



MINIMUM-FUEL TRAJECTORY DESIGN IN
MULTIPLE DYNAMICAL ENVIRONMENTS
UTILIZING DIRECT TRANSCRIPTION
METHODS AND PARTICLE SWARM
OPTIMIZATION

THESIS

Alfredo G. Zurita, Jr., 2d Lt, USAF
AFIT-ENY-MS-16-M-250

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-MS-16-M-250

MINIMUM-FUEL TRAJECTORY DESIGN IN MULTIPLE DYNAMICAL
ENVIRONMENTS UTILIZING DIRECT TRANSCRIPTION METHODS AND
PARTICLE SWARM OPTIMIZATION

THESIS

Presented to the Faculty
Department of Astronautics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Astronautical Engineering

Alfredo G. Zurita, Jr., B.S.

2d Lt, USAF

March 2016

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

MINIMUM-FUEL TRAJECTORY DESIGN IN MULTIPLE DYNAMICAL
ENVIRONMENTS UTILIZING DIRECT TRANSCRIPTION METHODS AND
PARTICLE SWARM OPTIMIZATION

THESIS

Alfredo G. Zurita, Jr., B.S.
2d Lt, USAF

Committee Membership:

Maj Christopher D. Geisel, Ph.D.
Chair

William E. Wiesel, Ph.D.
Member

Matthew J. Dillsaver, Ph.D.
Member

Maj Stuart A. Stanton, Ph.D.
Member

Abstract

Particle swarm optimization, an evolutionary algorithm modeled after natural swarm behavior, is used to generate an initial guess for designing fuel-optimal trajectories in multiple dynamical environments. Trajectories designed in the vicinity of Earth use continuous or finite low-thrust burning and transfer from an inclined or equatorial circular low-Earth-orbit to a geostationary orbit. In addition, a trajectory from near-Earth to a periodic orbit about the cislunar Lagrange point with minimized impulsive burn costs is designed within a multi-body dynamical environment. Direct transcription is used in conjunction with a nonlinear optimizer to find locally-optimal trajectories given the particle swarm optimization generated initial guess. The near-Earth transfers are propagated at low-level thrust where neither the very-low-thrust spiral solution nor the impulsive transfer is an acceptable starting point. The very-high-altitude transfer is designed in a multi-body dynamical environment lacking a closed-form analytical solution. Swarming algorithms excel at finding global optima given a small number of design parameters. When continuous control time histories are needed, employing a polynomial parameterization approach in conjunction with particle swarm optimization successfully generates feasible solutions for small problems. For dynamical environments in which chaos is present, such as in a circular restricted three-body system, particle swarm optimization gains utility due to a more global search for the solution, but may be more sensitive to boundary constraints. Computation time and constraint weighting are areas where a swarming algorithm is weaker than other approaches. The design methodologies employed are useful when an initial guess is not available for unorthodox trajectories or for designing in a complex dynamical environment.

AFIT-ENY-MS-16-M-250

To Haha, Chichi, Mahiely, Lala, Maggie, and Paddy.

Acknowledgments

The completion of this thesis would not have been possible without the support, guidance, and expertise of several individuals who assisted me throughout the research process. Without their assistance, I fear I would still be lost in a “chaotic sea.”

First and foremost, I would like to acknowledge my thesis adviser and three-body problem guru, Maj Christopher D. Geisel. His attention to detail and guiding questions proved to be extremely invaluable and helpful. I especially appreciate his willingness to explore the details of a given problem yielding better understanding as well as his availability for mentoring. Our weekly research meetings proved to be very advantageous in practicing how to present and talk about technical subject matter. Lastly, his Numerical Methods for Orbit Design class gave me a true appreciation for much of the subject matter in this research and became one of my favorite courses taken in my academic career.

Dr. William E. Wiesel, someone I regard as a legend in the astronautics community, also has my sincere appreciation. Our discussions always optimally pointed me in the right direction. I found myself gaining a greater understanding and appreciation for the nuggets of knowledge he imparts as I followed his direction. Also, his sense of humor in the classroom setting was always a breath of fresh air during the rigors of academics.

Dr. Matthew J. Dillsaver initiated my learning in the field of optimization. Our one-on-one sessions in parameter optimization gave me an enhanced understanding of the fundamentals of optimization, something I desperately needed prior to tackling more complex problems. Also, his expertise in computer implementation of optimization schemes helped solve issues I had along the way.

Maj Stuart A. Stanton proved to be an invaluable research sponsor. His guidance helped pinpoint the question my research sought to answer as well as provided a

macroscopic perspective on the relevance of this investigation. His helpful optimization guidance was a source of motivation due to his enthusiasm about the material.

Dr. Alan Jennings was an excellent source of knowledge about evolutionary algorithms. I gained an understanding about their strengths and limitations early on in the research process and was given useful advice on applying them in this investigation.

Lastly, to the Astrodynamics and Space Applications research group, thank you for listening to me talk about my research on a weekly basis and providing insightful questions and avenues for improvement. Our jokes and sharing of experiences helped keep the process fun and enjoyable.

Alfredo G. Zurita, Jr.

Table of Contents

	Page
Abstract	iv
Acknowledgments	vi
List of Figures	x
List of Tables	xv
List of Abbreviations	xvi
1. Introduction	1
1.1 Motivation	1
1.2 Research Objectives	4
1.3 Decision Tree for Trajectory Optimization Techniques	5
1.4 Document Preview and Special Notes	7
1.5 Chapter Summary	9
2. Background and Literature Review	10
2.1 The Two-Body Problem	10
2.1.1 Historical Context	10
2.1.2 Two-Body Derivation	12
2.1.3 Classical Orbital Elements	20
2.1.4 Modified Equinoctial Elements	25
2.1.5 Perturbation Methods and Accounting for Oblate Earth Effects	26
2.2 The Three-Body Problem	29
2.2.1 The Circular Restricted Three-Body Problem	30
2.2.2 Insight into the CR3BP	37
2.2.3 Equations of Variation	41
2.2.4 Targeting Trajectories	45
2.2.5 Stability of the Equilibrium Points	48
2.2.6 Dynamical Systems Theory	51
2.3 Optimization Fundamentals and Techniques	56
2.3.1 Parameter Optimization	58
2.3.2 Optimal Control and Indirect Transcription	62
2.3.3 Direct Transcription and Shooting	65
2.3.4 Runge-Kutta Shooting	69
2.3.5 Particle Swarm Optimization	71
2.4 Spacecraft Propulsion	76
2.5 Relevant Works in Literature	79
2.6 Chapter Summary	82

	Page
3. Low-Thrust Near-Earth Trajectory Design	83
3.1 Methodology	84
3.1.1 Low-Thrust Transfer Model	84
3.1.2 PSO Initial Guess	89
3.1.3 NLP Improvement	92
3.2 Results	94
3.2.1 Continuous Thrust Planar LEO to GEO Transfer	95
3.2.1.1 PSO Initial Guess Generation	96
3.2.1.2 NLP Improvement	98
3.2.2 Multiple-burn Planar LEO to GEO Transfer	101
3.2.2.1 PSO Initial Guess Generation	101
3.2.2.2 NLP Improvement	103
3.2.3 Continuous Thrust Non-Coplanar LEO to GEO Transfer	106
3.2.3.1 PSO Initial Guess Generation	107
3.2.3.2 NLP Improvement	112
3.2.4 Multiple-burn Non-Coplanar LEO to GEO Transfer	117
3.2.4.1 PSO Initial Guess Generation	117
3.2.4.2 NLP Improvement	119
3.2.5 Practicality of the Trajectories	122
3.3 Chapter Summary	126
4. Impulsive High-Altitude Trajectory Design	128
4.1 Methodology	129
4.1.1 Targeting Periodic Orbits	130
4.1.2 LPO Invariant Manifold	134
4.1.3 PSO Initial Guess and NLP Improvement	138
4.2 Results	139
4.2.1 PSO Initial Guess Generation	144
4.2.2 NLP Improvement	148
4.3 Chapter Summary	153
5. Conclusions	155
5.1 Limitations and Future Work Recommendations	158
Bibliography	161

List of Figures

Figure		Page
1	Decision tree for space trajectory optimization in the current investigation	6
2	Two-body motion in an inertial reference frame	13
3	Conic sections, adapted from Bate, Mueller, and White [1]	20
4	Orbit size, adapted from Bate, Mueller, and White [1]	22
5	COEs in Earth-centered inertial frame, adapted from Chobotov [2]	25
6	Three-body system in inertial reference frame	32
7	Barycentric rotating reference frame, adapted from Stuart [3]	34
8	Earth-Moon equilibrium points in the nondimensional barycentric rotating frame	39
9	Lagrange point associated zero-velocity curves	40
10	Example spacecraft trajectory propagated for 13.64 days in barycentric rotating frame with associated ZVCs	42
11	Change in Jacobi constant for the example spacecraft trajectory in the first 200 seconds and the entire 13.64 day propagation time	42
12	Notional diagram of a 2-D saddle $\times$ 2-D center equilibrium point, adapted from Geisel [4]	50
13	Periodic orbit in the vicinity of L1 with associated stable and unstable eigenvectors and ZVCs in nondimensional rotating barycentric frame	54
14	Stable and unstable manifold trajectories emanating from periodic orbit in the vicinity of L1 in nondimensional rotating barycentric frame	57
15	Local and global extrema, adapted from Arora [5]	59

Figure		Page
16	Discretized state and control	66
17	Notional diagram of single shooting	68
18	Three-step Runge-Kutta integration, adapted from Conway [6]	70
19	Diagram of the PSO position velocity update process	74
20	Acceleration vs. effective exhaust velocity, reproduced from Sutton and Biblarz [7]	78
21	Thrust pointing angles	85
22	Singular arc control example	90
23	Diagram of a grid refinement scheme	93
24	PSO generated continuous thrust, LEO to GEO, coplanar transfer, 0.639 day transfer (black arrows correspond to thrust pointing directions)	97
25	NLP improved continuous thrust, LEO to GEO, planar transfer, 0.607 day transfer (black arrows correspond to thrust pointing directions)	99
26	Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, planar transfer	100
27	PSO generated multiple-burn, LEO to GEO, planar transfer, 0.773 day transfer	103
28	NLP improved multiple-burn, LEO to GEO, planar transfer, 0.904 day transfer	105
29	Control time history and osculating elements for NLP improved multiple-burn, LEO to GEO, planar transfer	105
30	PSO generated continuous thrust, LEO to GEO, non-coplanar transfer, 3-D view, 0.664 day transfer (black arrows correspond to thrust pointing directions)	109
31	PSO generated continuous thrust, LEO to GEO, non-coplanar transfer, x - y view, 0.664 day transfer (black arrows correspond to thrust pointing directions)	109

Figure		Page
32	Control time history for PSO generated continuous thrust, LEO to GEO, non-coplanar transfer	110
33	PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.654 day transfer (black arrows correspond to thrust pointing directions)	110
34	PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.654 day transfer (black arrows correspond to thrust pointing directions)	111
35	Control time history for PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation	111
36	NLP improved continuous thrust, LEO to GEO, non-coplanar transfer, 3-D view, 0.650 day transfer (black arrows correspond to thrust pointing directions)	113
37	NLP improved continuous thrust, LEO to GEO, non-coplanar transfer, x - y view, 0.650 day transfer (black arrows correspond to thrust pointing directions)	113
38	Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, non-coplanar transfer	114
39	NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.624 day transfer (black arrows correspond to thrust pointing directions)	115
40	NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.624 day transfer (black arrows correspond to thrust pointing directions)	115
41	Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation	116

Figure	Page
42	Comparison of osculating elements between NLP improved continuous thrust, LEO to GEO, non-coplanar transfers with and without J2 perturbation effects. 116
43	PSO generated multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.785 day transfer 118
44	PSO generated multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.785 day transfer 118
45	Control time history for PSO generated multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation 119
46	NLP improved multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.802 day transfer 120
47	NLP improved multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.802 day transfer 120
48	Control time history and osculating elements for NLP improved multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation 121
49	Targeting a planar periodic orbit near L1 in nondimensional rotating barycentric frame 132
50	One revolution of the targeted planar periodic orbit near L1 in nondimensional rotating barycentric frame 132
51	PSO convergence on LPO for 30 particles and 100 iterations 134
52	Family members of the L1 Lyapunov family in nondimensional rotating barycentric frame 135
53	Stable manifold tube propagated from LPO in nondimensional rotating barycentric frame, propagated for 43.4 days 136
54	Zoomed in view of stable manifold tube propagated from LPO in nondimensional rotating barycentric frame 137

Figure		Page
55	Initial LEO altitude orbit and associated ZVCs in rotating barycentric frame, propagated for 1.52 hours	140
56	Chosen stable manifold trajectory and associated ZVCs in rotating barycentric frame, propagated for 86.8 days	142
57	Zoomed in view of chosen stable manifold trajectory and associated ZVCs in rotating barycentric frame	143
58	Nineteen PSO generated trajectories for various stable manifold insertion points in barycentric rotating frame (ZVCs not shown)	146
59	Three impulsive transfer from LEO to L1 LPO in rotating barycentric frame, 23.4 day transfer (ZVCs not shown)	150
60	Three impulsive transfer from LEO to L1 LPO in Earth-centered inertial frame, 23.4 day transfer	151

List of Tables

Table		Page
1	Parameter ranges for conic sections	23
2	Orbit inclinations	24
3	Characteristic quantities for the CR3BP	33
4	CR3BP Earth-Moon Lagrange points	39
5	Test case thrust specifications	87
6	Initial and terminal conditions for LEO to GEO coplanar transfer	95
7	Initial and terminal conditions for LEO to GEO non-coplanar transfer	106
8	Initial and terminal conditions for LEO to GEO non-coplanar transfer using MEEs	107
9	Low-thrust transfers comparison	123
10	PSO trajectory specifications	147
11	LEO to LPO trajectory impulse magnitudes	152

List of Abbreviations

2BP	-	Two-body problem
COEs	-	Classical orbital elements
CR3BP	-	Circular restricted three-body problem
EA	-	Evolutionary algorithm
EOMs	-	Equations of motion
EOVs	-	Equations of variation
GA	-	Genetic algorithm
GEO	-	Geosynchronous orbit
GPS	-	Global positioning system
KKT	-	Karush-Kuhn-Tucker
LEO	-	Low-Earth-orbit
LPO	-	Libration point orbit
MEEs	-	Modified equinoctial elements
NLP	-	Nonlinear programming
nd	-	Nondimensional
PSO	-	Particle swarm optimization
PSOIG	-	Particle swarm optimization generated initial guess
RK	-	Runge-Kutta
SNOPT	-	Sparse nonlinear optimizer
STM	-	State transition matrix
TPBVP	-	Two-point boundary value problem
ZVCs	-	Zero-velocity curves

MINIMUM-FUEL TRAJECTORY DESIGN IN MULTIPLE DYNAMICAL ENVIRONMENTS UTILIZING DIRECT TRANSCRIPTION METHODS AND PARTICLE SWARM OPTIMIZATION

1. Introduction

The field of optimization has broad relevance to nearly all technical disciplines and fields. Optimization serves to answer what the lowest or highest value a desired performance measurement can achieve given certain variables and constraints. The desire to design minimum-fuel trajectories arises in the context of space because efficient use of fuel reduces cost and allows for greater payload mass utilization. There are a variety of optimization techniques equipped to operate in the variety of dynamical environments used to model trajectories in space. However, most of these techniques require an educated initial guess about the behavior of the trajectory. In complex dynamical environments, an acceptable initial guess may not be readily available. To remedy this concern, evolutionary algorithms are chosen for investigation due to their ability to operate without an initial guess. Specifically, particle swarm optimization is utilized due to its algorithmic simplicity. Given the generated initial guess, further improvement is conducted by more robust numerical optimization techniques. The trajectories designed are chosen based on current and future relevance to military operations.

1.1 Motivation

During the transition into the early twenty-first century, the space domain has seen an increase in traffic and usage. In this evolving environment, the United States has

maintained a competitive edge over other nations in its space assets and technologies. However, that edge may gradually be dulled or jeopardized as other space-faring entities modernize and improve their own space programs.

Although the space domain is largely unpopulated and empty, it is popularly described as a “congested, contested, and competitive” environment [8]. The reason for this disparity is due to the existence of regions in the vicinity of Earth that hold greater utility over other regions. Prime examples of this are geostationary and Tundra orbits, where the spacecraft’s orbital period matches that of the Earth’s period of rotation, allowing it to stay above a desirable region for long periods of time if not indefinitely. This feature of matched periods is extremely attractive for satellites that need to provide constant coverage such as SiriusXM®, DirecTV®, and various communication and weather satellites [9,10]. As a result, this altitude band is highly populated and frequented. In addition, low-Earth-orbit (LEO), often defined to be greater than 160 kilometers and less than 2,000 kilometers in altitude, is also rich in space assets due to its accessibility, affordability, and proximity benefits. The benefits of this “ultimate high ground” result in a congested environment; thus, the United States Air Force may find that new trajectory design strategies or new areas of operations afford additional asset protection [11].

The President of the United States Barack H. Obama, in the 2011 National Security Space Strategy, stated that a key necessity for space planning is to stress the requirements of mission continuation and sustaining operations. In response, the “availability of alternate means for mission accomplishment” not only adds flexibility, but bolsters the resiliency of assets [12]. From a supplemental fact sheet released by the Department of Defense and in the 2013 United States Air Force Posture Statement, “resilience is the ability of an architecture to support the function necessary for mission success in spite of hostile action or adverse conditions” [13,14]. A recent

white paper from the Office of the Assistant Secretary of Defense further breaks down resiliency into six underpinnings: disaggregation, protection, distribution, proliferation, diversification, diversification, and deception [15]. The principle of protection is most relevant to this investigation because it calls for measures to “enable satellite operators to restore function, capabilities, or capacity after a natural or anthropogenic adverse event” [15]. An anthropogenic event is one which originates from human activity. The satellite AsiaSat 3 demonstrated asset protection as well as the utility of high-altitude, multi-body, trajectory design in its response to a mission threatening contingency [16]. In essence, it was able to overcome the prohibitive fuel cost of a large orbital inclination change by performing two successive lunar flybys in order to be placed into its desired geosynchronous orbit. Applying this type of strategy to military assets could very well increase resiliency especially in the area of protection. In addition, trajectories designed in a complex dynamical environment acquire a level of unpredictability that can also improve assets in their levels of “avoidance,” which can be described as measures taken to prevent potential threats [13].

Low-thrust maneuvers are typically performed due to their high fuel efficiency and are often executed by electrical propulsion systems. Historically, chemical propulsion systems have been the predominant choice for space systems due to flight heritage and reliability. However, in recent years, electric propulsion has gained popularity as well as successful flight demonstration and research. In fact, Boeing, the main GPS satellite provider to the Air Force, has been transitioning to electric propulsion and has even demonstrated successful operation of an “all-electric satellite” [17]. Although the lower thrust levels associated with electric propulsion may provide more fuel efficiency, this comes at the cost of longer time of flight. Designing at thrust acceleration levels between very-low-thrust and high-thrust gives the designer an intermediate option for balancing fuel expenditure and time of flight.

Cost has always been one of the major concerns for operating in space. Not only are support and maintenance expensive, but the mission designer must also be cognizant of the cost of maneuvers and station-keeping. The optimization techniques applied in this investigation operate as means for cost reduction in both mission support as well as mission operating costs. Primarily, the techniques are used to decrease the cost of satellite maneuvers by finding an optimal path for the satellite to fly with minimal fuel expenditure. The tools showcased in this investigation also aid the mission designer when operating at thrust accelerations without viable heuristic approximations or in complex dynamical environments.

1.2 Research Objectives

The fundamental question of this investigation is how to design minimum-fuel trajectories when an analytical approximation is unavailable and/or when the dynamical environment is complex. The process of designing an optimal trajectory, regardless of discipline, typically requires some sort of optimization technique. When the trajectory is to be designed in a nonlinear environment, as is the case for most problems designed in a space environment, an initial guess must be given to the chosen optimization scheme. The accuracy of the initial guess can dictate whether or not the optimal solution is found or if one is found at all. Particle swarm optimization (PSO) belongs to a class of optimizers called evolutionary algorithms that are unique in that they mimic natural behavior. They have the advantage of a more global search of the problem design space and do not require an initial guess. This investigation offers PSO as a viable and effective technique for generating an initial guess when one is not readily available. The initial guess is subsequently given to a more robust nonlinear programming (NLP) solver for improvement. There are two problems investigated to demonstrate the efficacy of this design strategy.

1. Low-thrust, fuel-optimal, continuous and multiple-burn transfers from a coplanar and non-coplanar low-Earth-orbit to a geostationary orbit designed in a two-body dynamical model with and without oblate Earth effects
2. Impulsive transfer from near-Earth to a periodic orbit about the Earth-Moon cislunar (between Earth and Moon) collinear Lagrange point with minimized burn costs designed in a multi-body dynamical environment

The first set of test cases is modeled in a restricted two-body (Earth-satellite) system with and without oblate Earth perturbation effects. The transfers are propagated at thrust acceleration levels where the literature reviewed in this investigation lacks analytical optimal solutions for an initial guess. Therefore, PSO is used to generate the initial guess that is provided to a more robust nonlinear optimizer.

The second case utilizes a circular restricted three-body system as well as invariant manifold dynamics. Due to the complexity of the dynamical environment and existence of chaotic (sensitive to initial conditions) regions of the phase space in the circular restricted three-body problem (CR3BP), a closed-form analytical solution and thereby an initial guess is unavailable. However, analytic work is used to provide insight into the problem so that the search space is more efficiently bounded, computation time is reduced, and convergence of the PSO algorithm is enhanced. The PSO-generated initial guess (PSOIG) is then improved via a nonlinear programming solver.

1.3 Decision Tree for Trajectory Optimization Techniques

When designing and optimizing spacecraft trajectories, the choice of techniques to employ depends on multiple factors. Such factors that affect the decisions in the current investigation are the dynamical model, the thrust level, the burn profile, the

transcription method, and the solving algorithm. Many of the decisions in the current work are motivated by whether or not an initial guess is readily available. Figure 1 provides a graphical depiction of the decision tree for this investigation.

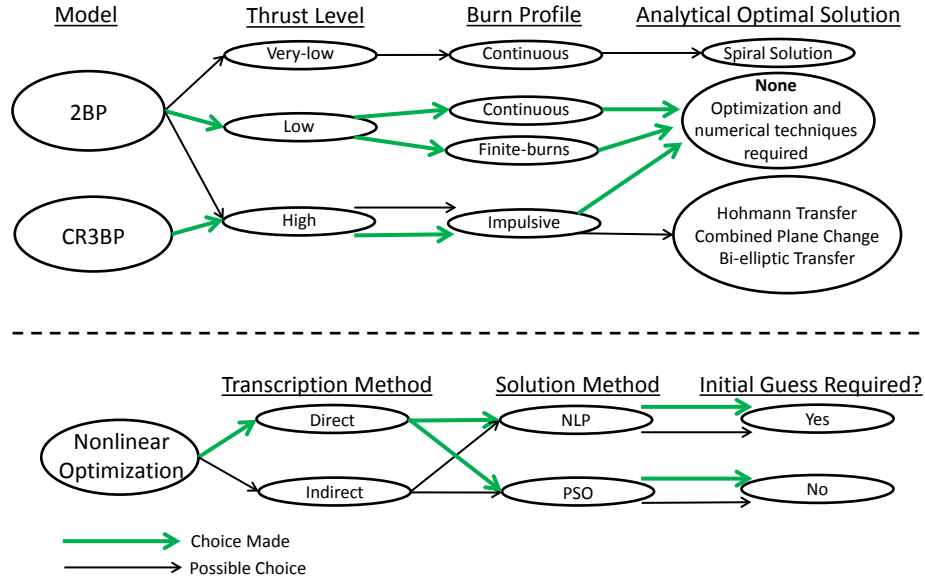


Figure 1. Decision tree for space trajectory optimization in the current investigation

In the decision tree, the green arrows denote decision routes that are taken in the current investigation. Even though the green paths are the ones taken, the black paths are also viable depending on the circumstances. The specific rationale behind each choice is elucidated in future discussions. However, the decision tree is offered to provide context for the current design approach within neighboring options. The relevant literature discussing many of the possible paths is offered in Section 2.5. The choice to offer the previous works at the end of Chapter 2 is motivated by a desire to present the background theory first in order to understand the significance of each previous contribution.

1.4 Document Preview and Special Notes

The body of this investigation attempts to demonstrate the flexibility and efficacy of using PSO as a method for generating a useful initial guess for minimum-fuel spacecraft trajectories, especially when one is not available. To do this, multiple test cases are devised with different constraints and in multiple dynamical environments. The relevant background, methodology, results, and analysis are organized as follows:

- Chapter 2: A derivation of the two-body problem (2BP) as well as a description of different coordinates used in this investigation are provided. The circular-restricted three body problem and relevant design tools are then introduced along with dynamical systems theory. Summaries of parameter and functional optimization are given, followed by descriptions of the optimization methods utilized in the design process. The fundamentals of propulsion are summarized, and a synopsis of previous contributions is given.
- Chapter 3: A polynomial-based approach is used to parameterize the control time histories of low-thrust transfers in the two-body problem. PSO is used to optimize the polynomial coefficients of a sequence of polynomials that approximate the optimal control time histories of the trajectories. The initial guesses are then given to a nonlinear programming algorithm for improvement. The chapter begins by detailing the methodology for generating the initial guesses as well as improving them. Next, inclination changes, variable burn profiles, and oblate Earth effects are factored into the design scenarios, and the results are presented. The chapter concludes with a discussion of the feasibility of the trajectories as well as how PSO performed.
- Chapter 4: PSO is used to determine the optimal time, magnitude, and direction of an impulsive burn to target an insertion point on a libration point orbit's

stable manifold trajectory. A range of insertion points is targeted by PSO to provide multiple initial guesses for improvement. Once on the stable manifold trajectory, the satellite coasts until a final burn is made to enter the desired libration point orbit (LPO). The design methodology is given at the beginning of the chapter followed by the trajectory design and results. A discussion of the final trajectory as well as the utility of PSO is given to conclude the chapter.

- Chapter 5: A discussion of the results, limitations of the current investigation, recommendations for future work, and concluding remarks are given.

As a special note, Chapters 3 and 4 are organized to contain their specific test case methodologies as well as their respective results. This choice is made to prevent confusion when transitioning between different dynamical models. Also, depending on the dynamical environment, the design approaches and processes applied vary and necessitate independent attention. It is important to state that Chapters 3 and 4 are not independent of each other, but they offer two scenarios in which PSO may be useful to the mission designer.

Before continuing, it is also important to highlight that the chosen propulsive specifications used in the low-thrust scenarios are on the upper fringes of what is possible with current technology. This choice is made to create a particularly unique and difficult dynamical environment. Any resulting low-thrust trajectories should be treated as future potential design options when the required propulsive technologies are more readily available.

A final area for special attention directly relates to the dynamical model being used for trajectory design. Nondimensionalized units are utilized throughout this investigation for numerical efficiency as well as greater quantitative intuition when designing. The characteristic quantities vary based on the dynamical model in use. Therefore, attention should be given to the specific dynamical model as well as how

the nondimensionalization for each dynamical model is conducted. The parameter defined as μ also varies depending on the dynamical model. In the context of the restricted two-body problem, μ is the gravitational parameter of the Earth. For the Earth-Moon CR3BP, μ is a mass ratio that depends on the mass of the Earth and the Moon. Due to this difference, attention to the dynamical model in focus is important.

1.5 Chapter Summary

This chapter introduced the problem, motivated the current investigation, and summarized the organization of this document. The next chapter details the necessary theory and context of the present investigation.

2. Background and Literature Review

The background theory and context for the present investigation are presented in this chapter. First, the two-body problem as well as necessary coordinates are derived and presented for use in the low-thrust transfers. Next, the circular restricted three-body problem and relevant design tools are introduced for direct application in the high-altitude trajectory design. The optimization techniques that are employed for trajectory design are explained in the optimization section. Finally, the fundamentals of satellite propulsion as well as relevant works in literature are given. Overall, this chapter provides the theory behind the methodologies used in the next two chapters as well as the scholarly research context of the present investigation.

2.1 The Two-Body Problem

In order to demonstrate the flexibility and efficacy of the methods employed in this investigation, the initial investigations are conducted in a simple, but relevant, dynamical model for satellite trajectories, the restricted two-body problem with perturbing accelerations caused by an oblate Earth. This simple model is used before transitioning into a more complex multi-body dynamical environment, specifically, the circular restricted three-body problem.

2.1.1 Historical Context

The 2BP was originally devised in order to explain the motion of celestial bodies. However, it was not the first model used to explain planetary motion. Claudius Ptolemy in the second century A.D. is often credited with one of the first attempts at explaining celestial motion. His “Ptolemaic scheme” is centered near the Earth with the planets revolving in a large circle called the *deferent* and with smaller circles

called *epicycles* [18]. It was not until Nicholas Copernicus in 1543 that a heliocentric model was seriously proposed [18]. The Copernican system is a rearrangement of the “Ptolemaic scheme” and matches the motion of the planets with greater accuracy due to the transition to a sun-centered system. In 1609, Galileo’s improvements on existing telescopes allowed him to take astronomical measurements and observations precise enough to confirm the validity of a heliocentric solar system [19]. Johannes Kepler is another significant contributor to describing the motion of the planets. By working under the most skilled astronomical observer at the time, Tycho Brahe, Kepler had access to Brahe’s reliable observational data and, during the early 17th century, was able to devise his three laws of planetary motion [20]. The laws state that [2]:

1. The motion of the planets trace out ellipses with the sun located at a focus.
2. The line drawn from the sun to a planet sweeps out equal area in equal time.
3. The square of the period of a planet is proportional to the cube of the semimajor axis of the orbit.

Isaac Newton published his *Principia Mathematica Philosophia Naturalis* in 1687. The document contains many of the discoveries necessary to answer the question of *why* the planets move according to Kepler’s laws [21]. In it, Newton presented his three laws of motion:

1. A particle at rest remains at rest, and a particle in motion remains in motion unless the net force acting on the particle is non-zero.
2. The force exerted on a particle is equal to the time rate of change of its momentum.
3. For every force there is always an equal and opposite reaction force.

The three laws of motion sufficiently describe motion for most everyday occurrences, but they cannot singularly explain the motion of stars and planets. It is Newton's law of universal gravitation that provides another fundamental axiom necessary to explain the motion of celestial bodies. It states that two objects exert an attractive force on each other that is proportional to the product of the masses and is inversely proportional to the square of the distance between them [1]. In vector notation form, it is expressed in equation (1)

$$\mathbf{F}_g = \frac{Gm_1m_2}{r^2}\hat{r} \quad (1)$$

where G is the universal gravitational constant, m_1 and m_2 are the masses of the two bodies in attraction, and r is the distance between the two bodies. In the current investigation, bold symbols denote vector quantities and the $\hat{}$ symbol is used to denote vectors of unit length. Newton's universal law of gravitation serves as a starting point for solving the problem of two bodies.

2.1.2 Two-Body Derivation

The following two-body problem derivation is modeled after Section 2.2 in Wiesel as well as Section 1.3 in Bate, Mueller, and White [1, 18]. Just as the name implies, in the 2BP, there are only two bodies taken into consideration. It is assumed that the system is closed and that the bodies are not affected by any external forces and the only internal force allowed is the gravitational force of attraction [1]. Also, the two bodies are assumed to have spherically symmetric gravity fields and center of masses located at their respective geometric centers [1]. These assumptions do not prohibit one mass from being larger than the other or even equal in size as in a binary star system. In order to model the relative motion of two bodies due to gravitational attraction, a reference frame must be chosen. It is important to choose an inertial

reference frame, or one that is not accelerating. A nuance here is that the reference frame can have constant velocity without violating the requirement. In this model, the inertial reference frame of choice is a Cartesian, orthonormal triad denoted with the symbols $\hat{X}$, $\hat{Y}$, and $\hat{Z}$.

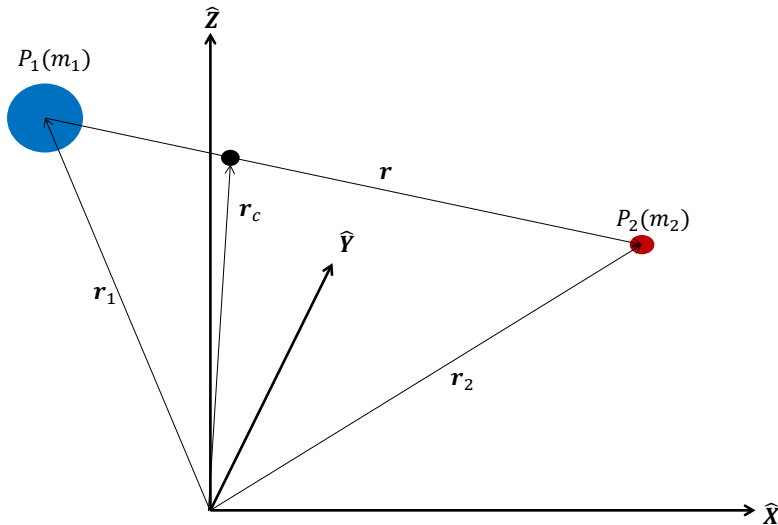


Figure 2. Two-body motion in an inertial reference frame

In Figure 2, $\mathbf{r}_c$ is the vector from the origin to the system center of mass. Also, $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$. The positions, P_1 and P_2 , define the locations of m_1 and m_2 , respectively. Applying Newton's second law and the universal law of gravitation to both P_1 and P_2 yields equations (2a) and (2b).

$$m_1 \ddot{\mathbf{r}}_1 = \frac{Gm_1 m_2}{r^2} \hat{\mathbf{r}} \quad (2a)$$

$$m_2 \ddot{\mathbf{r}}_2 = \frac{-Gm_1 m_2}{r^2} \hat{\mathbf{r}} \quad (2b)$$

Equations (2a) and (2b), written in component form, represent six second-order, non-linear, coupled ordinary differential equations, thus, twelve constants of the motion are necessary to solve the system [18]. Adding equations (2a) and (2b) together results in equation (3).

$$m_1\ddot{\mathbf{r}}_1 + m_2\ddot{\mathbf{r}}_2 = 0 \quad (3)$$

The next step is to define the vector from the origin to the center of mass of the system in terms of $\mathbf{r}_1$ and $\mathbf{r}_2$.

$$\mathbf{r}_c = \frac{m_1\mathbf{r}_1 + m_2\mathbf{r}_2}{m_1 + m_2} \quad (4)$$

Substituting equation (4) into (3) and simplifying yields equation (5).

$$\ddot{\mathbf{r}}_c = 0 \quad (5)$$

Integrating equation (5) produces six arbitrary constants or half that is necessary to solve the system defined by equations (2a) and (2b). A more physical interpretation of equation (5) is that the center of mass of the system is non-accelerating, or it moves linearly with constant velocity [18].

The next step requires contextual knowledge of the N -body problem. The N -body problem is much like the problem formulated in this section; however, instead of two bodies attracted gravitationally, N bodies interact with each other. For the N -body problem, a total of $6N$ integrals of the motion are required to solve the system. For the two-body problem, $N = 2$, therefore, twelve integrals of the motion are required. Regardless of the value for N , there are only ten known integrals of the motion in the N -body problem. The first six have already been demonstrated via the conservation of linear momentum of the system's center of mass. The next three are due to the conservation of angular momentum, with the last integral of the motion coming from the conservation of the system mechanical energy. In the present derivation, if a different approach is not taken, the end result would be a total of ten integrals of the motion found when a total of twelve are needed. Thus, a reduction to a relative 2BP

is conducted.

Instead of solving for the motion of both masses, now the focus is placed on defining the motion of the relative vector between both masses, $\mathbf{r}$. To solve for the relative motion between the two bodies, divide (2a) by m_1 and (2b) by m_2 then subtract equation (2a) from (2b). The result is

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \quad (6)$$

where μ is equal to $G(m_1 + m_2)$. Since this investigation is concerned with the trajectories of artificial satellites whose masses generally pale in comparison to the mass of the Earth, it is acceptable to simplify to $\mu = G(m_1)$. When making this assumption, that $m_2 \ll m_1$, the model is considered to be *restricted*. According to Wiesel, μ is used in place of G or m_1 not only for concise notation, but also because μ is much easier to calculate to a desirable precision. “The problem lies in the fact the G can only be measured in exceedingly delicate laboratory experiments with known masses, while the product μ can be determined by accurate tracking of earth satellites” [18]. It is important to note that this definition of μ differs from the definition used in sections where a third body is considered; therefore, it is extremely important to take note of which model is being used for trajectory design.

Equation (6) defines the relative 2BP. Employing the current reduction generated a system that requires six integrals of the motion; however, the previous six that were found as a result of the conservation of linear momentum no longer apply. The equations of motion (EOMs) defined in equation (6), as they stand, can be used to propagate the trajectory of a satellite in the 2BP. The next steps are taken to gain more insight via the remaining constants of the motion. Also, six constants are still needed in order to consider the system solved.

From physics, the radial gravity field from Newton’s universal law of gravitation

is conservative, meaning an energy integral is conserved. To prove this, dot equation (6) with $\dot{\mathbf{r}}$.

$$\dot{\mathbf{r}} \cdot \ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \cdot \dot{\mathbf{r}} \quad (7)$$

Consider for simplicity in the next few steps the restricted 2BP. The left-hand side of the equation is equal to the time rate of change of specific kinetic energy where specific means per unit mass of the spacecraft (m_2) [18].

$$\frac{\dot{T}}{m_2} = \frac{1}{2} \frac{d}{dt} [\dot{\mathbf{r}} \cdot \dot{\mathbf{r}}] = \frac{1}{2} [\dot{\mathbf{r}} \cdot \ddot{\mathbf{r}} + \ddot{\mathbf{r}} \cdot \dot{\mathbf{r}}] = \dot{\mathbf{r}} \cdot \ddot{\mathbf{r}} \quad (8)$$

In equation (8), T is the kinetic energy of the spacecraft. The right-hand side of equation (7) requires more manipulation. This dot product is equal to the magnitude of $\mathbf{r}$ times the projection of the velocity vector ($\dot{\mathbf{r}} = \mathbf{v}$) in the radial direction. The projection is the radial velocity, which is equal to the time rate of change of the magnitude of $\mathbf{r}$ [18]. Therefore, the right-hand side of equation (7) becomes

$$-\frac{\mu}{r^3} r \dot{r} = -\frac{\mu}{r^2} \dot{r} \quad (9)$$

The right-hand of equation (9) is the perfect time derivative of the specific potential energy where V is the potential energy [18].

$$-\frac{d}{dt} \left(-\frac{\mu}{r} \right) = -\frac{\mu}{r^2} \dot{r} = -\frac{\dot{V}}{m_2} \quad (10)$$

So, equation (7), after substituting in the new expressions on both sides, becomes

$$\frac{\dot{T}}{m_2} = -\frac{\dot{V}}{m_2} \quad (11)$$

Integrating equation (11) is the final step to show that total energy is conserved.

$$\varepsilon = \frac{T}{m_2} + \frac{V}{m_2} \quad (12)$$

$$\varepsilon = \frac{1}{2}v^2 - \frac{\mu}{r} \quad (13)$$

In equations (12) and (13), the integration constant, ε , is equal to specific mechanical energy and is a constant of the motion. Specific mechanical energy is equal to the sum of specific kinetic energy and specific potential energy. Substituting in the expressions for specific kinetic energy and specific potential energy yields the “vis-viva” equation in equation (13). Since the rate of change of the specific kinetic energy is equal to the negative rate of change of the specific potential energy, if the satellite increases in radius, it must consequently slow down to satisfy conservation of energy. The opposite is also true. This type of insight offers potential heuristics to employ when modeling in the two-body problem. Specific mechanical energy is the first of six required integrals of the motion to solve the relative 2BP.

To generalize this conclusion, in the general 2BP, the “specific” terms are per unit mass of a reduced mass. The reduced mass, m_{red} , is defined as

$$m_{red} = \frac{m_1 m_2}{m_1 + m_2} \quad (14)$$

When making the assumption in the restricted 2BP, that $m_2 \ll m_1$, the reduced mass approximately equals m_2 .

The next conserved quantity is angular momentum. To prove this, first, cross equation (6) with $\mathbf{r}$.

$$\mathbf{r} \times \ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \times \mathbf{r} = \mathbf{0} \quad (15)$$

Equation (15) is equal to a zero vector because a vector is being crossed with itself.

The left-hand side of equation (15) is equal to the time derivative of $\mathbf{r} \times \dot{\mathbf{r}}$. This is a similar operation to the step taken in equation (8) via the chain rule. Integrating $\frac{d}{dt}(\mathbf{r} \times \dot{\mathbf{r}}) = \mathbf{0}$ yields the next integral of the motion [18].

$$\mathbf{r} \times \dot{\mathbf{r}} = \mathbf{H} \quad (16)$$

Knowing that the cross-product produces a vector orthogonal to the radius and velocity vectors, the specific angular momentum, $\mathbf{H}$, always points out of the orbital plane and in a fixed direction. Thus, the motion of the satellite is confined to a plane. Also, since the specific angular momentum is a three dimensional vector, three additional constants of the motion are identified. The remaining two constants are necessary to define the shape and orientation of the orbit. At the current step, four out of the six required constants of the motion to solve the relative 2BP have been defined.

