

Copyright

by

Chad Wilson Smith

2008

**An Onboard Targeting Algorithm with Earth-Return
Applications**

by

Chad Wilson Smith, B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2008

**An Onboard Targeting Algorithm with Earth-Return
Applications**

**Approved by
Supervising Committee:**

To My Parents:

Delbert Lawrence and Michele Kay Smith

And My Grandparents:

Herbert and Mae Lentsch

Acknowledgments

This research was performed at the University of Texas at Austin and funded by the National Aeronautics and Space Administration under NASA Grant Number MNX07AR46G. Through this grant I was allowed the honor of working for Mike Weeks and Chris D'Souza. I would like to thank them for being excellent mentors during my study at UT and creating a wonderful working environment.

In acknowledging those most instrumental to this research, my greatest gratitude is reserved for my advisor, Dr. Belinda Marchand, for her guidance and motivation. Without her level of interest in my academic endeavors, my research could not have reached its potential. Through working with Dr. Marchand, I attained a most rewarding educational experience at UT.

I would also like to thank Dr. Dale Chimenti and Dr. Cesar Ocampo for their additional influences in my route to academic achievement. Their ability to articulate even the most challenging of concepts provided me with invaluable knowledge in my course work. Furthermore, the individual interest they showed me during my studies encouraged me to dedicate myself to the subject matter and develop my passion.

In addition to my advisor and professors, I would like to thank Stacy Weinsten, Rob Maddock, Dr. Ted Sweetser, Richard Mattingly, Fernando Abillera, and the rest of my colleagues in the Inner Planet Mission Analysis Group at the Jet Propulsion Laboratory. JPL provided me with a job that I had dreamed of

my entire life. However, the experience could not have been so fulfilling without my wonderful coworkers. It is because of them that my passion for astrodynamics continued to grow, ultimately motivating me to attend graduate school.

I am also indebted to my fellow UT colleagues Stuart Stanton, Divya Thakur, and Sara Scarritt for their instrumental influence during my graduate studies. Each of them went above and beyond their responsibilities as research colleagues and made my time at UT much more enjoyable.

Finally, I would like to thank my friends and family. Without them, this would all remain a dream.

CHAD WILSON SMITH

The University of Texas at Austin

August 2008

An Onboard Targeting Algorithm with Earth-Return Applications

Chad Wilson Smith, M.S.E
The University of Texas at Austin, 2008

Supervisor: Belinda Marchand

Over the course of the next decade, NASA plans to send a crew exploration vehicle (CEV) to the Moon as part of the Constellation program. The development of a suitable onboard Earth-return targeting algorithm is one of many tasks required in the execution of this mission. The targeting algorithm envisioned is capable of targeting a specified Earth entry state, obeys the propulsive limits of the spacecraft, and remains executable at any point along the return transfer.

A modified differential correction scheme is selected as a candidate for the targeting algorithm. This process, known as the Two Level Corrector (TLC), acts to find a feasible solution in the neighborhood of an initial guess. As its name implies, the TLC process consists of two separate phases. The first phase employs a basic targeter to achieve position continuity across the transfer, while the second phase acts to achieve velocity continuity and satisfy any additional constraints.

The goal of this investigation is to augment the theory of the TLC process to accommodate the requirements of an onboard targeting algorithm for the Earth-return portion of the upcoming Orion lunar missions, which includes Trans-Earth Injection (TEI) and trajectory correction maneuvers (TCM's). The work presented in this investigation demonstrates the algorithm as a proof of concept. Different mission constraints are targeted over a series of examples to provide supporting evidence. In addition, a finite burn formulation is devised and adapted to the TLC process. Finally, an end-to-end numerical optimization of the transfer is presented and contrasted against a converged solution from the targeting algorithm.

Contents

| | |
|--|-------------|
| Acknowledgments | v |
| Abstract | vii |
| List of Tables | xiii |
| List of Figures | xiv |
| Chapter 1 Introduction | 1 |
| 1.1 Problem Description | 1 |
| 1.2 Prior Work | 4 |
| 1.3 Document Organization | 7 |
| Chapter 2 Background Information | 9 |
| 2.1 Coordinate Frames | 9 |
| 2.2 Dynamical Model | 11 |
| 2.3 The Variational Equation | 14 |
| Chapter 3 Impulsive Trajectory Targeting | 18 |
| 3.1 Level-One Correction using Impulsive Maneuvers | 19 |
| 3.1.1 Time Dependent Level-One Correction | 20 |

| | | |
|------------------------------------|--|-----------|
| 3.2 | Level-Two Correction | 22 |
| 3.3 | Constraints | 27 |
| 3.3.1 | Position Vector | 30 |
| 3.3.2 | Velocity Vector | 32 |
| 3.3.3 | Altitude | 32 |
| 3.3.4 | Earth Entry Latitude | 33 |
| 3.3.5 | Earth Entry Longitude | 34 |
| 3.3.6 | Entry Flight Path Angle | 38 |
| 3.3.7 | Earth Entry Flight Path Azimuth | 40 |
| 3.3.8 | Time | 44 |
| 3.3.9 | Constraining Individual Maneuvers | 44 |
| 3.3.10 | Constraining the Total $\Delta\bar{V}$ | 45 |
| 3.3.11 | Constraining the Departure Velocity | 46 |
| 3.3.12 | Constraining a Patch State | 46 |
| Chapter 4 Targeting Results | | 48 |
| 4.1 | Initial Guess | 49 |
| 4.1.1 | Burn 1 | 49 |
| 4.1.2 | Burn 2 | 51 |
| 4.1.3 | Burn 3 | 52 |
| 4.1.4 | Application of the Initial Guess Generator | 54 |
| 4.1.5 | Patch Point Selection | 56 |
| 4.2 | Numerical Integration | 58 |
| 4.3 | Interpreting the Results | 58 |
| 4.4 | TLC Results: Five Example Cases | 59 |

| | | |
|------------------|--|------------|
| Chapter 5 | Two Level Correction with Finite Thrust | 70 |
| 5.1 | Alterations to the Physical Model | 71 |
| 5.2 | Level-One Correction Using Finite Thrust | 73 |
| 5.3 | Level-Two Correction Using Finite Thrust | 74 |
| 5.4 | Example Case | 81 |
| | | |
| Chapter 6 | Numerical Optimization | 83 |
| 6.1 | Parameter Optimization Theory | 84 |
| 6.1.1 | Unconstrained Parameter Optimization | 84 |
| 6.1.2 | Parameter Optimization with Equality Constraints | 85 |
| 6.1.3 | Parameter Optimization with Inequality Constraints | 86 |
| 6.2 | Unconstrained Minimization | 87 |
| 6.2.1 | One-Dimensional Line Search | 88 |
| 6.2.2 | First-Order Methods | 93 |
| 6.2.3 | Second-Order Methods | 95 |
| 6.3 | Constrained minimization | 96 |
| 6.3.1 | Penalty Functions and Unconstrained Minimization | 97 |
| 6.3.2 | Constrained Minimization Techniques | 99 |
| 6.3.3 | Finite Differencing | 100 |
| 6.3.4 | Optimization Techniques | 102 |
| 6.4 | Applied Numerical Optimization | 103 |
| 6.4.1 | Three-Burn TEI Optimization Process | 103 |
| 6.4.2 | Three-Burn TEI Optimization Examples | 109 |
| | | |
| Chapter 7 | TLC vs. Optimal | 121 |
| | | |
| Chapter 8 | Conclusions and Suggested Future Investigations | 125 |

| | |
|---------------------|------------|
| Bibliography | 127 |
| Vita | 130 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | Summary of the TLC Example Results | 68 |
| 4.2 | Summary of the Initial Guess Used in Each Example Case | 68 |
| 4.3 | Summary of the Entry States for Example Cases | 69 |

List of Figures

| | | |
|------|--|----|
| 1.1 | Trans-Earth Injection as Currently envisioned [1] | 3 |
| 1.2 | Early Apollo Guidance Strategy [2] | 6 |
| 1.3 | Angle Measurement For Determining Guidance Error [2] | 6 |
| 2.1 | ECI frame | 10 |
| 2.2 | MCI frame | 11 |
| 2.3 | Differential vs. Variation [3] | 17 |
| 3.1 | Lambert Targeting Example | 21 |
| 3.2 | (a) Level 1 correction (b) Level 2 correction applied over three patch points | 23 |
| 3.3 | Velocity continuity enforced on a simple example | 28 |
| 3.4 | Targeting Position | 31 |
| 3.5 | Targeting Altitude | 34 |
| 3.6 | Right Ascension | 36 |
| 3.7 | Flight Path Angle | 39 |
| 3.8 | Targeting the Flight Path Angle | 40 |
| 3.9 | Targeting the Flight Path Azimuth | 43 |
| 3.10 | Targeting the Altitude While Holding the Second Patch Point Fixed | 47 |

| | | |
|------|---|----|
| 4.1 | Rendition of Maneuver 1 | 51 |
| 4.2 | Rendition of Maneuver 2 | 52 |
| 4.3 | Rendition of Maneuver 3 | 53 |
| 4.4 | Example of Initial Guess Implementation at the Moon, MCI | 54 |
| 4.5 | EMS Angle Vs. Days from jd=2457000.5 | 55 |
| 4.6 | EMS Angle Vs. Days from jd=2457000.5 | 55 |
| 4.7 | Maneuver Sum vs. Days from jd=2457000.5 | 56 |
| 4.8 | Patch Point Selection | 57 |
| 4.9 | Case 1: Initial Guess (MCI) | 61 |
| 4.10 | Case 1: Converged Solution (ECI) | 62 |
| 4.11 | Case 2: Converged Solution (ECI) | 63 |
| 4.12 | Case 3: Initial Guess (MCI) | 64 |
| 4.13 | Case 3: Converged Solution (ECI) | 64 |
| 4.14 | Case 4: Initial Guess (MCI) | 65 |
| 4.15 | Case 4: Converged Solution (ECI) | 66 |
| 4.16 | Case 5: Initial Guess (MCI) | 67 |
| 4.17 | Case 5: Converged Solution (ECI) | 68 |
| 5.1 | An Example of an Finite Thrust trajectory | 74 |
| 5.2 | Targeting Altitude with a Thrust-Coast Arc | 80 |
| 5.3 | Thrust-Coast-Coast-Coast-Coast Arc | 82 |
| 6.1 | Representation of an Interval of Uncertainty | 89 |
| 6.2 | Representation of an Interval of Uncertainty in terms of σ | 90 |
| 6.3 | Reduction of the Interval of Uncertainty | 91 |
| 6.4 | Gradient Method Applied to a Quadratic Function: 7 Iterations | 94 |

| | | |
|------|--|-----|
| 6.5 | Variable Metric Method With a BFGS Update Applied to a Quadratic Function: 4 Iterations | 97 |
| 6.6 | Gradient Method Applied to a Quadratic Function: 17 Iterations . . | 98 |
| 6.7 | Variable Metric Method With a BFGS Update Applied to a Quadratic Function: 6 Iterations | 99 |
| 6.8 | Single and Multiple Shooting Optimization Methods | 104 |
| 6.9 | Multiple shooting applied to the lunar TEI problem | 105 |
| 6.10 | Optimization Example 1: (left) Transfer in ECI, (right) Transfer in MCI | 112 |
| 6.11 | Optimization Example 1: Close View of Lunar Departure (MCI) . . | 112 |
| 6.12 | Optimization Example 2: (left) Transfer in ECI, (right) Transfer in MCI | 114 |
| 6.13 | Optimization Example 2: Close View of Lunar Departure (MCI) . . | 114 |
| 6.14 | Optimization Example 3: (left) Transfer in ECI, (right) Transfer in MCI | 117 |
| 6.15 | Optimization Example 3: Close View of Lunar Departure (MCI) . . | 117 |
| 6.16 | Optimization Example 4: (left) Transfer in ECI, (right) Transfer in MCI | 119 |
| 6.17 | Optimization Example 4: Close View of Lunar Departure (MCI) . . | 120 |
| 7.1 | Comparison 1: Targeting vs. Optimal (ECI) | 122 |
| 7.2 | Comparison 1: Targeting vs. Optimal (MCI) | 123 |
| 7.3 | Comparison 2: Targeting vs. Optimal (ECI) | 124 |
| 7.4 | Comparison 2: Targeting vs. Optimal (MCI) | 124 |

Nomenclature

| | |
|-----------------|--------------------------------|
| α | Two Level Corrector Constraint |
| $(\bar{\cdot})$ | Vector Notation |
| \bar{a} | Acceleration Vector |
| \bar{b} | Controls Vector in TLC Process |
| \bar{c} | Constraint Vector |
| \bar{e} | Error Vector in TLC Process |
| \bar{h} | Angular Momentum Vector |
| \bar{R} | Position Vector |
| \bar{u} | Thrust Direction |
| \bar{V} | Velocity Vector |
| \bar{X} | State Vector |
| \bar{x} | Parameter Vector |
| \cdot | Dot Product |

| | |
|-----------------|-------------------------------------|
| $\Delta\bar{V}$ | Change in Velocity Vector (Delta-V) |
| Δ | Change |
| δ | Variation |
| \dot{m} | Mass Flow Rate |
| Γ | Penalty Function Weight |
| γ | Flight Path Angle |
| $\hat{(\cdot)}$ | Unit Vector Notation |
| \hat{e} | East Direction |
| \hat{n} | North Direction |
| \hat{r} | Down Direction |
| \mathbb{C} | Moon |
| μ | Gravitational Parameter |
| ν | Lagrange Multiplier |
| \odot | Sun |
| Ω | Longitude of Ascending Node |
| ω | Argument of Periapsis |
| \oplus | Earth |
| ∂ | Partial Derivative |
| ϕ | Latitude |

| | |
|---------------------------|--|
| Ψ | Lagrange Multiplier |
| σ | Step Size |
| τ_0 | Time From Epoch Along the Initial Orbit |
| Θ | Right Ascension |
| θ | Longitude |
| θ_{g0} | Greenwich Hour Angle |
| $(\tilde{\cdot})$ | Matrix Notation |
| $\tilde{\Phi}_{t_f, t_0}$ | State Transition Matrix From t_0 to t_f |
| \tilde{I} | Identity Matrix |
| \tilde{M} | State Relationship Matrix in TLC Process |
| \times | Cross Product |
| φ | Portion of the Cost Function that Contains Independent Variables |
| g_0 | Gravity Acceleration at Sea Level |
| G_x | First Partial Derivative of Numerical Function With Respect to x |
| G_{xx} | Hessian, or Second Partial Derivative With Respect to x |
| I_{sp} | Specific Impulse |
| t_0 | Initial Time |
| t_f | Final Time |
| Az | Flight Path Azimuth |

BFGS Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton Method

CEV Crew Exploration Vehicle

d Differential

ecc Eccentricity

ECI Earth-Centered Inertial Frame

EME2000 Earth-Mean Equator Frame Referenced With Respect to j2000

EMS Earth-Moon-Spacecraft Angle

G Numerical Function to be Minimized by NLP

HOT's Higher Order Terms

inc Inclination

J Cost Function

jd Julian Date

k Indicates a General Patch Point number

m Mass

MCI Moon-Centered Inertial Frame

NLP Nonlinear Program

OCT Optimal Control Theory

ODS One-Dimensional Search

t Time

TCM Trajectory Correction Maneuvers

TEI Trans-Earth Injection

TLC Two Level Corrector

TPBVP Two-Point Boundary Value Problem

UT Universal Time

Chapter 1

Introduction

On January 14th, 2004 President Bush stood at NASA headquarters in Washington D.C. and spoke words that had a large and lasting impact on the members of the space community across the nation.

Today I announce a new plan to explore space and extend a human presence across our solar system. We will begin the effort quickly, using existing programs and personnel. We will make steady progress one mission, one voyage, one landing at a time.

In particular, the new plan for the U.S.-manned space-flight program mentioned here includes the first flights of a new Crew Exploration Vehicle (CEV), by 2014, and the return of humans to the Moon before 2020.

1.1 Problem Description

While traveling to the Moon is a task previously accomplished, the latest initiative poses a different set of objectives and challenges. These upcoming lunar missions

serve to strengthen the existing knowledge base and technical confidence necessary to take man beyond the Moon. Rather than providing a quick means of lunar transportation, the CEV incorporates the advantages of modern technology and improves on the method employed during the Apollo era. It is also explicitly required, for these upcoming missions, that the CEV have the ability to return safely and autonomously to Earth even in the event that communication is lost with the ground. This gives way to the necessity of an onboard targeting algorithm.

The factors that influence the design of this guidance algorithm are: limited knowledge of physical data, limited knowledge of instrumentation errors, limited data processing capability aboard the spacecraft, and accuracy of the numerical guidance method itself. These four items represent the sources of guidance error. Since the first two cannot be improved by the selection of the algorithm, the third and fourth items are the focus of this study.

The Earth-return guidance algorithm is required to be applicable to two phases of the return trajectory. The first is referred to as Trans-Earth injection (TEI). A three-burn TEI sequence as it is currently envisioned [4]: 1) raises the apoapsis of the lunar parking orbit to allow for a period of up to 48 hours, 2) adjusts the orbital plane at apoapsis to coincide with the desired entry plan at Earth, and 3) injects the spacecraft onto an Earth-bound trajectory from periapsis. An artist's rendition of this three-burn scenario is illustrated in Figure 1.1 [1]. Of course, conceptually, the three-burn sequence relies on two-body dynamics and is therefore only an approximation to the true sequence. An optimal three-burn TEI sequence, for example, loosely follows this format in the full nonlinear dynamical model, changing plane and various orbit characteristics with the execution of each maneuver. Once the TEI sequence is complete, Earth entry targeting ensues. This

step is comprised of a series of small trajectory correction maneuvers (TCM's) that offset errors due to the execution of TEI or other errors occurring during transit (i.e. crew movement, jettisons, etc.).

The algorithm itself requires operational efficiency and reliability. Efficiency is associated with small computation times that are not taxing to the onboard processors. This computation constraint removes any potential for loading a numerical optimizer onboard the shuttle, as even the most well-tuned trajectory optimization routine is computationally burdensome to the CEV, "Orion". Reliability of the algorithm then implies the ability to find a usable solution consistently in the application to the Earth-return.

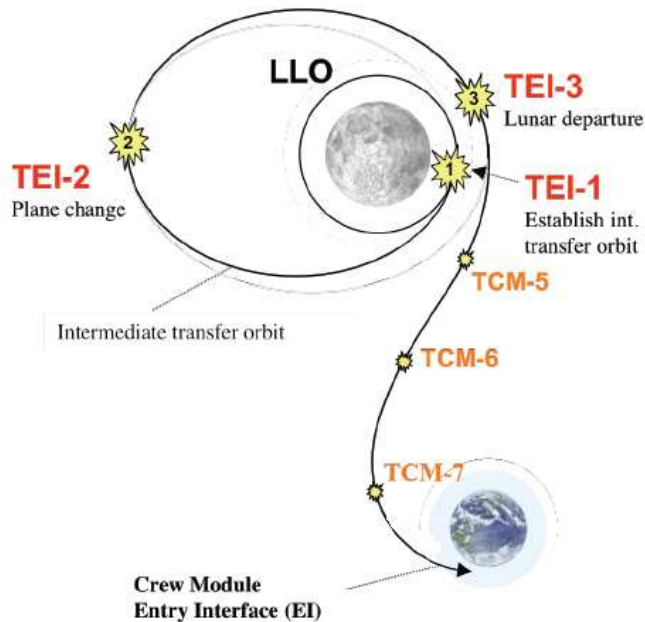


Figure 1.1: Trans-Earth Injection as Currently envisioned [1]

Given the time and mission constraints present on this midcourse guidance

algorithm, a Two Level Corrector (TLC) is identified as a suitable option for onboard targeting. This algorithm identifies nearby solutions that satisfy the imposed set of constraints. The process requires an initial guess, or reference trajectory, and is sensitive to the quality of this guess.

1.2 Prior Work

Onboard guidance is an area that has been of interest since the beginning of space flight. In general, the guidance process involves: 1) vehicle state measurement, 2) computation of control actions necessary to properly adjust position and velocity, and 3) delivery of suitable adjustment commands to the vehicle's control system. The guidance algorithm under consideration for this investigation pertains to the second of these three components in the guidance process. The goal of this type of targeting algorithm is to provide onboard trajectory guidance using minimal computation.

While the work previously completed in the area of midcourse guidance algorithms is extensive, it can be highlighted by a discussion of implicit and explicit techniques. A classification of these methods has previously been performed by Slater and Stern [5]. Implicit methods involve a linearized approximation to the dynamics with respect to some reference trajectory, while explicit methods attempt to model perturbations using perturbed Keplerian orbits. Traditionally, explicit methods require more computation but are also more robust because they do not require a pre-computed reference trajectory. On the other hand, it is assumed that the guidance routine ultimately serves as a backup and that radio communications are maintained throughout the midcourse correction. Subsequently, the accuracy of the implicit method is enhanced with the updated reference trajectories based on

ground support data.

Examples of implicit methods include multi-body and two-body linearized dynamics, each of which rely on a process called differential corrections. A differential corrections procedure works to achieve a trajectory that is continuous in position, given an initial state and a targeted terminal state. The process uses a linear approximation of the spacecraft's motion with respect to some reference. The formulation of the method itself closely resembles a Newton-root finding technique.

Differential corrections is the basis for the implicit guidance algorithm used during the Apollo missions and is also the basis of the first level of the implicit TLC process presented in this study. A summary of the Apollo guidance effort is presented by Battin [2]. This summary indicates that the initial Apollo missions use a reference trajectory that initiates at the departure state and terminates at the entry state. When any deviation occurs from this reference, the implicit guidance technique uses a combination of matrix operations and the position displacement to determine a velocity correction. This idea is captured in Figure 1.2 [2].

During later developments [2], a different approach is employed that abandons the use of a pre-existing reference trajectory. In place of the pre-determined reference, the current best estimate of the spacecraft's state is propagated forward using numerical integration. Linearizing about this arc, the error in the final state with respect to the targeted state is used to evaluate the required TCM. The disparity between the final state and the targeted state is represented by a heading angle, A , measured with respect to the stars. This is shown Figure 1.3 [2]. Given six measurements of this angle, along with the basic navigation equation, the correction needed in both the position and velocity can be extrapolated.

The TLC process possesses enhanced targeting capabilities in contrast with

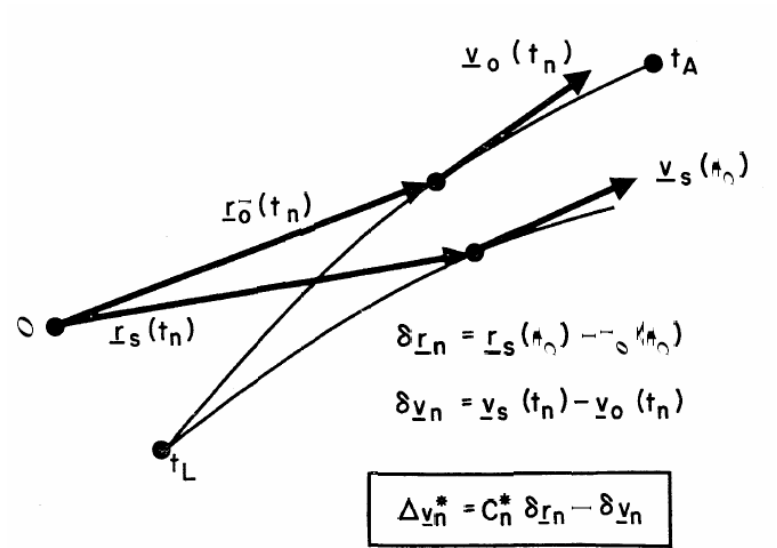


Figure 1.2: Early Apollo Guidance Strategy [2]

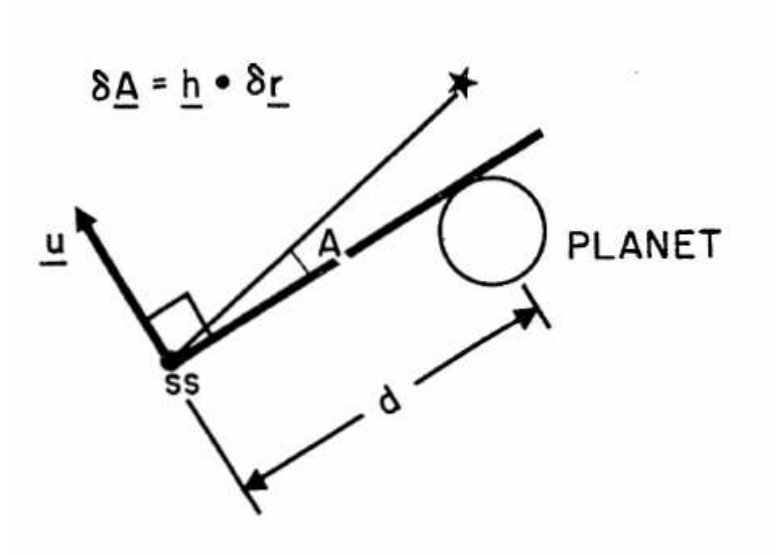


Figure 1.3: Angle Measurement For Determining Guidance Error [2]

any of the implicit or explicit guidance methods previously mentioned. Of course, it is likely that it also requires greater computational overhead. Applied to the TEI sequence, the single maneuver targeting employed by the Apollo algorithm leads to increased propulsive costs. Although, the theory is still suitable for TCM's. As opposed to the Apollo guidance algorithm, the TLC process is capable of targeting multiple maneuvers and entry constraints simultaneously. Howell and Pernicka are the initial founders of the TLC process [6, 7], and Marchand, Howell, and Wilson later expanded the capability of the algorithm through the addition of constraint targeting [8, 9].

1.3 Document Organization

This document is composed of eight chapters. The discussion begins with the basic theoretical foundation and evolves into a detailed discussion of the results.

In Chapter 2, the basic framework for the problem is introduced. Coordinate frames are defined, the state representation of choice is summarized, and the equations of motion are derived. The dynamics are governed by the Earth-Moon-Sun gravity field, and the relative n-body equations of motion are used in the development of the mathematical model. A linearized approximation is subsequently employed to identify the variational equation for the system.

In Chapter 3, The TLC process is discussed in detail. The method for ensuring position continuity is first described. Subsequently, the discussion evolves and addresses the process that is used to achieve velocity continuity and provide for constraint targeting. A sample set of target constraints is presented. This includes a the velocity continuity constraints, entry constraints, maneuver constraints, intermediate patch state constraints, and others.

In Chapter 4, several examples involving the application of the TLC algorithm are presented. First, a methodology for generating a suitable initial guess generation is summarized. Following this, the detailed series of results illustrates the effectiveness of the TLC approach.

In Chapter 5, the TLC theory is adapted to incorporate finite burns. Historically, the TLC process relies on impulsive maneuvers to control the characteristics of the targeting. The addition of finite burns to the TLC process is a new and exciting development resulting from this investigation and more accurately represents large maneuvers that span over a non-insignificant amount of time. An example is then presented to supplement the theory.

In Chapter 6, numerical optimization techniques are applied to the Earth-return as a basis for comparison against the targeting results. The theory of basic parameter optimization and numerical techniques is presented. The methods discussed are then applied to identify the optimal three-burn transfers considered in Chapter 4. These results serve to validate the TLC process and compare the resulting suboptimal targeting solutions to the optimal transfer paths.

In Chapter 7, a comparison between the optimal and the targeted solutions is made. This provides for a true measure of the optimality of the targeting results achieved in Chapter 4. Close attention is paid to the differences in these results, and ultimately a greater understanding of the TLC application is achieved.

In Chapter 8, conclusions and future recommendations for study are presented. Potential improvements are suggested, and analysis of the current work is performed.

Chapter 2

Background Information

This chapter develops the background information necessary for constructing the TLC process. In doing so, the nonlinear equations of motion are first presented, followed by a linearized approximation to these dynamics. This linearization, summarized in this investigation by the variational equation, provides the basis for the TLC algorithm.

2.1 Coordinate Frames

In this investigation, the directional axes of the working frame are consistent with the Earth-Mean Equator of J2000 (EME2000) frame [10]. In this frame, the x-axis is defined in the direction of the vernal equinox (cross product of the Earth's spin axis and the Earth's angular momentum about the sun) at noon on January 1, 2000, the z-axis is defined along the Earth's spin axis, and the y-axis completes the triad. The origin of the coordinate system is defined by either Earth Centered Inertial (ECI) or Moon Centered Inertial (MCI). Transformations between the ECI and MCI frames simply consists of an origin translation.

To formally define these transformations, let \odot denote the Sun, \oplus the Earth, and \lrcorner the Moon. The position vector from the Earth to the Moon is defined as $\bar{R}_{\oplus \rightarrow \lrcorner}(t)$, where the first subscript denotes the origin and the last subscript denotes the target body. This quantity is extracted from the DE405 ephemeris, and is subsequently employed in all transformations between ECI and MCI coordinates, as shown in Equations 2.1 and 2.2. A visual representation of each of these frames is depicted in Figures 2.1 and 2.2

$$\bar{R}_{\oplus \rightarrow s/c}(t) = \bar{R}_{\oplus \rightarrow \lrcorner}(t) + \bar{R}_{\lrcorner \rightarrow s/c}(t), \quad (2.1)$$

$$\bar{R}_{\lrcorner \rightarrow s/c}(t) = -\bar{R}_{\oplus \rightarrow \lrcorner}(t) + \bar{R}_{\oplus \rightarrow s/c}(t). \quad (2.2)$$

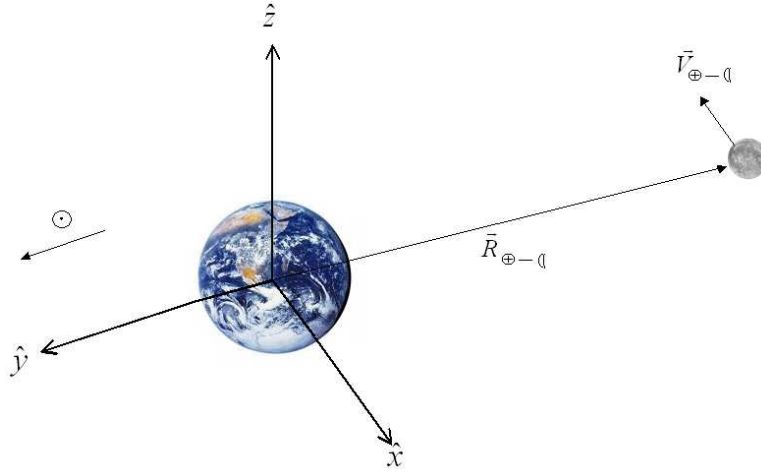


Figure 2.1: ECI frame

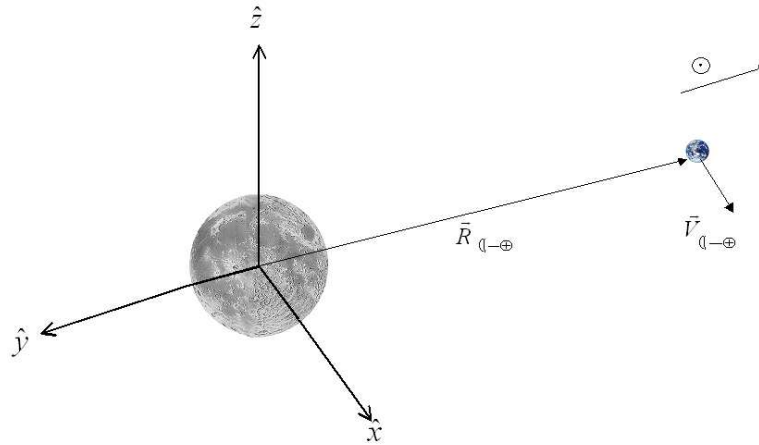


Figure 2.2: MCI frame

2.2 Dynamical Model

Over the course of an interplanetary mission, a spacecraft is exposed to various perturbing forces (i.e. imparts an acceleration), including: solar radiation pressure, drag, central body oblateness effects, third body oblateness effects, gravity, and magnetic radiation. The extent to which these forces affect the motion of the spacecraft depends on the nature of the mission. The governing dynamics are dictated by the spacecraft's environment, which in this case is the Earth-Moon system. Due to the short transfer times and sole presence within the Earth-Moon gravity field, gravitational effects are orders of magnitude larger than any other over the course of the Earth-return transfer leg (terminal point of the algorithm is entry interface). However, within the Earth-Moon gravity field, the gravitational effects due to the Sun are largely present, and therefore are considered along with the Earth and the Moon.

