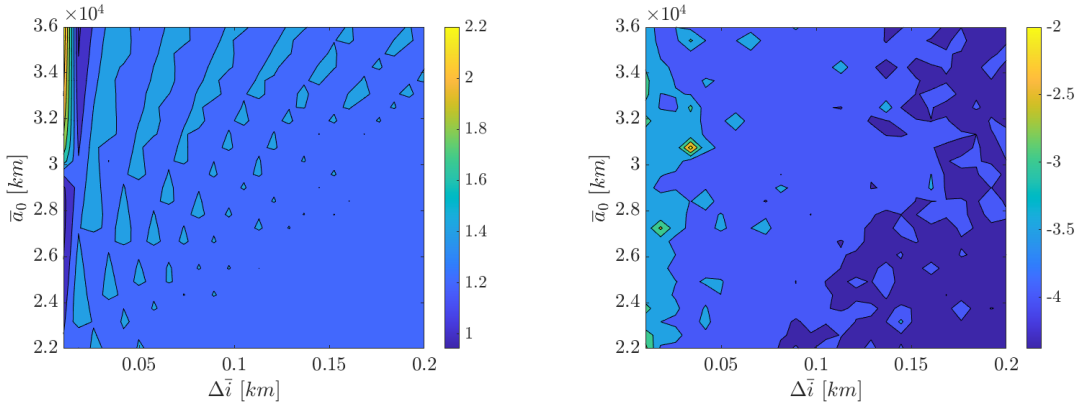
Figure D.6. Sample Accuracy and Computational Costs of $\Delta\overline{e}$ Estimates

D.3 Accuracy and Cost Plots for $\Delta\bar{i}$ Estimates



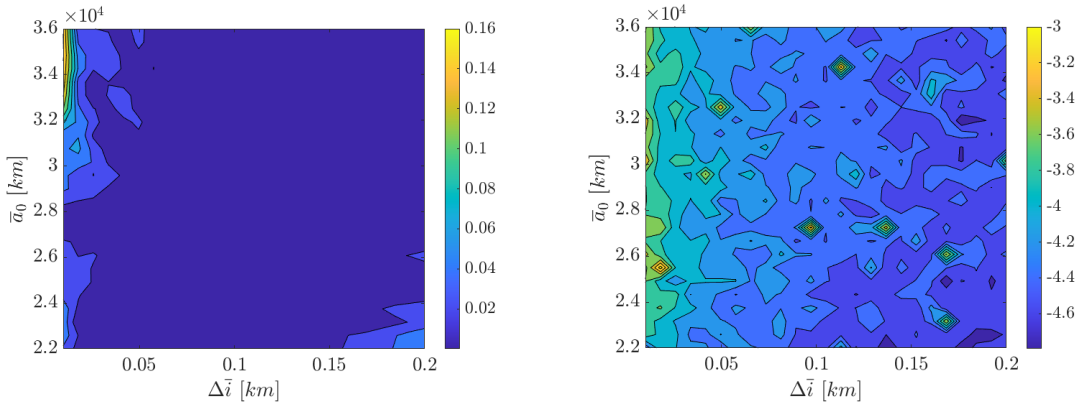
(a) $\delta\Delta V$ by $\bar{a}_0$ (b) δ Computation Eariance by $\bar{a}_0$

Figure D.7. Sample Accuracy and Computational Costs of $\Delta\bar{i}$ Estimates



(a) $\delta\Delta V$ by $\bar{a}_0$ (b) δ Computation Variance by $\bar{a}_0$

Figure D.8. Sample Accuracy and Computational Costs of $\Delta\bar{i}$ Estimates



(a) $\delta\Delta V$ by $\overline{a}_0$

(b) δ Computation Variance by $\overline{a}_0$

Figure D.9. Sample Accuracy and Computational Costs of $\Delta\bar{i}$ Estimates

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: MATLAB Code to Replicate Findings

E.0.1 General Functions

```
    function [G,m_earth,m_moon,m_sun,...
        mu_earth,mu_moon,a_earth,a_moon,r_moon,r_earth,moon_SOI]...
          = Cislunar_constants

%{
This script, "Cislunar_constants.m", defines and assigns numerical
values to various physical and astronomical constants used in cislunar
orbital mechanics calculations.
 These constants include
 gravitational constant (G), mass of Earth (m_earth), mass of
Moon (m_moon), mass of Sun (m_sun),
 gravitational parameter of Earth (mu_earth), gravitational
parameter of Moon (mu_moon), semi-major axis of Earth's orbit (a_earth),
 semi-major axis of Moon's orbit (a_moon), radius of Moon
(r_moon), radius of Earth (r_earth) and the radius of the sphere
of influence of the Moon (moon_SOI).
 These constants are all in standard S.I. units (km, kg, s)
%}

G=6.673*10^-20; %km^3/(kg s^2)
m_earth=5.9722E24; %kg
m_moon=0.07346E24; %kg
m_sun=1988500E24; %kg
mu_earth=0.39860E6; %km^3/s^2
mu_moon=0.00490E6; %km^3/s^2
a_earth=149.598E6; %km
a_moon=0.3844E6; %km
r_moon= 1738.1; %km
r_earth=6378; %km
moon_SOI=a_moon*(m_moon/m_earth)^(2/5); %km
end
```

```
    % ----------------------------------------------------------------------
% FUNCTION: LatexTitles
%ChatGPT Description
% DESCRIPTION:
% This function sets the default interpreter for all graphics objects to
% LaTeX.
%
% INPUTS:
% None
%
% OUTPUT:
% None
%----------------------------------------------------------------------
function LatexTitles

list_factory = fieldnames(get(groot,'factory'));
index_interpreter = find(contains(list_factory,'Interpreter'));
for i = 1:length(index_interpreter)
    default_name = strrep(list_factory{index_interpreter(i)},'factory','default');
    set(groot, default_name,'latex');
end
end
```