According to Wiesel, there have been many “ad hoc” approaches to extracting the shape and orientation information from the EOMs [18]. Section 2.4 in Wiesel, Section 1.5 in Bate, Mueller, and White, and Section 3.4 in Chobotov provide, complete derivations [1, 2, 18]. The first step is to cross both sides of equation (6) with $\mathbf{H}$.

$$\ddot{\mathbf{r}} \times \mathbf{H} = -\frac{\mu}{r^3} \mathbf{r} \times \mathbf{H} \quad (17)$$

Using the vector identity, $\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = \mathbf{B}(\mathbf{A} \cdot \mathbf{C}) - \mathbf{C}(\mathbf{A} \cdot \mathbf{B})$, and expanding the right side yields

$$\begin{aligned} \mathbf{r} \times \mathbf{H} &= \mathbf{r} \times (\mathbf{r} \times \dot{\mathbf{r}}) \\ \mathbf{r} \times \mathbf{H} &= \mathbf{r}(\mathbf{r} \cdot \dot{\mathbf{r}}) - \dot{\mathbf{r}}(\mathbf{r} \cdot \mathbf{r}) \end{aligned} \quad (18)$$

Since $\mathbf{r} \cdot \mathbf{r} = r^2$ and $\mathbf{r} \cdot \dot{\mathbf{r}} = r\dot{r}$, equation (17) can be rewritten as shown in equation (19) [18].

$$\frac{d}{dt}(\dot{\mathbf{r}} \times \mathbf{H}) = \frac{\mu}{r}\dot{\mathbf{r}} - \frac{\mu\dot{r}}{r^2}\mathbf{r} \quad (19)$$

The right-hand side of the previous equation is equal to the time derivative of $\mu\mathbf{r}/r$. Applying this knowledge results in a form of the equation that can be integrated.

$$\frac{d}{dt}(\dot{\mathbf{r}} \times \mathbf{H} - \mu\frac{\mathbf{r}}{r}) = \mathbf{0} \quad (20)$$

The final step to prove the next integral of the motion is integrating equation (20) to yield the equation below.

$$\dot{\mathbf{r}} \times \mathbf{H} - \mu\frac{\mathbf{r}}{r} = \mu\mathbf{e} \quad (21)$$

The eccentricity vector, $\mathbf{e}$, is a constant vector that lies in the plane of the orbit. Within the orbit plane, $\mathbf{e}$ points to *perigee* of the orbit or the point of closest approach to the Earth. Conversely, the term *apogee* is defined as the point of furthest distance from the Earth. The fact that $\mathbf{e}$ must point within the plane of the orbit means that it provides two, and only two, additional constants of the motion. However, since four have already been defined and six are required to solve the relative 2BP, with the addition of the constant vector, $\mathbf{e}$, the problem is considered to be solved. The most satisfying conclusion of the relative 2BP requires more manipulation, where an additional dot product of both sides of equation (21) with $\mathbf{r}$ is taken and the angle ν between $\mathbf{e}$ and $\mathbf{r}$ is introduced. The resulting orbit equation is

$$r = \frac{H^2}{\mu(1 + e\cos\nu)} \quad (22)$$

where e is the scalar eccentricity of the orbit, and ν is called the *true anomaly*. True anomaly is defined as the angle from $\mathbf{e}$ to $\mathbf{r}$ in the direction of motion or following the “right-hand-rule” with the thumb pointing along $\mathbf{H}$. Equation (22) provides the

magnitude of $\mathbf{r}$ in terms of an angle corresponding to the position of the satellite and other constants (H, e, μ). The magnitude of e provides information about the shape of the orbit, which is discussed in the next section. The satisfying conclusion is the not so obvious realization that equation (22) is the polar form for a conic section [18]. That is, all orbits in the 2BP must be one of the five conic sections: circle, ellipse, parabola, hyperbola, or a line.

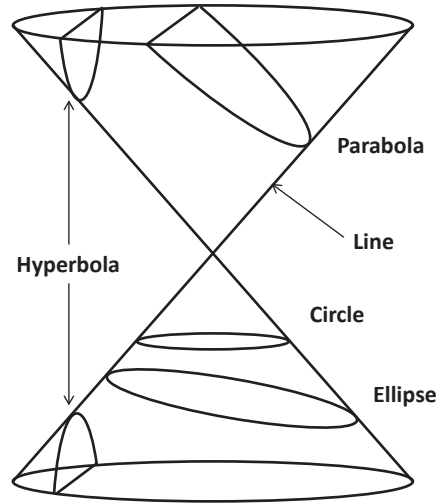


Figure 3. Conic sections, adapted from Bate, Mueller, and White [1]

2.1.3 Classical Orbital Elements

For most near-Earth satellite applications, the orbits are either circular or elliptical. The Earth, being the primary body and modeled as a point mass, is located at the center of the circle or at one focus of the ellipse. Since geometric parameters are useful in visualizing satellite orbits, the classical orbital elements (COEs) are now introduced.

There are six standard COEs used to define the size, shape, and orientation of a satellite's orbit as well as the current satellite position in that orbit. Before defining the elements, a convenient inertial reference frame is first introduced. With the

satellite orbiting the Earth, a geocentric-equatorial coordinate system is often chosen. Instead of $\hat{X}$, $\hat{Y}$, and $\hat{Z}$ used in the previous derivation as the arbitrary inertial reference frame, $\hat{I}$, $\hat{J}$, and $\hat{K}$ is now used. The origin of the frame is centered at the geometric center of the “spherical” earth. The $\hat{I}$ axis points in the vernal equinox direction, and the $\hat{K}$ axis points to the North Pole [2]. The vernal equinox is considered to be a sufficiently inertial direction (although it drifts slowly over time) and points toward the first point of the Aries constellation. To be precise, the exact direction points to where the first point of Aries was in 4,000 B.C. The vernal equinox direction is also equivalently, and more practically, defined as the vector from the Earth to the Sun on the first day of Spring in the Northern Hemisphere. Lastly, the $\hat{J}$ axis completes the right-handed triad. See Figure 5 for a visual representation of the inertial reference frame.

The first orbital element known as the semimajor axis, a , defines the size of the orbit. For a circle, it is equal to the radius, and for an ellipse, it is equal to half the length of the major or longest axis. The semiminor axis, b , appears frequently, but is not one of the COEs. It is equal to a in a circle and is half the length of the minor axis of an ellipse. Lastly, the semilatus rectum, p , is another important parameter and is depicted in Figure 4.

In describing the motion of satellites, it is helpful to define a few additional terms. The period of an orbit in terms of the semimajor axis is given by the expression

$$\text{Period} = 2\pi\sqrt{\frac{a^3}{\mu}} \quad (23)$$

The equation makes intuitive sense because one orbit spans 2π radians and because the mean motion, n , about the orbit is given by equation (24).

$$n = \sqrt{\frac{\mu}{a^3}} \quad (24)$$

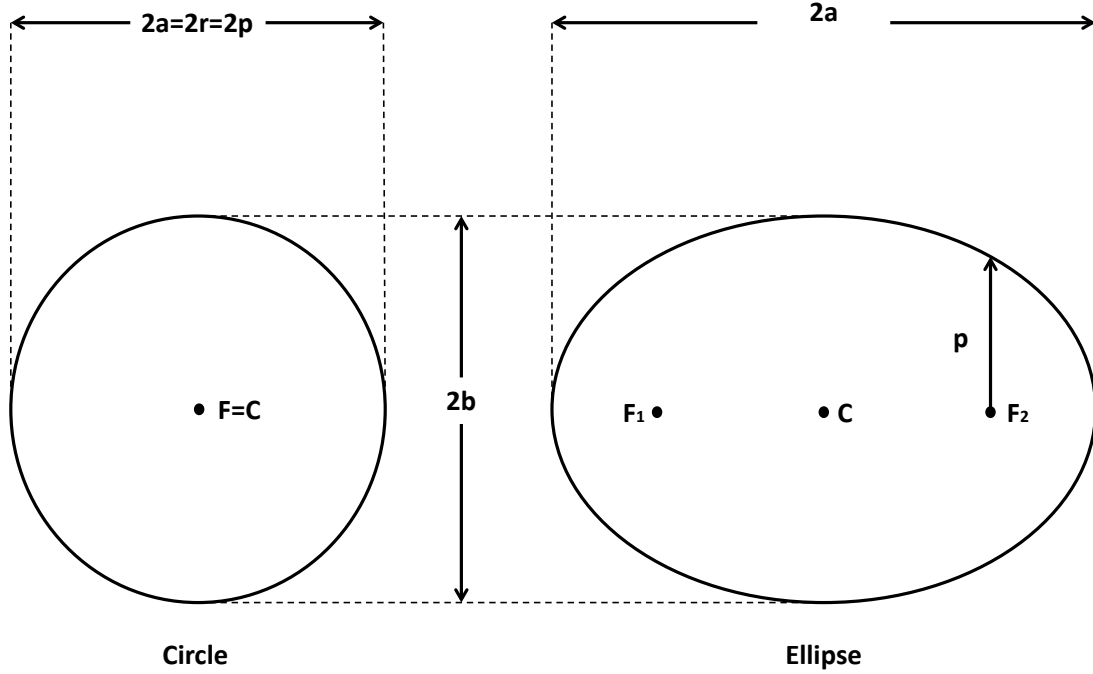


Figure 4. Orbit size, adapted from Bate, Mueller, and White [1]

Multiplying the mean angular velocity or mean motion by the angular span of an orbit logically yields the period of one revolution. Even though the satellite's angular velocity is, in general, variable (constant in circular orbits), the mean motion of the spacecraft can be used to calculate the period for one revolution.

The second orbital element already introduced is the eccentricity, e , which is equal to the magnitude of the eccentricity vector $\mathbf{e}$. It defines the shape of the orbit and is a dimensionless parameter. The table below provides ranges of eccentricity and semimajor axis values and their associated conic sections [2].

Table 1. Parameter ranges for conic sections

Semimajor axis	Eccentricity	Conic Section
$a = r$	0	Circle
$a > 0$	$0 < e \leq 1$	Ellipse
$a \rightarrow \infty$	1	Parabola
$a < 0$	$e \geq 1$	Hyperbola
$a(\varepsilon)$, a is a function of ε	1	Line: Degenerate Ellipse, Parabola, or Hyperbola

The polar form of a conic section in equation (22) can also be written in terms of the semilatus rectum and COEs by the expressions below.

$$r = \frac{p}{1 + e \cos \nu} \quad (25a)$$

$$r = \frac{a(1 - e^2)}{1 + e \cos \nu} \quad (25b)$$

An important note is that equation (25a) applies to circles, ellipses, parabolas, and hyperbolas whereas equation (25b) applies to circles, ellipses, and hyperbolas. These equations are useful in that they provide a relationship between the radius of the satellite's orbit and COEs.

The next three COEs define the orientation of the orbit plane. Inclination, i , can be thought of as the tilt of the orbit plane. It is the angle measured from $\mathbf{K}$ to $\mathbf{H}$ [1]. Below are the varying orbit types based on inclination [2].

Table 2. Orbit inclinations

Range	Orbit Type
$i = 0, 180$	Equatorial
$i = 90$	Polar
$0 \leq i < 90$	Prograde Motion
$90 < i \leq 180$	Retrograde Motion

The right ascension of the ascending node, Ω , is defined as the angle between $\mathbf{I}$ and $\mathbf{n}$, where $\mathbf{n}$ is the nodal vector that points to the ascending node. The ascending node is the point where the satellite makes a south-to-north crossing of the equatorial plane [2]. Ω can be thought of as the “swivel” of the orbit and ranges from 0 to less than 360 degrees.

The argument of perigee, ω , is measured from $\mathbf{n}$ to $\mathbf{e}$. It is the angle that orients the perigee of the orbit within the orbit plane [2]. ω ranges from 0 to less than 360 degrees.

Lastly, true anomaly, ν , is measured from $\mathbf{e}$ to the current position, $\mathbf{r}$, of the satellite. True anomaly can also range from 0 to less than 360 degrees. True anomaly is the only COE that is time varying in the 2BP.

With all six COEs, it is possible to completely define the motion of the satellite. There are additional elements available for degenerate cases such as when $i = 0$ or $e = 0$; however, they are not introduced here. Section 2.3 of Bate, Mueller, and White is an excellent reference for the alternate orbital elements [1].

There are a multitude of element sets that can be used depending on the application. In addition to rectangular and polar elements, there are also equinoctial elements that are formulated such that existing singularities are relocated to more convenient locations [18].

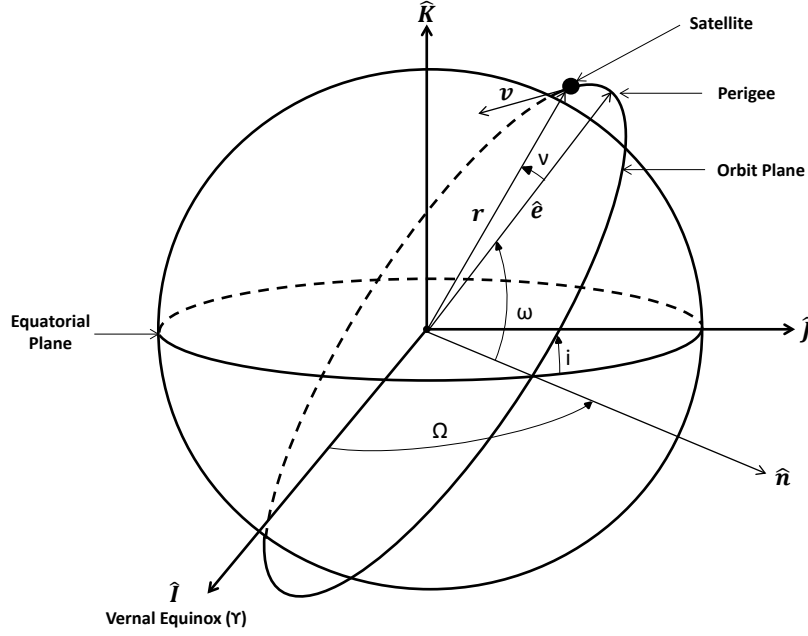


Figure 5. COEs in Earth-centered inertial frame, adapted from Chobotov [2]

Modified equinoctial elements are used in this investigation and are discussed in the next section.

2.1.4 Modified Equinoctial Elements

When implementing a perturbation into the 2BP or any dynamical model for spacecraft trajectories, it is worth using coordinates that emphasize the effects of the disturbing acceleration [18]. This is because, in many cases, the disturbing acceleration due to the perturbing force pales in comparison to the more dominant accelerations such as the gravitational acceleration toward a nearby massive body. In the 2BP, using the Lagrange planetary equations de-emphasizes the dominant two-body motion and allows perturbations such as the Earth's oblateness effects to be more accurately modeled. A potential issue with using the Lagrange Planetary equations is the existence of singularities at low eccentricities and inclinations [18]. Since the final orbit used for the 2BP test cases in the next chapter is circular and equatorial, choosing a slightly different set of coordinates is necessary. Modified equinoctial

elements (MEEs) remove the eccentricity singularity and relocate the inclination singularity to 180 degrees. An inclination of 180 degrees defines a retrograde equatorial orbit, which is not commonly used. The derivation of the MEEs can be found in Walker with the final equations displayed below [22].

$$\begin{aligned}
p &= a(1 - e^2) \\
f &= e \cos(\omega + \Omega) \\
g &= e \sin(\omega + \Omega) \\
h &= \tan\left(\frac{i}{2}\right) \cos(\Omega) \\
k &= \tan\left(\frac{i}{2}\right) \sin(\Omega) \\
L &= \Omega + \omega + \nu
\end{aligned} \tag{26}$$

The equations in (26) serve to replace the COEs when oblate Earth effects are incorporated into the dynamical model. The next section discusses methods for including perturbations in a dynamical model.

2.1.5 Perturbation Methods and Accounting for Oblate Earth Effects

For greater model accuracy, perturbing accelerations can be included to account for additional forces that the spacecraft may encounter. Common perturbing accelerations that are included in spacecraft trajectory models are drag due to the Earth's atmosphere, gravitational pull from additional celestial bodies, deformities in the initially assumed spherical gravity potential of the Earth, and solar radiation pressure. In addition, propulsive acceleration due to low-thrust on a spacecraft can also be modeled as a perturbation.

In practice, there are two approaches to modeling perturbations on a satellite. A general perturbations method analytically models the perturbing forces and includes

them in the EOMs such that they are still integrable. General perturbations theory relies on the assumption that the perturbing accelerations are small compared to the dominant forces in the model [23]. Conversely, a special perturbations method incorporates the perturbations into the EOMs and uses numerical integration to solve for particular solutions. In one sense, special perturbations is synonymous with numerical integration of the EOMs. The assumption of small perturbing forces is not necessary for a special perturbations method, but as stated in the previous section, it is often helpful to choose coordinates that emphasize the effects of the perturbations. Special perturbations is a useful and often a necessary approach when modeling in a dynamical environment that does not have a closed-form analytical solution or when including perturbations that are not sufficiently small for a general perturbations method.

In the current investigation, a special perturbations method is used to include perturbations such as low-thrust acceleration and the oblate Earth effects into the restricted 2BP model when conducting low-thrust trajectory design. The perturbations are small enough such that perturbing a reference 2BP solution via numerical integration sufficiently approximates the motion of the spacecraft, but large enough that a general perturbations method is not used. In transitioning to trajectory design in the CR3BP, a special perturbations method is used; however, the restricted 2BP no longer provides a sufficient reference solution because the third body perturbations are more significant at the super-GEO altitudes used for the trajectory design. This motivates the inclusion of third body gravitational effects into the EOMs to formulate the CR3BP dynamical environment that has no closed-form analytical solution. As such, numerical integration, or special perturbations, is required, but the third body effects are explicitly included in the reference solution. This is different from the low-thrust cases that use a restricted 2BP reference solution that is integrable (not

requiring numerical integration) when unperturbed.

For higher fidelity in modeling the gravity of the Earth, deviations from a spherically symmetric gravity potential can be included. Rather than only including the assumed Newtonian point mass potential, $V = -\mu/r$, the Earth's gravity potential can be expanded in a complete summation of geopotential terms defined as

$$V = -\frac{\mu}{r} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{R_{\oplus}}{r} \right)^n P_n^m \sin(\delta) [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \quad (27)$$

where $R_{\oplus}$ is the equatorial radius of the Earth, λ is the longitude and δ is the latitude. The P_n^m terms are the associated Legendre polynomials, and the C_{nm} and S_{nm} coefficients are specific to the gravity model [18]. In the current investigation, the oblateness of the Earth is only considered because it is the largest deviation in the Earth's actual geopotential from the Newtonian point mass potential. Any additional terms are smaller in magnitude, but can be included to add fidelity to the model. Physically, the Earth's oblateness is the extra mass or “bulge” that exists at the equator. In equation (27), the C_{20} coefficient accounts for the equatorial bulge and is related to J_2 . J_2 is the dimensionless parameter that is used when approximating the oblateness effects in a special perturbations method [18]. In many sources, the term J_2 is often synonymous with oblate Earth perturbative effects. Additional information on perturbation theory and the expansion of the Earth's geopotential can be found in Wiesel [23].

In a simple understanding of the secular effects of J_2 over time, the J_2 perturbation causes a regression of the right ascension of the ascending node and an advance of perigee. This behavior is evident in the expressions in equation (28).

$$\begin{aligned}\dot{\Omega} &= -\frac{3nJ_2R_{\oplus}^2}{2a^2(1-e^2)^2}\cos(i) \\ \dot{\omega} &= -\frac{3nJ_2R_{\oplus}^2}{2a^2(1-e^2)^2}\left[\frac{5}{2}\sin^2(i) - 2\right]\end{aligned}\tag{28}$$

In the expressions, J_2 is equal to 0.001082 for the Earth. When using COEs and a special perturbations method, it is straightforward to include the effects of J_2 during the numerical integration process. In Chapter 3, a similar approach is taken, but new expressions are introduced for use with MEEs. The next section provides the theory and derivations for a circular restricted three-body model.

2.2 The Three-Body Problem

After deriving the solution to the motion of two-bodies, it may seem that celestial motion can be captured by simple and elegant solutions such as conics. In reality, there are an “infinite” number of bodies in the universe all interacting gravitationally. For applications in the vicinity of the Earth, the gravitational pull of the planets in the solar system are either considered trivial or, for higher fidelity, modeled as a “small” perturbing force. Even the gravitational pulls of the Sun and the Moon are “small” for near-Earth applications. For trajectories above geosynchronous altitude or GEO (approximately 35,786 km altitude), but within the vicinity of the Earth-Moon system, it is not necessarily sufficient to only consider the gravitational pull of the Earth with the Moon’s gravity modeled as a “small” perturbation. Doing so would assume that the reference solution of a perturbed conic section is a “good enough” initial guess for the motion of the spacecraft. At altitudes higher than GEO, the Moon’s gravity begins to play a more important role for one to accurately describe the motion of the satellite. As such, analytically incorporating the Moon’s gravity into a higher-fidelity, multi-body dynamical model, as done in the CR3BP, provides a more accurate initial guess of the spacecraft’s motion. The CR3BP presents a very

complex and interesting problem as well as greater accuracy for super-GEO altitude transfers and a much wider range of possible motion.

Two-body motion is considered to be virtually the only problem in celestial mechanics with a closed-form analytical solution [18]. Thus, increasing fidelity to a three-body model is not a simple undertaking. However, techniques have been developed to gain insight into the problem and facilitate trajectory design in a circular restricted three-body dynamical environment. A motivation for expending the effort required to transition into a three-body model is the benefit of a vast design space that boasts a wide range of possible motion. This affords the mission designer not only greater quantitative accuracy at high altitudes but also greater freedom to explore additional possible qualitative behavior for trajectory design.

2.2.1 The Circular Restricted Three-Body Problem

The following discussion attempts to introduce the CR3BP as well as useful design tools, but it is not an extensive coverage. Szebehely's seminal work, *Theory of Orbits*, provides most of the foundation behind the following discussion and can also be referenced for additional information on the CR3BP [24]. Other helpful references include Roy [25] and Murray and Dermott [26].

In a similar fashion to the 2BP derivation, starting with Newton's universal law of gravitation is most convenient. Instead of assuming two bodies, now the law is applied to three separate bodies where the objective is to describe the motion of the third body. The resulting expression is given below

$$m_3\ddot{\mathbf{r}}_3 = -\frac{Gm_3m_1}{r_{13}^3}\mathbf{r}_{13} - \frac{Gm_3m_2}{r_{23}^3}\mathbf{r}_{23} \quad (29)$$

where m_1 is the mass of the larger primary body, m_2 is the mass of the smaller primary body, and m_3 is the mass of the spacecraft. For this investigation, P_1 is the position

of the Earth, P_2 is for the Moon, and P_3 corresponds to the spacecraft. The position vectors are defined such that the first subscript defines the base of the vector and the second subscript defines the location of the head. For example, $\mathbf{r}_{12}$ points from P_1 to P_2 . Equation (29) encompasses eighteen first-order differential equations. Since the motion of P_1 and P_2 are coupled with the motion of P_3 , their motion, expressed in the equations below, must also be solved for [24].

$$\begin{aligned} m_1 \ddot{\mathbf{r}}_1 &= -\frac{Gm_1m_2}{r_{21}^3} \mathbf{r}_{21} - \frac{Gm_1m_3}{r_{31}^3} \mathbf{r}_{31} \\ m_2 \ddot{\mathbf{r}}_2 &= -\frac{Gm_2m_1}{r_{12}^3} \mathbf{r}_{12} - \frac{Gm_2m_3}{r_{32}^3} \mathbf{r}_{32} \end{aligned} \tag{30}$$

In terms of integrals of the motion, eighteen are required but only ten are available from the inertial N -body problem, thus, the system cannot be solved. As stated in the previous section, there are ten known integrals of the motion in the N -body problem. Six integrals of the motion are from the conservation of system linear momentum, three are from the conservation of system angular momentum, and the last one is from the conservation of system mechanical energy [24]. Similarly to the restricted 2BP, the three-body problem can be simplified, though not solved, by making a few assumptions as well as transitioning into a more convenient and insightful reference frame.

The first step in simplifying the three-body system is to make a few assumptions. As assumed in the restricted 2BP, the mass of the spacecraft is considered to be significantly smaller than the masses of the primaries. Doing so means the motion of the two primaries is unaffected by the mass of m_3 . This reduces the motion of the primaries to the original two-body system in the previous section, or the two-body problem. A final assumption is the motion of the primaries is circular about their system barycenter [24]. If viewing from the center of either of the primary bodies, the motion of the other primary is circular as well. This assumption, while not necessary,

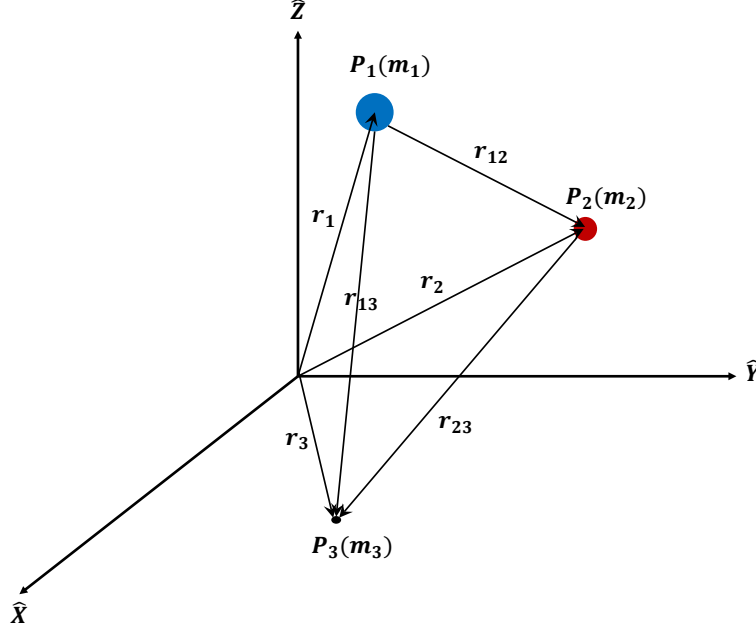


Figure 6. Three-body system in inertial reference frame

simplifies the analysis of the problem. Collectively making these assumptions reduced the general three-body problem into the CR3BP.

Modeling the CR3BP in a rotating, or synodic, reference frame has been found to admit an integral of the motion. The rotating frame, R , in this investigation is denoted by $\hat{x}$, $\hat{y}$ and $\hat{z}$. The origin of the frame is set at the barycenter, B , of the primaries. The $\hat{x}$ direction points through the positions of the primaries from P_1 to P_2 . $\hat{y}$ is orthogonal to $\hat{x}$ in the plane of the primaries. Lastly, $\hat{z}$ is aligned with the angular momentum vector of the primaries. It is important to note that the motion of the primaries is planar in the synodic frame, but P_3 is generally free to move spatially. Yet, in this investigation, only planar P_3 motion is investigated.

Since the CR3BP does not have a known closed-form analytical solution, numerical methods are typically used. To improve numerical performance as well as provide additional quantitative intuition, the problem is nondimensionalized. The characteristic parameters for the nondimensionalization are defined in Table 3.

Table 3. Characteristic quantities for the CR3BP

Symbol	Unit	Definition	Dimensional Value
l^*	Length	Distance between the primaries	384,400 km
m^*	Mass	Total system mass, $m_1 + m_2$	$6.046 \cdot 10^{24}$ kg
t^*	Time	$\sqrt{\frac{l^{*3}}{Gm^*}}$	4.342 days

The characteristic time, t^* , is defined such that the nondimensional gravitational parameter, μ , and the nondimensional mean motion of the primaries, n , are conveniently equal to 1. t^* is also equal to the time it takes the rotating frame of the primaries to sweep through 1 radian, resulting in a period of the primaries that is 2π . To nondimensionalize a dimensional parameter, one divides by the appropriate characteristic quantity. For example, if a distance L is given in kilometers, the equivalent nondimensional distance is $L_{nd} = \frac{L}{l^*}$. The inverse of the previous operation can be conducted to transform from nondimensional units to dimensional units as long as the appropriate characteristic quantities are applied.

The rotation angle, θ , about the $\hat{Z}$ axis defines the rotation angle to transform from barycentric inertial coordinates to barycentric rotating coordinates and is equal to $\theta = n\tau$ where τ is nondimensional time [4]. Since n is equal to 1 nondimensionally, then $\dot{\theta} = 1$, where the time derivative is with respect to nondimensional time τ . The positions of the primaries are defined in terms of the system mass ratio, $\mu = \frac{m_2}{m_1+m_2}$, which for the Earth-Moon system equals 0.0121505865505687. A special note is that this definition of μ in the CR3BP varies depending on the author and is also different from the gravitational parameter in the 2BP.

To define the motion of the spacecraft (P_3), the EOMs of interest are those that govern the vector $\boldsymbol{\rho}$ depicted in Figure 7. A derivation of the EOMs in rotating barycentric coordinates is presented in Szebehely [24] and Murray and Dermott [26].

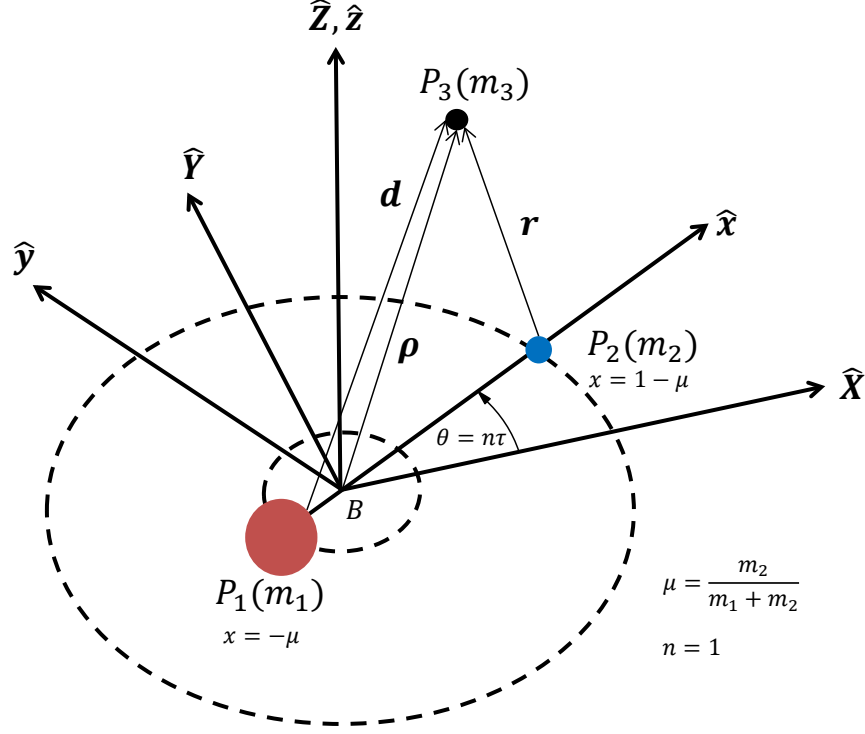


Figure 7. Barycentric rotating reference frame, adapted from Stuart [3]

The same end result can be achieved by first rewriting equation (29) in nondimensional units and in terms of the parameters in Figure 7 with respect to an inertial reference frame centered on B . The resulting second-order nondimensional vector ordinary differential equation is

$$\ddot{\boldsymbol{\rho}} = -\frac{(1-\mu)}{d^3}\mathbf{d} - \frac{\mu}{r^3}\mathbf{r} \quad (31)$$

where $\ddot{\boldsymbol{\rho}}$ is the acceleration of $\boldsymbol{\rho}$ with respect to nondimensional time and the inertial reference frame centered on B . Since $\mathbf{d}$ and $\mathbf{r}$ can easily be written in inertial or rotating coordinates, the remaining step is to express the left side of equation (31) in rotating barycentric coordinates, x , y , and z . To do this, the transport theorem must be applied twice on $\boldsymbol{\rho} = x\hat{x} + y\hat{y} + z\hat{z}$. The transport theorem provides the necessary relationship to take derivatives when using multiple reference frames. The relationship is $\frac{{}^I d}{dt}[\] = \frac{{}^R d}{dt}[\] + \boldsymbol{\omega}^{RI} \times [\]$, where the superscripts I and R signify the reference frame and $\boldsymbol{\omega}^{RI}$ is the angular velocity vector of the R frame with respect

to the I frame. Applying the transport theorem twice on $\boldsymbol{\rho}$ results in the equation below.

$$\ddot{\boldsymbol{\rho}} = (\ddot{x} - 2\dot{y} - x)\hat{x} + (\ddot{y} + 2\dot{x} - y)\hat{y} + \ddot{z}\hat{z} \quad (32)$$

Even though the system is written in barycentric rotating frame coordinates, the derivatives are still taken with respect to an inertial reference frame. That is, the dynamics or described motion are still consistent with what would be seen by an inertial observer, but expressed in non-inertial coordinates. Setting equation (32) equal to equation (31) yields a system of three scalar second-order nondimensional ordinary differential equations describing the motion of P_3 under the gravitational influence of the two primaries. The resulting scalar equations after a minor rearrangement of terms are

$$\ddot{x} = x + 2\dot{y} - \frac{(1 - \mu)(x + \mu)}{d^3} - \frac{\mu(x - 1 + \mu)}{r^3} \quad (33a)$$

$$\ddot{y} = y - 2\dot{x} - \frac{y(1 - \mu)}{d^3} - \frac{\mu y}{r^3} \quad (33b)$$

$$\ddot{z} = -\frac{z(1 - \mu)}{d^3} - \frac{\mu z}{r^3} \quad (33c)$$

where $d = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$. Since the current investigation focuses on planar motion in the CR3BP, to isolate motion to the x - y plane, set z equal to zero. Notice time does not explicitly appear in the CR3BP EOMs; this is because the system is time-invariant. That is, a solution in the CR3BP is independent of time and valid for any other equivalent time span that it encompasses [4]. Lastly, the system of differential equations is highly nonlinear, coupled, and does not currently have a known closed-form analytical solution. In applying the CR3BP EOMs, it is useful to define a “pseudo potential” that expresses the new gravity

potential in this model [27].

$$U^* = \frac{1 - \mu}{d} + \frac{\mu}{r} + \frac{x^2 + y^2}{2} \quad (34)$$

The EOMs can now be rewritten in more concise notation below.

$$\ddot{x} - 2\dot{y} = U_x^* \quad (35a)$$

$$\ddot{y} + 2\dot{x} = U_y^* \quad (35b)$$

$$\ddot{z} = U_z^* \quad (35c)$$

It is often helpful to transform back to an Earth-centered inertial frame to gain a different perspective on a designed trajectory or particular orbit. The transformation between the barycentric rotating frame and the P_1 centered inertial frame is given by the equations below [4].

$$X = (x + \mu)\cos(\theta) - y\sin(\theta) \quad (36a)$$

$$Y = (x + \mu)\sin(\theta) + y\cos(\theta) \quad (36b)$$

$$Z = z \quad (36c)$$

$$\dot{X} = -(x + \mu)\sin(\theta) - y\cos(\theta) + \dot{x}\cos(\theta) - \dot{y}\sin(\theta) \quad (36d)$$

$$\dot{Y} = (x + \mu)\cos(\theta) - y\sin(\theta) + \dot{x}\sin(\theta) - \dot{y}\cos(\theta) \quad (36e)$$

$$\dot{Z} = \dot{z} \quad (36f)$$

The next section expands upon the CR3BP by providing useful insight necessary for the trajectory design process.

2.2.2 Insight into the CR3BP

An important aspect of the CR3BP is that there exists regions of the phase space where the solutions are sensitive to small changes in the initial conditions. In modern nomenclature, this quality is termed deterministic chaos [28]. Due to the existence of chaotic behavior, vastly different qualitative behaviors can exist in close proximity. This fact provides an exciting yet exceedingly challenging design environment. A useful result of deriving the system in rotating coordinates is the existence of an integral of the motion. In 1836, the German mathematician Carl Jacobi discovered the “integral of relative energy” or the Jacobi constant, C [26,29]. The Jacobi constant is an energy integral and, although it is not the total system mechanical energy, it is an “energy-like” quantity [18]. The Jacobi constant is a function of position and velocity and is given by

$$C = x^2 + y^2 + \frac{2(1-\mu)}{d} + \frac{2\mu}{r} - V^2 \quad (37a)$$

$$C = 2U^* - V^2 \quad (37b)$$

where $V = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$. Wiesel provides an apt interpretation of the Jacobi integral when it is transformed back into inertial position and velocity components when he states, “the resulting expression resembles a combination of the total energy of the third mass and its total angular momentum” [18]. It is important to note that authors define C differently, thus, attention is needed when referencing other works. Given the discovery of the Jacobi constant, George William Hill defined accessible regions at particular energy levels in 1878 [24]. The boundaries of the accessible regions are provided by zero-velocity curves (ZVCs) or, in spatial dimensions, zero-velocity surfaces. When the relative velocity of the system is set to zero for a particular Jacobi

constant value, equation (37b) reduces to

$$C = 2U^* \quad (38)$$

The set of coordinates that satisfies equation (38) defines the ZVCs. Regions violating the ZVCs are not physically realizable because they correspond to imaginary magnitudes of relative velocity [25]. The formulation for C used in this investigation implies that a decrease in C corresponds to an increase in the “energy” level of the spacecraft. The opposite is also true. Changing the C value and examining the ZVCs shows an evolution of the accessible regions. In discussion about accessible regions, forbidden regions correspond to $V^2 < 0$, and accessible regions are defined by $V^2 > 0$. A similar yet important note is that an exterior region describes an accessible region that is separated from an interior region by a shaded forbidden region. That is, for a spacecraft starting in an interior region at a certain C , there may be physically realizable (unshaded) regions that the spacecraft cannot reach due to a closed boundary of ZVCs. This also means the same spacecraft at the same energy level starting in the exterior region would not be able to access the interior region. Figure 9 shows the ZVCs corresponding to the value of Jacobi constant associated with each of the CR3BP equilibrium points.

As just hinted, the CR3BP has five equilibrium solutions known as Lagrange points or libration points [24]. The Lagrange points correspond to coordinates where the relative velocity and acceleration are zero or $U_x^* = U_y^* = U_z^* = 0$ [27, 29]. The three Lagrange points that lie on the x -axis are known as the collinear points and were discovered by Leonhard Euler in 1765 [4, 18]. The remaining two, found in 1836 by Joseph-Louis Lagrange, form equilateral triangles with the primaries and are known as the triangular points [18, 24]. Figure 8 shows the locations of the Earth-Moon equilibrium points in the nondimensional barycentric rotating frame. The locations

in nondimensional units (nd) of the five Earth-Moon libration points as well as their corresponding Jacobi constants are given in Table 4.

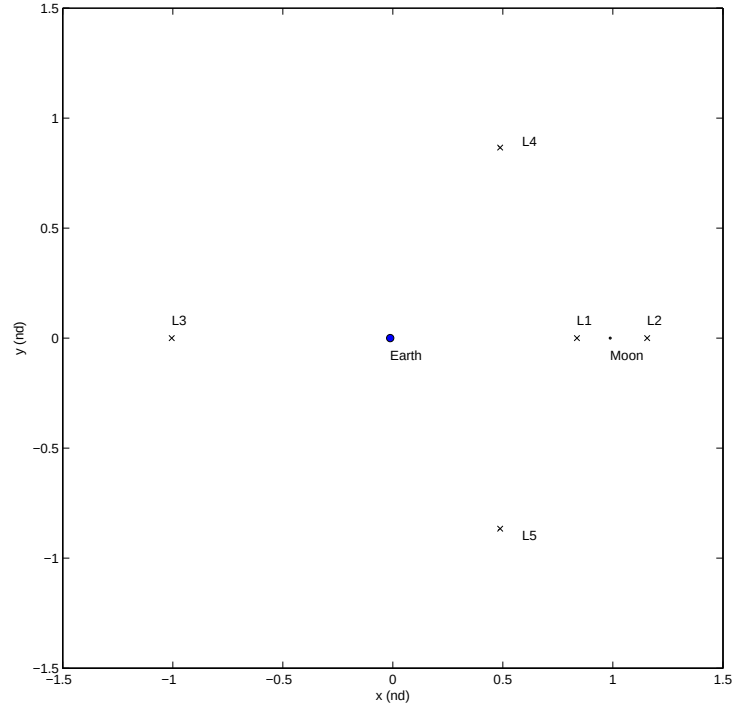


Figure 8. Earth-Moon equilibrium points in the nondimensional barycentric rotating frame

Table 4. CR3BP Earth-Moon Lagrange points

L-point	x (nd)	y (nd)	z (nd)	C
L1	0.836915121142417	0	0	3.18834112642610
L2	1.15568216906384	0	0	3.17216046839511
L3	-1.00506264620231	0	0	3.01214715162089
L4	0.487849413449431	0.866025403784439	0	2.98799705020295
L5	0.487849413449431	-0.866025403784439	0	2.98799705020295

The numbering of the Lagrange points can vary depending on the source. For this investigation, the cislunar Lagrange point is L1, where cislunar defines the body

of space between the Earth and the Moon. The translunar, or “beyond the Moon” point, is labeled L2, and the trans-Earth, “beyond the Earth”, point is L3. L4 is the leading equilateral point whereas L5 is the trailing equilateral point.

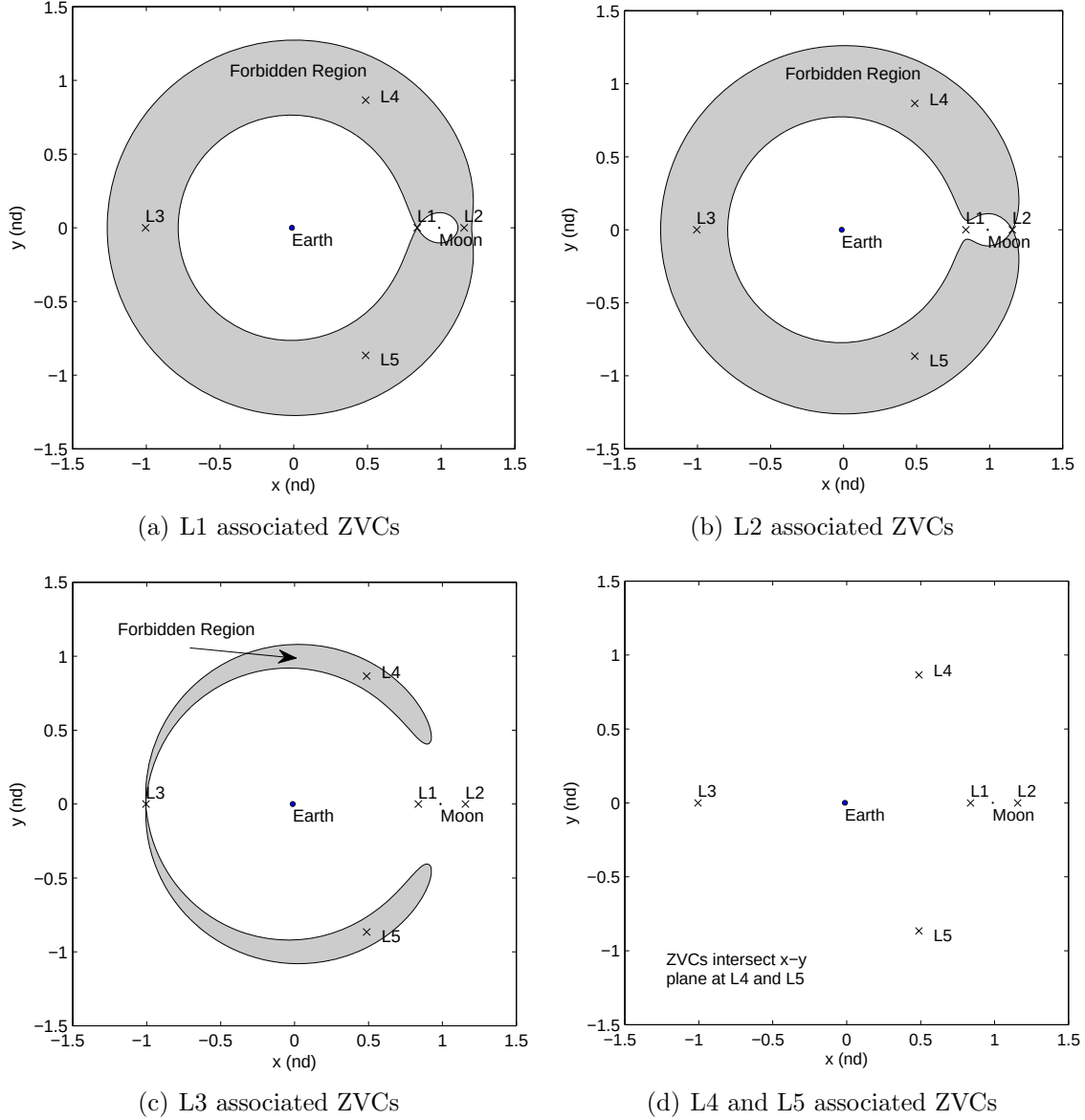


Figure 9. Lagrange point associated zero-velocity curves

Figure 9 shows the ZVCs associated with each of the Lagrange Points where the gray regions are forbidden at the specified energy level. The ZVCs in Figure 9(a) are associated with the energy level of L1 and show that the “L1 gateway” is almost open.