The equations of motion that govern the path of a spacecraft in a gravity

field of n-bodies are well-documented. A sample formulation is presented in Bate, Mueller, and White [11]. The goal is to express the equations of motion relative to the center of the Moon or the Earth. To that end, consider the inverse square law,

$$F_g = -\frac{GMm}{R^2}. \quad (2.3)$$

Equation 2.3 contains the scalar form of Newton's law of gravity, however this expression may be extended to capture the force in a particular direction,

$$\bar{F}_g = -\frac{GMm}{R^2} \frac{\bar{R}}{R}. \quad (2.4)$$

In the presence of n gravitational perturbations are present, the net force acting on the spacecraft is

$$\bar{F}_g = Gm_i \sum_{\substack{j=1 \\ j \neq i}}^n -\frac{m_j}{R_{ji}^2} \bar{R}_{ji}. \quad (2.5)$$

Subsequently, the spacecraft acceleration due to the net force in Equation 2.5 is

$$\ddot{\bar{R}} = -G \sum_{\substack{j=1 \\ j \neq i}}^n \frac{m_j}{R_{ji}^2} \bar{R}_{ji}. \quad (2.6)$$

This equation captures $\ddot{\bar{R}}$ in an inertial frame, and \bar{R} is a vector that originates at an inertial point, such as the solar system barycenter. In this study, it is most convenient to represent the equations of motion relative to a moving reference point, such as the Earth or the Moon. For example, consider two arbitrary gravitating bodies (labeled as 1 and 2). The inertial accelerations of each body in an n-body

gravitational field are determined as

$$\ddot{\bar{R}}_1 = -G \sum_{j=2}^n \frac{m_j}{R_{j1}^2} \bar{R}_{j1}, \quad (2.7)$$

$$\ddot{\bar{R}}_2 = -G \sum_{\substack{j=1 \\ j \neq 2}}^n \frac{m_j}{R_{j2}^2} \bar{R}_{j2}. \quad (2.8)$$

Thus, the relative acceleration of body 2 with respect to body 1 is given by

$$\ddot{\bar{R}}_{12} = \ddot{\bar{R}}_2 - \ddot{\bar{R}}_1. \quad (2.9)$$

Substituting Equations 2.7 and 2.8 into Equation 2.9 yields the inertial acceleration of body 2 with respect to body 1,

$$\ddot{\bar{R}}_{12} = G \sum_{\substack{j=1 \\ j \neq 2}}^n -\frac{m_j}{R_{j2}^2} \bar{R}_{j2} - G \sum_{j=2}^n -\frac{m_j}{R_{j1}^2} \bar{R}_{j1}. \quad (2.10)$$

Equation 2.10 is further simplified by extracting the relative accelerations of bodies 1 and 2 from the summation,

$$\ddot{\bar{R}}_{12} = \left(-\frac{Gm_1}{R_{12}^3} \bar{R}_{12} - G \sum_{\substack{j=1 \\ j \neq 2}}^n \frac{m_j}{R_{j2}^2} \bar{R}_{j2} \right) - \left(-\frac{Gm_2}{R_{21}^3} \bar{R}_{21} - G \sum_{j=2}^n \frac{m_j}{R_{j1}^2} \bar{R}_{j1} \right). \quad (2.11)$$

Furthermore, note that $\bar{R}_{12} = -\bar{R}_{21}$. Thus, the classical form of the relative n-body equations of motion is determined as,

$$\ddot{\bar{R}}_{12} = -\frac{G(m_1 + m_2)}{R_{12}^3} \bar{R}_{12} + \sum_{j=3}^n -Gm_j \left(\frac{\bar{R}_{j2}}{R_{j2}^3} - \frac{\bar{R}_{j1}}{R_{j1}^3} \right). \quad (2.12)$$

Equation 2.12 suggests that the relative equations of motion associated with the

ECI and MCI frames are given by Equations 2.13 and 2.14, respectively.

$$\ddot{\bar{R}}_{\oplus \rightarrow s/c} = -\frac{G(m_{\oplus})}{R_{\oplus \rightarrow s/c}^3} \bar{R}_{\oplus \rightarrow s/c} - Gm_{\odot} \left(\frac{\bar{R}_{\odot \rightarrow s/c}}{R_{\odot \rightarrow s/c}^3} - \frac{\bar{R}_{\odot \rightarrow \oplus}}{R_{\odot \rightarrow \oplus}^3} \right) - Gm_{\zeta} \left(\frac{\bar{R}_{\zeta \rightarrow s/c}}{R_{\zeta \rightarrow s/c}^3} - \frac{\bar{R}_{\zeta \rightarrow \odot}}{R_{\zeta \rightarrow \odot}^3} \right) \quad (2.13)$$

$$\ddot{\bar{R}}_{\zeta \rightarrow s/c} = -\frac{G(m_{\zeta})}{R_{\zeta \rightarrow s/c}^3} \bar{R}_{\zeta \rightarrow s/c} - Gm_{\odot} \left(\frac{\bar{R}_{\odot \rightarrow s/c}}{R_{\odot \rightarrow s/c}^3} - \frac{\bar{R}_{\odot \rightarrow \zeta}}{R_{\odot \rightarrow \zeta}^3} \right) - Gm_{\oplus} \left(\frac{\bar{R}_{\oplus \rightarrow s/c}}{R_{\oplus \rightarrow s/c}^3} - \frac{\bar{R}_{\oplus \rightarrow \odot}}{R_{\oplus \rightarrow \odot}^3} \right) \quad (2.14)$$

In Equations 2.13 and 2.14, the mass of the spacecraft is assumed to be negligible, and the positions of the celestial bodies are actively retrieved at each step of the integration from the DE405 Ephemeris. The DE405 ephemeris provides the position and velocity of all nine planets, and the Moon, in EME2000 coordinates, the working frame of choice in this study.

2.3 The Variational Equation

The underlying theory of the TLC targeting algorithm in this investigation relies on the linearized representation of the state equation,

$$\dot{\bar{X}} = F(\bar{X}). \quad (2.15)$$

In Equation 2.15, \bar{X} is a general state vector that defines the three-dimensional position and velocity information for the spacecraft,

$$\bar{X} = \begin{bmatrix} \bar{R} \\ \bar{V} \end{bmatrix}_{6 \times 1}. \quad (2.16)$$

The time derivative of Equation 2.16,

$$\dot{\bar{X}} = \begin{bmatrix} \bar{V} \\ \bar{a} \end{bmatrix}_{6 \times 1}, \quad (2.17)$$

provides three-dimensional velocity and acceleration information for the spacecraft.

A linearized approximation to Equation 2.15 is obtained from a Taylor series expansion [12] truncated to first-order. In particular, this expansion will be applied to the displacement of the function from some reference state, \bar{X}^*

$$F(\bar{X}) - F(\bar{X}^*) = \frac{\partial F}{\partial \bar{X}}|_* (\bar{X} - \bar{X}^*) + \frac{1}{2!} \frac{\partial^2 F}{\partial \bar{X}^2}|_* (\bar{X} - \bar{X}^*)^2 \dots + \text{Higher Order Terms} \quad (2.18)$$

For notational purposes, the differential of the state relative to the reference is represented by

$$d\bar{X} = \bar{X} - \bar{X}^*. \quad (2.19)$$

Substituting Equation 2.19 into Equation 2.18 yields

$$d\dot{\bar{X}} = \frac{\partial F}{\partial \bar{X}}|_* d\bar{X} + \frac{1}{2!} \frac{\partial^2 F}{\partial \bar{X}^2} d\bar{X}^2 \dots + \text{HOT}'s. \quad (2.20)$$

Note that in Equation 2.20 the differential of the state, $d\bar{X}$, does not cancel the denominator of the partial derivative. The denominator assumes an infinitesimal change in \bar{X} where the differential is small but finite [12].

From the expression in Equation 2.20, a linear approximation for the total deviation of \bar{X} from \bar{X}^* is achieved by truncating the Taylor series to include only the first-order term,

$$d\dot{\bar{X}} = \frac{\partial F}{\partial \bar{X}}|_* d\bar{X}. \quad (2.21)$$

In Equation 2.21, $\frac{\partial F}{\partial \bar{X}}$ is the Jacobian of the system, and represented by $A(t)$ [13]. Since the Earth-Moon dynamics defined in this study are formulated in the inertial frame, the Jacobian is expressed by the form,

$$\frac{\partial \dot{\bar{X}}}{\partial \bar{X}} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ \frac{\partial \ddot{\bar{R}}}{\partial \bar{R}} & 0_{3 \times 3} \end{bmatrix}. \quad (2.22)$$

In Equation 2.22, the lower-left submatrix is computed directly from Equation 2.13 or 2.14 depending on the desired origin. Equations 2.23 shows the partial derivative for the ECI frame, and Equation 2.24 summarizes the partial derivative for the MCI frame.

$$\frac{\partial \ddot{\bar{R}}}{\partial \bar{R}} = Gm_{\odot} \left(\frac{3\bar{R}_{s/c \rightarrow \odot} \bar{R}_{s/c \rightarrow \odot}^T}{R_{s/c \rightarrow \odot}^5} - \frac{1}{R_{\oplus \rightarrow \odot}^3} I_{3 \times 3} \right) + Gm_{\zeta} \left(\frac{3\bar{R}_{s/c \rightarrow \zeta} \bar{R}_{s/c \rightarrow \zeta}^T}{R_{s/c \rightarrow \zeta}^5} - \frac{1}{R_{\oplus \rightarrow \zeta}^3} I_{3 \times 3} \right) \quad (2.23)$$

and,

$$\frac{\partial \ddot{\bar{R}}}{\partial \bar{R}} = Gm_{\odot} \left(\frac{3\bar{R}_{s/c \rightarrow \odot} \bar{R}_{s/c \rightarrow \odot}^T}{R_{s/c \rightarrow \odot}^5} - \frac{1}{R_{\zeta \rightarrow \odot}^3} I_{3 \times 3} \right) + Gm_{\oplus} \left(\frac{3\bar{R}_{s/c \rightarrow \oplus} \bar{R}_{s/c \rightarrow \oplus}^T}{R_{s/c \rightarrow \oplus}^5} - \frac{1}{R_{\zeta \rightarrow \oplus}^3} I_{3 \times 3} \right) \quad (2.24)$$

The solution to the first-order differential equation in Equation 2.21 is well known and depends on the state transition matrix

$$d\bar{X}_t = \tilde{\Phi}_{t,t_0} d\bar{X}_{t_0}. \quad (2.25)$$

Also, the calculation of the state transition matrix is governed by

$$\dot{\tilde{\Phi}} = \tilde{A} \tilde{\Phi}_{t,t_0}. \quad (2.26)$$

Initially at $t=t_0$, $\tilde{\Phi}_{t_0,t_0} = \tilde{I}$, where \tilde{I} is the $N \times N$ identity matrix, and N represents the length of \bar{X} .

Of course, the differential of the spacecraft state in Equation 2.25, $d\bar{X}$, is associated with a specific fixed value of t . To allow additional flexibility, it is advantageous to consider Equation 2.25 in the context of a non-contemporaneous variation. Non-contemporaneous analysis accommodates the time varying nature of the linearized dynamics in this investigation. Figure 2.3 illustrates a visual representation of contemporaneous vs. non-contemporaneous and thus details the difference

between a differential and a variation. This discrepancy between differentials and variations is outlined in many texts, with one such example given by Weinstock [14] and another by Hull [3]. The non-contemporaneous form of Equation 2.23 is written as

$$(\delta\bar{X}(t) + \bar{X}(t)dt) = \tilde{\Phi} \left(\delta\bar{X}(t_0) + \dot{\bar{X}}(t_0)dt \right). \quad (2.27)$$

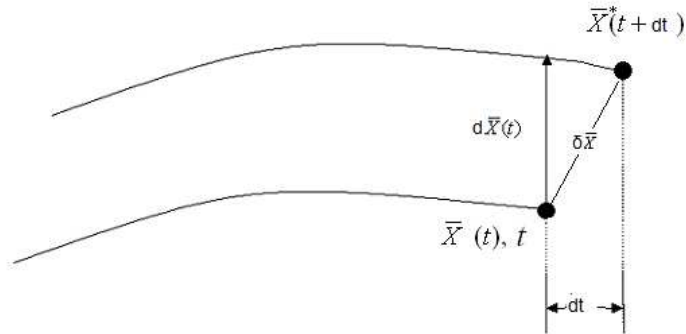


Figure 2.3: Differential vs. Variation [3]

Equation 2.27 is referred to as the variational equation. This variational equation is the basis of the targeting algorithms described in this investigation.

Chapter 3

Impulsive Trajectory Targeting

The two-level corrector is the basis for this targeting investigation. This procedure is an iterative one. The first of the two iterative procedures that constitute the process satisfies position continuity by making small adjustments to the velocity. This occurs across all specified nodes, or patch points, along a provided reference path. The second procedure uses the displacement of positions and times of these patch points to satisfy the selected set of imposed constraints (including velocity continuity). For every Level-Two correction that takes place, a series of Level-One corrections is performed sequentially across all patch states. As a numerical technique, these may be considered as the major and minor iterations of the routine.

The TLC algorithm [9], is developed in a generalized framework. That is, it is applicable to any regime that follows the form in Equation 2.15. The focus of this chapter is the development and description of the ideas that encompass the TLC process. This discussion begins with the basic concepts of the Level-One process and evolves into a detailed description of the constraint targeting with the Level-Two process.

3.1 Level-One Correction using Impulsive Maneuvers

A Level-One correction is analogous to Lambert's problem. Lambert's problem entails finding the transfer arc in a two-body regime that connects a given initial and final position vector for a fixed time of flight. Using the two-body approximation, a simple iterative technique permits a solution to this problem. However, in the Earth-Moon-Sun gravity field, two-body approximations are inappropriate. Therefore, rather than a Lambert solver, an iterative method that employs the variational equation is chosen. To that end, consider Equation 2.27 from time t_0 to t_f ,

$$\begin{bmatrix} \delta \bar{R}_f^- - \bar{V}_f^- \delta t_f \\ \delta \bar{V}_f^- - \bar{a}_f^- \delta t_f \end{bmatrix} = \begin{bmatrix} \tilde{A}_{t_f, t_0} & \tilde{B}_{t_f, t_0} \\ \tilde{C}_{t_f, t_0} & \tilde{D}_{t_f, t_0} \end{bmatrix} \begin{bmatrix} \delta \bar{R}_0^+ - \bar{V}_0^+ \delta t_0 \\ \delta \bar{V}_0^+ - \bar{a}_0^+ \delta t_0 \end{bmatrix}. \quad (3.1)$$

Note that the '+' and '-' superscripts denote the state just after and just before the patch point. These states are considered to be infinitely close to the patch point. Also, the submatrices \tilde{A}_{t_f, t_0} , \tilde{B}_{t_f, t_0} , \tilde{C}_{t_f, t_0} , \tilde{D}_{t_f, t_0} correspond to the components of the state transition matrix $\frac{\partial \bar{R}_{t_0}}{\partial \bar{R}_{t_f}}$, $\frac{\partial \bar{R}_{t_0}}{\partial \bar{V}_{t_f}}$, $\frac{\partial \bar{V}_{t_0}}{\partial \bar{R}_{t_f}}$, $\frac{\partial \bar{V}_{t_0}}{\partial \bar{V}_{t_f}}$. In a fixed time of flight targeting problem: t_0 , \bar{R}_0 , and t_f are fixed. Thus, $\delta \bar{R}_0 = \delta t_f = \delta t_0 = 0$ and the first vector expression of Equation 3.1 reduces to

$$\delta \bar{R}_f^- = \tilde{B}_{t_f, t_0} \delta \bar{V}_0^+. \quad (3.2)$$

In a linear sense, the velocity correction required to meet the target is

$$\delta \bar{V}_0^+ = \tilde{B}_{t_f, t_0}^{-1} \delta \bar{R}_f^-. \quad (3.3)$$

In determining the inverse in Equation 3.3, it is important to note that $\tilde{B}_{t_f, t_0}^{-1} \neq \tilde{B}_{t_0, t_f}$. This identity only holds for the entire state transition matrix, not its submatrices.

Equation 3.3 represents the fundamental component of a standard differential corrections procedure [11] used to determine the velocity necessary at the departure point to reach the terminal position. A Level-One correction is then the iterative application of the linearized approximation in Equation 3.3 to the nonlinear problem. The solution is successfully identified once the final position is achieved within some specified tolerance.

As an example, consider initial state $\bar{X}_0 = \begin{bmatrix} 16378.145 & 0 & 0 & 0 & 3.694 & 0 \end{bmatrix}$ and terminal target position $\bar{R} = \begin{bmatrix} -10378.22 & 0 & 0 \end{bmatrix}$, with respect to the ECI frame. Eight iterations of the Level-One correction achieves the targeted position within the tolerance of 10^{-4} km. The required $|\Delta\bar{V}|$ is 1329 m/s. The result is illustrated in Figure 3.1. This plot shows, however, that the desired terminal position is nearly achieved after only 2 iterations of the Level-One correction procedure.

3.1.1 Time Dependent Level-One Correction

The traditional Level-One algorithm assumes the final time is a fixed parameter. However, this need not necessarily be imposed. If t_f were not fixed, the first vector expression of Equation 3.1 is given as

$$\delta\bar{R}_f^- = \tilde{B}_{t_f, t_0} \delta\bar{V}_0^+ + \bar{V}_f^- \delta t_f. \quad (3.4)$$

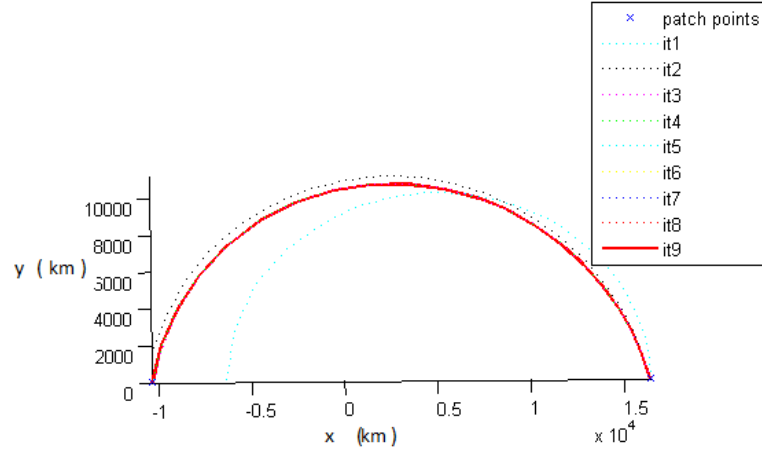


Figure 3.1: Lambert Targeting Example

Equation 3.4 is written in vector form as

$$\delta \bar{R}_f = \begin{bmatrix} \tilde{B}_{t_f, t_0} \bar{V}_f^- \end{bmatrix} \begin{bmatrix} \delta \bar{V}_0^+ \\ \delta t_{t_f} \end{bmatrix}. \quad (3.5)$$

Equation 3.5 represents an under-determined system. That is, $\delta \bar{R}_0^+$ and δt_{t_f} represent four control parameters for three targets. Therefore, there are more unknown variables than equations and less dependent variables than independent variables. Given this, an infinite number of solutions exist. A convenient choice for the solution of Equation 3.5 is the minimum norm solution. The minimum norm method [15] identifies a solution to

$$\bar{e} = \tilde{M} \bar{b}, \quad (3.6)$$

that minimizes \bar{b} ,

$$\bar{b} = \tilde{M}^T \left(\tilde{M} \tilde{M}^T \right)^{-1} \bar{e}. \quad (3.7)$$

In the case of a Level-One correction, as that in Equation 3.4, applying a minimum norm solution identifies the smallest possible $\delta \bar{V}_0^+$ and δt_f that achieve the desired $\delta \bar{R}_f = 0$. This is shown mathematically as

$$\begin{bmatrix} \delta \bar{V}_0^+ \\ \delta t_f \end{bmatrix} = \tilde{M}^T \left(\tilde{M} \tilde{M}^T \right)^{-1} \delta \bar{R}_f. \quad (3.8)$$

In Equation 3.8, $\tilde{M} = \begin{bmatrix} \tilde{B}_{t_0, t_f} & \bar{V}_f^- \end{bmatrix}$, $\bar{e} = \begin{bmatrix} \delta \bar{V}_0^+ & \delta t_f \end{bmatrix}^T$, and $\bar{b} = \delta \bar{R}_f$.

3.2 Level-Two Correction

The Level-One targeting algorithm is a very useful tool and has many applications. However, the method is ineffective when the number of targets exceeds the number of control variables. This problem is addressed by the Level-Two process [6]. While the Level-One process adjusts the velocity at the patch states to satisfy position continuity, the Level-Two correction applies small adjustments in the positions and times of the patch states to achieve velocity continuity.

A simple example of three consecutive patch points is employed to aid in this discussion. Consider applying the Level-One correction process to this three patch point example where: k-1 denotes the initial point, k represents the interior point, and k+1 gives terminal point. Applying the Level-One corrections process to this example results in a set of consecutive arcs such as those illustrated in Figure 3.2(a). As a consequence of the nature of the Level-One process, a non-zero maneuver exists at patch points k-1 and k. The Level-Two process is a linear

iterative process that is capable of searching the neighborhood of Figure 3.2(a) for a solution that is continuous in velocity. Such a solution is represented in Figure 3.2(b). Mathematically, the goal is to achieve

$$\Delta \bar{V}_k = 0 = \bar{V}_k^+ - \bar{V}_k^- . \quad (3.9)$$

Given the expression in Equation 3.9, it is easily deduced that

$$\bar{V}_k^+ = \bar{V}_k^- . \quad (3.10)$$

Subsequently, it follows that the variation in the velocity on either side of k must

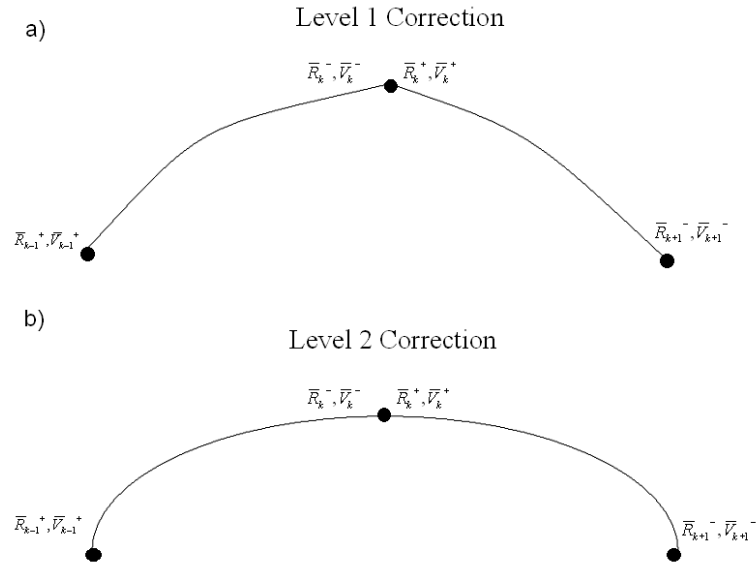


Figure 3.2: (a) Level 1 correction (b) Level 2 correction applied over three patch points

follow the same relationship,

$$\delta\bar{V}_k^+ = \delta\bar{V}_k^-. \quad (3.11)$$

This implies that,

$$\delta\Delta\bar{V}_k = 0 = \delta\bar{V}_k^+ - \delta\bar{V}_k^-. \quad (3.12)$$

Equation 3.12 is equivalent to Equation 3.9. To enforce the velocity continuity constraint, it is necessary for Equation 3.12 be written in terms of the desired controls. For the three patch state case in Figure 3.2, these include: $\delta\bar{R}_{k-1}, \delta t_{k-1}, \delta\bar{R}_k, \delta t_k, \delta\bar{R}_{k+1}$, and δt_{k+1} . The linear approximation in Equation 2.27 is used to achieve this relation [8, 9]. Applying this variational equation to the second arc in the three patch point example gives

$$\begin{bmatrix} \delta\bar{R}_k - \bar{V}_k^+ \delta t_k \\ \delta\bar{V}_k^+ - \bar{a}_k^+ \delta t_k \end{bmatrix} = \begin{bmatrix} \tilde{A}_{k,k+1} & \tilde{B}_{k,k+1} \\ \tilde{C}_{k,k+1} & \tilde{D}_{k,k+1} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_{k+1} - \bar{V}_{k+1}^- \delta t_{k+1} \\ \delta\bar{V}_{k+1}^- - \bar{a}_{k+1}^- \delta t_{k+1} \end{bmatrix}. \quad (3.13)$$

Note that the ‘+’ and ‘-’ superscripts do not apply to the position terms, as the position continuity here is assumed to have already been achieved by the Level-One process. As an intermediate step in defining $\delta\bar{V}_k^+$ in terms of the control variables, the vector $\delta\bar{V}_{k+1}^-$ is determined from the first vector expression of Equation 3.13.

$$\begin{aligned} \delta\bar{V}_{k+1}^- &= \tilde{B}_{k,k+1}^{-1} \delta\bar{R}_k - \tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \delta\bar{R}_{k+1} - \tilde{B}_{k,k+1}^{-1} \bar{V}_k^+ \delta t_k + \\ &\quad \left(\tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \bar{V}_{k+1}^- + \bar{a}_{k+1}^- \right) \delta t_{k+1} \end{aligned} \quad (3.14)$$

Subsequently, $\delta\bar{V}_k^+$ is determined from substituting Equation 3.14 into the second

vector expression of Equation 3.13.

$$\begin{aligned}
\delta\bar{V}_k^+ = & \left(\tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1} \right) \delta\bar{R}_k + \left(\bar{a}_k^+ - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\bar{V}_k^+ \right) \delta t_k + \\
& \left(\tilde{C}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1} \right) \delta\bar{R}_{k+1} + \\
& \left(-\tilde{C}_{k,k+1}\bar{V}_{k+1}^- + \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1}\bar{V}_{k+1}^- \right) \delta t_{k+1}. \quad (3.15)
\end{aligned}$$

Equation 3.15 represents an important result, as the variation in $\delta\bar{V}_k^+$ is expressed in terms of the variations in the positions and times of the current and subsequent patch point. Similarly, employing the variational equation from k-1 to k yields an expression for $\delta\bar{V}_k^-$,

$$\begin{aligned}
\delta\bar{V}_k^- = & \left(\tilde{C}_{k,k-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1} \right) \delta\bar{R}_{k-1} + \\
& \left(-\tilde{C}_{k,k-1}\bar{V}_{k-1}^+ + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1}\bar{V}_{k-1}^+ \right) \delta t_{k-1} + \\
& \left(\tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1} \right) \delta\bar{R}_k + \left(\bar{a}_k^- - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\bar{V}_k^- \right) \delta t_k. \quad (3.16)
\end{aligned}$$

Finally, the substitution of Equation 3.15 and Equation 3.16 into Equation 3.12 leads to an expression for the variation in impulsive $\Delta\bar{V}$ at point k in terms of the desired controls,

$$\delta\Delta\bar{V}_k = \begin{bmatrix} \tilde{M}_{R_{k-1}} & \tilde{M}_{t_{k-1}} & \tilde{M}_{R_k} & \tilde{M}_{t_k} & \tilde{M}_{R_{k+1}} & \tilde{M}_{t_{k+1}} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_{k-1} \\ \delta t_{k-1} \\ \delta R_k \\ \delta t_k \\ \delta\bar{R}_{k+1} \\ \delta t_{k+1} \end{bmatrix}. \quad (3.17)$$

The submatrices in Equation 3.17 are,

$$\begin{aligned}
\tilde{M}_{R_{k-1}} &= \left(-\tilde{C}_{k,k-1} + \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{A}_{k,k-1} \right), \\
\tilde{M}_{t_{k-1}} &= \left(\tilde{C}_{k,k-1} \bar{V}_{k-1}^+ - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{A}_{k,k-1} \bar{V}_{k-1}^+ \right), \\
\tilde{M}_{R_k} &= \left(\tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \right), \\
\tilde{M}_{t_k} &= \left(\bar{a}_k^+ - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \bar{V}_k^+ - \bar{a}_k^- + \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \bar{V}_k^- \right), \\
\tilde{M}_{R_{k+1}} &= \left(\tilde{C}_{k,k+1} - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \right), \\
\tilde{M}_{t_{k+1}} &= \left(-\tilde{C}_{k,k+1} \bar{V}_{k+1}^- + \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \bar{V}_{k+1}^- \right)
\end{aligned} \tag{3.18}$$

Equation 3.17 contains more controls than constraints and is thus an under-determined system with an infinite number of solutions. Using the minimum norm solution given by Equation 3.7 provides the smallest changes in the positions and times of the three patch points that lead to a zero $\Delta \bar{V}$ at point k.