E.0.2 Translating Between State Representations

Source: [16]

```
    % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  function coe = coe_from_sv(R,V,mu)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Developed and made available by Vallado
% This function computes the classical orbital elements (coe)
% from the state vector (R,V) for a given central body (mu)
% using Algorithm 4.1.
%
```

```
  mu   - gravitational parameter (km^3/s^2)
  R    - position vector in the geocentric equatorial frame (km)
  V    - velocity vector in the geocentric equatorial frame (km)
  r, v - the magnitudes of R and V
  vr   - radial velocity component (km/s)
  H    - the angular momentum vector (km^2/s)
  h    - the magnitude of H (km^2/s)
  incl - inclination of the orbit (rad)
  N    - the node line vector (km^2/s)
  n    - the magnitude of N
  cp   - cross product of N and R
  RA   - right ascension of the ascending node (rad)
  E    - eccentricity vector
  e    - eccentricity (magnitude of E)
  eps  - a small number below which the eccentricity is considered
         to be zero
  w    - argument of perigee (rad)
  TA   - true anomaly (rad)
  a    - semimajor axis (km)
  pi   - 3.1415926...
  coe  - vector of orbital elements [h e RA incl w TA a]

  User M-functions required: None
%}
% ---------------------------------------------

eps = 1.e-10;

r    = norm(R);
v    = norm(V);

vr   = dot(R,V)/r;

H    = cross(R,V);
h    = norm(H);

%...Equation 4.7:
incl = acos(H(3)/h);

%...Equation 4.8:
```

```
N    = cross([0 0 1],H);
n    = norm(N);

%...Equation 4.9:
if n ~= 0
    RA = acos(N(1)/n);
    if N(2) < 0
        RA = 2*pi - RA;
    end
else
    RA = 0;
end

%...Equation 4.10:
E = 1/mu*((v^2 - mu/r)*R - r*vr*V);
e = norm(E);

%...Equation 4.12 (incorporating the case e = 0):
if n ~= 0
    if e > eps
        w = acos(dot(N,E)/n/e);
        if E(3) < 0
            w = 2*pi - w;
        end
    else
        w = 0;
    end
else
    w = 0;
end

%...Equation 4.13a (incorporating the case e = 0):
if e > eps
    TA = acos(dot(E,R)/e/r);
    if vr < 0
        TA = 2*pi - TA;
    end
else
    cp = cross(N,R);
    if cp(3) >= 0
```

```
        TA = acos(dot(N,R)/n/r);
    else
        TA = 2*pi - acos(dot(N,R)/n/r);
    end
end

%...Equation 4.62 (a < 0 for a hyperbola):
a = h^2/mu/(1 - e^2);

coe = [h e RA incl w TA a];

  end  %coe_from_sv
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
    % ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  function [r, v] = sv_from_coe(coe,mu)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Developed and made available by Vallado
  This function computes the state vector (r,v) from the
  classical orbital elements (coe).

  mu   - gravitational parameter (km^3;s^2)
  coe  - orbital elements [h e RA incl w TA]
         where
             h    = angular momentum (km^2/s)
             e    = eccentricity
             RA   = right ascension of the ascending node (rad)
             incl = inclination of the orbit (rad)
             w    = argument of perigee (rad)
             TA   = true anomaly (rad)
  R3_w - Rotation matrix about the z-axis through the angle w
  R1_i - Rotation matrix about the x-axis through the angle i
  R3_W - Rotation matrix about the z-axis through the angle RA
  Q_pX - Matrix of the transformation from perifocal to geocentric
         equatorial frame
  rp   - position vector in the perifocal frame (km)
  vp   - velocity vector in the perifocal frame (km/s)
  r    - position vector in the geocentric equatorial frame (km)
```

```
  v    - velocity vector in the geocentric equatorial frame (km/s)

  User M-functions required: none
%}
% ----------------------------------------------

h    = coe(1);
e    = coe(2);
RA   = coe(3);
incl = coe(4);
w    = coe(5);
TA   = coe(6);

%...Equations 4.45 and 4.46 (rp and vp are column vectors):
rp = (h^2/mu) * (1/(1 + e*cos(TA))) * (cos(TA)*[1;0;0] + sin(TA)
    *[0;1;0]);
vp = (mu/h) * (-sin(TA)*[1;0;0] + (e + cos(TA))*[0;1;0]);

%...Equation 4.34:
R3_W = [ cos(RA)  sin(RA)  0
        -sin(RA)  cos(RA)  0
            0        0     1];

%...Equation 4.32:
R1_i = [1       0          0
        0   cos(incl)  sin(incl)
        0  -sin(incl)  cos(incl)];

%...Equation 4.34:
R3_w = [ cos(w)  sin(w)  0
        -sin(w)  cos(w)  0
           0       0     1];

%...Equation 4.49:
Q_pX = (R3_w*R1_i*R3_W)';

%...Equations 4.51 (r and v are column vectors):
r = Q_pX*rp;
v = Q_pX*vp;

```

```
%...Convert r and v into row vectors:
r = r';
v = v';

end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

E.1 Analytical Estimates

E.1.1 $\Delta\overline{a}$

```
%Dan Gregorio 22NOV2022 Carmel, CA
%{
%ChatGPT Comments
% This function calculates the delta-V required for a continuous
low-thrust maneuver from an elliptical orbit with semi-major axis
a_0, eccentricity e_0 to another
 elliptical orbit with semi-major axis a_f and eccentricity e_f, using the
 gravitational parameter mu of the central body.

% The calculation is based on the use of elliptic integrals, specifically the
% complete elliptic integral of the first kind (F_lc) and the complete elliptic
% integral of the second kind (E_lc) to determine the ratio of the two integrals.


-Implements DiCarlo and Vasile's Analytic  calculation for Variation of the
Semi-major Axis with Tangential Thrust case 3.1 p. 5/38
-Takes as input:
    -an initial semimajor axis and eccentricity, a_0 and e_0
    -a target final semimajor axis and eccentricity, a_f and e_f
    -gravitational parameter of central body, mu
-Note, relies on MATLAB's complete elliptic integral of the first kind
(ellipke)
```

```
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions
for low-thrust orbit transfers,    Celest Mech Dyn Astr,
vol. 133, no. 7, p. 33, Jul. 2021,
doi: 10.1007/s10569-021-10033-9.

%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function DeltaV_An = AnalyticDV_max_a(a_0, e_0, a_f, e_f, mu)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%a and mu in km

e_bar=.5*(e_0+e_f);
 , and  [1]
m=(4*e_bar)/(1+e_bar)^2;
[F_lc,E_lc]=ellipke(m);%calculate elliptic integrals of first and second kind
 [1]
f_e=(2/(1-e_bar))*E_lc+(2/(1+e_bar))*F_lc;
  [1]
DeltaV_An=(2*pi*abs(sqrt(mu/a_0)-sqrt(mu/a_f)))/((1-e_bar^2)*f_e);

end
```

E.1.2 $\Delta\overline{e}$

```
%-----------------------------------------------------------------------
%Dan Gregorio 22NOV2022 Carmel, CA
% FUNCTION: AnalyticDV_delta_e_only
%ChatGPT Comments
% DESCRIPTION:
% This function calculates the deltaV required to change the eccentricity of
% an orbit from e_0 to e_f, assuming a constant semi-major axis a, using the
% analytic solution.
%
% INPUTS:
% a: Semi-major axis of the orbit [km]
```

```
% e_0: Initial eccentricity
% e_f: Final eccentricity
% mu: Gravitational parameter of the central body [km^3/s^2]
%
% OUTPUT:
% DeltaV_An: DeltaV required to change the eccentricity of the orbit [km
    /s]
%-----------------------------------------------------------------------
function DeltaV_An = AnalyticDV_delta_e_only(a, e_0, e_f, mu)

DeltaV_An = 2/3*sqrt(mu/a)*abs(asin(e_f-e_0));