In Figure 9(b), that same gateway is much wider due to a lower C value or higher energy associated with L2. Decreasing C and increasing the energy level to that of L3 is depicted in Figure 9(c) with the L1 and L2 gateways open and the L3 gateway at the cusp of allowing passage [27]. Finally, at the energy levels associated with L4 and L5 shown in Figure 9(d), the ZVCs are just leaving the x - y plane and only exist as points at L4 and L5. In leaving the plane at this “energy” level, there are still zero-velocity surfaces that bound the out-of-plane motion; however, the zero-velocity surfaces exist at only two points, L4 and L5 in the x - y plane. Therefore, at this C value, planar motion is nearly unbounded. Increasing the “energy” level from that of L4 and L5 would cause the zero-velocity surfaces to depart from the x - y plane and continue to increase in out-of-plane separation.

As a final note, an added benefit to having an integral of the motion is that numerical integration error accumulation can be checked by monitoring the change in C . In the CR3BP, rapid error accumulation may occur when motion is close to a primary body. Figures 10 and 11 give an example of Jacobi constant error tracking for sensitive and less sensitive motion, where “sensitive” implies more error. For a given trajectory, seeing only very small variations in the Jacobi constant allows the user to trust the numerically propagated results and acquire a sense of the level of error accumulation. In the current investigation, checking the change in C is used as a means of verifying numerically generated results when designing in the CR3BP. The next section introduces equations of variation as a way to gain insight into the dynamical behavior nearby a particular trajectory.

2.2.3 Equations of Variation

The equations of variation (EOVs) provide a tool for gaining information about neighboring trajectories to a particular reference trajectory. This is exceedingly rel-

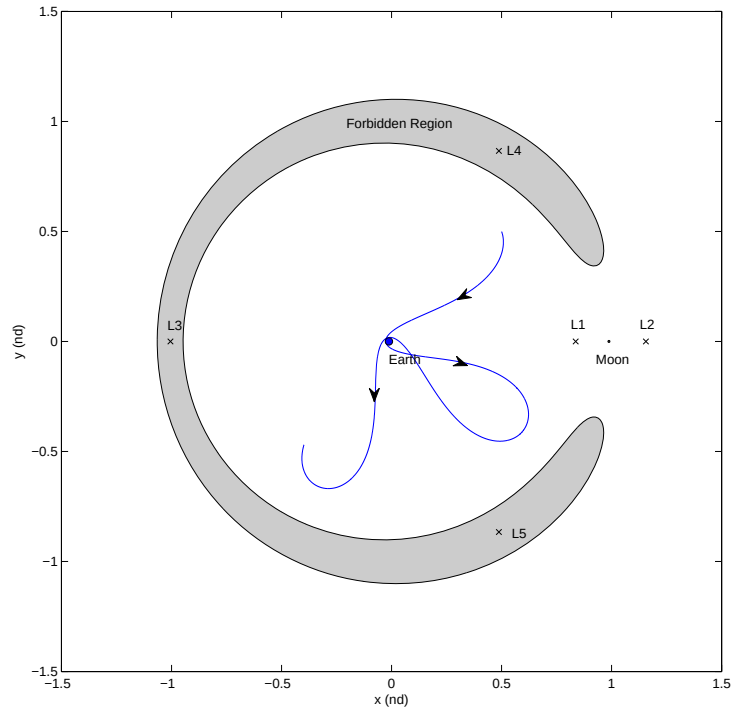


Figure 10. Example spacecraft trajectory propagated for 13.64 days in barycentric rotating frame with associated ZVCs

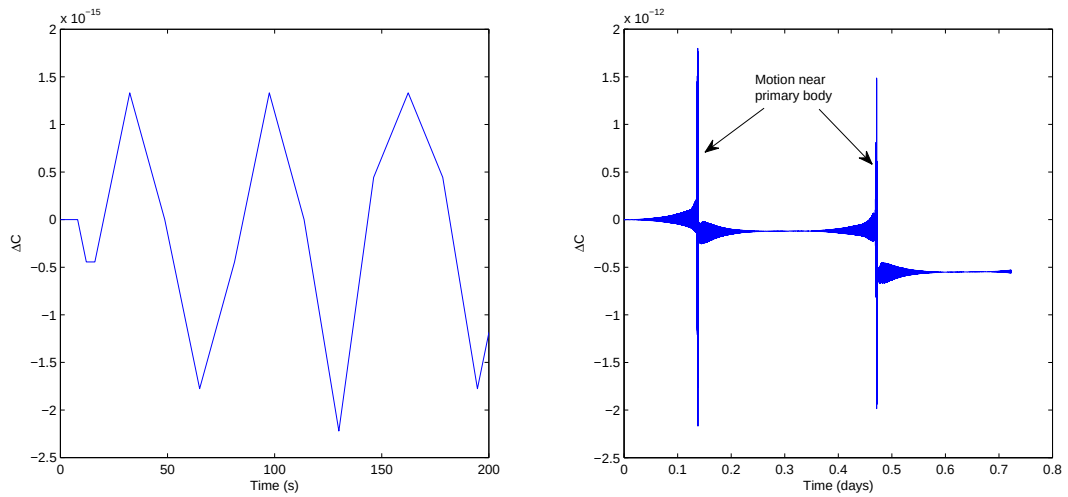


Figure 11. Change in Jacobi constant for the example spacecraft trajectory in the first 200 seconds and the entire 13.64 day propagation time

evant for systems or models where the EOMs must be numerically integrated. In many spacecraft trajectory design applications, this approach is taken when special perturbations are involved. In the CR3BP, due to the lack of a closed-form analytical solution, numerical methods are necessitated and the EOVs become a very useful design tool. Using a first-order Taylor series expansion and perturbing a reference trajectory by small variations yields the linear variational EOMs

$$\ddot{\xi} - 2\dot{\eta} = U_{xx}^* \xi + U_{xy}^* \eta + U_{xz}^* \zeta \quad (39a)$$

$$\ddot{\eta} + 2\dot{\xi} = U_{xy}^* \xi + U_{yy}^* \eta + U_{yz}^* \zeta \quad (39b)$$

$$\ddot{\zeta} = U_{xx}^* \xi + U_{yz}^* \eta + U_{zz}^* \zeta \quad (39c)$$

where ξ , η , and ζ are perturbations in the x , y , and z directions respectively [27]. It is important to note that these equations are linear and are attempting to approximate the behavior of solutions in the vicinity of a reference trajectory. Since the dynamical environment is, in general, highly nonlinear, care must be taken to ensure that the variations and time spans used are reasonably small [27]. The variational EOMs can be written in state space form as

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \\ \ddot{\xi} \\ \ddot{\eta} \\ \ddot{\zeta} \end{bmatrix} = A(t) \begin{bmatrix} \xi \\ \eta \\ \zeta \\ \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \end{bmatrix} \quad (40)$$

where $A(t)$ is a 6×6 matrix with the form

$$A(t) = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ B & \Omega \end{bmatrix} \quad (41)$$

with the elements B and Ω defined as follows

$$B = \begin{bmatrix} U_{xx}^* & U_{xy}^* & U_{xz}^* \\ U_{xy}^* & U_{yy}^* & U_{yz}^* \\ U_{xz}^* & U_{yz}^* & U_{zz}^* \end{bmatrix} \quad (42)$$

$$\Omega = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (43)$$

Introducing the A matrix is necessary to be able to characterize how a variation in the initial conditions of the reference trajectory impacts the state at the final time. The general solution to equation (40) is written below and answers that very question.

$$\delta \mathbf{x}(t) = \Phi(t, t_0) \delta \mathbf{x}(t_0) \quad (44)$$

In equation (44), a variation in the initial conditions can be premultiplied by the state transition matrix (STM), Φ , to linearly approximate the variation in the state at a future time [23]. The STM is governed by the following equation

$$\dot{\Phi}(t, t_0) = A(t) \Phi(t, t_0) \quad (45)$$

with the initial conditions

$$\Phi(t_0, t_0) = I \quad (46)$$

where I is the 6×6 identity matrix. Numerically integrating the STM using equation (45) along with the EOMs allows for information about the neighboring solutions to be propagated as well. In the spatial CR3BP, the STM is a 6×6 matrix defined by the partial derivative of the current state with respect to the initial state.

$$\Phi = \frac{\partial \mathbf{x}}{\partial \mathbf{x}_0} = \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial z_0} & \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial x_0} & \frac{\partial z}{\partial y_0} & \frac{\partial z}{\partial z_0} & \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}}{\partial x_0} & \frac{\partial \dot{x}}{\partial y_0} & \frac{\partial \dot{x}}{\partial z_0} & \frac{\partial \dot{x}}{\partial \dot{x}_0} & \frac{\partial \dot{x}}{\partial \dot{y}_0} & \frac{\partial \dot{x}}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}}{\partial x_0} & \frac{\partial \dot{y}}{\partial y_0} & \frac{\partial \dot{y}}{\partial z_0} & \frac{\partial \dot{y}}{\partial \dot{x}_0} & \frac{\partial \dot{y}}{\partial \dot{y}_0} & \frac{\partial \dot{y}}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}}{\partial x_0} & \frac{\partial \dot{z}}{\partial y_0} & \frac{\partial \dot{z}}{\partial z_0} & \frac{\partial \dot{z}}{\partial \dot{x}_0} & \frac{\partial \dot{z}}{\partial \dot{y}_0} & \frac{\partial \dot{z}}{\partial \dot{z}_0} \end{bmatrix} \quad (47)$$

An important feature of the STM in the CR3BP is that its determinant is equal to unity. By Liouville's theorem, this means the dynamical “flow” of the system is incompressible [27, 30]. Another explanation is that the “volume” occupied in the phase space by a trajectory is preserved [4]. This property has numerical benefits in that the determinant of the STM can be used to measure numerical integration error.

Another benefit of the STM is that it holds information regarding the sensitivity of the reference trajectory to changes in the initial conditions. In other words, it can be an effective gauge on how chaotic the current operating region is. More importantly, the EOVs can be exploited to target trajectories and solve TPBVPs in the CR3BP. Targeting via differential corrections is discussed in the next section.

2.2.4 Targeting Trajectories

When a continuous trajectory between two states is desired, differential corrections based on the Newton-Raphson method can be used target a precise solution. In the context of the CR3BP, a two-point boundary value problem (TPBVP) is often

formulated such that the initial state conditions or times of flight are solved in order to target a desired final state. A TPBVP seeks a solution to the system's EOMs given specified starting and end conditions. It is worth noting that this section introduces optimization concepts that are further detailed in Section 2.3. Also, the targeting scheme being introduced does not aim at finding the optimal solution to a TPBVP, but a feasible solution if one exists. In general, a set of n design variables must be defined in the free-variable array $\mathbf{X}$.

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{bmatrix} \quad (48)$$

Given the design variables, the objective is to drive a set of m equality constraints, $\mathbf{F}(\mathbf{X})$, to zero within a satisfactory tolerance.

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} F_1(\mathbf{X}) \\ F_2(\mathbf{X}) \\ F_3(\mathbf{X}) \\ \vdots \\ F_m(\mathbf{X}) \end{bmatrix} = \mathbf{0} \quad (49)$$

To implement inequality constraints into $\mathbf{F}(\mathbf{X})$, slack variables can be added to the constraints and included as design parameters. The corrections procedure needs to be initialized with a first guess, $\mathbf{X}_0$. A guess-and-check method or conic approximation can be used in the CR3BP for less chaotic regions or near-Earth segments to provide the initial conditions. However, PSO is a tempting and viable method to

supply the initial guess in this investigation. Given $\mathbf{X}_0$, a first-order Taylor series expansion is used to approximate a small change from the reference trajectory $\mathbf{X}_0$.

$$\mathbf{F}(\mathbf{X}) \approx \mathbf{F}(\mathbf{X}_0) + D\mathbf{F}(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0) + H.O.T \quad (50)$$

The higher order terms ($H.O.T$) are dropped resulting in a linear approximation. In order to update the initial states at every iteration, the Jacobian matrix, $D\mathbf{F}(\mathbf{X})$, is needed. The $m \times n$ Jacobian matrix is defined below.

$$D\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \frac{\partial F_1}{\partial X_2} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \frac{\partial F_2}{\partial X_1} & \frac{\partial F_2}{\partial X_2} & \cdots & \frac{\partial F_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial X_1} & \frac{\partial F_m}{\partial X_2} & \cdots & \frac{\partial F_m}{\partial X_n} \end{bmatrix} \quad (51)$$

It is worth noting that, depending on the constraints, elements that populate the $D\mathbf{F}(\mathbf{X})$ may be found in the STM in equation (47). If the number of n design variables equals the number of m constraints, then the $D\mathbf{F}(\mathbf{X})$ matrix is square and invertible. By the implicit function theorem, this implies that if a local solution exists, it is unique [31]. To iteratively drive the constraint array, $\mathbf{F}(\mathbf{X})$, to zero, an update equation is used.

$$\mathbf{X}_{i+1} = \mathbf{X}_i - D\mathbf{F}(\mathbf{X}_i)^{-1}\mathbf{F}(\mathbf{X}_i) \quad (52)$$

Convergence conditions are met once the error, $\|\mathbf{F}(\mathbf{X}_i)\| < \epsilon$ where ϵ is the convergence tolerance. When $n > m$, the TPBVP has infinitely many solutions. A common practice is to choose the next guess closest to the previous guess via the minimum-norm. This approach uses the pseudo-inverse of $D\mathbf{F}(\mathbf{X})$ in the update equation below

$$\mathbf{X}_{i+1} = \mathbf{X}_i - D\mathbf{F}(\mathbf{X}_i)^T [D\mathbf{F}(\mathbf{X}_i)D\mathbf{F}(\mathbf{X}_i)^T]^{-1} \mathbf{F}(\mathbf{X}_i) \quad (53)$$

where the superscript T denotes a matrix transpose. Convergence for a differential corrections scheme using the Newton-Raphson is quadratic; however, this depends on the proximity of the initial guess to the actual solution [32]. This general corrections scheme allows the designer to target orbits and trajectories in the CR3BP. Also, this method of targeting is a popular tool to solve a variety of TPBVPs in the CR3BP [3, 4, 33, 34]. As is shown in Chapter 4, it can even be used to target periodic solutions. The next section seeks to characterize the dynamics in the vicinity of the equilibrium points in the CR3BP.

2.2.5 Stability of the Equilibrium Points

Analyzing the stability of the equilibrium points, or Lagrange points, in the CR3BP provides information about the dynamical motion in the vicinity of the points themselves. A relevant question is: will a spacecraft in the vicinity of a specific Lagrange point remain near the point or depart as time progresses? Answering this question not only tells the designer how expensive long-term missions near the Lagrange points will cost in terms of station-keeping but also sets the stage for a more complete analysis of the dynamical “flow” near the equilibrium solution using dynamical systems theory.

From Lyapunov stability analysis, a solution to the equation $\dot{\mathbf{x}} = f(\mathbf{x}, t)$, denoted $\psi(t)$, is Lyapunov stable if given any $\epsilon > 0$, there exists a $\delta > 0$ such that any solution of $\phi(t)$ satisfying

$$|\phi(t_0) - \psi(t_0)| < \delta \quad (54)$$

for time $t \geq t_0$ satisfies

$$|\phi(t) - \psi(t)| < \epsilon \quad (55)$$

To be considered asymptotically stable, the solution $\phi(t)$ must approach $\psi(t)$ as $t \rightarrow \infty$. The CR3BP libration points are solutions, $\psi(t)$, to the system EOMs, thus, characterizing their respective stability provides information about motion in their vicinity. At the equilibrium points, the A matrix in equation (41) is constant, with six constant eigenvalues, λ_n , for $n = 1, 2, \dots, 6$. In a linear system, examining the real parts of the eigenvalues yields Lyapunov stability information based on the following rule set:

- **Complex Eigenvalues:** If all the roots have negative real parts, the equilibrium point is asymptotically stable. If one or more of the complex roots has a positive real part, then the equilibrium point is unstable.
- **Pure Imaginary Eigenvalues:** For an equilibrium point with all pure imaginary roots, the linear stability is marginally stable. The same is true for an equilibrium point with some pure imaginary roots and with only negative real parts for any remaining eigenvalues.
- **Real Eigenvalues:** If the eigenvalues are negative, the equilibrium point is asymptotically stable. Conversely, if any root has a positive real portion, the equilibrium point is unstable.

Any marginally stable characterizations of the equilibrium point cannot be extended to the nonlinear model. Asymptotically stable and unstable characterizations of the equilibrium point do not share the same restriction and can be extended to the nonlinear system. A nuance not captured in the rule set is that a system with only zero valued eigenvalues is characterized by unstable behavior. Evaluating the CR3BP

libration points for Lyapunov stability results in the collinear points exhibiting stable, unstable, and marginally stable modes. Since this investigation focuses on the planar CR3BP, the collinear points can be characterized as 2-D saddle $\times$ 2-D centers in the x - y plane. The 2-D saddle consists of a 1-D unstable mode and a 1-D stable mode. The 2-D center corresponds to a two-dimensional center subspace. A center subspace is characterized by oscillatory, marginally stable motion. The triangular points, due to purely imaginary roots, are marginally stable, or 2-D center $\times$ 2-D center $\times$ 2-D centers spatially or 2-D center $\times$ 2-D centers in the planar Earth-Moon CR3BP. The characterization of the triangular points depends on the system μ value. For μ values greater than a critical value of 0.03852, the triangular points are unstable [24]. The Earth-Moon μ value is less than the critical μ value, thus, the triangular points exhibit marginally stable behavior. Again, conclusions about the nonlinear behavior of the motion near the triangular points cannot be made based on the linear stability analysis. Figure 12 gives a notional depiction of the 2-D saddle and 2-D center subspaces for one of the collinear Lagrange points.

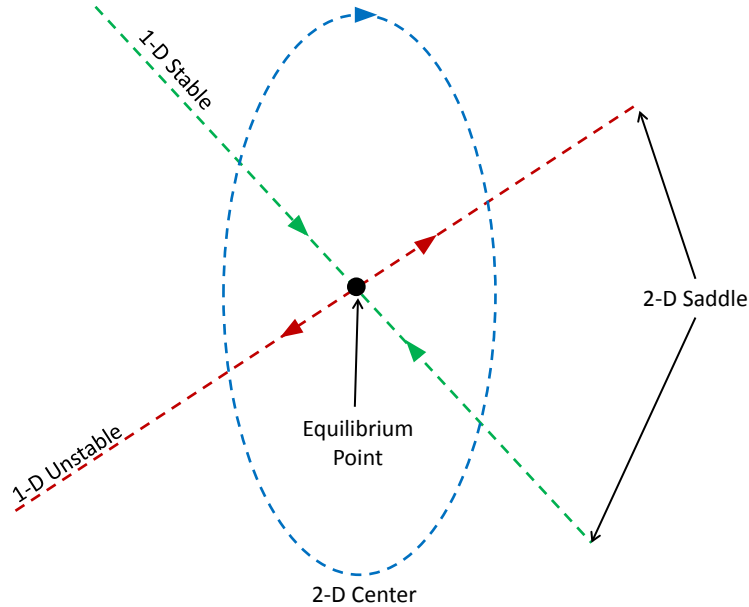


Figure 12. Notional diagram of a 2-D saddle $\times$ 2-D center equilibrium point, adapted from Geisel [4]

The figure shows that the motion in the center subspace is bounded and oscillatory. The 1-D unstable subspace departs the equilibrium point and the 1-D stable subspace approaches the equilibrium point. Together, the 1-D stable and 1-D unstable subspaces make up a 2-D saddle. The next section expands the stability analysis to periodic orbits by introducing dynamical systems theory.

2.2.6 Dynamical Systems Theory

Dynamical systems theory is used in the present investigation to gain a visualization of the “flow” of the dynamics through the use of invariant manifolds. The dynamical systems theory presented in this section is based on the works of Parker and Chua, Guckenheimer and Holmes, and Wiggins [31, 35, 36].

Prior to generating the invariant manifolds, a stability analysis of the periodic solutions must first be made. A periodic solution, in this context, means that after a finite amount of time, the spacecraft, given an initial condition, returns to the same initial condition in both position and velocity within a certain tolerance. The stability analysis of the motion near a periodic solution does not follow the same rule set used when evaluating the stability of equilibrium points. A different approach using Floquet theory is employed in order to gain similar information. Given a periodic solution in the CR3BP, the A matrix is no longer constant as is the case for the equilibrium points, but consists of time-varying periodic terms [27]. Choosing a fixed point along the periodic orbit and generating the monodromy matrix from the STM is the first step. The monodromy matrix is equal to the STM of the periodic orbit after exactly one period, t_p , has elapsed. Floquet theory states that the monodromy matrix can be written in terms of a periodic function, $F(t)$, and Jordan normal form matrix J such that

$$\Phi(t_p, t_0) = F(t_0)e^{Jt_p}F^{-1}(t_0) \quad (56)$$

where $F(t_p)$ consists of the eigenvectors of the monodromy matrix and $F(t_p) = F(t_0)$ [23, 27]. The matrix J is usually a diagonal matrix where the diagonal elements, ω_i , are termed the Poincaré exponents. The eigenvalues of the monodromy matrix, called the characteristic multipliers, λ_i , are related to Poincaré exponents via the simple expression below.

$$\lambda_i = e^{\omega_i t_p} \quad (57)$$

The monodromy matrix is easily computed by numerically propagating the STM for a full cycle using equation (45). The STM is propagated along with spacecraft state itself when numerically integrating the CR3BP EOMs in equations (33). From Floquet theory, evaluating the eigenvalues of the monodromy matrix provides the necessary information regarding the stability of a periodic orbit. The rule set for this set of eigenvalues is as follows:

- If any eigenvalue of the monodromy matrix lies outside the unit circle, the periodic orbit is unstable.
- If an eigenvalue of the monodromy matrix lies within the unit circle, the motion along the associated eigenvector is asymptotically stable.
- If an eigenvalue of the monodromy matrix lies on the unit circle, the motion along the associated eigenvector is marginally stable.

Using this linear stability analysis, asymptotically stable and unstable characterizations of the orbit extend to the nonlinear model, but the same extension cannot be made for marginally stable characterizations of the periodic solution. In the CR3BP, all eigenvalues of the monodromy matrix appear as conjugate pairs, thus, for a periodic orbit to be marginally stable, all of the eigenvalues must lie on the unit circle [3].

Due to the monodromy matrix stemming from a periodic orbit, one of the conjugate pairs must be of unity real value [35]. This implies that a periodic solution cannot be asymptotically stable due to the requirement that all of the eigenvalues be within the unit circle. Expanding this reasoning, if one of the eigenvalues does lie within the unit circle, then the conjugate eigenvalue must exist outside of the unit circle. This conjugate pair corresponds to a 2-D saddle emanating from any fixed point along the periodic orbit. Also, conjugate eigenvalues that exist on the unit circle create 2-D center subspaces emanating from a fixed point along the periodic orbit.

The instability of a periodic orbit is not necessarily an undesirable quality. As is shown, this allows for trajectories to asymptotically approach the unstable periodic orbit. However, once approximately “on” the unstable periodic orbit, station-keeping must be conducted due to the instability drifting. Conversely, marginally stable periodic orbits do not have invariant manifold tubes asymptotically approaching them, therefore, getting to them may be more expensive in terms of fuel. However, this means that staying “on” the marginally stable periodic orbit may be easier due to the decreased level of station-keeping required.

To calculate the eigenvalues along the periodic orbit, it is possible to exploit the periodicity of the solution [3]. For the periodic solution, the eigenvalues of the monodromy matrix are independent of the starting point and can be easily extracted after propagating the STM for a full cycle. For the eigenvectors of the monodromy matrix, it has been shown that they are not independent of the starting point and need to be transitioned from the start point to the desired point via the STM using the expression

$$\mathbf{E}_i(t) = \Phi(t, t_0)\mathbf{E}_i(t_0) \quad (58)$$

where $\mathbf{E}_i(t)$ is the eigenvector corresponding to the i th eigenvalue at time t [27]. In the spatial six-dimensional phase space, there are six eigenvalues corresponding to

each point along the periodic orbit; there are four eigenvalues in the planar problem. Applying equation (58), the stable and unstable eigenvectors at 20 points along a periodic orbit about the Earth-Moon L1 point are shown in Figure 13.

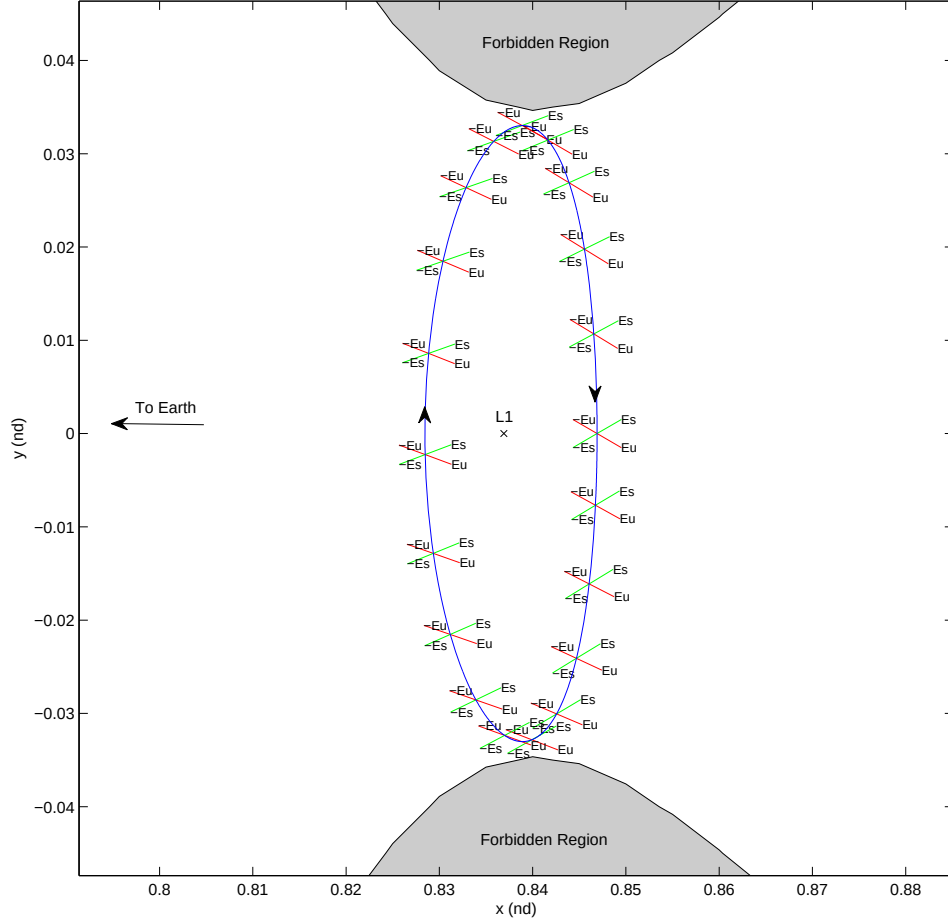


Figure 13. Periodic orbit in the vicinity of L1 with associated stable and unstable eigenvectors and ZVCs in nondimensional rotating barycentric frame

In the figure, the stable eigenvector directions are green and denoted E_s , and the unstable eigenvector directions are red and denoted E_u . The negative signs are used to specify which direction along the eigenvector is being used. The gray region shows the forbidden region corresponding to the value of the Jacobi constant of the blue periodic orbit. The black arrowheads indicate the direction of motion of the spacecraft moving along the periodic orbit.

Propagating a spacecraft along the eigenvector directions corresponding to the

2-D saddle subspace results in asymptotic behavior. For the unstable directions, the spacecraft tends to depart the periodic orbit in positive time whereas the spacecraft asymptotically approaches in the stable eigenvalue directions. If attempting to approach the periodic orbit, the spacecraft should approach on a stable manifold trajectory. To propagate the manifold trajectories emanating from the periodic orbit eigenvectors, a displacement value d is implemented so that propagation does not take prohibitively long to depart or approach the libration point orbit (LPO). Even if the computer is fast enough to overcome the asymptotic behavior near the periodic orbit given a very small d , numerical integration error tends to accumulate for longer integration times. Choosing an appropriate d value is not as simple as arbitrarily picking a large number either. If the d value is too large, then the propagated trajectory may not be a “good enough” approximation for the stable or unstable manifold trajectory. In this investigation, stepping off using a dimensional displacement of 50 km is acceptable for the Earth-Moon system [4]. The displacement is not applied in just position, but is also applied to the velocity. This is done by first normalizing the eigenvectors (in position only) using the equation below

$$\hat{\mathbf{E}} = \frac{\mathbf{E}}{\sqrt{x_E^2 + y_E^2 + z_E^2}} \quad (59)$$

where x_E , y_E , and z_E are the position components of the eigenvector $\mathbf{E}$. After normalizing, the displacement is applied to compute the initial conditions for the approximate manifold trajectories via the expressions below

$$\begin{aligned} \mathbf{X}_s(t_0) &= \mathbf{X}(t) \pm d\hat{\mathbf{E}}^s(t) \\ \mathbf{X}_u(t_0) &= \mathbf{X}(t) \pm d\hat{\mathbf{E}}^u(t) \end{aligned} \quad (60)$$

where $\mathbf{X}_s(t_0)$ is the initial position to propagate along the stable subspace, $\mathbf{X}_u(t_0)$ is

the initial position to propagate in the unstable direction, $\hat{\mathbf{E}}^s$ is the stable eigenvector emanating from the period orbit at time t along the orbit, and $\hat{\mathbf{E}}^u$ is the unstable eigenvector. Propagating in the stable direction must occur in *negative time* in order to depart the periodic orbit. Conversely, the unstable manifold directions require a propagation in forward time. When trajectories emanating from a collection of points along the periodic orbit are propagated, a manifold “tube” appears which serves as a geometric representation of the dynamical “flow” asymptotically approaching and departing the unstable periodic orbit. The stable and unstable manifold tubes emanating from the same periodic orbit near L1 in Figure 13 are shown in Figure 14. As denoted by the directions of motion in Figure 14, the stable manifold trajectories approach the periodic orbit, and the unstable manifold trajectories depart. Four manifold tubes are shown in the figure corresponding to the “positive” and “negative” directions of the unstable and stable eigenvector directions. Figure 14 only depicts the positions of the manifold trajectories, however; for the spacecraft to be “on” a particular manifold, the spacecraft must be at the exact position and velocity of the manifold trajectory.

The CR3BP, the insight into the CR3BP, and the associated design tools discussed in this section are employed in Chapter 4 to design a trajectory from near-Earth to a periodic orbit in the vicinity of L1. The next section discusses the theory behind optimization and details specific methods that are used in this investigation.

2.3 Optimization Fundamentals and Techniques

Optimization can be divided into many different classifications and is a broad topic of interest. Perhaps the two most fundamental branches of classifications are parameter optimization and functional optimization. In parameter optimization, commonly referred to as optimal design, the parameters are time-invariant and limit the prob-

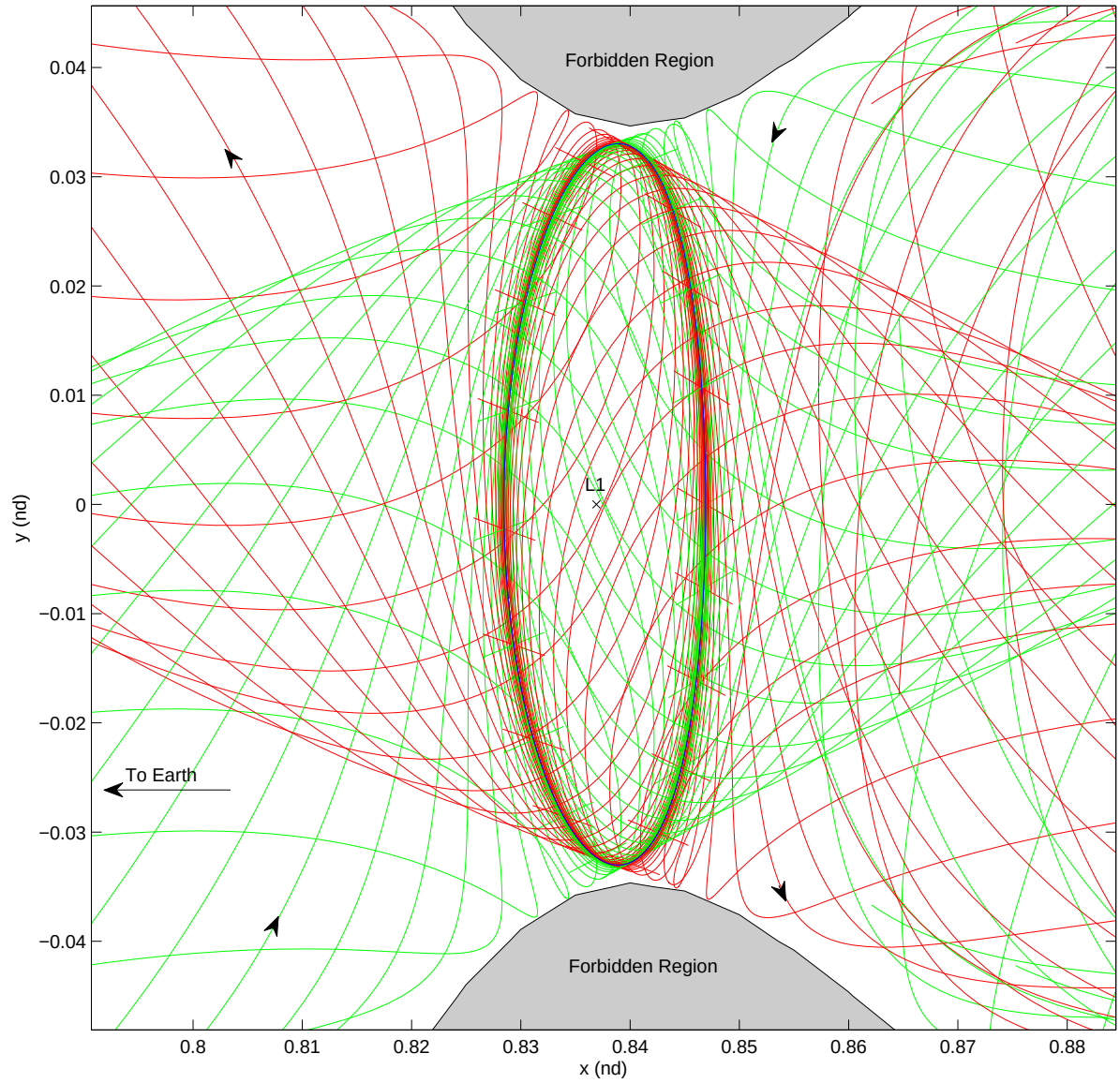


Figure 14. Stable and unstable manifold trajectories emanating from periodic orbit in the vicinity of L1 in nondimensional rotating barycentric frame

lem to a finite dimension. For functional optimization, frequently referred to as optimal control theory, one or more of the parameters are dynamic or functions of time, thus the problem is no longer of finite dimension. The reason why is it called functional optimization is because the function being minimized is a function of a function, or a *functional* [37]. Another important distinguishing category is between *direct* and *indirect* methods of optimization. When concerned with parameter optimization, indirect methods exploit the optimality criteria or Karush-Kuhn-Tucker conditions (KKT) to solve for optimal solutions, while direct methods are those that start with an initial guess and search the design space iteratively for a solution [5]. The KKT first-order necessary conditions are a way to check optimality of an equality or inequality constrained problem [5]. Conversely, when discussing optimal control, indirect methods solve for the solution via the necessary conditions from the calculus of variations, whereas direct methods transcribe the optimal control problem into a simpler-to-handle parameter optimization problem [6]. This investigation focuses on direct methods to generate optimal solutions.

2.3.1 Parameter Optimization

The optimization problems in this investigation are directly transcribed into parameter optimization problems, thus, it is necessary to introduce the concepts behind parameter optimization. The following discussion and notation is modeled after the parameter optimization section in Longuski, Guzmán, and Prussing [37].

Parameter optimization is concerned with finding extrema through methods that stem from calculus. An extremum can be a maximum or a minimum as well as either local to a region or global to the entire function.

Figure 15 demonstrates the difference between local and global extrema. Since the x -domain is bounded, defining the global maximum and minimum is relatively

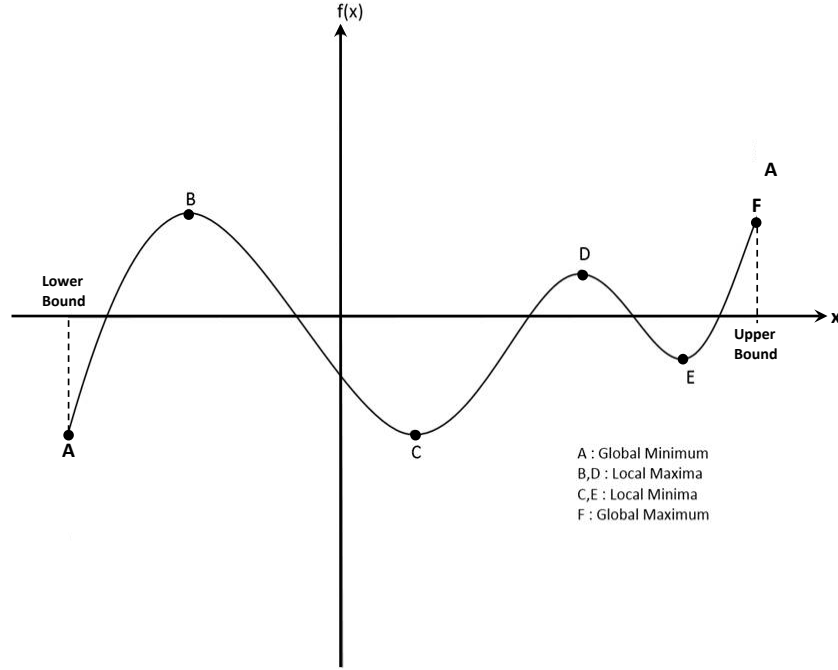


Figure 15. Local and global extrema, adapted from Arora [5]

straightforward. In other cases, such as if $f(x)$ is unbounded, a global optimum may or may not exist depending on the end behavior of the function. Given that parameter optimization is concerned with finding maxima and minima, the problem is typically stated as:

Find: $\mathbf{X}$

such that the function J is minimized

$$J = f(\mathbf{X})$$

The function J is a scalar performance index called a cost function while $\mathbf{X}$ is a constant vector of n dimensions [37]. It is important to note that the following discussion and the current investigation assumes the cost function is to be minimized; if a maximum is to be found, one simply adds a negative sign in front of the function.

From calculus, we know that at a maximum or minimum the first derivative of the function is equal to zero, thus, we are able to define the first-order necessary condition for a local minimum.

$$\frac{\partial f(\mathbf{X}^*)}{\partial \mathbf{X}_i} = 0 \quad (61)$$

Since the first derivative only identifies stationary points, the second derivative is needed in order to identify the nature of the extrema. A second derivative less than zero corresponds to a maximum, and a second derivative greater than zero corresponds to a minimum. For a minimum, the second-order necessary condition states that the Hessian of the function at $\mathbf{X}^*$ must be positive semidefinite or positive definite [5] where the Hessian is defined as

$$H(\mathbf{X}^*) = \left[\frac{\partial^2 f}{\partial X_i \partial X_j} \right]; \quad i = 1 \text{ to } n, \quad j = 1 \text{ to } n \quad (62)$$

The second-order sufficiency condition for the local extrema states that if the Hessian is positive definite at the point $\mathbf{X}^*$, then $\mathbf{X}^*$ is a local minimum for the function $f(\mathbf{X})$. Positive definiteness can be discerned via the n eigenvalues of the Hessian matrix [5]. If they are all positive, greater than zero, then the matrix is positive definite. If one or more of the eigenvalues is zero, then the matrix is positive semidefinite. A point satisfying the first-order necessary condition, the second-order necessary condition, and the second-order sufficiency condition is indeed a local minimum. The necessary conditions are required, however; they may be met at a point that is not a local minimum. Therefore, the sufficiency condition is included to ensure the point is a local minimum.

So far in this development, the functions have been unconstrained; however, it is likely that a real system being optimized will have one or more constraints that must be satisfied. A convenient way to handle constraints in an optimization problem is by using Lagrange multipliers. First and foremost, the constraint must be expressed

algebraically in the form

$$\phi_m(X_1, \dots, X_n) = 0 \quad (63)$$

where $m < n$ so that there are $n - m$ independent variables. If there is an equal number of constraints as there are parameters ($n = m$), then there can only be one or no solutions. Lastly, if there are more constraints than parameters, then the problem is over constrained and there are no solutions [37]. It is important to note that the constraints are assumed to be linearly independent such that there is no redundancy in the problem formulation. Once all m constraints have been expressed, m scalar constants known as Lagrange multipliers ($\lambda_1 \dots \lambda_m$) can be introduced to the problem. This is done by considering what is known as the Lagrangian function below

$$L(X_1, \dots, X_n, \lambda_1, \dots, \lambda_m) = J + \lambda_1 \phi_1 + \dots \lambda_m \phi_m \quad (64)$$

The Lagrangian essentially transforms the constrained system into an unconstrained optimization problem where the equations below must now be solved for the unknowns [37].

$$\frac{\partial L}{\partial X_i} = 0; \quad i = 1 \text{ to } n \quad (65)$$

$$\frac{\partial L}{\partial \lambda_j} = \phi_j; \quad j = 1 \text{ to } m \quad (66)$$

The reason why the Lagrange multipliers are appended is because, in practice, it is easier to solve the $n + m$ equations when compared to solving the constrained system with fewer unknowns [37]. For the system above, the same necessary and sufficient conditions from the unconstrained optimization problem apply; however, there are now an additional m equations for the partials with respect to the Lagrange multipliers that must also be handled.