To gain insight into the application of Equation 3.17, consider the trajectory illustrated in Figure 3.3. The blue crosses represent the position of a sample set of Earth centered patch states, where the initial and final points are identical to those in Figure 3.1. The initial and final states are fixed, but all interior states are allowed to vary to satisfy velocity continuity over the interior patch points. The expression

in Equation 3.19 represents the application of Equation 3.17 to this example.

$$\begin{bmatrix} \Delta \bar{V}_2 \\ \Delta \bar{V}_3 \\ \Delta \bar{V}_4 \\ \Delta \bar{V}_5 \end{bmatrix}_{12 \times 3} = \begin{bmatrix} \tilde{M}_{R_1} & \tilde{M}_{t_1} & \tilde{M}_{R_2} & \tilde{M}_{t_2} & \tilde{M}_{R_3} & \tilde{M}_{t_3} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} \\ \bar{0} & \bar{0} & \tilde{M}_{R_2} & \tilde{M}_{t_2} & \tilde{M}_{R_3} & \tilde{M}_{t_3} & \tilde{M}_{R_4} & \tilde{M}_{t_4} & \bar{0} & \bar{0} & \bar{0} & \bar{0} \\ \bar{0} & \bar{0} & \bar{0} & \bar{0} & \tilde{M}_{R_3} & \tilde{M}_{t_3} & \tilde{M}_{R_4} & \tilde{M}_{t_4} & \tilde{M}_{R_5} & \tilde{M}_{t_5} & \bar{0} & \bar{0} \\ \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \tilde{M}_{R_4} & \tilde{M}_{t_4} & \tilde{M}_{R_5} & \tilde{M}_{t_5} & \tilde{M}_{R_6} & \tilde{M}_{t_6} \\ \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \bar{0} & \tilde{M}_{R_5} & \tilde{M}_{t_5} & \tilde{M}_{R_6} & \tilde{M}_{t_6} \end{bmatrix}_{12 \times 24} \begin{bmatrix} \bar{R}_1 \\ t_1 \\ \bar{R}_2 \\ t_2 \\ \bar{R}_3 \\ t_3 \\ \bar{R}_4 \\ t_4 \\ \bar{R}_5 \\ t_5 \\ \bar{R}_6 \\ t_6 \end{bmatrix}_{24 \times 3} \quad (3.19)$$

Note, even though $\delta \bar{R}_1, \delta t_1$ and $\delta \bar{R}_6, \delta t_6$ are present on the right hand side of the equation, their values are set to zero to satisfy the fixed end point constraints. Direct application of the TLC process indicated by Equation 3.19 achieves the solution in Figure 3.3 in 4 iterations. After these 4 iterations, the magnitude of the ΔV at the initial patch point is 192 m/s and all interior maneuvers are successfully driven to zero. The blue crosses and dotted line represent the initial patch states and the trajectory, respectfully. Clearly, this startup arc represents a poor initial guess due to the large velocity discontinuities across interior portion of the trajectory. Still, the TLC process is able to identify a solution that is continuous in velocity in a matter of a few iterations.

3.3 Constraints

Thus far, the formulation of the TLC scheme presented only describes how to achieve a velocity continuity constraint. However, the TLC process is applicable to any constraint that can be expressed in terms of the control variables, in this case the variations in the positions and times of the patch points. Constraints may be imposed

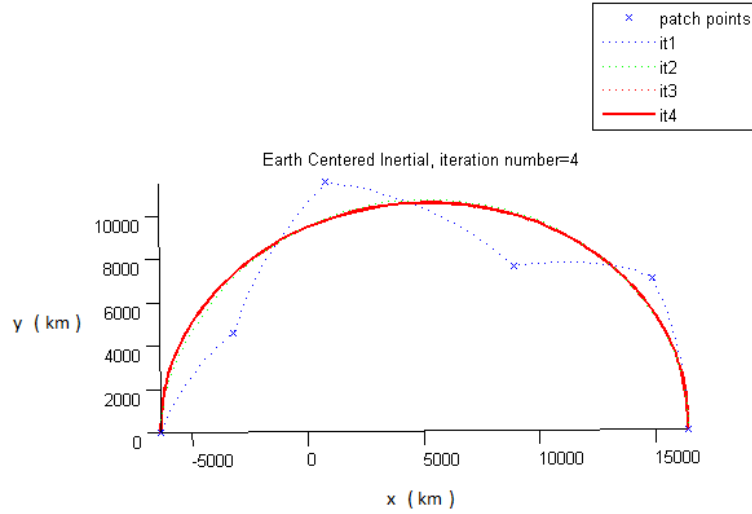


Figure 3.3: Velocity continuity enforced on a simple example

at the end state (final patch point), the initial state, or any interior point along the transfer. In this investigation, a constraint at the k^{th} patch point is represented by the notation, α_k .

By definition, a constraint imposed at the k^{th} patch point has an explicit dependence on the position and time at that patch point. However, the constraint does not have an explicit dependence on any of the remaining patch states. On the other hand, it is possible for the constraints to be a function of \bar{V}_k^+ or \bar{V}_k^- , which implies an implicit dependence (displacement of position and times affects

the velocity) on the neighboring patch states through the following,

$$\begin{aligned}
\delta\alpha_k &= \left(\frac{\partial\alpha_k}{\partial\bar{V}_k^-} \frac{\partial\bar{V}_k^-}{\partial\bar{R}_{k-1}} \right) \delta\bar{R}_{k-1} + \left(\frac{\partial\alpha_k}{\partial\bar{V}_k^-} \frac{\partial\bar{V}_k^-}{\partial t_{k-1}} \right) \delta t_{k-1} + \\
&\quad \left(\frac{\partial\alpha_k}{\partial\bar{R}_k} + \frac{\partial\alpha_k}{\partial\bar{V}_k^+} \frac{\partial\bar{V}_k^+}{\partial t_k} + \frac{\partial\alpha_k}{\partial\bar{V}_k^-} \frac{\partial\bar{V}_k^-}{\partial\bar{R}_k} \right) \delta\bar{R}_k + \\
&\quad \left(\frac{\partial\alpha_k}{\partial t_k} + \frac{\partial\alpha_k}{\partial\bar{V}_k^+} \frac{\partial\bar{V}_k^+}{\partial t_k} + \frac{\partial\alpha_k}{\partial\bar{V}_k^-} \frac{\partial\bar{V}_k^-}{\partial t_k} \right) \delta t_k + \\
&\quad \left(\frac{\partial\alpha_k}{\partial\bar{V}_k^+} \frac{\partial\bar{V}_k^+}{\partial\bar{R}_{k+1}} \right) \delta\bar{R}_{k+1} + \left(\frac{\partial\alpha_k}{\partial\bar{V}_k^+} \frac{\partial\bar{V}_k^+}{\partial t_{k+1}} \right) \delta t_{k+1} \tag{3.20}
\end{aligned}$$

Equation 3.20 demonstrates the operation of the chain rule [12] in obtaining the desired implicit dependence on the control variables. The partial derivatives of \bar{V}_k^+ and \bar{V}_k^- in this expression are obtained from 3.15 and 3.16,

$$\frac{\partial\bar{V}_k^-}{\partial\bar{R}_{k-1}} = \left(C_{k,k-1} - D_{k,k-1} B_{k,k-1}^{-1} A_{k,k-1} \right), \tag{3.21}$$

$$\frac{\partial\bar{V}_k^-}{\partial t_{k-1}} = \left(-C_{k,k-1} \bar{V}_{k-1}^+ + D_{k,k-1} B_{k,k-1}^{-1} A_{k,k-1} \bar{V}_{k-1}^+ \right) \delta t_{k-1}, \tag{3.22}$$

$$\frac{\partial\bar{V}_k^-}{\partial\bar{R}_k} = \left(D_{k,k-1} B_{k,k-1}^{-1} \right), \tag{3.23}$$

$$\frac{\partial\bar{V}_k^-}{\partial t_k} = \left(\bar{a}_k^- - D_{k,k-1} B_{k,k-1}^{-1} \bar{V}_k^- \right), \tag{3.24}$$

$$\frac{\partial\bar{V}_k^+}{\partial\bar{R}_k} = \left(D_{k,k+1} B_{k,k+1}^{-1} \right), \tag{3.25}$$

$$\frac{\partial\bar{V}_k^+}{\partial t_k} = \left(\bar{a}_k^+ - D_{k,k+1} B_{k,k+1}^{-1} \bar{V}_k^+ \right), \tag{3.26}$$

$$\frac{\partial\bar{V}_k^+}{\partial\bar{R}_{k+1}} = \left(C_{k,k+1} - D_{k,k+1} B_{k,k+1}^{-1} A_{k,k+1} \right), \tag{3.27}$$

$$\frac{\partial\bar{V}_k^+}{\partial t_{k+1}} = \left(-C_{k,k+1} \bar{V}_{k+1}^- + D_{k,k+1} B_{k,k+1}^{-1} A_{k,k+1} \bar{V}_{k+1}^- \right) \delta t_{k+1}. \tag{3.28}$$

With the definition of the partials in Equations 3.21 through 3.28, the only partial derivatives in Equation 3.20 that remain undetermined for a given constraint are

$\frac{\partial \alpha_k}{\partial R_k}, \frac{\partial \alpha_k}{\partial t_k}, \frac{\partial \alpha_k}{\partial V_k^-}, \frac{\partial \alpha_k}{\partial V_k^+}$. To implement a constraint into the TLC algorithm, the rows of \bar{e} and \tilde{M} in Equation 3.17 are augmented to include the error and the partial derivatives associated with the constraint. The augmented system is still of the form,

$$\bar{e} = \tilde{M}\bar{b}. \quad (3.29)$$

However, expanding this augmented system yields

$$\begin{bmatrix} \delta \Delta \bar{V}_k \\ \delta \alpha_k \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta \bar{V}_k}{\partial b} \\ \frac{\partial \alpha_k}{\partial b} \end{bmatrix} \begin{bmatrix} \delta \bar{R}_k \\ \delta t_k \end{bmatrix} \quad (3.30)$$

The second row in Equation 3.30 represents the vector expression in Equation 3.20. A minimum norm solution of Equation 3.30 identifies the smallest changes in position and time that meet the desired constraints.

Sections 3.3.1 through 3.3.12 summarize a sample set of constraints that are applicable to the present study. In each of these sections, the constraint is stated and the associated partial derivatives are determined. The set of constraints presented includes results previously identified by Marchand, Howell, and Wilson [8, 9], as well as some new formulations relevant to the Orion TEI.

3.3.1 Position Vector

The position vector constraint at the k^{th} patch point [8],

$$\alpha_{pos} = \bar{R}_k - \bar{R}_{k-specifid}, \quad (3.31)$$

is applicable at any point along a trajectory, including the end point or any other patch state. This constraint, along with others position related constraints like:

altitude, latitude, and longitude can all be targeted via the Level-One process. However, it is useful to incorporate these as Level-Two constraints in the event that additional constraints are imposed. The only non-zero partial derivative of Equation 3.31 with respect to the control variables is

$$\frac{\partial \alpha_{pos}}{\partial \bar{R}_k} = \tilde{I}_{3 \times 3}. \quad (3.32)$$

An example of position constraint targeting is shown in Figure 3.4. The position

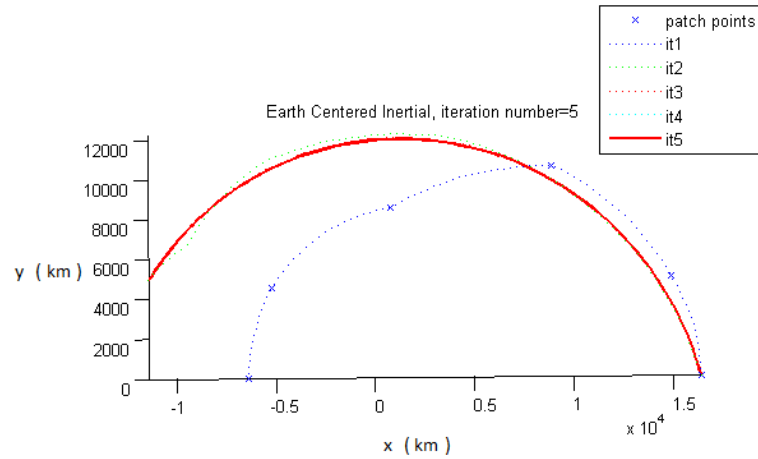


Figure 3.4: Targeting Position

constraint is applied at the final patch point. The numerical values used for the initial and targeted values are

$$\begin{aligned} \text{Initial position} &: \begin{bmatrix} -6378.145 & 0.0 & 0 \end{bmatrix} \text{ km}, \\ \text{Targeted position} &: \begin{bmatrix} -11378.145 & 5000.0 & 0 \end{bmatrix} \text{ km}. \end{aligned} \quad (3.33)$$

The TLC process is able to converge on the neighborhood of the desired solution within two iterations. Five iterations are required to achieve the pre-specified convergence tolerance.

3.3.2 Velocity Vector

The velocity vector constraint [8], much like the position vector constraint, is applicable at any point along the trajectory. In addition, the velocity either before ('-') or after ('+') the k^{th} patch point may be targeted by

$$\alpha_{v^-} = \bar{V}_k^- - \bar{V}_{k-specifield}, \quad (3.34)$$

$$\alpha_{v^+} = \bar{V}_k^+ - \bar{V}_{k-specifield}. \quad (3.35)$$

The non-zero partial derivatives associated with Equations 3.34 and 3.35 are listed below,

$$\frac{\partial \alpha_{v^-}}{\partial \bar{V}_k^-} = \tilde{I}_{3 \times 3}, \quad (3.36)$$

$$\frac{\partial \alpha_{v^+}}{\partial \bar{V}_k^+} = \tilde{I}_{3 \times 3}. \quad (3.37)$$

3.3.3 Altitude

The altitude constraint [8] is more specifically a constraint imposed on the radial distance from the origin. This is merely an alteration of Equation 3.28, where at the k^{th} patch point,

$$\alpha_{alt} = |\bar{R}_k| - R_{specifield}. \quad (3.38)$$

The partial derivative required for targeting Equation 3.38 is

$$\frac{\partial \alpha_{alt}}{\partial \bar{R}_k} = \frac{1}{2} (\bar{R}_k^T \bar{R}_k)^{-\frac{1}{2}} 2\bar{R}_k^T \tilde{I}_{3 \times 3}. \quad (3.39)$$

Equation 3.39 is simplified to

$$\frac{\partial \alpha_{alt}}{\partial \bar{R}_k} = \frac{\bar{R}_k}{|\bar{R}_k|}. \quad (3.40)$$

Figure 3.5 shows the implementation of an altitude constraint on the terminal patch state. The initial and targeted altitude values are

$$\begin{aligned} \text{Initial altitude} &: & 0 \text{ km}, \\ \text{Targeted altitude} &: & 3000 \text{ km}. \end{aligned} \quad (3.41)$$

The altitude is less restrictive than the position vector constraint in Equation 3.32. Thus, the TLC is able to identify a solution in the neighborhood of the initial guess that satisfies the specified target.

3.3.4 Earth Entry Latitude

Mathematically speaking, the geocentric latitude at the k^{th} patch point [8],

$$\alpha_{lat} = \frac{\bar{R}_{entry} \cdot \hat{z}}{|\bar{R}_{entry}|} - \sin(\phi)_{specified}, \quad (3.42)$$

is simply the inverse sine of the projection of the position vector onto the Earth's z-axis. To avoid quadrant ambiguity, it is useful to target the sine of the latitude rather than the latitude itself. The partial derivative associated with Equation 3.42 is,

$$\frac{\partial \alpha_{lat}}{\partial \bar{R}_k} = \frac{\hat{z}}{|\bar{R}_k|} - \frac{\bar{R}_k}{|\bar{R}_k|} \sin(\phi). \quad (3.43)$$

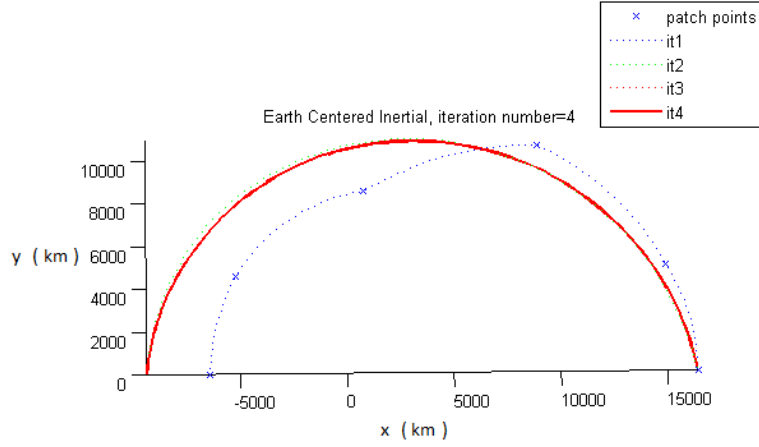


Figure 3.5: Targeting Altitude

3.3.5 Earth Entry Longitude

The calculation of longitude is more involved than the constraints outlined previously. Because it is defined within the Earth's body-fixed rotating frame, rather than an inertial frame, the calculation of the longitude includes both the elapsed time and the rotation of the Earth. The equation for the longitude [16] at the k^{th} patch point is

$$\theta_k = \tan^{-1} \frac{\bar{R}_k \cdot \hat{y}}{\bar{R}_k \cdot \hat{x}} - \theta_{g0} - \omega_e (t_k - t_0). \quad (3.44)$$

However, to avoid quadrant ambiguity, it is advantageous to consider the cosine of the longitude rather than the longitude itself,

$$\cos(\theta_k) = \cos \left(\tan^{-1} \left(\frac{\bar{R}_k \cdot \hat{y}}{\bar{R}_k \cdot \hat{x}} \right) - \theta_{g0} - \omega_e (t_k - t_0) \right). \quad (3.45)$$

For notational purposes, the right ascension term in Equation 3.45 is expressed as the angle Θ , shown by

$$\Theta_k = \tan^{-1}\left(\frac{\bar{R}_k \cdot \hat{y}}{\bar{R}_k \cdot \hat{x}}\right). \quad (3.46)$$

A formulation for targeting the right ascension itself is included in the previous investigation by Marchand, Howell, and Wilson [9]. Using a cosine trigonometric identity, Equation 3.45 is expressed in the more convenient form,

$$\begin{aligned} \alpha_{lon} = & \cos(\Theta_k)\cos(-\theta_{g0} - \omega_e(t_k - t_0)) - \\ & \sin(\Theta_k)\sin(-\theta_{g0} - \omega_e(t_k - t_0)) - \cos(\theta)_{specified}. \end{aligned} \quad (3.47)$$

Equation 3.47 contains the Greenwich hour angle, θ_{g0} , in its computation. An approximation for θ_{g0} is summarized by Vallado [16]. The computation itself depends on knowledge of the universal time (UT), in fractions of days from midnight on the current day. The expressions in Equation 3.48 are sufficient for the calculation of the Greenwich hour angle, however, additional effort is required to ensure that θ_{g0} is defined between 0 and 2π .

$$\begin{aligned} jd0 &= jd - UT, \\ UT_1 &= \frac{jd0 - 2451545}{36525}, \\ \theta_{g0} &= 100.4606184 + (36000.77004)UT_1 + (0.000387933)UT_1^2 - \\ & \quad (2.583 \times 10^{-8})UT_1^3. \end{aligned} \quad (3.48)$$

Also note that in Equation 3.47, Δt is found using

$$(t_k - t_0) = (86400)UT. \quad (3.49)$$

To evaluate the partial derivatives of Equation 3.47, it is useful to recognize the geometric properties of the right ascension shown in Figure 3.6. From this image,

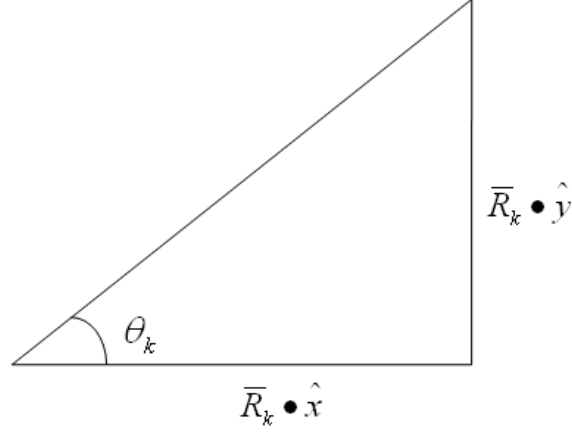


Figure 3.6: Right Ascension

the expressions for the cosine and sine of the right ascension in Equation 3.47 are obtained as:

$$\cos(\Theta_k) = \frac{\bar{R}_k \cdot \hat{x}}{\sqrt{(\bar{R}_k \cdot \hat{x})^2 + (\bar{R}_k \cdot \hat{y})^2}}, \quad (3.50)$$

$$\sin(\Theta_k) = \frac{\bar{R}_k \cdot \hat{y}}{\sqrt{(\bar{R}_k \cdot \hat{x})^2 + (\bar{R}_k \cdot \hat{y})^2}}. \quad (3.51)$$

The partial derivatives of the cosine and sine of the right ascension in Equations 3.50 and 3.51 are:

$$\frac{\partial \cos(\Theta_k)}{\partial \bar{R}_k} = \frac{\hat{x}^T \left(\sqrt{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2} \right) - \frac{(\hat{x}^T (\bar{R}_k^T \hat{x}) + \hat{y}^T (\bar{R}_k^T \hat{y})) \bar{R}_k^T \hat{x}}{\sqrt{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2}}}{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2}, \quad (3.52)$$

$$\frac{\partial \sin(\Theta_k)}{\partial \bar{R}_k} = - \frac{\hat{y}^T \left(\sqrt{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2} \right) - \frac{(\hat{x}^T (\bar{R}_k^T \hat{x}) + \hat{y}^T (\bar{R}_k^T \hat{y})) \bar{R}_k^T \hat{y}}{\sqrt{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2}}}{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2}. \quad (3.53)$$

Finally, the expressions for the non-zero partial derivatives of Equation 3.47 with respect to the control variables are given by

$$\begin{aligned}\frac{\partial \alpha_{lon}}{\partial \bar{R}_k} &= \frac{\partial \cos(\theta_k)}{\partial \bar{R}_k} \cos(-\theta_{g0} - \omega_e(t_k - t_0)) - \\ &\quad \frac{\partial \sin(\theta_k)}{\partial \bar{R}_k} \sin(-\theta_{g0} - \omega_e(t_k - t_0)),\end{aligned}\quad (3.54)$$

$$\begin{aligned}\frac{\partial \alpha_{lon}}{\partial t_k} &= \omega_e \frac{\bar{R}_k^T \hat{x}}{\sqrt{\bar{R}_k^T \hat{x}^2 + \bar{R}_k^T \hat{y}^2}} \sin(-\theta_{g0} - \omega_e \Delta t) + \\ &\quad \omega_e \frac{\bar{R}_k^T \hat{y}}{\sqrt{\bar{R}_k^T \hat{x}^2 + \bar{R}_k^T \hat{y}^2}} \cos(-\theta_{g0} - \omega_e \Delta t).\end{aligned}\quad (3.55)$$

Alternate forms of the longitude constraint in Equation 3.47 are also available. One such alternative involves targeting the sine of the longitude,

$$\begin{aligned}\alpha_{lon_b} &= \sin(\Theta_k) \cos(-\theta_{g0} - \omega_e(t_k - t_0)) + \\ &\quad \cos(\Theta_k) \sin(-\theta_{g0} - \omega_e(t_k - t_0)) - \sin(\theta)_{specified},\end{aligned}\quad (3.56)$$

and another simply targets the longitude value directly as

$$\alpha_{lon_c} = \theta_k - \theta_{specified}.\quad (3.57)$$

The partial derivatives associated with Equations 3.56 and 3.57 are

$$\begin{aligned}\frac{\partial \alpha_{lon_b}}{\partial \bar{R}_k} &= \frac{\partial \sin(\theta_k)}{\partial \bar{R}_k} \cos(-\theta_{g0} - \omega_e(t_k - t_0)) + \\ &\quad \frac{\partial \cos(\theta_k)}{\partial \bar{R}_k} \sin(-\theta_{g0} - \omega_e(t_k - t_0)),\end{aligned}\quad (3.58)$$

$$\begin{aligned}\frac{\partial \alpha_{lon_b}}{\partial t_k} &= \omega_e \frac{\bar{R}_k^T \hat{y}}{\sqrt{\bar{R}_k^T \hat{x}^2 + \bar{R}_k^T \hat{y}^2}} \sin(-\theta_{g_0} - \omega_e \Delta t) - \\ &\quad \omega_e \frac{\bar{R}_k^T \hat{x}}{\sqrt{\bar{R}_k^T \hat{x}^2 - \bar{R}_k^T \hat{y}^2}} \cos(-\theta_{g_0} - \omega_e \Delta t),\end{aligned}\quad (3.59)$$

$$\frac{\partial \alpha_{lon_c}}{\partial \bar{R}_k} = \frac{\hat{y}^T (\bar{R}_k^T \hat{x}) - \hat{x}^T (\bar{R}_k^T \hat{y})}{(\bar{R}_k^T \hat{x})^2 + (\bar{R}_k^T \hat{y})^2}, \quad (3.60)$$

$$\frac{\partial \alpha_{lon_c}}{\partial t_k} = -\omega_e. \quad (3.61)$$

Each formulation presents unique advantages and disadvantages. While targeting trigonometric functions allows distinct advantages in avoiding quadrant ambiguities, they present their own set of disadvantages as well. For example, the sine function double values between 0 and π , and the cosine function double values between $-\pi/2$ and $\pi/2$. Therefore, the best choice among these formulations depends on the initial and targeted angles themselves and having a variety of options becomes advantageous.

3.3.6 Entry Flight Path Angle

The entry flight path angle [9], γ , dictates the steepness of the incoming velocity vector at Earth entry. A visual representation of this angle is captured in Figure 3.7. Equation 3.62 shows the expression for the flight path angle at patch point k,

$$\alpha_{fpa} = \frac{\bar{R}^T \bar{V}_k^-}{R V_k^-} - \sin(\gamma)_{specified}. \quad (3.62)$$

The non-zero partial derivatives associated with Equation 3.62 are:

$$\frac{\partial \alpha_{fpa}}{\partial \bar{R}_k} = \frac{(\bar{V}_k^-)^T}{|\bar{R}_k| |\bar{V}_k^-|} - \sin(\gamma) \frac{\bar{R}_k^T}{|\bar{R}_k|^2}, \quad (3.63)$$

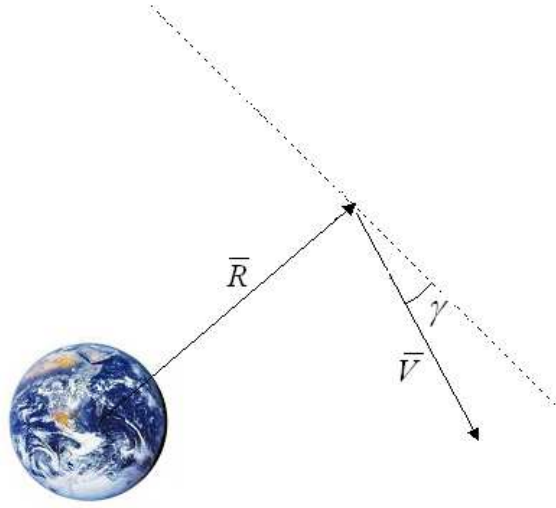


Figure 3.7: Flight Path Angle

$$\frac{\partial \alpha_{fpa}}{\partial \bar{V}_k^-} = \frac{(\bar{R}_k)^T}{|\bar{R}_k| |\bar{V}_k^-|} - \sin(\gamma) \frac{\bar{V}_k^{-T}}{|\bar{V}_k^-|^2}. \quad (3.64)$$

An example of the entry flight path angle constraint implemented into the standard set of example patch states is seen in Figure 3.8. The initial and final values of the flight path angle for this example are,

$$\begin{aligned} \text{Initial flight path angle} &: & 0 \text{ degrees} \\ \text{Targeted flight path angle} &: & -6 \text{ degrees.} \end{aligned} \quad (3.65)$$

Note that since the flight patch angle, γ , is reduced by six degrees, the angle of the incoming velocity vector is rotated toward the center of the Earth to achieve the solution.

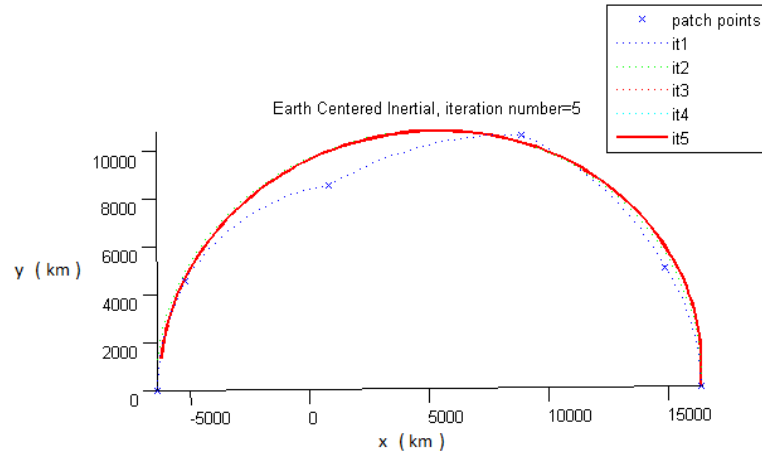


Figure 3.8: Targeting the Flight Path Angle

3.3.7 Earth Entry Flight Path Azimuth

The Flight Path azimuth is defined in a local spacecraft, North-East-Down frame.

This frame is defined by the following orthogonal unit vectors:

$$\begin{aligned}
 \hat{r} &= \frac{\bar{R}_k}{R_k}, \\
 \hat{e} &= \frac{\hat{z} \times \hat{r}}{|\hat{z} \times \hat{r}|}, \\
 \hat{n} &= \frac{\bar{R}_k \times (\hat{z} \times \hat{r})}{|\bar{R}_k \times (\hat{z} \times \hat{r})|}.
 \end{aligned} \tag{3.66}$$

The expression for the constraint itself is given by

$$\alpha_{az} = \frac{\bar{V}_k^- \cdot \hat{n}}{(\bar{V}_k^- \cdot \hat{n})^2 + (\bar{V}_k^- \cdot \hat{e})^2} - \cos(Az)_{specified}. \tag{3.67}$$

All non-zero partial derivatives associated with the flight path azimuth constraint in Equation 3.67 are captured by the following:

$$\frac{\partial \alpha_{az}}{\partial R_k} = \frac{(\bar{V}_k^-)^T \frac{\partial \hat{n}}{\partial R_k} \left(\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \right) - \frac{((\bar{V}_k^-)^T \hat{e}) \left((\bar{V}_k^-)^T \frac{\partial \hat{e}}{\partial R_k} \right) + ((\bar{V}_k^-)^T \frac{\partial \hat{n}}{\partial R_k}) ((\bar{V}_k^-)^T \hat{n})}{\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2}} ((\bar{V}_k^-)^T \hat{n})}{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \quad (3.68)$$

$$\frac{\partial \alpha_{az}}{\partial \bar{V}_k^-} = \frac{\hat{n}^T \left(\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \right) - \frac{((\bar{V}_k^-)^T \hat{e}) \hat{e}^T + ((\bar{V}_k^-)^T \hat{n}) \hat{n}^T}{\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2}} (\bar{V}_k^-)^T \hat{n}}{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2}. \quad (3.69)$$

Equations 3.68 and 3.69 contain partial derivatives of the coordinate frames with respect to the control variables. The first of these partials is expressed by

$$\frac{\partial \hat{e}}{\partial \bar{R}_k} = \left(\frac{\tilde{X}_z}{\bar{R}_k} - \frac{(\tilde{X}_z \bar{R}_k) (\tilde{X}_z \bar{R}_k)^T \tilde{X}_z}{[(\tilde{X}_z \bar{R}_k)^T (\tilde{X}_z \bar{R}_k)]^{\frac{3}{2}}} \right). \quad (3.70)$$

In Equation 3.70, \tilde{X}_z represents a cross-product in matrix multiplication form,

$$\tilde{X}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.71)$$

where for example,

$$\hat{z} \times \bar{R}_k = \tilde{X}_z \bar{R}_k. \quad (3.72)$$

Before the second of the required coordinate frame partial derivatives is given, it is useful to identify a vector expression for a triple cross-product,

$$\bar{R} \times (\hat{z} \times \hat{r}) = \hat{z} (\bar{R}_k^T \bar{R}_k) - \bar{R}_k (\bar{R}_k^T \hat{z}). \quad (3.73)$$

Given the simplification in Equation 3.73, the partial derivative of \hat{n} with respect to the position at the k^{th} patch point is

$$\frac{\partial \hat{n}}{\partial \bar{R}_k} = \frac{\partial}{\partial \bar{R}_k} \left(\frac{\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z})}{[(\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z}))^T (\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z}))]^{\frac{1}{2}}} \right). \quad (3.74)$$

Equation 3.74 reduces to

$$\begin{aligned} \frac{\partial \hat{n}}{\partial \bar{R}_k} &= \left(2\hat{z}\bar{R}_k^T - (\bar{R}_k^T \hat{z})\tilde{I}_{3 \times 3} - (\bar{R}_k \hat{z}^T) \right) (\chi)^{-\frac{1}{2}} - (\chi)^{-\frac{3}{2}} \\ &\quad (\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z}))^T \left(2\hat{z}\bar{R}_k^T - (\bar{R}_k^T \hat{z})\tilde{I}_{3 \times 3} - (\bar{R}_k \hat{z}^T) \right) \\ &\quad (\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z})), \end{aligned} \quad (3.75)$$

where,

$$\chi = [(\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z}))^T (\hat{z}(\bar{R}_k^T \bar{R}_k) - \bar{R}_k(\bar{R}_k^T \hat{z}))]. \quad (3.76)$$

An example of cosine azimuth targeting is provided by Figure 3.9, where the initial and final angles are specified as

$$\begin{aligned} \text{Initial Azimuth} &: \quad 90 \text{ degrees,} \\ \text{Targeted Azimuth} &: \quad 0 \text{ degrees.} \end{aligned} \quad (3.77)$$

Notice in this figure that in changing the azimuth by 90 degrees, the entire transfer rotates to satisfy the constraint.

Like the latitude, longitude, and flight path angle constraints, alternate targeting formulations exist. In choosing which to use, the numerical properties of each trigonometric function are considered. Depending on the imposed target values, one formulation may perform more efficiently than another. One alternative approach

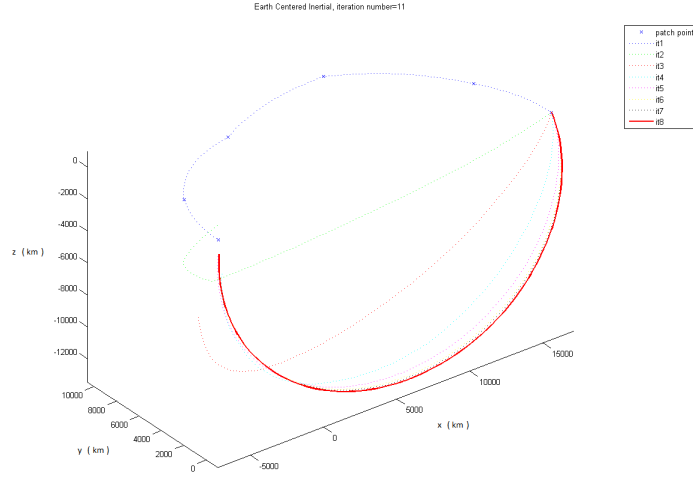


Figure 3.9: Targeting the Flight Path Azimuth

to targeting the flight path azimuth involves the sine, rather than the cosine,

$$\alpha_{az_b} = \frac{\bar{V}_k^- \cdot \hat{e}}{(\bar{V}_k^- \cdot \hat{n})^2 + (\bar{V}_k^- \cdot \hat{e})^2} - \sin(Az)_{specified}. \quad (3.78)$$

Another alternate formulation involves targeting the tangent of the flight path azimuth,

$$\alpha_{az_c} = \frac{\bar{V}_k^- \cdot \hat{e}}{\bar{V}_k^- \cdot \hat{n}} - \tan(Az)_{specified}. \quad (3.79)$$

All non-zero partial derivatives associated Equations 3.78 and 3.79 are captured by the following:

$$\frac{\partial \alpha_{az_b}}{\partial \bar{R}_k} = \frac{(\bar{V}_k^-)^T \frac{\partial \hat{e}}{\partial \bar{R}_k} \left(\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \right) - \frac{((\bar{V}_k^-)^T \hat{e}) \left((\bar{V}_k^-)^T \frac{\partial \hat{e}}{\partial \bar{R}_k} \right) + ((\bar{V}_k^-)^T \frac{\partial \hat{n}}{\partial \bar{R}_k}) ((\bar{V}_k^-)^T \hat{n})}{\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{e})^2}}}{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \quad (3.80)$$

$$\frac{\partial \alpha_{az_b}}{\partial \bar{V}_k^-} = \frac{\hat{e}^T \left(\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2} \right) - \frac{((\bar{V}_k^-)^T \hat{n}) \hat{n}^T + ((\bar{V}_k^-)^T \hat{e}) \hat{e}^T}{\sqrt{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2}} (\bar{V}_k^-)^T \hat{e}}{((\bar{V}_k^-)^T \hat{e})^2 + ((\bar{V}_k^-)^T \hat{n})^2}, \quad (3.81)$$

$$\frac{\partial \alpha_{az_c}}{\partial \bar{R}_k} = \frac{\left((\bar{V}_k^-)^T \frac{\partial \hat{e}}{\partial \bar{R}_k} \right) ((\bar{V}_k^-)^T \hat{n}) - \left((\bar{V}_k^-)^T \frac{\partial \hat{n}}{\partial \bar{R}_k} \right) ((\bar{V}_k^-)^T \hat{e})}{((\bar{V}_k^-)^T \hat{n})^2}, \quad (3.82)$$

$$\frac{\partial \alpha_{az_c}}{\partial \bar{V}_k^-} = \frac{\hat{e}^T ((\bar{V}_k^-)^T \hat{n}) - \hat{n}^T ((\bar{V}_k^-)^T \hat{e})}{((\bar{V}_k^-)^T \hat{n})^2}. \quad (3.83)$$

3.3.8 Time

Time constraints may be placed at any patch state along the transfer. It can be beneficial to constrain the terminal entry time of the transfer, and equally important to freeze an intermediate patch state for applications discussed in Sections 3.3.11 and 3.3.12. The expression used for constraining the time [8] at the k^{th} patch point is given by

$$\alpha_{time} = t_k - t_{desired}. \quad (3.84)$$

The only non-zero partial derivative is

$$\frac{\partial \alpha_t}{\partial t_k} = 1. \quad (3.85)$$

3.3.9 Constraining Individual Maneuvers

During the design process, it may be beneficial, or even necessary, to allow a maneuver at some point along the transfer. That is, if more constraints exist than controls for a particular problem, either a maneuver must be allowed or additional patch states must be added to enable convergence. To allow a maneuver in the TLC process, the rows corresponding to the maneuver point's velocity continuity constraint are removed. However, by simply removing the velocity continuity constraint, it is possible for the magnitude of the allowed maneuver to become exceedingly large. If

it is important for this maneuver to remain below a certain threshold, a constraint may be placed on the magnitude of the maneuver of the k^{th} patch point by

$$\alpha_{delv} = \sqrt{\Delta \bar{V}_k^T \Delta \bar{V}_k} - \Delta \bar{V}_{desired}. \quad (3.86)$$

The non-zero partial derivatives of Equation 3.86 with respect to the controls are

$$\frac{\partial \alpha_{delv}}{\partial \bar{V}_k^+} = \frac{\bar{V}_k^{+T}}{|\bar{V}_k^+|}, \quad (3.87)$$

and

$$\frac{\partial \alpha_{delv}}{\partial \bar{V}_k^-} = -\frac{\bar{V}_k^{-T}}{|\bar{V}_k^-|}. \quad (3.88)$$

3.3.10 Constraining the Total $\Delta \bar{V}$

While the maneuver magnitude constraint in section 3.3.9 has many applications, the existence of several maneuvers benefits from the use of a different constraint. Rather than individually constraining the maneuvers, the entire sum can be constrained by

$$\alpha_{delv} = \sum_{k=1}^m (\sqrt{\Delta \bar{V}_k^T \Delta \bar{V}_k} - \Delta \bar{V}_{desired}). \quad (3.89)$$

The partial derivative of Equation 3.89 with respect to the controls are

$$\frac{\partial \alpha_{delv}}{\partial \bar{V}_k^+} = \sum_{k=1}^m \left(\frac{\bar{V}_k^{+T}}{|\bar{V}_k^+|} \right), \quad (3.90)$$

$$\frac{\partial \alpha_{delv}}{\partial \bar{V}_k^-} = -\sum_{k=1}^m \left(\frac{\bar{V}_k^{-T}}{|\bar{V}_k^-|} \right). \quad (3.91)$$

This constraint allows for the reduction of the ΔV total to be less confined. When the maneuver sum is incorporated for a multiple maneuver transfer, the TLC process

distributes the maneuver reductions in the most dynamically compliant fashion to meet the constraints. Of course, like Section 3.3.9, the maneuver sum constraint is only applicable to interior maneuvers.

3.3.11 Constraining the Departure Velocity

The TLC process requires a maneuver at the first patch state to achieve the position of the second patch point. However, it is not always desirable to allow this maneuver to exist, and it can, in fact, be removed by requiring that \bar{V}_1^+ match \bar{V}_1^- ,

$$\alpha_{v0} = \bar{V}_1^+ - \bar{V}_1^-. \quad (3.92)$$

Of course, for this constraint to successfully remove the initial maneuver, the position and time constraints at the first patch point must also be enforced. The partial derivatives for Equation 3.92 are obtained from Equations 3.26 through 3.28 since $\delta\bar{V}_1^- = 0$ by definition.

3.3.12 Constraining a Patch State

An alternative for dealing with the initial maneuver dilemma in Section 3.3.11 is to create an artificial patch point or set of patch points prior to the first patch state. In doing this, it becomes possible to constrain the first maneuver as an interior patch point. The points prior to this are then subject to change within the TLC process, but their final values are not of consequence. Consider the plot in Figure 3.10. The first patch state in this plot is an artificial one, and the second patch state remains constant in its location throughout the iteration process. In this particular example, the maneuver at the constrained patch state is held to a value of zero.

To implement the fixed patch state constraint, the position, time, and \bar{V}_k^-

are each enforced on the first non-artificial patch state. Equations 3.31, 3.84, and 3.34 are used to accomplish this task.

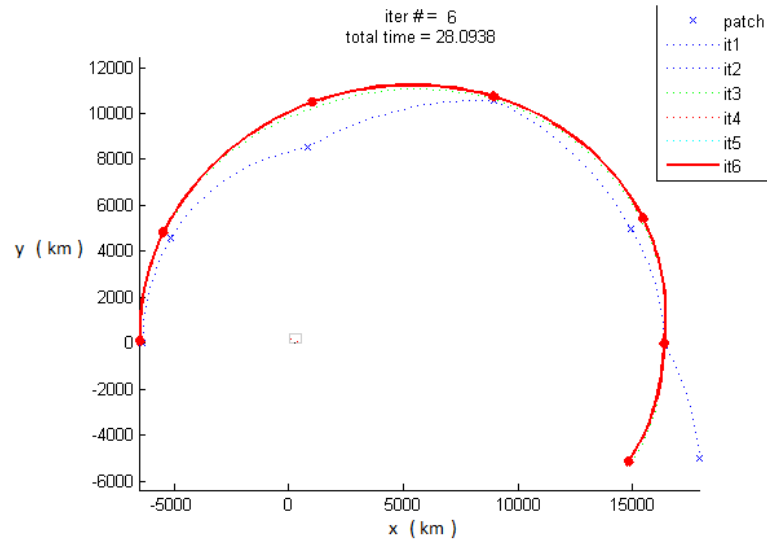


Figure 3.10: Targeting the Altitude While Holding the Second Patch Point Fixed

Chapter 4

Targeting Results

As previously mentioned in the first chapter, the goal of this investigation is to develop an algorithm that can: target Earth entry constraints, use less than 1.4 km/s of ΔV , target multiple maneuvers via a single execution of the routine, and converge to a solution with minimal computational time. This chapter shows that the TLC algorithm satisfies these requirements and succeeds in its application to the Earth-return targeting problem. Five example cases are presented to illustrate this.

The cases included in this chapter address the targeting problem represented by a complete end-to-end Earth-return. A full three-burn Earth-return scenario represents the most challenging application of the TLC algorithm. While this algorithm faces the challenge of being run at any point along the transfer, from TEI-1 to the final TCM, favorable convergence behavior in the end-to-end case provides assurance that the algorithm is adequate for any of these.

Consider, for example, the targeting of a midcourse correction maneuver. In this case, the first patch state in the TLC process is located somewhere between TEI-

3 and Earth entry, and targeting involves only one or two maneuvers. Furthermore, in the absence of a major catastrophe, the targeting errors are small and thus the required TCM's are on the order of 10's of m/s or smaller. Naturally, a targeting task of this caliber is much more trivial than the end-to-end undertaking.

The application of the TLC process to the example cases in this chapter is completed in two phases: generating the initial set of patch points and applying the TLC procedure to this initial guess. Section 4.1 describes the generation of the initial patch states for the Earth-return transfer.

4.1 Initial Guess

Since this algorithm relies on the minimum norm solution, the search for a solution remains in the realm of the initial reference trajectory. Clearly, this implies that the initial guess is an important part of the TLC process.

While creating a method for automatically generating the initial guess seems like a daunting task, simplification is attainable. That is, for this project, each application of this targeting algorithm involves one task: Earth-return targeting. In this application, the only variables that changes from one initial guess to another are the: departure date, arrival date, lunar orbit parameters, selection of active entry constraints, and particular entry constraint values. Applying simple two-body mechanics [11, 17, 18, 19] and Chapter 3's targeting methods, a straight-forward initial arc generation scheme is constructed.

4.1.1 Burn 1

The method for determining the first maneuver begins with the assumption of a circular lunar parking orbit. Therefore, only variations in altitude, inclination, and

ascending node are considered. If a non-circular initial orbit were required for the mission, this approach could easily be altered. However, without additional knowledge on the requirements for the initial orbit, a low-lunar circular orbit seems a reasonable choice.

The calculation of the maneuver itself begins by retrieving $\bar{R}_{\zeta \rightarrow \oplus}$ from the DE405 ephemeris at the targeted entry time. The departure point on the orbit is chosen as that which most nearly opposes the direction of this vector from the Moon to the Earth. This direction is given by

$$\bar{R}_{departure} = -\frac{\bar{R}_{\zeta \rightarrow \oplus}}{|\bar{R}_{\zeta \rightarrow \oplus}|}. \quad (4.1)$$

Once the departure point is chosen, the first maneuver is applied along the direction of the orbital velocity with a magnitude given by

$$\Delta V_1 = \sqrt{2 \left(-\frac{\mu}{R_{\tau_0} + R_{apo}} + \frac{\mu}{R_{\tau_0}} \right)} - \sqrt{\frac{\mu}{R_{\tau_0}}}. \quad (4.2)$$

The apoapsis radius in Equation 4.2 is given by,

$$R_{apo} = \left((period) \frac{\sqrt{\mu}}{2\pi} \right)^{\frac{2}{3}} - R_{\tau_0}. \quad (4.3)$$

This maneuver places the spacecraft on an intermediate transfer orbit. The period of the intermediate orbit is left as an input to the user, but is arbitrarily selected as 48 hours for generating the results in this investigation. An artist's rendition of this maneuver is seen in Figure 4.1.

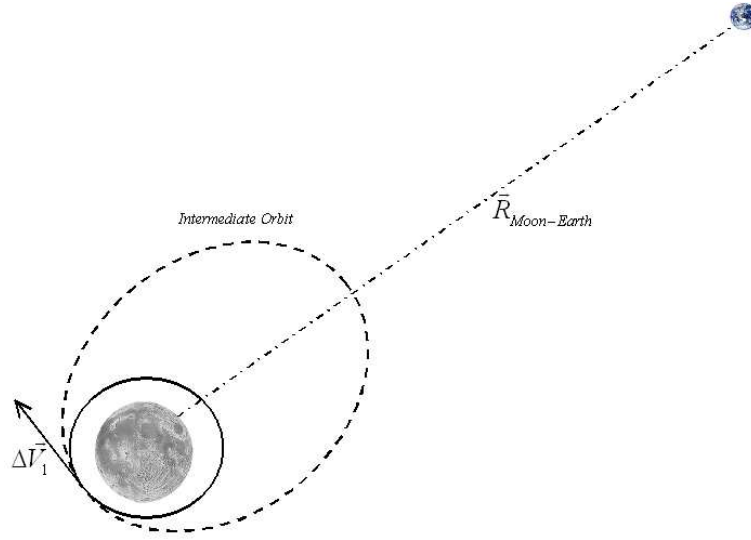


Figure 4.1: Rendition of Maneuver 1

4.1.2 Burn 2

The second maneuver occurs at the apoapsis of the intermediate transfer orbit. The position and velocity at this point are found by numerically integrating the state after burn 1 for half of the intermediate transfer orbit's period. This time is calculated by

$$period = t_1 = \frac{\pi}{\sqrt{\mu}} \frac{R_{\tau_0} + R_{apo}}{2}^{\frac{3}{2}}. \quad (4.4)$$

The maneuver applied at this state does not alter the magnitude of the velocity but rather rotates it. The vector rotation is performed such that the velocity vector at the apoapsis is aligned with $\vec{V}_{\oplus \rightarrow \zeta}(t_f)$ ($-\vec{V}_{\oplus \rightarrow \zeta}(t_f)$ if the initial lunar orbit is retrograde). The period of the intermediate orbit remains unchanged, because the line of apsides is perpendicular to the Earth's velocity. This rotation is performed

to reduce the plane change required by the third maneuver to target the Earth. An artist's rendition of this maneuver is seen in Figure 4.2.

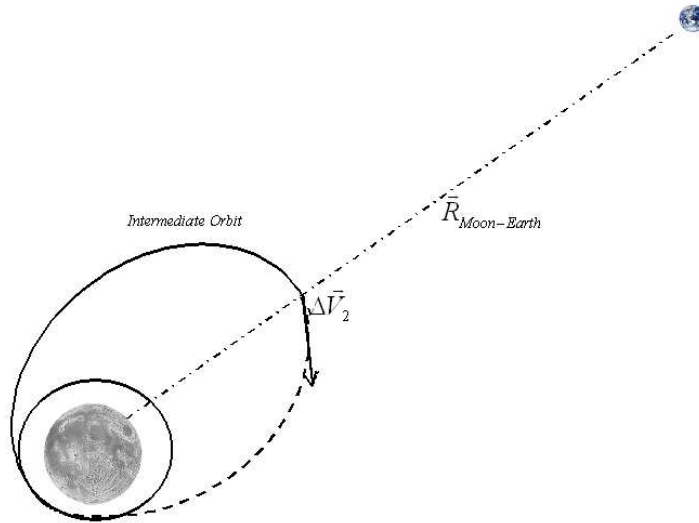


Figure 4.2: Rendition of Maneuver 2

4.1.3 Burn 3

The third maneuver is applied after propagating the burn 2 state forward until the periapsis is reached. For the first two maneuvers, two-body mechanics are assumed in estimating the magnitude and directions of the maneuvers. For the final maneuver, however, the targeting method in Chapter 3 is utilized.

First, a Level-One correction process is used to target the altitude, latitude, and longitude at the Earth. The velocity discontinuity created at the periapsis in this targeting process yields the third maneuver in the three-burn TEI guess generation. An artist's rendition of this maneuver is seen in Figure 4.3, and an actual image of

a three-burn transfer created with this generation process is shown in Figure 4.4.

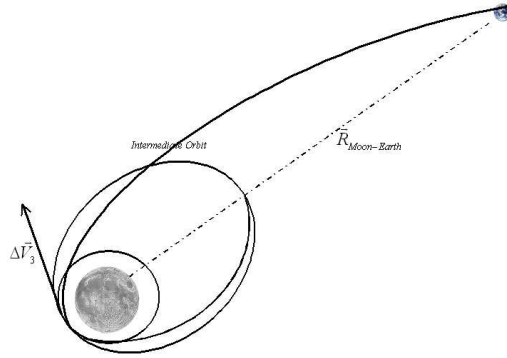


Figure 4.3: Rendition of Maneuver 3

Thus far, the initial arc generator described only targets the Earth entry position constraints. However, the velocity constraints are often important as well, and the Level-Two correction procedure is a viable method for satisfying these constraints.

To construct an initial guess that satisfies the position and velocity entry constraints, the Level-One process for calculating TEI-3 is replaced by a Level-Two. In doing so, the first step is to select several patch points along the arc generated by the Level-One procedure. Given these patch states, the Level-Two process described in Chapter 3 is applied in targeting the desired entry constraints.

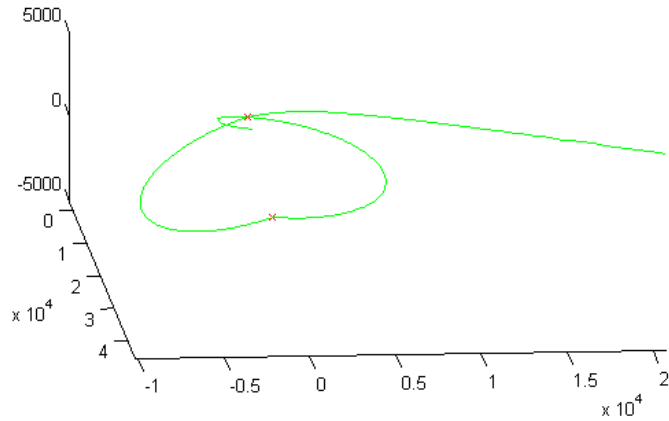


Figure 4.4: Example of Initial Guess Implementation at the Moon, MCI

4.1.4 Application of the Initial Guess Generator

The most useful capability of the initial guess generation tool described in Section 4.1 is its ability to satisfy the end constraints. This allows for the TLC's startup arc to be in the vicinity of the solution. Since the formulation of the TLC process relies on a minimum norm calculation, this proves invaluable in solving the selected examples in this study. However, care must be taken to recognize the limits of this initial guess arc formulation.

The primary limitation of this initial guess tool is identified at the first maneuver. That is, the desired direction defined by Equation 4.1 may not necessarily intersect the initial orbit. Consider the image in Figure 4.5. This departure point is not collinear with $\bar{R}_{\mathcal{C} \rightarrow \oplus}$. Because of this, a plane change is ultimately required at TEI-3 to target the Earth and maneuver costs are increased.

The disparity between the departure point and the Earth-Moon line can be measured via the Earth-Moon-Spacecraft (EMS) angle depicted in Figure 4.5. Across a lunar cycle, the EMS angle at the initial maneuver point of the TEI se-

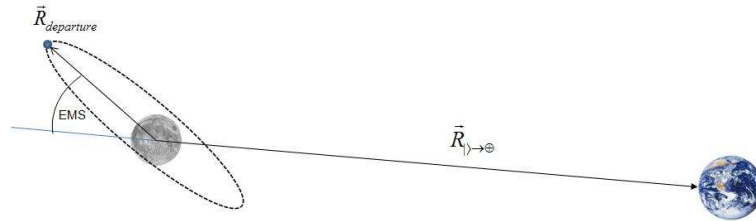


Figure 4.5: EMS Angle Vs. Days from jd=2457000.5

quence increases to a magnitude greater than 20 degrees. The evolution of this EMS value for the first maneuver is shown by the plot in Figure 4.6. Not surprisingly,

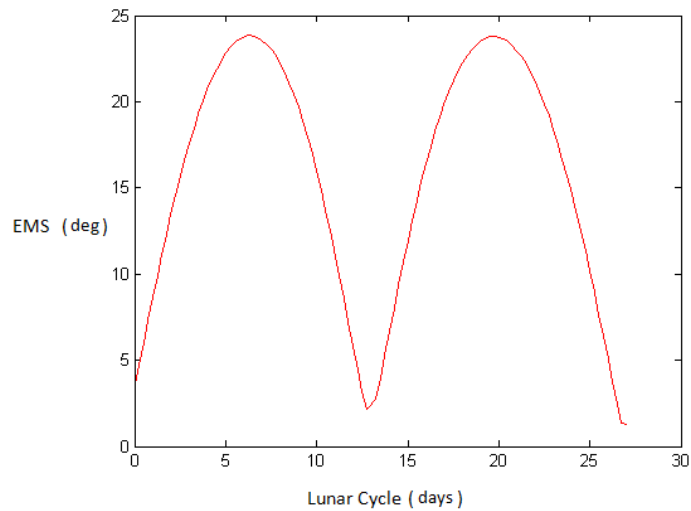


Figure 4.6: EMS Angle Vs. Days from jd=2457000.5

the trend in the sum of the maneuvers computed by the initial guess tool follows

the same trend across the lunar cycle as seen in Figure 4.7. The optimal transfer magnitudes are achieved via the methods described in Chapter 6, and are only included in this figure to provide as a frame of reference. Using the Julian Date epoch of 2457000.5 chosen for this investigation, the ΔV variation shown in Figure 4.7 is not an issue, but this trend should be acknowledged for applications of this guess generator outside this study.

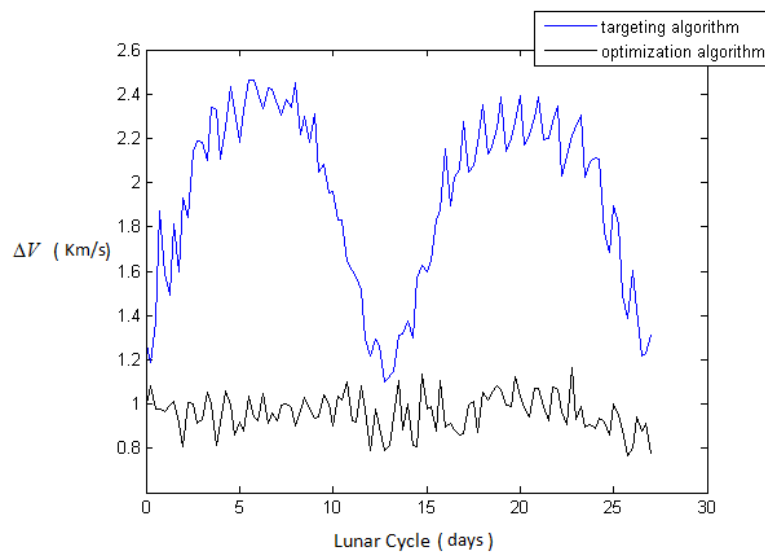


Figure 4.7: Maneuver Sum vs. Days from $jd=2457000.5$

4.1.5 Patch Point Selection

Beyond the generation of the initial guess itself, the placement of the patch points is a matter of concern. Since a Level-One correction is executed between patch points, the computational time increases with an increasing number of patch states, making their selection important.

There are some simple guidelines that are generally followed when selecting

these patch points from the startup arc. First, the patch points are chosen far enough apart. In doing this the computation time of the TLC process is not lengthy, and the number of patch states is not so excessive that the general shape of the transfer is constrained. On the other hand, care is also taken to include enough patch points to adequately represent the startup trajectory. The distribution is not necessarily evenly spaced in time, and dynamically sensitive regions require a level of different distributions of the patch states. Finally, the maneuvers along the initial guess arc are always included in the patch point set to provide an adequate representation of the initial guess arc to the TLC process.

An example of patch point selection is seen in Figure 4.8. The patch points selected (red crosses) here are chosen from the MCI frame, because it is more intuitive to identify the TEI sequence within this frame. Once these states are captured, they are transformed into the ECI frame in preparation for the TLC process.

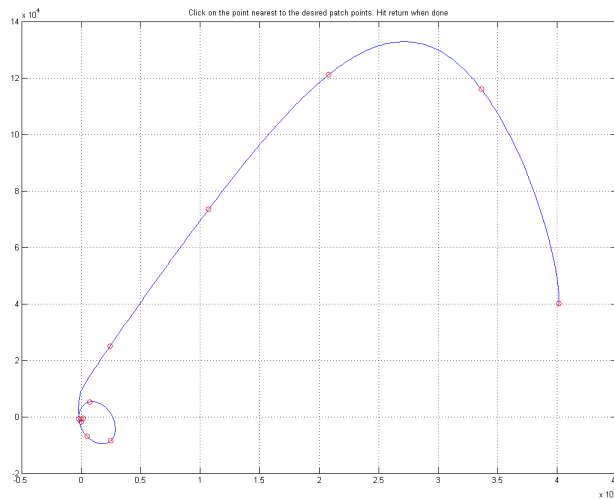


Figure 4.8: Patch Point Selection

4.2 Numerical Integration

The differential equations given by Equations 2.13, 2.14, and 2.26, represent the differential equations that must be solved, in both the initial guess generation and the TLC solution process. However, since these differential equations do not admit an exact solution, a numerical approximation to their solution is pursued instead. This requires the use of numerical integration. The integrator chosen for this study is based on an explicit Runge-Kutta 4-5 formula, using the Dormand-Prince pair [20].

4.3 Interpreting the Results

Using the TLC process and the initial guess generator described in Section 4.1, several example cases are generated. The results include a graphical representation of both the initial guess and the converged solution. Aside from the plots, additional information is included such as: entry state information, ΔV values, and an indication of the computational time. These are captured in Tables 4.1 through 4.3. In discussing the computation time, however, it is important to note that because this work is completed in MATLAB, the computation times are unnecessarily large. Ultimately being transferred to C code, these computation times are inflated by at least an order of magnitude. Convergence of the TLC process is identified by a vector that contains the set of constraint errors. When the magnitude of this vector is reduced below $1e-4$, convergence is achieved. Within the initial guess procedure, however, the convergence threshold is defined by $1e-3$ within the Level-Two component.

For consistency, the initial epoch and the lunar orbit parameters are held

constant across each of these five cases as

$$\begin{aligned}
 t_0 &= 2014 - 12 - 9 : 00 \text{ s}, \\
 a &= 100 \text{ km}, \\
 ecc &= 0^\circ, \\
 inc &= 135^\circ, \\
 \Omega &= 0 \text{ km/s}, \\
 \omega &= 0^\circ.
 \end{aligned} \tag{4.5}$$

To further enhance the level of comparison among the examples, the entry targets are also held fixed as

$$\begin{aligned}
 R &= 6500.065 \text{ km}, \\
 \phi &= -19.2041^\circ, \\
 \theta &= 134.5456^\circ, \\
 \gamma &= -6.03^\circ, \\
 Az &= 13.996^\circ.
 \end{aligned} \tag{4.6}$$

4.4 TLC Results: Five Example Cases

An ideal return involves using a minimum amount of propellant and successfully targeting of a pre-specified: altitude, latitude, longitude, flight path angle, and flight path azimuth at entry interface. However, in the event that failures occur onboard, including the loss of ground communication, the minimum requirement is that the CEV return safely to Earth within the available means (origin of the 1.4

km/s ΔV budget constraint).

The first example case presented here demonstrates the TLC's ability to target the altitude, latitude, longitude, and flight path angle constraints at entry interface. In addition, a constraint is imposed on the maneuver sum magnitude, arbitrarily set to 0.9 km/s.

The plot of the initial guess for this first case is seen in Figure 4.9. This arc is generated using the process described in Section 4.1. However, because so many features of the entry state are being constrained in this case, the Level-Two portion of the initial guess generation is difficult to converge. To allow more freedom in the convergence process, an additional maneuver is added in the initial guess process after TEI-3. This maneuver acts to increase the number of controls in the targeting process, and prevents TEI-3 from becoming unnecessarily large. In addition to this change, an artificial patch point is added prior to the periapsis, shifting TEI-3 to an interior patch point of the Level-Two process and allowing a maneuver sum constraint to be imposed. This 4-maneuver initial guess in Figure 4.9 meets the altitude, latitude, longitude, and flight path angle constraints, and requires only 1.16 km/s in total ΔV .

Applying the TLC process to a 4-burn initial guess arc naturally lends to a 4-burn solution. However, it is advantageous to reduce this to a 3-burn transfer. After omitting the patch point associated with the fourth maneuver and enforcing the altitude, latitude, longitude, flight path angle, and maneuver sum constraints in the TLC process, the solution in Figure 4.10 is obtained. Tables 4.1 through 4.3 indicate that only 10 iterations and 244.5 seconds of computation time are required to target this transfer. Combining this computation time with that required to construct the initial guess yields a total computational effort of six minutes and 19

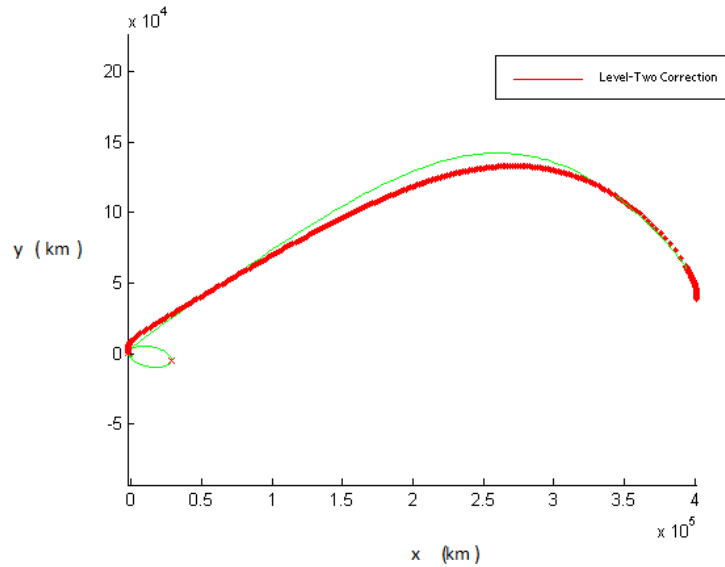


Figure 4.9: Case 1: Initial Guess (MCI)

seconds.

As opposed to the solution in Figure 4.10, the simpler approach to targeting the altitude, latitude, longitude, and flight path angle involves applying the TLC process directly to the guess in Figure 4.9 without omitting the fourth maneuver or imposing a maneuver sum constraint. This is the basis of the second case considered in this investigation. As expected, the result is a 4-burn transfer with a larger maneuver sum than the first case. A plot of the transfer is shown in Figure 4.11. Note that in comparing this result against case 1 in Table 4.1, the maneuver sum is increased by 328 m/s and the computation is decreased by 114.6 seconds. This captures the trade that exists between maneuver magnitude and TLC computation time. In imposing an additional constraint on the maneuver sum, such as in case 1, the total ΔV is reduced, but computation time is necessarily sacrificed. Still, in

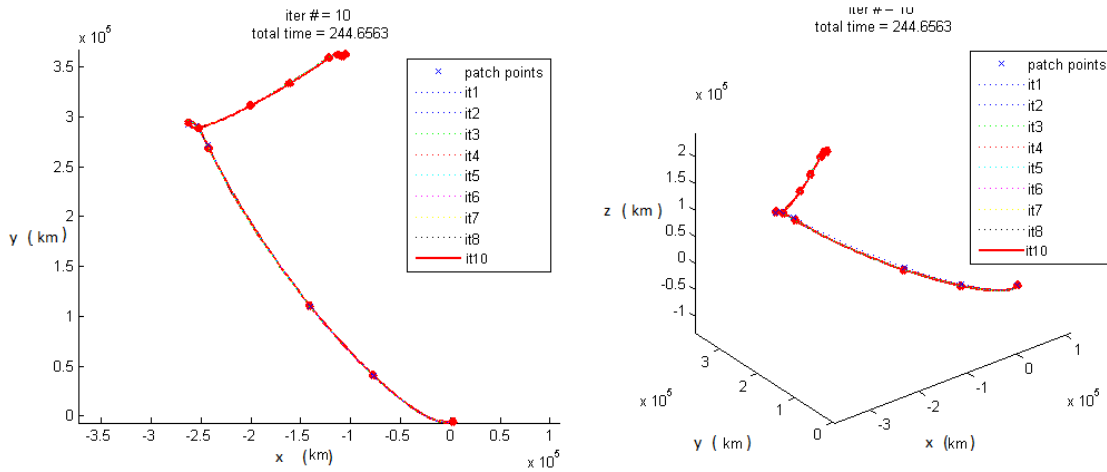


Figure 4.10: Case 1: Converged Solution (ECI)

both case 1 and case 2, all imposed constraints are satisfied and the 1.4 km/s ΔV budget constraint is met.

Cases 1 and 2 represent viable targeting strategies, targeting nearly every component of the entry state. It is important to note that the most essential of these constraints is the entry flight path angle, due to the limiting nature of entry heating constraints. Given this, a more basic set of requirements can be imposed by the autonomous onboard targeting algorithm and still provide for a safe entry in the event of communication loss. Imposing the altitude and flight path angle as the lone targeting requirements certainly admits variability in the landing site obtained at Earth, however it also provides the most essential features of a fail-safe return strategy. This is the basis for case 3 in this investigation.

Because the flight path angle is a component of the velocity vector, it is recommended that the Level-Two portion of the initial guess process again be imposed. This allows for all the entry constraints to once again be satisfied at the onset of

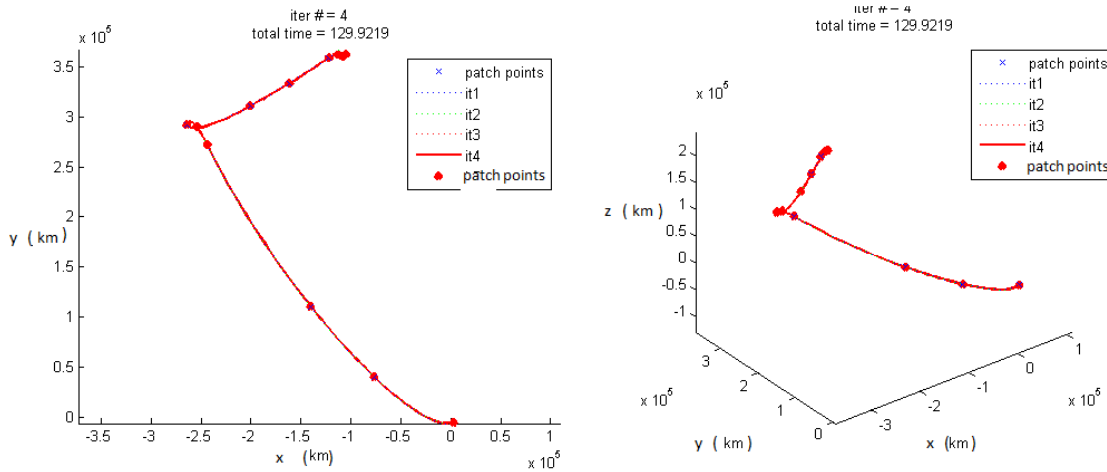


Figure 4.11: Case 2: Converged Solution (ECI)

the TLC process. The initial guess created for case 3 is shown in Figure 4.12. It satisfies the altitude and flight path angle constraints, requires only 4 iterations of the Level-Two process, and involves a mere ΔV of 1.21 km/s.

Selecting a set of patch states from Figure 4.12 and applying the TLC process yields the converged solution seen in Figure 4.13. The 124 seconds required for the initial guess and TLC computation is 140.4 seconds less than case 2. This is an expected result, as fewer entry constraints provide a simpler targeting task for the TLC process. This is representative of a trade that exists between the number of imposed constraints and the required computation time of the TLC process.

Note that no effort is made to reduce the total maneuver magnitude in Figure 4.13. While the sum of maneuvers for this case can be reduced by imposing a constraint, such as in case 1, it is significant to note that it is well below the 1.4 km/s budget without one. This is permitted by the initial guess generation process. The sole purpose of the process outlined in Section 4.1 is to provide the TLC with

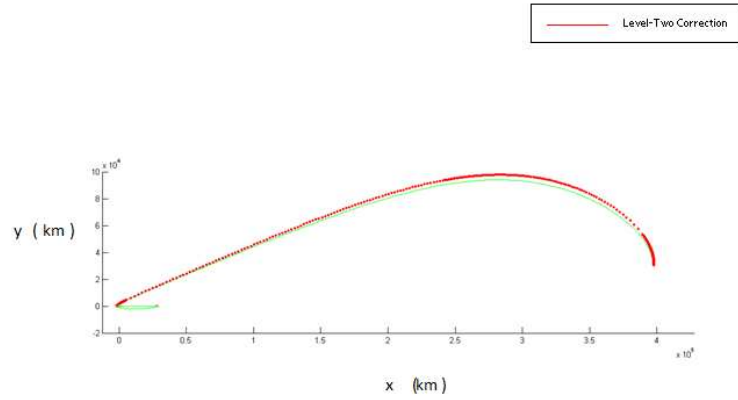


Figure 4.12: Case 3: Initial Guess (MCI)

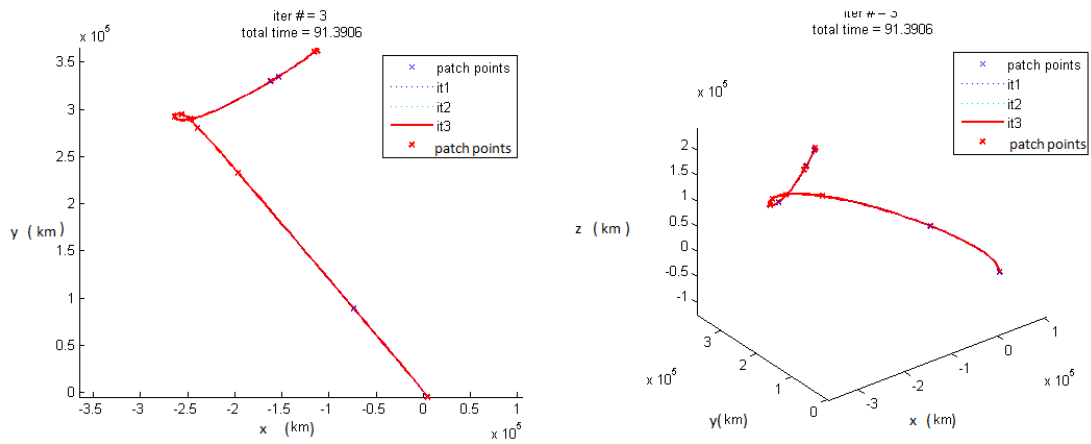


Figure 4.13: Case 3: Converged Solution (ECI)

an arc in the vicinity of a feasible and propulsively viable solution.

An alternate initial guess for targeting the altitude and flight path angle constraints is shown by the plot in Figure 4.14. This plot is also generated using the process described in Section 4.1, only without the inclusion of the Level-Two portion of the initial guess. In doing this, the entry flight path angle is not satisfied by the startup arc. The Level-One process (Chapter 3) only has the ability to target altitude, latitude, and longitude at a pre-specified entry time. However, the benefit in removing the Level-Two portion of the initial guess is that a mere 14.3 seconds of computation time are required to generate guess arc.

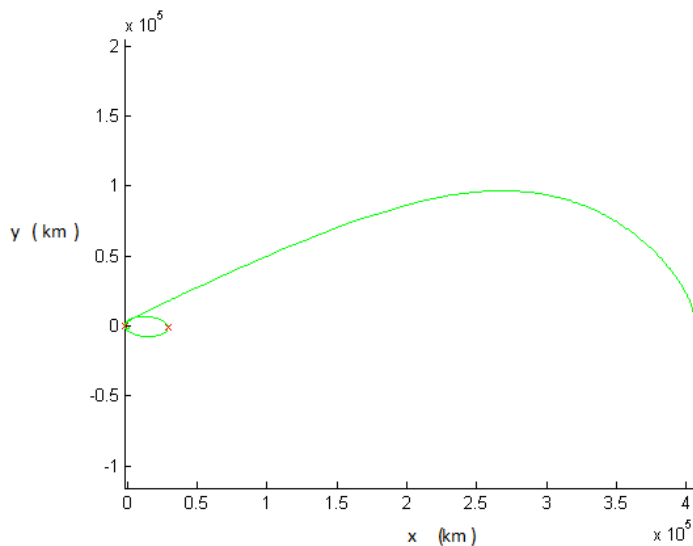


Figure 4.14: Case 4: Initial Guess (MCI)

Applying the TLC process to the initial guess in Figure 4.14 achieves the converged transfer seen in Figure 4.15. Unfortunately, the TLC computation time is 279.8 seconds greater than the value in case 3. A great deal of this computational effort is dedicated to satisfying the maneuver sum constraint that is imposed to

keep the total ΔV at or below 1.25 km/s. Unconstrained, the magnitude of the maneuvers becomes greater than 3 km/s in this example. For this particular case, the computational time saved in the initial guess procedure is more than undone by the loss in the TLC time required. As shown in case 3, this extraneous amount of computation time is preventable by providing a guess that satisfies the entry constraints. This case demonstrates the benefit of an initial guess that meets the entry requirements.

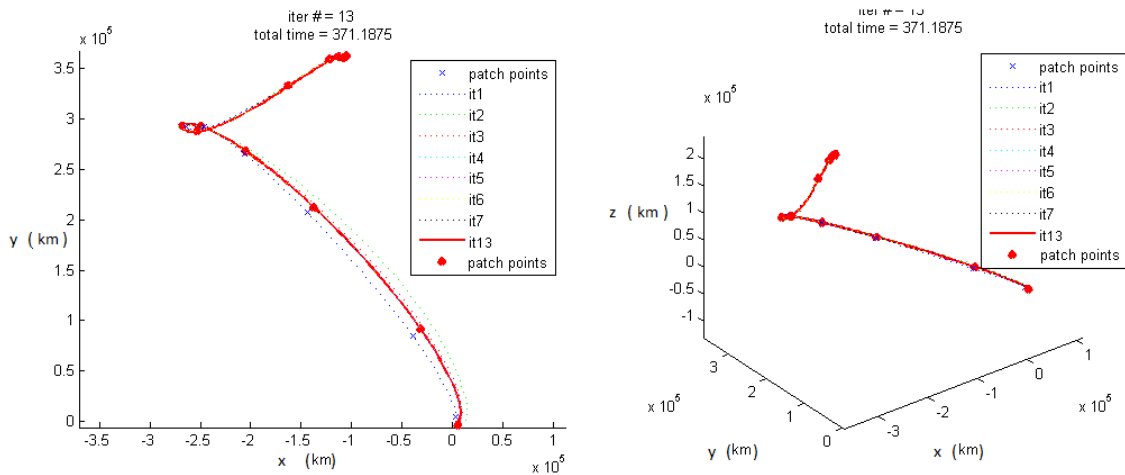


Figure 4.15: Case 4: Converged Solution (ECI)

Thus far, an example has not been considered that attempts to constrain the entire entry state in Equation 4.6. This is the basis for the final case in this section, where only the speed is not constrained, which is naturally about 11 km/s. It is advantageous in this case to once again: add an additional patch point, allow an extra maneuver, and constraint the maneuver sum in the initial guess generation process. Figure 4.16 shows this initial guess, which requires 25 iterations, 364 seconds of computation time, and a ΔV of 2.3 km/s. From Table 4.2 it is seen that

this maneuver magnitude is nearly twice that of any initial guess in cases 1-4.

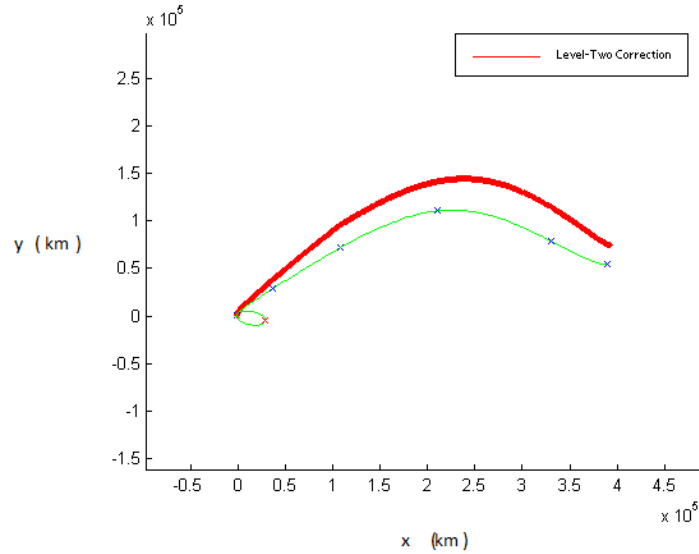


Figure 4.16: Case 5: Initial Guess (MCI)

Selecting patch points from Figure 4.16 and applying the TLC process with a maneuver sum constraint yields the converged solution in Figure 4.17. While 371 seconds of computational effort is required for this TLC solution, all imposed constraints are satisfied. This is a significant result given that the initial guess is 900 m/s above the ΔV budget constraint. Of course, while it is remarkable that the TLC process is still able to identify a solution in the vicinity of the startup arc, the computation time can be reduced through the use of an alternate initial guess.

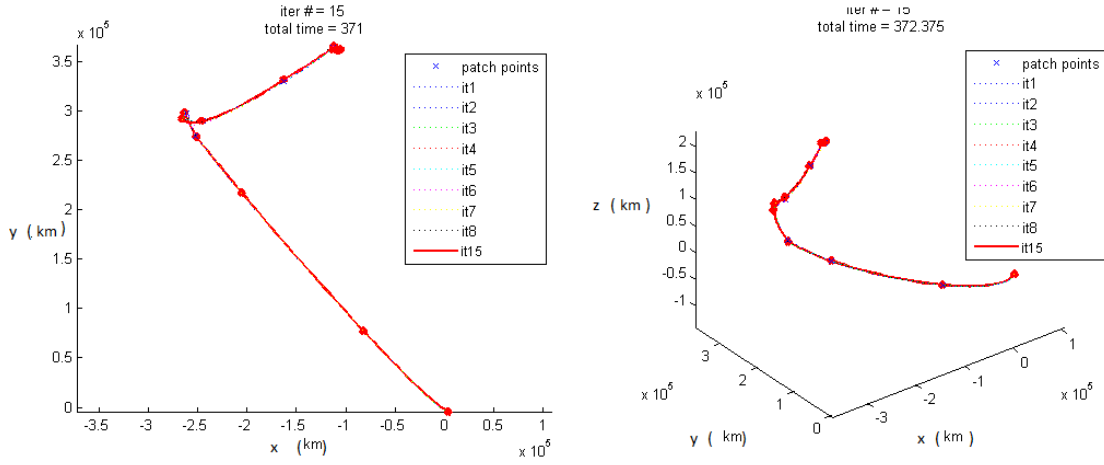


Figure 4.17: Case 5: Converged Solution (ECI)

| Case | Iter. | Time(s) | ΔV_1 | ΔV_2 | ΔV_3 | ΔV_4 | ΔV | Active Constraints |
|------|-------|---------|--------------|--------------|--------------|--------------|------------|---|
| 1 | 10 | 244.5 | 608.8 | 43.6 | 247.5 | 0 | 899.9 | $R, \phi, \theta, \gamma, \Delta V$ |
| 2 | 4 | 129.9 | 608.2 | 56.8 | 459.7 | 103.6 | 1228.3 | R, ϕ, θ, γ |
| 3 | 3 | 91.4 | 607.6 | 56.9 | 544.8 | 0 | 1209.3 | R, γ |
| 4 | 13 | 371.2 | 602.6 | 106.1 | 541.3 | 0 | 1250 | R, γ |
| 5 | 15 | 371 | 629.1 | 70.6 | 389.2 | 261.1 | 1350 | $R, \phi, \theta, \gamma, Az, \Delta V$ |

Table 4.1: Summary of the TLC Example Results

| Case | ΔV | Iter. | Time(s) |
|------|------------|-------|---------|
| 1 | 1.16 | 9 | 134.5 |
| 2 | 1.16 | 9 | 134.5 |
| 3 | 1.21 | 4 | 32.6 |
| 4 | 1.28 | 0 | 14.3 |
| 5 | 2.3 | 25 | 364 |

Table 4.2: Summary of the Initial Guess Used in Each Example Case

| Case | R(km) | ϕ (deg) | θ (deg) | V(km/s) | γ (deg) | Az(deg) | TOF(s) |
|------|----------|--------------|----------------|---------|----------------|---------|-----------|
| 1 | 6500.065 | -19.2041 | 134.5456 | 10.993 | -6.03 | 69.858 | 536758.77 |
| 2 | 6500.065 | -19.2041 | 134.5456 | 10.989 | -6.03 | 73.089 | 536518.99 |
| 3 | 6500.065 | -0.54747 | 125.5254 | 10.984 | -6.03 | 178.706 | 557034.59 |
| 4 | 6500.065 | -17.4617 | 134.8519 | 10.991 | -6.03 | 75.512 | 544232.45 |
| 5 | 6500.065 | -19.2041 | 134.5456 | 10.993 | -6.03 | 13.996 | 626183.16 |

Table 4.3: Summary of the Entry States for Example Cases

Chapter 5

Two Level Correction with Finite Thrust

Traditionally, the TLC process is comprised of impulsive maneuvers. Despite the fact that no realistic engine has the ability to instantaneously change the velocity of the spacecraft, impulsive maneuvers stand as an adequate approximation for the “standard” TEI maneuvers and Earth-return TCM’s. However, if an engine failure occurs onboard the CEV, the available thrust level decreases, and the burn duration for executing maneuvers increases. In this circumstance, the impulsive approximation is no longer adequate. Therefore, in the same spirit of the impulsive burn formulation in Chapter 3, a TLC procedure using finite burns is considered.

In developing this procedure, the Level-One correction is first determined, followed by the Level-Two portion. Furthermore, the constraints from Chapter 3 are reusable within the Level-Two process, and the same general rules apply. Finally, a few simple examples are presented at the end of the chapter to show the success achieved in the limited application of this procedure.

5.1 Alterations to the Physical Model

To begin, a new state representation and dynamical model are required. For example, a thrust arc depends on mass, mass flow, and burn direction information. In response to this, the state is appended to include this information,

$$\bar{X} = \begin{bmatrix} \bar{R} \\ \bar{V} \\ m \\ \dot{m} \\ \bar{u} \end{bmatrix}_{11 \times 1} . \quad (5.1)$$

Note, for this investigation it is assumed that the mass flow rate and thrust direction values remain constant across a thrust arc. The equations of motion then need to be altered as well to account for the acceleration produced by the thrusters. An expression for thrust is given by

$$T = -\frac{\dot{m}}{m} g_0 I_{sp} . \quad (5.2)$$

Using Equation 5.2, the equations of motion that follow are

$$\dot{\bar{X}} = \begin{bmatrix} \bar{V} \\ \ddot{\bar{R}} + \frac{T}{m} \frac{\bar{u}}{|\bar{u}|} \\ -\dot{m}_g \\ \ddot{m} \\ \dot{\bar{u}} \end{bmatrix}_{11 \times 1} , \quad (5.3)$$

where \ddot{R} is defined by Equation 2.13. Beyond this, changes are made to the linearized dynamics due to these alterations in the equations of motion shown in Equation 5.3. The Jacobian for this thrusting segment takes the new form,

$$\frac{\partial \dot{\bar{X}}}{\partial \bar{X}} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} \\ \frac{\partial \ddot{R}}{\partial R} & 0_{3 \times 3} & \frac{\partial \ddot{R}}{\partial m} & \frac{\partial \ddot{R}}{\partial \dot{m}_g} & \frac{\partial \ddot{R}}{\partial \bar{u}} \\ 0_{1 \times 3} & 0_{1 \times 3} & 0 & 1 & 0_{1 \times 3} \\ 0_{1 \times 3} & 0_{1 \times 3} & 0 & 0 & 0_{1 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 3} \end{bmatrix}. \quad (5.4)$$

The partial derivatives contained in Equation 5.4 are expressed by the following:

$$\frac{\partial \ddot{R}}{\partial R} = \sum_{\substack{j=1 \\ j \neq q,s}}^n Gm_j \left(\frac{3\bar{R}_{sj}\bar{R}_{sj}^T}{R_{sj}^5} - \frac{1}{R_{sj}^3} I_{3 \times 3} \right), \quad (5.5)$$

$$\frac{\partial \ddot{R}}{\partial m} = \frac{\dot{m}}{m^2} g_0 I_{sp} \frac{\bar{u}^T}{u}, \quad (5.6)$$

$$\frac{\partial \ddot{R}}{\partial \dot{m}} = -\frac{1}{m} g_0 I_{sp} \frac{\bar{u}^T}{u}, \quad (5.7)$$

$$\frac{\partial \ddot{R}}{\partial \bar{u}} = -\frac{\dot{m}}{m} g_0 I_{sp} \frac{1}{u} I_{3 \times 3} - \frac{\dot{m}}{m} g_0 I_{sp} (\bar{u}^T \bar{u})^{-3/2} \bar{u} \bar{u}^T. \quad (5.8)$$

Of course, the new Jacobian definition leads to a new state transition matrix. This is represented by $\tilde{\Phi}'$ in the variational equation,

$$\left(\delta \bar{X}_f^- - \dot{\bar{X}}_f^- \delta t_f \right) = \tilde{\Phi}'_{t_f, t_0} \left(\delta \bar{X}_0^+ - \dot{\bar{X}}_0^+ \delta t_0 \right). \quad (5.9)$$

Expanded, this variational equation is

$$\begin{bmatrix} \delta\bar{R}_f - \bar{V}_f^- \delta t_f \\ \delta\bar{V}_f^- - \bar{a}_f^- \delta t_f \\ \delta m_f^- - \dot{m}_f^- \delta t_f \\ \delta\dot{m}_f^- - \ddot{m}_f^- \delta t_f \\ \delta\bar{u}_f^- - \dot{u}_f^- \delta t_f \end{bmatrix} = \begin{bmatrix} \tilde{A}_{t_f,t_0} & \tilde{B}_{t_f,t_0} & \tilde{E}_{t_f,t_0} & \tilde{F}_{t_f,t_0} & \tilde{G}_{t_f,t_0} \\ \tilde{C}_{t_f,t_0} & \tilde{D}_{t_f,t_0} & \tilde{H}_{t_f,t_0} & \tilde{I}_{t_f,t_0} & \tilde{J}_{t_f,t_0} \\ \tilde{K}_{t_f,t_0} & \tilde{L}_{t_f,t_0} & \tilde{M}_{t_f,t_0} & \tilde{N}_{t_f,t_0} & \tilde{O}_{t_f,t_0} \\ \tilde{P}_{t_f,t_0} & \tilde{Q}_{t_f,t_0} & \tilde{R}_{t_f,t_0} & \tilde{S}_{t_f,t_0} & \tilde{T}_{t_f,t_0} \\ \tilde{U}_{t_f,t_0} & \tilde{V}_{t_f,t_0} & \tilde{W}_{t_f,t_0} & \tilde{X}_{t_f,t_0} & \tilde{Y}_{t_f,t_0} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_0 - \bar{V}_0^+ \delta t_0 \\ \delta\bar{V}_0^+ - \bar{a}_0^+ \delta t_0 \\ \delta m_0^+ - \dot{m}_0^+ \delta t_0 \\ \delta\dot{m}_0^+ - \ddot{m}_0^+ \delta t_0 \\ \delta\bar{u}_0^+ - \dot{u}_0^+ \delta t_0 \end{bmatrix}. \quad (5.10)$$

5.2 Level-One Correction Using Finite Thrust

Using Equation 5.10, the Level-One portion of the correction process is constructed. The controls over a thrusting segment used to match position continuity are \dot{m}_0 , \bar{u}_0 , and t_f . At the initial point along a thrusting segment: \bar{R}_0 , \bar{V}_0 , m_0 , and t_0 are fixed. Therefore, $\delta\bar{R}_0 = \delta\bar{V}_0 = \delta t_0 = \delta m_0 = 0$. Using this information, the first vector expression of Equation 5.10 reduces to

$$\delta\bar{R}_f = \tilde{F}_{t_f,t_0} \delta\dot{m}_0 + \tilde{G}_{t_f,t_0} \delta\bar{u}_0 + \bar{V}_f^- \delta t_f. \quad (5.11)$$

Equation 5.11 is rewritten in the form,

$$\delta\bar{R}_f = \begin{bmatrix} \tilde{F}_{t_f,t_0} & \tilde{G}_{t_f,t_0} & \bar{V}_f^- \end{bmatrix} \begin{bmatrix} \delta\dot{m}_0 \\ \delta\bar{u}_0 \\ \delta t_f \end{bmatrix}. \quad (5.12)$$

Finally, the minimum norm is applied to Equation 5.12 to determine a solution.

This is represented by

$$\begin{bmatrix} \delta m_0 \\ \delta \bar{u}_0 \\ \delta t_f \end{bmatrix} = \tilde{M}^T \left(\tilde{M} \tilde{M}^T \right)^{-1} \delta \bar{R}_0, \quad (5.13)$$

where $\tilde{M} = \begin{bmatrix} \tilde{F}_{t_f, t_0} & \tilde{G}_{t_f, t_0} & \bar{V}_f^- \end{bmatrix}$.

Equation 5.13 represents the Level-One finite thrust process. Note that while the mass flow is allowed to vary, a constraint for its magnitude can be implemented within the Level-Two process, if desired.

5.3 Level-Two Correction Using Finite Thrust

To outline the Level-Two process, it is useful to consider the image given in Figure 5.1. In this configuration of patch points: thrust is active across the first segment

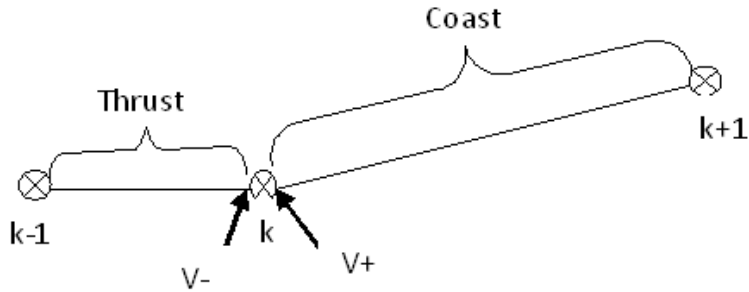


Figure 5.1: An Example of an Finite Thrust trajectory

to achieve the position at patch point k, an impulsive maneuver then exists at k

to achieve patch point k+1. Using the a Level-Two process, however, the position, time, mass flow, and thrust direction associated with these patch points can be manipulated such that the impulsive maneuver at patch point k is driven to zero. If this is accomplished it naturally follows that the thrust arc, rather than the impulsive maneuver, targets the position at patch point k+1. This yields a solely finite transfer and is thus a motivating example for this Level-Two corrections process.

Just as in the impulsive formulation, the condition for satisfying velocity continuity is

$$\delta\Delta\bar{V}_k = 0 = \delta\bar{V}_k^+ - \delta\bar{V}_k^- . \quad (5.14)$$

Once again, if Equation 5.14 is expressed in terms of the controls, velocity continuity can be achieved. However, unlike the impulsive case, the controls for the finite burn formulation are small perturbations (variations) in: mass flow, thrust direction, position, and time.

In solving for $\delta\bar{V}_k^-$ in terms of the controls, the variational equation for a thrusting arc is applied from k-1 to k,

$$\begin{bmatrix} \delta\bar{R}_k - \bar{V}_k^- \delta t_k \\ \delta\bar{V}_k^- - \bar{a}_k^- \delta t_k \\ \delta m_k^- - \dot{m}_k^- \delta t_k \\ \delta \dot{m}_k^- - \ddot{m}_k^- \delta t_k \\ \delta \bar{u}_k^- - \dot{\bar{u}}_k^- \delta t_k \end{bmatrix} = \begin{bmatrix} \tilde{A}_{k,k-1} & \tilde{B}_{k,k-1} & \tilde{E}_{k,k-1} & \tilde{F}_{k,k-1} & \tilde{G}_{k,k-1} \\ \tilde{C}_{k,k-1} & \tilde{D}_{k,k-1} & \tilde{H}_{k,k-1} & \tilde{I}_{k,k-1} & \tilde{J}_{k,k-1} \\ \tilde{K}_{k,k-1} & \tilde{L}_{k,k-1} & \tilde{M}_{k,k-1} & \tilde{N}_{k,k-1} & \tilde{O}_{k,k-1} \\ \tilde{P}_{k,k-1} & \tilde{Q}_{k,k-1} & \tilde{R}_{k,k-1} & \tilde{S}_{k,k-1} & \tilde{T}_{k,k-1} \\ \tilde{U}_{k,k-1} & \tilde{V}_{k,k-1} & \tilde{W}_{k,k-1} & \tilde{X}_{k,k-1} & \tilde{Y}_{k,k-1} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_{k-1} - \bar{V}_{k-1}^+ \delta t_{k-1} \\ \delta\bar{V}_{k-1}^+ - \bar{a}_{k-1}^+ \delta t_{k-1} \\ \delta m_{k-1}^+ - \dot{m}_{k-1}^+ \delta t_{k-1} \\ \delta \dot{m}_{k-1}^+ - \ddot{m}_{k-1}^+ \delta t_{k-1} \\ \delta \bar{u}_{k-1}^+ - \dot{\bar{u}}_{k-1}^+ \delta t_{k-1} \end{bmatrix} . \quad (5.15)$$

As an intermediate step in defining $\delta\bar{V}_k^-$ in terms of the control variables, the vector

$\delta\bar{V}_{k-1}^+$ is determined from the first vector expression of Equation 5.15,

$$\begin{aligned}\delta\bar{V}_{k-1}^+ = & \quad (-\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1})\delta\bar{R}_{k-1} + (\bar{a}_{k-1}^+ + \tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1}\bar{V}_{k-1}^+ + \tilde{B}_{k,k-1}^{-1}\tilde{E}_{k,k-1}\dot{m}_{k-1}^+ \\ & + \tilde{B}_{k,k-1}^{-1}\tilde{F}_{k,k-1}\dot{m}_{k-1}^+ + \tilde{B}_{k,k-1}^{-1}G_{k,k-1}\dot{u}_{k-1}^+) \delta t_{k-1} - (\tilde{B}_{k,k-1}^{-1}\tilde{E}_{k,k-1})\delta m_{k-1} \quad (5.16) \\ & - (\tilde{B}_{k,k-1}^{-1}\tilde{F}_{k,k-1})\delta\dot{m}_{k-1} - (\tilde{B}_{k,k-1}^{-1}k, k-1)\delta\bar{u}_{k-1} + (\tilde{B}_{k,k-1}^{-1})\delta\bar{R}_k - (\tilde{B}_{k,k-1}^{-1}\bar{V}_k^-)\delta t_k.\end{aligned}$$

Then, Equation 5.16 is substituted into the second vector equation of Equation 5.15 to yield

$$\begin{aligned}\delta\bar{V}_k^- = & \quad \left(\tilde{C}_{k,k-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1}\right)\delta\bar{R}_{k-1} + (-\tilde{C}_{k,k-1}\bar{V}_{k-1}^+ - \tilde{H}_{k,k-1}\dot{m}_{k-1}^+ \\ & - \tilde{I}_{k,k-1}\dot{m}_{k-1}^+ - \tilde{J}_{k,k-1}\dot{u}_{k-1}^+ + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1}\bar{V}_{k-1}^+ + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{E}_{k,k-1}\dot{m}_{k-1}^+ \\ & + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{F}_{k,k-1}\dot{m}_{k-1}^+ + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}G_{k,k-1}\dot{u}_{k-1}^+)\delta t_{k-1}^+ + \quad (5.17) \\ & \left(\tilde{H}_{k,k-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{E}_{k,k-1}\right)\delta m_{k-1}^+ + \left(\tilde{I}_{k,k-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{F}_{k,k-1}\right)\delta\dot{m}_{k-1}^+ + \\ & \left(\tilde{J}_{k,k-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{G}_{k,k-1}\right)\delta\bar{u}_{k-1}^+ + \left(\bar{a}_k^- - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\bar{V}_k^-\right)\delta t_k^- + \\ & \left(\tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\right)\delta\bar{R}_k.\end{aligned}$$

Applying the variational equation from $k+1$ to k , $\delta\bar{V}_k^+$ is found similarly as

$$\begin{aligned}\delta\bar{V}_k^+ = & \quad \left(\tilde{C}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1}\right)\delta\bar{R}_{k+1}^- + (-\tilde{C}_{k,k+1}\bar{V}_{k+1}^- \\ & + \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1}\bar{V}_{k+1}^-)\delta t_{k+1}^- + \\ & \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{E}_{k,k+1}\right)\delta m_{k+1}^- + \\ & \left(\bar{a}_k^+ - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\bar{V}_k^+\right)\delta t_k^+ + \left(\tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\right)\delta\bar{R}_k^+. \quad (5.18)\end{aligned}$$

In this case, however, the terms associated with the thrust drop out since no thrusting occurs across the coast arc. Given Equations 5.17 and 5.18, the condition stated

in Equation 5.14 is nearly defined in terms of the control parameters. However, aside from the control parameters, Equations 5.17 and 5.18 are also a function of mass. Since mass is a dependent variable, velocity continuity is only achievable if these terms are removed from the expression. This is done by expressing variations in mass in terms of the control variables.

Given that thrust occurs from $k-1$ to k , the total mass at the end of the thrust is found by

$$m_k = m_{k-1} + \dot{m}_{k-1}^+ (t_k - t_{k-1}), \quad (5.19)$$

where \dot{m} is negative for any realistic engine. Noting that Equation 5.19 is a function of m_{k-1} , \dot{m}_{k-1} , t_k , and t_{k-1} , it can be seen that changes in m_k occur if any changes are made to these values. Mathematically, this statement is represented by

$$\delta m_k = \left(\frac{\partial m_k}{\partial m_{k-1}} \right) \delta m_{k-1} + \left(\frac{\partial m_k}{\partial \dot{m}_{k-1}^+} \right) \delta \dot{m}_{k-1}^+ + \left(\frac{\partial m_k}{\partial t_{k-1}} \right) \delta t_{k-1} + \left(\frac{\partial m_k}{\partial t_k} \right) \delta t_k. \quad (5.20)$$

Evaluating the partials on the right hand side of Equation 5.20 provides the expressions:

$$\begin{aligned} \frac{\partial m_k}{\partial m_{k-1}} &= 1, \\ \frac{\partial m_k}{\partial \dot{m}_{k-1}^+} &= (t_k - t_{k-1}), \\ \frac{\partial m_k}{\partial t_{k-1}} &= -\dot{m}_{k-1}^+, \\ \frac{\partial m_k}{\partial t_k} &= \dot{m}_{k-1}^+. \end{aligned} \quad (5.21)$$

An expression for the variation in m_k with respect to variations in the control

parameters is found by substituting Equation 5.21 into Equation 5.20,

$$\delta m_k = \delta m_{k-1} + (t_k - t_{k-1}) \delta \dot{m}_{k-1}^+ + (-\dot{m}_{k-1}^+) \delta t_{k-1} + (\dot{m}_{k-1}^+) \delta t_k. \quad (5.22)$$

Since $\delta m_{k-1} = 0$ for the example in Figure 5.1, Equation 5.22 provides an expression for δm_k in terms of the control variables.

Substituting Equation 5.22 into Equation 5.17 and noting that $\ddot{m}_{k-1} = \dot{u}_{k-1} = 0$ for this example,

$$\begin{aligned} \delta \bar{V}_k^- = & \left(\tilde{C}_{k,k-1} - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{A}_{k,k-1} \right) \delta \bar{R}_{k-1} + (-C_{k,k-1} \bar{V}_{k-1}^+ - \tilde{H}_{k,k-1} \dot{m}_{k-1}^+ \\ & + \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{A}_{k,k-1} \bar{V}_{k-1}^+ + \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{E}_{k,k-1} \dot{m}_{k-1}^+) \delta t_{k-1}^+ + \\ & \left(\tilde{I}_{k,k-1} - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \tilde{F}_{k,k-1} \right) \delta \dot{m}_{k-1}^+ + \left(\tilde{J}_{k,k-1} - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} k, k-1 \right) \delta \bar{u}_{k-1}^+ + \\ & \left(\bar{a}_k^- - \tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \bar{V}_k^- \right) \delta t_k^- + \left(\tilde{D}_{k,k-1} \tilde{B}_{k,k-1}^{-1} \right) \delta \bar{R}_k. \end{aligned} \quad (5.23)$$

For a thrust-coast sequence, $m_k = m_{k+1}$, and therefore $\delta m_k = \delta m_{k+1}$. Using this information, substituting Equation 5.22 into Equation 5.18, and also noting that $\delta \dot{m}_{k+1}^- = \delta \bar{u}_{k+1}^- = \dot{m}_{k+1}^- = \ddot{m}_{k+1} = \dot{u}_{k+1} = 0$, the expression for $\delta \bar{V}_k^+$ becomes

$$\begin{aligned} \delta \bar{V}_k^+ = & \left(\tilde{C}_{k,k+1} - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \right) \delta \bar{R}_{k+1} + (-\tilde{C}_{k,k+1} \bar{V}_{k+1}^- + \\ & \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{A}_{k,k+1} \bar{V}_{k+1}^-) \delta t_{k+1}^- + \\ & \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{E}_{k,k+1} \right) (t_k - t_{k-1}) \delta \dot{m}_{k-1}^+ + \\ & \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{E}_{k,k+1} \right) (-\dot{m}_{k-1}^+) \delta t_{k-1} + \\ & + \left(\bar{a}_k^+ - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \bar{V}_k^+ + \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \tilde{E}_{k,k+1} \right) (\dot{m}_{k-1}^+) \right) \delta t_k^+ + \\ & \left(\tilde{D}_{k,k+1} \tilde{B}_{k,k+1}^{-1} \right) \delta \bar{R}_k^+. \end{aligned} \quad (5.24)$$

Substituting Equations 5.23 and 5.24 into Equation 5.14 yields the Level-Two por-

tion of the finite thrust TLC procedure,

$$\delta\Delta\bar{V}_k = \begin{bmatrix} \tilde{M}_{R_{k-1}} & \tilde{M}_{t_{k-1}} & \tilde{M}_{\dot{m}_{k-1}} & \tilde{M}_{\bar{u}_{k-1}} & \tilde{M}_{R_k} & \tilde{M}_{t_k} & \tilde{M}_{R_{k+1}} & \tilde{M}_{t_{k+1}} \end{bmatrix} \begin{bmatrix} \delta R_{k-1} \\ \delta t_{k-1} \\ \delta m_{k-1} \\ \delta \dot{m}_{k-1} \\ \delta \bar{u}_{k-1} \\ \delta R_k \\ \delta t_k \\ \delta R_{k+1} \\ \delta t_{k+1} \end{bmatrix}. \quad (5.25)$$

The components of the state-relationship matrix in Equation 5.25 are given by

$$\begin{aligned} \tilde{M}_{R_{k-1}} &= \left(-\tilde{C}_{k,k-1} + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1} \right), \\ \tilde{M}_{t_{k-1}} &= \left(\tilde{C}_{k,k-1}\bar{V}_{k-1}^+ + \tilde{H}_{k,k-1}\dot{m}_{k-1}^+ - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{A}_{k,k-1}\bar{V}_{k-1}^+ \right. \\ &\quad \left. - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{E}_{k,k-1}\dot{m}_{k-1}^+ + \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{E}_{k,k+1} \right) (-\dot{m}_{k-1}^+) \right), \\ \tilde{M}_{\dot{m}_{k-1}} &= \left(-\tilde{I}_{k,k-1} + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{F}_{k,k-1} \right) + \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{E}_{k,k+1} \right) (t_k - t_{k-1}), \\ \tilde{M}_{\bar{u}_{k-1}} &= \left(-\tilde{J}_{k,k-1} + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\tilde{G}_{k,k-1} \right), \\ \tilde{M}_{R_k} &= \left(\tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1} - \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1} \right), \\ \tilde{M}_{t_k} &= \left(\bar{a}_k^+ - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\bar{V}_k^+ - \bar{a}_k^- + \tilde{D}_{k,k-1}\tilde{B}_{k,k-1}^{-1}\bar{V}_k^- \right) + \\ &\quad \left(\tilde{H}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{E}_{k,k+1} \right) (\dot{m}_{k-1}^+), \\ \tilde{M}_{R_{k+1}} &= \left(\tilde{C}_{k,k+1} - \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1} \right), \\ \tilde{M}_{t_{k+1}} &= \left(-\tilde{C}_{k,k+1}\bar{V}_{k+1}^- + \tilde{D}_{k,k+1}\tilde{B}_{k,k+1}^{-1}\tilde{A}_{k,k+1}\bar{V}_{k+1}^- \right). \end{aligned} \quad (5.26)$$

An example of this methodology applied to a thrust-coast transfer is con-

tained in Figure 5.2. Here an altitude constraint is targeted along with the velocity continuity constraint at patch point k . The finite thrust TLC procedure requires more iterations than the impulsive TLC process, but nevertheless steady convergence to the solution is achieved.

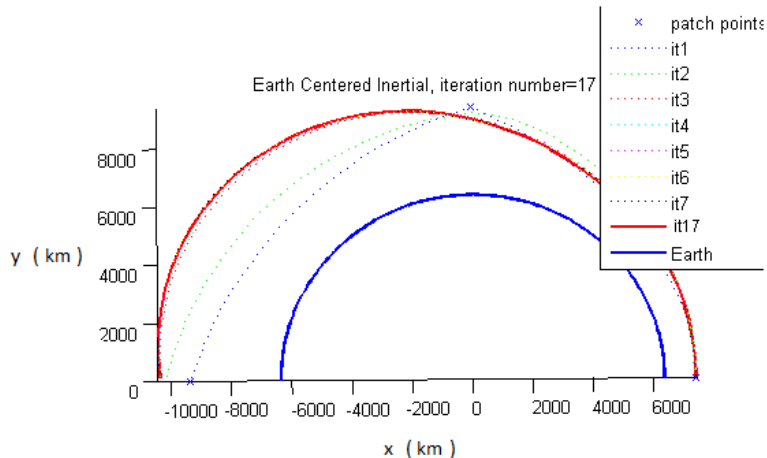


Figure 5.2: Targeting Altitude with a Thrust-Coast Arc

It is crucial to mention that at the beginning of a thrust arc in this implementation, it is assumed that the initial velocity is equivalent to the terminal velocity of the previous arc. Therefore, the only velocity discontinuities that exist are those between a thrust-coast or coast-coast segments. Thrust-thrust and coast-thrust scenarios do not have a maneuver at their intermediate patch point and thus a velocity continuity constraint is not required. For this reason, only the thrust-coast formulation is needed for the finite thrust TLC implementation.

5.4 Example Case

Before an example case is presented, the process for guessing these thrust parameters is described for the finite thrust implementation. For the finite thrust implementation, a guess for the mass flow and thrust direction is required in addition to the set of patch states. This is calculated via the rocket equation found in many texts including one written by Weisel [21]. Specifically, using pre-specified information about the I_{sp} , initial mass, and impulsive $\Delta\bar{V}$ required to achieve the next patch state,

$$\dot{m}_k = \frac{(1 - e^{-\frac{\Delta V_k}{I_{sp}g_0}})m_k}{t_{k+1} - t_k}, \quad (5.27)$$

and

$$\bar{u}_k = \frac{\Delta\bar{V}_k}{\Delta V_k}. \quad (5.28)$$

Given the simple initial guess procedure defined by Equations 5.27 and 5.28, along with the same set of patch points contained in Figure 3.3, the simple example case for the study is solved. Note that for this case, a 1000 km altitude constraint is imposed along with the velocity continuity constraints, and the initial mass is set to 1000 kg.

Figure 5.3 contains a solution that thrusts across the first arc and coasts across the remaining four. Therefore, in order to achieve the 1000 km altitude constraint with the exclusive use of finite thrust, the maneuvers at patch states 2,3,4, and 5 must be driven to zero by the TLC process. The maneuver at patch state 2 requires the formulation from this chapter, where 3 through 5 are treated with the impulsive TLC process. In determining this solution, the mass flow rate is only updated in the Level-One portion of the procedure, and in doing so a total of 25 iterations and 156 seconds are required to achieve this solution. With the

maneuvers at patch points 2-5 removed, the resulting thrust arc has a burn time of 1750.8 seconds, and a mass flow rate of -0.035 kg/s.

Note that this solution, like any other obtained by the TLC process is dictated by the initial guess. For instance, if the second patch state in this example is moved closer to the first, the solution would have a smaller burn time and a larger mass flow rate.

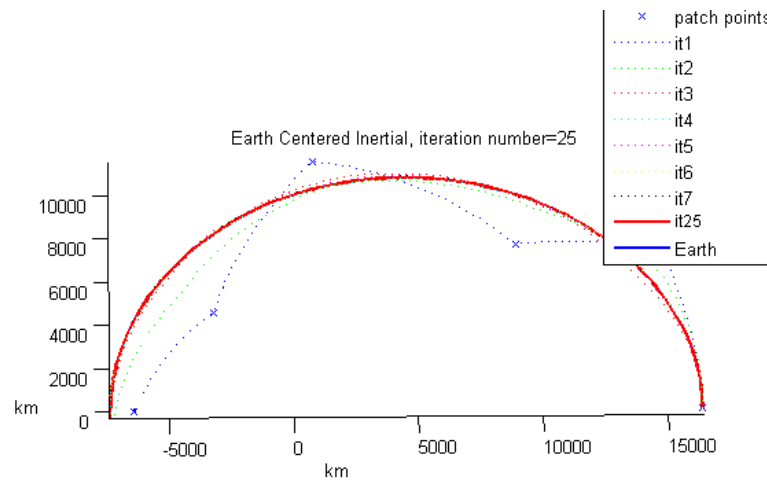


Figure 5.3: Thrust-Coast-Coast-Coast-Coast Arc

Figure 5.3 represents a simple application and an initial attempt at modeling finite thrust. This algorithm, however, has not yet been applied to an Orion transfer.

Chapter 6

Numerical Optimization

Numerically optimal solutions for several of the impulsive TEI transfers in Chapter 4 are presented in this chapter. Calculating an optimal TEI burn sequence enables enhanced analysis to be performed for this study. That is, since the targeting algorithm only provides a feasible solution, the presentation of optimal transfers in this chapter enables a comparison to be made between the TLC results and the optimal transfers themselves.

Before the optimization results are discussed, a set of introductory materials pertaining to numerical optimization is first presented. This introduction includes a description of parameter optimization theory, numerical optimization methods, and a few optimization techniques themselves. The parameter optimization theory and numerical optimization discussions provide insight into the theory behind the commercial nonlinear programming (NLP) software (VF13AD) utilized in Section 6.4.

6.1 Parameter Optimization Theory

From basic calculus [12], it is known that an extremum along a function is located where that function's derivative evaluates to zero. For example, a function, F , has a minimum with respect to the parameter, x , when

$$F = \frac{\partial F}{\partial x}(x) = 0. \quad (6.1)$$

Equation 6.1 is the basis for parameter optimization, and it is referred to as the first-order necessary condition for optimality [22]. Three general cases for parameter optimization are outlined: unconstrained, equality constrained, and inequality constrained. These methods are discussed in detail by Hull [3].

6.1.1 Unconstrained Parameter Optimization

Consider the scalar performance index,

$$J = \varphi(\bar{x}). \quad (6.2)$$

For a minimum, \bar{x}^* , it holds for all values of \bar{x} such that $\Delta J \geq 0$ for the function

$$\Delta J = \varphi(\bar{x}) - \varphi(\bar{x}^*). \quad (6.3)$$

Expanding Equation 6.3 using a Taylor Series Expansion,

$$\begin{aligned} \Delta J &= dJ + \frac{1}{2!}d^2J + HOT's, \\ \Delta J &= \varphi_x(\bar{x})dx - \frac{1}{2!}\varphi_x(\bar{x})dx + HOT's. \end{aligned} \quad (6.4)$$

Note that ‘dJ’ is a differential, and the differential of an independent variable is a constant. An expression for this differential is

$$dJ = \varphi_x(\bar{x})dx. \quad (6.5)$$

Given Equation 6.5, the first-order necessary conditions for a minimum are

$$\varphi_x = 0. \quad (6.6)$$

6.1.2 Parameter Optimization with Equality Constraints

Consider, once again, the scalar performance index given by Equation 6.2. Introducing a set of equality constraints,

$$\bar{\psi}(x) = 0. \quad (6.7)$$

The performance index can include these constraints through the use of a Lagrange multiplier [3], ν ,

$$J' = G = \varphi(\bar{x}) + \bar{\nu}^T \bar{\psi}(\bar{x}). \quad (6.8)$$

This augmented term seen in Equation 6.8 evaluates to zero when the constraints are met and therefore does not alter the cost.

To find a minimum of this function, the point at which slope of the function evaluates to zero is located. Given this, the differential is required to evaluate zero. Determining the differential of the cost function in Equation 6.8 involves the partial derivative with respect to both the independent variables \bar{x} and $\bar{\nu}$,

$$dJ = G_x dx + \psi d\nu. \quad (6.9)$$

Therefore, the first-order necessary conditions for a minimum are,

$$\begin{aligned} G_x &= 0, \\ \psi &= 0. \end{aligned} \tag{6.10}$$

6.1.3 Parameter Optimization with Inequality Constraints

A set of inequality constraints to the problem such as

$$\bar{\theta}(x) = x - x_{des} \geq 0, \tag{6.11}$$

can be converted to equality constraints with the use of slack variables [3], α^2 ,

$$x - x_{des} - \alpha^2 = 0. \tag{6.12}$$

The performance index may include these terms using the Lagrange multiplier, ν , under the same general reasoning as that in Equation 6.8,

$$J' = G = \varphi(\bar{x}) + \bar{\nu}^T(\theta - \alpha^2). \tag{6.13}$$

To find a minimum of Equation 6.13, the point at which slope of the function evaluates zero is located. Taking the differential of the cost function involves taking the partial derivatives with respect to the three independent variables \bar{x} , $\bar{\nu}$, and α ,

$$dJ = \varphi_x dx + \bar{\nu} dx - 2\bar{\nu}\alpha d\alpha + (\theta - \alpha^2) d\nu. \tag{6.14}$$

Therefore, the first-order necessary conditions for a minimum are

$$\begin{aligned}\varphi_x + \bar{\nu} &= 0, \\ -2\bar{\nu}\alpha &= 0, \\ \theta - \alpha^2 &= 0.\end{aligned}\tag{6.15}$$

Two solutions exist for the conditions introduced by Equation 6.15. One of these solutions is on the constraint boundary ($\alpha = 0$), and the other is off of the constraint boundary ($\bar{\nu} = 0$). Therefore, in determining the solution \bar{x} is chosen as the dependent variable and α the independent variable.

6.2 Unconstrained Minimization

For many problems, the minimum is achievable via the analytical approaches shown in Equations 6.2 through 6.15. However, for complex problems possessing complex cost functions and many constraints, it is much easier or even a necessary to solve the problem using numerical optimization methods. Lunar Trans-Earth Injection is an example of such a problem, and therefore the focus from this point forward is on the numerical approach to solving these types of problems.

Many numerical methods exist that are capable of minimization. The first criteria for categorizing these approaches is constrained versus unconstrained minimization. Unconstrained minimization is the simpler of these two classes and is the basis of this section's discussion. A much more extensive discussion on these classes of numerical optimization methods is performed by Bertsekas [23]. In this section, one-dimensional searches (ODS) are first presented. These are also sometimes referred to as zeroth-order methods. Beyond this, first and second-order methods are

also discussed.

6.2.1 One-Dimensional Line Search

The simplest form of unconstrained minimization is the one-dimensional search (ODS). This is a zeroth-order approximation problem, because it does not require any derivative information. In this method, the minimum of the function is sought out along a single search direction. The general form for a one-dimensional search method is

$$G(x + \sigma dx), \tag{6.16}$$

where σ is the step size.

The solution process involves two portions: bracketing the minimum, and locating the minimum. Bracketing the minimum requires several function evaluations along the search direction. Essentially, if a decrease in the function is followed by an increase of the same function at an increased step size, the minimum is bracketed between these function evaluations. This is referred to as up-down-up [24]. Still, even after the minimum is bracketed, a large interval of uncertainty can remain such as the example shown in Figure 6.1. A general method for bracketing the minimum is described by:

1. Given the value of $G(x)$, find $G(x + \sigma dx)$ where the search direction, dx , is already provided. Initially choose $\sigma = 1$.
2. Perform another function evaluation, only this time at $G(x + (\sigma + d\sigma)dx)$. Note that initially, $d\sigma = 2\Delta\sigma$, where $\Delta\sigma = 1$ is a reasonable choice as well.
3. Perform step 2 repeatedly until the desired up-down-up orientation is found. Increase $d\sigma$ by $4\Delta\sigma, 8\Delta\sigma$, etc. at each iteration.

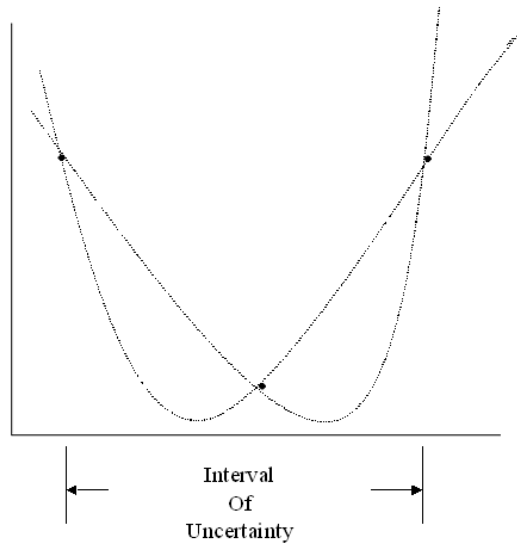


Figure 6.1: Representation of an Interval of Uncertainty

Many methods exist for a one-dimensional search. Some of the more notable are: the bisection method, Fibonacci search, golden ratio method, and curve fitting approach. While an exhaustive search of the function space is one avenue for finding the minimum of a function, line searches act to minimize the number of function evaluations needed to locate the minimum.

Bisection, Fibonacci, and Golden-Ratio

The golden ratio, Fibonacci, and bisection methods are all very similar in their methodology. In fact, the following algorithm description is applicable to all three:

1. Begin with 4 points along the function at step sizes of $\sigma_l, \sigma_1, \sigma_2, \sigma_u$, where σ_l and σ_u shown in Figure 6.2 correspond to the lower and upper bounds of the interval of uncertainty provided by the minimum bracketing shown in Figure

6.1.

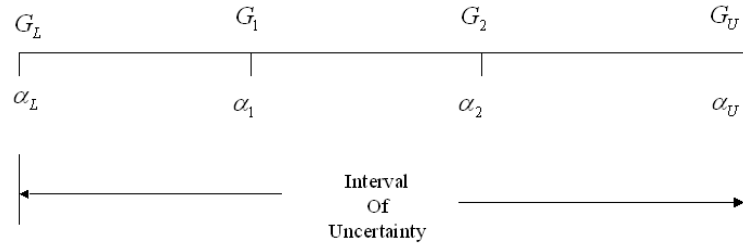


Figure 6.2: Representation of an Interval of Uncertainty in terms of σ

2. Reduce the interval of uncertainty:

- (a) If $G_1 \leq G_2$ then set $G_L = G_1$, $G_2 = G_2$, $G_U = G_U$, and recalculate a new σ_1, G_1 .
- (b) If $G_2 \leq G_1$ then set $G_U = G_2$, $G_1 = G_1$, $G_L = G_L$, and recalculate a new σ_2, G_2 .

3. Repeat step 2 until the minimum is found within some tolerance.

Given this general formulation for locating a minimum in a one-dimensional search, the lone difference that distinguishes bisection, from golden ratio, from Fibonacci is the manner in which the step sizes are updated. Figure 6.3 shows an evolution of the reduction of the interval of uncertainty. For the golden-ratio search,

$$L_{i+1} = (0.6218)L_i. \quad (6.17)$$

Therefore, the interval of uncertainty is reduced by 0.6218 on each successive iteration. This is superior to the bisection method, which reduces the interval of

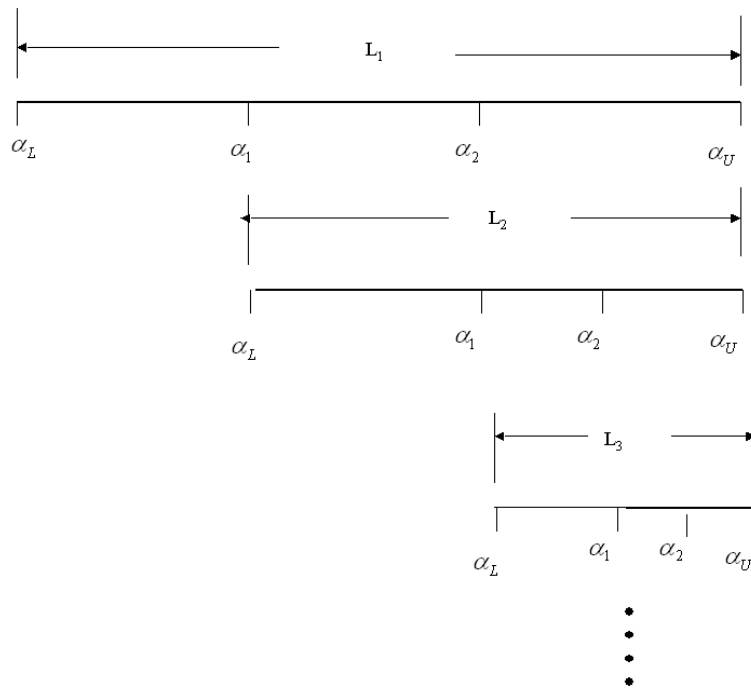


Figure 6.3: Reduction of the Interval of Uncertainty

uncertainty by 0.5 on each iteration. A slight improvement to these methods is made by using a ratio of Fibonacci numbers rather than the golden-ratio. That is, replacing 0.6218 with a ratio of Fibonacci numbers, $\frac{F_{n-i}}{F_{n-i+1}}$.

While the Fibonacci search is slightly more efficient than the golden-ratio method, this discrepancy in efficiency vanishes as n goes to infinity. Furthermore, Fibonacci faces a unique disadvantage in that the number of iterations must be specified beforehand in order to properly retrieve the required set of Fibonacci numbers for the search.

Curve-Fitting

A different one-dimensional search to those mentioned is the curve-fitting routine. In this method, a parabola is fit to the current function evaluations across the interval of uncertainty. Consider first the general equation for a parabola,

$$P = R\sigma^2 + S\sigma + T. \tag{6.18}$$

Three function evaluations are required to solve for R,S, and T using

$$\begin{aligned} P_1 &= R\sigma_1^2 + S\sigma_1 + T, \\ P_2 &= R\sigma_2^2 + S\sigma_2 + T, \\ P_3 &= R\sigma_3^2 + S\sigma_3 + T. \end{aligned} \tag{6.19}$$

Once R,S, and T are solved for, a parabola is fit to the data, and the minimum is found by

$$\begin{aligned}\frac{\partial P}{\partial \sigma} &= 0 = 2R\sigma + S, \\ \sigma_4 &= -\frac{S}{2R}.\end{aligned}\tag{6.20}$$

Convergence occurs when $G(\sigma) = P(\sigma)$ to within some tolerance. That is, when the parabolic approximation for the minimum is equivalent to the function evaluation at σ , the minimum of the function is achieved. However, if the current iteration does not yet satisfy these conditions, Up-Down-Up is chosen from the set of 4 σ 's and the process is repeated.

6.2.2 First-Order Methods

Thus far, several methods have been shown for finding the minimum along a particular search direction. First-Order methods act to determine a search direction in the n-dimensional space. One such numerical method is known as the gradient method [23], where the search direction is defined as

$$dx = -G_x^T(x).\tag{6.21}$$

This method is also referred to as the method of steepest descent, because it can be shown that this search direction produces the greatest decrease in the cost function. The following process defines this method:

1. Provide an initial guess, x .
2. Calculate $G(x)$ and $G_x(x)$.

3. One of the ODS methods mentioned in the previous section are employed to find the σ that minimizes the function along this search direction, G_x .
4. If the minimum is not yet achieved, steps 2 and 3 are repeated using the new value for x .

An example of the gradient method is shown in Figure 6.4. The function being minimized in this example is,

$$G(x_1, x_2) = x_1^2 + \frac{1}{2}x_2^2, \quad (6.22)$$

and the initial guess used for the minimization is $x = [-1.25, 1.0]$. After 7 iterations,

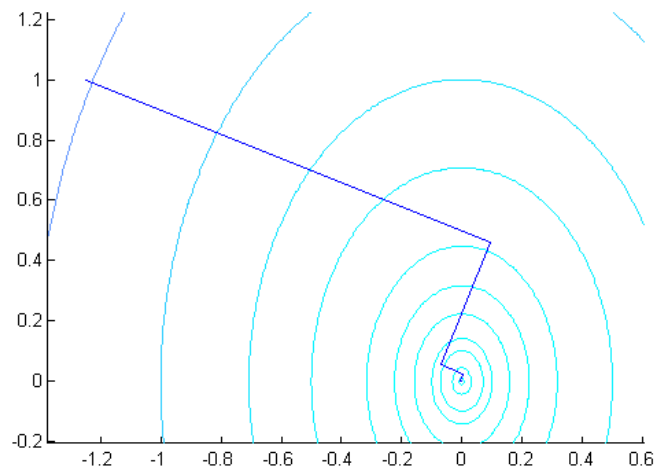


Figure 6.4: Gradient Method Applied to a Quadratic Function: 7 Iterations

the minimum is located at $x = [0.0, 0.0]$.

The function given by Equation 6.22, and the starting point of $x = [-1.25, 1.0]$, are also employed by the remaining examples in this chapter. This allows for adequate comparisons to be made between each of the methods.

6.2.3 Second-Order Methods

While the gradient method provides an adequate means for determining the search direction, other options exist. Second-order methods utilize first and second order information in determining the search direction for the minimum. For example, consider the Taylor series expansion,

$$\Delta G(x) = G(x + dx) - G(x) = G_x(x)dx + \frac{1}{2!}G_{xx}d^2x + HOT's. \quad (6.23)$$

Eliminating all terms beyond the second-order and solving for dx when $\frac{\partial \Delta G}{\partial dx} = 0$,

$$dx = -G_x^T(x)G_{xx}(x)^{-1}. \quad (6.24)$$

Equation 6.24 represents a Newton-Raphson Method [23]. In this method, however, a disadvantage exists in that $G_{xx}(x)$ must remain positive definite for the computation of this inverse to exist. A way to avoid this dilemma imposed by the Newton-Raphson method is to employ what is known as a Variable Metric Method [24]. This method addresses the positive definite issue by using an approximation for the inverse of G_{xx} (the Hessian). The approximation, H, is ensured to remain positive definite by definition. The process for using the Variable Metric method is:

1. Provide an initial guess, x.
2. Calculate $G(x)$, $G_x(x)$, and set H to an identity matrix.

3. Calculate $dx = -H_k G_x$.
4. One of the zeroth-order ODS methods mentioned in the previous section is employed to find the σ that minimizes the function along this search direction, dx .
5. If the minimum is not yet achieved ($G_x = 0$ to within some tolerance), set $k=k+1$, update H , calculate G_x at the new value of x , and return to step 3.

Several updates for H are available, and the method highlighted here is known as the BFGS update [23],

$$H_{k+1} = \left[I - \frac{\Delta x \Delta G_x^T}{\Delta x^T \Delta G_x} \right] H_k \left[I - \frac{\Delta G_x \Delta x^T}{\Delta G_x^T \Delta x} \right] + \frac{\Delta x \Delta x^T}{\Delta x^T \Delta G_x}. \quad (6.25)$$

Within Equation 6.25,

$$\begin{aligned} \Delta x &= x_{k+1} - x_k, \\ \Delta G_x &= G_{x_{k+1}} - G_{x_k}. \end{aligned} \quad (6.26)$$

An example of this method is shown in Figure 6.5. The function and initial guess are the same as those used in Figure 6.4. Compared to the gradient method, the number of iterations is decreased by nearly fifty percent for this particular example.

6.3 Constrained minimization

Often times in parameter optimization it is necessary to impose constraints. At the very least, the final boundary conditions are typically imposed as constraints.

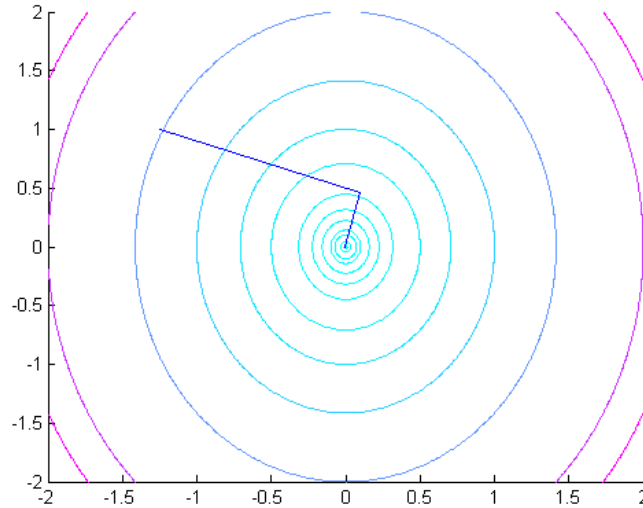


Figure 6.5: Variable Metric Method With a BFGS Update Applied to a Quadratic Function: 4 Iterations

Multiple methods exist finding a minimum subject to constraints. The first such method mentioned here involves the use of penalty functions.

6.3.1 Penalty Functions and Unconstrained Minimization

Constraints may be addressed using unconstrained minimization techniques, by employing the use of penalty functions [24]. Penalty functions are implemented by augmenting G with the constraint such that there is a penalty paid on the cost function whenever this constraint is violated. The general form of this expression is given by

$$G = G + \sum_{j=1}^m \max(0, g_j(\bar{x}))^2 + \sum_{l=1}^m (h(\bar{x}))^2. \quad (6.27)$$

Applying this technique to Equation 6.22, subject to the constraint

$$x_2 = x_1^2 + 0.1, \quad (6.28)$$

gives

$$G(x_1, x_2) = x_1^2 + \frac{1}{2}x_2^2 + \Gamma (\max(0, x_1^2 - x_2 + 0.1)), \quad (6.29)$$

where Γ is the penalty parameter.

Applying the gradient technique and variable metric techniques to Equation 6.29 results in the plots shown in Figures 6.6 and 6.7. As expected, the variable metric method is more efficient than the gradient method, but both converge to the minimum.

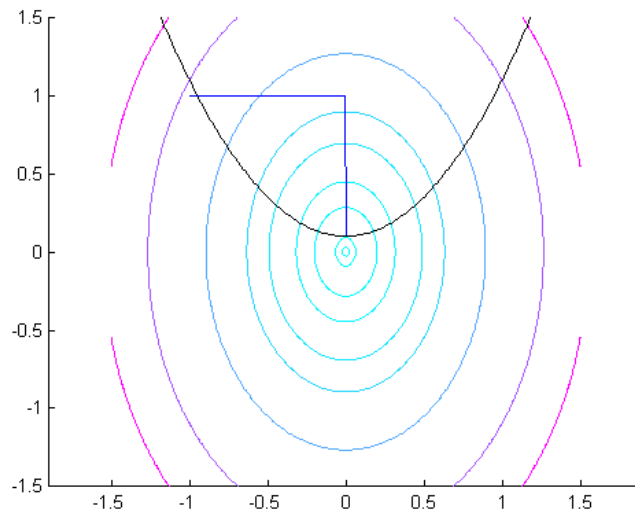


Figure 6.6: Gradient Method Applied to a Quadratic Function: 17 Iterations

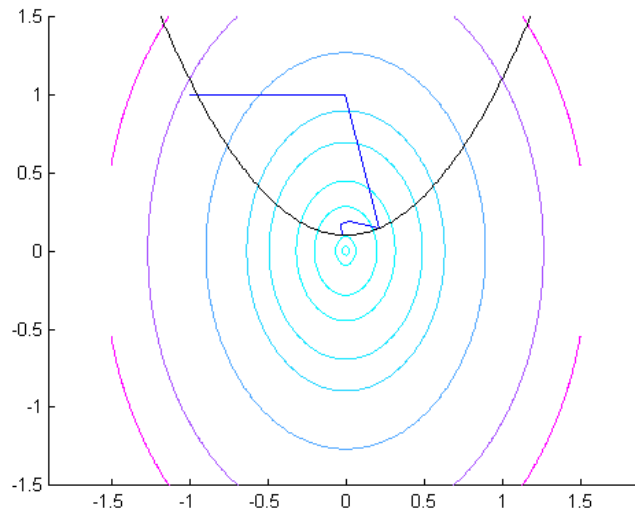


Figure 6.7: Variable Metric Method With a BFGS Update Applied to a Quadratic Function: 6 Iterations

6.3.2 Constrained Minimization Techniques

While the penalty function approach may work in many cases, it is still ambiguous in its application due to the arbitrary penalty weight. Many approaches exist that deal directly with the presence of constraints, using them to their advantage. Such algorithms include: the Method of Feasible Directions, Reduced Gradient Method, Robust Method of Feasible Directions, Augmented Lagrangian, and others [24]. While these methods represent viable options for creating a minimization code, many commercial codes exist for this same purpose. These commercial codes are exceedingly fast and robust. They are built by professional programmers, and are an excellent means for performing numerical optimization. Some of these commercial codes include: SNOPT, FMINCON(Matlab), KNITRO, VF13AD, and others.

VF13AD

Chosen for its availability and simplicity of implementation, VF13AD is implemented as the nonlinear program (NLP) of choice for this study. The algorithm itself is obtained for free from the Harwell function library, and runs on the FORTRAN platform. It utilizes Sequential Quadratic Programming (SQP) as a means for performing the minimization, finding a step away from the current point by minimizing a quadratic approximation to the problem. The variable metric technique described in Section 6.2.3 is utilized by VF13AD in the direction finding process. When constraints are present in the main calculation, the linear approximations to the constraints are included in the calculation of the search direction as well. Interestingly enough, however, VF13AD also uses a penalty function approach similar to Section 6.3.1 in dealing with constraints throughout the one dimensional search as part of what is known the watchdog technique [25]. Finally, while a thorough description of the ODS is proprietary information, Section 6.2.1 provides for good insight into the general thought process that is involved in a ODS process.

6.3.3 Finite Differencing

In each of the numerical methods presented here, aside from the one-dimensional search, derivatives are required. These derivatives dictate the search direction calculation, and they can be determined numerically. Numerical derivatives are computed using a technique called finite differencing [23]. While these derivatives are more computationally burdensome and less accurate than analytical derivatives, they are very convenient. That is, any time the optimization problem changes, the derivatives must be recomputed. By using finite differencing, the user is free to alter the problem without recomputing the derivatives. Because of this flexibility, finite

differencing is used to determine the derivatives within this numerical optimization investigation.

In determining the formula for finite differencing, it is useful to begin with a Taylor series expansion of the cost function,

$$\Delta G(x) = G(x + dx) - G(x) = G_x(x)dx + \frac{1}{2!}G_{xx}d^2x + HOT's. \quad (6.30)$$

Using the first order information only, a numerical approximation is found for the derivative by what is termed forward differencing,

$$G_x(x) = \frac{G(x + dx) - G(x)}{dx}. \quad (6.31)$$

Using a slightly different Taylor series expansion,

$$\Delta G(x) = G(x - dx) - G(x) = -G_x(x)dx + \frac{1}{2!}G_{xx}d^2x + HOT's, \quad (6.32)$$

an expression for backward differencing is similarly achieved,

$$G_x(x) = \frac{G(x - dx) + G(x)}{dx}. \quad (6.33)$$

Using second order information, a more accurate form of finite differencing, known as central differencing is achieved. This involves the subtraction of Equations 6.30 and 6.32. The expression for central differencing is

$$G_x = \frac{G(x + dx) - G(x - dx)}{2dx}. \quad (6.34)$$

It should be noted that the selection of dx heavily influences the accuracy

of finite differencing. If dx is chosen to be a large value, much of the behavior close to x is lost. Conversely, a value that is too small fails to capture what is happening sufficiently close to x . An adequate choice for dx is one that depends on the magnitude of x . Therefore, a convenient statement is

$$dx = \epsilon|x| \tag{6.35}$$

It can be shown that $\epsilon = 10^{-4}$ gives roughly a precision accurate to $1e-8$ when using central differences. [23]

6.3.4 Optimization Techniques

The realm of optimization techniques is divided into two subcategories, direct and indirect optimization. Direct optimization involves the parameterization, also known as transcription, of the problem. Specifically, this means representing the physical description of the optimization problem in terms of parameters and constraints. With a properly parameterized problem, a nonlinear programming code, such as VF13AD may be used to find the optimal solution subject to the constraints.

Indirect optimization, however, uses Optimal Control Theory (OCT) [3] to solve the Two-Point-Boundary-Value (TPBVP), where the term TPBVP implies any function that has constraints at the initial and final points. Using OCT, the first-order necessary conditions for a minimum are derived for the particular TPBVP. Once these are obtained, a root finding function rather than a nonlinear program, is used to obtain a solution to the first-order necessary conditions. A downside to the indirect approach, however, is that for any change to the optimization problem, the TPBVP changes and the first-order necessary conditions must be re-derived.

Direct Multiple Shooting

Direct optimization techniques include: single shooting, multiple shooting, collocation, and several others. The focus in this study remains on multiple shooting. The difference between single and multiple shooting is gathered directly from their names. In single shooting, a single shot (numerical integration from the initial boundary to the final boundary) is used to target the end constraints of the TP-BVP. In multiple shooting, the number of “shots” is free to be chosen, but after each of them a set of continuity constraints is enforced to ensure that the integration is continuous in position and velocity. An artist’s rendition of these two methods is shown in Figure 6.8. The initial and final points in this figure represent the TP-BVP, and the break shown in the multiple shooting arc represents the location of the continuity constraints.

6.4 Applied Numerical Optimization

The remaining sections in this chapter focus on the application of numerical optimization to the lunar TEI problem. Direct multiple shooting is utilized as the optimization technique of choice, and VF13AD is the NLP used for locating the optimal solution. Beginning with the parameterization of the problem, the solution process is discussed in detail. The chapter concludes with application of the optimization algorithm to several cases from Chapter 4.

6.4.1 Three-Burn TEI Optimization Process

To use VF13AD, the three-burn TEI problem is represented in terms of: parameters, constraints, and a cost function. Defining these parameters composes the process that has previously been referred to as parameterization. In setting up this problem,

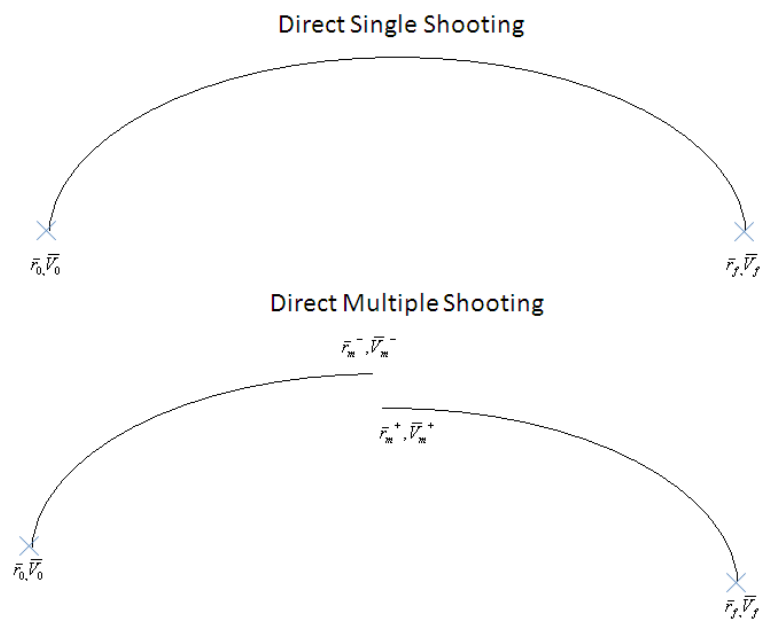


Figure 6.8: Single and Multiple Shooting Optimization Methods

only two “shots” are incorporated for the multiple shooting scheme, and one of them requires backward integration. That is, after the maneuver is applied at the lunar departure point, integration proceeds forward some specified time, TOF_1 , and then the specified Earth entry state is propagated backwards for some specified time TOF_2 . Figure 6.9 illustrates this process, where the circle at the beginning of the transfer represents the initial lunar orbit, and the end point represents the Earth entry point. Notice that the arrows switch direction across the second segment, this only implies that integration proceeds backward and not that the motion of the spacecraft itself is changing. Finally, the discontinuity in the states at the break point must be driven to zero for a feasible solution to be achieved.

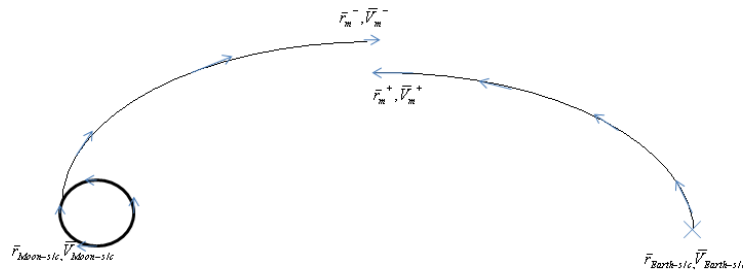


Figure 6.9: Multiple shooting applied to the lunar TEI problem

Using this approach, it is important to mention that the initial and final boundary constraints of the TPBVP are already enforced and do not need to be imposed as constraints within the NLP. That is, since the initial and terminal states are pre-specified and used to “shoot” to some intermediate discontinuity, the TPBVP is solved when the discontinuity constraints are satisfied, by definition.

With the parameterization of this problem described at a high level, the specific cost function, parameters, and constraints associated with this transfer are

defined. Since this optimization problem is intending to minimize the propulsive sum of the three maneuvers of the TEI sequence, the cost function is

$$J = \Delta V_1 + \Delta V_2 + \Delta V_3. \quad (6.36)$$

The independent variables that are subject to change within the NLP when searching to minimize the specific cost are referred to as parameters. For this problem these are: the time of departure along the initial orbit (could be replaced by the mean anomaly), the first maneuver, the flight time between the first and second maneuvers, the second maneuver, the flight time between the second and third maneuvers, the third maneuver, the flight time from the third maneuver to the discontinuity point, the flight time from the Earth to the discontinuity point, and the Earth entry state. This is captured by the parameter vector,

$$\bar{x}_p = \begin{bmatrix} \tau_0 \\ \Delta \bar{V}_1 \\ TOF_1 \\ \Delta \bar{V}_2 \\ TOF_2 \\ \Delta \bar{V}_3 \\ TOF_3 \\ TOF_4 \\ \bar{r}_{entry} \\ \bar{V}_{entry} \end{bmatrix}_{20 \times 1}. \quad (6.37)$$

Finally, the constraints are contained in Equation 6.38 through 6.50, where the first eleven constraints are equality constraints, and the final six constraints are

inequality (satisfied when greater than or equal to zero).

$$c_{1-3} : 0 = \bar{r}_{TOF_3} - \bar{r}_{TOF_4} \quad (6.38)$$

$$c_{4-6} : 0 = \bar{V}_{TOF_3} - \bar{V}_{TOF_4} \quad (6.39)$$

$$c_7 : 0 = r_{entry} - r_{specified} \quad (6.40)$$

$$c_8 : 0 = \tau_0 + TOF_1 + TOF_2 + TOF_3 + TOF_4 - t_{target} \quad (6.41)$$

$$c_9 : 0 \geq \sin(\gamma_{entry}) - \sin(\gamma_{specified}) \quad (6.42)$$

$$c_{10} : 0 \geq \sin(\phi_{specified}) - \sin(\phi_{entry}) \quad (6.43)$$

$$c_{11} : 0 \geq \cos(\theta_{specified}) - \cos(\theta_{entry}) \quad (6.44)$$

$$c_{12} : 0 \geq r_{periapsis} - r_{minimum} \quad (6.45)$$

$$c_{13} : 0 \geq r_{maximum} - r_{apoapsis} \quad (6.46)$$

$$c_{14} : 0 \geq TOF_1 \quad (6.47)$$

$$c_{15} : 0 \geq TOF_2 \quad (6.48)$$

$$c_{16} : 0 \geq TOF_3 \quad (6.49)$$

$$c_{17} : 0 \geq TOF_4 \quad (6.50)$$

Starting with Equation 6.38, the list of requirements imposed by this constraint vector includes: continuity constraints, an altitude constraint, a time of flight constraint, a flight path angle constraint, a latitude constraint, a longitude constraint, a periapsis constraint, an apoapsis constraint, and a set of feasible time constraints. It is important to note that the number of active constraints cannot and does not exceed the number of parameters. A problem with more constraints than parameters cannot be numerically solved.

Given the cost, set of parameters, and constraints defined for this Earth-

return problem, the NLP is ready to be incorporated. The solution process for using the NLP to solve this problem is described, at a high level, by the following calculation steps:

1. Retrieve the position and velocity vectors on the initial orbit corresponding to the time, τ_0 , in the parameter vector.
2. Add $\Delta\bar{V}_1$ to the velocity vector of this state.
3. Integrate forward for the amount of time specified by the parameter TOF_1 .
4. Add $\Delta\bar{V}_2$ to the velocity vector of this state.
5. Integrate forward for the amount of time specified by the parameter TOF_2 .
6. Add $\Delta\bar{V}_3$ to the velocity vector of this state.
7. Integrate forward for the amount of time specified by the parameter TOF_3 .
8. Given the final conditions contained within the parameter vector, integrate backward for the amount of time specified by parameter TOF_4 .
9. Calculate the gradients of the cost function and the constraints with respect to the parameters in \bar{x}_p .
10. Provide the NLP with the values of the cost function, constraints, and the gradient information (tuned finite differencing in this case).
11. The NLP updates \bar{x}_p based on the search direction (gradients) and steps 1-10 are iterated upon until the cost is minimized.

Note that during the solution process, it is advantageous to scale all the parameters such that they are on the order of unity. Beyond this, it is also necessary to constrain the maximum step size of each parameter within VF13AD. This problem is

sensitive and large changes in the parameters create “chattering” in the search for the minimum, introducing convergence difficulty.

Initial Guess

With the problem properly parameterized, all that remains is to create a good initial guess for the parameter vector in Equation 6.37. A good initial guess is crucial to the success rate of convergence. The initial guess chosen here corresponds to the same as that outlined in Chapter 4. However, Level-Two correction is not incorporated across the terminal leg of the Earth-return, only the Level-One. Therefore, while the initial guess satisfies the desired position constraints, it does not satisfy the velocity. Finally, note that since the initial guess provided here is continuous in position, the position portion of the continuity constraints in Equation 6.38 is also satisfied initially.

6.4.2 Three-Burn TEI Optimization Examples

Four cases are shown to provide a set of optimal transfers that provide insight into the optimality of the targeting solutions in Chapter 4. For these examples, two types of figures are presented. The first of these figures represents the optimal transfer in the ECI frame. The second of these figures captures the transfer in the MCI frame. It is also useful to take notice of the red crosses that appear on the MCI plots. These represent the maneuver locations along the initial guess arc.

Optimization Example 1

Using the same initial conditions used for the five cases in Chapter 4, with the exception of a change to the inclination, the following entry parameters are targeted:

$$\begin{aligned}altitude &= 100 \text{ km}, \\ \theta &= 134.5456^\circ, \\ \phi &= -19.2041^\circ, \\ gamma &= -6.03^\circ,\end{aligned}\tag{6.51}$$

Constraining the flight time to match example case 1 of Chapter 4, the propulsive cost for this optimal transfer is 993.8 m/s, and the computation time was 15 minutes and 12 seconds. Note that the inequality constraints for the periapsis altitude and the apoapsis altitude in Equations 6.45 and 6.46 are pushed to their boundaries. This results in a periapsis altitude of 100 km and an apoapsis that corresponds to

a 48 hour orbit period. The parameter vector that yields these results is,

$$\bar{x}_p = \begin{bmatrix} \tau_0 \\ \Delta\tilde{V}_1 \\ TOF_1 \\ \Delta\tilde{V}_2 \\ TOF_2 \\ \Delta\tilde{V}_3 \\ TOF_3 \\ TOF_4 \\ \bar{r}_{entry} \\ \tilde{V}_{entry} \end{bmatrix} = \begin{bmatrix} 4707.699223 \text{ s} \\ 0.3178893398 \text{ km/s} \\ -0.3623749700 \text{ km/s} \\ -0.4806142515 \text{ km/s} \\ 82260.3293 \text{ s} \\ 0.2432063955D - 01 \text{ km/s} \\ -0.7464241746D - 01 \text{ km/s} \\ 0.3512764443D - 01 \text{ km/s} \\ 86002.81284 \text{ s} \\ 0.1903882892D + 00 \text{ km/s} \\ 0.8813369223D - 01 \text{ km/s} \\ -0.8701368577D - 01 \text{ km/s} \\ 16188.92969 \text{ s} \\ 347568.3540 \text{ s} \\ 2610.026500 \text{ km} \\ -5559.204778 \text{ km} \\ -2138.667586 \text{ km} \\ 9.223798159D \text{ km/s} \\ 3.966088047 \text{ km/s} \\ 4.456202428 \text{ km/s} \end{bmatrix}. \quad (6.52)$$

The initial guess and optimal solution are captured visually by the plots in Figures 6.10 and 6.11. Figure 6.11 shows that the optimal solution is similar to the well-formed initial guess. The primary difference between the initial guess and the optimal solution is the location of the departure point on the initial orbit.

An interesting characteristic of this optimal transfer that cannot be easily observed in these figures is the active altitude constraint at the moon. In searching for the optimal transfer, the outgoing departure from the moon is driven to the boundary of the lunar surface. If left unconstrained, the optimal solution would pass directly through the moon. This, of course, is not a viable solution and therefore the constraint is imposed. This is an interesting result, and it suggests that a retrograde lunar orbit might provide for more favorable geometry during the departure sequence.