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.1.3 $\Delta\bar{i}$

```
%-----------------------------------------------------------------------
% FUNCTION: AnalyticDV_del_max_i
%
% DESCRIPTION:
% This function calculates the deltaV required to change the inclination
     of
% an orbit from i_0 to i_f, assuming a circular orbit at a_0 altitude,
    using
% the analytic solution. Estimates will exhibit increasing error as
% eccentricity increases.
%
% INPUTS:
% a_0: Altitude of the circular orbit [km]
% i_0: Initial inclination [deg]
% i_f: Final inclination [deg]
% mu: Gravitational parameter of the central body [km^3/s^2]
%
% OUTPUT:
% DeltaV_An: DeltaV required to change the inclination of the orbit [km/
    s]
%-----------------------------------------------------------------------
```

```

function DeltaV_An = AnalyticDV_del_max_i(a_0, i_0, i_f, mu)
function DeltaV_An = AnalyticDV_del_max_i(a_0, i_0, i_f, mu)

delta_i = abs(abs(i_f)-abs(i_0));

DeltaV_An = (pi/2)*sqrt(mu/a_0)*delta_i;
end
```

E.1.4 $\Delta\overline{\Omega}$

```
%-----------------------------------------------------------------------
% FUNCTION: AnalyticDV_del_max_Omega
%ChatGPT description and comments
% DESCRIPTION:
% This function calculates the deltaV required to change the longitude
    of
% ascending node of an orbit from Omega_0 to Omega_f, assuming a
    circular
% orbit at a_0 altitude and an initial inclination of i_0, using the
% analytic solution.
%
% INPUTS:
% Omega_0: Initial longitude of ascending node [deg]
% i_0: Initial inclination [deg]
% a_0: Altitude of the circular orbit [km]
% Omega_f: Final longitude of ascending node [deg]
% mu: Gravitational parameter of the central body [km^3/s^2]
%
% OUTPUT:
% DeltaV_An: DeltaV required to change the longitude of ascending node
% of the orbit [km/s]
%-----------------------------------------------------------------------
function DeltaV_An = AnalyticDV_del_max_Omega(Omega_0,i_0,a_0, Omega_f,
    mu)

delta_Omega = abs(abs(Omega_f)-abs(Omega_0));

```

```
DeltaV_An = (pi/2)*sqrt(mu/a_0)*delta_Omega*sin(i_0);
end
```

E.1.5 $\Delta\overline{\omega}$

```
%-----------------------------------------------------------------------
% FUNCTION: AnalyticDV_delta_w_only
%ChatGPT Description
% DESCRIPTION:
% This function calculates the deltaV required to change the argument of
% periapsis of an orbit from w_0 to w_f, assuming a constant semi-major
% axis a and eccentricity e, using the analytic solution.
%
% INPUTS:
% a: Semi-major axis of the orbit [km]
% e: Eccentricity of the orbit
% w_0: Initial argument of periapsis [deg]
% w_f: Final argument of periapsis [deg]
% mu: Gravitational parameter of the central body [km^3/s^2]
%
% OUTPUT:
% DeltaV_An: DeltaV required to change the argument of periapsis of the
    orbit [km/s]
%-----------------------------------------------------------------------
%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function DeltaV_An = AnalyticDV_delta_w_only(a,e, w_0, w_f, mu)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

p=a*(1-e^2);
DeltaV_An = sqrt(mu/p)*abs(w_f-w_0);

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.2 Integration Functions

E.2.1 Dynamics

```
%Dan Gregorio 22NOV2022 Carmel, CA
function dGdt = VV_dynamics_wrapper(t,G,T,control_test,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss's equations designed to take a control history (alpha)
%(Equation 2 in Di Carlo and Vasile 2021).
%Unpack control(s)
alpha_in=control_test(:,1)
beta_in=control_test(:,2)


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%interpolation control(s)
alpha = interp1(T,alpha_in,t,'spline');%also with 'linear'
beta=0;%interp1(T,beta_in,t,'spline'); %because beta is 0 for the
%test hard coded to save processing and memory

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
nu = w + theta; %Argument of latitude


%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon.*cos(beta).*(e.*sin(theta).*sin(alpha)+...
(p/r).*cos(alpha));

edot = epsilon.*cos(beta)./h*((p*sin(theta).*sin(alpha)+...
((p+r)*cos(theta)+r*e)).*cos(alpha));
```

```
idot = epsilon*(r/h)*cos(nu).*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i).*sin(beta);

wdot = 1/(h*e)*epsilon.*cos(beta)*(-p*cos(theta).*sin(alpha)+...
(p+r).*sin(theta).*cos(alpha))...
-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon.*sin(beta);

thetadot = (h/r^2)+epsilon/(e*h).*cos(beta)*...
(p*cos(theta).*sin(alpha)-(p+r)*sin(theta).*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)

% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
```

E.2.2 Integration of $\Delta\overline{a}$ Strategy

```
%Dan Gregorio 15JAN2023 Carmel, CA
%{
-Integrates Gauss' equations as developed by Battin and DiCarlo and
    vasile
Implemented by Hudson and Gregorio

-Takes as input:
    -a vector of classical orbital elements [a e i Omega omega theta],
        COE0
    -a target final semimajor axis, af
    -value of the magnitude of the low-thrust force, epsilon
    -gravitational parameter of central body, mu
```

```
    -radius of central body, RCB
-All required sub-functions included as nested functions(GaussEquations_3_1
[ODEs] and delta_a_event [event fn]). N.B. that if separating these
subfuncitons global variables must be set to feed variable af trigger to
events function. Not for the faint of heart.
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust orbit transfers,
% Celest Mech Dyn Astr, vol. 133, no. 7, p. 33, Jul. 2021, doi: 10.1007/s10569-021-10033-9.



%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [T,Y,alpha,beta]=Gauss_Maneuver_Exfil_max_a_15s(COE0,af,epsilon,mu,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%Unpack Constants
constants.af=af; %Final (objective) Semimajor axis (km)
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)

%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset(...
    'Stats','on',...
    'events',@delta_a_event); %uses termination criteria event
```

```

%Timespan
t_span = [0 ToF_Max];

%Builds initial conditions vector for ODE Solver
G0= [a_0;
     e_0;
     i_0;
     RA_0;
     w_0;
     theta_0];

%Propagate Gauss' Equations
[T Y] = ode15s(@(T,Y) GaussEquations_3_1(T,Y,constants), t_span, G0,
    options);
theta_history=Y(:,6);e_history=Y(:,2);
alpha=atan2(e_history.*sin(theta_history),1+e_history.*cos(theta_history
    ));
beta=zeros(length(alpha),1);

%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_3_1(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
```

```
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
nu = w + theta; %Argument of latitude

%Thrust angle calculations
alpha=atan2(e*sin(theta),1+e*cos(theta));
beta=0;

%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+(p/r)*cos(
    alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));

idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*(p*cos(theta)*sin(alpha)-...
(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
```

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function


%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    function [value, isterminal, direction] = delta_a_event(T,Y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low to
high
%}
value       = (Y(1) > af);
isterminal = 1;   % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.2.3 Integration of $\Delta\overline{e}$ Strategy

```
%Dan Gregorio 15JAN2023 Carmel, CA
%{
-Integrates Gauss' equations as developed by Battin and DiCarlo and vasile
Implemented by Hudson and Gregorio

-Takes as input:
    -a vector of classical orbital elements [a e i Omega omega theta], COE0
    -a target final semimajor axis, af
    -value of the magnitude of the low-thrust force, epsilon
    -gravitational parameter of central body, mu
    -radius of central body, RCB
-All required sub-functions included as nested functions(GaussEquations_3_1
[ODEs] and delta_a_event [event fn]). N.B. that if separating these
```

```
subfuncitons global variables must be set to feed variable af trigger to
events function. Not for the faint of heart.
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust
    orbit transfers,    Celest Mech Dyn Astr, vol. 133, no. 7, p. 33,
    Jul. 2021, doi: 10.1007/s10569-021-10033-9.