In order to deal with inequality constraints, slack variables are appended to the

constraints to transform them into equality constraints with Lagrange multipliers still in place to further reduce the system [5]. The inequality constraints take on the form below

$$\phi_m(X_1, \dots, X_n) + s^2 = 0 \quad (67)$$

where s is a slack variable. The slack variable serves as another scalar unknown allowing design variables with values satisfying the interior feasible region of the inequality to act as if they are satisfying the constraint at the boundary.

The nonlinear programming (NLP) problem is essentially parameter optimization with a nonlinear objective function, nonlinear constraints, or both. In many sources, NLP and parameter optimization are synonymous. An NLP solver, as is used in this investigation, is an algorithm equipped to solve a parameter optimization by satisfying the KKT conditions.

2.3.2 Optimal Control and Indirect Transcription

Functional optimization, or optimal control, adds a level of complexity to optimal design in that it is based on the calculus of variations, where functionals are being minimized or maximized. Since the optimal control problem in this investigation is directly transcribed, as opposed to indirectly transcribed, into a TBVP, the classical optimal control problem is only introduced. For greater detail on indirect optimal control theory, refer to Bryson and Ho [38].

To highlight the disparity in complexity between parameter optimization and optimal control, Betts states, “the optimal control problem may be interpreted as an extension of the NLP problem to an infinite number of variables” [32]. The extension to an infinite number of variables is due to the necessity of defining the optimal control continuously over the entire trajectory. Fundamentally, the problem requires

solving for the control time history $\mathbf{u}(t)$ such that J is minimized

$$J = \phi[\mathbf{X}(t_f), t_f] + \int_{t_0}^{t_f} L[\mathbf{X}(t), \mathbf{u}(t), t] dt \quad (68)$$

where the first term on the right-hand-side is the terminal cost and the integral is the path cost [38]. t_0 and t_f designate the initial and final time, respectively. The Lagrangian L is similar in form to equation (64); however, now it is a function of the control $\mathbf{u}(t)$ as well as the system state variables $\mathbf{X}(t)$ and time t . The cost function in equation (68), given the initial conditions, is subject to the system state equations

$$\dot{\mathbf{X}} = \mathbf{f}[\mathbf{X}(t), \mathbf{u}(t), t] \quad (69)$$

the boundary conditions

$$\boldsymbol{\psi}[\mathbf{X}(t_0), \mathbf{u}(t_0), t_0] = \mathbf{0} \quad (70)$$

$$\boldsymbol{\psi}[\mathbf{X}(t_f), \mathbf{u}(t_f), t_f] = \mathbf{0}$$

and the path constraint vector

$$\mathbf{g}[\mathbf{X}(t), \mathbf{u}(t), t] = \mathbf{0} \quad (71)$$

The system defined above is commonly known as the problem of Bolza due to the existence of both the terminal and path costs in the cost function J . Without the path cost, the problem is reduced to Mayer form and conversely, if only the path cost is present, it is known as the Lagrange Problem [37].

Analogous to the method used in Section 2.3.1, the resulting augmented cost function after appending arbitrary multipliers to satisfy the constraints is written as [38]

$$\hat{J} = [\phi + \boldsymbol{\nu}^T \boldsymbol{\psi}]_{t_f} + \int_{t_0}^{t_f} \left\{ H[\mathbf{X}(t), \mathbf{u}(t), t, \boldsymbol{\lambda}, \boldsymbol{\mu}] - \dot{\mathbf{X}} \right\} dt \quad (72)$$

where the Hamiltonian of the system is defined as

$$H[\mathbf{X}(t), \mathbf{u}(t), t, \boldsymbol{\lambda}, \boldsymbol{\mu}] = L[\mathbf{X}(t), \mathbf{u}(t), t] + \boldsymbol{\lambda}^T(t) \mathbf{f}[\mathbf{X}(t), \mathbf{u}(t), t] + \boldsymbol{\mu}^T(t) \mathbf{g}[\mathbf{X}(t), \mathbf{u}(t), t] \quad (73)$$

The Lagrange multipliers, $\boldsymbol{\nu}$, are appended to the boundary constraints, the *costate variables*, $\boldsymbol{\lambda}$, are adjoined to the differential constraints, and the multipliers, $\boldsymbol{\mu}$, correspond to the path constraints [32]. In Section 2.3.1, the first partial derivatives of the Lagrangian were taken and set equal to zero to provide the necessary conditions for optimality. In this case, the first variation of the augmented function is set equal to zero ($\delta \hat{J} = 0$), resulting in, after extensive derivation, the necessary optimality conditions known as the Euler-Lagrange equations [32].

$$\dot{\boldsymbol{\lambda}} = -\mathbf{H}_X^T \quad (74a)$$

$$\mathbf{0} = \mathbf{H}_u^T \quad (74b)$$

$$\dot{\mathbf{x}} = \mathbf{H}_\lambda^T \quad (74c)$$

The subscripts are shorthand notation denoting the variables in which partial derivatives are taken. A rigorous derivation of the Euler-Lagrange equations can be found in Meirovitch or Greenwood [30, 39]. Satisfying the boundary conditions as well as the Euler-Lagrange equations yields the optimal solution to the problem of Bolza.

This technique showcases the classical approach for indirectly transcribing the optimal control problem into a TPBVP. Solving the indirectly transcribed system for simple optimization problems can be completed analytically; however, for most complex problems, a numerical optimization method, such as an NLP algorithm, is commonly used [40]. It is important to note that using a numerical optimization technique on an indirectly transcribed system does not mean a direct approach is being taken. Indirect vs. direct optimization is a matter of distinguishing between

the transcription method employed, not on the method used to solve the resulting TPBVPs.

A reasonable question to ask is: why not always use the indirect transcription method? In the context of a well defined scenario, given a user with a sufficient optimal control theory background, indirect transcription may not pose a problem. However, if the problem is not well defined, such as the case where multiple constraint combinations and formulations are to be tested, re-derivation of the system needs to be taken. Extensive re-derivation is not desirable and can lead to additional avenues for introducing errors. Another disadvantage to the indirect method is, when path constraints are introduced, such as a variable number of finite burn arcs, the constrained-arc sequence much be known *a priori* [32]. Requiring the user to already have the required insight to satisfy this concern is not always practical, especially in complex systems. Lastly, an initial guess for the costates is required such that the numerical optimization method can converge. Consistent with the theme of this investigation, a search algorithm such as PSO may be used to find the costates and has been shown to be a viable technique [41]. However, direct transcription, while not without its own disadvantages, is the approach employed in this investigation to solve a variety of problems.

2.3.3 Direct Transcription and Shooting

The primary difference between indirect and direct transcription methods is that direct methods discretize the state and control arcs into a sequence of finite segments whereas indirect transcription uses a continuous state and control. Also, direct methods do not explicitly apply the Euler-Lagrange equations nor do they require extensive knowledge of optimal control theory [40]. Without requiring applied optimal control theory, it is easier to change problem formulations and substitute in differ-

ent constraints. Direct methods, developed after indirect methods, are a hallmark of the advent of modern computing due to the greater computational expenditure required [3].

Direct transcription methods divide the controlled segment(s) into n_s intervals

$$t = [t_0, t_1, t_2, \dots, t_M] \quad (75)$$

where M is the number of nodes such that $M = n_s + 1$ [32]. The state $\mathbf{X}(t)$ and control $\mathbf{u}(t)$ is then defined at each node.

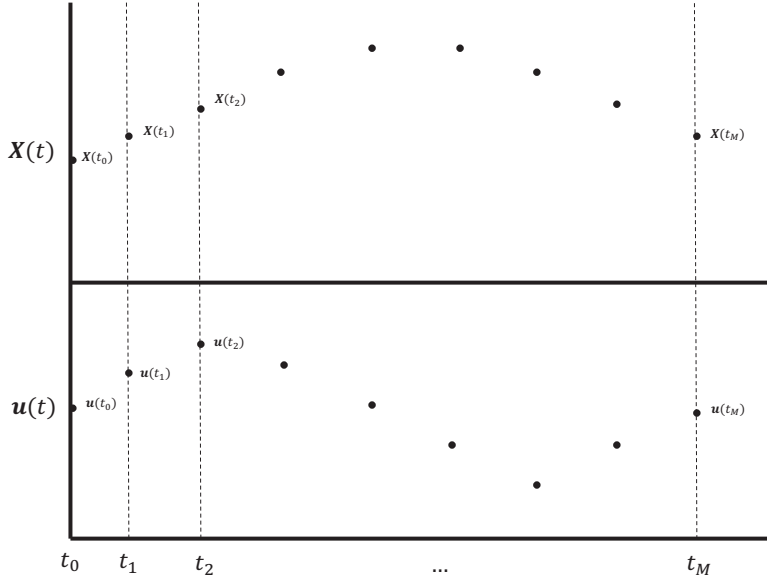


Figure 16. Discretized state and control

After discretizing the states and control time histories, the values of the states at each of the nodes can be calculated via numerical integration, and the control values can be treated as parameters to be optimized via NLP. Any path constraints are enforced at the nodes, and boundary constraints are enforced at the initial and terminal conditions. This approach describes a very simple direct technique known as direct single shooting. Shooting is an appropriate name because the initial state can be thought of as shot, via numerical integration, to the final time and checked

for constraint violation as well as an improving objective function value. Multiple shooting is a more powerful technique that can be applied when the shape of the optimal trajectory is known by inserting variable state parameters along the trajectory to also be optimized along with the control parameters. The variable state parameters give the initial condition a qualitative shape where the optimizer hopefully finds a locally optimal solution that inherits these qualities.

For both shooting methods, the initial conditions as well as time can be set as parameters to be optimized. Other trajectory design parameters such as thrust efficiency, throttling, and pointing angles may be included in the design parameter array $\mathbf{P}$.

$$\mathbf{P}^T = [U, T, D]^T \quad (76)$$

where U is the collection of all the control parameters, T contains the durations of all of the segments, and D hold any remaining design parameters. The cost function J is a function of Φ , such that the optimization problem is to minimize J

$$J = \Phi(\mathbf{P}) \quad (77)$$

subject to the bounds

$$B_l \leq \left\{ \begin{array}{c} \mathbf{P} \\ \mathbf{C}(\mathbf{P}) \end{array} \right\} \leq B_u \quad (78)$$

The upper and lower bounds of the constraints are designated by B_u and B_l , respectively [6]. The array $\mathbf{C}$ is the collection of all of the nonlinear constraints; path constraints evaluated at the nodes are also collected in the $\mathbf{C}(\mathbf{P})$ term. This formulation is remarkably simple and allows for easy implementation of a variety of constraints.

Shooting does not limit the number of controlled segments to unity. In fact,

a shooter only requires a function Φ to transform a given set of initial conditions and control variables into the end state. The number of controlled segments and uncontrolled segments can be variable. For example, if a multiple-burn trajectory is being optimized, the number of burns and coasts can be left up to the optimizer to decide. Allowing the duration of each of the burn and coast segments to be a parameter, unneeded segments may collapse to zero [6]. This is a powerful benefit when compared to indirect methods that require intuition into the shape or nature of the optimal solution. Figure 17 depicts a notional diagram of single shooting where $\mathbf{X}^*(t)$ is the optimal trajectory and $\mathbf{X}_i(t)$ is the i th guess for the trajectory.

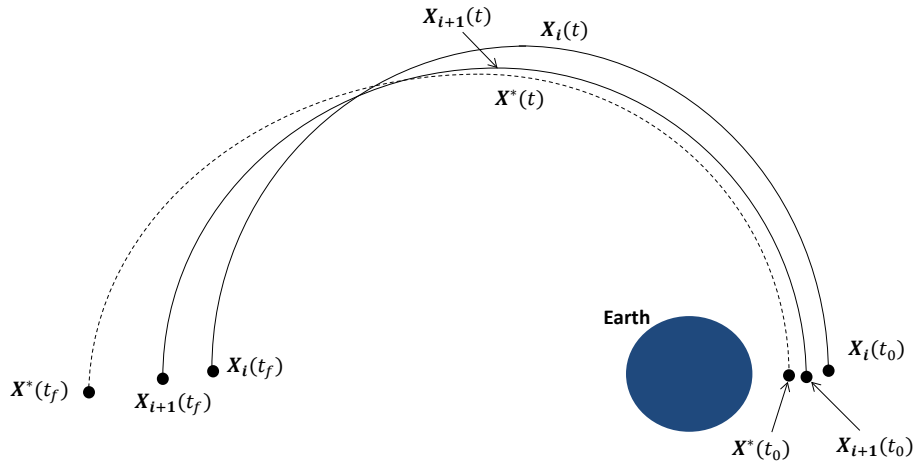


Figure 17. Notional diagram of single shooting

The NLP solver iteratively approaches the optimal trajectory given a baseline while ensuring the constraints are met. The KKT conditions are checked by the NLP solver to verify that a locally optimal solution has been found. The search direction and convergence behavior depends on the problem and specific algorithm chosen. In most cases, the NLP solver utilizes gradient information to dictate the search direction. Providing this information to the NLP solver can reduce computation time. Some solvers can run without the gradient information provided by the user and numerically determine the gradients via finite differencing, but this often results

in longer run times. The gradients of the scalar cost function as well as the constraints can be provided in the form

$$\nabla J = \frac{\partial J}{\partial \mathbf{P}} \quad (79)$$

$$\nabla \mathbf{C} = \frac{\partial \mathbf{C}}{\partial \mathbf{P}} \quad (80)$$

For problems that are highly discretized, the parameter array $\mathbf{P}$ can be very large. This often results in sparse gradient and Hessian matrices; sparse means a high number of zero elements. Detailed information on various NLP algorithms and techniques for handling sparse large-scale problems can be found in Betts [32].

2.3.4 Runge-Kutta Shooting

As mentioned, shooting requires a function Φ to transform the initial conditions to the end state. Due to the insertion of continuous control and non-integrable segments, numerical integration is required. A widely used integration scheme is the Runge-Kutta (RK) method. Numerical integration lends itself to direct transcription due to the fact that the integration occurs over a discretized time interval. For each step in time, multiple RK integration steps can be taken. In this investigation, three integration steps are chosen for each time step. An advantage of RK shooting is that additional control parameters can be inserted at additional points between the nodes. Figure 18 shows the three-step RK integration scheme.

In Figure 18, the subscript n designates a particular node. Also, the control parameters between the nodes are designated as v . With three integration steps, five additional control parameters can be inserted per time step. Having additional control parameters is beneficial when large changes in the control occur over short periods of time or at a much faster rate than the state parameters [6]. This advantage

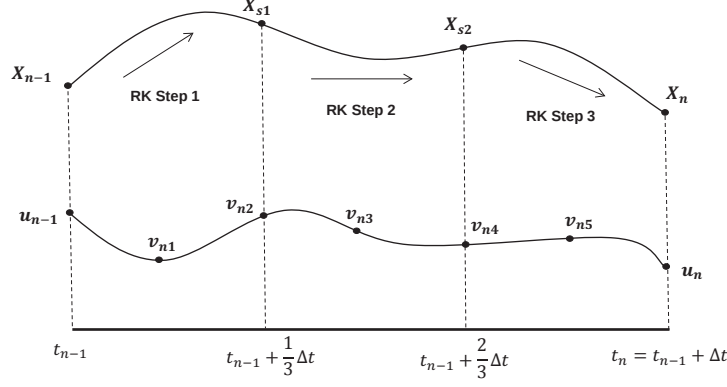


Figure 18. Three-step Runge-Kutta integration, adapted from Conway [6]

proves to be particularly useful for the low-thrust trajectories. The intermediate state approximations, X_s , are not explicitly outputted, but they are necessary for the three step process. Each RK step is a fourth-order approximation. Higher order RK integration schemes exist and may provide increased precision, but at greater computational expense. The fourth-order RK integration equations are presented below [6].

$$\mathbf{X}_{s1}^1 = \mathbf{X}_{n-1} + \frac{1}{6}\Delta t f(\mathbf{X}_{n-1}, \mathbf{u}_{n-1}) \quad (81a)$$

$$\mathbf{X}_{s1}^2 = \mathbf{X}_{n-1} + \frac{1}{6}\Delta t f(\mathbf{X}_s^1, \mathbf{v}_{n1}) \quad (81b)$$

$$\mathbf{X}_{s1}^3 = \mathbf{X}_{n-1} + \frac{1}{6}\Delta t f(\mathbf{X}_s^2, \mathbf{v}_{n1}) \quad (81c)$$

$$\mathbf{X}_{s1}^4 = \mathbf{X}_{n-1} + \frac{1}{12}\Delta t [(f(\mathbf{X}_{n-1}, \mathbf{u}_{n-1}) + 2f(\mathbf{X}_s^1, \mathbf{v}_{n1}) + 2f(\mathbf{X}_s^2, \mathbf{v}_{n2}) + f(\mathbf{X}_s^3, \mathbf{v}_{n2})] \quad (81d)$$

$$\mathbf{X}_{s1} = \mathbf{X}_{s1}^4 \quad (81e)$$

To calculate the second step, $\mathbf{X}_{s1}$ is used as the initial conditions with subsequent intermediate control parameters inserted as necessary. The third step repeats the same process and yields the approximation for $\mathbf{X}_n$. Since numerical integration is an approximation, it is important to choose a time step short enough to yield accurate results but long enough such that the computation time is reasonable. The appropri-

ate level of discretization depends on the problem and is discussed on a case-by-case basis in subsequent chapters.

2.3.5 Particle Swarm Optimization

Particle swarm optimization (PSO) is an inherently different optimization approach from the previously introduced methods that has recently gained popularity in its application to spacecraft trajectory optimization. It officially falls under a broader series of optimizers termed evolutionary algorithms (EA), where the most well-known is the genetic algorithm (GA) [6]. EAs numerically search the design space through methods modeled after behavior found in nature. The advantages of using EAs are that an initial guess is not necessarily required, and they are claimed to be more likely to find a global minimum in the design space when compared to other methods [6].

GAs, while still EAs, are unique in that they mimic the Darwinian process of natural selection. Each individual in a genetic algorithm is encoded, much like DNA, with a string of binary values that represent candidate solutions. Over the course of an iteration, individuals can undergo genetic processes such as genetic cross-over and mutation. After the desired number of iterations, it is expected that the individuals evolve toward an optimal or satisfactory solution [42].

The PSO algorithm, in particular, mimics the behavior of flocking birds and schooling fish and was developed by Kennedy and Eberhart in 1995 [43]. Its attractiveness stems from the fact that it has fewer algorithmic parameters to specify and fine-tune compared to GAs and is simpler to implement by lacking operators such as cross-over and mutation [5]. The PSO algorithm starts out by defining the flock or swarm. Each agent or particle is randomly assigned values for each of the design parameters. The swarm, as a whole, is essentially scattered about the design

space. Each particle then evaluates its own “fitness” based on the system’s cost function and remembers its personal best fitness after each iteration [43]. A global best is also tracked by the swarm to designate the best solution found up to the current iteration by the entire population. After each iteration, the position and velocity of each particle is updated such that the particle moves toward its own best solution as well as the global best solution. A simple PSO algorithm from Arora is formally presented below [5]:

1. Initialize the system by specifying N_p , c_1 , c_2 , and k_{max} , which are the number of particles in the swarm, the cognitive parameter, social parameter, and max number of iterations, respectively. The values of c_1 and c_2 range between 0 and 4, but they are usually set to 2.
2. Randomly assign each particle a location in the design space where the initial location of the i th particle is written $x^{(i,0)}$. Also, evaluate the cost function for each of the particle locations, $J(x^{(i,0)})$ for $i = 1$ to N_p , and determine the best solution among all the particles, $x_G^{(k)}$, where $k = 0$.
3. Calculate the velocity for each of the particles using the equation below

$$\mathbf{v}^{(i,k+1)} = \mathbf{v}^{(i,k)} + c_1 r_1 (\mathbf{x}_p^{(i,k)} - \mathbf{x}^{(i,k)}) + c_2 r_2 (\mathbf{x}_G^{(k)} - \mathbf{x}^{(i,k)}); \quad i = 1 \text{ to } N_p$$

where the variables are defined as the following.

$\mathbf{v}^{(i,k+1)}$ Velocity to update the i th particle for the next iteration

$\mathbf{v}^{(i,k)}$ Velocity of the i th particle at the k th iteration

$\mathbf{x}^{(i,k)}$ Position of the i th particle at the k th iteration

$\mathbf{x}_p^{(i,k)}$ Best position the i th particle has seen up to the k th iteration

$\mathbf{x}_G^{(k)}$ Best position the swarm has seen up to the k th iteration

r_1, r_2 Random number between 0 and 1

For initialization purposes, the vectors $\mathbf{v}(i, 0)$ are set to zero.

4. Update the positions of the particles using the equation below.

$$\mathbf{x}^{(i,k+1)} = \mathbf{x}^{(i,k)} + \mathbf{v}^{(i,k+1)}; \quad i = 1 \text{ to } N_p$$

Also, make sure the positions of the particles are within the parameter bounds such that

$$\mathbf{x}_{Lower} \leq \mathbf{x}^{(i,k+1)} \leq \mathbf{x}_{Upper}$$

$\mathbf{x}_{Lower}$ Vector of lower bounds for all design parameters

$\mathbf{x}_{Upper}$ Vector of upper bounds for all design parameters

If $\mathbf{x}^{(i,k+1)}$ violates one or more of the bounds, set the values of the parameters in violation equal to the nearest bounds.

5. Update the personal and global best solutions by checking the following

$$\text{If } J(\mathbf{x}^{(i,k+1)}) \leq J(\mathbf{x}_P^{(i,k)}), \text{ then } \mathbf{x}^{(i,k)P} = \mathbf{x}^{(i,k)}$$

$$\text{else } \mathbf{x}_P^{(i,k+1)} = \mathbf{x}_P^{(i,k)}; \quad i = 1 \text{ to } N_p$$

$$\text{If } J(\mathbf{x}^{(i,k+1)}) \leq J(\mathbf{x}_G), \text{ then } \mathbf{x}_G = \mathbf{x}_P^{(i,k+1)}; \quad i = 1 \text{ to } N_p$$

6. After step 5, check to see if $k = k_{max}$. If so, stop and report $\mathbf{x}_G$ as the solution.

If not, add 1 to k and return to step 2.

Figure 19 gives a notional depiction of how each particle conducts its position velocity update at each iteration. On the left-hand side of the figure, the particles

in the swarm populate the design space, $\mathbf{P}$, where the positions of the particles correspond to candidate solutions. For the right-hand side, a zoomed in view of a single particle is given. The zoomed in particle, at each iteration, performs a position and velocity update based on its current velocity, $\mathbf{v}^{(i,k)}$, personal best position, $\mathbf{x}_P^{(i,k)}$, and global best position, $\mathbf{x}_G^k$.

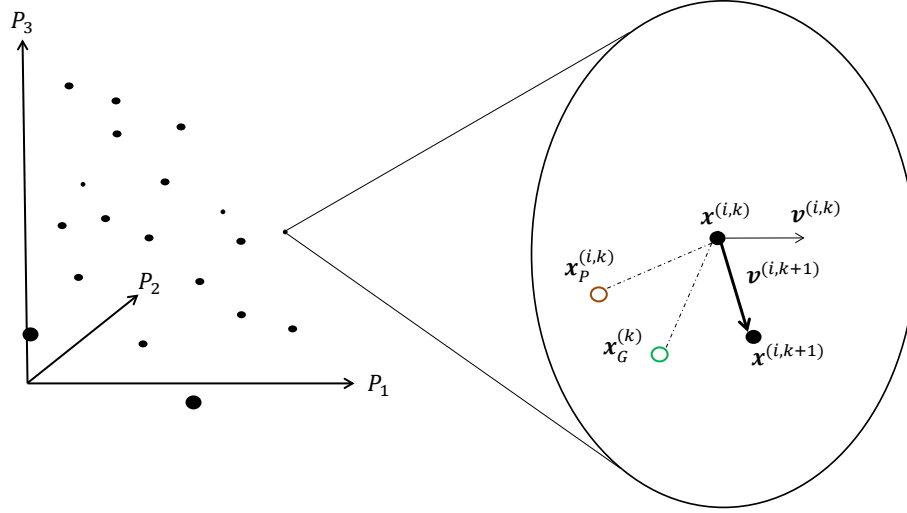


Figure 19. Diagram of the PSO position velocity update process

The algorithm presented can be employed in order to handle an unconstrained optimization problem. However, in many cases, equality or inequality constraints must also be satisfied in order to find a feasible solution. In the context of EAs, Koziel and Michalewicz identified four categories of approaches that can be taken: penalty functions, feasible solution preservation, distinguishing between feasible and infeasible solutions, and hybrids of the previous categories [44]. The method employed in this investigation is a penalty function approach where the penalty depends on the type of constraint.

First off, inequality constraints pose less of a problem to the PSO because they decrease the size of the feasible search space without removing a degree of freedom. To account for them, the particle can be assigned an infinite fitness value if one or more

of the inequality constraints are violated. In addition, the velocity components corresponding to the violated parameters must be set to zero to ensure that the velocity update is only affected by the cognitive and social components of its movement [41]. The algorithm includes modified steps 2 and 3 such that for all N_p particles, one evaluates the inequality constraints. If any are violated, one sets $J(x^{(i,k)}) = \infty$ and $v^{(i,k+1)} = 0$.

Equality constraints are slightly more problematic because they do decrease the degrees of freedom of the problem by m equality constraints. The approach employed in this investigation is to append a penalty function to the system cost function such that

$$\tilde{J} = J + \sum_{r=1}^m \alpha_r |\phi_r(x^{(i,k)})| \quad (82)$$

where there are m equality constraints defined below.

$$\phi_r(x^{(i,k)}) = 0; \quad r = 1 \text{ to } m \quad (83)$$

The weighting coefficients in front of the equality constraints need to be carefully chosen such that there is balance between constraint violation and algorithm conditioning [41]. If the weighting coefficient is too high, the optimizer may have difficulty satisfying the constraint, whereas a very low chosen value may soften the constraint to less than the desired convergence. Methods for choosing these coefficients can vary widely in complexity. Prasad proposed a class of variable penalty functions in the context of a nonlinear programming problem in order to improve performance [45]. For this investigation, a trial-and-error approach was used and checked based on desired convergence tolerances.

As with many EAs, the PSO algorithm presented is just one of several variants that have been formulated and utilized. There are algorithms that call for variable

social and cognitive parameters through weighting based on the progress to the final iteration. Such schemes are set up to emphasize the cognitive parameters at the beginning of the iterative process for diversification and then to see a shift to higher weighting of the social parameter so that the particles drift strongly toward the global best by means of intensification [46]. Other authors have used random numbers for defining the algorithmic parameters at each iteration thereby increasing the stochastic nature of the approach [6]. Variation can also be seen in local and global formulations. Some algorithms may record local best solutions for particles in the vicinity of each other instead of tracking the global best in order to mitigate the risk of the algorithm converging on a local optimum [43]. Ultimately, care must be taken when choosing which type of variant to apply to different systems. Also, fine-tuning of algorithmic parameters may be necessary for acceptable optimizer performance.

2.4 Spacecraft Propulsion

The fundamentals of thrust and propulsion essentially describe a direct application of Newton’s second and third laws. Propulsive force is generated by ejecting mass from the rocket body or satellite at high velocities much like “for every action there is an equal and opposite reaction” [7]. In addition, the force generated by a propulsion system typically happens over a finite time duration. The force of thrust integrated over time is known as total impulse, I_t .

$$I_t = \int_0^t F dt \quad (84)$$

To measure efficiency, total impulse per unit weight of propellant, or specific impulse, I_{sp} , is used. Assuming a constant thrust level and mass flow rate, I_{sp} is expressed in equation (85)

$$I_{sp} = \frac{I_t}{m_p g_0} \quad (85)$$

where m_p is the total effective mass of propellant expelled and g_0 is equal to 9.81 m/s^2 . Note that g_0 is *always* the acceleration due to gravity at Earth sea level regardless of the local gravity field [7]. In this investigation, constant thrust, F , is assumed, thus, equation (85) can be rewritten as

$$I_{sp} = \frac{F}{\dot{m} g_0} = \frac{F}{\dot{w}} \quad (86)$$

where $\dot{m}$ is the mass flow rate. Another useful metric is the effective exhaust velocity, c , which is the average equivalent velocity ejected from the body [7].

$$c = I_{sp} g_0 = \frac{F}{\dot{m}} \quad (87)$$

The effective exhaust velocity is essentially equivalent to the I_{sp} , the difference being a factor of g_0 ; therefore, the two are interchangeable [7]. A more tangible measure of efficiency is the propellant mass fraction, which is the ratio of the propellant mass to the initial mass of the system [7]. The relevant expressions are

$$\zeta = \frac{m_p}{m_0} \quad (88)$$

$$m_f = m_o - m_p \quad (89)$$

where ζ is the propellant mass fraction, m_0 is the initial mass of the spacecraft, and m_f is the final mass of the spacecraft after the transfer has occurred. In most propulsion literature, ζ is the propellant mass fraction of the propulsion system itself; however, in this investigation it is used for the entire satellite.

Depending on the magnitude of the force, desired ranges of acceleration and ef-

efficiencies can only be attained by certain classes of propulsion systems. In general, high-thrust, but lower propellant efficiency, is attained by chemical propulsion systems. Conversely, low-thrust, high-efficiency performance is provided by electric propulsion.

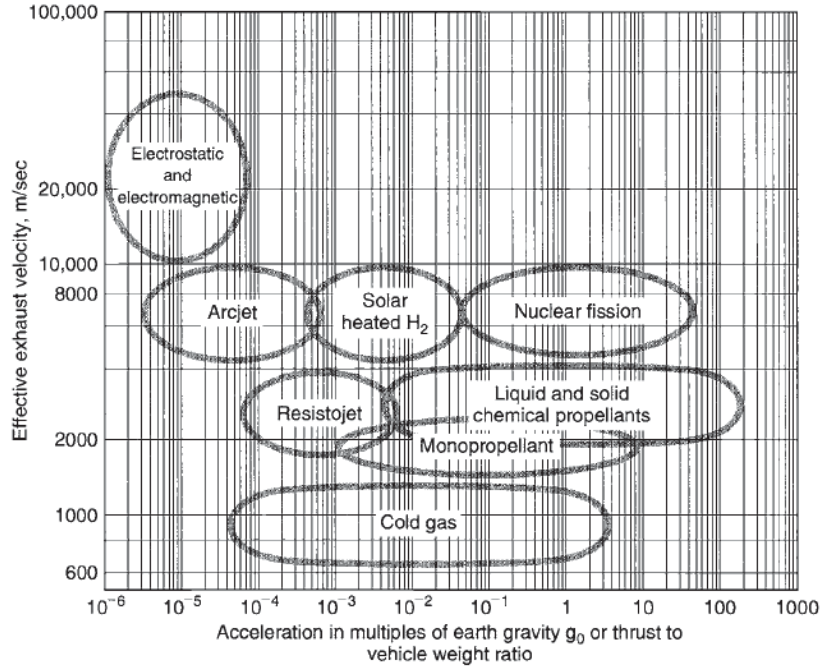


Figure 20. Acceleration vs. effective exhaust velocity, reproduced from Sutton and Biblarz [7]

The low-thrust 2BP trajectories in this investigations assume propulsive acceleration levels of $0.01g$ (0.0981 m/s^2) and an effective exhaust velocity of 11.59 km/s ($I_{sp} = 1,181 \text{ s}$). From Figure 20, this puts the propulsion system at the upper fringes of the “Solar heated H_2 ” regime [7]. The reason for choosing this particular performance level is due to the lack of analytical approximations available for low-thrust transfers [2]. According to Chobotov, there are three ranges of thrust accelerations; their associated solution strategies are given below:

- High-Thrust ($T/W_0 \approx 0.5$ to 1.0): Impulsive burns is a viable assumption.

- Low-Thrust ($T/W_0 \approx 10^{-2}$ to 10^{-1}): Numerical optimization is required.
- Very-Low-Thrust ($T/W_0 \approx 10^{-5}$): Thrust is modeled as a perturbation and continuous first-order analytic spiral solution can be used.

Since the chosen thrust specifications for the 2BP trajectories fall into the “low-thrust” range, there is a need for the numerical optimization techniques as well an initial guess strategy. In contrast, for the impulsive transfers in the CR3BP, the need for an initial guess strategy aided by the analytical insight available is dictated by the complexity of the dynamical environment and not by the propulsive acceleration level. That is, in the CR3BP, there is no known closed-form analytical solution, thus, an initial guess is not readily available.

2.5 Relevant Works in Literature

The main bodies of scholarly works relevant to this investigation are those concerning continuous and low-thrust transfers, direct transcription methods, particle swarm optimization, and the circular restricted three-body problem. Chapters 3 and 4 present the following two sets of test cases, respectively:

1. Low-thrust, fuel-optimal, continuous and multiple-burn transfers from a coplanar and non-coplanar low-Earth-orbit to a geostationary orbit designed in a two-body dynamical model with and without oblate Earth effects
2. Impulsive transfer from near-Earth to a periodic orbit about the Earth-Moon cislunar collinear Lagrange point with minimized burn costs designed in a multi-body dynamical environment

For the first set of cases, Edelbaum provides an analytic approach to the optimal very-low-thrust transfer between circular coplanar and non-coplanar orbits [47]. The

method models the acceleration due to thrust as a perturbation and finds the optimal solution to be a many-revolution spiral. Since the solution requires the perturbing acceleration to be very small compared to dominant gravitational acceleration, this solution only applies to very-low-thrust accelerations. Wiesel and Alfano further generalize Edelbaum’s work to allow for constant thrust as opposed to constant acceleration [48]. Burt provides the secular rates of change for the classic orbital elements given very low levels of thrust, and Pollard extends the work to account for multiple finite burn profiles and steering laws [49,50]. For high-thrust transfers between coplanar circular orbits of radius ratios less than 11.9, a two-impulse Hohmann transfer is optimal in terms of change in velocity, ΔV , required [51]. Since the orbit radius ratio between LEO to GEO is approximately equal to 6, the Hohmann transfer provides the optimal transfer for high-thrust acceleration levels. When non-coplanar transfers are investigated in the current work, the optimal high-thrust acceleration transfer is assumed to use a combined plane change as the second impulse. The combined plane change as the second impulse means all of the necessary inclination change occurs instantaneously at the second impulse in addition to the recircularization required to enter the target orbit. This assumption of using a combined plane change as the second impulse is not necessarily optimal, but is sufficiently near-optimal for comparison purposes in the current investigation.

The 2BP cases are intentionally run at thrust acceleration levels where neither very-low-thrust spiral nor the Hohmann transfer are sufficiently accurate for an initial guess. The difference in optimal solutions is due to a transition from gravitationally dominant motion in the very-low-thrust case to thrust dominant motion in the impulsive case. This transition as well as the required ΔV across the thrust acceleration spectrum for multiple orbit radius ratios are well documented in Vallado [52]. An analytical approximation for coplanar transfers at mid-level thrust accelerations is

provided by Spencer and Culp [53]. This analytical approximation does not allow for inclination changes, thus, it could not be used for all of the two-body test cases. In relaxing the continuous burn constraint and allowing intermittent coast arcs, the required ΔV for a given transfer is lowered. The optimal solution depends on the number of burns allowed and approaches the ΔV of the Hohmann transfer as the number of burns increases. This is because, as the number of burns increases, the required duration for the burns decreases until the transfer consists of an infinite number of small impulses. Pelouch and Redding et al. provide optimal control histories and gravity loss information for finite-burn transfers between LEO and geostationary orbits [54, 55].

Particle swarm optimization, as well as evolutionary algorithms in general, has gained a significant amount of traction in the early twenty-first century. PSO, in particular, is chosen due to its algorithmic simplicity while still boasting a freedom from requiring an initial guess. Many authors use PSO for a variety of applications. Works of notable relevance are those that apply it to space mission design. Pontani and Conway demonstrate PSO's ability to solve continuous and finite transfers as well as periodic libration orbits in the CR3BP in multiple submissions [6, 56, 57]. Genetic algorithms are applied to calculate interplanetary trajectories in multiple works [58, 59, 60]. The polynomial interpolation method used in this investigation to parameterize continuous control time histories for PSO to optimize is detailed in Wall and Conway [61]. In addition, Russell and Shampine as well as Hargraves and Paris have also employed similar polynomial-based interpolation schemes [62, 63]. The foundation for the piecewise polynomial interpolation or spline interpolation method is based off of the work of Birkoff and de Boor [64].

Trajectory design in the CR3BP poses a unique challenge that can be approached from a variety of angles. In the second set of test cases, the final target is insertion

into an L1 periodic orbit. Howell and Pernicka demonstrate successful numerical determination of periodic in-plane and out-of-plane orbits about the collinear Lagrange points [33]. Conway and Pontani demonstrate PSO’s ability to accurately, though less efficiently, calculate the periodic orbits as well [6]. Invariant manifold dynamics serve as an efficient path to approaching the periodic orbit by following the “flow” of the system dynamics in the second test case. An introduction to manifold dynamics in the context of the Earth-Moon libration points is found in Grebow and Stuart [3, 34]. Martin and Conway use a direct transcription technique to optimize a low-thrust transfer between a geostationary transfer orbit and a low lunar orbit [65]. Evolutionary algorithms are also used to calculate transfers from LEO to the Earth-Moon libration points and LPOs by McMahon et al. and Abraham et al. [66, 67]. Their cases are very similar to the ones posed in Chapter 4; however, in this investigation PSO is used in conjunction with NLP.

2.6 Chapter Summary

The background knowledge provided in this chapter focused on the 2BP, CR3BP, direct optimization techniques, and satellite propulsion. The theory given for each section is not exhaustive, but is tailored toward the methodologies employed in the following chapters. The material presented in this chapter is employed to design minimum-fuel trajectories in the 2BP, in the 2BP plus oblate Earth effects, and in the CR3BP. PSO is used to generate the initial guesses of the trajectories to subsequently be improved upon by an NLP algorithm. The direct transcription methods utilized for each of the test cases vary; however, they all resemble a form of direct single shooting. The next chapter focuses on the near-Earth low-thrust trajectories.

3. Low-Thrust Near-Earth Trajectory Design

Altitudes below GEO are considered to be near-Earth for trajectory design in the current investigation. Within this regime, the 2BP is a very good approximation for the dynamical environment an actual satellite experiences. For greater fidelity, additional perturbations such as higher degrees of the Earth's geopotential, air drag, and solar radiation pressure can be included in the model. Analytical solutions in the 2BP are available for the very-low-thrust and impulsive transfer. At the very-low-thrust and impulsive thrust acceleration levels, even when perturbations are added, the analytical solutions still serve as reasonable initial guesses. However, analytical optimal solutions to use as an initial guess are not available for low-thrust (as opposed to very-low-thrust) transfers with or without perturbations. As a result, an initial guess generation method is necessary when operating around the low-thrust acceleration level to initiate any NLP solvers. PSO is employed as a way to generate the initial guess. A polynomial approach is used to parameterize the continuous control time history thereby minimizing the number of design variables PSO must optimize.

This chapter provides the methodology and results for low-thrust trajectories designed in two-body and two-body with oblate Earth effects models. The trajectories start at a low-altitude (300 km) circular orbit and end at a circular geosynchronous altitude ($\sim 35,786$ km) orbit. Coplanar and non-coplanar transfers are investigated where the non-coplanar trajectories start at 28.5 degrees and end at 0 degrees inclination. In addition, the effects of allowing for finite burning as opposed to continuous thrusting are investigated.

3.1 Methodology

This section includes the low-thrust transfer model equations and state space representations that are used to numerically propagate the spacecraft trajectories. Also, details are given on the polynomial parameterization that PSO uses to generate the low-thrust trajectory initial guesses. Lastly, specifics are provided on how the particle swarm generated initial guess (PSOIG) is given to the NLP solver for improvement.

In the present chapter, all simulations range in computation time from less than a few minutes to a few hours depending on the number of particles and iterations used by the PSO and the number of design variables in NLP. The times corresponds to elapsed times in MATLAB[®] (Version: 8.1.0.605 (R2013a); benchmark: 0.2930, 0.3178, 0.2058, 0.3201, 0.6751, 0.5911) [68]. Also, the computer used runs a 64-bit Windows 7 operating system with 4GB of RAM and an Intel(R) Celeron(R) CPU E3400 @2.60 GHz processor.

3.1.1 Low-Thrust Transfer Model

To optimize a minimum-fuel continuous or finite burn trajectory, the amount of time the thruster is active directly corresponds to the amount of fuel spent. For the continuous thrust trajectory, this means that the time of flight is equal to the thrusting duration. Similarly, for finite burns, the thrust time is equal to the sum of the burn arc durations. Thus, the focus of the optimizer is to minimize the burn time. However, when formulating the problems, time of flight or total burn time is not the only independent variable in the trajectory. The optimality of the trajectories also depends on the control time histories of the thrust pointing angle. Allowing the thrust pointing vector to vary over the flight provides ample variability for optimization to occur. For the continuous thrust cases, the time of flight as well as the thrust pointing time history are the parameters to be optimized. Extending to a finite burn model,

the time of flight is segmented into individual burn arc and coast arc durations to be optimized. To define the thrust pointing angle, two angles, α and β , are used. The thrust pointing angle corresponding to the direction within the instantaneous orbit plane is α . The angle β is the angle that defines the out-of-plane thrust component. Figure 21 depicts the pointing angles where $\mathbf{T}$ is the thrust vector.

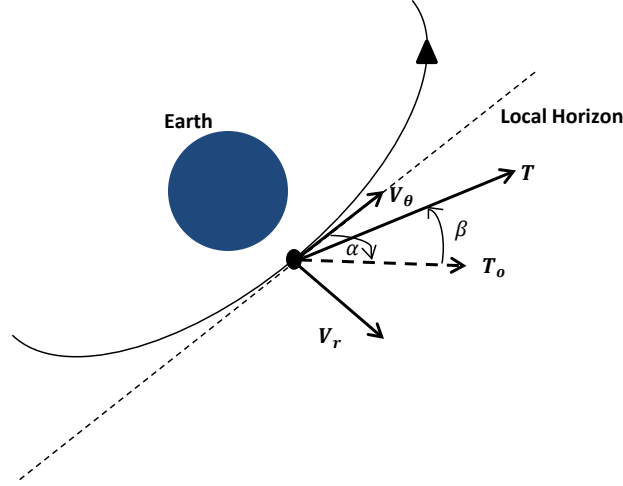


Figure 21. Thrust pointing angles

In Figure 21, α is measured clockwise from the local horizon and β is measured from T_o to T , where T_o is the projection of T onto the instantaneous orbit plane. A positive valued β angle occurs when T points out-of-plane in the northern direction and vice versa. The velocity vector is parameterized into its radial ($\mathbf{V}_r$) and tangential ($\mathbf{V}_\theta$) components.

The dynamics of the satellite are modeled in the restricted two-body problem where the EOMs are

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3} \mathbf{r} = \mathbf{a}_d \quad (90)$$

This equation is very similar to equation (6); however, $\mathbf{a}_d$ is added as a disturbing acceleration [32]. In this chapter, the disturbing accelerations are the propulsive

acceleration and the perturbing acceleration due to the oblateness of the Earth if it is included. Rewriting equation (90) in terms of tangential and radial components and assuming planar motions results in the state equations below

$$\mathbf{X} = \begin{bmatrix} V_r \\ V_\theta \\ r \end{bmatrix} \quad (91)$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{V}_r \\ \dot{V}_\theta \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{\mu-rV_\theta^2}{r^2} + \frac{T}{m}\sin(\alpha) \\ -\frac{V_r V_\theta^2}{r} + \frac{T}{m}\cos(\alpha) \\ V_r \end{bmatrix} \quad (92)$$

In equation (92), the T/m term is the thrust per unit mass or disturbing acceleration due to thrust. The EOMs in equation (92) are strictly for the planar scenarios since only the in-plane pointing angle, α , is being factored in. If the spacecraft is assumed to provide constant thrust, NOT constant acceleration, then the thrust to mass ratio at a given time can be written as

$$\frac{T}{m} = \frac{cn_0}{c - n_0 t} \quad (93)$$

where c is the effective exhaust velocity and n_0 is the thrust to mass ratio at the initial time [6]. Constant acceleration is not assumed because the variable mass of the spacecraft due to propellant usage is included in the T/m term. All of the test cases in this chapter use the same thrust specifications given in Table 5.

Table 5. Test case thrust specifications

Parameter	Dimensional	Nondimensional
c	11.58 km/s	1.5 DU/TU
n_0	0.01g	0.01 DU/TU ²

These particular values were originally taken from Conway to validate preliminary results but then used for all of the low-thrust transfers due to the interesting results [6]. In this chapter, all numerical simulations are run using the following nondimensional units for increased computational performance

$$\begin{aligned}
 1 \text{ DU} &= r_0 = 6678.137 \text{ km} \\
 1 \text{ TU} &= \sqrt{\frac{\text{DU}^3}{\mu}} = 14.41 \text{ min}
 \end{aligned} \tag{94}$$

where $\mu = 398,600.5 \text{ km}^3/\text{s}^2$. Since the initial radius, r_0 for all of the transfers in this section is at the same altitude LEO (300 km), the values for DU and TU remain the same. An important note is that DU and TU are NOT to be confused with the canonical DU and TU used in the 2BP where DU is equal to the radius of the Earth.

A different state space representation is offered for the transfers that include out-of-plane motion. MEEs are used in this state space representation to facilitate the inclusion of J_2 and emphasize its effects. The MEEs are defined in Section 2.1.4. Using MEEs and building the state space representation of the EOMs results in the system shown in equation (95) [69].

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{p} \\ \dot{f} \\ \dot{g} \\ \dot{h} \\ \dot{k} \\ \dot{L} \end{bmatrix} = \begin{bmatrix} 0 & \frac{2p}{w} \sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}} \sin(L) & \sqrt{\frac{p}{\mu}} \frac{(w+1)\cos(L)+f}{w} & -\sqrt{\frac{p}{\mu}} \frac{g[h\sin(L)-k\cos(L)]}{w} \\ -\sqrt{\frac{p}{\mu}} \cos(L) & \sqrt{\frac{p}{\mu}} \frac{(w+1)\sin(L)+g}{w} & \sqrt{\frac{p}{\mu}} \frac{f[h\sin(L)-k\cos(L)]}{w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}} \frac{s^2}{2w} \cos(L) \\ 0 & 0 & \sqrt{\frac{p}{\mu}} \frac{s^2}{2w} \sin(L) \\ 0 & 0 & \sqrt{\frac{p}{\mu}} \frac{h\sin(L)-k\cos(L)}{w} \end{bmatrix} \begin{bmatrix} a_r \\ a_\theta \\ a_w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{\mu p} \left(\frac{w}{p}\right)^2 \end{bmatrix} \quad (95)$$

$$s^2 = 1 + h^2 + k^2 \quad (96)$$

$$w = \frac{p}{r} = 1 + f\cos(L) + g\sin(L)$$

The acceleration components are pointing in the radial direction a_r , along the local horizon a_θ , and in the orbit plane normal a_w direction. The MEE EOMs allow for spatial satellite motion. The accelerations due to the thruster are given by the equations below.

$$\mathbf{a}_{Thruster} = \begin{bmatrix} \frac{T}{m} \sin(\alpha) \cos(\beta) \\ \frac{T}{m} \cos(\alpha) \\ \frac{T}{m} \sin(\beta) \end{bmatrix} \quad (97)$$

Notice that both in-plane and out-of-plane pointing angles are now incorporated. When incorporating J_2 as another disturbing accelerations, the equations provided by Kechichian using MEEs are employed [70].

$$\mathbf{a}_{J_2} = \begin{bmatrix} -\frac{3\mu J_2 R_\oplus^2}{2r^4} \left(1 - 12 \frac{[h\sin(L)-k\cos(L)]^2}{s^4}\right) \\ -\frac{12\mu J_2 R_\oplus^2}{r^4} \frac{[h\sin(L)-k\cos(L)][h\cos(L)+k\sin(L)]}{s^4} \\ -\frac{6\mu J_2 R_\oplus^2}{r^4} \frac{[h\sin(L)-k\cos(L)][1-k^2-h^2]}{s^4} \end{bmatrix} \quad (98)$$

Additional discussion on the J_2 perturbation can be found in Section 2.1.5. In

the present investigation, the coplanar transfers do not utilize MEEs nor do they incorporate J_2 perturbative effects. J_2 is only included in the designated non-coplanar transfers. The next section details the methodology used to generate an initial guess using PSO for the low-thrust transfer model provided in this section.

3.1.2 PSO Initial Guess

When using an NLP solver, converging on an optimal solution typically depends on the proximity of the initial guess to the final answer. While robust numerical algorithms that have wide convergence ranges exist, the sensitivity of the problem can significantly decrease the likelihood of convergence. Since PSO searches the solution space without the need for an initial guess, it is a viable option for initial guess generation. PSO is not a “one-size-fits-all” approach; it works best when the dimension of the search space is as small as possible. The question that needs to be answered then is: how does one parameterize the continuous thrust pointing angle history into a small finite set of parameters? A variety of approaches can be taken.

One approach for parameterizing the continuous control time history is to discretize the continuous control time history using a large time step. PSO then optimizes the control values that are held constant for each time step. The resulting solution looks like a stair-step function with the approximate shape of the optimal solution. This approach was initially investigated; however, finding the right time step that yielded viable results proved to be a time consuming and iterative process. Also, for the continuous thrust cases, the issue of singular arcs became predominant. A singular arc arises when the optimal control solution is not unique [32]. An example of a singular-arc solution is shown in Figure 22. This problem arises due to the lack of path constraints limiting the rate of change of the control, thus, allowing for an erratic solution. Placing path constraints can potentially solve the issue; however, an

interpolation method is chosen in this investigation to more efficiently decrease the size of problem.

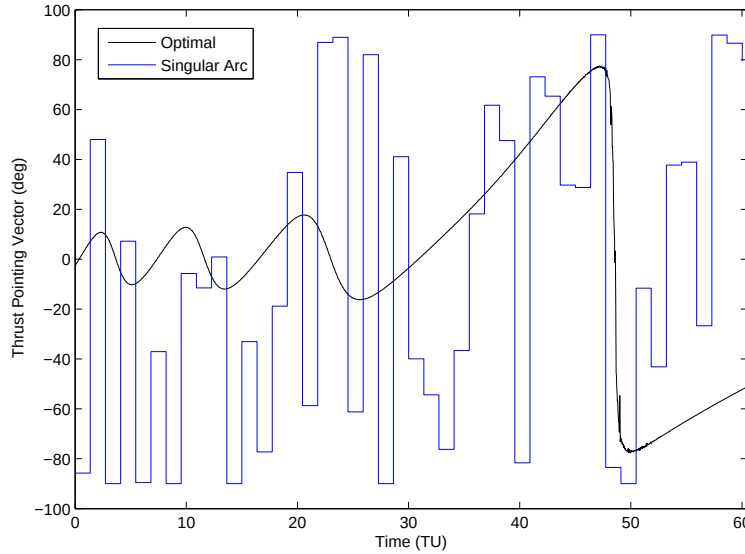


Figure 22. Singular arc control example

Another approach for parameterizing the continuous control time history that holds promise is to model the control time history as a stringed together set of polynomials frequently termed spline interpolation. This approach is shown to be effective for spacecraft trajectory design in Wall and Conway [61]. The order of the polynomial can be experimented with, though increasing the degree of the polynomial can cause problematic oscillations due to Runge’s phenomenon [71]. The problematic oscillations that arise due to Runge’s phenomenon occur at the edges of an interval that is interpolated using polynomials. The problem is exacerbated by increasing the degree of the polynomial(s) used for the interpolation. A set of fourth-order polynomials proves to be sufficient in the current investigation while still avoiding the problematic oscillations at the edges. PSO optimizes the burn time(s) as well as the polynomial coefficients that define the control angles throughout the burn(s). The polynomials

take the form

$$c_n(t) = k_{n1}(t - \frac{\tau_i}{2n_s}\Delta t)^4 + k_{n2}(t - \frac{\tau_i}{2n_s}\Delta t)^3 + k_{n3}(t - \frac{\tau_i}{2n_s}\Delta t)^2 + k_{n4}(t - \frac{\tau_i}{2n_s}\Delta t) + k_{n5} \quad (99)$$

where $c_n(t)$ is the control value given by the n th polynomial at time t , τ_i is the number of time increments the entire controlled trajectory is partitioned into, n_s is the number of polynomials used, Δt is the time-step size, and $k_{n(1-5)}$ are the fourth-order polynomial coefficients. The polynomials are centered at the middle of their respective time intervals.

The PSO algorithm optimizes the polynomial coefficients and burn time(s) in the design parameter array $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & \cdots & k_{n5} & T \end{bmatrix}^T \quad (100)$$

where T contains the burn time(s). The bounds on the coefficients are set such that the pointing angle stays within ± 180 degrees. The bounds are calculated using the equations below [61]

$$\begin{bmatrix} -\frac{\pi}{(\frac{t_f}{n_s n_b})^4} \\ -\frac{\pi}{(\frac{t_f}{n_s n_b})^3} \\ -\frac{\pi}{(\frac{t_f}{n_s n_b})^2} \\ -\frac{\pi}{(\frac{t_f}{n_s n_b})} \\ -\pi \end{bmatrix} \leq \begin{bmatrix} k_{n1} \\ k_{n2} \\ k_{n3} \\ k_{n4} \\ k_{n5} \end{bmatrix} \leq \begin{bmatrix} \frac{\pi}{(\frac{t_f}{n_s n_b})^4} \\ \frac{\pi}{(\frac{t_f}{n_s n_b})^3} \\ \frac{\pi}{(\frac{t_f}{n_s n_b})^2} \\ \frac{\pi}{(\frac{t_f}{n_s n_b})} \\ \pi \end{bmatrix} \quad (101)$$

where n_b is the number of burns assumed in the transfer. For the continuous thrust transfers, n_b is assumed to be 1. A particular difficulty with this method is that discontinuities are allowed at the nodes between polynomials. Continuity can be en-

forced; however, in this investigation, the polynomial end-points are left unrestricted as to allow for potentially fast changes in the control as well as greater freedom for the PSO to optimize. In implementing PSO, a penalty function method as defined in equation (82) is used to construct the modified scalar cost function.

Due to the stochastic elements in the PSO algorithm, the random number generator seed is set to 0 in MATLAB[®] at the beginning of a simulation [68]. The random number generation algorithm is left as the default ‘twister’ algorithm. This is used to ensure results are repeatable by the user and by the reader. In addition, the cognitive and social coefficients, c_1 , and c_2 , in the PSO algorithm are set to 1.49445 per recommendations from Conway [6]. For PSO, optimality conditions are not checked; the final solution is the global best candidate solution once the iteration count reaches k_{max} . The next section discusses NLP improvement on the PSOIGs.

3.1.3 NLP Improvement

The NLP solver used in this investigation is the function *fmincon* in the MATLAB[®] Optimization ToolBox. The function *fmincon* allows the user to supply the nonlinear cost and nonlinear constraint function as well as choose the specific algorithm the solver uses [68]. For the trajectories in this chapter, the “interior-point” algorithm is chosen due to the number of design parameters that *fmincon* must handle. The choice is based on recommendations found in the Mathworks “Choosing a Solver” documentation stating that the interior-point algorithm “handles large, sparse problems, as well as small dense problems” [72]. The PSOIGs are given to NLP by discretizing the polynomial-based control time history and supplying the time parameters. The NLP solver then optimizes the control time histories at each time increment. The time step is initially set to be “reasonably” small such that the NLP solver is dealing with approximately 100 design parameters.

To increase the granularity of the resulting answers, a simple mesh refinement scheme is used. The previously converged answer is doubled in size with the new intermediate control values set equal to the average value of the two bordering control values. The diagram in Figure 23 depicts the refinement strategy.

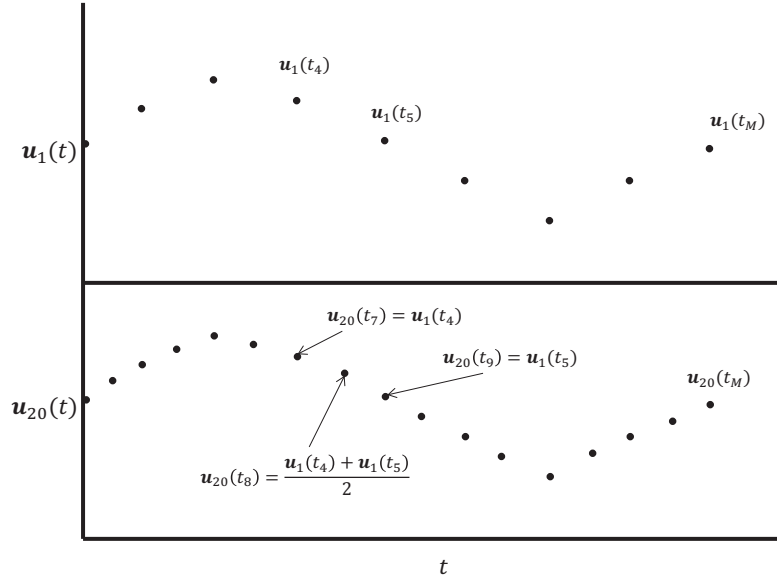


Figure 23. Diagram of a grid refinement scheme

After a few iterations of grid refinement, the trajectories should have an acceptable level of granularity. The size of the design parameters array tends to be on the order of 10^3 and 10^4 . The level of desired fidelity as well as allowable computation time dictates the number of refinements completed. Due to the size of the problem after a few iterations, solvers that are equipped to handle large sparse problems are recommended. Using *fmincon*, the “interior-point” algorithm performs well; however, another recommended solver for large problems, due to its exploitation of the sparse matrices, is the Sparse Nonlinear Optimizer (SNOPT) [73]. Given an inevitably sparse problem after grid refinement, gradient information is not manually provided to *fmincon* for the trajectories in this chapter.

In employing *fmincon*, the function tolerance ‘FunTol’ and constraint tolerance

‘ConTol’ are left at the default values of 10^{-6} . The function tolerance and constraint tolerance directly correspond to the tolerances that must be met for optimality. In order to prevent premature termination of the function, the maximum function iterations ‘MaxFunIter’ is set to 10,000. The maximum number of function iterations becomes relevant when the number of design parameters is on the order of 10^4 . All NLP improvements in this chapter are executed successfully with an *fmincon* exit flag of 2. This error flag means that a local minimum is found such that the constraints are satisfied to the designated constraint tolerance and a change in the free-variables causes a change in the objective function smaller than the function tolerance.

3.2 Results

The results in this section are organized such that the TPBVP boundary conditions are first presented followed by generation of the initial guess via PSO. Lastly, the NLP improved solution is given. In order to provide a measure of the utility of PSO for each test case, an effort to manually create a viable initial guess for the NLP solver is attempted. The manually generated initial guesses are simple and are an attempt to model default values that may be used by a designer. In certain instances, a previously generated NLP solution from a simpler case is input as the initial guess. The test cases are conducted in the following order:

1. Continuous low-thrust LEO to GEO planar transfer
2. Low-thrust multiple-burn LEO to GEO planar transfer
3. Continuous low-thrust LEO to GEO non-coplanar transfer
4. Continuous low-thrust LEO to GEO non-coplanar transfer with oblate Earth effects

5. Low-thrust multiple-burn LEO to GEO non-coplanar transfer with oblate Earth effects

The test cases are constructed to gradually increase the complexity of the transfers. This serves the purpose of systematically testing the limits of PSO. A comparison of the continuous low-thrust LEO to GEO non-coplanar transfers with and without oblate Earth affects is provided. Lastly, trajectories are compared in terms of fuel usage against the very-low-thrust and impulsive burn equivalent transfers. As global optimality is not guaranteed in this investigation, all results can only be labeled as “locally optimal” to the initial guess provided to the NLP solver.

3.2.1 Continuous Thrust Planar LEO to GEO Transfer

The first test case is the simplest as it is a low-thrust continuous transfer from LEO to GEO. Since the trajectory is between two coplanar, equatorial orbits, the control time history only requires the in-plane angle α . The initial and terminal conditions are given in Table 6. The satellite is propagated using the equations in (92). The next section discusses PSOIG generation given these boundary conditions.

Table 6. Initial and terminal conditions for LEO to GEO coplanar transfer

	Initial Conditions	Terminal Conditions
r (km)	6,678.137	42,164
V_r (km/s)	0	0
V_θ (km/s)	7.7257	3.0747
e	0	0
i (deg)	0	0

3.2.1.1 PSO Initial Guess Generation

To generate the initial guess with PSO, the pointing angle time history $\alpha(t)$ is segmented into five fourth-order polynomials. Each of the polynomials spans twenty-six time steps. Employing the Runge-Kutta integration scheme introduced in section 2.3.4, twenty-six time-steps corresponds to 182 control point insertions. Across all five polynomials, 910 independent control parameters are used in the numerical integration of the trajectory. With each fourth-order polynomial consisting of five coefficients, twenty-five polynomial coefficients are to be optimized. In addition to the polynomial coefficients, the parameter array $\mathbf{P}$ contains the time of flight t_f . A total of twenty-six parameters make up the free-variable array.

$$\mathbf{P} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & \cdots & k_{55} & t_f \end{bmatrix}^T \quad (102)$$

The bounds on the polynomial coefficients are set by equation (101), and the time of flight must be between 1 and 100 TU (1 TU = 14.41 minutes). The equality constraints enforced in the trajectory are

$$\phi(x^{i,k}) = \begin{bmatrix} V_r(t_f) \\ V_\theta(t_f) - \sqrt{\frac{\mu}{r_2}} \\ r(t_f) - r_2 \end{bmatrix} = \mathbf{0} \quad (103)$$

where r_2 is the final orbit radius of 42,164 km. The constraints are set up so that the terminal conditions correspond to a circular orbit of the correct radius. Given the constraints, the modified scalar cost function, $\tilde{J}$, being minimized is defined as the following

$$\tilde{J} = \|\phi_1\| + \|\phi_2\| + \|\phi_3\| + 0.01t_f \quad (104)$$

The coefficient for t_f was experimented with until reasonable results were generated. A small coefficient is needed in order to scale the time of flight so that it is on the same order of magnitude as the constraints. A high time of flight coefficient yielded a minimized time of flight with a complete disregard by the optimizer to satisfying the constraints. The resulting PSOIG using 100 particles (N_p) and 1,000 max iterations (k_{max}) is shown in Figure 24.

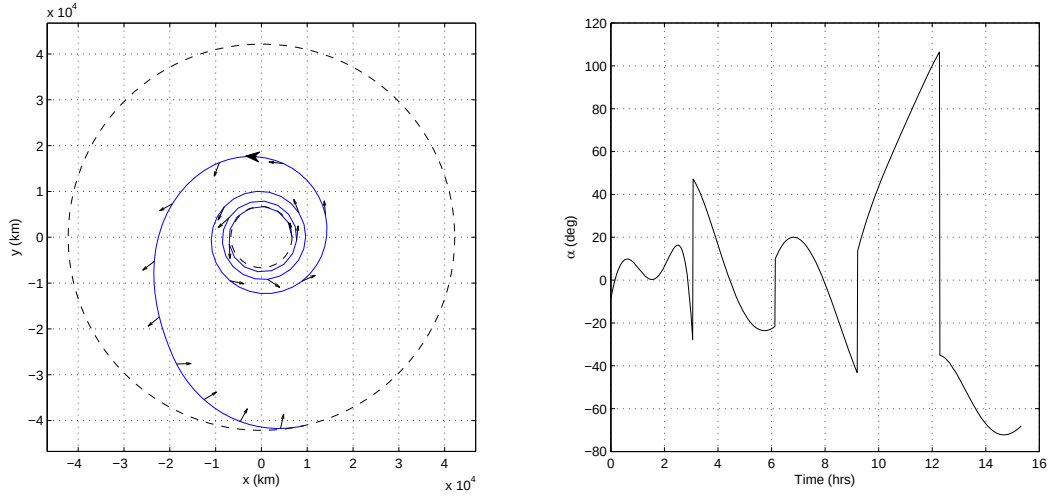


Figure 24. PSO generated continuous thrust, LEO to GEO, coplanar transfer, 0.639 day transfer (black arrows correspond to thrust pointing directions)

The small black arrows denote the pointing angle measured with respect to the local horizon at the time corresponding to the point at the base of the arrow. The larger black arrowheads in all of the plots in this investigation are used to indicate the direction of motion of the spacecraft. The pointing angle qualitatively depicts oscillatory behavior, though the discontinuities at the end-points of the polynomials significantly affect the smoothness of the control history. The next section discusses NLP improvement on this initial guess and presents the results.

3.2.1.2 NLP Improvement

In order to pass the PSOIG into *fmincon*, the number of discretized segments is set equal to 100. Using the polynomial coefficients, the control value corresponding to the times at each of the 100 increments is calculated using equation (99). With the control values and the time of flight from PSO, the initial guess $\mathbf{P}$ takes on the form of a 101×1 array below

$$\mathbf{P} = \begin{bmatrix} u_{t_1} & u_{t_2} & u_{t_3} & \cdots & u_{t_{100}} & t_f \end{bmatrix}^T \quad (105)$$

The bounds on the pointing angle are now narrower and must lie between ± 90 degrees. The same bound of 1 to 100 TU on time of flight is still enforced. The function *fmincon* is well equipped to handle constraints, and as such, the constraints and cost function can be handled independently. The equality constraint vector takes on the same form as in equation (103) and the cost function being minimized is simply

$$J = t_f \quad (106)$$

In order to prevent a singular arc issue, path constraints are also added as a separate inequality constraint array. The inequality constraints are formulated such that the time rate of change of the control does not exceed a certain value. In this investigation, thirty degrees per second is set as the maximum. Thirty degrees per second is chosen because this is faster than a typical spacecraft rotation rate, and sufficiently bounds the rate of change of the control [74]. To enforce this constraint, the rate of change between each supplied control value is calculated by dividing the difference in two sequential control values by the change in time. It is important to note that the time step between controls is one-seventh the time step between states given a three-step RK integration scheme. The inequality constraint vector is

a separately defined array in *fmincon* and takes on the form below.

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_{tf} \end{bmatrix} \leq 30 \text{ deg/s} \quad (107)$$

After supplying the PSOIG to *fmincon* and after successful convergence designated by an exit flag of 2, the fidelity of the trajectory is increased using the continuation scheme in Section 3.1.3. The resulting trajectory and the pointing angle time history are shown in Figure 25 and Figure 26, respectively.

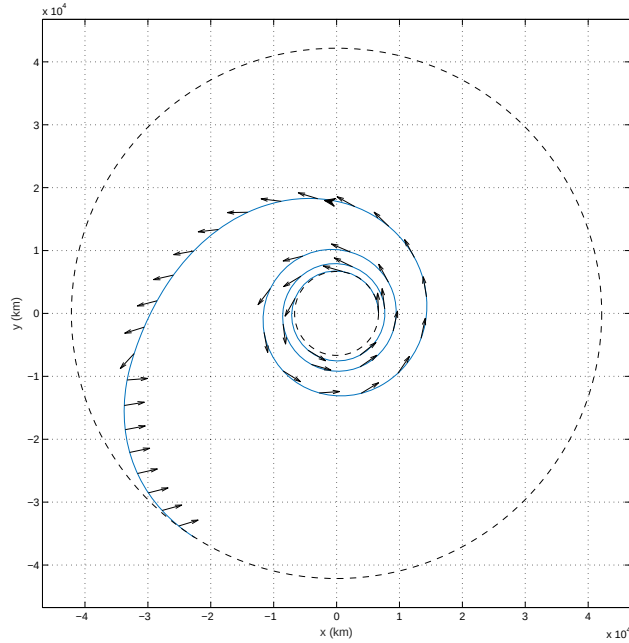


Figure 25. NLP improved continuous thrust, LEO to GEO, planar transfer, 0.607 day transfer (black arrows correspond to thrust pointing directions)

The optimal continuous pointing angle is oscillatory with a frequency corresponding to the period of each subsequent revolution. This is exemplified by comparing the approximately 3.5 revolutions in the trajectory to the roughly 3.5 periods in the control. Also, it appears the amplitude of the pointing angle grows with each rev-

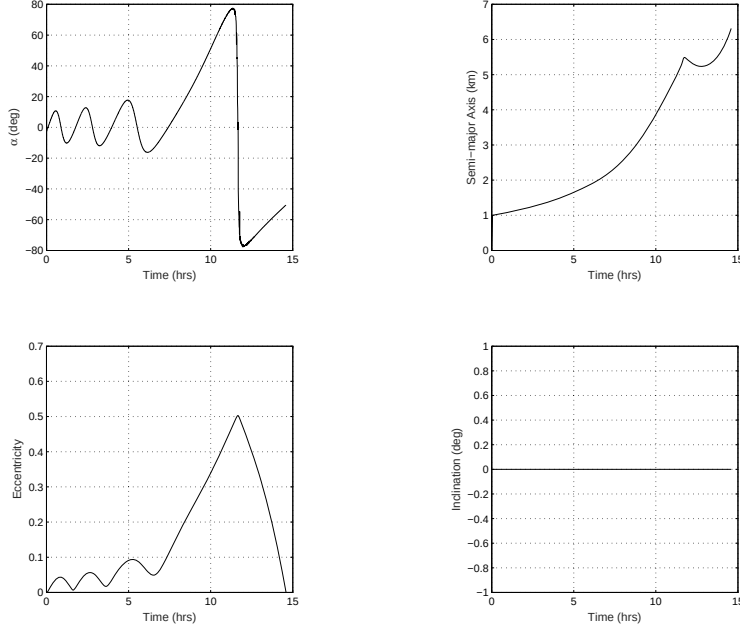


Figure 26. Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, planar transfer

olution, with a final oscillation attaining nearly 80 degrees. The amplitude of the oscillations in the control is a visual depiction of the level of inefficiency in the trajectory. The inefficiency can be explained by examining the instantaneous COEs, or osculating orbital elements. The time histories of the COEs show that the eccentricity of the transfer remains near zero for the first three revolutions and then increases dramatically when the satellite is nearing the target altitude. Ideally, the eccentricity would remain zero for the entire trajectory and the performance in terms of ΔV would approach that of Edelbaum's very-low-thrust spiral transfer. Using Edelbaum's equation for the ΔV between these orbit sizes yields an optimal transfer at 4.65 km/s with a time of flight (assuming $n_0 = 0.001g$) of 53.8 days. However, the optimal continuous-thrust transfer at this acceleration level ($n_0 = 0.01g$) is 0.608 days with a ΔV of 6.019 km/s. There is definitely a tradespace between ΔV and time of flight when using continuous thrust and varying the thrust acceleration level. As a form of results validation, the ΔV at this orbit transfer ratio as well as thrust

acceleration level matches that of the results shown Vallado [52]. Also, a very similar trajectory at the same thrust acceleration level is given in Conway [6].

The manual initial guess also supplied to *fmincon* to gauge the NLP convergence window was an array of 0 degree α angles. A guess-and-check method was required for choosing a reasonable time of flight and time step size such that *fmincon* converged. Conversely, PSO required user determination of appropriate constraint weighting. For this simple trajectory, both courses of action required the same level of user effort and yielded the same final trajectory.

3.2.2 Multiple-burn Planar LEO to GEO Transfer

The next test case relaxes the continuous-burn restriction and allows for intermittent coast arcs between burn arcs. The trajectory is still between two coplanar, equatorial orbits, and the control time history only requires the in-plane angle α . The boundary conditions for the TPBVP are the same as in Table 6. The satellite is still propagated using the same equations (92).

3.2.2.1 PSO Initial Guess Generation

For the PSOIG generation, the pointing angle time history, $\alpha(t)$, is segmented into five individual burn arcs. Each of the five polynomials defines the control for an individual burn segment. The same amount of nodes and control inputs are used as in the previous continuous-thrust example. The single time variable used in the previous test case is now expanded into the times of each of the five burns as well as the four coast arcs between them. A total of thirty-four parameters make up the free-variable array.

$$\mathbf{P} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & \cdots & k_{55} & \mathbf{T} \end{bmatrix}^T \quad (108)$$

where $\mathbf{T}$ is defined as

$$\mathbf{T} = \begin{bmatrix} t_{b1} & t_{b2} & \cdots & t_{b5} & t_{c1} & t_{c2} & \cdots & t_{c5} \end{bmatrix}^T \quad (109)$$

The total burn time is a sum of all the burn times, t_{bi} , and the total flight time is a sum of all the burn times and coast times, t_{ci} . The bounds on the pointing angles are still defined by equation (101); however, an n_b value of 5 must now be used. The burn times and coast times are bounded between 0.0001 and 30 TU to allow for enough variability in solutions, but also to prevent a single segment from dominating and causing a collapse in all of the other arcs. The constraints are defined by equation (103). The modified cost function is now a function of the constraints as well as the total burn time, t_{bt} , and total coast time, t_{ct} .

$$\tilde{J} = \|\phi_1\| + \|\phi_2\| + \|\phi_3\| + 0.01t_{bt} + 0.001t_{bc} \quad (110)$$

The 0.01 coefficient in front of the total burn time functions similarly to the coefficient in front of the time of flight in the previous case; however, it is helpful to factor in the total coast time into the modified cost with a smaller coefficient. This is done to minimize superfluous coasting. In practice, the PSO algorithm, given the modified cost function, seeks to minimize the burn time before minimizing the coast time. The values of the constraint weighting coefficients were user-determined using trial-and-error. The resulting PSOIG using 500 particles and 1,000 max iterations is shown in Figure 27.

The pointing angle displays oscillatory behavior with discontinuities at the coasting arcs. Even though four coast arcs were allotted, the converged trajectory only uses two. It is interesting to note that the most inefficient portion of the previous transfer, marked by the highest amplitude in the pointing angle, is now a coasting

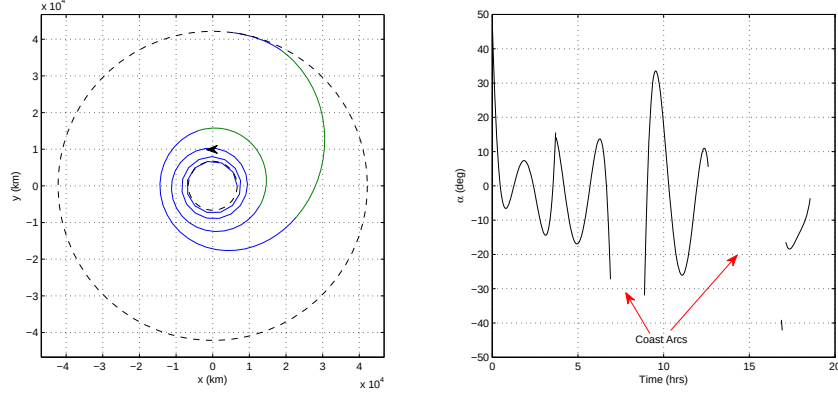


Figure 27. PSO generated multiple-burn, LEO to GEO, planar transfer, 0.773 day transfer

segment. This implies that the inefficiency in the previous trajectory is best improved upon by replacing the problem section with a coast arc. The next section discusses NLP improvement of this guess and presents the locally optimal results.

3.2.2.2 NLP Improvement

Using the polynomial coefficients from the PSOIG, the control values correspond to twenty-five time increments along each of the burns. This means that the time step size for each burn depends on the length of the burn. Short burn segments have smaller times steps and vice versa. With the control values and the burn and coast segment durations included, the initial guess $\mathbf{P}$ takes on the form of a 109×1 array below

$$\mathbf{P} = \begin{bmatrix} u_{t_1} & u_{t_2} & u_{t_3} & \cdots & u_{t_{100}} & T \end{bmatrix}^T \quad (111)$$

The bounds are ± 90 degrees for the control parameters and between 0.0001 and 30 TU for the burn and coast times. The definition of T is given in equation (109) The equality constraint equations remain the same from equation (103) and inequality constraints, enforced along the burn arcs, are the same as in equation (107). The

total time of flight is minimized in the scalar cost function J .

$$J = t_{b1} + t_{b2} + \cdots + t_{b5} + t_{c1} + t_{c2} + \cdots + t_{c5} \quad (112)$$

The converged trajectory, control time history, and osculating elements are shown in Figures 28 and 29. The black arrows are removed from the finite-burn plots in order to declutter the plot as well as allow the reader to better distinguish between the burn and coast segments. The thrust pointing angle displays similar oscillatory behavior to the continuous-thrust case; however, the amplitude of the oscillations are much smaller. The solution uses three instead of the two coasting arcs used in the PSOIG and burns for 0.479 days with a total flight time of 0.904 days. The ΔV significantly improved by 1.56 km/s from the continuous case to 4.457 km/s. This is better than the Edelbaum very-low-thrust solution but is still worse in terms of ΔV than the impulsive transfer that only requires 3.892 km/s. Ideally, an infinite number of impulsive burns would be competed at the instantaneous perigee of each revolution such that the orbit is very gradually raised. This hypothetical transfer would approach the ΔV of the impulsive transfer. These results are consistent with numbers predicted in Pelouch for a multi-burn transfer between LEO and GEO [54].

Many different non-PSOIGs were supplied to *fmincon* in order to gauge the sensitivity of the problem as well as the convergence window. Supplying *fmincon* with an initial guess equivalent to the converged continuous-thrust solution did not work because the solver tended not to depart the given solution. However, when non-zero values on the order of 1 TU were input for each of the coast durations, the solver was able to converge on a very similar solution to that of Figure 28. Using the PSOIG, a few iterations on the constraint coefficients in equation (110) were necessary before an acceptable initial guess was generated. Both methods are viable; however, the non-PSOIG method built off of the solution from the previous test case. Such

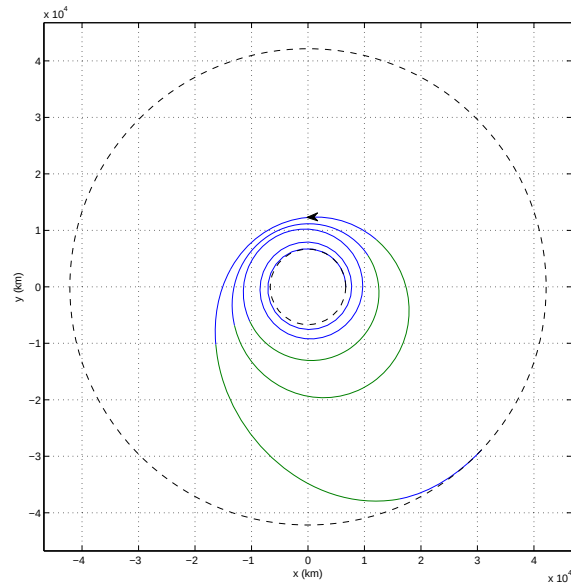


Figure 28. NLP improved multiple-burn, LEO to GEO, planar transfer, 0.904 day transfer

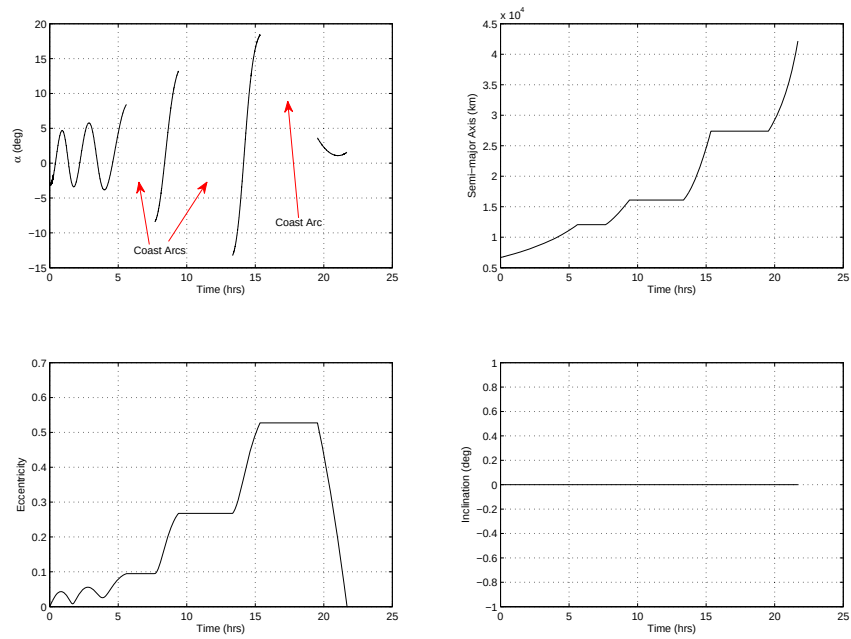


Figure 29. Control time history and osculating elements for NLP improved multiple-burn, LEO to GEO, planar transfer

a previously generated solution may not be available to the user depending on the design scenario. Conversely, despite iteratively searching for acceptable constraint coefficients, the PSOIG method does not require an initial guess for the motion of the spacecraft.

3.2.3 Continuous Thrust Non-Coplanar LEO to GEO Transfer

For this next test case, out-of-plane motion is now added to the burns in order to allow for inclination changes and for added complexity. The transfer starts at an inclined LEO altitude orbit and terminates at an equatorial GEO altitude orbit. The table below shows the initial and terminal conditions.

Table 7. Initial and terminal conditions for LEO to GEO non-coplanar transfer

	Initial Conditions	Terminal Conditions
r (km)	6,678.137	42,164
V_r (km/s)	0	0
V_θ (km/s)	7.7257	3.0747
e	0	0
i (deg)	28.5	0

An initial inclination of 28.5 degrees is chosen in order to simulate a launch from Cape Canaveral to LEO where a post orbit insertion transfer to GEO is still required. These non-coplanar transfers are conducted using MEEs due to the eventual inclusion of J_2 . Table 8 shows the boundary conditions using MEEs.

Table 8. Initial and terminal conditions for LEO to GEO non-coplanar transfer using MEEs

	Initial Conditions	Terminal Conditions
p (km)	6,678.137	42,164
f (deg)	0	0
g (deg)	0	0
h (deg)	0	0
k (deg)	14.551	0
L (deg)	0	Free

The L parameter is left free as the position on the final orbit is not constrained. The trajectories are propagated using the EOMs given in equation (95) with the acceleration equal to $\mathbf{a}_{Thruster}$ when J_2 is not included and equal to $\mathbf{a}_{Thruster} + \mathbf{a}_{J2}$ when it is included.

3.2.3.1 PSO Initial Guess Generation

To generate the initial guess using PSO, five fourth-order polynomials are still used for $\alpha(t)$; however, an additional five polynomials must be included to approximate $\beta(t)$. Each polynomial spans twenty-six time-steps, thus, with 182 control point insertions and two control parameters at each step, there are now 1,820 independent control parameters used in the numerical integration of the trajectory. Having ten polynomials translates into fifty-one free-variables where the final parameter is the time of flight. The $\mathbf{P}$ array is defined as

$$\mathbf{P} = \begin{bmatrix} k_{11} & \cdots & k_{15} & k_{21} & \cdots & k_{55} & q_{11} & \cdots & q_{15} & q_{21} & \cdots & q_{55} & t_f \end{bmatrix}^T \quad (113)$$

where q_{nj} is the β angle polynomial coefficient corresponding the to n th polynomial and the j th coefficient. The bounds on the coefficients are calculated using equation (101) with n_b equal to 1 and t_f bounded between 1 and 100 TU. The equality constraints, after the transformation to MEEs, now take on the form

$$\phi(x^{i,k}) = \begin{bmatrix} p(t_f) - r_2 \\ f(t_f) \\ g(t_f) \\ h(t_f) \\ k(t_f) \end{bmatrix} = \mathbf{0} \quad (114)$$

With the five boundary constraints defined, the modified cost function is defined very similarly to equation (104).

$$\tilde{J} = \|\phi_1\| + \|\phi_2\| + \|\phi_3\| + \|\phi_4\| + \|\phi_5\| + 0.01t_f \quad (115)$$

Again, the time of flight must be scaled down such that it is on the same order of magnitude as the constraints. The resulting PSOIG using 100 particles and 1,000 max iterations is shown in 3-D in Figure 30. The x - y plane projection is shown in Figure 31.

The black arrows in the 3-D view depict the spatial thrust pointing direction. In the x - y view, the black arrows only depict the in-plane angle, α . The control time histories in Figure 32 communicate a high volatility in the pointing angles. It is difficult to discern the qualitative nature of the control time history; however, a large β angle appears near the end of the trajectory.

Before conducting NLP improvement on the PSOIG, the same trajectory is solved again with PSO, but with the inclusion of $\mathbf{a}_{J2}$. The same methodology for the previously generated guess is used; the resulting trajectory and control time history

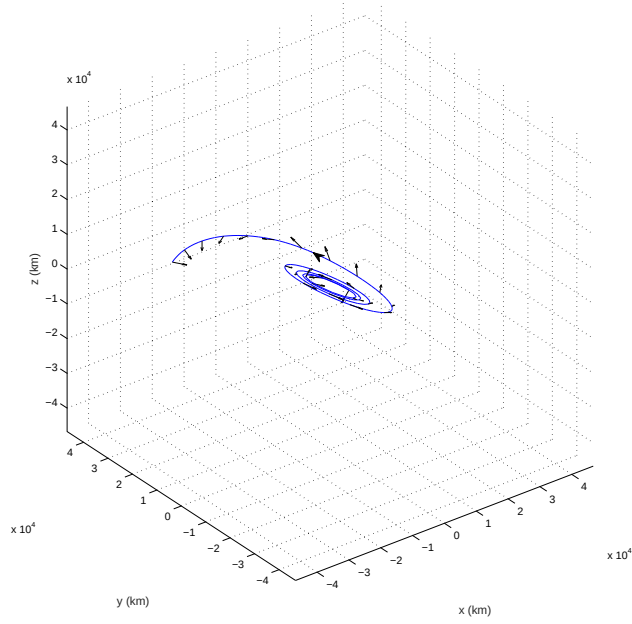


Figure 30. PSO generated continuous thrust, LEO to GEO, non-coplanar transfer, 3-D view, 0.664 day transfer (black arrows correspond to thrust pointing directions)

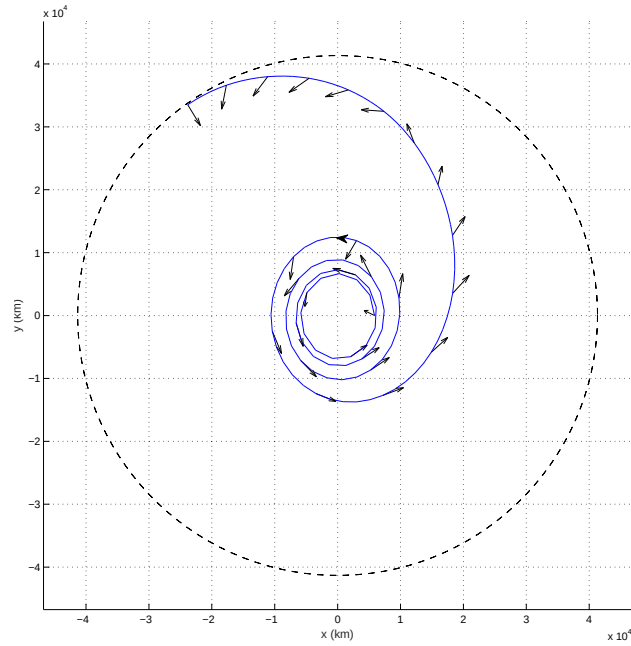


Figure 31. PSO generated continuous thrust, LEO to GEO, non-coplanar transfer, x - y view, 0.664 day transfer (black arrows correspond to thrust pointing directions)

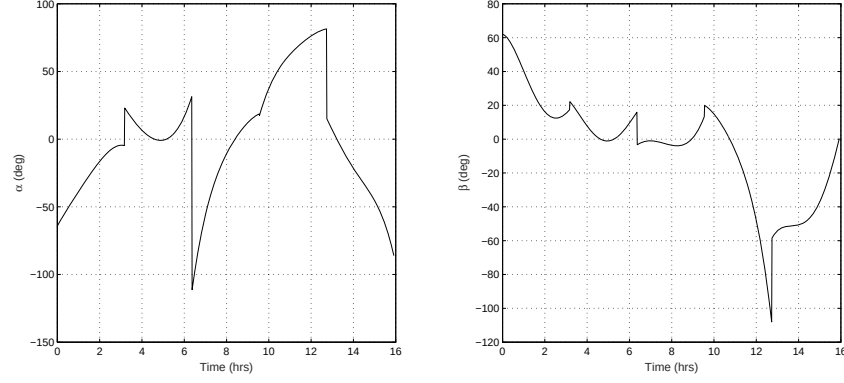


Figure 32. Control time history for PSO generated continuous thrust, LEO to GEO, non-coplanar transfer

are shown in Figures 33, 34, and 35. The results are similar to those of the non- J_2 PSOIG; however, the oscillatory nature of the pointing angles is more discernible in this case. A large α and β angle is required near the termination of the trajectory. The two PSOIGs are given to the NLP solver for improvement in the next section.

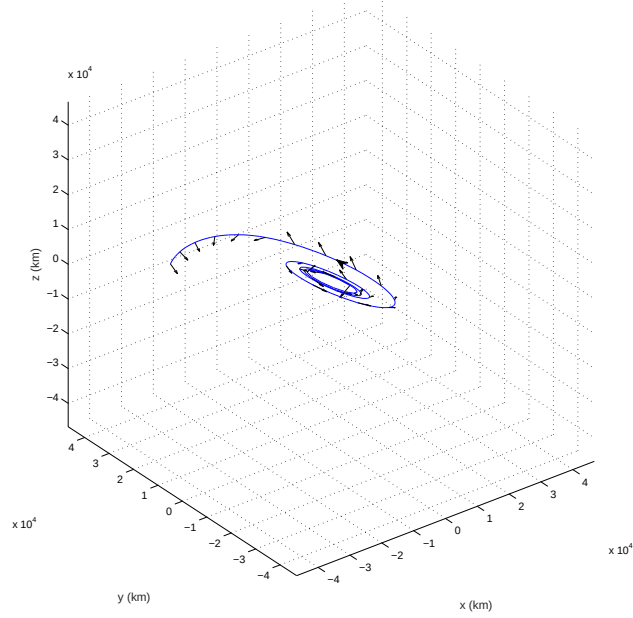


Figure 33. PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J_2 perturbation, 3-D view, 0.654 day transfer (black arrows correspond to thrust pointing directions)

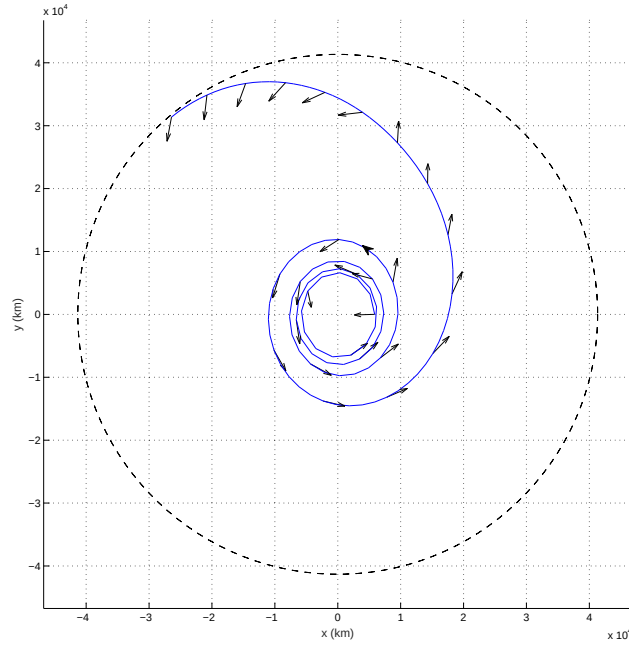


Figure 34. PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.654 day transfer (black arrows correspond to thrust pointing directions)

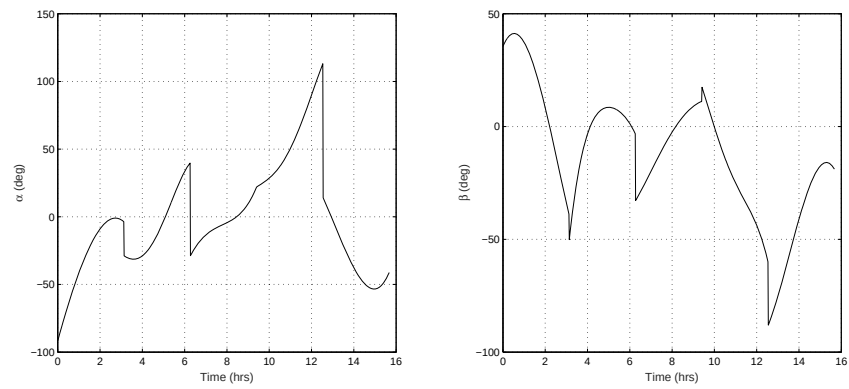


Figure 35. Control time history for PSO generated continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation

3.2.3.2 NLP Improvement

Passing the PSOIG for improvement to *fmincon* now requires handling both the in-plane and out-of-plane control time histories. After calculating the control values for each time increment using the PSOIG polynomials coefficients, the parameter array $\mathbf{P}$ takes on a similar form to equation (105), but with 100 in-plane and 100 out-of-plane variables, denoted u_t and w_t , respectively. The control parameters are then followed by t_f in the parameter array.

$$\mathbf{P} = \begin{bmatrix} u_{t_1} & u_{t_2} & \cdots & u_{t_{100}} & w_{t_1} & w_{t_2} & \cdots & w_{t_{100}} & t_f \end{bmatrix}^T \quad (116)$$

The bounds are ± 90 degrees for the control parameters and between 1 and 100 TU for burn time. The cost function is still to minimize t_f and the path constraints are enforced such that $\dot{u}_t$ remains less than or equal to thirty degrees per second. The equality constraints are defined by equation (114). After successful convergence and increasing the fidelity of the transfers, the resulting trajectory without J_2 is shown in Figures 36, 37, and 38.

The converged trajectory has a time of flight of 0.65 days, which results in a ΔV of 6.247 km/s. From the 3-D view, it is easy to discern the 28.5 degree inclination of the initial orbit. Also, it appears that the addition of an inclination change from LEO to GEO does not significantly affect the in-plane pointing when compared to the planar transfers nor is there a large difference in time of flight. As a result, the ΔV increase from the planar trajectory is on the order of 200 m/s. Using a combined plane change at the second impulse in a two-impulse maneuver, the associated cost for just changing the inclination is 363 m/s. An explanation as to why the continuous thrust transfer requires less ΔV than the impulsive optimum for the inclination change is that the excess in-plane thrust is diverted into the required out-of-plane corrections. This would effectively lower the required max amplitude of the in-plane angle as well

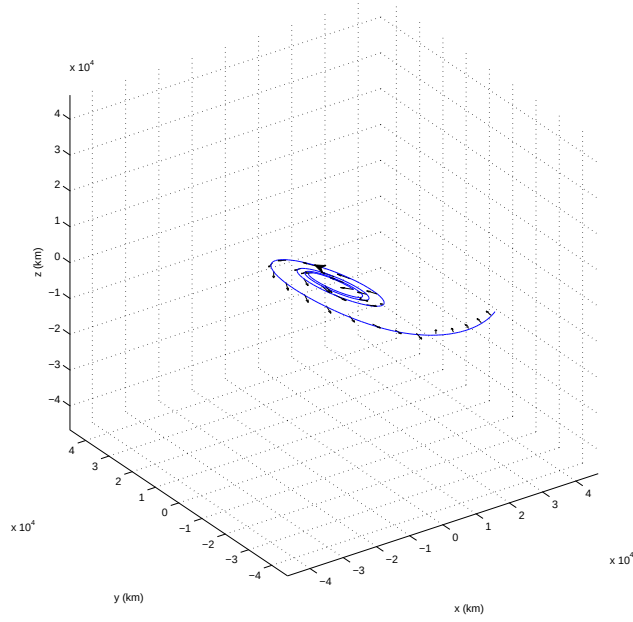


Figure 36. NLP improved continuous thrust, LEO to GEO, non-coplanar transfer, 3-D view, 0.650 day transfer (black arrows correspond to thrust pointing directions)

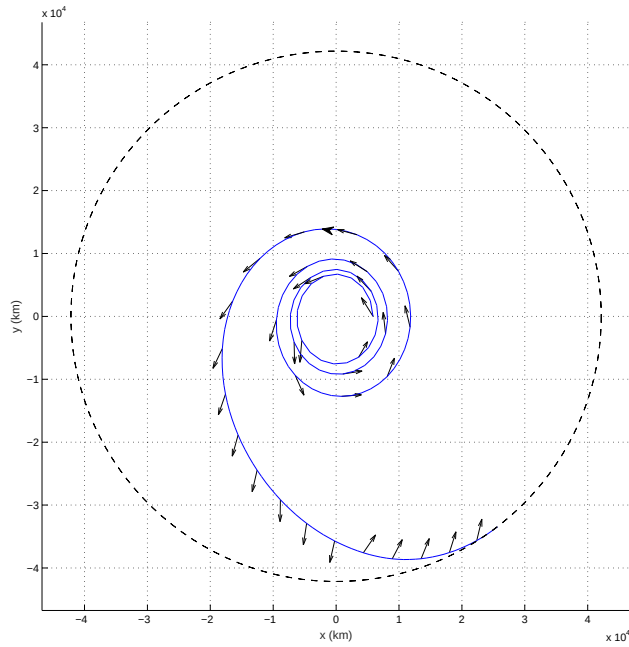


Figure 37. NLP improved continuous thrust, LEO to GEO, non-coplanar transfer, x - y view, 0.650 day transfer (black arrows correspond to thrust pointing directions)

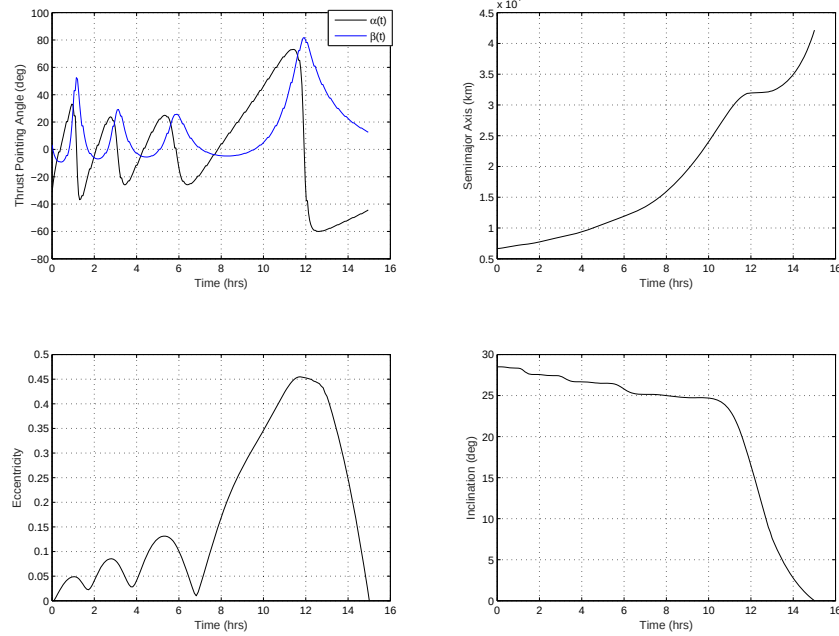


Figure 38. Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, non-coplanar transfer

as lower the cost of the plane change. These results are validated by an equivalent transfer conducted in Herman [75]. The NLP improved trajectory including the J_2 perturbation is shown in Figures 39, 40, and 41.

As expected, the inclusion of J_2 only slightly changes the optimal trajectory. Interestingly, the required ΔV is decreased by 18 m/s compared to the transfer without J_2 . A comparison of the osculating elements for both transfers is shown in Figure 42. The deviations in the optimal trajectories due to the inclusion of J_2 is mostly seen during the highest rates of change of the COEs. This is likely a result of the minor deviations in the models building up to be corrected in the “fast” portion or period of greatest variation in the COEs as well as the control amplitudes.

Manually creating an initial guess to supply to *fmincon* in this scenario was very difficult without using the PSOIG. Previous techniques of using zeroed control time histories and guessing at the time of flight did not work. The system was also very sensitive to the step size used. If a step size below 1.5 TU or greater than 2 TU

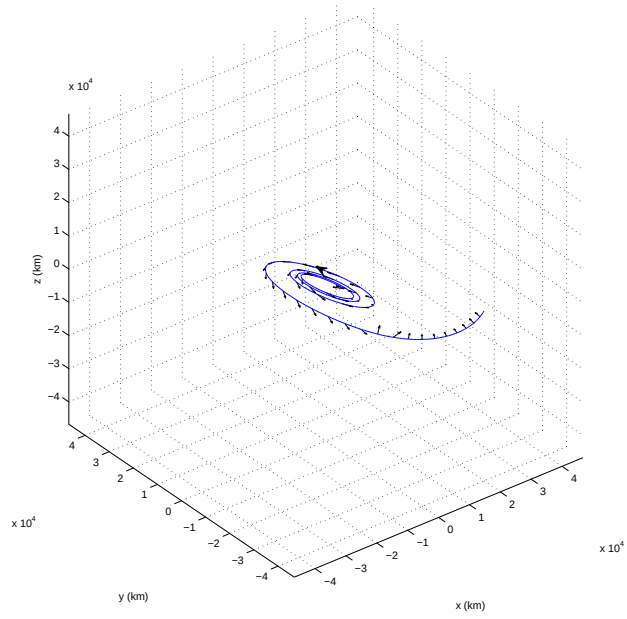


Figure 39. NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.624 day transfer (black arrows correspond to thrust pointing directions)

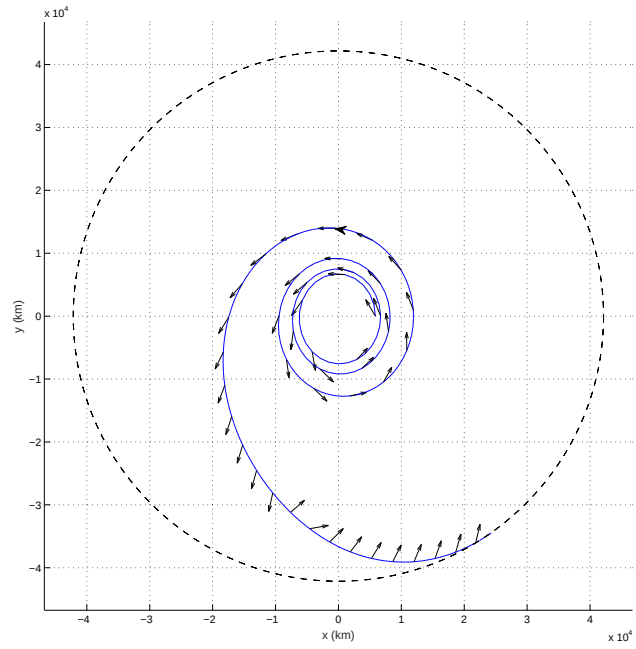


Figure 40. NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.624 day transfer (black arrows correspond to thrust pointing directions)

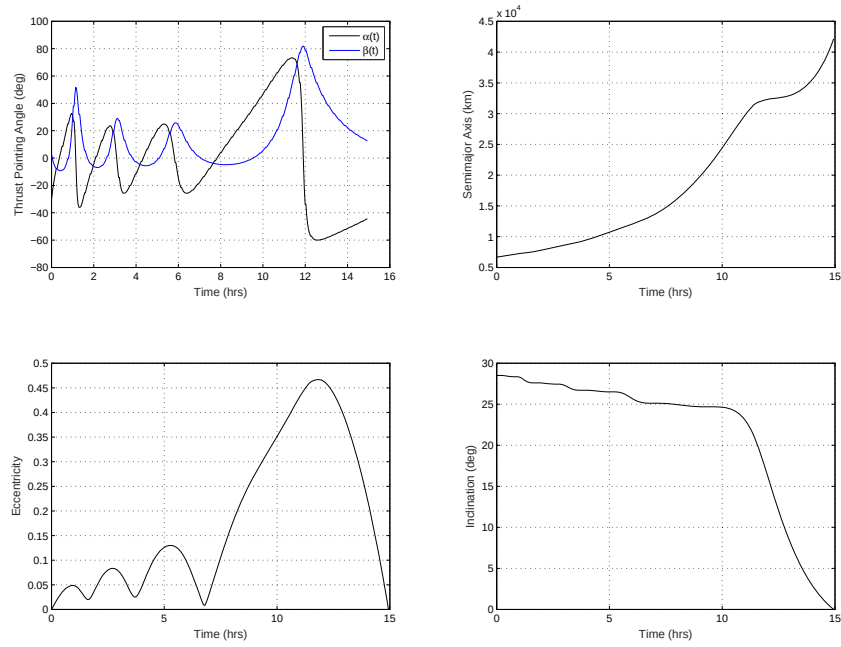


Figure 41. Control time history and osculating elements for NLP improved continuous thrust, LEO to GEO, non-coplanar transfer with J2 perturbation

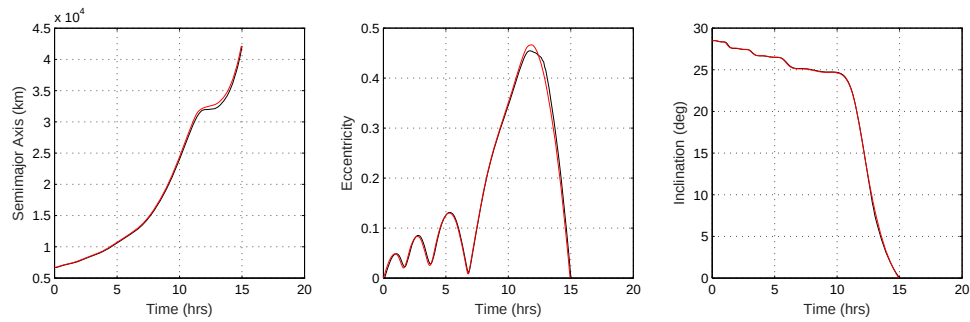


Figure 42. Comparison of osculating elements between NLP improved continuous thrust, LEO to GEO, non-coplanar transfers with and without J2 perturbation effects.

with zeroed control time histories was used as an initial guess, the NLP solver did not converge. The benefit of using the PSOIG was definitely seen in this test case. However, doubling the problem size due to an additional pointing angle affected PSO in that the initial guesses were not as close to the locally optimal solution. Even with PSO showing signs of being over-encumbered by a highly dimensioned search space, the initial guesses were still within the NLP convergence window.

3.2.4 Multiple-burn Non-Coplanar LEO to GEO Transfer

3.2.4.1 PSO Initial Guess Generation

This final scenario provides the highest level of complexity in this chapter as spatial motion, finite burns, and J_2 are all incorporated. Just as in the planar case, relaxing the continuous thrust constraint manifests itself as the addition of time parameters to define the duration of sequential burning and coasting arcs. With ten polynomials, five for the in-plane control and five for the out-of-plane control, each spanning twenty-six time steps, the PSO now needs to optimize a $\mathbf{P}$ of fifty-nine parameters. The constraints are defined by equation (114) and the bounds are defined by equation (101) where n_s is 5. The burn and coast times must be between 0.0001 and 30 TU. Lastly, the modified cost function is equivalent to equation (115). Optimizing the non-coplanar multiple-burn trajectory via PSO using 1,000 particles and 1,000 iterations yields the results in Figures 43, 44, and 45.

The thrust pointing angle is near continuous with very short coast arcs. Assuming the optimal solution takes advantage of larger coast arcs, it appears PSO is showing signs of difficulty with the dimension of search space. Even though the oscillatory behavior of the pointing angles is present, the trajectory does not improve upon the continuous thrust case with an increase in ΔV from 6.6295 km/s to 7.874 km/s. Higher iterations or a greater number of particles could be used, but at a greater

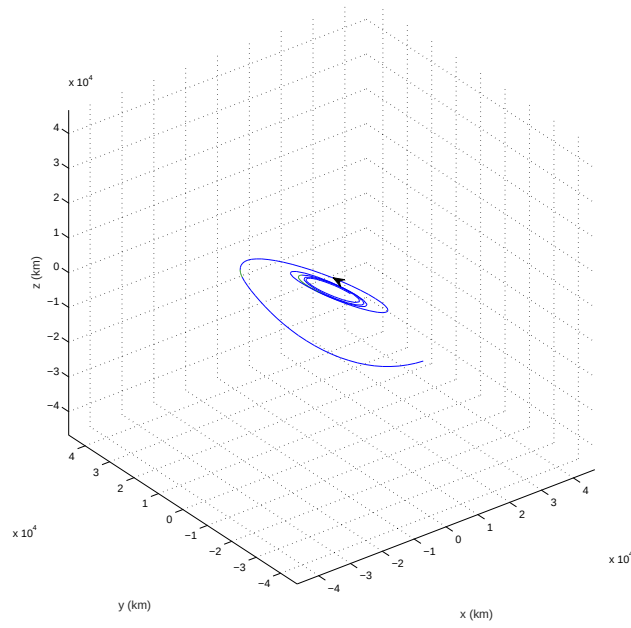


Figure 43. PSO generated multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.785 day transfer

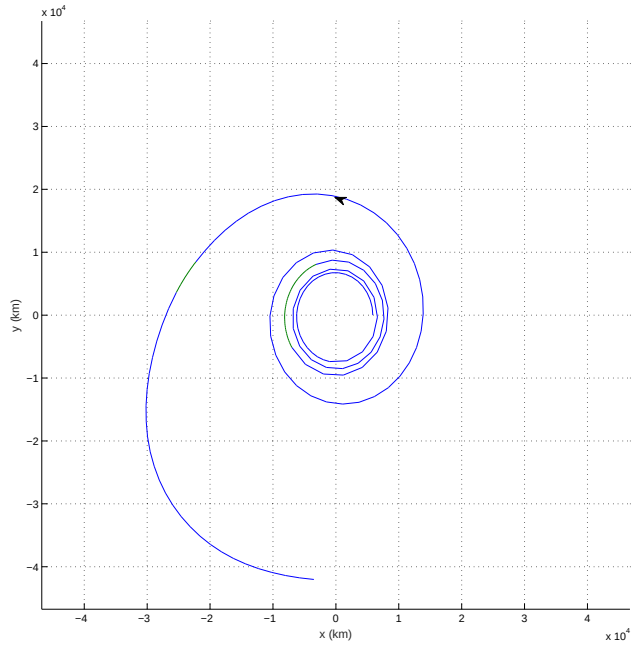


Figure 44. PSO generated multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.785 day transfer

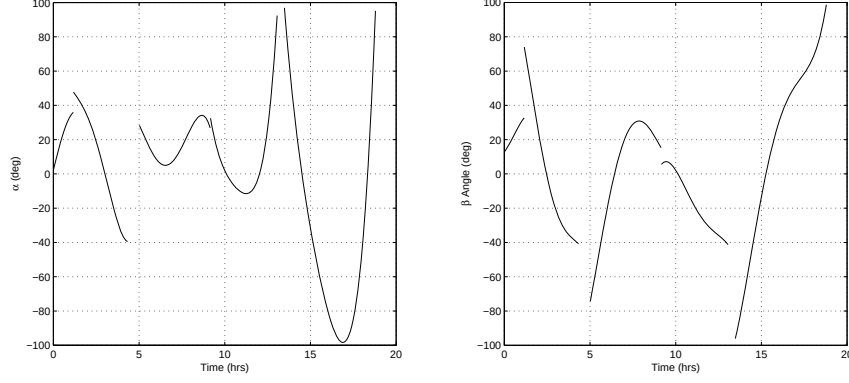


Figure 45. Control time history for PSO generated multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation

computation cost. This sub-optimal initial guess is provided to the NLP solver and is still within the convergence window of the NLP solver.

3.2.4.2 NLP Improvement

The NLP improvement follows closely with previous methodology. The parameter array $\mathbf{P}$ is very similar to equation (111) with the addition of the out-of-plane w_t values at each of the time increments.

$$\mathbf{P} = \begin{bmatrix} u_{t_1} & u_{t_2} & \cdots & u_{t_{100}} & w_{t_1} & w_{t_2} & \cdots & w_{t_{100}} & T \end{bmatrix}^T \quad (117)$$

The bounds are ± 90 degrees for the control parameters and between 0.0001 and 30 TU for the burn and coast times. The constraints are equivalent to equation (114), and the scalar cost function is equal to the total time of flight. The converged trajectory given the previously generated PSOIG is shown in Figures 46, 47, 48.

The level of fidelity is noticeably less than those of the previous NLP improvements. Even though the continuation scheme was conducted, due to the size of the problem, increasing the granularity of the plot could not be done while still yielding an *fmincon* exit flag of 2. The amount of iterations and function evaluations allowed

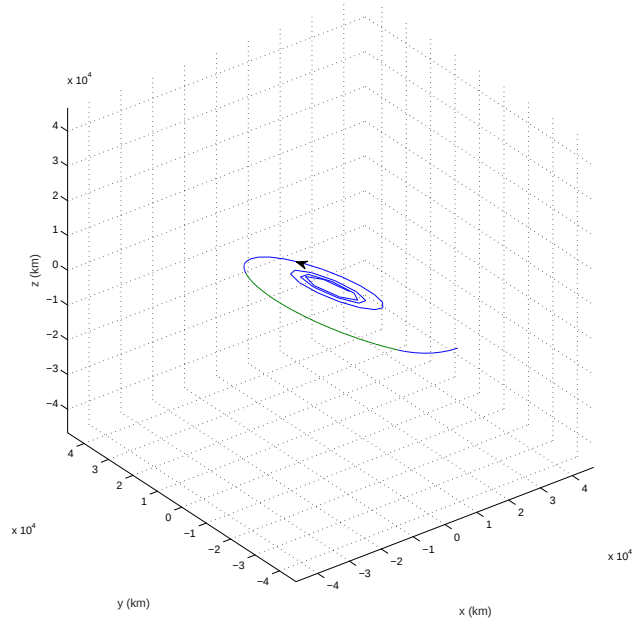


Figure 46. NLP improved multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, 3-D view, 0.802 day transfer

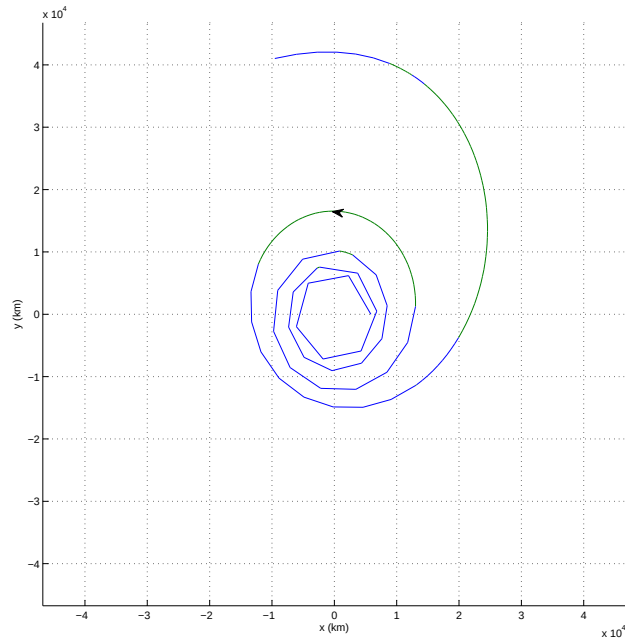


Figure 47. NLP improved multi-burn, LEO to GEO, non-coplanar transfer with J2 perturbation, x - y view, 0.802 day transfer

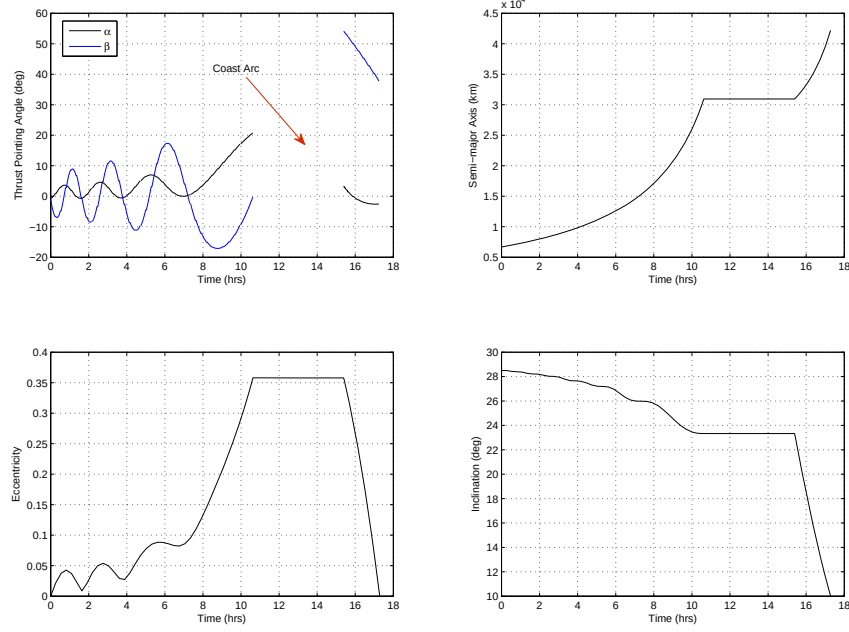


Figure 48. Control time history and osculating elements for NLP improved multiple-burn, LEO to GEO, non-coplanar transfer with J2 perturbation

could have been increased, but doing so would have been computationally expensive. The lower fidelity plot after successful convergence is presented instead.

The results show an improvement in ΔV from the continuous thrust non-coplanar transfer. This trajectory requires 0.513 days of burn time and lasts 0.802 days total. The burn time translates into 4.85 km/s, down from the 6.22 km/s of the continuous thrust transfer. The transfer only utilizes two coasting arcs, which is a local optimum given the supplied PSOIG that only has one significant coast.

For this case, successful convergence was also acquired using the continuous thrust trajectory as an initial guess. This technique, however, takes advantage of knowledge gained from a previously conducted optimization scenario. Arguably, PSO performs well with little to no knowledge about the optimal trajectory required.

3.2.5 Practicality of the Trajectories

The methodology employed demonstrated success in calculating locally optimal minimum-fuel trajectories. A valid question is to then ask: how practical are these trajectories to actually fly? To aid in the following analysis, Table 9 provides the quantitative characteristics of all of the calculated trajectories as well as the equivalent analytical optimum solutions for the very-low-thrust and impulsive thrust levels.

The table, in addition to the burn time, time of flight, and the ΔV of the converged trajectories, provides the mass fraction, m_f/m_0 , and propellant mass, m_p , required to conduct the transfer assuming a 3,000 kg final mass and the I_{sp} corresponding to the thrust acceleration level ($n_0 = 0.01g$). The 3,000 kg final mass is an assumed on-orbit final mass for a typical GPS satellite. In other words, this mass is the final mass, m_f , once the spacecraft has reached its final mission altitude. Actual payload mass insertion can be calculated using an assumed propulsion mass structural factor of 10%, meaning that the satellite payload mass to GEO is 2,700 kg [76]. The final column of the table gives the required orbit insertion mass, m_{s0} , that the launch vehicle must provide to LEO in order to guarantee a 2,700 kg payload mass to GEO given the required propellant mass for the trajectory. For the coplanar and non-coplanar sets of test cases, the very-low-thrust transfer is continuously burning whereas the impulsive transfer is calculated using a Hohmann transfer or combined plane change. The I_{sp} values for the analytic optimal solutions are typical for a high efficiency ion thruster (10,000 s) or an on-board chemical propellant system (300 s). The PSO and NLP solvers all used the same performance levels of $n_0=0.01g$ and $I_{sp}=1181$ s.

The results show that forcing the spacecraft to thrust continuously is sub-optimal at the 0.01g thrust level. When finite burns are allowed, the ΔV drops to less than that of the very-low-thrust case, but is still surpassed by the impulsive transfer. More indicative of the fuel-efficiency, the mass fraction shows that using the 0.01g level of

Table 9. Low-thrust transfers comparison

Scenario/Solver	Burn Time (days)	Transfer Time (days)	ΔV (km/s)	m_f/m_0	m_p (kg)	m_{s0} (kg)
Continuous, $\Delta i = 0^\circ$						
PSO	0.6393	0.6393	6.4338	0.5739	2,226	5,226
NLP	0.6079	0.6079	6.0192	0.5948	2,043	5,043
Finite, $\Delta i = 0^\circ$						
PSO	0.5055	0.7730	4.7607	0.6631	1,524	4,524
NLP	0.4791	0.9039	4.4573	0.6807	1,407	4,407
Very-low-thrust ($n_0=0.001$, $I_{sp}=10,000$)	53.832	53.832	4.6510	0.9537	146	3,145
Impulsive-two-burn ($I_{sp}=300$)	-	0.2198	3.8926	0.2664	8,260	11,260
Continuous, $\Delta i = 28.5^\circ$						
PSO	0.6638	0.6638	6.7673	0.5577	2,379	5,379
NLP	0.6503	0.6503	6.2468	0.5833	2,143	5,143
Continuous, $\Delta i = 28.5^\circ$, J2						
PSO	0.6537	0.6537	6.6295	0.5643	2,315	5,315
NLP	0.6239	0.6239	6.2290	0.5842	2,135	5,135
Finite, $\Delta i = 28.5^\circ$, J2						
PSO	0.7392	0.7849	7.8746	0.5068	2,918	5,918
NLP	0.5133	0.8015	4.8517	0.6579	1,559	4,559
Very-low-thrust ($n_0=0.001$, $I_{sp}=10,000$)	68.8755	68.8755	5.9508	0.9411	187	3,187
Impulsive-combined ($I_{sp}=300$)	-	0.2198	4.2559	0.2354	9,739	12,739

thrust acceleration keeps the transfers below one day and requires significantly less propellant than the impulsive transfers. While the best mass fractions are gained by using a very-low-thrust spiral transfer, the transfers takes upwards of 50 days to complete. If a reasonable level of fuel-efficiency as well as transfer speed is desired, then thrust acceleration levels around 0.01g are an acceptable option.

For a typical payload that is sent to GEO, the launch vehicle does not drop the payload off at LEO. Instead, a geosynchronous transfer orbit (GTO) is used as the initial parking orbit. A GTO is a highly elliptical orbit that typically has a low perigee altitude of a few hundred kilometers and an apogee at GEO altitude. For impulsive GTO to GEO transfers, the ΔV is about 1.5 km/s for the planar transfer and 1.8 km/s for the 28.5 deg non-coplanar transfer. Compared to the 3.89 km/s required for an impulsive LEO to GEO transfer, the GTO to GEO insertion system is required to provide 2.5 km/s lower in terms of ΔV to reach the final orbit. Naturally, the ΔV disparity is incurred by the launch vehicle, but it has been shown to be more efficient in terms of fuel used by the launch vehicle to use a GTO to GEO boost stage instead of a boost stage from LEO [76]. Using the GTO to GEO ΔV s in conjunction with a chemical propulsion system having an I_{sp} of 300 s, the propellant masses required for the coplanar and non-coplanar case are 1,994 kg and 2,530 kg, respectively. Comparing these numbers to the NLP solutions for the LEO to GEO finite low-thrust planar and non-coplanar transfers, the required propellant masses are lower at 1,407 kg and 1,559 kg, respectively. This implies that the same payload, assuming the low-thrust and impulsive-thrust propulsion systems are the same mass, can reach GEO for less propellant starting at a less expensive parking orbit for the launch vehicle. The conclusion that low-thrust when compared to impulsive thrust is more fuel efficient is not new; however, in terms of practicality, the finite low-thrust LEO to GEO transfers are feasible options. This is based on the fact that

approximately 1,500 kg of propellant used by a boost stage to insert a 2,700 kg payload to GEO is feasible with current technology. Also, compared to the m_{s0} column in Table 9, launch vehicles are able to easily provide upwards of 9,000 kg to LEO [77,78]. In fact, in terms of possible masses to LEO given current launch vehicle technology, all of the required initial masses are feasible. However, feasibility does not dictate practicality. For the most practical option in terms of fuel usage and time of flight, a GTO-GEO finite burn transfer using low-thrust should be conducted. This combines the benefit of a higher energy parking orbit with the fuel efficiency of low-thrust propulsion. This particular transfer is not considered for optimization in the current investigation, but is foreseeably straightforward to conduct given the proposed methodology.

Up to this point, very-low-thrust options have not been discussed. If the user is willing to wait fifty or more days for final orbit insertion, then haste is not a concern. However, given a starting point in LEO, the very-low-thrust satellite must gradually conduct the orbit raising and inclination change over hundreds of revolutions. Other concerns such as extended time in the Van Allen radiation belts as well as power capabilities for constant thrusting become much more relevant. It may be worthwhile to investigate hardening the satellite to radiation as well as having a robust on-board power system to employ very-low-thrust options for the orbit transfer phase of the mission. Also, using a very-low-thrust acceleration level starting from GTO may provide a more acceptable time of flight while still gaining the very high fuel efficiency. Overall, the conclusion is that, for GEO based satellites, using a GTO parking orbit is the most practical to relieve the propulsive burden on the satellites. Also, finite burn transfers at low-thrust offer a middle ground between fuel efficiency and time of flight.

3.3 Chapter Summary

This chapter provided the methodology and results for multiple near-Earth LEO to GEO low-thrust transfers. Due to the fact that PSO operates best when optimizing a small amount of design variables, a polynomial-based approach was used to approximate the optimal continuous time history of the thrust pointing angles for each trajectory. All of the PSOIGs given to the NLP solver were within the convergence window with most of the initial guesses being relatively close to the locally optimal solution. The LEO to GEO transfers, while not the most practical, did elucidate the benefits of using low-thrust as well as finite burning. A more practical transfer and an avenue for future work would be to apply the methodology in this chapter to low-thrust GTO to GEO transfers.

For the first several cases, learning how to weigh the constraints vs. the time of flight or burn time and coast time proved to be the factor that required the most user experimentation. In other instances, increasing the number of particles or iterations was done if the desired level of convergence was not met. For the continuous, planar trajectory, supplying *fmincon* with an acceptable non-PSOIG was relatively easy and only required a few changes to the step size. Manual creation of the initial guess for the finite-burn and non-coplanar trajectories required more finesse. While PSO demonstrated the ability to generate a “good-enough” initial guess for all of the scenarios, once the number of design parameters reached fifty or more, it was difficult to generate an initial guess that was in close proximity to the final converged solution. More effort could have been made to allow for longer run times with larger swarms; however, keeping the computation time within one hour for the PSOIG was desired. As a final note, the transfers generated in this chapter are not claimed to be globally optimal, but local to the PSOIG supplied to the NLP solver.

The next chapter seeks to test the efficacy of PSO as a means for initial guess

generation by subjecting it to a highly nonlinear dynamical environment in which chaos is present, the CR3BP.

4. Impulsive High-Altitude Trajectory Design

When operating at altitudes higher than GEO, the effects of the Moon's gravity become much more significant. Modeling the Moon's gravity as a small perturbation much like the inclusion of J_2 in the previous chapter is a valid approach for altitudes up to GEO, but greater accuracy and insight may be gained by transitioning into a three-body model for super-GEO altitudes. The circular restricted three-body model is used as the dynamical environment for the methods and trajectory design showcased in this chapter due to a focus on very-high-altitude operations. Apart from the accuracy gained by using this model at super-GEO altitudes, the range of possible behavior is also expanded. The trajectories that result from numerically integrating the CR3BP equations of motion are not Keplerian as they exhibit a high level of nonlinearity and may exist in regions of the phase space where chaos is present. Designing in this complex dynamical environment is difficult and as such, an initial guess for a desired transfer is not always readily available due to the lack of a closed-form analytical solution for the CR3BP. PSO plays the role of initial guess generation in the design approach employed in this chapter.

The methodology presented in this chapter first utilizes a differential corrections method for numerically solving TPBVPs in the CR3BP. Using this approach, periodic solutions are calculated using targeting and PSO. Finally, dynamical systems theory is applied with a focus on invariant manifolds associated with periodic orbits near the Earth-Moon L1 Lagrange point. The results section presents the design of a three-impulse trajectory from low-Earth-altitude (300 km) to a libration point orbit (LPO) about the Earth-Moon L1 equilibrium point. A segment of the trajectory takes place on an approximation for a stable manifold trajectory in order to efficiently approach the target LPO. Lastly, it is important to emphasize that the present chapter only explores motion in the planar (x - y) CR3BP.

4.1 Methodology

This section covers targeting periodic orbits, identifies the particular invariant manifold emanating from the design LPO, and details the methods used in the current investigation for PSOIG generation and NLP improvement in the CR3BP.

All states, $\mathbf{X}$, in this chapter are propagated via the EOMs given by the system of equations (33). The numerical integration is conducted using MATLAB[®]'s built-in ordinary differential equations (ODE) solver, *ode45*. The solver is a one-step, fourth-order, fifth-order corrections, Runge-Kutta integrator. *ode45* benefits from a variable step size whereas the Runge-Kutta integration in the previous chapter uses a fixed step. A variable step size is advantageous in regions of the phase space that are sensitive or chaotic because the time step can be decreased to maintain accuracy. A variable step size also provides faster computation for less chaotic regions where increasing the step does not introduce significant error. The relative tolerance 'RelTol' and absolute tolerance 'AbsTol' in the odeset options are both set to 10^{-13} for the numerical integration conducted in this chapter. This level of fidelity corresponds to meter accuracy in position and sub-mm/s accuracy in velocity, a necessity when operating in a chaotic environment. For additional documentation on the ODE solvers in MATLAB[®], reference Shampine and Reichelt [79].

For each simulation in the present chapter, the elapsed computation time ranges from a few minutes for each PSO solved TPBVP to no more than an hour for NLP improvement. The times correspond to elapsed time in MATLAB[®] (Version: 8.1.0.605 (R2013a); benchmark: 0.2930, 0.3178, 0.2058, 0.3201, 0.6751, 0.5911) [68]. Also, the computer used runs a 64-bit Windows 7 operating system with 4GB of RAM and an Intel(R) Celeron(R) CPU E3400 @2.60 GHz processor.

4.1.1 Targeting Periodic Orbits

In the CR3BP an infinite number of periodic orbits can be computed. Periodic, in this context, means that after a finite amount of time the spacecraft, given an initial condition, returns to the same initial conditions in both position and velocity within a satisfactory tolerance. Periodic orbits of particular interest are those near the Lagrange points. To target a periodic orbit about one of the collinear libration points, an interesting symmetry can be exploited. For every trajectory there is a mirror image trajectory across the x - z plane that runs in negative time [80]. Using this knowledge, starting on the x - z plane with velocity solely in the y direction and targeting the next perpendicular crossing of the x -axis results in half of a periodic orbit. The remaining half can be found by reflecting the current half across the x - z plane.

For an initial guess of the LPO, a linear approximation is found in Szebehely [24]. Using this approach for motion near L1 in the planar CR3BP, the initial state, $\mathbf{X}_{LPO}(t_0)$, is given below in nondimensional and dimensional units in the barycentric rotating frame.

$$\mathbf{X}_{LPO}(t_0) = \begin{bmatrix} x_0 \\ y_0 \\ \dot{x}_0 \\ \dot{y}_0 \end{bmatrix} = \begin{bmatrix} 0.846915121142417 \\ 0 \\ 0 \\ -0.083722733189462 \end{bmatrix} = \begin{bmatrix} 325,554.172567146 \text{ km} \\ 0 \text{ km} \\ 0 \text{ km/s} \\ -0.081716841593909 \text{ km/s} \end{bmatrix} \quad (118)$$

The initial conditions correspond to an initial condition between L1 and the Moon on the x -axis with a perpendicular crossing in the negative y direction. Targeting the next perpendicular crossing with the x -axis requires that the initial velocity in the y direction be variable in addition to time. The error that is being minimized consists

of the final y position and final velocity in the x direction. Thus, the constraint or error vector $\mathbf{F}(\mathbf{X})$ is written below.

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} y(t_f) \\ \dot{x}(t_f) \end{bmatrix} = \mathbf{0} \quad (119)$$

Constructing the $D\mathbf{F}$ matrix results in the square matrix ($n = m = 2$) below.

$$D\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \frac{\partial y}{\partial y_0} & \dot{y} \\ \frac{\partial \dot{x}}{\partial y_0} & \ddot{x} \end{bmatrix} \quad (120)$$

Since the $D\mathbf{F}$ matrix is square, equation (52) can be used as the update equation. It is worth noting that the elements in the first column can be extracted from the STM. For an initial guess for time, the time associated with first x -axis crossing ($t = 5.515117$ days) is used. An event function within *ode45* is employed to ensure that the numerical integration stops at the x -axis. Given the event function, time updates are ignored in this differential corrections process. The convergence tolerance for the corrections scheme, ϵ , is set to 10^{-13} . Figure 49 shows the convergence behavior for targeting half of the periodic orbit.

The numbers at the end of the individual arcs correspond to the current iteration where the first iteration is 0. As shown in Figure 49, the first and second iterations are not perfect perpendicular crossings; however, starting at the fourth iteration, differentiating between the iterations becomes difficult as only minor corrections are subsequently made. To generate the remaining half of the periodic orbit, a reflection across the x - z plane is made resulting in the LPO shown in Figure 50.

The period of the targeted LPO is 5.515117 days. The L1 Lyapunov orbit can also be calculated accurately via PSO. Using the initial state in equation (118), PSO is set up to optimize the initial velocity and time. The cost function evaluated at

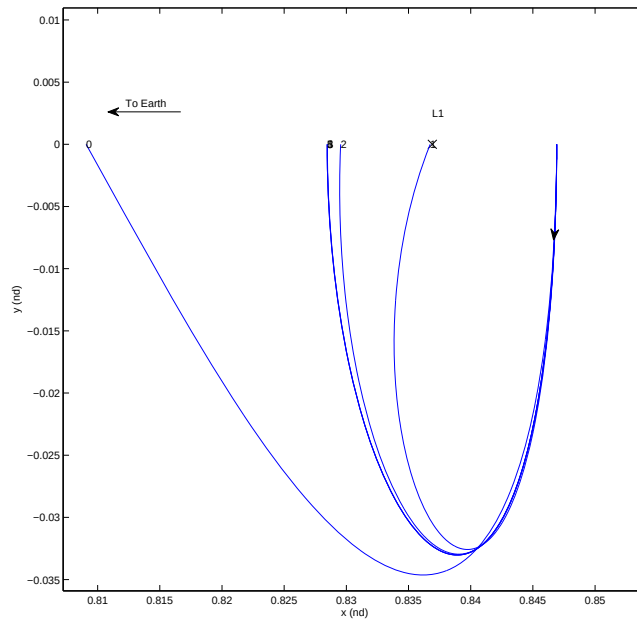


Figure 49. Targeting a planar periodic orbit near L1 in nondimensional rotating barycentric frame

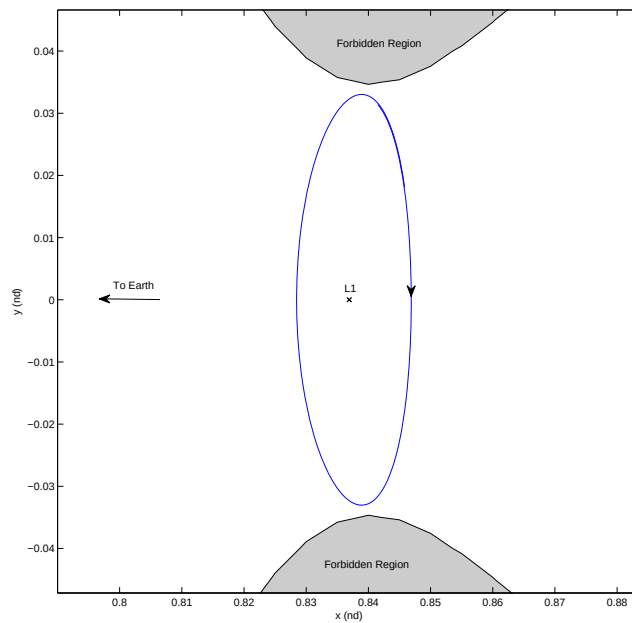


Figure 50. One revolution of the targeted planar periodic orbit near L1 in nondimensional rotating barycentric frame

each iteration is given in equation (121).

$$J(P) = x(t_f) - x(t_0) + V_x(t_f) \quad (121)$$

where

$$\mathbf{P} = \begin{bmatrix} V_x(t_o) \\ t_f \end{bmatrix} \quad (122)$$

and

$$\begin{bmatrix} -0.01 \\ 1 \end{bmatrix} \leq \mathbf{P} \leq \begin{bmatrix} -1 \\ 3 \end{bmatrix} \quad (123)$$

The cost function J is minimized for a periodic orbit after one revolution with a perpendicular crossing at the x -axis. The bounds are set such that the PSO does not converge on a solution with minimal displacement from the initial condition and is within a reasonable range of C values for the LPO. Also, constraint weighting is not used due to an already well-conditioned and low-dimensioned search space. Figure 51 shows the convergence behavior of 30 particles in 100 iterations. The presented LPO is used as the final target orbit for the three impulse trajectory designed in this chapter.

In the CR3BP, periodic orbits exist in families. If a different periodic orbit is desired, a continuation scheme can be used to generate more members from the same orbit family. For this particular planar family, stepping in the x direction from the previous initial condition and using the initial velocity from the previous LPO as an initial guess allows the targeting scheme to converge on a new family member. The step in the x direction cannot be too large or else the targeter may have difficulty converging. Conversely, a very small step size can be computationally expensive if a range of family members is desired. This continuation scheme is used to generate twenty neighboring periodic orbits within the same L1 Lyapunov family. All twenty-

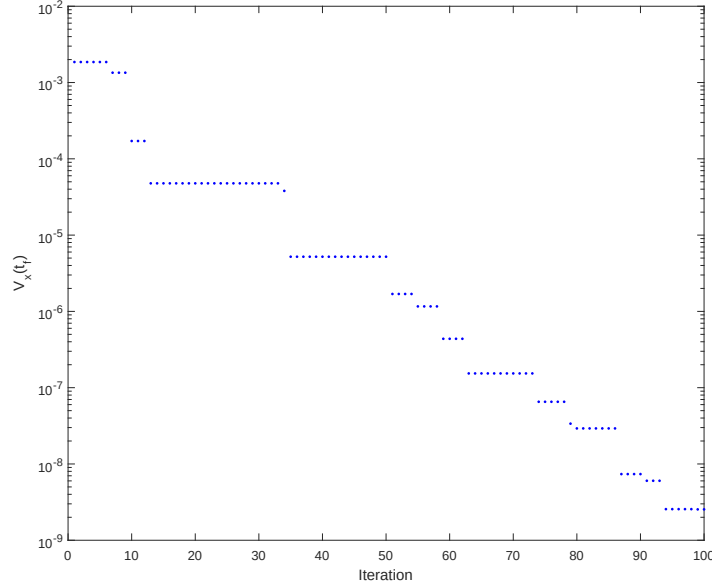


Figure 51. PSO convergence on LPO for 30 particles and 100 iterations

one L1 Lyapunov family members are shown in Figure 52. The next section exploits dynamical systems theory in order to generate an approximation for the invariant stable manifold tube emanating from the first LPO targeted in this section.

4.1.2 LPO Invariant Manifold

The trajectory designed and presented in Section 4.2 employs a coast arc on an approximation for an invariant stable manifold trajectory approaching an unstable LPO about L1. A coast segment on the invariant manifold is motivated by an expected efficiency when following the natural dynamical “flow” in a model. In other words, invariant manifold trajectories represent “free” transfers (zero ΔV) to and from periodic orbits in infinite time due to their asymptotic behavior. Using approximations of the invariant manifold slightly increases the required ΔV , but allows the LPO approach phase to be conducted in a reasonable amount of time. The relevant dynamical systems theory required for generating the manifold used is provided in Section 2.2.6. The manifold tube of interest for the trajectory designed in this chap-

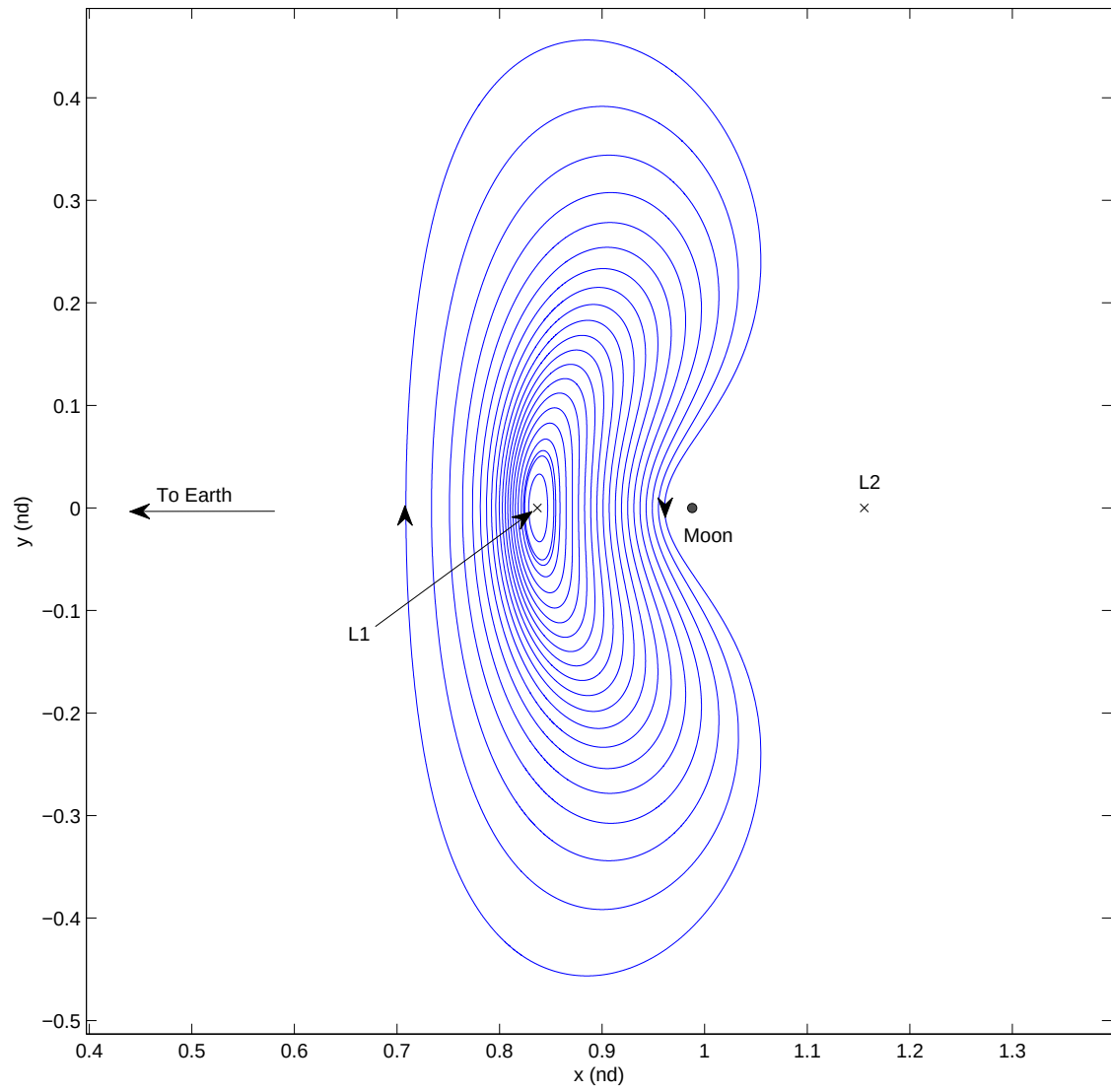


Figure 52. Family members of the L1 Lyapunov family in nondimensional rotating barycentric frame

ter is the stable invariant manifold that “departs” (in negative time) the design LPO toward the Earth. An approximation for the stable manifold tube, propagated for 43.4 days in negative time, is shown in Figure 53 with a zoomed in plot in Figure 54. The manifold tube associated ZVCs are also shown in the figures.

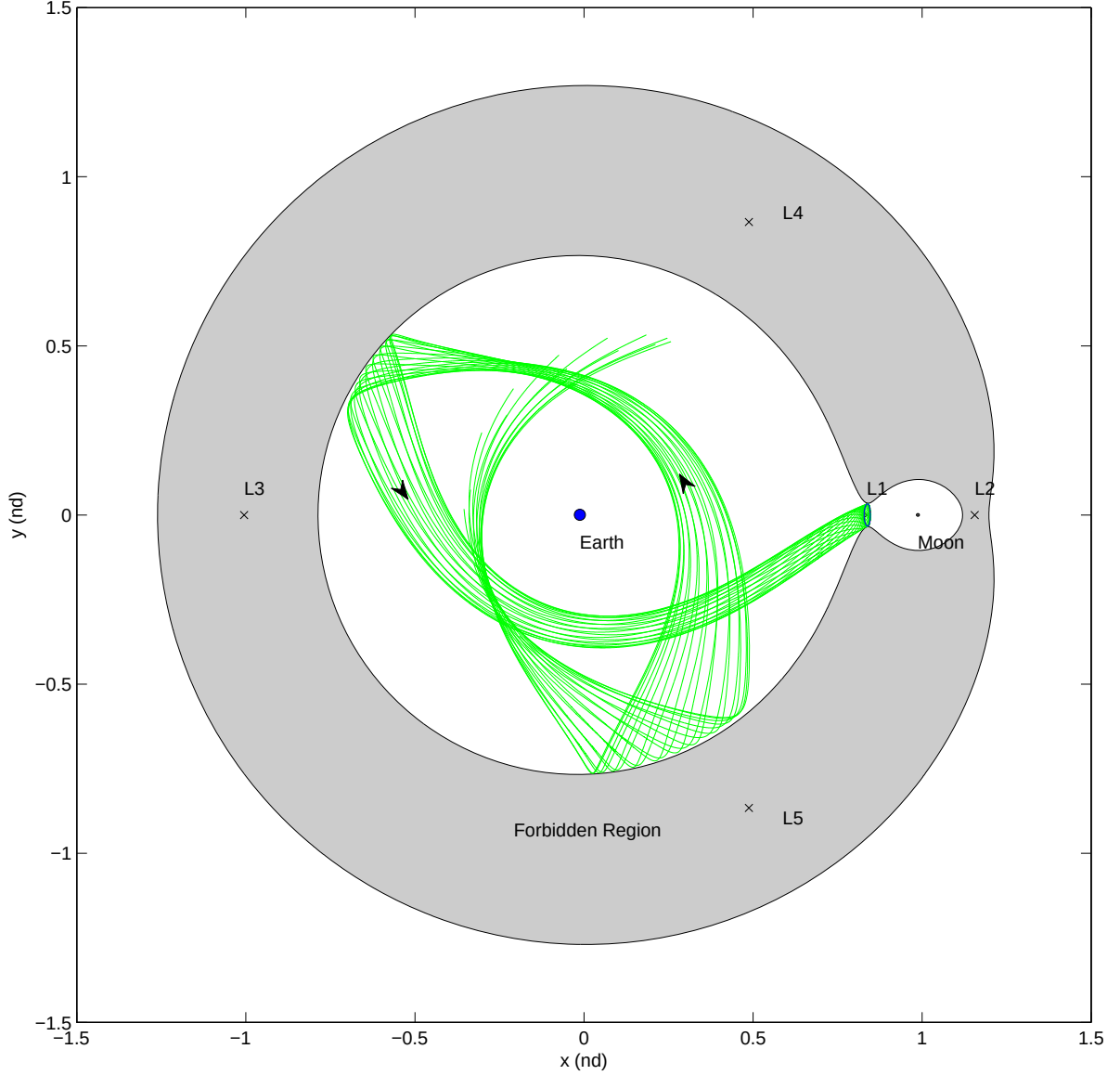


Figure 53. Stable manifold tube propagated from LPO in nondimensional rotating barycentric frame, propagated for 43.4 days

At the value of Jacobi constant associated with the stable manifold tube ($C = 3.18339545917064$), the “L1 gateway” is open, but the “L2 gateway” is closed.

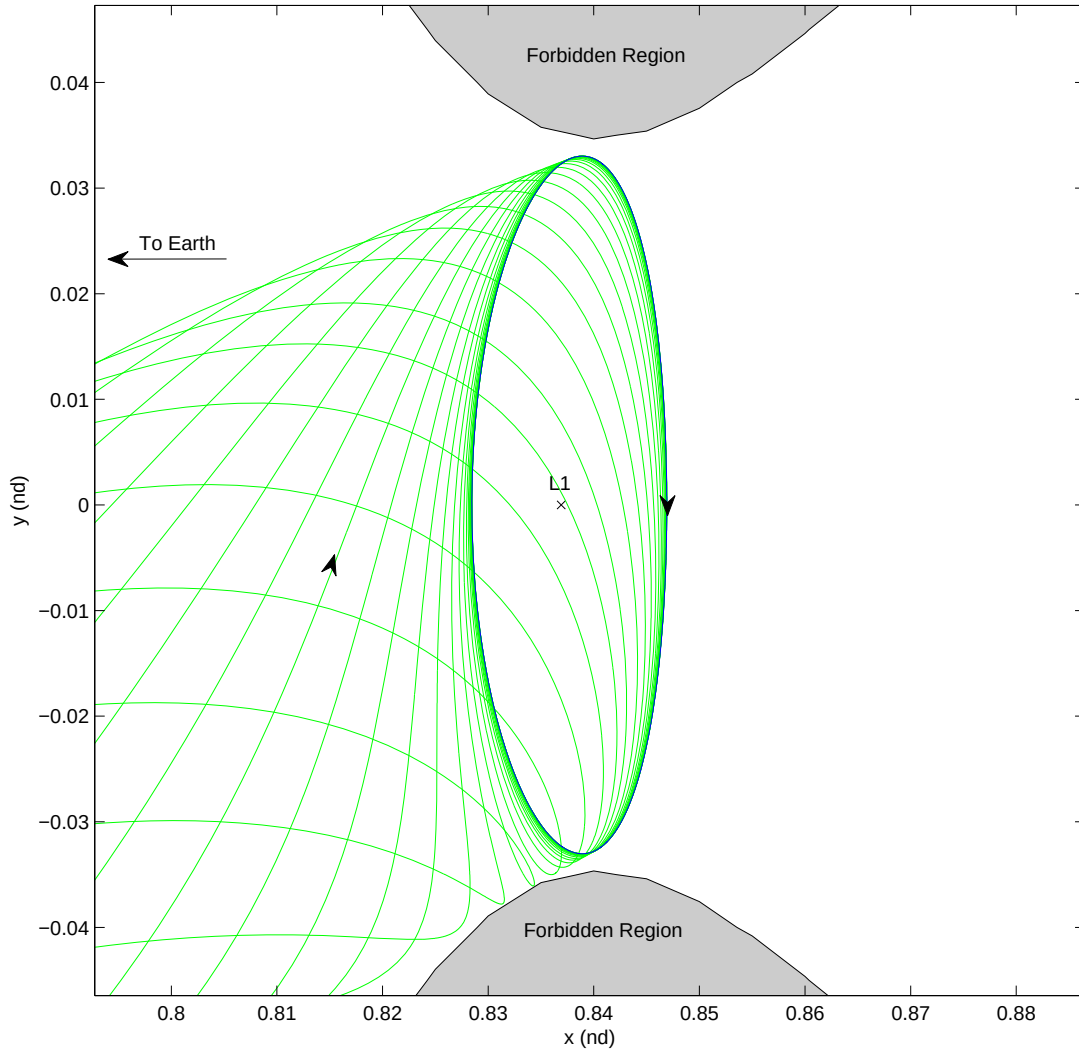


Figure 54. Zoomed in view of stable manifold tube propagated from LPO in non-dimensional rotating barycentric frame

Thus, motion around the Earth and the Moon is possible; however, the transitioning from the interior region to the exterior region beyond L2 and L3 is impossible without propulsion. For the trajectory designed in this chapter, a free-variable to be optimized is the location in which the spacecraft begins coasting on the stable invariant manifold. Specifically, the insertion point on a specified stable manifold trajectory is a design parameter.

4.1.3 PSO Initial Guess and NLP Improvement

To generate the PSOIG in this chapter, the free-variables are straightforward and a creative parameterization scheme is not required. Since impulsive burns are assumed, the PSO is set up to optimize boundary conditions for the TPBVP, which for a portion of the transfer, manifests as ΔV components. For a three impulse transfer, the number of design parameters is small, thus, PSO should not have difficulty with the dimension of the search space. Unfortunately, a difficulty due to the complexity of the dynamical environment is encountered. The difficulty arises in attempting to find appropriate constraint weighting factors. For PSO, optimality conditions are not checked, the final solution is the global best candidate solution once the iteration count reaches k_{max} .

The PSOIGs generated in the next section are passed to *fmincon* for improvement in terms of ΔV and continuity. A benefit of using impulsive burns is that control variables do not have to be inserted during the integration. As such, the numerical integration between impulses is numerically integrated using *ode45*. Without the need for control insertion, the problem size is much smaller compared to the transfers in the previous chapter, therefore, providing gradient information to the NLP solver is recommended. The gradient information takes on the form of equations (79) and (80), noting that *fmincon* requires the transposes of each of the gradient matrices

when they are supplied. In employing *fmincon* for NLP improvement, the function tolerance ‘FunTol’ and constraint tolerance ‘ConTol’ are left at the default values of 10^{-6} . The function tolerance and constraint tolerance directly correspond to the tolerances that must be met for optimality. The next section details the design and presents the results for the LEO to LPO three impulse transfer.

4.2 Results

The trajectory designed in this section begins at LEO altitude (300 km) and ends on the LPO generated in the previous section. The transfer consists of three impulsive burns where the objective is to minimize the total magnitude of the impulsive burns. The first burn is used to leave LEO and approach the stable manifold trajectory insertion point. After arriving at the manifold insertion point, a second burn is used to get “on” the manifold trajectory in terms of velocity, as position continuity has already been met within a certain tolerance. After coasting on the stable manifold trajectory, a final burn is used to enter the LPO.

To define the boundary conditions, first the particular LEO is defined by the following initial state in nondimensional units.

$$\mathbf{X}_{LEO}(t_0) = \begin{bmatrix} 0.00522229846503999 \\ 0 \\ 0 \\ 7.53 \end{bmatrix} \quad (124)$$

Due to the existence of an additional gravitational body, this LEO is not Keplerian in the sense that it is not perfectly circular or periodic. Instead, this approximation of a 300 km altitude LEO returns to almost the initial point after one period. The approximate “period” of this orbit is 1.5216 hours. In order to allow for a variable

LEO departure point, the first free parameter is the time that the LEO orbit is propagated, t_{LEO} , before the first impulse occurs. The initial LEO as well as its associated ZVCs are shown in Figure 55.

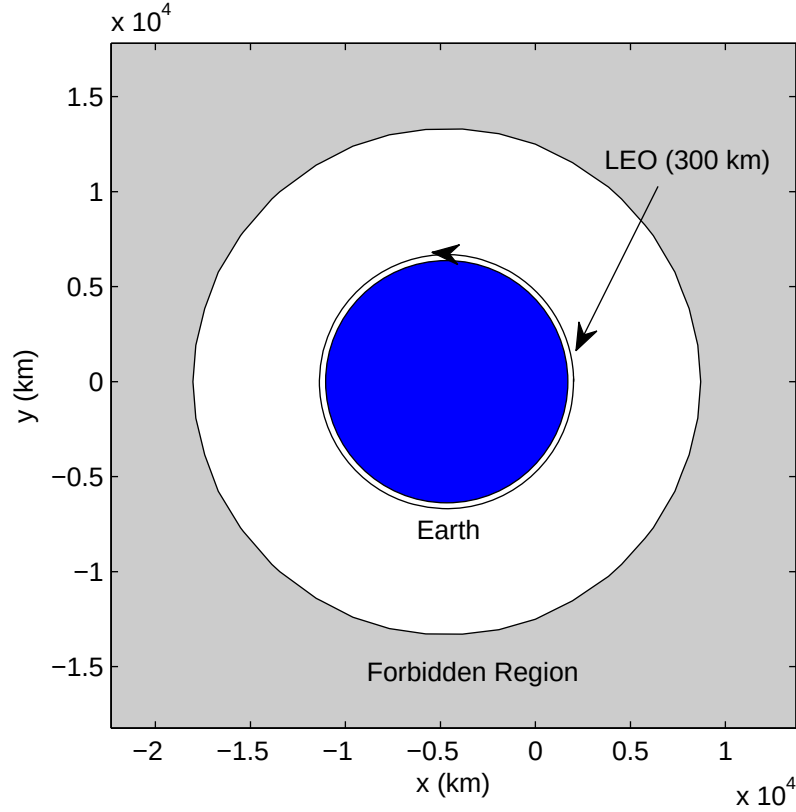


Figure 55. Initial LEO altitude orbit and associated ZVCs in rotating barycentric frame, propagated for 1.52 hours

The ZVCs in the figure show that, at the initial energy level, the motion of the spacecraft is bounded to stay within the vicinity of the Earth. A similar region is accessible about the Moon; however, motion in that accessible region requires the spacecraft to begin within that accessible region. Due to the boundedness, it is clear why an impulsive burn(s) is required to reach the LPO.

The next two free parameters are the velocity in the x direction, V_{x1} , and velocity in the y direction, V_{y1} , after the first impulse occurs. The fourth parameter is the time of flight that the state after the first impulse is propagated, t_1 . The optimizer seeks to

choose V_{x1} , V_{y1} , and t_1 such that the spacecraft stops exactly at the manifold insertion point after t_1 has elapsed while still minimizing the initial impulse made. The final design parameter is the manifold insertion point. This is encompassed by a single time parameter t_M , that defines the amount of time the stable manifold trajectory is propagated in reverse time from the LPO. The initial state $X_s(t_0)$ for the chosen stable manifold trajectory in this scenario is given below in nondimensional units.

$$\mathbf{X}_s(t_0) = \begin{bmatrix} 0.841518629433691 \\ 0.031415828337320 \\ 0.019345235864132 \\ -0.023166570216752 \end{bmatrix} \quad (125)$$

The stable manifold trajectory with the initial conditions in equation (125) is shown in Figure 56 with a zoomed-in view in Figure 57.

Per the method used to generate the stable manifold trajectory given in Section 2.2.6, an important note is that a “true” asymptotic stable manifold trajectory is not used. The stable manifold trajectory used and depicted in Figures 56 and 57 is an approximation of the true asymptotic stable manifold trajectory due to the displacement, d , initially incurred. From the figures, the stable manifold trajectory, in many instances, approaches its associated ZVCs; however, they are not crossed during the integration time. The ZVCs show that the “L1 gateway” is open to the extent required to allow the LPO to exist around the L1 point. This assumes only a minor burn is required near the LPO. In reality, the ZVCs would adjust according to the small change in C incurred after the final burn. The next section discusses and presents the PSOIG for the three impulse trajectory.

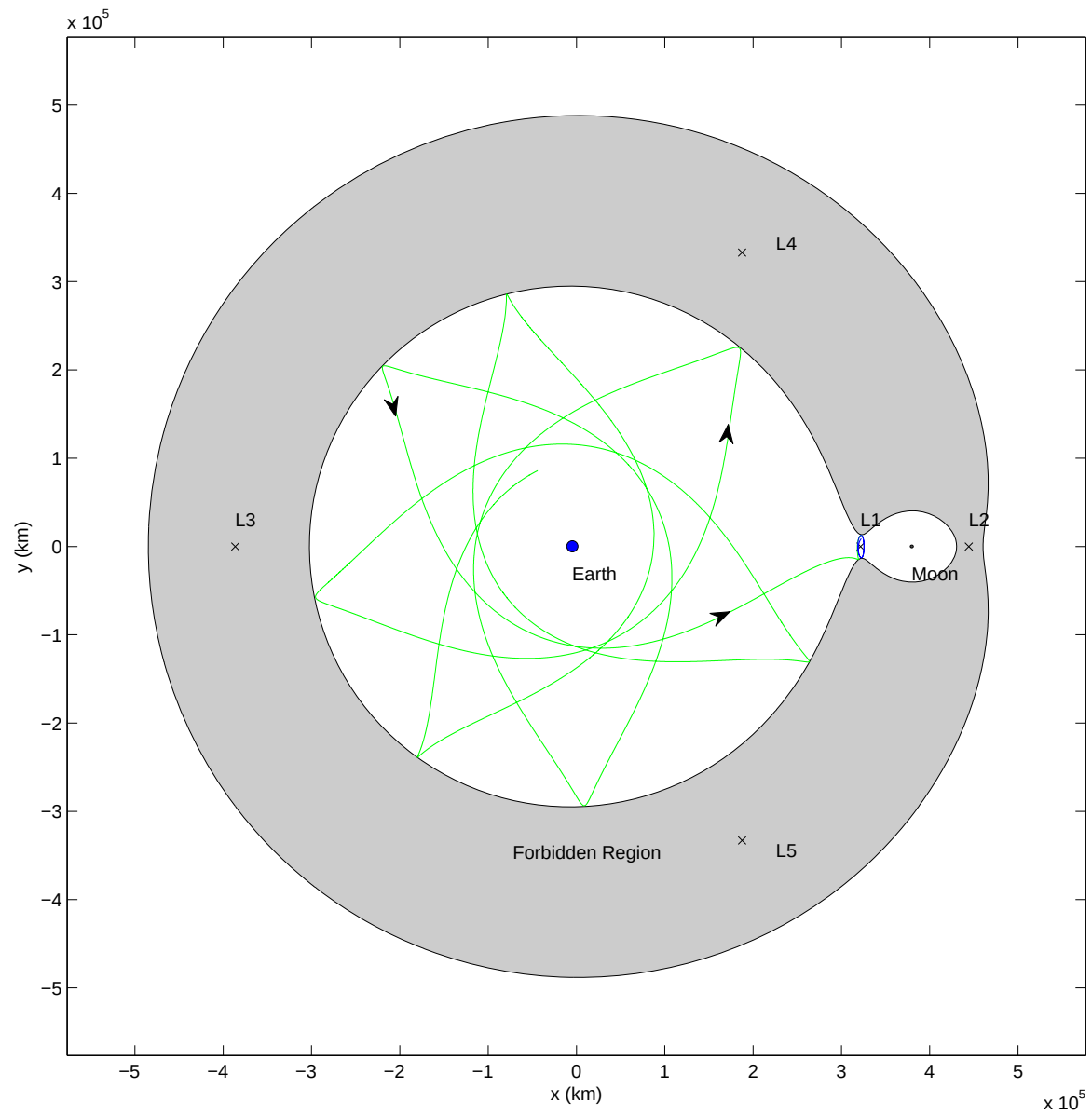


Figure 56. Chosen stable manifold trajectory and associated ZVCs in rotating barycentric frame, propagated for 86.8 days

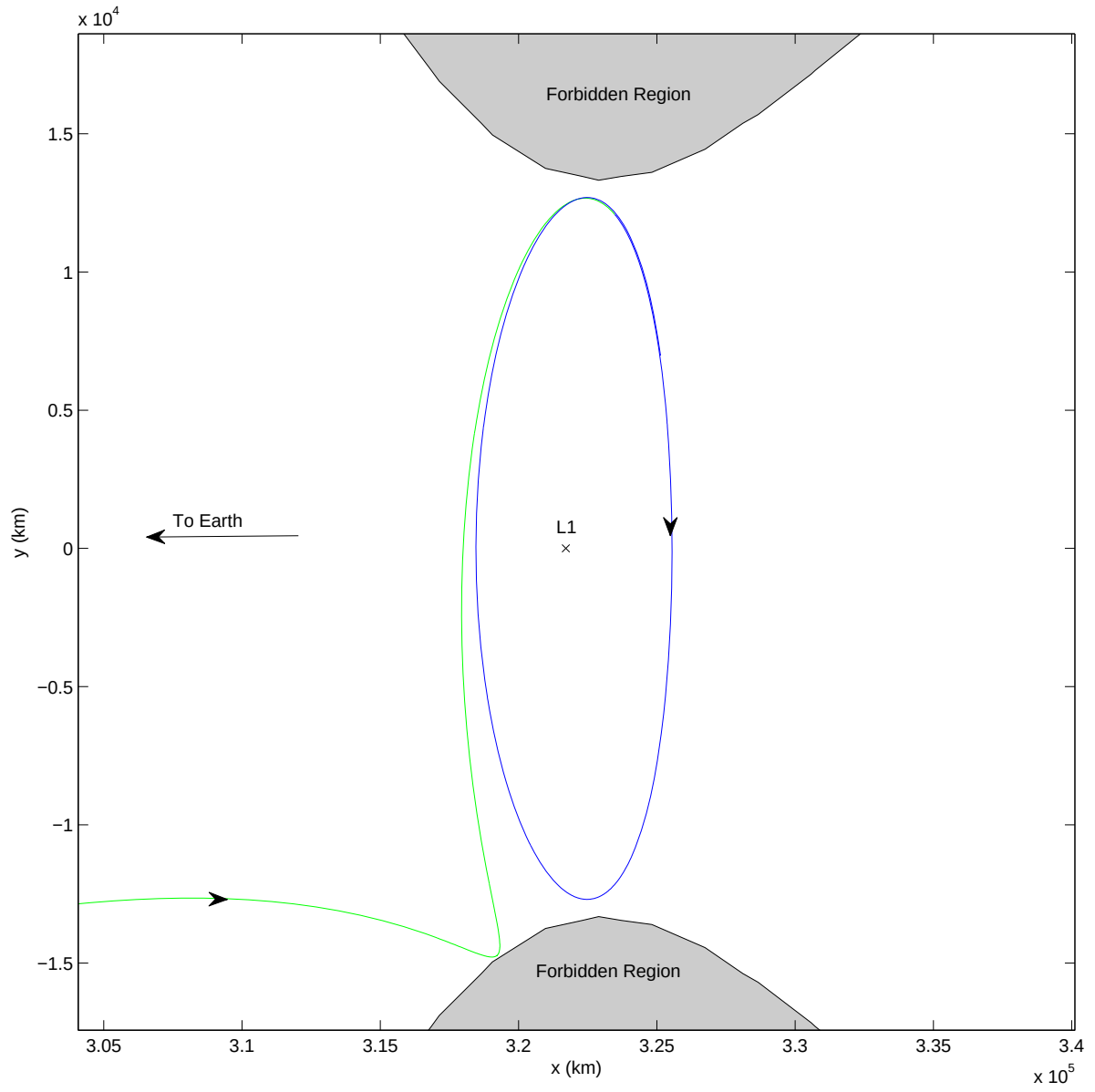


Figure 57. Zoomed in view of chosen stable manifold trajectory and associated ZVCs in rotating barycentric frame

4.2.1 PSO Initial Guess Generation

Initially, the PSO was set up to solve for all five of the parameters previously defined (t_{LEO} , V_{x1} , V_{y1} , t_1 , and t_M). However, this did not provide the desired results as the optimizer tended to converge on the first particle that came close to satisfying the boundary conditions at the randomly guessed manifold insertion point. Changing the weighting of the constraints in the augmented cost function, after several trials, did not result in improved convergence behavior. As a result, the following PSO setup is conducted for a range of manifold insertion points starting at 1 nondimensional time all the way to 10 nondimensional time with a 0.5 nondimensional time step. This corresponds to nineteen individual TPBVPs that PSO must solve.

As previously stated, the variable manifold insertion point is removed from the free-variable array such that $\mathbf{P}$ is now defined as

$$\mathbf{P} = \begin{bmatrix} t_{LEO} \\ V_{x1} \\ V_{y1} \\ t_1 \end{bmatrix} \quad (126)$$

where the bounds in nondimensional units on $\mathbf{P}$ are given by

$$\begin{bmatrix} 0.000001 \\ -13 \\ -13 \\ 0.01 \end{bmatrix} \leq \mathbf{P} \leq \begin{bmatrix} 0.0146 \\ 13 \\ 13 \\ 5 \end{bmatrix} \quad (127)$$

The velocity component bounds are dictated by first calculating the velocity required to be at the energy level associated with the stable manifold at the barycenter ($x = 0$, $y = 0$). The required velocity is 10.1 nondimensional units. Given the as-

sumption that an efficient trajectory conducts most of the required change in energy in the first impulse at LEO and only requires a smaller burn to get onto the manifold trajectory, an efficient trajectory does not require any one velocity component to be much more than 10.1 nondimensionally. Therefore, the velocity bounds are set to be just above the 10.1 nondimensional velocity requirement. The time of flight bound of 5 nondimensional time units (21.7 days) is set so that the segment from LEO to the manifold insertion point does not take an unreasonably long amount of time. Lastly, the t_{LEO} bounds correspond to one revolution of the LEO.

The constraints are set up to enforce position continuity at the manifold insertion point.

$$\phi(x^{i,k}) = \begin{bmatrix} x_M(t_M) - x_1(t_1) \\ y_M(t_M) - y_1(t_1) \end{bmatrix} = \mathbf{0} \quad (128)$$

In addition to the satisfying the constraints, the cost function is set to minimize the total ΔV . Since the final burn is a function of the manifold trajectory chosen already, the third burn is not included in the cost function. The modified cost function is defined as

$$\tilde{J} = \|\phi_1\| + \|\phi_2\| + 0.1(\Delta V_1 + \Delta V_2) \quad (129)$$

where

$$\begin{aligned} \Delta V_1 &= \sqrt{[\dot{x}_{LEO}(t_{LEO}) - \dot{x}_1(t_0)]^2 + [\dot{y}_{LEO}(t_{LEO}) - \dot{y}_1(t_0)]^2} \\ \Delta V_2 &= \sqrt{[\dot{x}_M(t_M) - \dot{x}_1(t_1)]^2 + [\dot{y}_M(t_M) - \dot{y}_1(t_1)]^2} \end{aligned} \quad (130)$$

The ΔV s are scaled down by a factor of 0.1 to prioritize satisfying the position continuity constraints in equation (128) prior to optimizing the total change in velocity. The PSO results for each solved TPBVP corresponding to different manifold insertion points using 500 particles and 500 iterations are shown in Figure 58.