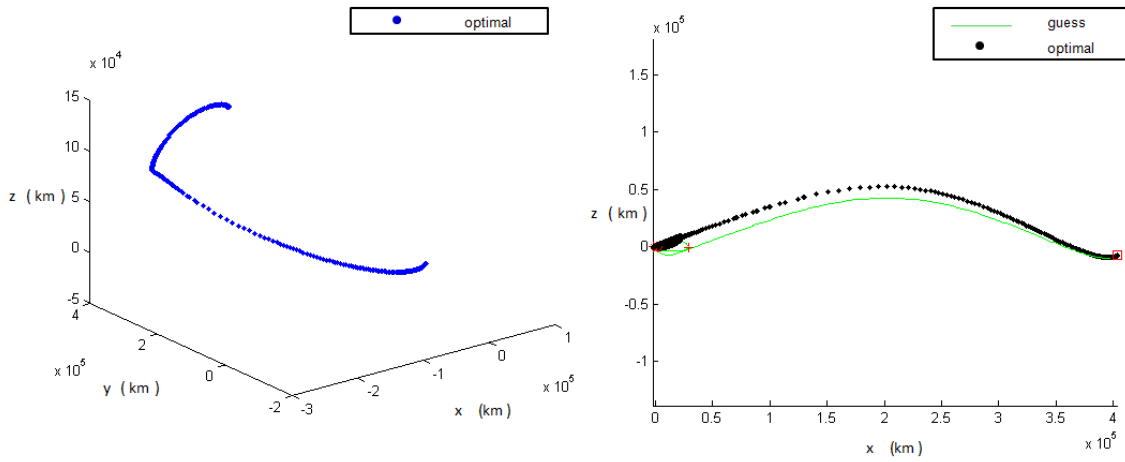


Figure 6.10: Optimization Example 1: (left) Transfer in ECI, (right) Transfer in MCI

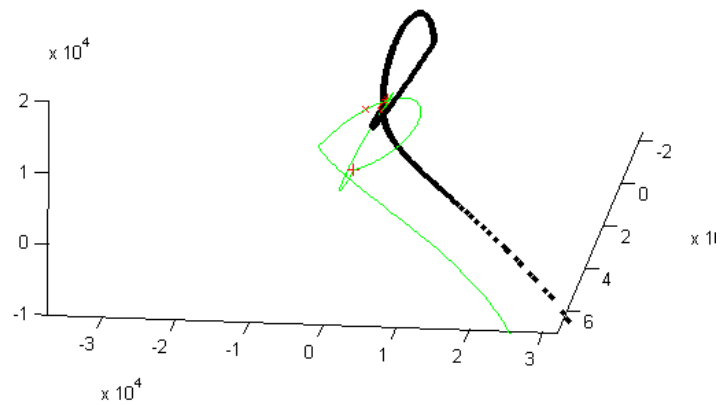


Figure 6.11: Optimization Example 1: Close View of Lunar Departure (MCI)

Optimization Example 2

In this example, the initial lunar orbit inclination is altered from the first example to match the five cases in Chapter 4. Given this minor change, the total maneuver sum of the optimal transfer is reduced to a mere 864.4 m/s, with a computation time of 14 minutes and 48 seconds. With the total ΔV reduced by nearly 130 m/s, it is concluded that the retrograde orbit allows for a more favorable transfer for this particular opportunity. In addition, with the flight time constrained to the same value used as the first optimization example, this solution represents the optimal version of the targeting solution in case 1 of Chapter 4. The parameter vector providing this result is

$$\bar{x}_p = \begin{bmatrix} \tau_0 \\ \Delta \bar{V}_1 \\ TOF_1 \\ \Delta \bar{V}_2 \\ TOF_2 \\ \Delta \bar{V}_3 \\ TOF_3 \\ TOF_4 \\ \bar{r}_{entry} \\ \bar{V}_{entry} \end{bmatrix} = \begin{bmatrix} 4038.176472 \text{ s} \\ 0.1661900723 \text{ km/s} \\ 0.5905783806 \text{ km/s} \\ -0.2312936547D - 01 \text{ km/s} \\ 97366.16025 \text{ s} \\ 0.1957442993D - 01 \text{ km/s} \\ 0.8284395627D - 02 \text{ km/s} \\ -0.1029461067D - 01 \text{ km/s} \\ 78248.79192 \text{ s} \\ 0.6961636456D - 01 \text{ km/s} \\ 0.2142923772D + 00 \text{ km/s} \\ 0.2609877661D - 01 \text{ km/s} \\ 14074.55085D \text{ s} \\ 343000.4455 \text{ s} \\ 2610.025462 \text{ km} \\ -55592.05266D \text{ km} \\ -2138.667586 \text{ km} \\ 9.333667092 \text{ km/s} \\ 4.191411468 \text{ km/s} \\ 4.005499735D \text{ km/s} \end{bmatrix}. \quad (6.53)$$

The initial guess and optimal solution for this example are illustrated in Figures 6.12 and 6.13. While the periapsis and apoapsis constraints are once again pushed to their boundaries by the intermediate transfer orbit, the outgoing departure does not violate the lunar surface proximity constraint.

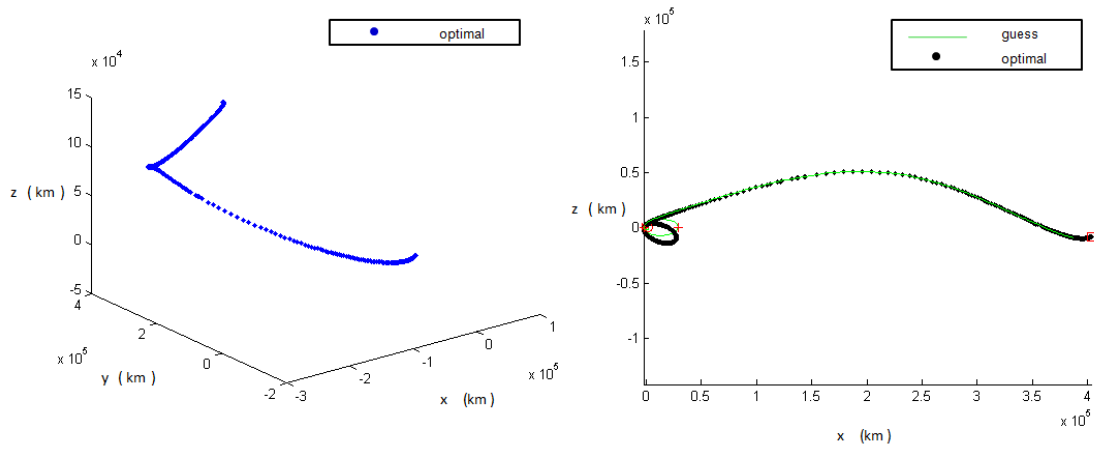


Figure 6.12: Optimization Example 2: (left) Transfer in ECI, (right) Transfer in MCI

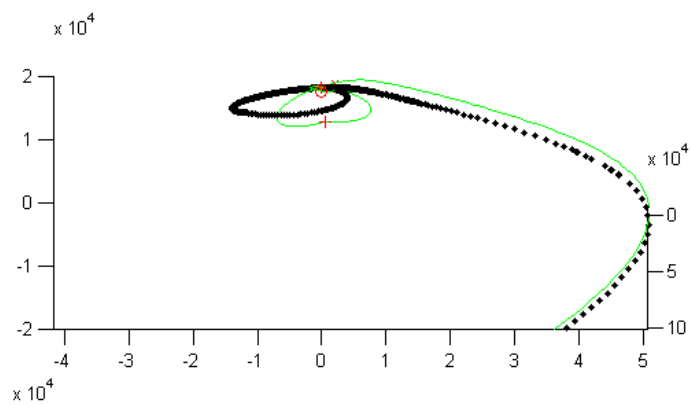


Figure 6.13: Optimization Example 2: Close View of Lunar Departure (MCI)

The loop seen in the left hand plot in Figure 6.12 is an artifact of the retrograde nature of the transfer. Since the velocity at the periapsis of the 48-hour lunar orbit is greater than that of the Moon itself (with respect to the Earth) the motion appears to circle in the ECI frame. It is also interesting to note, that TEI-3 occurs at the center of this loop, since it is performed at the periapsis.

Optimization Example 3

The goal of this example is to target an additional entry constraint. Namely, an attempt is made to impose the flight path azimuth constraint contained in Equation 4.6 such that the entry conditions are

$$\begin{aligned}
 altitude &= 100 \text{ km} \\
 10.5 &\leq V \leq 12 \text{ km/s} \\
 \theta &= 134.5456^\circ \\
 \phi &= -19.2041^\circ \\
 gamma &= -6.03^\circ \\
 Az &= 13.996^\circ
 \end{aligned}$$

The time of flight constraint is imposed to match case five of Chapter 4. By doing this, a direct comparison of the targetting solution in case five can be made to the optimal transfer achieved in this example.

The total ΔV cost associated with the converged optimal transfer is 1644.4 m/s and the computation time is 24 minutes and 37 seconds. This is nearly twice the ΔV of the second example. With the addition of the flight path azimuth constraint, along with the extended flight time, this example represents a very specific transfer.

That is, by fixing the initial orbit, the transfer time, and the Earth entry geometry, the optimization routine does not have much freedom in changing the transfer. In fact, in order to achieve the specific entry geometry, the third maneuver is pushed well beyond periapsis. The parameter vector associated with this solution is

$$\bar{x}_p = \begin{bmatrix} \tau_0 \\ \Delta \bar{V}_1 \\ TOF_1 \\ \Delta \bar{V}_2 \\ TOF_2 \\ \Delta \bar{V}_3 \\ TOF_3 \\ TOF_4 \\ \bar{r}_{entry} \\ \bar{V}_{entry} \end{bmatrix} = \begin{bmatrix} 2868.961028 \text{ s} \\ -0.3338508486 \text{ km/s} \\ 0.4940615234 \text{ km/s} \\ -0.9642915071D - 01 \text{ km/s} \\ 114373.0840 \text{ s} \\ -0.2167507346D \text{ km/s} \\ 0.1148761710D - 01 \text{ km/s} \\ 0.2320227999D + 00 \text{ km/s} \\ 109799.6903 \text{ s} \\ 0.5692856160D + 00 \text{ km/s} \\ 0.4593708169D + 00 \text{ km/s} \\ 0.1425026618D + 00 \text{ km/s} \\ 26811.67413 \text{ s} \\ 372329.7780 \text{ s} \\ 3856.566658 \text{ km} \\ -4779.528190D \text{ km} \\ -2138.667586 \text{ km} \\ 4.952854053 \text{ km/s} \\ 1.213182497 \text{ km/s} \\ 9.728632751 \text{ km/s} \end{bmatrix}. \quad (6.54)$$

The initial guess and optimal solution for this example are shown in Figures 6.14 and 6.15. In contrast to the first two optimization examples presented in this section, the converged solution is considerably different than the initial guess. Figure 6.15 provides a close view of the third maneuver displacement from periapsis.

Optimization Example 4

The final investigation here is motivated by the result in example 3 as well as the targeting result in case five of Chapter 4. Since the targeting solution provides a better transfer, in terms of propulsion, an optimal 4-burn transfer solution becomes intriguing. This investigation, however, reveals that example 3 is merely a local

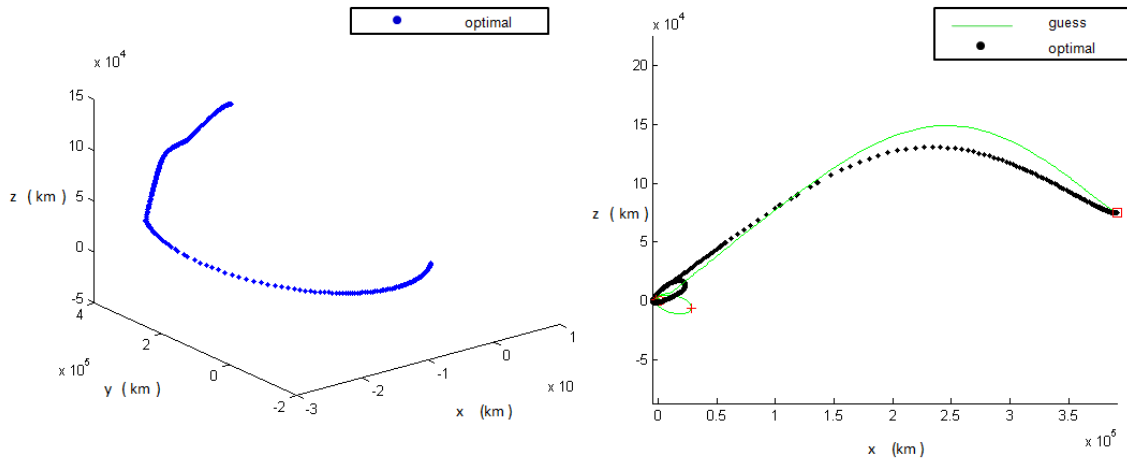


Figure 6.14: Optimization Example 3: (left) Transfer in ECI, (right) Transfer in MCI

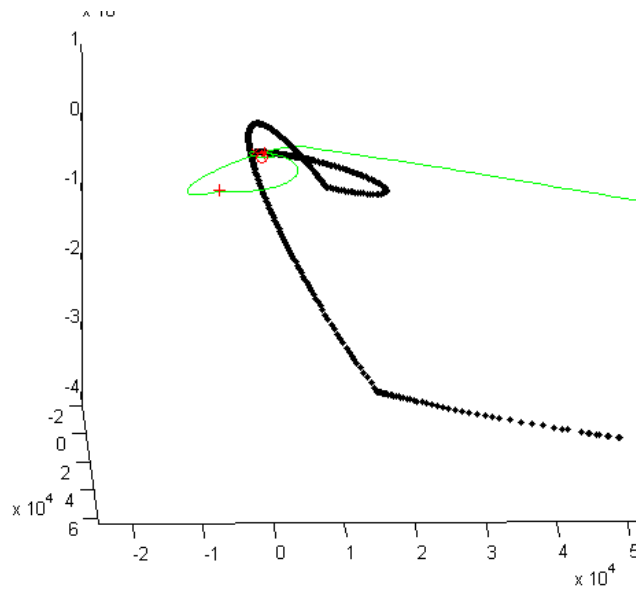


Figure 6.15: Optimization Example 3: Close View of Lunar Departure (MCI)

minimum and that an improved 3-burn transfer exists elsewhere in the solution space. In particular, while searching for the optimal 4-burn solution, the optimizer found a 3-burn transfer that possesses a total cost of 979.9 m/s. The associated computation time with this solution is 22 minutes and 11 seconds. The parameter vector yielding this solution is

$$\bar{x}_p = \begin{bmatrix} \tau_0 \\ \Delta \bar{V}_1 \\ TOF_1 \\ \Delta \bar{V}_2 \\ TOF_2 \\ \Delta \bar{V}_3 \\ TOF_3 \\ TOF_4 \\ \bar{r}_{entry} \\ \bar{V}_{entry} \end{bmatrix} = \begin{bmatrix} 3488.185047 \text{ s} \\ -0.1241954614D - 02 \text{ km/s} \\ 0.4016601027D + 00 \text{ km/s} \\ -0.8625754356D - 01 \text{ km/s} \\ 195371.93284000 \text{ s} \\ 0.2229732349D - 01 \text{ km/s} \\ 0.4016673021D + 00 \text{ km/s} \\ -0.8916254821D - 01 \text{ km/s} \\ 237985.6479 \text{ s} \\ 0.1445962955D - 01 \text{ km/s} \\ 0.6168768813D - 01 \text{ km/s} \\ 0.1437265194D + 00 \text{ km/s} \\ 0.9795531082D - 30 \text{ s} \\ 189337.4217 \text{ s} \\ 3856.566714 \text{ km} \\ -4779.528190D \text{ km} \\ -2138.667586 \text{ km} \\ 4.9515023 \text{ km/s} \\ 1.212851516 \text{ km/s} \\ 9.725978118 \text{ km/s} \end{bmatrix}. \quad (6.55)$$

The second maneuver placed at apoapsis for the plane change is zero in this solution, thus affording a 3-burn transfer from the 4-burn search. Therefore, the burn scenario for this unique solution: 1) raises apoapsis and creates an intermediate transfer orbit, 2) departs from periapsis, and 3) performs a midcourse maneuver to complete the necessary plane change for targeting the Earth entry state.

The initial guess and optimal solution for this example are included in Figures 6.16 and 6.17. In this solution, the intermediate transfer is smaller than the other examples. Also, rather than immediately departing from the periapsis, the spacecraft remains in the intermediate orbit for several revolutions. This is an arti-

fact of the large flight time constraint imposed, and is evident in the left-hand plot of Figure 6.16. The right-hand plot of Figure 6.17 then does well in providing a view of the plane change involved in the third maneuver.

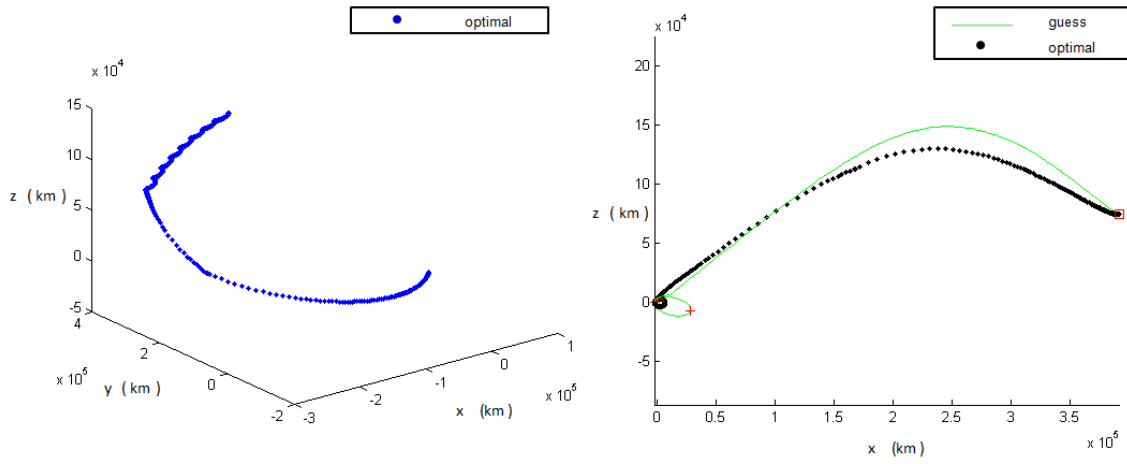


Figure 6.16: Optimization Example 4: (left) Transfer in ECI, (right) Transfer in MCI

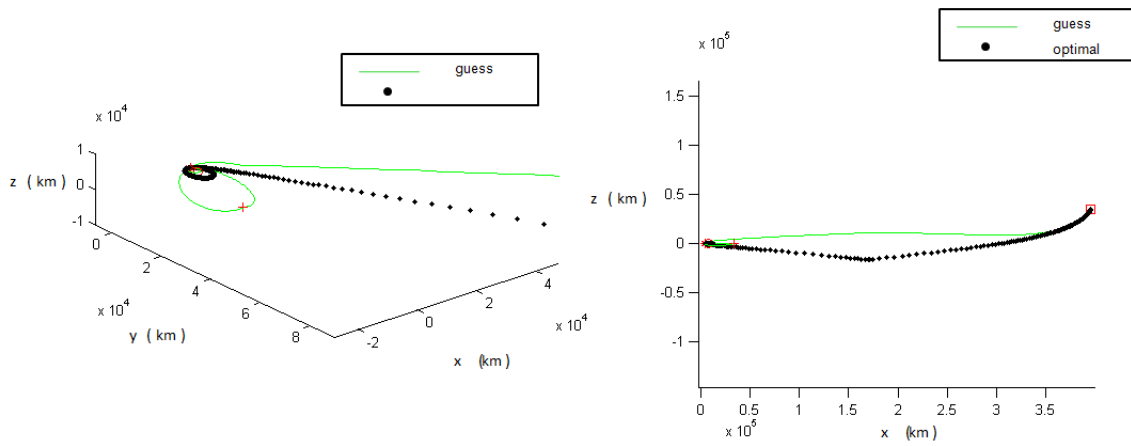


Figure 6.17: Optimization Example 4: Close View of Lunar Departure (MCI)

Chapter 7

TLC vs. Optimal

With the optimization results presented in Chapter 6, the optimality of the targeting solutions in Chapter 4 is evaluated. A great measure of success for the Earth-return TLC guidance algorithm is the propulsive optimality of its solutions.

Example 2 in Chapter 6 presents the optimal three-burn TEI transfer subject to the altitude, latitude, longitude, and flight path angle entry constraints. This transfer requires approximately 860 m/s of impulsive ΔV . The guidance algorithm provides a solution satisfying these same constraints with only 0.9 km/s. This shows that through creating a good initial guess, a near optimal solution is achievable with the TLC process. To illustrate the extent to which the targeter's solution approaches the optimal, Figures 7.1 and 7.2 show a superimposition of their solutions in the ECI and MCI frames. With the two transfers being nearly identical, the only real disparity that remains is a function of the departure point on the initial orbit.

It is also of interest to compare the computation times for these two solutions. One of the primary goals mentioned at the beginning of this investigation is the minimization of computational effort required by the onboard guidance algo-

rithm. In the case shown in Figures 7.1 and 7.2, the optimal solution required eight minutes and 19 seconds more than the targeter. Furthermore, it is important to note that the optimization is performed on a FORTRAN platform, while the TLC in MATLAB. As mentioned previously, experience shows that a compilable language such as FORTRAN is typically more than an order of magnitude faster than MATLAB. Given this information, the contrast in computation time is actually much larger than the observed values.

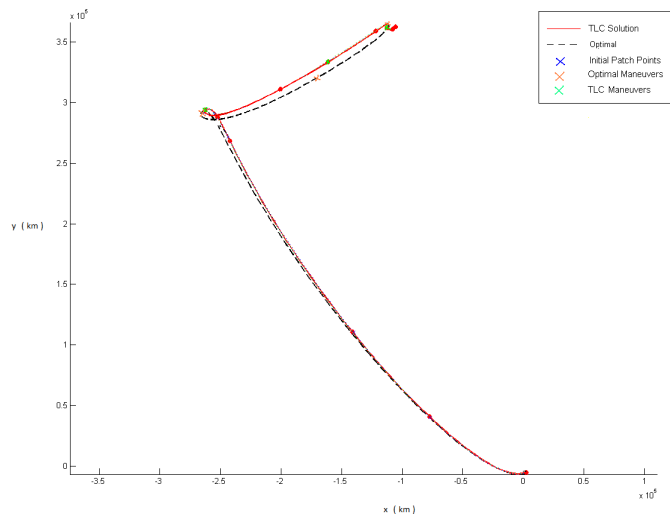


Figure 7.1: Comparison 1: Targeting vs. Optimal (ECI)

Example 4 in Chapter 6 summarizes the optimal transfer for the most challenging case addressed by this study. The targeter also addresses this problem in case 5 of Chapter 4. Figures 7.3 and 7.4 show their solutions superimposed in the ECI and MCI frames. Of course, since the optimal solution's maneuver sum is over 350 m/s less than the targeter's solution, some disparity between these solutions is expected. However, considering the difference in the burn scenarios taken by each

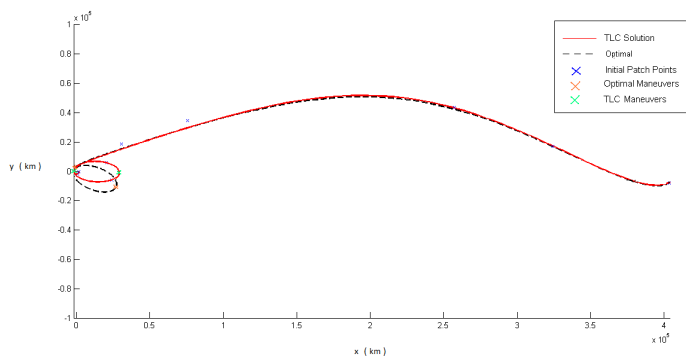


Figure 7.2: Comparison 1: Targeting vs. Optimal (MCI)

approach, their solutions are surprisingly similar. The non-uniform motion seen at the beginning of the optimal transfer illustrates that the spacecraft is still in orbit for a significant portion of the transfer time. This is an artifact of the increase in the magnitude of the flight time constraint imposed to match the targeter's result. For this example, the computation time of the optimal solution requires nine minutes and 56 seconds more than the TLC algorithm to reach a solution. As observed in this example, creating an initial guess in the vicinity of the optimum is not a straight-forward task. Moreover, the failure to create a good initial guess results in a ΔV such as the 350 m/s in this case.

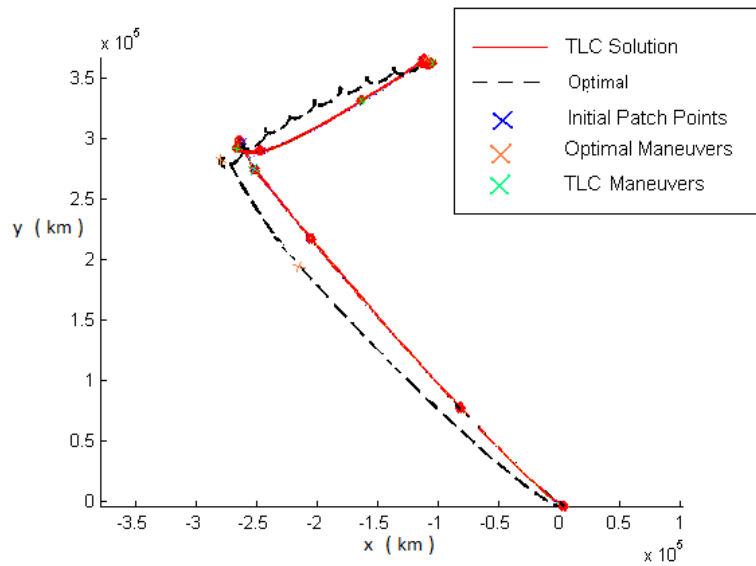


Figure 7.3: Comparison 2: Targeting vs. Optimal (ECI)

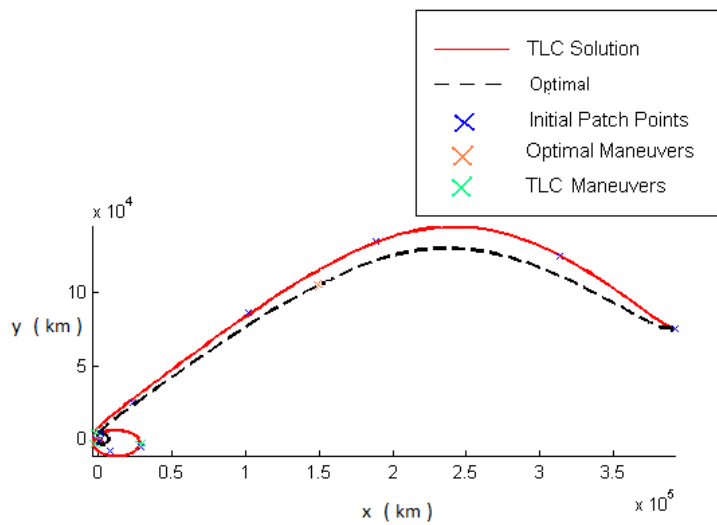


Figure 7.4: Comparison 2: Targeting vs. Optimal (MCI)

Chapter 8

Conclusions and Suggested Future Investigations

The TLC algorithm in this study demonstrates the required capabilities for an Orion onboard guidance routine. First, the entry constraints outlined in Chapter 3 provide a method for targeting a specific entry state at the Earth. Also, a maneuver sum constraint, fixed patch state constraint, and initial maneuver constraint, and simple initial guess generator offer a means for maintaining propulsive costs below the 1.4 km/s ΔV budget. Finally, the application of the TLC process itself demonstrates the ability to both robustly and quickly converge.

Beyond the impulsive demonstration of the TLC algorithm constructed for this study, a prototype for a finite thrust TLC is presented. This extension of the theory beyond an impulsive stature is a potential avenue for constructing an onboard guidance algorithm that employs more realistic thruster models. In the event of the loss of an engine, a finite thrust model is invaluable.

The completion of this study provides several avenues for future investigation.

First, the initial efforts presented on the guess generation, finite thrust, and TEI optimization leave room for extended applications and enhancements. In addition to these tasks, additional constraint formulation are required to enhance the flexibility of Earth entry targeting (geodetic latitude, downrange, crossrange, etc.). Finally, with the onboard application of the guidance algorithm requiring autonomy, the entire process needs to run without a user in the loop. Accomplishing this task involves automating the initial guess and patch point generation.

Bibliography

- [1] D’Souza, C. and Crain, T., “Orion Cislunar Guidance and Navigation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2007, Paper No. AIAA 2007-6681.
- [2] Battin, R., Hoag, D., and Craig, D., *Apollo Astronaut’s Guidance and Navigation Course Notes*, Massachusetts Institute of Technology, 1962.
- [3] Hull, D. G., *Optimal Control Theory*, Springer, 2003.
- [4] Condon, G. L., Dawn, T. F., Merriam, R. S., Sostaric, R. R., and Westhelle, C. H., “CEV Trajectory Design Considerations For Lunar Missions,” *30th Annual AAS Guidance and Control Conference*, 2007, Paper No. AAS 07-075.
- [5] Slater, G. and Stern, R., “Simplified Midcourse Guidance Techniques,” *Stepping Stones To Mars Meeting*, 2007, Paper No. AIAA-1966-2018.
- [6] Pernicka, H., *The Numerical Determination of Lissajous Orbits in the Circular Restricted Three-Body Problem*, Master’s thesis, Purdue University, 1986.
- [7] Howell, K. C. and Pernicka, H. J., “Numerical determination of Lissajous trajectories in the restricted three-body problem,” *Celestial Mechanics*, Vol. 41, 1988, pp. 107–124.

- [8] Wilson, R. S. and Howell, K. C., “Trajectory Design in the Sun-Earth-Moon System Using Lunar Gravity Assists,” *Journal of Spacecraft and Rockets*, Vol. 35, 1998, pp. 191–198.
- [9] Marchand, B., Howell, K., and Wilson, R., “Improved Corrections Process for Constrained Trajectory Design in the n-Body Problem,” *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007.
- [10] Kaplan, G. H., Josties, F. J., Angerhofer, P. E., Johnston, K. J., and Spencer, “Definition and realization of the International Celestial Reference System by VLBI Astrometry of Extragalactic Objects,” Tech. rep., IERS, 1997, IERS Technical Note 23.
- [11] Bate, R., Mueller, D., and White, J., *Fundamentals of Astrodynamics*, Dover Publications, New York, 1971.
- [12] Stewart, J., *Calculus: Early Vectors*, Brooks/Cole Pub Co., 1998.
- [13] Antsaklis, P. J. and Michel, A. N., *Calculus: Early Vectors*, McGraw-Hill, 1997.
- [14] Weinstock, R., *Calculus of Variations with Applications to Physics and Engineering*, Dover, 1974.
- [15] Bretscher, O., *Linear Algebra with Applications*, Prentice Hall, 2001.
- [16] Vallado, D., *Linear Algebra with Applications*, Microcosm Press, 2001.
- [17] Moulton, F. R., *An Introduction to Celestial Mechanics*, Dover, 1970.
- [18] Bond, V. R. and Allman, M. C., *Modern Astrodynamics: Fundamentals and Perturbation Methods*, Princeton University Press, 1996.

- [19] Curtis, H. D., *Orbital Mechanics for Engineering Students*, Elsevier, 2005.
- [20] Dormand, J. R. and Prince, P. J., “A Family of Embedded Runge-Kutta Formulae,” *Journal of Computational and Applied Mathematics*, Vol. 6, No. 1, 1980, pp. 19–26.
- [21] Weisel, W. E., *Spaceflight Dynamics*, McGraw-Hill, 1997.
- [22] Bryson, A. and Ho, Y., *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere Pub. Corp., New York, 1975.
- [23] Bertsekas, D. P., *Nonlinear Programming*, Athena Scientific, 1999.
- [24] Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Gill, 1984.
- [25] Powell, M. J. D., “Extensions to Subroutine VF02AD,” *Systems Modeling and Optimization, Lecture Notes in Control and Information Sciences*, Vol. 38, Springer-Verlag, New York, 1982, pp. 529–538.
- [26] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, 1999.
- [27] Adamo, D. R., “Apollo 13 Trajectory Reconstruction Via State Transition Matrices,” *30th Annual AAS Guidance and Control Conference*, 2007, Paper No. AAS 07-330.

Vita

Chad Wilson Smith was born in Waterloo Iowa on the 20th of November, 1981. He received his Bachelor of Science in Aerospace Engineering from Iowa State University in 2004. At that time, Chad W. Smith accepted an aerospace engineering position in the Inner Planet Mission Analysis Group at NASA's Jet Propulsion Laboratory in Pasadena, California. During his time at JPL he enhanced his engineering skills, applying mission analysis and astrodynamics to real applications. Departing JPL in the Fall of 2006 on a leave of absence, Chad began the pursuit of a graduate degree at The University of Texas at Austin. Upon accepting this M.S.E. Degree in Aerospace Engineering, Chad plans to pursue M.B.A and M.E.M. degrees at the Kellogg School of Business and the McCormick School of Engineering at Northwestern University.

Permanent Address: 7546 Wagner Road

Cedar Falls, IA 50613

This thesis was typeset with $\text{\LaTeX} 2_{\epsilon}$ ¹ by the author.

¹ $\text{\LaTeX} 2_{\epsilon}$ is an extension of \LaTeX . \LaTeX is a collection of macros for \TeX . \TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.