%N.B. to execute for a REDUCTION of eccentricity between e_0 and e_f,
    alpha
%= theta + pi

%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [T,Y,alpha,beta]=Gauss_Maneuver_Exfil_max_e_15s(COE0,e_f,
    epsilon,mu,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%Unpack Constants
%constants.i_f = i_f; %Final (objective)  inclination
constants.e_f = e_f; %Final (objective) eccentricity
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)

%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset(...
    'Stats','on',...
    'events',@delta_a_event); %uses termination criteria event

%Timespan
```

```
t_span = [0 ToF_Max];

%Builds initial conditions vector for ODE Solver
G0= [a_0;
      e_0;
      i_0;
      RA_0;
      w_0;
      theta_0];

%Propagate Gauss' Equations
[T Y] = ode15s(@(T,Y) GaussEquations_3_1(T,Y,constants), t_span, G0,
    options);
theta_history=Y(:,6);e_history=Y(:,2);
alpha=theta_history; %Change to alpha = thata_history+pi for reduction
    of eccentricity
beta=zeros(length(alpha),1);

%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_3_1(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
```

```
nu = w + theta; %Argument of latitude
E = acos((e+cos(nu))/(1+e*cos(nu))); %Eccentric Anom


%Thrust angle calculations

alpha=theta; %Change to alpha = thata_history+pi for reduction of
    eccentricity
beta=0;

%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+(p/r)*cos(
    alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));

idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*...
(p*cos(theta)*sin(alpha)-(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];
```

```

end %GaussEquations
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function


%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    function [value, isterminal, direction] = delta_a_event(T,Y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low to
high
%}
value       = (Y(2) > e_f);
isterminal = 1;    % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.2.4 Integration of $\Delta\bar{i}$ Strategy

```
%Dan Gregorio 15JAN2023 Carmel, CA
%{
-Integrates Gauss' equations as developed by Battin and DiCarlo and vasile
Implemented by Hudson and Gregorio

-Takes as input:
    -a vector of classical orbital elements [a e i Omega omega theta], COE0
    -a target final semimajor axis, af
    -value of the magnitude of the low-thrust force, epsilon
    -gravitational parameter of central body, mu
    -radius of central body, RCB
-All required sub-functions included as nested functions(
```

```
    GaussEquations_3_1
[ODEs] and delta_a_event [event fn]). N.B. that if separating these
subfuncitons global variables must be set to feed variable af trigger to
events function. Not for the faint of heart.
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust
    orbit transfers,    Celest Mech Dyn Astr, vol. 133, no. 7, p. 33,
    Jul. 2021, doi: 10.1007/s10569-021-10033-9.



%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [T,Y,alpha,beta]=Gauss_Maneuver_Exfil_max_i_15s(COE0,i_f,
    epsilon,mu,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%Unpack Constants
constants.i_f=i_f; %Final (objective) Semimajor axis (km)
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)

%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset(...
    'Stats','on',...
    'events',@delta_i_event); %uses termination criteria event

%Timespan
t_span = [0 ToF_Max];
```

```
%Builds initial conditions vector for ODE Solver
G0= [a_0;
     e_0;
     i_0;
     RA_0;
     w_0;
     theta_0];

%Propagate Gauss' Equations
[T Y] = ode15s(@(T,Y) GaussEquations_3_1(T,Y,constants), t_span, G0, options);
theta_history=Y(:,6);w_history=Y(:,5);
beta=sign(cos(w_history+theta_history))*sign(i_f-i_0)*(pi/2);
alpha=zeros(length(beta),1);



%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_3_1(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
```

```
nu = w + theta; %Argument of latitude

%Thrust angle calculations
alpha=0;
beta=sign(cos(nu))*sign(i_f-i_0)*(pi/2);

%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+...
(p/r)*cos(alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));

idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*(p*cos(theta)*sin(alpha)-...
(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

```

```matlab

%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    function [value, isterminal, direction] = delta_i_event(T,Y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low to
high
%}
value      = (Y(3) > i_f);
isterminal = 1;    % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.2.5 Integration of $\Delta\overline{\Omega}$ Strategy

```matlab
%Dan Gregorio 15JAN2023 Carmel, CA
%{
-Integrates Gauss' equations as developed by Battin and DiCarlo and vasile
Implemented by Hudson and Gregorio

-Takes as input:
    -a vector of classical orbital elements [a e i Omega omega theta], COE0
    -a target final semimajor axis, af
    -value of the magnitude of the low-thrust force, epsilon
    -gravitational parameter of central body, mu
    -radius of central body, RCB
-All required sub-functions included as nested functions( GaussEquations_3_1
[ODEs] and delta_a_event [event fn]). N.B. that if separating these
subfuncitons global variables must be set to feed variable af trigger to
events function. Not for the faint of heart.
```

```
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust
    orbit transfers,    Celest Mech Dyn Astr, vol. 133, no. 7, p. 33,
    Jul. 2021, doi: 10.1007/s10569-021-10033-9.



%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [T,Y,alpha,beta]=Gauss_Maneuver_Exfil_max_RA_15s(COE0,RA_f,
    epsilon,mu,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%pack Constants
constants.RA_f=RA_f; %Final (objective) Semimajor axis (km)
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)

%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset(...
    'Stats','on',...
    'events',@delta_a_event); %uses termination criteria event

%Timespan
t_span = [0 ToF_Max];

%Builds initial conditions vector for ODE Solver
G0= [a_0;
     e_0;
```

```
      i_0;
      RA_0;
      w_0;
      theta_0];

%Propagate Gauss' Equations
[T Y] = ode15s(@(T,Y) GaussEquations_3_1(T,Y,constants), t_span, G0, options);
theta_history=Y(:,6);w_history=Y(:,5);nu_history = w_history + theta_history; %Argument of latitude
beta=sign(RA_f-RA_0)*sign((sin(nu_history)))*(pi/2);
alpha=zeros(length(beta),1);



%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_3_1(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
nu = w + theta; %Argument of latitude

%Thrust angle calculations
```

```
alpha=0;
beta=sign(RA_f-RA_0)*sign((sin(nu)))*(pi/2);

%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+...
(p/r)*cos(alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));

idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*(p*cos(theta)*sin(alpha)-...
(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function