In Figure 58, the highlighted green trajectory is the approximation of the stable manifold trajectory. The different colored segments each represent the PSOIG for

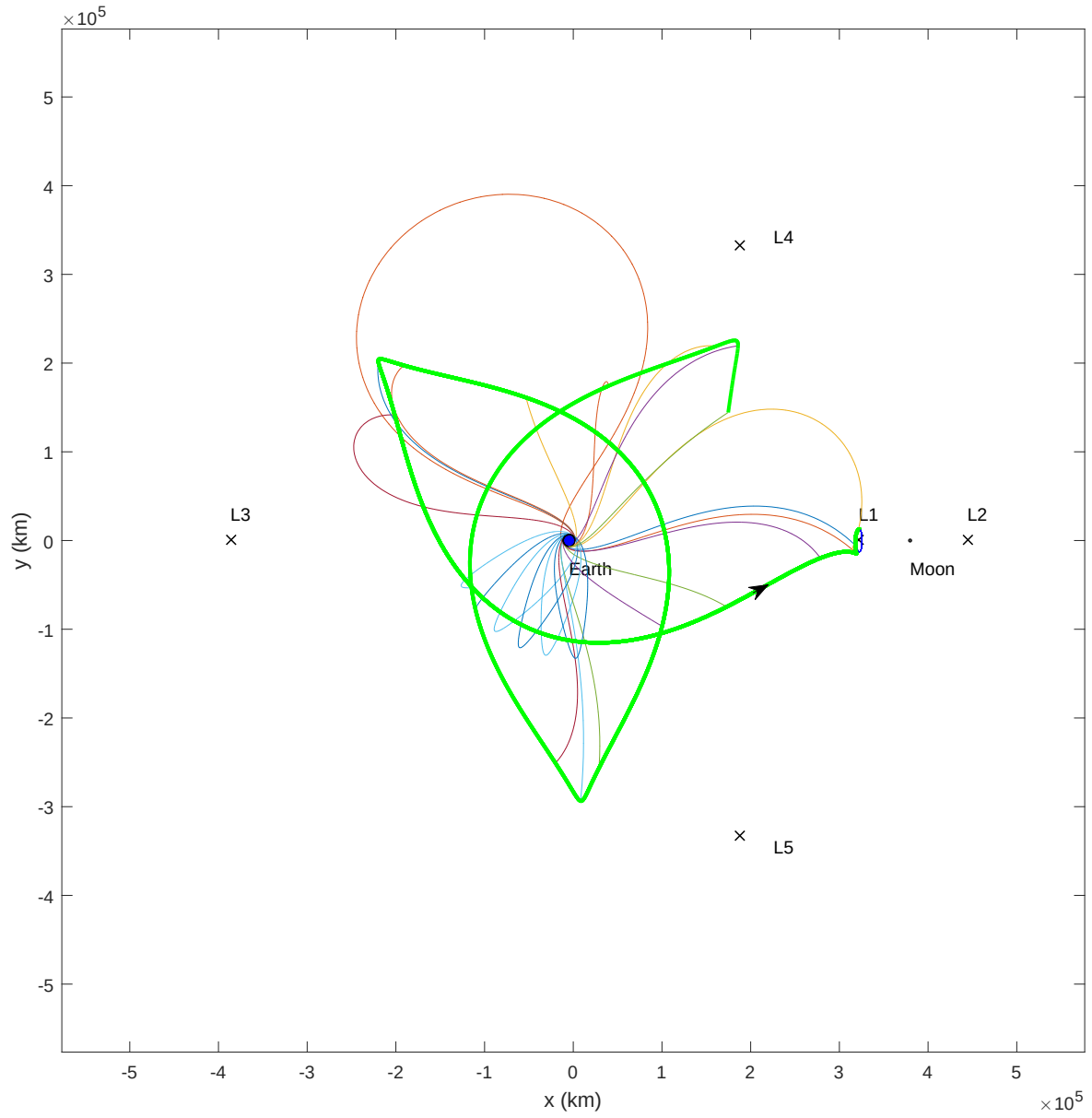


Figure 58. Nineteen PSO generated trajectories for various stable manifold insertion points in barycentric rotating frame (ZVCs not shown)

a particular insertion point onto the manifold trajectory while still allowing for a variable LEO departure point. Table 10 shows the ΔV s for each of the PSOIGs.

Table 10. PSO trajectory specifications

t_M (nd)	t_{LEO} (hours)	ΔV_1 (km/s)	ΔV_2 (km/s)	t_1 (days)	$\Delta V_1 + \Delta V_2$ (km/s)
1.0	1.0298	3.6258	0.9860	2.4668	4.6118
1.5	0.0198	3.0808	0.9682	13.0274	4.0491
2.0	1.1203	3.3818	0.7866	5.3286	4.1683
2.5	0.8401	3.1614	0.6427	2.4446	3.8041
3.0	0.9104	3.9425	1.0043	1.0494	4.9468
3.5	0.4028	2.8713	1.4637	5.6253	4.3350
4.0	0.1272	3.2976	0.7257	4.0586	4.0233
4.5	0.0104	3.0608	0.7494	2.5571	3.8102
5.0	0.0104	3.0665	0.7304	3.0726	3.7969
5.5	0.0104	4.8887	1.3379	0.7348	6.2266
6.0	0.7918	3.8150	1.6669	0.9466	5.4819
6.5	0.6717	3.8242	2.4501	1.0699	6.2744
7.0	0.6972	4.2843	3.6618	0.7881	7.9462
7.5	0.6974	4.7339	0.7181	1.9452	5.4519
8.0	1.5439	3.2870	1.5622	5.3894	4.8492
8.5	1.1585	2.9466	1.0466	2.0285	3.9932
9.0	1.1073	3.0335	0.8384	2.5217	3.8718
9.5	1.5216	8.0775	0.9003	2.0629	8.9778
10.0	1.1059	3.4118	2.6505	0.6937	6.0623

The lowest two-impulse ΔV is 3.7969 km/s at the stable manifold trajectory insertion point corresponding to a t_M of 5 nondimensional time units, or 21.7 days.

Even though some of the guesses show lower ΔV than others, it cannot be assumed that any particular PSOIG results in the smallest ΔV after NLP improvement. As a result, all of the initial guesses are supplied individually to *fmincon* in the next section.

4.2.2 NLP Improvement

To improve the PSOIGs using *fmincon*, a few additions need to be made to the setup. First, due to the increased robustness of *fmincon* compared to PSO, in addition to the four parameters varied in the previous section, *fmincon* also varies t_M in the free-variable array. Thus, $\mathbf{P}$ is modified and expressed below.

$$\mathbf{P} = \begin{bmatrix} t_{LEO} \\ V_{x1} \\ V_{y1} \\ t_1 \\ t_M \end{bmatrix} \quad (131)$$

As noted in the previous chapter, *fmincon* is able to handle the cost function and constraint array separately. The cost function is formulated in order to minimize the ΔV s for the first two impulses.

$$J = \Delta V_1 + \Delta V_2 \quad (132)$$

where

$$\begin{aligned} \Delta V_1 &= \sqrt{[\dot{x}_{LEO}(t_{LEO}) - \dot{x}_1(t_0)]^2 + [\dot{y}_{LEO}(t_{LEO}) - \dot{y}_1(t_0)]^2} \\ \Delta V_2 &= \sqrt{[\dot{x}_M(t_M) - \dot{x}_1(t_1)]^2 + [\dot{y}_M(t_M) - \dot{y}_1(t_1)]^2} \end{aligned} \quad (133)$$

In addition, the constraints are identical to those used in the PSO setup in equation (128).

$$\mathbf{C}(\mathbf{P}) = \begin{bmatrix} x_M(t_M) - x_1(t_1) \\ y_M(t_M) - y_1(t_1) \end{bmatrix} = \mathbf{0} \quad (134)$$

The bounds on the variables are identical to those used in the previous section in equation (127) with the addition of the bounds on t_M set to 0.00001 and 15 nondimensional time. The bounds on t_M span the PSOIGs and allow additional freedom to increase the last guess by 5 nondimensional time units.

The gradient information for the objective function and the constraints is supplied to *fmincon*. First, the supplied objective function gradient is given below.

$$\frac{\partial J}{\partial \mathbf{P}} = \left[\begin{array}{ccccc} \frac{\partial J}{\partial t_M} & \frac{\partial J}{\partial V_x} & \frac{\partial J}{\partial V_y} & \frac{\partial J}{\partial t_1} & \frac{\partial J}{\partial t_{LEO}} \end{array} \right]^T \quad (135)$$

Effort must be made to understand the functional dependencies when defining each of the elements. A crucial distinction is that when taking the partial derivative with respect to t_M , the fact that t_M corresponds to negative time must be considered with an additional negative sign. For the constraint gradient, the matrix takes on the form below.

$$\frac{\partial \mathbf{C}}{\partial \mathbf{P}} = \left[\begin{array}{ccccc} \frac{\partial C_1}{\partial t_M} & \frac{\partial C_1}{\partial V_x} & \frac{\partial C_1}{\partial V_y} & \frac{\partial C_1}{\partial t_1} & \frac{\partial C_1}{\partial t_{LEO}} \\ \frac{\partial C_2}{\partial t_M} & \frac{\partial C_2}{\partial V_x} & \frac{\partial C_2}{\partial V_y} & \frac{\partial C_2}{\partial t_1} & \frac{\partial C_2}{\partial t_{LEO}} \end{array} \right]^T \quad (136)$$

In the constraint gradient matrix, the first two columns are elements in the STM, and the fourth column has elements equal to zero. Again, the partial derivatives with respect to t_M need to take into account the negative time aspect previously mentioned. Supplying the gradient information improves the speed of *fmincon* because the gradients do not need to be calculated via finite differencing, which can be computationally expensive.

After supplying *fmincon* with all of the PSOIGs, the best trajectory in terms of ΔV was found using the $t_M = 5$ nondimensional time PSOIG. The trajectory is shown in both the barycentric rotating frame and an Earth-centered inertial frame in Figures 59 and 60 respectively.

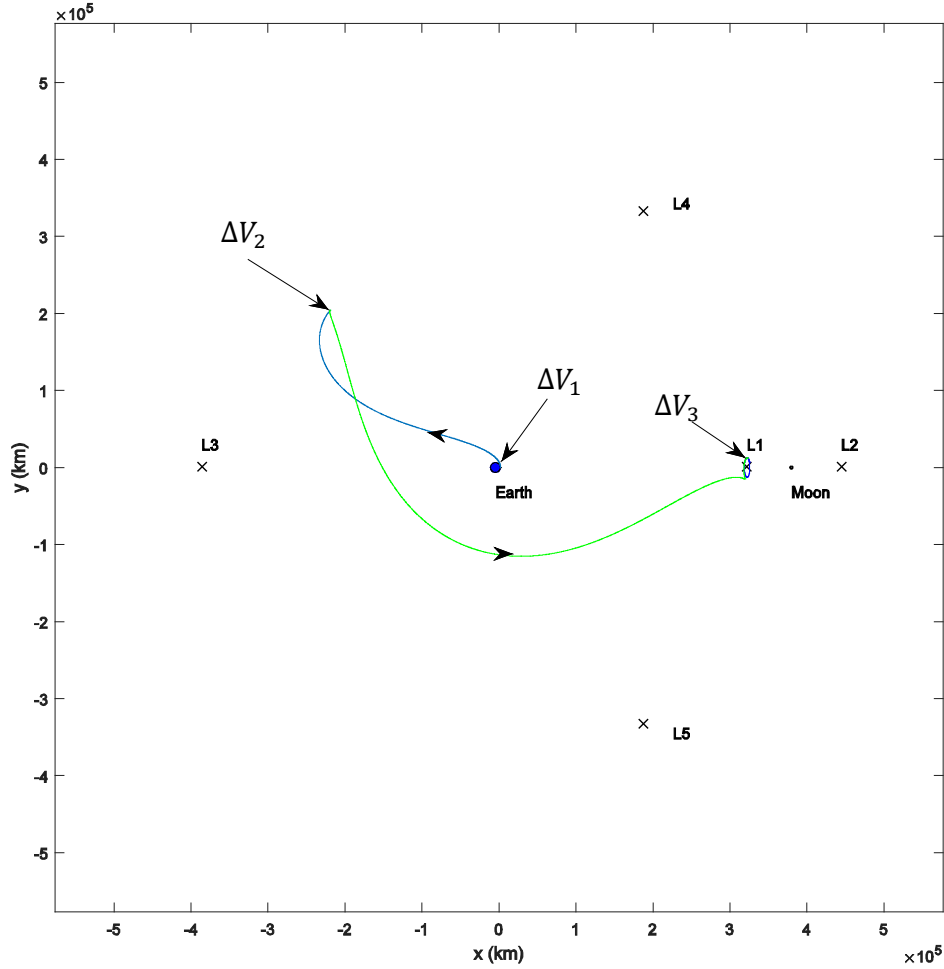


Figure 59. Three impulsive transfer from LEO to L1 LPO in rotating barycentric frame, 23.4 day transfer (ZVCs not shown)

In Figures 59 and 60, it is apparent how different the same trajectory can look depending on the reference frame. The barycentric rotating frame offers a simple view where the relative distances between the spacecraft and the equilibrium points or the primaries is obvious. In the more traditional Earth-centric inertial view, the actual flight path is much easier to discern. A nuance is that the Earth-centric inertial

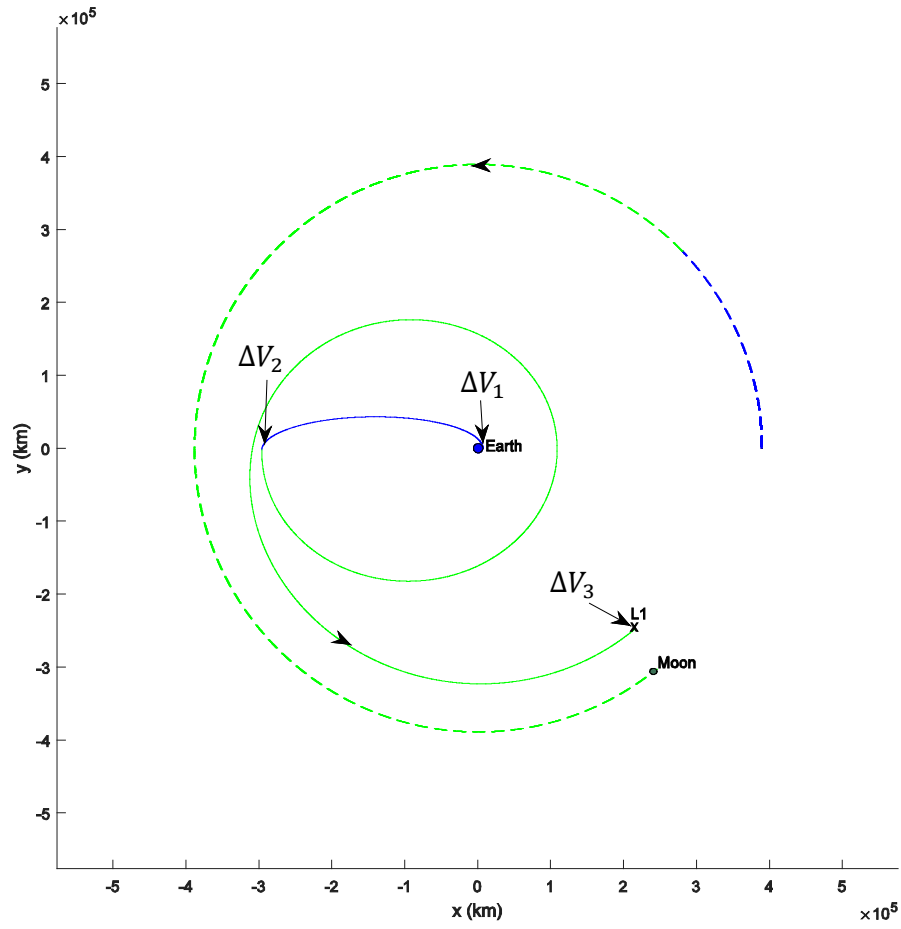


Figure 60. Three impulsive transfer from LEO to L1 LPO in Earth-centered inertial frame, 23.4 day transfer

view depicted is not equatorial, but at the inclination of the plane of the Moon's orbit about the Earth. Both the rotating and inertial frames offer an invaluable and necessary perspective.

The total transfer requires 3.49 km/s where the final burn occurs at the first intersection in position of the stable manifold trajectory approximation and the LPO and costs 0.00115 km/s (1.15 m/s). Note that the true stable manifold trajectory would asymptotically approach the LPO instead of readily supplying an intersection in position. The magnitudes for all three impulses are given in Table 11.

Table 11. LEO to LPO trajectory impulse magnitudes

Impulse	Magnitude (km/s)
ΔV_1	2.92635
ΔV_2	0.57102
ΔV_3	0.00115
ΔV_{total}	3.49851

The time of flight from the initial starting point on the x -axis to LPO insertion is 23.4033 days. The final NLP improved $\mathbf{P}$ array is given below.

$$\mathbf{P} = \begin{bmatrix} t_{LEO} \\ V_{x1} \\ V_{y1} \\ t_1 \\ t_M \end{bmatrix} = \begin{bmatrix} 6.1488 \times 10^{-5} \text{ days} \\ -0.0801471798017611 \text{ km/s} \\ 10.5278443326929 \text{ km/s} \\ 3.3396 \text{ days} \\ 20.0636 \text{ days} \end{bmatrix} \quad (137)$$

The PSOIGs on either side of the $t_M = 5$ guess within 1 nondimensional time converged on the same answer, providing insight as to the convergence window for this locally optimal solution. The other PSOIGs resulted in different converged solutions

at different manifold trajectory insertion points, but for greater ΔV .

As explored in the previous chapter, the question of the practicality of the designed transfer is important. Since the burns are assumed to be impulsive, is it also assumed that the spacecraft is utilizing chemical propulsion with an I_{sp} of 300 s. For an approximate spacecraft mass, NASA’s ARTEMIS mission that sent two satellites to the Earth-Moon L1 and L2 points had a total weight of approximately 250 kg [81]. Assuming an addition 250 kg for the insertion vehicle, the insertion vehicle would require 1,136 kg of fuel, or a total wet mass of 1,636 kg. The number of launch vehicles that can satisfy this mass to LEO requirement is large; however, it may be impractical to fly a design to LEO that requires thousands of kilograms of fuel, when an initial GTO parking orbit is more practical. Prior to conducting such a mission, additional simulations would need to be conducted for a GTO starting orbit in the spatial CR3BP (allows 3-D motion out of the plane of the primaries) allowing for a variable LPO insertion point to make the trajectory more practical.

4.3 Chapter Summary

This chapter covered the methodology and results for designing a three impulse transfer from LEO altitude to an LPO about the Earth-Moon cislunar collinear Lagrange point. The design takes advantage of an approximation of an invariant stable manifold trajectory coast arc to efficiently approach the target LPO. The manifold trajectory is efficient in the fact that it exploits dynamical systems theory to match the dynamical “flow” of the environment. PSO, while initially intended to provide an initial guess for the entire three impulse transfer, had constraint weighting difficulties and was instead used to solve simpler TPBVPs across a range of values for the stable manifold trajectory insertion point. The initial guesses were then supplied to *fmincon* and optimized with the stable manifold trajectory insertion point included as a design

parameter. The final trajectory required 3.49 km/s in ΔV and 23.4 days in time of flight. A final and important note is that the trajectory designed is at best a locally optimal given the PSOIG, and a claim of global optimality cannot be made. The next and final chapter summarizes the present work and provides a discussion on the collective conclusions found in the current investigation.

5. Conclusions

The purpose of this investigation is to provide a viable methodology for designing spacecraft trajectories that are unorthodox or exist in a complex dynamical environment. The existence of robust optimization techniques allows for a variety of problems and scenarios to be solved; however, these algorithms required an initial guess to be initialized. The sensitivity of the system dictates the quality of the initial guess required. For the trajectories designed in this investigation, an initial guess is not readily available and many of the more complex design scenarios exhibited limited regions of convergence. Particle swarm optimization is offered as a tool to generate the initial guess for a locally optimal minimum-fuel trajectory.

PSO boasts the ability to globally search the solution space without any initial conditions. In addition, the algorithm is very simple to implement with a small number of algorithmic parameters when compared to other evolutionary algorithms. PSO is not without faults and has difficulty enforcing equality constraints. In this investigation, a penalty method is employed thereby adding the constraints to the cost function after they have been scaled by a user defined coefficient. The main concern when applying PSO to a new problem is choosing constraint weighting factors such that the algorithm is not ill-conditioned [6]. Due to this concern, some insight into the dynamical environment is useful such that the bounds on the design variable are tight yet not over-constraining. For example, in the case of the trajectory designed in the CR3BP, a constant energy-like quantity unique to the dynamical system is used to provide smart bounds on two of the design variables. Throughout the investigation, experimenting with the PSO coefficients until acceptable results and levels of convergence were gained proved to be more productive than experimenting with manually created initial guesses to supply to the NLP solver. The methodology that worked well and was employed throughout the investigation can be summarized by

the following steps:

Step 1: Directly transcribe the problem into a TPBVP and define the system EOMs.

Step 2: Construct the terminal and path constraints that must be satisfied for the solution to be feasible and define the augmented cost function using a penalty function system.

Step 3: Scale the constraints as well as the cost index such that a dominant term or a large disparity in magnitude does not exist in the PSO augmented cost function. Run the PSO.

Step 4: Check the converged solution. If premature convergence occurs, a change in the constraint weighting coefficients may be warranted. If the constraint weighting factors are not an issue, increase the swarm size for a more exhaustive search or the iteration count for higher levels of convergence.

Step 5: Give the PSOIG to the NLP solver. If the NLP solver does not converge, Step 4 may require additional attention. The problem may also be too large for the PSO to generate an initial guess within the NLP convergence window. If the design space subjected to PSO is greater than fifty dimensions, make simplifying assumptions or apply tighter bounds.

In employing the proposed methodology to the spacecraft trajectories in the current investigation, several conclusions can be made.

1. **The polynomial parameterization approach used for continuous control, used in conjunction with PSO, is useful for small problems.**

In the near-Earth trajectory designs, the polynomial parametrization or spline interpolation approach is successfully employed to turn a continuous time history into a finite number of parameters. However, for more complex scenarios to

be optimized, a decrease in the proximity of the initial guesses to locally optimal solutions is seen. The decrease in the “goodness” of the PSOIGs is a function of the dimension of the design space and not an indication of an inherent limit to the parameterization approach. For example, in the finite planar transfer, the PSOIG proved to be a much better estimate of the optimal solution when compared to the PSOIG for the non-coplanar transfer. The qualitative shape of the control that the polynomial approach needed to approximate was not radically different for either transfer. What did change was a doubling of the problem size. The PSOIGs given fifty or higher dimensioned search spaces appeared to show a greater departure from a locally optimal solution than results from the other smaller dimensioned problems. Based on these results, the polynomial approach, when used in conjunction with PSO, needs to be applied to problems that can be parameterized to a small, finite set of design variables.

2. PSO requires some intuition to properly weigh constraints.

While PSO boasts a freedom from requiring an initial guess, when attempting to solve a constrained problem, some intuition is still required. For example, for the finite burn trajectories, the knowledge that the desired optimal solution does not include superfluous coasting was factored into the appropriate constraint weighting coefficient. Also, in the CR3BP, where there is arguably less readily available intuition to exploit, the constraint weighting for the five parameter problem proved to be too difficult. The problem of constraint weighting is well documented and currently an unsolved problem. However, for difficult problems, systematically augmenting the constraint weighting may be less onerous than manually creating an initial guess within the NLP convergence window.

3. PSO performance can be enhanced with smart bounds.

Another avenue to implement intuition about the problem at hand is through smart bounding of the design parameters. An example of this is seen in Chapter 4 when the value of Jacobi constant was used to bound the design parameters for the three impulse transfer. While not required, smart bounds allow the user to successfully generate an acceptable PSOIG for fewer particles and iterations. Care must be taken to ensure that viable designs are not accidentally being removed by applying overly-restrictive bounds. Also, in applying stringent bounds, a global search of the entire design space is not being conducted. However, for very complex environments, a global search of a justifiably bounded search space may be sufficient.

4. PSO excels at parameter optimization.

When control parameters do not need to be inserted into the shooting problem, PSO excels at solving the TPBVP. As is shown in Chapter 4, PSO can provide an initial guess to the differential correction algorithm if not solve the targeting problem itself. The reason why PSO should not replace differential corrections is that it is computationally expensive and does not exploit the EOVs to efficiently tailor its search directions. PSO also demonstrates the ability to target periodic orbits in the CR3BP, thus, the design applications of PSO are not limited to spacecraft transfers, but may also be applied to exploring other possible behavior in a complex design space.

5.1 Limitations and Future Work Recommendations

The current investigation is marked by limitations in certain areas. First, the PSO algorithm utilized is kept constant throughout all of the test cases. That is, all

algorithmic parameters are kept constant and other PSO variants are not explored. Using other variants such as local PSOs may produce improved results in a few or all of the test cases. In addition, the polynomial interpolation method used for the low-thrust cases only utilizes fourth-order polynomials. An in-depth investigation is required to determine the optimal order of the polynomials or if other parameterization methods not using polynomials are better suited for the test cases conducted. Further stressing the PSO-to-NLP methodology used in the current investigation can be done by including additional, more complex constraints and design parameters. Doing so would allow the user to better discern the limits of this design approach. Next, the trial-and-error method for calculating appropriate constraint weighting factors is a coarse approach and should be further refined in future work. Lastly, a major advantage of EAs is that they are claimed to boast a more global search of the design space. Even though this claim is made, a conclusion about the global optimality of the trajectories designed in the present work cannot be made. Further investigation into the global optimality of trajectories designed using PSO should be conducted to fully exploit the benefit of a more global search.

Based on the given conclusions and limitations, potential areas for future work or in-depth investigation are as follows:

- Extend all the test cases conducted in the current investigation to similar transfers starting at a GTO as opposed to LEO such that more practical solutions are generated.
- Explore the trade-space between changing the degree of the polynomial vs. the number of polynomials to parameterize continuous functions. Also, investigate other parameterization schemes such as using a Fourier series and having PSO optimize the Fourier coefficients.

- Enforce continuity between the polynomial chains as well as allow variable orders of polynomials for different segments. For high rate of change control segments, greater degree polynomials may be advantageous.
- Conduct a more extensive investigation on PSO constraint weighting sensitivity, or employ more advanced techniques such as variable constraint weighting.
- Increase the complexity of the low-thrust transfers by including throttling, variable specific impulse (I_{sp}), and additional constraints such as power restriction due to time in eclipse.
- Use low-thrust as opposed to impulsive burns to fly to the insertion point on the L1 LPO stable manifold. Also, allow for the particular manifold trajectory chosen to be an additional design variable.
- Investigate how local-best vs. global-best PSO variants may be used in the CR3BP to see if PSO can handle additional parameters after the initial shooting process.
- Incorporate a hybrid technique where PSO provides the initial guess to a differential corrections scheme to shoot between two boundary conditions. The converged trajectory is handled within a parent PSO routine that varies the manifold trajectory and insertion point.
- Investigate post-optimality techniques to validate the level of optimality (local or global) of any converged trajectory.

The list provided is not exhaustive but provides a guideline as to where further investigation seems most desirable given the results of the current investigation.

Bibliography

1. R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. New York: Dover, 1971.
2. V. A. Chobotov, *Orbital Mechanics*, 2nd ed. University of Michigan: AIAA, 1991.
3. J. Stuart, *Fuel-Optimal Low-Thrust Transfers Between Libration Point Orbits*, Master's Thesis, Purdue University, 2011. [Online]. Available: https://engineering.purdue.edu/people/kathleen.howell.1/Publications/Masters/2011_Stuart.pdf
4. C. D. Geisel, *Spacecraft Orbit Design in the Circular Restricted Three-Body Problem Using Higher-Dimensional Poincaré Maps*, Ph.D. dissertation, Purdue University, 2013. [Online]. Available: https://engineering.purdue.edu/people/kathleen.howell.1/Publications/Dissertations/2013_Geisel.pdf
5. J. Arora, *Optimum Design*, 3rd ed. Waltham, MA: Academic Press, 2012.
6. B. A. Conway, *Spacecraft Trajectory Optimization*. New York: Cambridge, 2010.
7. G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 8th ed. Hoboken, NJ: Wiley, 2010.
8. Department of Defense, "Joint Publication 3-14: Space Operations," 2013. [Online]. Available: http://dtic.mil/doctrine/new_pubs/jp3_14.pdf [Accessed: 2015-09-22]
9. "SiriusXM: Satellite Radio." [Online]. Available: <http://www.siriusxm.com/> [Accessed: 2016-01-26]
10. "DIRECTV." [Online]. Available: <http://www.directv.com/> [Accessed: 2016-01-26]
11. J. J. Sellers, *Understanding Space*, 3rd ed. New York: McGraw Hill, 2005.
12. Department of Defense, "DoD Directive 3100.10," 2012. [Online]. Available: <http://dtic.mil/whs/directives/corres/pdf/310010p.pdf> [Accessed: 2015-09-22]
13. Department of Defense, "FACT SHEET: Resilience of Space Capabilities," 2011. [Online]. Available: http://www.defense.gov/Portals/1/features/2011/0111_nsss/docs/DoDFactSheet-Resilience.pdf [Accessed: 2016-09-22]
14. United States Air Force, "USAF Posture Statement 2013," Washington DC, 2013. [Online]. Available: <http://www.dtic.mil/dtic/tr/fulltext/u2/a584783.pdf> [Accessed: 2016-03-05]

15. Office of the Assistant Secretary of Defense for Homeland Defense and Global Security, “Space Domain Mission Assurance : A Resilience Taxonomy,” 2015.
16. National Aeronautics and Space Administration, “AsiaSat 3.” [Online]. Available: <http://nssdc.gsfc.nasa.gov/nmc/masterCatalog.do?sc=1997-086A> [Accessed: 2015-09-09]
17. Boeing, “Boeing: Worlds First All-Electric Propulsion Satellite Begins Operations,” 2015. [Online]. Available: <http://boeing.mediaroom.com/2015-09-10-Boeing-World-s-First-All-Electric-Propulsion-Satellite-Begins-Operations> [Accessed: 2015-09-22]
18. W. E. Wiesel, *Spaceflight Dynamics*, 3rd ed. Beaver Creek, OH: Aphelion Press, 2010.
19. M. A. Finocchiaro, *The Galileo Affair: A Documentary History*. Oakland, CA: University of California Press, 1989.
20. J. Dreyer, *History of the Planetary Systems from Thales to Kepler*. Cambridge, United Kingdom: Cambridge University Press, 1906.
21. I. Newton, “Philosophiæ Naturalis Principia Mathematica,” 1687.
22. M. J. H. Walker and J. Ireland, B., Owens, “A Set of Modified Equinoctial Elements,” *Celestial Mechanics*, vol. 36, pp. 409–419, 1985.
23. W. E. Wiesel, *Modern Astrodynamics*, 2nd ed. Beaver Creek, OH: Aphelion Press, 2010.
24. V. Szebehely, *Theory of Orbits*. New York: Academic Press, 1967.
25. A. E. Roy, *Orbital Motion*, 4th ed. New York: Taylor and Francis Group, 2005.
26. C. D. Murray and S. F. Dermott, *Solar System Dynamics*. Cambridge, United Kingdom: University of Cambridge Press, 1999.
27. C. D. Geisel, “MECH733A Graduate Course: Numerical Methods for Orbit Design.” Wright-Patterson Air Force Base, OH: Air Force Institute of Technology, 2015.
28. H. G. Shuster and W. Just, *Deterministic Chaos: An Introduction*, 4th ed. Hoboken, NJ: Wiley, 2006.
29. H. Curtis, *Orbital Mechanics: For Engineering Students*. Burlington, MA: Elsevier Science, 2005.
30. M. Leonard, *Methods of Analytical Dynamics*. New York: Dover, 2010.

31. S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. New York: Springer, 1990.
32. J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2010.
33. K. C. Howell and H. J. Pernicka, "Numerical Determination of Lissajous Trajectories in the Restricted Three-Body Problem," *Celestial Mechanics*, vol. 41, 1988.
34. D. Grebow, *Generating Periodic Orbits in the Circular Restricted Three-Body Problem with Applications to Lunar South Pole Coverage*, Master's Thesis, Purdue University, 2006. [Online]. Available: https://engineering.purdue.edu/people/kathleen.howell.1/Publications/Masters/2006_Grebow.pdf
35. T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*. Ann Arbor, MI: Springer, 1989.
36. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York: Springer, 1983.
37. J. M. Longuski, J. J. Guzmán, and J. E. Prussing, *Optimal Control with Aerospace Applications*. New York: Springer, 2014.
38. A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Washington DC: Taylor and Francis Group, 1975.
39. D. T. Greenwood, *Classical Dynamics*. New York: Dover, 1997.
40. S. A. Stanton, *Finite Set Control Transcription for Optimal Control Applications*, Ph.D. dissertation, University of Texas at Austin, 2009. [Online]. Available: <https://repositories.lib.utexas.edu/bitstream/handle/2152/6631/stantons82670.pdf?sequence=2&isAllowed=y>
41. M. Pontani and B. A. Conway, "Swarming Theory Applied to Space Trajectory Optimization," in *Spacecraft Trajectory Optimization*, 2010, pp. 263–293.
42. D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. New York: Springer, 2002.
43. R. Kennedy, J., Eberhart, "Particle Swarm Optimization," in *Proceedings of IEEE International Conference on Neural Networks*, Perth, Western Australia, 1995.
44. S. Koziel and Z. Michalewicz, "Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19–44, 1999.

45. B. Prasad, "A Class of Generalized Variable Penalty Methods," *Journal of Optimization Theory and Applications*, vol. 35, no. 2, pp. 159–182, 1981.
46. T. Matsui and M. Sakawa, "Particle Swarm Optimization Combining Diversification and Intensification Through Complex Networks for Nonlinear Programming Problems," in *IEEE International Conference on Neural Networks*, Brisbane, Australia, 2012, pp. 1363–1367.
47. T. N. Edelbaum, "Propulsion Requirements for Controllable Satellites," *American Rocket Society*, vol. 31, no. 8, pp. 1079–1089, 1961.
48. W. E. Wiesel and A. Salvatore, "Optimal Many-Revolution Orbit Transfer," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 1, pp. 155–157, 1985.
49. E. Burt, "On Space Manoeuvres with Continuous Thrust," *Planetary and Space Science*, vol. 15, no. 1, pp. 103–122, 1967.
50. J. E. Pollard, "Simplified Approach for Assessment of Low-Thrust Elliptical Orbit Transfers," in *Proceedings of the International Electric Propulsion Conference 1997 (IEPC97)*, no. 1, Cleveland, OH, 1997, pp. 979–986.
51. W. Hohmann, *The Attainability of Heavenly Bodies*. Washington DC: NASA Technical Translation F-44, 1960.
52. D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 3rd ed. New York: Microcosm Press, 2007.
53. D. B. Spencer and R. D. Culp, "Designing Continuous-Thrust Low-Earth-Orbit to Geosynchronous-Earth-Orbit Transfers," *Journal of Spacecraft and Rockets*, vol. 32, no. 6, 1995.
54. J. J. Pelouch, "Low-Thrust Chemical Orbit Transfer Propulsion," in *Fifteenth Joint Propulsion Conference Proceedings*, Las Vegas, NV, 1979.
55. J. V. Breakwell and D. C. Redding, "Optimal Low-Thrust Transfers to Synchronous Orbit," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 2, pp. 148–155, 1984.
56. M. Pontani and B. Conway, "Optimal Low-Thrust Orbital Maneuvers via Indirect Swarming Method," *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 272–292, 2014.
57. M. Pontani and B. Conway, "Minimum-Fuel Finite-Thrust Relative Orbit Maneuvers via Indirect Heuristic Method," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 5, pp. 913–924, 2015. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/1.G000157>

58. V. L. Rauwolf, G. A., Coverstone-Carroll, "Near-Optimal Low-Thrust Orbit Transfers Generated by a Genetic Algorithm," *Journal of Spacecraft and Rockets*, vol. 33, no. 6, 1996.
59. J. C. Eisenreich, *Optimization of Low Thrust Spacecraft Trajectories Using a Genetic Algorithm*, Master's Thesis, University of Texas at Austin, 1998.
60. C. R. Bessette and D. Spencer, "Identifying Optimal Interplanetary Trajectories through a Genetic Approach," in *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, CO, 2006.
61. B. Wall and C. B. A., "Near-Optimal Low-Thrust Earth-Mars Trajectories via a Genetic Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, 2005.
62. R. D. Russell and L. F. Shampine, "A Collocation Method for Boundary Value Problems," *Numerical Mathematics*, vol. 19, 1972.
63. C. R. Hargraves and S. W. Paris, "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, 1987.
64. G. Birkhoff and C. R. de Boor, "Piecewise Polynomial Interpolation," in *Approximation of Functions*. New York: Elsevier Science, 1965.
65. C. Martin and B. A. Conway, "Optimal Low-Thrust Trajectories Using Stable Manifolds," in *Spacecraft Trajectory Optimization*. New York: Cambridge, 2010, ch. 9, pp. 238–262.
66. J. McMahon, W. Anthony, T. Critz, M. Nazari, M. Deilami, A. Larsen, E. Butcher, and G. Born, "Optimal Transfers with Guidance to the Earth-Moon L1 and L3 Libration Points using Invariant Manifolds: A Preliminary Study," in *AIAA/AAS Astrodynamics Specialist Conference*, Minneapolis, MN, 2012.
67. A. J. Abraham, D. B. Spencer, and T. J. Hart, "Particle Swarm Optimization of 2-Manuver, Impulsive Transfers From LEO to Lagrange Point Orbits via Shooting," in *24th International Symposium on Space Flight Dynamics*, Laurel, MD, 2014.
68. The MathWorks Inc., "MATLAB," Version: 8.1.0.604, Natick, MA, 2013.
69. R. D. Falck, "Optimization of Low-Thrust Spiral Trajectories by Collocation," in *AIAA/AAS Astrodynamics Specialist Conference*. Minneapolis, MN: AIAA, 2012.
70. J. A. Kechichian, "The Treatment of the Earth Oblateness Effect in Trajectory Optimization in Equinoctial Coordinates," *Acta Astronautica*, vol. 40, no. 1, 1997.

71. J. Epperson, “On the Runge Example,” *The American Mathematical Monthly*, vol. 94, pp. 329–341, 1987.
72. The Mathworks Inc., “Choosing a Solver: Optimization Decision Table.” [Online]. Available: <http://www.mathworks.com/help/optim/ug/choosing-a-solver.html> [Accessed: 2016-01-31]
73. P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *Society for Industrial and Applied Mathematics*, vol. 47, no. 1, pp. 99–131, 2005.
74. J. R. Wertz, D. F. Everett, and J. J. Puschell, *Space Mission Engineering: The New SMAD*. Hawthorne, CA: Microcosm Press, 2011.
75. A. Herman, “Trajectory Modeling and Optimization For Future Space Systems,” Air Force Research Lab, Kirtland Air Force Base, NM, Tech. Rep., 1998.
76. F. Duret, “Staging or Upper Stage Reignition,” *Acta Astronautica*, vol. 51, no. 1, pp. 33–38, 2002.
77. Los Angeles Air Force Base, “Global Positioning System (GPS) III Satellite,” 2014. [Online]. Available: <http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=18830> [Accessed: 2016-02-15]
78. United Launch Alliance, “Delta IV Payload Planner’s Guide, June 2013,” 2013. [Online]. Available: http://www.ulalaunch.com/uploads/docs/Launch_Vehicles/Delta_IV_Users_Guide_June_2013.pdf [Accessed: 2016-02-15]
79. L. F. Shampine and M. W. Reichelt, “The MATLAB ODE Suite.” [Online]. Available: <http://www.mathworks.com> [Accessed: 2016-02-09]
80. A. E. Roy and M. W. Ovenden, “On the Occurrence of Commensurable Mean Motions in the Solar System: The Mirror Theorem,” *MNRAS*, vol. 115, no. 3, pp. 296–309, 1955.
81. National Aeronautics and Space Administration, “ARTEMIS: The First Earth Moon Libration Orbiter,” 2009. [Online]. Available: http://www.nasa.gov/mission_pages/themis/news/artemis-orbit.html [Accessed: 2016-02-16]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, 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 Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 24-03-2016		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) Aug 2014 - Mar 2016	
4. TITLE AND SUBTITLE Minimum-Fuel Trajectory Design in Multiple Dynamical Environments Utilizing Direct Transcription Methods and Particle Swarm Optimization				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Zurita, Alfredo G., Jr., 2d Lt, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENY-MS-16-M-250	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL Space Vehicles Directorate 3550 Aberdeen Avenue SE Kirtland AFB NM, 87117-5776 POC: N/A Principal DoD Space Advisor Staff 1670 Air Force Pentagon Washington DC, 20330-1670C 703-693-5799 POC: Maj Stuart A. Stanton				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RV	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT Particle swarm optimization is used to generate an initial guess for designing fuel-optimal trajectories in multiple dynamical environments. Trajectories designed in the vicinity of Earth use continuous or finite low-thrust burning and transfer from an inclined or equatorial circular low-Earth-orbit to a geostationary orbit. In addition, a trajectory from near-Earth to a periodic orbit about the cislunar Lagrange point with minimized impulsive burn costs is designed within a multi-body dynamical environment. Direct transcription is used in conjunction with a nonlinear optimizer to find locally-optimal trajectories given the initial guess. The near-Earth transfers are propagated at low-level thrust where neither the very-low-thrust spiral solution nor the impulsive transfer is an acceptable starting point. The very-high-altitude transfer is designed in a multi-body dynamical environment lacking a closed-form analytical solution. Swarming algorithms excel given a small number of design parameters. When continuous control time histories are needed, employing a polynomial parameterization facilitates the generation of feasible solutions. For design in a circular restricted three-body system, particle swarm optimization gains utility due to a more global search for the solution, but may be more sensitive to boundary constraints. Computation time and constraint weighting are areas where a swarming algorithm is weaker than other approaches.					
15. SUBJECT TERMS spacecraft trajectory optimization, optimization, particle swarm optimization, low-thrust, multi-body dynamical environment, circular restricted three-body problem, direct transcription, nonlinear programming					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Christopher D. Geisel, Ph.D., AFIT/ENY
U	U	U	UU	184	19b. TELEPHONE NUMBER (include area code) (937)255-3636 x4237; christopher.geisel@afit.edu