%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    function [value, isterminal, direction] = delta_a_event(T,Y)
```

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low to
high
%}
value      = (Y(4) > RA_f);
isterminal = 1;   % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.2.6 Integration of $\Delta\overline{\omega}$ Strategy

```
%Dan Gregorio 15JAN2023 Carmel, CA
%{
-Integrates Gauss' equations as developed by Battin and DiCarlo and vasile
Implemented by Hudson and Gregorio

-Takes as input:
    -a vector of classical orbital elements [a e i Omega omega theta], COE0
    -a target final semimajor axis, af
    -value of the magnitude of the low-thrust force, epsilon
    -gravitational parameter of central body, mu
    -radius of central body, RCB
-All required sub-functions included as nested functions(GaussEquations_3_1
[ODEs] and delta_a_event [event fn]). N.B. that if separating these
subfuncitons global variables must be set to feed variable af trigger to
events function. Not for the faint of heart.
%}
%[1] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust orbit transfers,    Celest Mech Dyn Astr, vol. 133, no. 7, p. 33,
```

```
    Jul. 2021, doi: 10.1007/s10569-021-10033-9.



%~~MAIN FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [T,Y,alpha,beta]=Gauss_Maneuver_Exfil_w_15s(COE0,w_f,epsilon,mu
    ,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%Unpack Constants
constants.w_f=w_f; %Final (objective) Semimajor axis (km)
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)
if w_f>w_0
    constants.alpha = pi/2;
else %assumes wf<w0
    constants.alpha = 3*pi/2;
end
%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset(...
    'Stats','on',...
    'events',@delta_a_event,...
    'reltol',1E-12,'abstol',1e-12); %uses termination criteria event

%Timespan
t_span = [0 ToF_Max];

%Builds initial conditions vector for ODE Solver
```

```
G0= [a_0;
     e_0;
     i_0;
     RA_0;
     w_0;
     theta_0];

%Propagate Gauss' Equations
[T Y] = ode45(@(T,Y) GaussEquations_3_1(T,Y,constants), t_span, G0, options);
theta_history=Y(:,6);w_history=Y(:,5);
alpha=constants.alpha*ones(length(T),1);
beta=zeros(length(T),1);



%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_3_1(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
nu = w + theta; %Argument of latitude

```

```
%Thrust angle calculations
alpha=constants.alpha;
beta=0;

%Differential Equations
%=====================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+...
(p/r)*cos(alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));

idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*(p*cos(theta)*sin(alpha)-...
(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function


%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
    function [value, isterminal, direction] = delta_a_event(T,Y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low to
high
%}
value       = (Y(5) > w_f);
isterminal = 1;   % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.3 Analysis Scripts

E.3.1 Single Maneuver Analysis

```
    function [TOF,a_f,e_f,DV_Gauss]=...
    Gauss_PointManeuver_Analysis_Control_Plots...
    (T,Y,alpha,beta,mu,epsilon)
LatexTitles
a = Y(:,1);
e = Y(:,2);
i = Y(:,3);
RA = Y(:,4);
w = Y(:,5);
theta = Y(:,6);
h = sqrt(mu.*a.*(1-e.^2));


[G,m_earth,m_moon,m_sun,...
          mu_earth,mu_moon,...
          a_earth,a_moon,...
          r_moon,r_earth,...
          moon_SOI] = Cislunar_constants;
```

```
r = zeros(length(Y),3);
v = zeros(length(Y),3);
n = 0;
while n<length(Y)
    n=n+1;
    coe = [h(n) e(n) RA(n) i(n) w(n) theta(n)]; %coe  - orbital elements [h e RA incl w TA]
    [r(n,:), v(n,:)] = sv_from_coe(coe,mu);
end

figure

%plots of numeric assessments to sanity check

tiledlayout(4,2)
 nexttile
 plot3(r(:,1),r(:,2),r(:,3), 'Color' , '#A24646');hold on;
 rcb=r_moon;
[x,y,z] = sphere;
 mun = surf(x*rcb, y*rcb, z*rcb);
 % Load Moon Image
        topoMoon = imread('moon.jpg');
        % Set it on MOON
        set(mun,'facecolor','texture',...
                 'cdata',im2double(topoMoon),...
                 'edgecolor','none');hold off
 title('Position [km]')
 %moon texture from
%Ashish Meshram (Meet) (2023). 3D Visualization of SUN, EARTH and Moon (https://github.com/meetashish/sun-earth-moon), GitHub. Retrieved February 23, 2023.


 nexttile
 plot(T,alpha,T,beta)
 legend('$\alpha$ [rad]','$\beta$ [rad]')
 title('Control History [rad]')

nexttile
```

```
 plot(T,Y(:,1))%plot semimajor axis in km over time in sec
 title('Semi-major Axis [km]')

nexttile
 plot(T,Y(:,2))
 title('Eccentricity')

nexttile
 plot(T,Y(:,3))
 title('Inclination [rad]')

nexttile
 plot(T,Y(:,4))
 title('$\Omega$ [rad]')

 nexttile
 plot(T,Y(:,5))
 title('$\omega$ [rad]')

nexttile
 plot(T,Y(:,6))
 title('$\theta$ [rad]')

 %sanity check
TOF=T(end);
a_f=Y(end,1);
e_f=Y(end,2);
DV_Gauss=epsilon*TOF;
end
```

E.3.2 Variational measurement

```
    %accuracy plot development
function [Variation_DeltaV,Variation_Comptime]=...
Gauss_max_a_DeltaV_deviation_15s_analytic...
(COE0,af,epsilon,mu,days)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```

```
%Unpack inputs and match to ODEs
a_0 = COE0(1);
e_0 = COE0(2);
i_0 = COE0(3);
RA_0 = COE0(4);
w_0 = COE0(5);
theta_0 = COE0(6);

%Unpack Constants
constants.af=af; %Final (objective) Semimajor axis (km)
constants.thrust=epsilon; %acceleration
constants.mu=mu; %Earth Gravitational parameter (km^3/s^2)

%Arbitrary ToF maximum - Change as needed set to 22 days for now
ToF_Max=86400*days;

%ODE Solver options
options = odeset('events',@delta_a_event); %uses termination criteria
    event

%Timespan
t_span = [0 ToF_Max];

%Builds initial conditions vector for ODE Solver
G0= [a_0;
     e_0;
     i_0;
     RA_0;
     w_0;
     theta_0];

%Propagate Gauss' Equations
tic
[T G] = ode15s(@(T,Y) GaussEquations_max_a(T,Y,constants), t_span, G0,
    options);
numcomptime=toc;
DeltaV_num=T(end)*constants.thrust;

tic
DeltaV_An = AnalyticDV_max_a(a_0, e_0, af, G(end,2), mu);
```

```
ancomptime=toc;

Variation_DeltaV = abs(DeltaV_An-DeltaV_num)/abs(DeltaV_num);
Variation_Comptime = assess_difference(ancomptime,numcomptime);

%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function [dGdt] = GaussEquations_max_a(t,G,constants)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Gauss' equations (Equation 2 in Di Carlo and Vasile 2021).
%variable "constants.... consider how to globalize....


epsilon= constants.thrust; %Magnitude of accelleration force
mu = constants.mu; %Gravitational constant of central body

%Unpack the input
a = G(1); %Semimajor axis (km)
e = G(2); %Eccentricity
i = G(3); %Inclination
RA = G(4); %Right ascencion of the node (radians)
w = G(5); %Argument of perigee (radians)
theta = G(6); %True anomoly

%Intermediate calculations
p = a*(1-e^2); %semilatus rectum
h = sqrt(mu*p); %specific angular momentum % was mu/p here
r = h^2/mu/(1 + e*cos(theta)); %The radius
nu = w + theta; %Argument of latitude

%Thrust angle calculations
alpha=atan2(e*sin(theta),1+e*cos(theta));
beta=0;

%Differential Equations
%==================================================================
adot = ((2*a^2)/h)*epsilon*cos(beta)*(e*sin(theta)*sin(alpha)+...
(p/r)*cos(alpha));

edot = epsilon*cos(beta)/h*((p*sin(theta)*sin(alpha)+...
((p+r)*cos(theta)+r*e))*cos(alpha));
```

```
idot = epsilon*(r/h)*cos(nu)*sin(beta);

RAdot = epsilon*(r/h)*sin(nu)/sin(i)*sin(beta);

wdot = 1/(h*e)*epsilon*cos(beta)*(-p*cos(theta)*sin(alpha)+...
(p+r)*sin(theta)*cos(alpha))-(r*sin(nu)*cos(i))/(h*sin(i))*epsilon*sin(
    beta);

thetadot = (h/r^2)+epsilon/(e*h)*cos(beta)*(p*cos(theta)*sin(alpha)-...
(p+r)*sin(theta)*cos(alpha));
%=====================================================================
% thetadot = h/(r^2); %thetadot approx deriv (DiCarlo Vasile 2021 pp4)
    also


% Assemble output
dGdt = [adot;
        edot;
        idot;
        RAdot;
        wdot;
        thetadot];

end %GaussEquations
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function

%~~SUB-FUNCTION~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    function [value, isterminal, direction] = delta_a_event(T,Y)
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Terminates integration of Gauss' Equations when final a crossed from low
    to
high
%}
value      = (Y(1) > af);
isterminal = 1;   % Stop the integration
direction  = 0;
    end
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of sub-function
```

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%{
Calculated the order of magnitude difference betweeen two numbers
%}
function order_of_magnitude_difference = assess_difference(a, b)

magnitude_a = log10(abs(a));
magnitude_b = log10(abs(b));

order_of_magnitude_difference = magnitude_a - magnitude_b;

end

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~End of main-function
```

E.3.3 Regional Accuracy Plots

```
    %Numeric DeltaV calcs for Plots

clc,clear
LatexTitles
[G,m_earth,m_moon,m_sun,...
          mu_earth,mu_moon,...
          a_earth,a_moon,...
          r_moon,r_earth,...
          moon_SOI] = Cislunar_constants;

epsilon=10E-7;
mu=mu_earth;
RCB=r_earth;
days=500;

%Run1 low e low a0

e0 = 0.1;
i0 = 0.1;
```

```
RA0 = 0.1;
w0 = 0.1;
theta0 = pi;

a0low=7500;
a0high=36000;
slice=15;
arange = linspace(a0low,a0high,slice);
transfer=linspace(1500,10000,slice);


Variation_DeltaV=zeros(slice,slice);
Variation_Comptime=Variation_DeltaV;

for m = 1:slice
    a0=arange(m);
    for n = 1:slice
    af = arange(m)+transfer(n);
    coe0=[a0 e0 i0 RA0 w0 theta0]';

    [Variation_DeltaV(m,n),Variation_Comptime(m,n)]=...
    Gauss_max_a_DeltaV_deviation_15s_analytic...
    (coe0,af,epsilon,mu,days);
end
end


contourf(transfer,arange,Variation_DeltaV)
subtitle('$\delta \Delta V=\frac{\Delta V_{Num}-\Delta V_{An}}{\Delta V_{Num}}$ , $\overline{e}=0.1$')
colorbar;xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{a}_0 \ [km]$')
contourf(transfer,arange,Variation_Comptime)
subtitle('$\delta$ Computation Time, $\overline{e}=0.1$')
colorbar,xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{a}_0 \ [km]$')


%Run 2 high e mod a0
e0 = 0.7;
```

```
i0 = 0.1;
RA0 = 0.1;
w0 = 0.1;
theta0 = pi;

a0low=22000;
a0high=36000;
slice=15;
arange = linspace(a0low,a0high,slice);
transfer=linspace(2000,10000,slice);


Variation_DeltaV=zeros(slice,slice);
Variation_Comptime=Variation_DeltaV;



for m = 1:slice
    a0=arange(m);
    for n = 1:slice
    af = arange(m)+transfer(n);
    coe0=[a0 e0 i0 RA0 w0 theta0]';

    [Variation_DeltaV(m,n),Variation_Comptime(m,n)]=...
    Gauss_max_a_DeltaV_deviation_15s_analytic...
    (coe0,af,epsilon,mu,days);
end
end


% p=surf(transfer,arange, Variation_DeltaV);
% view([0.91431 90.00000])
% p.FaceColor='interp';%p.EdgeColor="none"
% colorbar


contourf(transfer,arange,Variation_DeltaV)
subtitle('$\delta \Delta V=\frac{\Delta V_{Num}-\Delta V_{An}}{\Delta V_
    {Num}}$ , $\overline{e}=0.7$')
colorbar,xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{a}_0
```

```
    \ [km]$')
contourf(transfer,arange,Variation_Comptime)
subtitle('$\delta$ Computation Time , $\overline{e}=0.7$')
colorbar,xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{a}_0
    \ [km]$')

%Fixed a0 BIG, Varying eccentricity, Varying Transfer
e0 = 0.1;
i0 = 0.1;
RA0 = 0.1;
w0 = 0.1;
theta0 = pi;

a0low=7500;
a0high=36000;

a0=a0high;

slice=15;
arange = linspace(a0low,a0high,slice);
erange = linspace(0.01, 0.9,slice);
transfer=linspace(1500,10000,slice);


Variation_DeltaV=zeros(slice,slice);
Variation_Comptime=Variation_DeltaV;

for m = 1:slice
    e0=erange(m);
    for n = 1:slice
    af = a0+transfer(n);
    coe0=[a0 e0 i0 RA0 w0 theta0]';

    [Variation_DeltaV(m,n),Variation_Comptime(m,n)]=...
    Gauss_max_a_DeltaV_deviation_15s_analytic...
    (coe0,af,epsilon,mu,days);
end
end


```

```
contourf(transfer,erange,Variation_DeltaV)
subtitle('$\delta \Delta V=\frac{\Delta V_{Num}-\Delta V_{An}}{\Delta V_{Num}}$ , $\overline{a}_0=3.6\times 10^6$, $\overline{e}_0=0.1$')
colorbar;xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{e}_0$')
contourf(transfer,arange,Variation_Comptime)
subtitle('$\delta$ Computation Time,  $\overline{a}_0=3.6\times 10^6$, $\overline{e}_0=0.1$')
colorbar,xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{e}_0$')


%Fixed a0 small, Varying eccentricity, Varying Transfer
e0 = 0.1;
i0 = 0.1;
RA0 = 0.1;
w0 = 0.1;
theta0 = pi;

a0low=7500;
a0high=36000;

a0=10000;

slice=15;
arange = linspace(a0low,a0high,slice);
erange = linspace(0.01, 0.9,slice);
transfer=linspace(1500,10000,slice);


Variation_DeltaV=zeros(slice,slice);
Variation_Comptime=Variation_DeltaV;

for m = 1:slice
    e0=erange(m);
    for n = 1:slice
    af = a0+transfer(n);
    coe0=[a0 e0 i0 RA0 w0 theta0]';
    [Variation_DeltaV(m,n),Variation_Comptime(m,n)]=...
    Gauss_max_a_DeltaV_deviation_15s_analytic...
```

```
    (coe0,af,epsilon,mu,days);
end
end


contourf(transfer,erange,Variation_DeltaV)
subtitle('$\delta \Delta V=\frac{\Delta V_{Num}-\Delta V_{An}}{\Delta V_{Num}}$ , $\overline{a}_0=1\times 10^6$')
colorbar;xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{e}_0$')
contourf(transfer,arange,Variation_Comptime)
subtitle('$\delta$ Computation Time,  $\overline{a}_0=3.6\times 10^6$')
colorbar,xlabel('$\Delta \overline{a} \ [km]$'),ylabel('$\overline{e}_0$')
```

E.4 Major Evaluations of Optimality

The included code in these sections requires access to a DIDO license and is available upon request from the Dudley Knox Library. All code in this section adapted from [13].

List of References

[1] L. Yue, Q. Yingjing, and J. Wuxing, "Orbit injection control of low thrust low energy Moon return via STK," in *2018 Chinese Control And Decision Conference (CCDC)*. Shenyang, CHN: IEEE, June 2018, pp. 3783–3788 [Online]. Available: https://doi.org/10.1109/CCDC.2018.8407780

[2] D. C. Folta, N. Bosanac, I. Elliott, L. Mann, R. Mesarch, and J. Rosales, "Astrodynamics Convention and Modeling Reference for Lunar, Cislunar, and Libration Point Orbits," NASA, Technical Publication (TP) 20220014814, Oct. 2022 [Online]. Available: https://ntrs.nasa.gov/citations/20220014814

[3] M. J. Holzinger, C. C. Chow, and P. Garretson, "A Primer on Cislunar Space," Department of Defense, U.S. Air Force, AFRL, 26 Electronic Parkway, Rome, NY, USA., Rep., 2020.

[4] M. Di Carlo and M. Vasile, "Analytical solutions for low-thrust orbit transfers," *Celestial Mechanics and Dynamical Astronomy*, vol. 133, no. 7, p. 33, July 2021 [Online]. Available: https://doi.org/10.1007/s10569-021-10033-9

[5] Y. Liu, Y. Qian, and W. Jing, "Study on Low Energy Moon Return Orbit Construction and Low Thrust Propulsion Control Strategy," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. Xiamen, CHN: IEEE, Aug. 2018, pp. 1–6 [Online]. Available: https://doi.org/10.1109/GNCC42960.2018.9019023

[6] J. C. Moltz, *The Politics of Space Security, Strategic Restraint and the Pursuit of National Interests*, 2nd ed. Stanford, CA, USA: Stanford University Press, 2011.

[7] J. C. Moltz, "Crowded orbits: Conflict and cooperation in space," *International Affairs*, vol. 91, no. 2, pp. 413–414, 2015.

[8] D. F. Lawden, *Optimal trajectories for space navigation.* (Butterworths mathematical texts). London, GBR: Butterworths, 1963.

[9] R. R. Bate, *Fundamentals of Astrodynamics*. New York, NY, USA: Dover Publications, 1971.

[10] J. E. Prussing, *Orbital Mechanics*. New York, NY, USA: Oxford University Press, 1993.

[11] *Space Mission Engineering : The New SMAD* (Space technology library ; v. 28). Hawthorne, CA, USA: Microcosm Press, 2011.

[12] H. D. Curtis, *Orbital Mechanics for Engineering Students*, revised fourth edition. ed. (Elsevier aerospace engineering series.). Oxford, GBR: Butterworth-Heinemann, An imprint of Elsevier, 2021 - 2021.

[13] I. M. Ross, *A primer on Pontryagin's Principle in Optimal Control*. Carmel, CA, USA: Collegiate Publishers, 2009.

[14] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, Inc., Jan. 1999 [Online]. Available: https://doi.org/10.2514/4.861543

[15] A. E. Petropoulos, "Simple control laws for low-thrust orbit transfers," Jet Propulsion Laboratory, Sunnyvale, CA, USA, Rep., 2003 [Online]. Available: https://dataverse.jpl.nasa.gov/dataset.xhtml?persistentId=hdl:2014/38468

[16] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 3rd ed. (Space technology library ; v. 21). New York, NY, USA: Springer, 2007.

[17] M. J. Holzinger, C. C. Chow, and P. Garretson, "A Primer on Cislunar Space," USAF, Rep., 2021.

[18] J. S. Hudson and D. J. Scheeres, "Reduction of Low-Thrust Continuous Controls for Trajectory Dynamics," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 780–787, May 2009 [Online]. Available: https://doi.org/10.2514/1.40619

[19] S. P. Hughes, R. H. Qureshi, S. D. Cooley, and J. J. Parker, "Verification and Validation of the General Mission Analysis Tool (GMAT)," in *AIAA/AAS Astrodynamics Specialist Conference*. San Diego, CA, USA: American Institute of Aeronautics and Astronautics, Aug. 2014 [Online]. Available: https://doi.org/10.2514/6.2014-4151

[20] W. C. Nelson, *Space Mechanics* (Prentice-Hall international series in space technology). Englewood Cliffs, NJ, USA: Prentice-Hall, 1962.

[21] R. Deutsch, *Orbital Dynamics Of Space Vehicles* (Prentice-Hall international series in space technology). Englewood Cliffs, NJ, USA: Prentice-Hall, 1964.

[22] J. P. Marec, *Optimal Transfers Between Close Elliptical Orbits* (NASA TT ; F-554). Washington, DC, USA: National Aeronautics and Space Administration, 1969.

[23] J. P. Marec, *Optimal Space Trajectories* (Studies in Astronautics ; v. 1). Amsterdam, NLD: Elsevier Scientific Pub. Co., 1979.

[24] V. A. Brumberg, *Analytical Techniques of Celestial Mechanics*. Berlin, DEU: Springer-Verlag, 1995.

[25] E. Burt, "On space manoeuvres with continuous thrust," *Planetary and Space Science*, vol. 15, no. 1, pp. 103–122, 1967.

[26] J. E. Pollard, "Simplified analysis of low-thrust orbital maneuvers," Aerospace Corporation, CA, USA, Rep., 2000.

[27] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y.-S. Zhao, "Low-thrust minimum-fuel optimization in the circular restricted three-body problem," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 8, pp. 1501–1510, 2015.

[28] J. S. Parker and R. L. Anderson, *Low-energy lunar trajectory design* (JPL deep-space communications and navigation series). Somerset, GBR: WILEY, 2014.

[29] Z. Hall, D. Schwab, R. Eapen, and P. Singla, "Reachability-based approach for search and detection of maneuvering cislunar objects," in *AIAA SCITECH 2022 Forum*.

[30] M. T. Ozimek and K. C. Howell, "Low-thrust transfers in the earth-moon system, including applications to libration point orbits," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 533–549, 2010.

[31] J. T. Betts and S. O. Erb, "Optimal low thrust trajectories to the moon," *SIAM Journal on Applied Dynamical Dystems*, vol. 2, no. 2, pp. 144–170, 2003.

[32] K. Rivera, A. Zara, D. Aguilar-Marsillach, M. Holzinger, I. Elliott, and N. Bosanac, "Patterns of Life and Maneuver Detection for Cislunar Trajectory Maintenance," 2021.

[33] J. Woodburn and J. H. Seago, "Orbit determination of satellites in lunar orbit using an optimal sequential filter," in *Spaceflight Mechanics 2008, VOL 130, Parts 1 and 2* (Advances in the Astronautical Sciences). San Diego: Univelt Inc, 2008, vol. 130, pp. 61–79.

[34] C. A. Kluever and K.-R. Chang, "Near-optimal low-thrust lunar trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 2, pp. 494–496, 1996.

[35] J. P. Vinti, "Zonal harmonic perturbations of an accurate reference orbit of an artificial satellite," *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, vol. 67B, no. 4, p. 191, Oct. 1963 [Online]. Available: https://doi.org/10.6028/jres.067B.016

[36] "HPOP Results Example" [Online]. Available: https://help.agi.com/stk/11.0.1/Content/hpop/example_HPOPResults.htm

[37] G. E. O. Giacaglia, J. P. Murphy, and T. L. Felsentreger, "A semi-analytic theory for the motion of a lunar satellite," *Celestial Mechanics*, vol. 3, no. 1, pp. 3–66, Mar. 1970 [Online]. Available: https://doi.org/10.1007/BF01230432

[38] "Lunar Constants and Models Document," Sep. 2005 [Online]. Available: https://www.hq.nasa.gov/alsj/lunar_cmd_2005_jpl_d32296.pdf

[39] J. Peypouquet, *Convex Optimization in Normed Spaces: Theory, Methods and Examples* (SpringerBriefs in Optimization). Springer International Publishing, 2015 [Online]. Available: https://books.google.com/books?id=puwEogEACAAJ

[40] L. Pontryagin, *The Mathematical Theory of Optimal Processes* (Classics of Soviet mathematics). Routledge, 2018, vol. 4.

[41] I. M. Ross, "Enhancements to the dido optimal control toolbox," Naval Postgraduate School, Monterey, CA, USA, Rep., 2020 [Online]. Available: https://arxiv.org/abs/2004.13112

[42] I. M. Ross, Q. Gong, and P. Sekhavat, "Low-thrust, high-accuracy trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 921–933, 2007.

[43] D. Morante, M. Sanjurjo Rivo, and M. Soler, "A survey on low-thrust trajectory optimization approaches," *Aerospace*, vol. 8, no. 3, pp. 88–, 2021.

[44] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences*, vol. 102, pp. 76–98, 2018.

[45] E. Galperin, "Reflections on bellman's optimality principle," *Nonlinear Analysis*, vol. 63, no. 5, pp. e707–e713, 2005.

[46] C. Short, P. Ghosh, and A. Claybrook, "Revisiting trajectory design with stk astrogator part 1," in *American Astronomical Society Conference*, San Diego, CA, USA, January 2019 [Online]. Available: https://www.agi.com/getmedia/5fc77cdf-9a7c-4f06-a825-6e7bd49284e9/RevisitingTrajectoryDesign_Final.pdf?ext=.pdf

[47] C. Short, P. Ghosh, and A. Claybrook, "Revisiting trajectory design with stk astrogator part 1," in *American Astronomical Society Conference*, San Diego, CA, USA, January 2019 [Online]. Available: https://www.agi.com/getmedia/5fc77cdf-9a7c-4f06-a825-6e7bd49284e9/RevisitingTrajectoryDesign_Final.pdf?ext=.pdf

[48] P. Ghosh, "A survey of the methods available for the design of many-revolution low-thrust planetocentric trajectories," in *American Astronomical Society Conference*, San Diego, CA, USA, January 2019 [Online]. Available: https://www.academia.edu/44268097/AAS_19_297

[49] T. McElrath, "Exploration of the sparse nonlinear optimizer profile (SNOPT) in astrogator," Analytical Graphics, Inc., Rep., December 2019 [Online]. Available: https://analyticalgraphics.force.com/faqs/articles/Knowledge/Tips-and-Tricks-for-the-Sparse-Nonlinear-Optimizer-SNOPT-Profile-in-Astrogator

[50] I. M. Ross, Q. Gong, M. Karpenko, and R. J. Proulx, "Scaling and balancing for high-performance computation of optimal controls," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 10, pp. 2086–2097, 2018.

[51] Y. Meng, H. Zhang, and Y. Gao, "Low-thrust minimum-fuel trajectory optimization using multiple shooting augmented by analytical derivatives," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 3, pp. 662–677, 2019.

[52] C. A. Kluever and S. R. Oleson, "Direct approach for computing near-optimal low-thrust earth-orbit transfers," *Journal of Spacecraft and Rockets*, vol. 35, no. 4, pp. 509–515, 1998.

[53] M. Di Carlo, J. M. Romero Martin, and M. Vasile, "Camelot: Computational-analytical multi-fidelity low-thrust optimisation toolbox," *CEAS Space Journal*, vol. 10, no. 1, pp. 25–36, 2018.

[54] I. M. Ross, Q. Gong, and P. Sekhavat, "Low-thrust, high-accuracy trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 921–933, July 2007 [Online]. Available: https://doi.org/10.2514/1.23181

[55] A. Ruggiero, P. Pergola, S. Marcuccio, and M. Andrenucci, "Low-thrust maneuvers for the efficient correction of orbital elements," in *Proceedings of the 32nd International Elec-tric Propulsion Conference*, Sep. 2011, pp. IEPC–2011–102 [Online]. Available: http://electricrocket.org/IEPC/IEPC-2011-102.pdf

[56] D. F. Lawden, *Optimal Trajectories for Space Navigation.* (Butterworths mathematical texts). London, GBR: Butterworths, 1963.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE