UNIVERSITY OF CALIFORNIA,
IRVINE

Classifications of Time-Optimal Constant-Acceleration Earth-Mars Transfers

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTERS OF SCIENCE

in Mechanical and Aerospace Engineering

by

Jesse A. Campbell

Thesis Committee:
Professor Kenneth D. Mease, Chair
Professor Faryar Jabbari
Professor Tammy Smecker-Hane

2014

DEDICATION

To
my family, whose unwavering love and support has guided me through uncertain times
my friends, whose unfailing encouragement inspires me to ever greater heights
and to anyone else whose personal voyage it is to explore the cosmos.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor Benjamin Villac, for offering his guidance and support throughout my graduate career. Without him this thesis and all that is sure to follow would not be possible.

I would also like to thank everyone in the Flight Dynamics and Controls Laboratory at UC Irvine for offering their help and camaraderie over the years, and for constantly inspiring me to better myself everyday.

And thank you Don, for providing the spark of inspiration.

Partial support of this thesis comes from the National Aeronautics and Space Administration (NASA) Astrodynamics Reach Grant, In-Space Propulsion Technology Development program.

ABSTRACT OF THE THESIS

Classifications of Time-Optimal Constant-Acceleration Earth-Mars Transfers

By

Jesse A. Campbell

Masters of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2014

Professor Kenneth D. Mease, Chair

A parametric study of Earth-Mars time-optimal constant-acceleration transfers is presented. Various continuation methods are introduced to efficiently compute large numbers of transfers for compilation into a database. Once computed, transfers are subsequently classified by observed shared characteristics, and the relationship between classifications and properties of the optimal control problem is explored.

Chapter 1

Introduction

Low thrust propulsion systems have been around for many decades[14] and much research [10, 12, 5, 16, 15, 9] has been conducted into how to design trajectories for missions incorporating low-thrust propulsion systems, including time-optimal objectives. In general, these low-thrust systems are characterized by high fuel efficiency (or high specific impulse) and continuous long-term thrusting sequences, which makes them ideal for missions to distant parts of the solar system. However, this benefit comes with unique challenges since low-thrust systems do not benefit from the impulse assumption applicable to high-thrust systems. Fortunately, newer developments such as VASIMR[1] (currently in development) and NSTAR (currently aboard NASA's Dawn mission[3]) have illustrated that low-thrust electric propulsion systems have improved considerably in efficiency and thrusting power since the first tests of electric propulsion were conducted (as with SERT 2[11, 4]). Should these trends continue, the future is sure to promise unique opportunities for trajectory designers wanting to incorporate continuous propulsion into their missions.

In an effort to complement previous work investigating minimized flight times for low thrust propulsion[10, 12, 5, 16, 15, 9], this paper presents an initial parametric study of time-

optimal trajectories followed by spacecraft with continuous-thrust capabilities between two arbitrary orbits. The ultimate goal is that of providing a searchable database of trajectories appropriate for use as first initial guesses in the design of other continuous-thrust missions.

1.1 Problem Constraints

In the most general formulation of this problem, a database of a dimension equal to the number of degrees of freedom relevant to the problem being investigated. The complexity of this database therefore increases geometrically with more degrees of freedom and a higher resolution (number of sampled nodes) in each dimension. For example, in three-dimensions the initial and final orbits introduce at minimum twelve degrees of freedom, quickly making this approach intractable for even a small number of sample points in each dimension. Additional degrees of freedom, such as propulsion system characteristics, spacecraft mass, and the inclusion of planetary fly by's, serve to increase complexity even further. In order to fix this problem of intractability, a number of assumptions governing dynamics and spacecraft characteristics are introduced in this investigation so as to reduce the number of variables to two: $\Delta\theta_f$ (final difference in true anomaly) and A (acceleration magnitude).

1.1.1 Coplanar Circular Concentric Orbits

The first simplification assumes the initial and final orbits are coplanar, circular and concentric. This fixes nine of the twelve degrees of freedom so that now only three remain: initial and final radii and final difference in true anomaly. If we assert that the two orbits correspond with those of the Earth and Mars, then the radii are fixed and only one spatial dimension needs to be varied. While quite a simplification, many realistic problems can still be modeled to first order by orbits of this variety. And while the ratio between initial and

final orbital radii, as well as the magnitude of the primary body's gravitational strength ($\mu = GM_{\odot}$) will vary from problem to problem, many orbit raising maneuvers can still benefit by an investigation into this type of generic framework. This is similar to how one can get a general sense of how all Hohmann transfers operate by looking at the properties of only Earth-Mars Hohmann transfers; while not an exhaustive approach, these types of transfers may still highlight important aspects shared by all such continuous-thrust transfers.

1.1.2 Constant Acceleration

One method of reducing the number of degrees of freedom associated with the spacecraft and its propulsion system is to make the assumption of constant accelerating thrust. That is, assert that

$$\frac{T(t)}{m(t)} \equiv A \tag{1.1}$$

where $A > 0$ is constant over the entire trajectory. If we also assume that the I_{sp} of the spacecraft is constant, then thrust is described as follows

$$T(t) = -\dot{m}(t) I_{sp} g_0 \tag{1.2}$$

where g_0 is the standard gravitational acceleration at Earth's surface. Combining Equation 1.1 with Equation 1.2 and integrating through time gives an exponential expression for the mass of the spacecraft.

$$m(\Delta t) = m_0 e^{-\frac{A}{I_{sp} g_0} \Delta t} \tag{1.3}$$

If we assume $\frac{A}{I_{sp} g_0} \ll 1$ in addition to A being constant, then we get that mass (and therefore thrust) are both approximately constant over the whole transfer. For a given desired payload

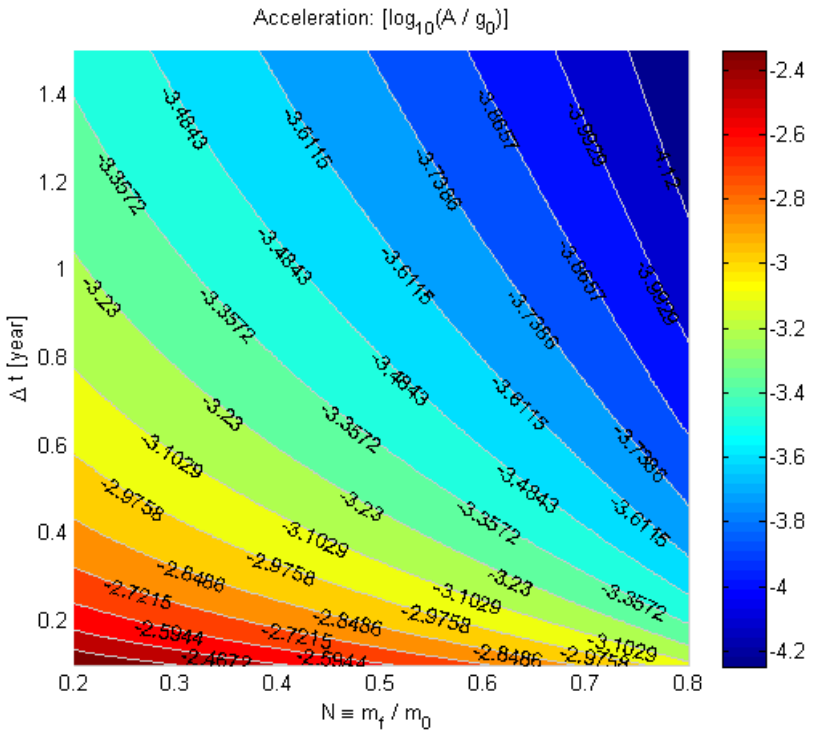


Figure 1.1: Minimum required acceleration magnitude (A) and maximum allowable transfer time (Δt) as a function of desired payload fraction ($\frac{m_f}{m_0} \in [0.2, 0.8]$) at a specified specific impulse ($I_{sp} = 12000s$).

fraction ($N \equiv \frac{m_f}{m_0}$), Figure 1.1 shows the minimum required acceleration (A) and maximum allowable transfer time (Δt) such that the final payload fraction stays above N for a given I_{sp} (in the case of Figure 1.1, $I_{sp} = 12000sec$, the upper limit of VASIMR's capabilities [1]).

1.1.3 Patched Conics Assumption

In a patched conics approximation, it is assumed that the gravitational force of the Earth and Mars are negligible throughout the entire transfer. This is justified by assuming that the spacecraft begins and ends its journey at the outermost boundaries of the relevant body's sphere of influence (6.18×10^{-3} AU for Earth and 3.86×10^{-3} AU for Mars). Since these

spheres of influence are relatively small in comparison to the overall transfer (approximately less than one percent of the total distance traveled in a worst case scenario), the spacecraft can be assumed to begin and end its motion at the presumed center of mass of the Earth and Mars, leaving and entering at parabolic escape and approach trajectories.

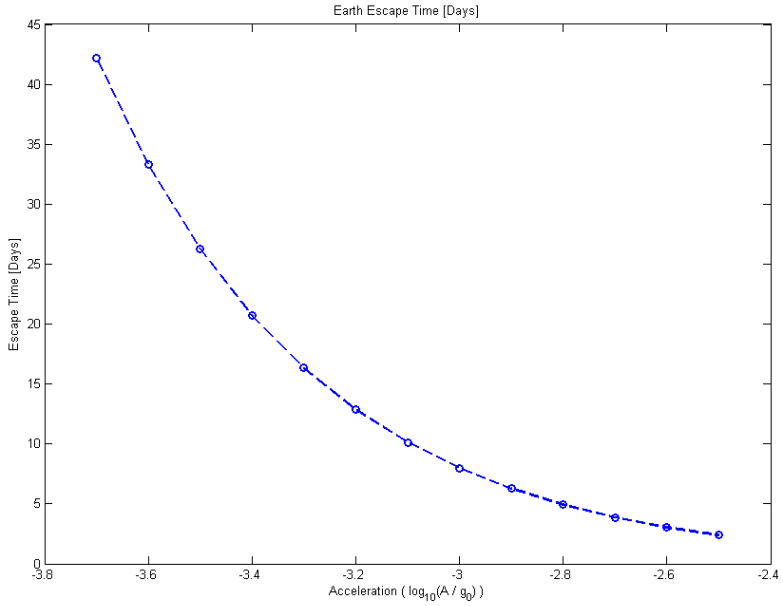
Earth escape and Mars capture times can be calculated for each value of acceleration considered in this database and added afterwards to the total transfer time before any optimized trajectories are found. For Earth escape, because the optimal method of increasing the orbital energy of a spacecraft with continuous thrust is to assume that the direction of thrust points along the spacecraft's tangential velocity, time to escape is found by integrating forward in time the following system dynamics of Equation 1.4 until $\|\dot{\mathbf{r}}(t_{esc})\| = \sqrt{\mu/R(t_{esc})}$, where t_{esc} is the time of escape (see Figure 1.2a).

$$\ddot{\mathbf{r}}(t) + \frac{\mu}{r(t)^3}\mathbf{r}(t) = A\frac{\dot{\mathbf{r}}(t)}{\|\dot{\mathbf{r}}(t)\|} \tag{1.4}$$

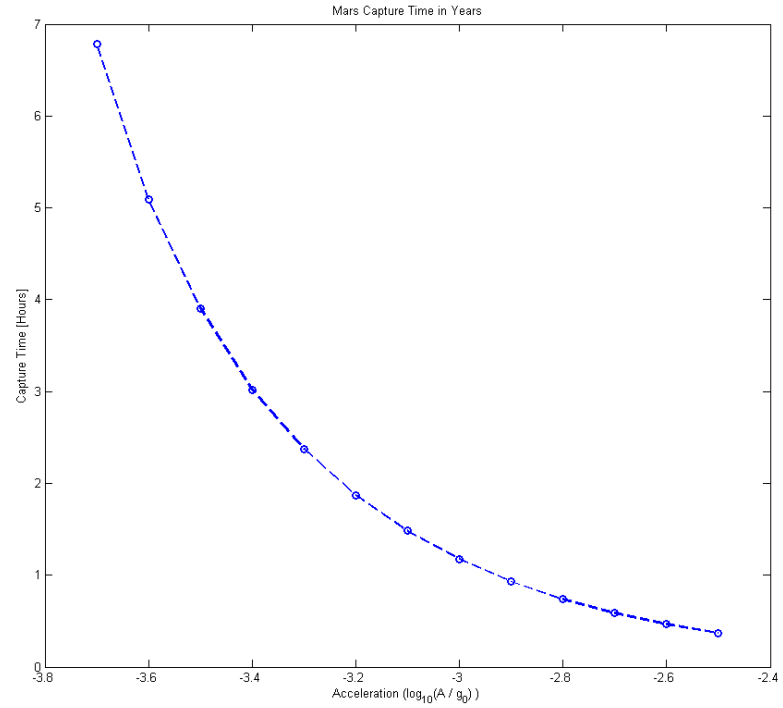
For Mars capture, the spacecraft is assumed to start at zero velocity relative to Mars at a distance equal to Mars' sphere of influence. The time for capture can then be found by using an acceleration tangential to velocity (the same dynamics as Equation 1.4) until the periapsis of the resulting orbit matches the target distance away from Mars.[1]. If one calculates such trajectories for a target periapsis equal to twice Mars' radius, the results can be seen in Figure 1.2b.

Now only two degrees of freedom remain to vary in computing a database for Earth-Mars transfers: the final relative true anomaly ($\Delta\theta_f$) and acceleration magnitude (A). This makes the problem tractable, although the process of computing even a small database is time consuming without stream-lining the process of trajectory optimization, as will be seen in the next chapter.

[1] At $t = 0$, since the spacecraft is assumed stationary, assume that the direction of thrust is pointing perpendicular with respect to a line drawn between the spacecraft and the center of mass of Mars (in either direction)



(a) Earth escape time (in days) as a function of acceleration (logarithmic).



(b) Mars capture time (in hours) as a function of acceleration (logarithmic).

Figure 1.2: Earth escape and Mars capture as functions of acceleration.

Chapter 2

Problem Formulation and Procedure

To describe the system dynamics two-dimensional polar coordinates are used, where $R(t)$ represents the radius and $\theta(t)$ the angle of the spacecraft's position (see Figure 2.1). The angle that the spacecraft points its thrusters is defined with respect to its local polar coordinate system and is denoted by $u(t)$ (see Figure 2.2). These choices were made because, as will be seen in Chapter 3, different solutions are found when the final Mars angle $\theta(t_f)$ differs by integer increments of 2π.

Additionally, the spacecraft is assumed to have an initial velocity that results in a parabolic escape trajectory from the Earth parallel to the Earth's circular motion about the Sun. In other words, the spacecraft starts in a circular orbit about the Sun at the same distance as the Earth's semimajor axis (1.0 AU). The final position of the spacecraft is uniquely determined by its final relative true anomaly ($\Delta\theta_f \equiv \theta(t_f) - \theta(t_0)$), and the velocity is likewise constrained to keep the resulting orbit circular at the semimajor axis of Mars (1.5 AU).

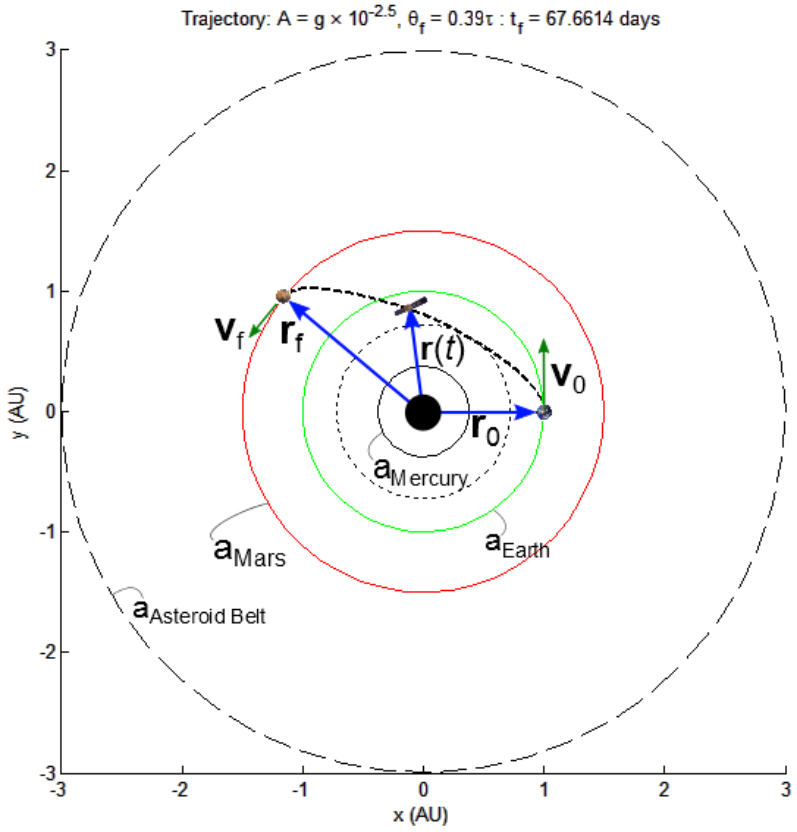


Figure 2.1: Boundary conditions fixed at initial times ($\mathbf{r}_0$, $\mathbf{v}_0$) and final times ($\mathbf{r}_f$, $\mathbf{v}_f$).

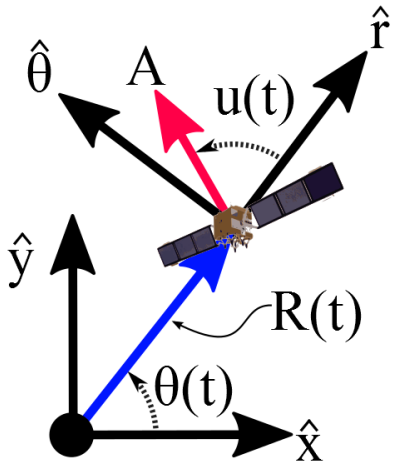


Figure 2.2: $u(t)$ is the angle with respect to local coordinates of the acceleration vector

This fixing of the initial and final states results in a two-point boundary value problem

$$\ddot{\mathbf{r}}(t) + \frac{\mu}{r(t)^3}\mathbf{r}(t) = A\hat{\mathbf{u}}(t) \tag{2.1}$$

with initial and final conditions on position and velocity

$$\begin{cases} \mathbf{r}(t_0) & = \mathbf{r}_0, \quad \mathbf{r}(t_f) & = \mathbf{r}_f \\ \dot{\mathbf{r}}(t_0) & = \mathbf{v}_0, \quad \dot{\mathbf{r}}(t_f) & = \mathbf{v}_f \end{cases} \tag{2.2}$$

where μ is the gravitational constant of the Sun, $\mathbf{r}(t) = R(t)(cos\theta(t), sin\theta(t))^\intercal$ is the position of the spacecraft, $\ddot{\mathbf{r}}(t)$ is the second time derivative of $\mathbf{r}(t)$, A is the constant magnitude of acceleration, $\hat{\mathbf{u}}(t) = (cos(u(t)), sin(u(t)))^\intercal$ is the unit vector specifying the direction of acceleration, and the vectors $\mathbf{r}_0$, $\mathbf{r}_f$, $\mathbf{v}_0$, and $\mathbf{v}_f$ are constant vectors specifying the initial and final positions and initial and final velocities of the spacecraft, respectively.

2.1 Optimal Control Formulation

In order to solve this problem for the optimal thrusting angle, $u(t)$, the methods of optimal control are necessary. Let the state vector be defined such that $\mathbf{x}(t) \equiv [R(t), \theta(t), V(t), \Omega(t)]^\intercal$, where $V(t)$ and $\Omega(t)$ are the time dependent radial velocity and angular velocity of the spacecraft, respectively. The control vector naturally becomes just the thrusting angle, $u(t)$,

and the cost to minimize is $\mathcal{J} = t_f - t_0$, over which the optimal control is found such that

$$\begin{pmatrix} \dot{R}(t) \\ \dot{\theta}(t) \\ \dot{V}(t) \\ \dot{\Omega}(t) \end{pmatrix} = \begin{pmatrix} V(t) \\ \Omega(t) \\ Acos(u(t)) + R(t)\Omega(t)^2 - \frac{\mu}{R(t)^2} \\ \frac{1}{R(t)}(Asin(u(t)) - 2V(t)\Omega(t)) \end{pmatrix} \tag{2.3a}$$

$$\begin{pmatrix} a_{Mercury} \\ -0.01c \\ \frac{-0.01c}{a_{Mercury}} \end{pmatrix} \leq \begin{pmatrix} R(t) \\ V(t) \\ \Omega(t) \end{pmatrix} \leq \begin{pmatrix} a_{AsteroidBelt} \\ +0.01c \\ \frac{+0.01c}{a_{Mercury}} \end{pmatrix} \tag{2.3b}$$

$$\begin{pmatrix} R(t_0) \\ \theta(t_0) \\ V(t_0) \\ \Omega(t_0) \end{pmatrix} = \begin{pmatrix} a_{Earth} \\ 0 \\ 0 \\ \sqrt{\frac{\mu}{a_{Earth}^3}} \end{pmatrix} \tag{2.3c}$$

$$\begin{pmatrix} R(t_f) \\ \theta(t_f) \\ V(t_f) \\ \Omega(t_f) \end{pmatrix} = \begin{pmatrix} a_{Mars} \\ \Delta\theta_f \\ 0 \\ \sqrt{\frac{\mu}{a_{Mars}^3}} \end{pmatrix} \tag{2.3d}$$

$$0 \leq \sqrt{V(t)^2 + R(t)^2\Omega(t)^2} \leq 0.01c \tag{2.3e}$$

where $a_{Mercury}$ is the semimajor axis of the orbit of Mercury ($0.375AU$) and $a_{AsteroidBelt}$ is the semimajor axis of the asteroid belt ($3.0AU$).

If gravity were to be ignored completely, then there would exist an analytic solution[6] for all possible initial and final boundary conditions. However, since $\mu \neq 0$, numerical methods must be used to find solutions. The specific numerical solver chosen for this step is GPOPS-II[13],

a Matlab software package implementing Legendre-Gauss-Radau pseudospectral collocation, a direct method of solution.[1]

However, useful results can still be obtained through analytic investigation. Using standard optimal control, the Hamiltonian can be written as follows,[2] assuming the costates are denoted by $\boldsymbol{\lambda}^\intercal(t) = [\lambda_R(t), \lambda_\theta(t), \lambda_V(t), \lambda_\Omega(t)]$.

$$\mathcal{H} = 1 + \boldsymbol{\lambda}^\intercal \dot{\mathbf{x}} = 1 + \lambda_R V + \lambda_\theta \Omega + \lambda_V (R\Omega^2 - \frac{\mu}{R^2} + A cos(u)) + \lambda_\Omega \frac{1}{R}(-2V\Omega + A sin(u)) \quad (2.4)$$

The first order necessary conditions then become, since this problem is time-optimal and the control $u(t)$ is not *explicitly* bounded

$$\mathcal{H}(t) = 0 \quad (2.5)$$

$$tan(u(t)) = \frac{1}{R(t)} \frac{\lambda_\Omega(t)}{\lambda_V(t)} \quad (2.6)$$

where the first equation is a result of the fact that $\mathcal{H}(t_f) = 0$ and that $\mathcal{H}(t)$ is constant for time-optimal problems. The second equation comes from the minimum principle, $\nabla_{u(t)} \mathcal{H}(t) = 0$. In this form the costates, $\lambda_V(t)$ and $\lambda_\Omega(t)$, are solutions to nonlinear ODEs and thus have no closed-form solution in general, so we cannot proceed any further without knowing what the costates are beforehand.

What these two conditions provide now is a method of checking for the validity of a solution. If these conditions are violated, then the solution cannot possibly be a (local) minimum of the Hamiltonian, and thus not a (local) solution to the optimal control problem. However, as these are only necessary and not sufficient conditions, this doesn't guarantee in any way that any solutions which do *not* violate these constraints are *actually* (local) solutions to the problem.

[1] GPOPS-II allows its users to specify which underlying nonlinear programming problem solver to use, the two choices being SNOPT[8] and IPOPT[2]. SNOPT was chosen for this problem.

[2] The dependence on time of each variable is implied and has been left out for clarity.

The only issue left to address now is that of determining good initial guesses for the numerical solver. Since this problem involves generating a database of trajectories whose boundary conditions are "nearby" each other, a continuation method would serve well to speed up the building process while efficiently reusing work already done to find previous solutions. The next section will review a few such continuation methods.

2.2 Initial Guess Generation Methods

2.2.1 Analytic Solution for $\mu \equiv 0$

For different steps in the database generation process different types of methods are required in order to find further solutions. In the very first step, no previous solutions can be called upon, so the first trajectory we try to find is one that has a high enough acceleration such that the analytic solution[6] for the $\mu \equiv 0$ case provides a good approximation. If we assume that $A \gg \frac{\mu}{R^2}$ over an entire trajectory, then we can use such a solution as guesses for the optimal control solver. The general solution results in a bilinear tangent law for the control angle of the following form

$$tan(u(t)) = \frac{C_1 t + C_2}{C_3 t + C_4} \tag{2.7}$$

where C_1, C_2, C_3, and C_4 are constants in time that can be solved for numerically through equations given in [6].[3] However, if we choose to find a solution for a trajectory where $\Delta\theta_f$ is chosen in such a way as to minimize the velocity normal to the straight-line drawn between the initial and final positions, then we can assume the problem is approximately

[3]One should note that Equation 2.6 is just a more general form of the bilinear tangent law formulated in polar coordinates, where the costates are not merely linear functions of t in general, a consequence of the inclusion of an inverse-square force.

one-dimensional, and the following control law applies

$$u(t) = \begin{cases} 0 & t < t_S \\ \pi & t > t_S \end{cases} \tag{2.8}$$

where t_S is a characteristic "switching time" which in this case becomes $t_S = 2\sqrt{\frac{a_{Mars} - a_{Earth}}{A}}$ for a given acceleration magnitude, A. The specific case in which the transverse velocity is minimized is when the initial and final velocities projected onto the vector joining the initial and final positions in each orbit, when dotted together, are at a maximum value. That is

$$\max \mathbf{v}_0^p \cdot \mathbf{v}_f^p \tag{2.9}$$

$$\mathbf{v}_0^p = \frac{\mathbf{v}_0 \cdot \Delta\mathbf{r}}{\Delta\mathbf{r} \cdot \Delta\mathbf{r}} \Delta\mathbf{r} \tag{2.10}$$

$$\mathbf{v}_f^p = \frac{\mathbf{v}_f \cdot \Delta\mathbf{r}}{\Delta\mathbf{r} \cdot \Delta\mathbf{r}} \Delta\mathbf{r} \tag{2.11}$$

$$\Delta\mathbf{r} = \mathbf{r}_f - \mathbf{r}_0 \tag{2.12}$$

The specific angle at which this happens for the values of Earth and Mars orbits given in Equations 2.3c and 2.3d is $\Delta\theta_f \approx 0.2677\pi$. Using the one-dimensional approximation for the control law at this angle, an initial guess can be made that allows for convergence to a (locally) time-optimal trajectory, as shown in Figure 2.3. The control seen there exhibits a characteristic "bang-bang" behavior, typical of the analytic one-dimensional solution.

2.2.2 Linear Geometric Continuation (LGC)

After finding at least one optimal trajectory using the method described above, a method of computing higher quality guesses in a more automated fashion is needed. Consequently,

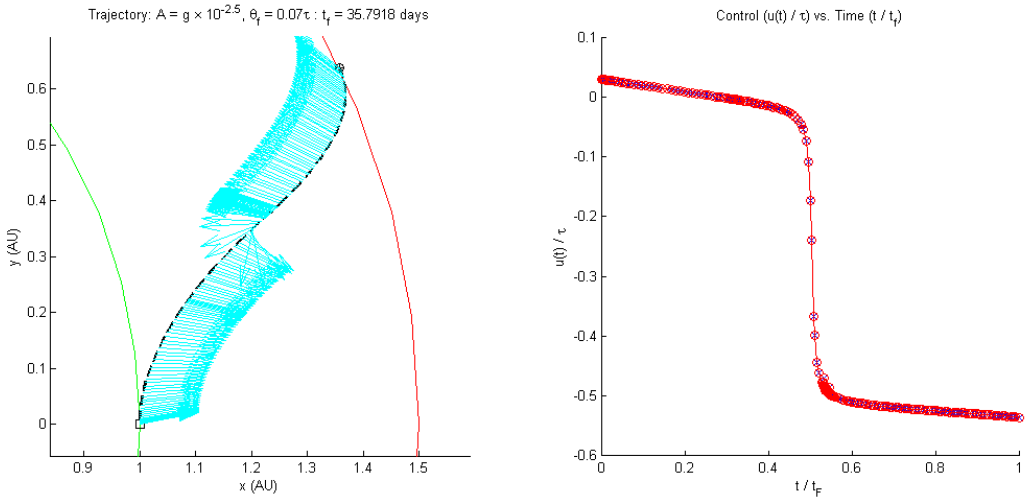


Figure 2.3: Plot of trajectory (left) and corresponding control (right) for acceleration magnitude $A = 10^{-2.5} g_0 \gg \frac{\mu}{R^2}$ over the entire trajectory.

a simple continuation method is devised, based on a linear approximation of the differences in boundary conditions between a previously computed solution and the current problem to solve, here referred to as linear geometric continuation, or LGC.

LGC works by generating a guess for the position, $\mathbf{r}_G(t)$, and acceleration angle, $u_G(t)$, by reusing a previously computed trajectory, $\mathbf{r}^O(t)$, known a priori to converge quickly, along with the final (known) boundary conditions on the state, represented by $\mathbf{r}^O(t_f)$ and $\dot{\mathbf{r}}^O(t_f)$. These boundary conditions are assumed to be close to the desired boundary conditions, $\mathbf{r}_f$ and $\mathbf{v}_f$, such that the magnitude of their relative differences is small. Since the problem of database construction requires an exhaustive iteration over a large number of closely-related trajectories, this method has the additional benefit of reusing what has already been computed, adding an element of efficiency.

LGC is described mathematically by the following set of equations, where $\mathbf{r}_G(t)$ and $\mathbf{v}_G(t)$ are the desired guess vectors for position and velocity, respectively.

$$\mathbf{r}^G(\delta t^G) = \mathbf{r}^O(\delta t^O) + R_{\delta\theta(\delta t^O)}\boldsymbol{\delta r}(\delta t) \tag{2.13a}$$

$$\mathbf{v}^G(\delta t^G) = \dot{\mathbf{r}}^O(\delta t^O) + R_{\delta\theta(\delta t^O)}\boldsymbol{\delta v}(\delta t) \tag{2.13b}$$

$$\boldsymbol{\delta r}(\delta t) = (\mathbf{r}_f - \mathbf{r}^O(t_f))\delta t \tag{2.13c}$$

$$\boldsymbol{\delta v}(\delta t) = (\mathbf{v}_f - \dot{\mathbf{r}}^O(t_f))\delta t \tag{2.13d}$$

$$R_{\delta\theta(\delta t^O)} = \begin{pmatrix} cos(\theta_f - \theta(\delta t^O)) & -sin(\theta_f - \theta(\delta t^O)) \\ sin(\theta_f - \theta(\delta t^O)) & cos(\theta_f - \theta(\delta t^O)) \end{pmatrix} \tag{2.13e}$$

$$\delta t^O = (t_f^O - t_0^O)\delta t + t_0^O \tag{2.13f}$$

$$\delta t^G = (t_f^G - t_0^G)\delta t + t_0^G \tag{2.13g}$$

$$\delta t \in [0, 1] \tag{2.13h}$$

Here, the matrix $R_{\delta\theta(\delta t)} \in \mathbb{R}^2$ is a rotation matrix designed to rotate the vector $\boldsymbol{\delta r}(\delta t)$ by an angle $\delta\theta(\delta t) = (\theta_f - \theta(\delta t))$ so that the angle between $\boldsymbol{\delta r}(\delta t)$ and the local tangential direction, $\hat{\boldsymbol{\theta}}(\delta t)$, is held constant. This idea was motivated by the observation that, for small perturbations $\boldsymbol{\delta r}(\delta t)$, the angle between successively computed trajectories was found to be approximately constant over the course of the normalized time interval, δt (see Figure 2.4). By using a normalized time interval, δt, the new time interval for $\mathbf{r}^G(t)$ can be computed easily when a guess for the final time, t_f^G, is provided. The value for t_f^G is found by computing the arc length of the original trajectory, denoted by C^O, and the newly generated guess trajectory, denoted by C^G, and simply multiplying the normalized time interval by the ratio in arc lengths and then adding the initial time from the original trajectory. Note that calculating the arclength for C^G doesn't require t_f^G since the entire trajectory is found geometrically without regard to time, and so generally this integration becomes a sum over

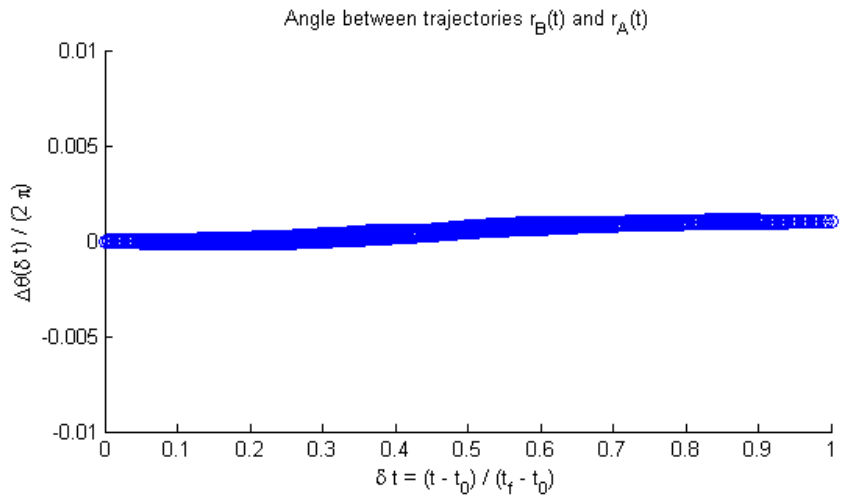


Figure 2.4: Plots for all successive pairs of trajectories within a database of trajectories with constant acceleration $A = 10^{-2.9} g_0$ and maximum mesh errors less than 2×10^{-4}. The approximate equality between the angles $\theta^A(\delta t^A)$ and $\theta^B(\delta t^B)$, namely $\Delta\theta = cos^{-1}(sin(\theta^B(\delta t^B) - \theta^A(\delta t^A)) \approx 0$, is clearly visible here.

discrete data points.

$$t_f^G = \frac{C^O}{C^G}\delta t + t_0^O \tag{2.14a}$$

$$\frac{C^O}{C^G} = \frac{\int_{t_0^O}^{t_f^O} \sqrt{(R^O\Omega^O)^2 + (V^O)^2}\, dt}{\int_0^1 \sqrt{(R^G\Omega^G)^2 + (V^G)^2}\, d\delta t} \tag{2.14b}$$

Once the vector fields $\mathbf{r}^G(t)$ and $\mathbf{v}^G(t)$ are computed, the guess for the control angle, $u^G(t)$, is computed through an application of inverse dynamics. To do this, the guessed vectors $\mathbf{r}^G(t)$ and $\mathbf{v}^G(t)$ are inserted into the left-hand side of Equation 2.1 and a control vector, $\boldsymbol{f}^G(t)$, is computed. From this new vector the angle $u^G(t)$ can be approximated in the following fashion:

$$\boldsymbol{f}^G(t) = \dot{\mathbf{v}}^G(t) + \frac{\mu}{(r^G(t))^3}\mathbf{r}^G(t) \tag{2.15a}$$

$$u^G(t) = atan2(f_y, f_x) \tag{2.15b}$$

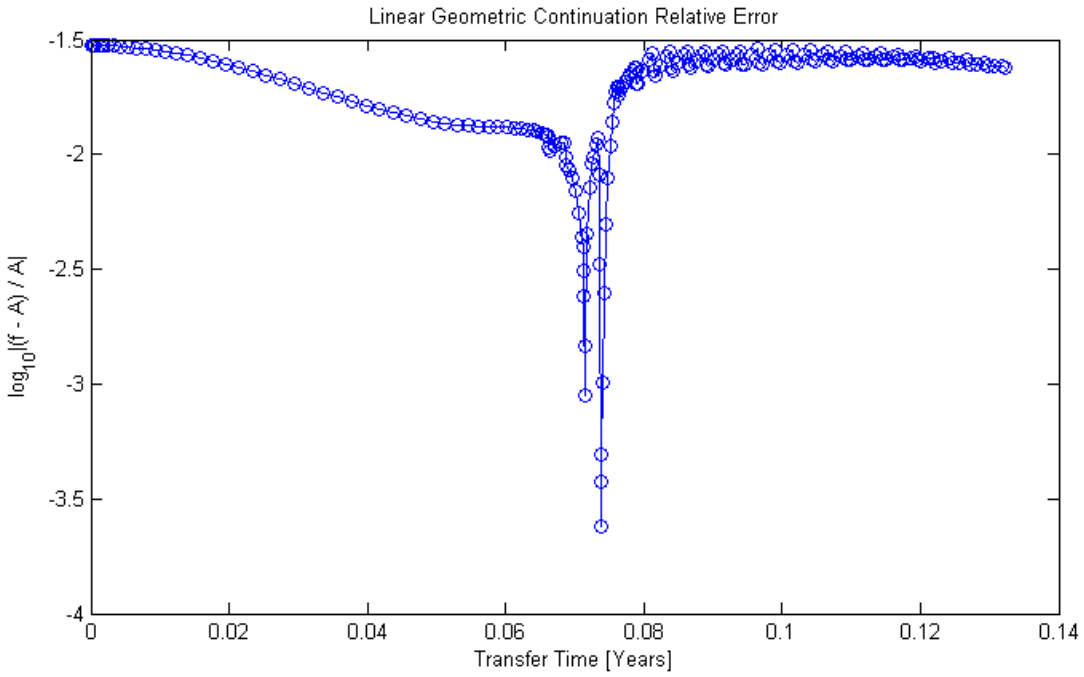


Figure 2.5: Plot of $log_{10}(||\boldsymbol{f}^G| - A|/A)$, a measure of the error incurred in the application of inverse dynamics on the trajectory produced by LGC.

where the function $atan2(f_y, f_x)$ is the inverse tangent of $\frac{f_y}{f_x}$ which takes into account the quadrants from which the components of the vector $\boldsymbol{f}^G(t)$ came.

LGC Error Analysis

In general the vector $\boldsymbol{f}^G(t)$ is not a vector with magnitude equal to the constant acceleration magnitude, A. This fact intrinsically introduces a level of error into the desired quantity, $u^G(t) = atan2(\boldsymbol{f}_y, \boldsymbol{f}_x)$. But assuming small values of $|\boldsymbol{\delta r}(t)|/|\mathbf{r}(t)|$ and $|\boldsymbol{\delta v}(t)|/|\dot{\mathbf{r}}(t)|$, this error is small if the difference in magnitude, $||\boldsymbol{f}^G(t)| - A|$, is also small. A plot of this quantity for one such trajectory can be seen in Figure 2.5. From this figure it appears that this assumption seems to hold over the entire interval.

One advantage of computing the control in this fashion is that regardless of the nature of the solution, the guess $\mathbf{r}^G(t)$ satisfies the boundary conditions of the state dynamics, thereby

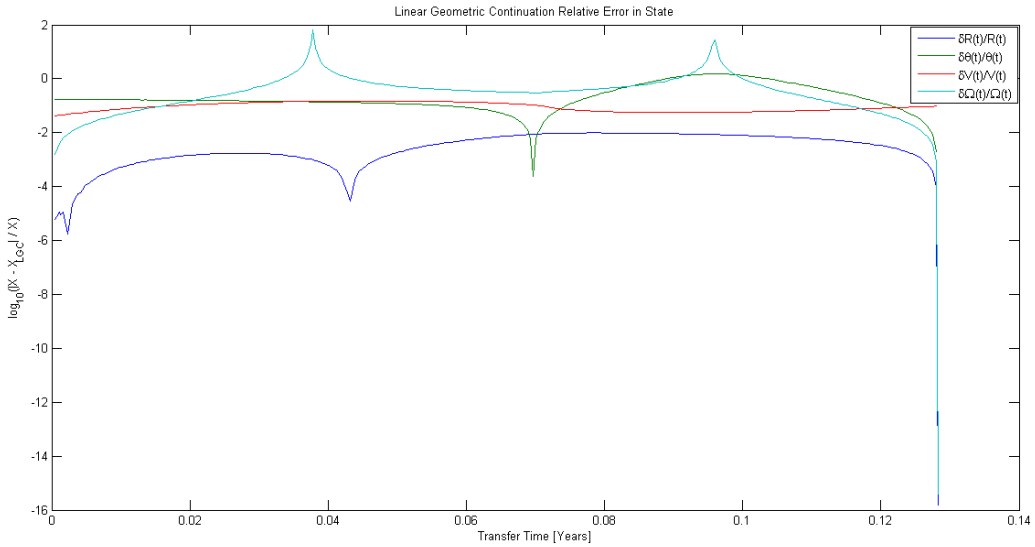


Figure 2.6: Plot of $log_{10}(|X^O - X^{LGC}|/X^O)$, the relative error between the LGC guess and the solution it was attempting to approximate.

guaranteeing that the control at least approximates a solution that also adheres to the boundary conditions. This means that for vectors $\boldsymbol{\delta r}$ and $\boldsymbol{\delta v}$ that are small in magnitude compared with $\mathbf{r}(t)$, the deviation in $u^G(t)$ ought also to be small. Additionally, the control for the original state, $\hat{\mathbf{u}}(t)$, is not used directly in generating $u^G(t)$, and instead relies on *only* the state, $[\mathbf{r}(t), \dot{\mathbf{r}}(t)]$ [4] generated by the optimal control solver to provide an approximation to the control $u(t)$.

Computing the error between an LGC guess and an actual converged solution is shown in Figure 2.6. The largest contributor to error appears to be the angular velocity, but all other values appear to stay relatively close to the target trajectory. This shows that LGC approximates the desired solution well enough for guess generation, although for trajectories that make close approaches to the Sun the increase in $\Omega(t)$ may increase error.

[4] different than using the velocity generated by the optimal control solver, $\mathbf{v}(t)$, since $\dot{\mathbf{r}}(t) \neq \mathbf{v}(t)$ exactly in general

2.2.3 Difference Continuation (DC)

When two or more solutions exist near a desired boundary condition, another method of continuation can be used, referred to here as Difference Continuation (DC). The basic idea is to take the difference between two nearby solutions and add that difference to the closer of the two solutions. Then, by scaling and rotating appropriately such that the boundary conditions match, the control can be solved for in a manner similar to LGC. Given two trajectories $\mathbf{x}^B(t)$ and $\mathbf{x}^A(t)$, where $\mathbf{x}^B(t)$ is closer to the desired final conditions, $\mathbf{x}_f$, then the guess in position and velocity ($\mathbf{r}^G(t)$ and $\mathbf{v}^G(t)$, respectively) can be calculated in the following manner.

$$\mathbf{r}^G(\delta t^G) = \mathbf{r}^B(\delta t^B) + \alpha R_{\delta\theta(\delta t^B)} \boldsymbol{\delta r}(\delta t) \tag{2.16}$$

$$\mathbf{v}^G(\delta t^G) = \dot{\mathbf{r}}^B(\delta t^B) + \beta R_{\delta\theta(\delta t^B)} \boldsymbol{\delta v}(\delta t) \tag{2.17}$$

$$\boldsymbol{\delta r}(\delta t) = (\mathbf{r}^B(\delta t^B) - \mathbf{r}^A(\delta t^A)) \tag{2.18}$$

$$\boldsymbol{\delta v}(\delta t) = (\dot{\mathbf{r}}^B(\delta t^B) - \dot{\mathbf{r}}^A(\delta t^A)) \tag{2.19}$$

$$R_{\delta\theta(\delta t^B)} = \begin{pmatrix} cos(\theta_f - \theta(\delta t^B)) & -sin(\theta_f - \theta(\delta t^B)) \\ sin(\theta_f - \theta(\delta t^B)) & cos(\theta_f - \theta(\delta t^B)) \end{pmatrix} \tag{2.20}$$

$$\alpha = \frac{\|\mathbf{r}_f - \mathbf{r}^B(t_f^B)\|}{\|\mathbf{r}^B(t_f^B) - \mathbf{r}^A(t_f^A)\|} \tag{2.21}$$

$$\beta = \frac{\|\mathbf{v}_f - \dot{\mathbf{r}}^B(t_f^B)\|}{\|\dot{\mathbf{r}}^B(t_f^B) - \dot{\mathbf{r}}^A(t_f^A)\|} \tag{2.22}$$

$$\delta t^{B/A} = (t_f^{B/A} - t_0^{B/A})\delta t + t_0^{B/A} \tag{2.23}$$

$$\delta t^G = (t_f^G - t_0^G)\delta t + t_0^G \tag{2.24}$$

$$\delta t \in [0, 1] \tag{2.25}$$

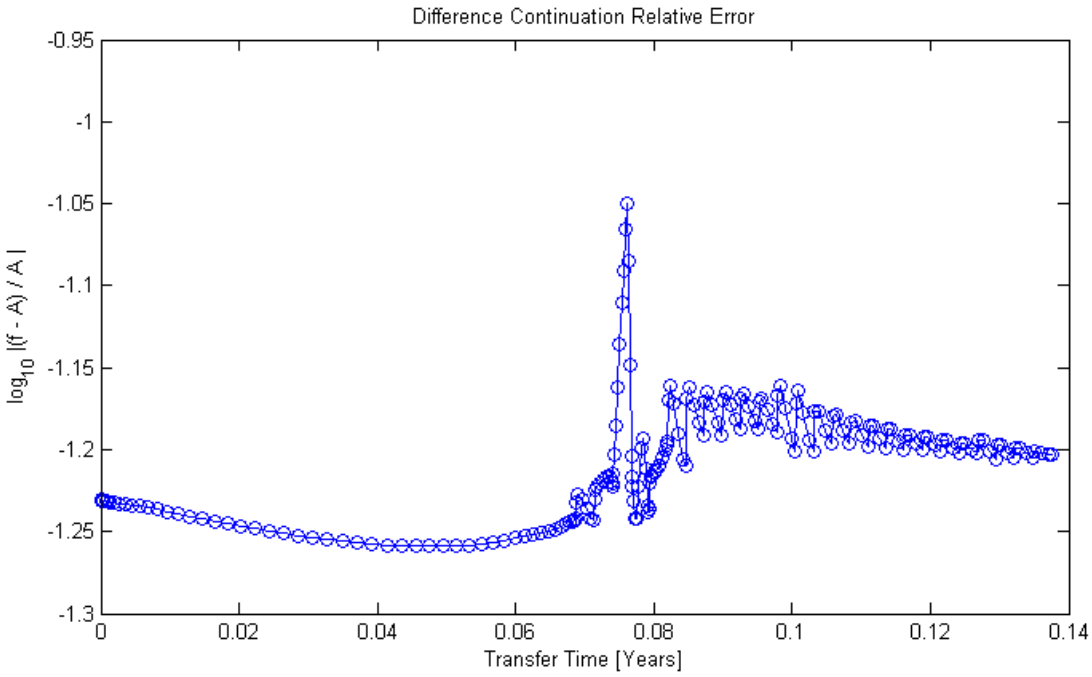


Figure 2.7: Plot of $log_{10}(||\boldsymbol{f}^G| - A|/A)$, a measure of the error incurred in the application of inverse dynamics on the trajectory produced by DC.

The primary benefit of this method is that the way in which the difference in final boundary conditions is applied to the previous solution, $r^B(t)$, is different depending on how the solution is actually changing between two solutions instead of just linearly as assumed by LGC. However, this method still suffers the same problems that LGC does, namely that if even one of the previous solutions begins to diverge then that error will be carried through with the guess, especially if the state dynamics are not being satisfied.

DC Error Analysis

As seen with LGC, DC generates the control angle $u(t)$ through an application of inverse dynamics, and so similar errors to LGC are expected. A plot of of the error in acceleration vector magnitudes for the same trajectory as in Figure 2.5 is shown in Figure 2.7. Close inspection shows that the errors are actually larger for DC than for LGC, but even so DC is seen to approximate the actual solution better in every coordinate than LGC does, as seen in

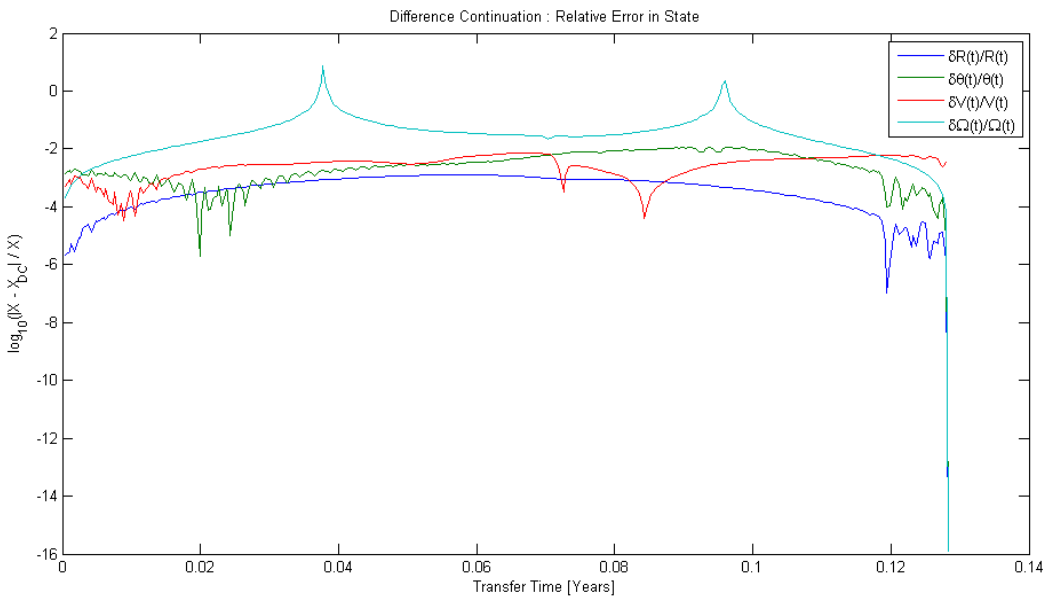


Figure 2.8: Plot of $log_{10}(|X^O - X^{DC}|/X^O)$, the relative error between the DC guess and the solution it was attempting to approximate.

Figure 2.8. But it should be noted that error is still allowed to accumulate at a rate similar to LGC, and so another method of reducing error ought to be investigated.

2.2.4 Linearized Optimal Difference Continuation (LODC)

The next attempt concerns linearizing the difference between any two trajectories and solving for that difference as an optimal control problem, given the difference in two previous solutions as a guess. The optimized difference can then be added to the closer of the previous solutions to produce a guess for the original optimal control problem. Effectively, this is not much different that solving two optimal control problems successively with DC as a guess, but since the first optimal control problem is linearized, the additional computational effort is worth the expected increase in accuracy.

LODC assumes that there exist state and control trajectories, $\mathbf{x}^o(t)$ and $u^o(t)$ respectively, that are known to be a converged solution to a nearby problem that needs to be optimized.

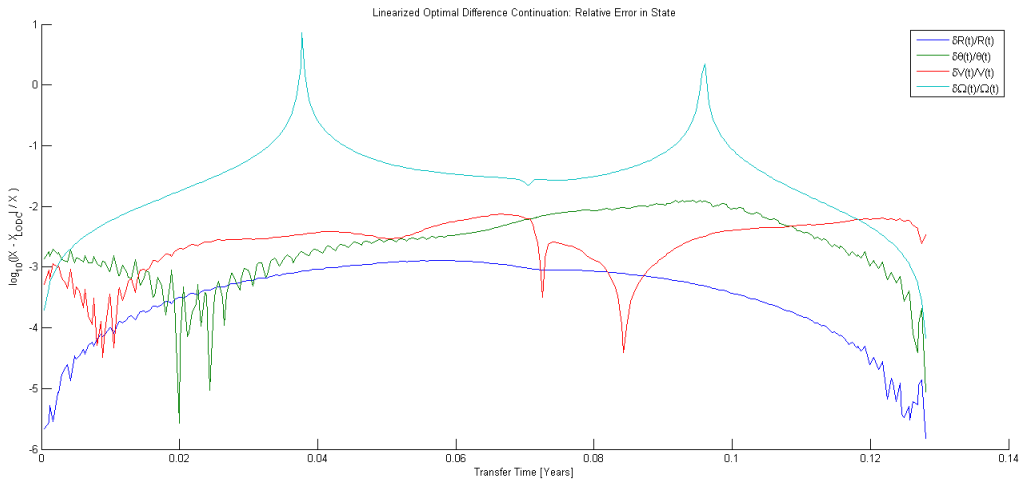


Figure 2.9: Plot of $log_{10}(|X^O - X^{LODC}|/X^O)$, the relative error between the LODC guess and the solution it was attempting to approximate.

Performing standard linearization about these quantities produces state dynamics of the following form

$$\delta\dot{\mathbf{x}}(t) = \nabla_{\mathbf{x}(t)}\boldsymbol{f}[\mathbf{x}^o, \mathbf{u}^o; t]\delta\mathbf{x}(t) + \nabla_{\mathbf{u}(t)}\boldsymbol{f}[\mathbf{x}^o, \mathbf{u}^o; t]\delta\mathbf{u}(t) \tag{2.26a}$$

$$\nabla_{\mathbf{x}(t)}\boldsymbol{f}[\mathbf{x}^o, \mathbf{u}^o; t] = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ (\Omega_o^2 + 2\frac{\mu}{R_o^3}) & 0 & 0 & 2R_o\Omega_o \\ \frac{-1}{R_o^2}(-2V_o\Omega_o + Asin(u_o)) & 0 & -2\frac{\Omega_o}{R_o} & -2\frac{V_o}{R_o} \end{pmatrix} \tag{2.26b}$$

$$\nabla_{\mathbf{u}(t)}\boldsymbol{f}[\mathbf{x}^o, \mathbf{u}^o; t] = \begin{pmatrix} 0 & 0 & -Asin(u_o) & \frac{A}{R_o}cos(u_o) \end{pmatrix}^\intercal \tag{2.26c}$$

$$\tag{2.26d}$$

where $\delta\mathbf{x}(t_0) \equiv [0; 0; 0; 0]$ and $\delta\mathbf{x}(t_f) = \delta\mathbf{x}_f$ are the boundary conditions associated with these dynamics.

LODC Error Analysis

Since the first step in generating a guess via LODC is to provide another guess based on DC, the error is expected to be only somewhat better than using DC alone. However, while relative state errors were not observed to be significantly better than with DC, convergence was found to be somewhat more robust. This is likely due to the fact that the numerical solver is simply working longer on finding a solution. It is for this reason that LODC is the method chosen to construct the vast majority of the database discussed in the next chapter.

Chapter 3

Results and Classifications

3.1 Database Construction

Irrespective of continuation methods used or number and granularity of dimensions sampled, the same basic steps apply in the construction of any database of time-optimal trajectories with a finite number of dimensions.

1. Using a known analytic solution (here, the $\mu \equiv 0$ case), find one time-optimal solution that is expected to closely mimic the known analytic solution (denoted as a "seed" trajectory).

2. Using a one-trajectory continuation method (here, LGC), continue the database in one dimension (e.g. acceleration magnitude) for a desired number of sample points.

3. Then, again using a one-trajectory continuation method (here, LGC), continue exactly one step in each dimension orthogonal to the first (e.g. final true anomaly).

4. Finally, using a multi-trajectory continuation method (here, LODC), continue the database for a desired number of sample points in every orthogonal dimension, starting again at step 2 for any additional dimensions.

For the problem presented here of minimum-time Earth-Mars transfers, a database with accelerations varying (logarithmically) between $10^{-2.5}g$ to $10^{-3.7}g$ is constructed. In order to find local, as well as global minima, a number of databases were actually computed, each with a different initial "seed" trajectory. These were later combined and the minimum transfer time selected from points within the database with identical acceleration and physically indistinguishable terminal points in Mars's orbit (i.e. values of $\Delta\theta_f$ invariant under addition of integer multiples of 2π). The first database constructed uses a "seed" trajectory with $\theta_f = 0$ continued in positive and negative directions until $\Delta\theta_f = 2\pi$ and $\Delta\theta_f = -\pi$ (respectively), with sampling densities of 101 and 51 trajectories (respectively) at every acceleration. The second database starts at $\theta_f = \pi$ and continues in the negative direction for to $\Delta\theta_f = 0.0$ with a sampling density of 51 trajectories at every acceleration. A third database, at a constant acceleration of $A = 10^{-2.9}g_0$, was also computed with a higher sampling density of 1533 trajectories. The "seed" trajectory had a final true anomaly $\theta_f = 0$ and proceeded in the positive and negative directions for approximately one full and one half turn (respectively). The databases are summarized in Table 3.1. The reason for separately constructing a number of different databases with different initial "seed" trajectories is that it was initially suspected that traversing the same final true anomalies from different directions might allow the numerical solver to find optimal trajectories with qualitatively different

Database (Direction)	Traversed θ Range / Values	Accel. Range	# of Nodes
Q0 (Positive)	$\Delta\theta_f \in [+0.000\pi, +2.000\pi]$	$A \in [10^{-3.7}g_0, 10^{-2.5}g_0]$	$N = 1313$
Q0 (Negative)	$\Delta\theta_f \in [+0.000\pi, -1.000\pi]$	$A \in [10^{-3.7}g_0, 10^{-2.5}g_0]$	$N = 663$
Qπ (Negative)	$\Delta\theta_f \in [+1.000\pi, +0.000\pi]$	$A \in [10^{-3.7}g_0, 10^{-2.5}g_0]$	$N = 663$
A29 (Positive)	$\Delta\theta_f \in [+0.000\pi, +2.210\pi]$	$A = 10^{-2.9}g_0$	$N = 1000$
A29 (Negative)	$\Delta\theta_f \in [+0.000\pi, -1.262\pi]$	$A = 10^{-2.9}g_0$	$N = 533$

Table 3.1: Database list

behaviors. The same was thought to be true of trajectories with physically indistinguishable, but numerically distinct, final true anomalies invariant under the addition of integer multiples of 2π. The former case was not confirmed, but the latter case was indeed supported by the appearance of sudden changes in trajectory behavior (as will be seen in the section "Branching Points"). However, before any definitive claims can be made concerning the properties of local solutions, it must first be determined whether or not these trajectories are even local minima to begin with.

3.2 Database Filtering

After computing trajectories using the algorithm described above, trajectories which are definitely not (local) solutions need to be pruned from the database. To do this, the first order necessary conditions (FONC), found analytically in Equations 2.5 and 2.6, are used. Using the FONC in this form is justified by the fact that (1) this problem concerns time-optimality and (2) that the control variable $u(t)$ is effectively unconstrained.

$$\mathcal{H} = 1 + \lambda_R V + \lambda_\theta \Omega + \lambda_V (R\Omega^2 - \frac{\mu}{R^2} + Acos(u)) + \frac{\lambda_\Omega}{R}(-2V\Omega + Asin(u)) \equiv 0 \quad (3.1)$$

$$\lambda_V tan(u) - \frac{\lambda_\Omega}{R} = 0 \quad (3.2)$$

Once the deviation from each of these FONC is computed for every trajectory in every database, then the solutions that did not converge to within a certain tolerance (specifically $10^{-0.8}$ in $\mathcal{H}$ and $10^{-1.2}$ in $\mathcal{H}_{u(t)}$, as presented in Figure 3.1) can be pruned. The rest of the database should not contain any solutions which are not local solutions. [1] Overall,

[1] Technically this is different than saying that all remaining solutions are local solutions, as these are only *necessary* and not *sufficient* conditions for optimality that are being tested.

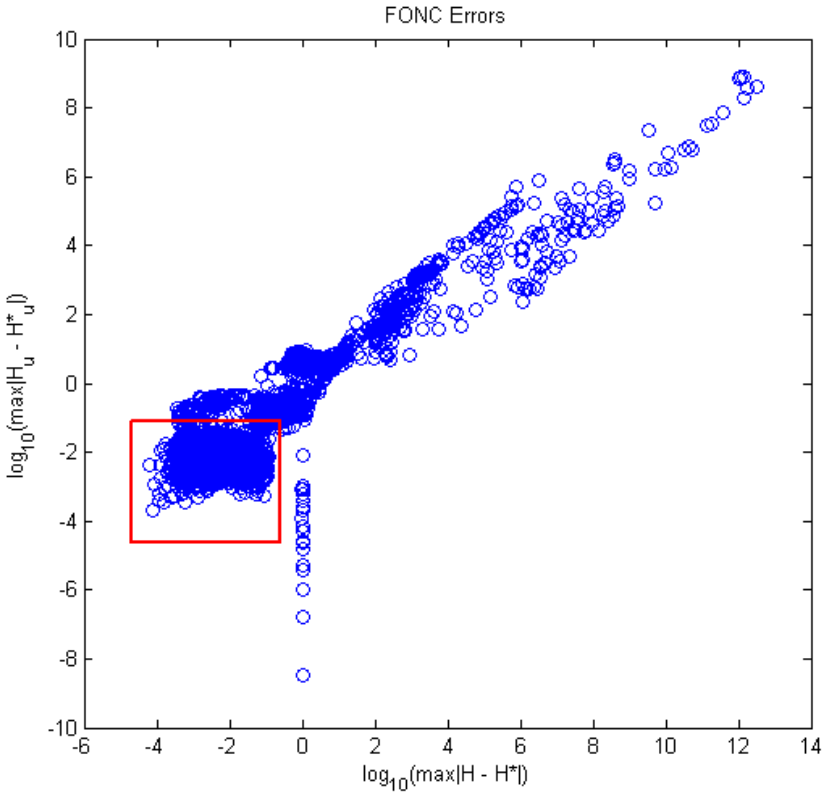


Figure 3.1: Plot of errors associated with meeting the FONC's. The red rectangle indicates the region of acceptable trajectories (error range: $\mathcal{H} < 10^{-0.8}$ and $\mathcal{H}_u < 10^{-1.2}$).

only about 56.9% of trajectories from all databases were not disqualified for not being local solutions. Most of these pruned solutions tended to be in the lower levels of acceleration, perhaps indicating that solutions to these problems may begin to qualitatively diverge rapidly from the analytic solution for $\mu \equiv 0$ (a form of initial guess which seems to perform well for higher accelerations.)

3.3 Branching Points

After filtering, interesting patterns can now be sought with the knowledge that they should reflect physical phenomena rather than numerical instabilities. The first pattern that can be noted is the existence of "branching" or "switching points," i.e. points in a database where

Acceleration Magnitude	Switching Point [rad]	Difference [rad]
$10^{-2.5}g_0$	$\theta_S \approx 1.30\pi$	undefined
$10^{-2.6}g_0$	$\theta_S \approx 1.38\pi$	$\Delta\theta_S = 0.08\pi$
$10^{-2.7}g_0$	$\theta_S \approx 1.48\pi$	$\Delta\theta_S = 0.10\pi$
$10^{-2.8}g_0$	$\theta_S \approx 1.60\pi$	$\Delta\theta_S = 0.12\pi$
$10^{-2.9}g_0$	$\theta_S \approx 1.74\pi$	$\Delta\theta_S = 0.14\pi$

Table 3.2: Branching Points.

two equal acceleration trajectories meeting in the same place in Mars's orbit at the same time exhibit qualitatively different control behaviors. As can be seen for $A = 10^{-2.9}g_0$ in Figure 3.2, the branching point takes on the specific value $\Delta\theta_f \approx 1.734\pi$ (see Figure 3.3 for the two branching trajectories). Perhaps unsurprisingly though, this point also corresponds to the maximum transfer time between these two orbits. This is because the minimum is being taken between corresponding elements of two observably continuous sets, which in general produces a discontinuity at the shared maximum minimal value. In a similar vein, the minimum occurs within the boundaries of these two sets and not at the endpoints.

The actual computed branching points for each set of trajectories are summarized in Table 3.2. A linear rate of increase in the switching angle, θ_S, with acceleration is observed. A quadratic relationship between θ_S and the difference in acceleration exponent, $\delta\alpha \equiv 0.1$, can approximate this. Letting $A = A_0 10^{-\delta\alpha n}$, where $A_0 = 10^{-2.15}g_0$, $n \in [3.5, 7.5]$, and $\theta_{min} = 1.17\pi$, we can write the following when $A \in [10^{-2.9}g_0, 10^{-2.5}g_0]$.

$$\theta_S(n) \cong \theta_{min} + (n\delta\alpha)^2 \tag{3.3}$$

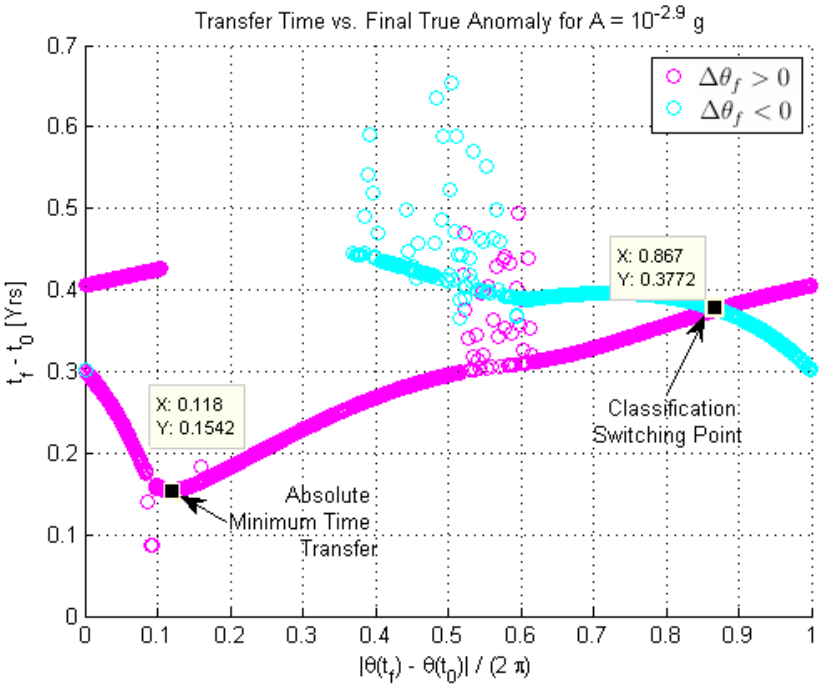
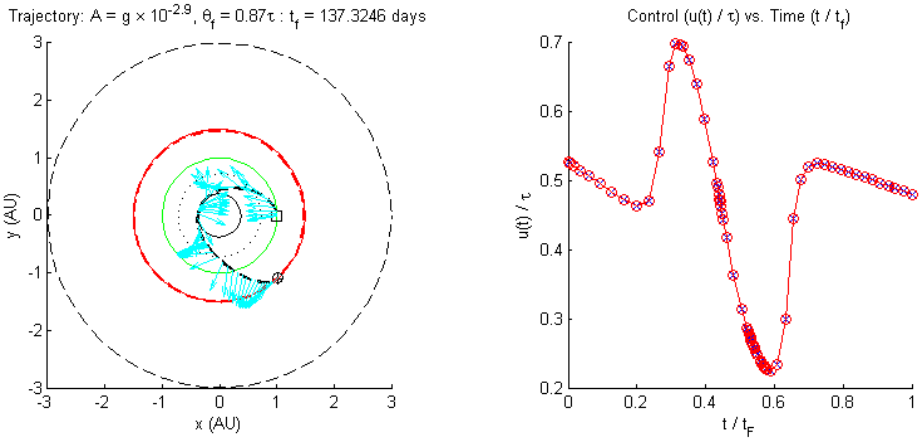


Figure 3.2: Plot of transfer time versus final true anomaly (mod 2π). Classification switching point and maximum transfer time are seen to coincide.

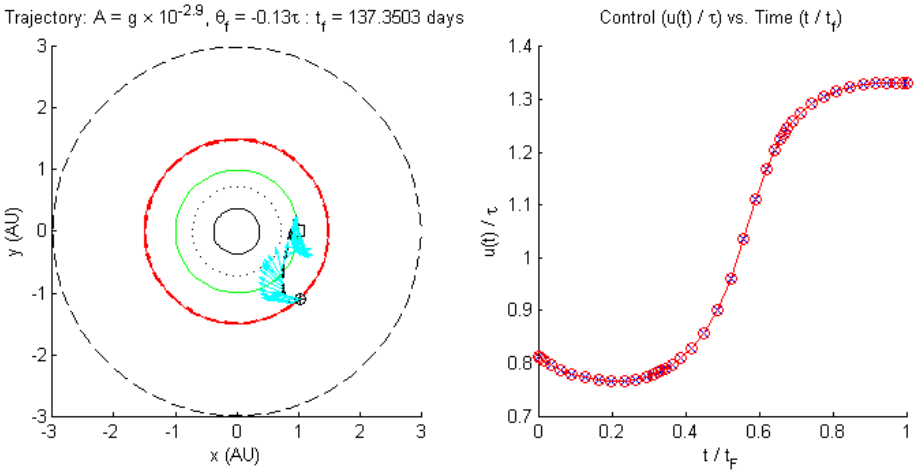
3.4 Transfer Classifications

3.4.1 Database Visualization

In order to analyze the thousands of presumed locally-optimal trajectories, a "radial plot" is presented so as to convey the (minimal) total transfer time, final Mars true anomaly, and time-varying characteristics of any given state or control variable, all at a specific acceleration magnitude (see Figures 3.4, 3.5, and 3.6). The way in which these figures should be interpreted is as follows. Each line radiating outward represents a single trajectory. Its angular position with respect to the center of the inner circle is exactly $\Delta\theta_f$ for that trajectory, and the length of that line is proportional to the total transfer time, t_f, of every other trajectory in that plot. The colors along each line denote the value of the variable being displayed. For $A = 10^{-2.5}$ to $A = 10^{-2.9}$ we have complete sets of converged solutions, so such "radial plots" can be constructed for entire Mars orbits. Beyond this range, however, the errors in



(a) $\Delta\theta_f = 1.74\pi$



(b) $\Delta\theta_f = -0.26\pi$

Figure 3.3: Branching point for trajectories at $A = 10^{-2.9}g_0$. Qualitatively different control behaviors are easily observed in this case.

FONC are too large for too many trajectories to be able to compose complete sets, so only accelerations within the aforementioned range are plotted in such a manner.

Classifications are based on the observed changes in the time derivatives of the state, control, and other physical characteristics. Since these databases are discrete sets of points that can have noise associated with the numerical solution process, it is handy to have these radial plots to consult. Note that while all the radial plots shown here are for the specific case of $A = 10^{-2.9}g_0$, this qualitative analysis applies to higher accelerations as well.

3.4.2 Control Classifications

Three distinct regions were found in Figure 3.4 that illuminated the distinct differences in the nature of the control that was being used. As seen in Figure 3.4a, there are three "branching" points that are clearly visible. One occurs at $\Delta\theta_f \approx 0.25\pi$, another at $\Delta\theta_f \approx 1.25\pi$, and a third at $\Delta\theta_f \approx 1.75\pi$. The first branch point corresponds very well with the point at which the analytic solution for the $\mu \equiv 0$ case is most closely approximating the trajectory. Transfers within the green region of the inner annulus are the most similar to a "bang-bang" control, as seen by the single switching point (boundary between red and blue regions in Figure 3.4a). The second branch point corresponds with the point at which the minimum radius constraint is active. The hard "bang-bang" control is forced to smooth itself out so as to keep the trajectory above that minimum radius. For trajectories in the blue region, the additional variation in control is most likely due to this phenomenon. The final branch point also happens to be a switching point where the transfer time is at a maximum and the qualitative nature of the trajectory discontinuously changes so as to maintain a minimum transfer time. Looking back at the first control branch point, we see that the transfers in the red region exhibit another type of "bang-bang" behavior, but with two additional switching points corresponding to the two additional blue lines as seen across that classification.

3.4.3 State Classifications

Additional classifications can be identified by looking at various quantities derived directly from the state solution to each optimal control problem. The quantities investigated here are the radial position, radial velocity, tangential velocity, and the time derivative in tangential velocity, all of which produce six more classifications.

Radial Position and Velocity

As with the control classifications, there are three distinct regions visible in Figure 3.5 with three corresponding branch points occuring at $\Delta\theta_f \approx 0.25\pi$, $\Delta\theta_f \approx 0.75\pi$, and $\Delta\theta_f \approx 1.75\pi$. The first branch point corresponds exactly with the first branch point in the control classifications, indicating that in these variables the trajectories most closely approximating the $\mu \equiv 0$ case present a boundary between different trajectory behaviors. The second branch point approximately corresponds with a point at which spacecraft begin traveling inside of Earth's orbit and begin to plunge closer and closer to the Sun with increasing $\Delta\theta_f$. The final branch point, as with the control classifications, also corresponds with the switching point in transfer time.

Trajectories in the green region correspond with trajectories that most closely represent a linear "bang-bang" solution, where trajectories in the blue region begin to take advantage of the inverse-square gravitational force in minimizing transfer time. This behavior gets more pronounces as solutions approach the red region barrier. It is also apparent here that transfers in the blue region begin to be constrained by minimum radius, and so spend a long time at that constant minimum radius throughout the transfer, being forced to travel in a circle for a finite amount of time.

Tangential Velocity and $\frac{d(R\Omega)}{dt}$

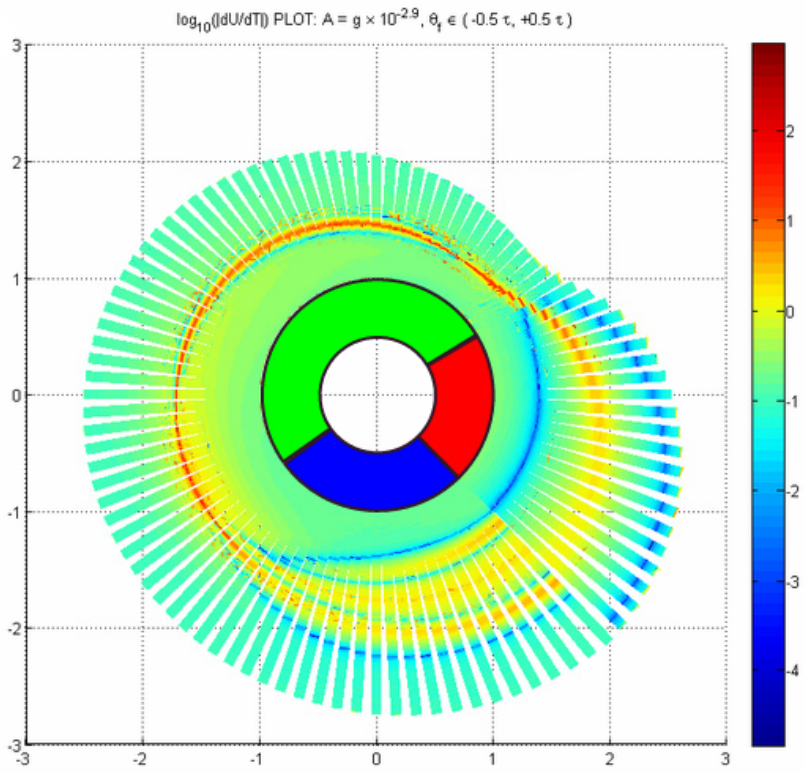
The final set of classifications identified are associated with the tangential velocity of the spacecraft as it transfers between the Earth and Mars. As with the other types of classifications, the first and last branch points correspond to the same points in space, namely the $\mu \equiv 0$ case and the switching point. The middle branch point differs from the other two sets, however, and is defined by the emergence of a soft approach to the final tangential angle. Trajectories beyond this point introduce more than one point where $\frac{d(R\Omega)}{dt} = 0$.

Of a particularly striking note is how obvious the transition across the switching point in the tangential velocity space appears (Figure 3.6b). Trajectories in the red classification all attempt to "back-track" or reverse the initial orbital velocity in order to intercept Mars, while all other trajectories attempt to increase tangential velocity before rendezvous.

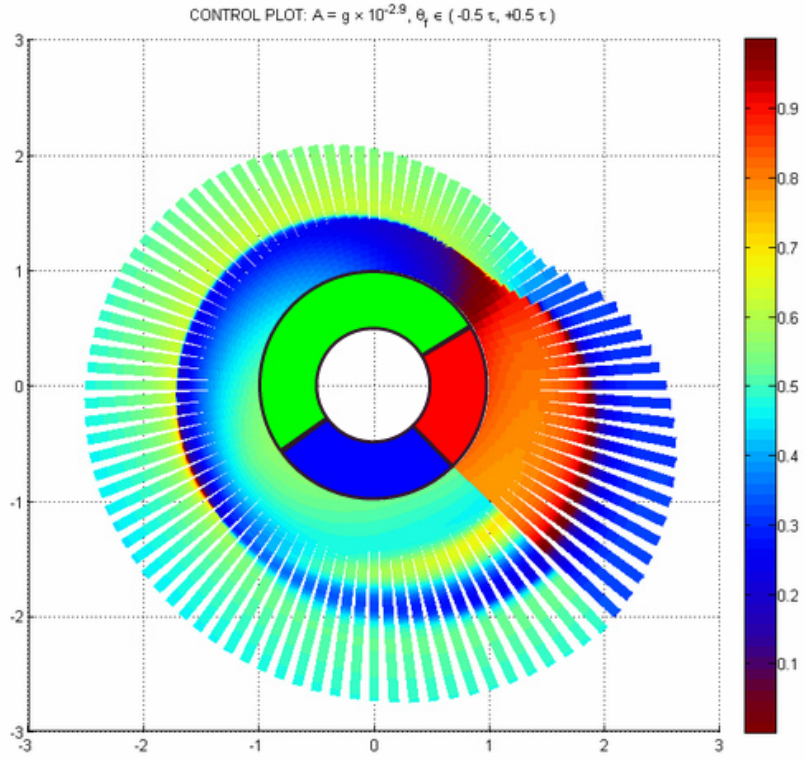
3.5 "Free Orbits" and Database Extensions

As a final note, consider what would happen should the spacecraft's propulsion system cut out partway through one of the time-optimal transfers discussed above (see Figure **??**). For most cases, it would end up in a closed orbit about the Sun (in the rest it would end up in a hyperbolic escape trajectory out of the Solar System or an orbit that crashes into the Sun, see Figure 3.8). Since all the orbits described above are consistent with the FONC required of locally optimal trajectories (Equations 3.1 and 3.2), it should be true that any subset of those trajectories also represents another local solution between two different sets of boundary conditions. This is because the FONC are constant relations over the entire trajectory time. That is, if one were to look at a non-trivial subset, $[t_0', t_f'] \subset [t_0, t_f]$ of the original trajectory, those state, costate and control variables automatically satisfy the FONC, and thus represent candidates for a local solution between the specified boundary conditions.

Therefore, given any specific transfer from the above database, new pairs of orbits analogous to the Earth-Mars transfer case considered here can undergo an identical database construction method as outlined above. It should be noted, however, that these trajectories are only postulated to be *local* solutions, especially if the original trajectory from which the new database is derived is only a local solution.

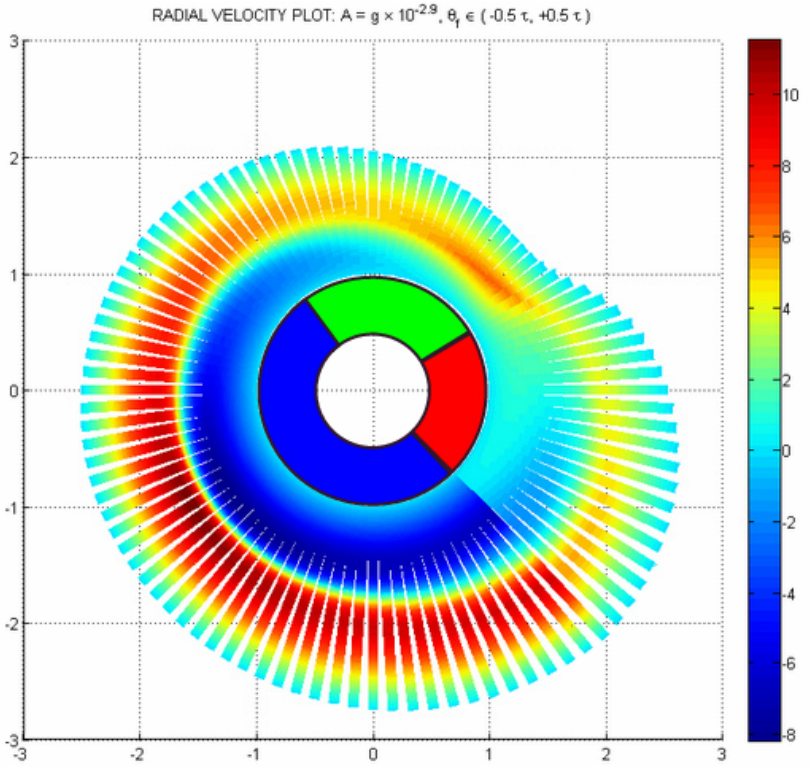


(a) Control derivative radial plot

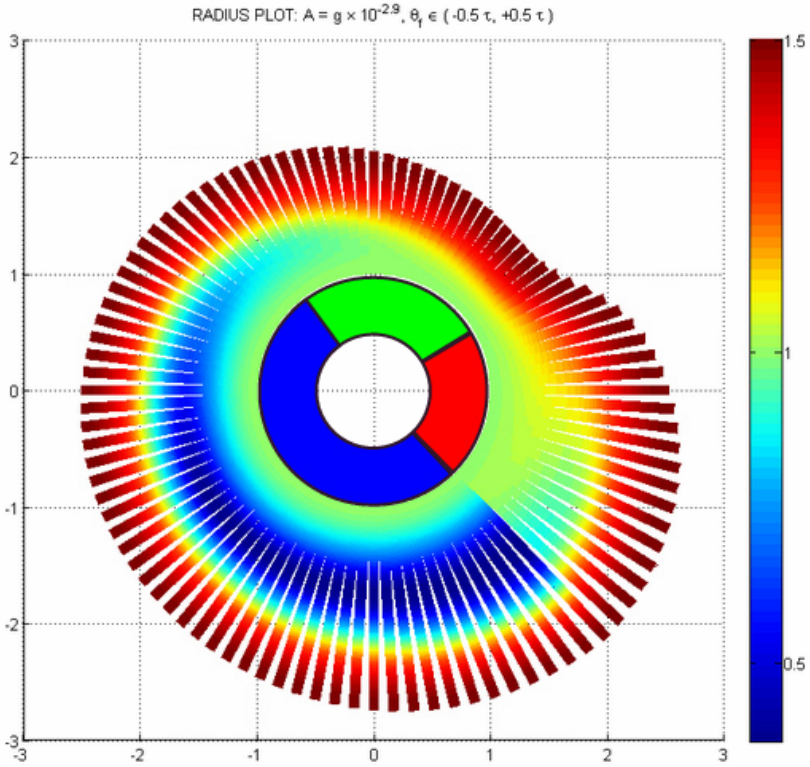


(b) Control radial plot

Figure 3.4: Classifications denoted by red, blue and green regions in inner annulus.

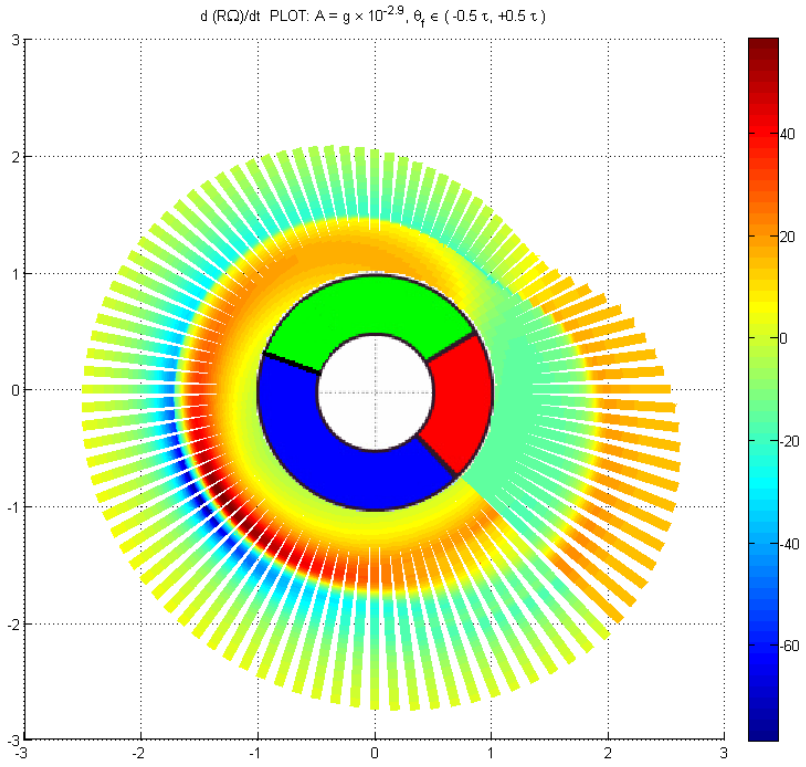


(a) Radial velocity radial plot

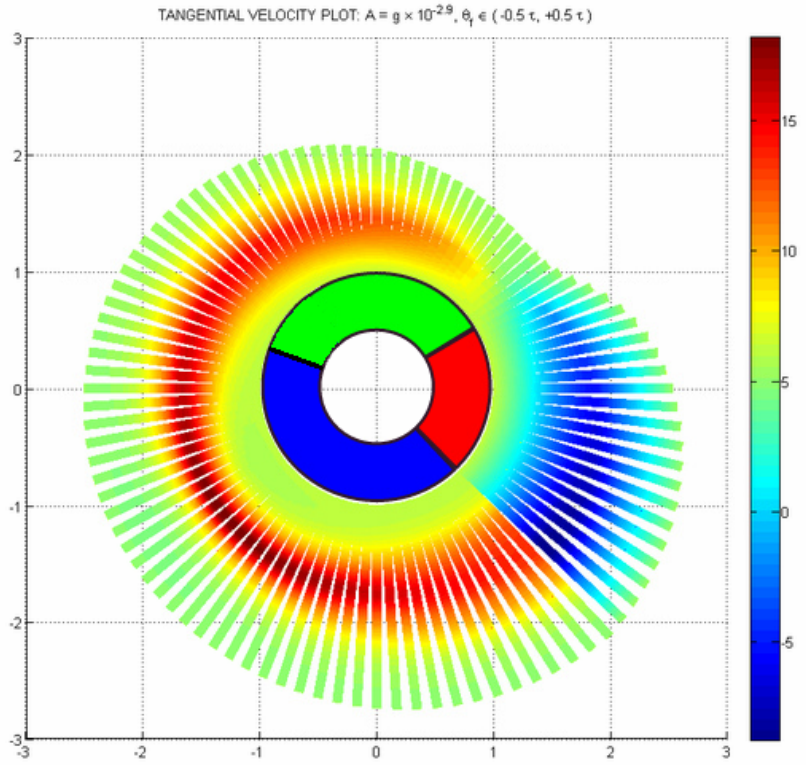


(b) Radial distance radial plot

Figure 3.5: Classifications denoted by red, blue and green regions in inner annulus.

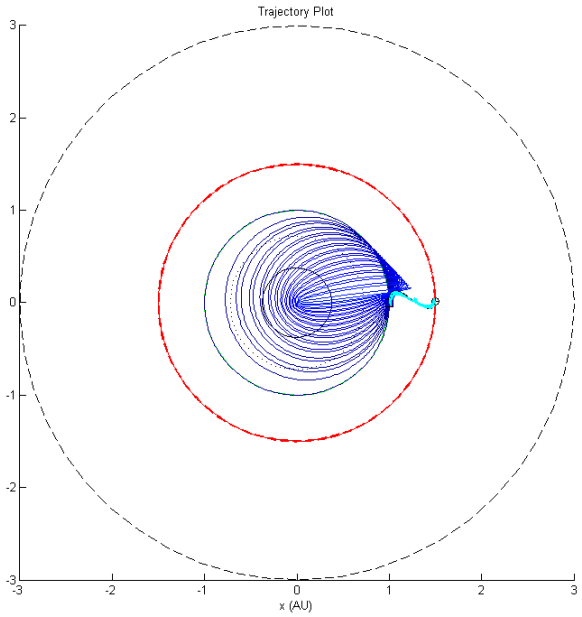


(a) Tangential velocity derivative radial plot

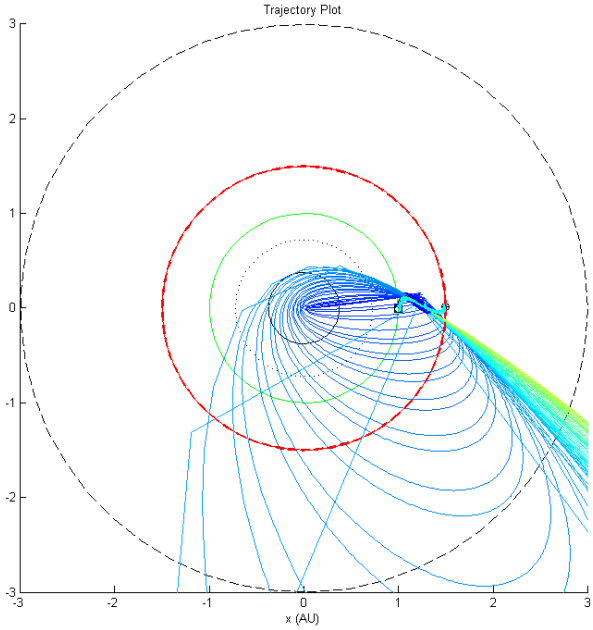


(b) Tangential velocity radial plot

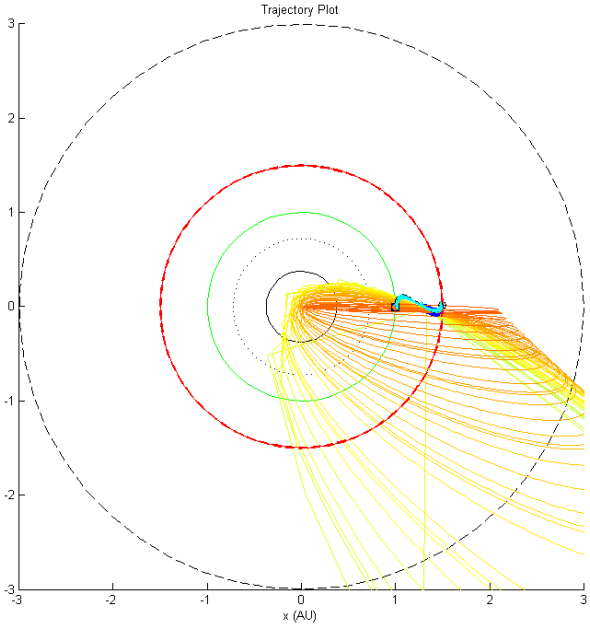
Figure 3.6: Classifications denoted by red, blue and green regions in inner annulus.



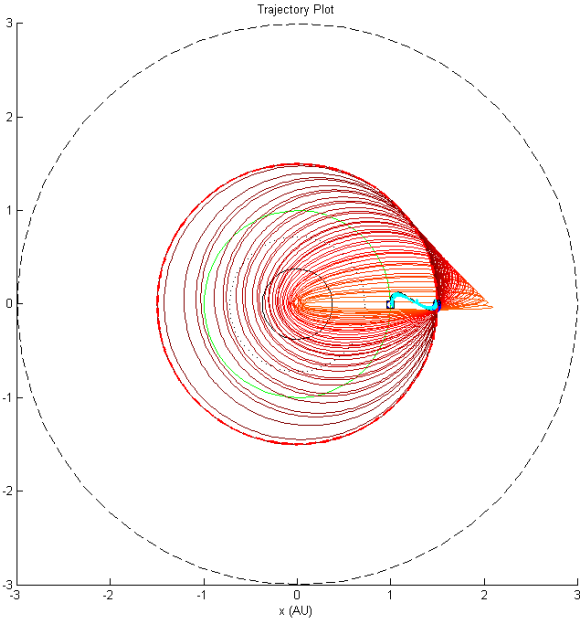
(a) Extended orbits for $\delta t \in [0, 0.15]$.



(b) Extended orbits for $\delta t \in [0.15, 0.55]$.



(c) Extended orbits for $\delta t \in [0.55, 0.77]$.



(d) Extended orbits for $\delta t \in [0.77, 1]$.

Figure 3.7: **Free Orbits**: Each figure above shows the extended free orbits for a seed trajectory with $A = 10^{-2.5}g$ and $\theta_f = 0.0$.

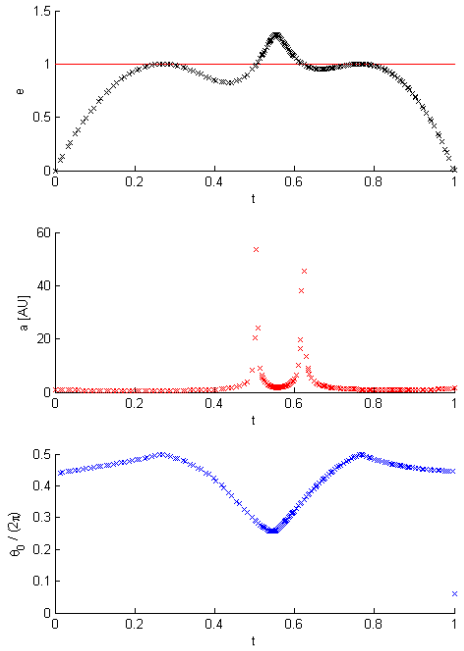


Figure 3.8: The eccentricity (top), semimajor axis (middle) and final true anomaly (bottom) of all the extended orbits depicted in Figure **??**

Chapter 4

Conclusion

A method of database construction was described for the exploration of constant-acceleration time-optimal Earth-Mars transfers and an analysis of its results was presented. Various continuation methods were designed and presented in order to efficiently construct such a database while maintaining the numerical stability of found solutions. Each continuation method was compared and its corresponding realm of applicability assigned based on its various strengths and weaknesses. After a number of example databases were computed their results were compiled and filtered based on the fact that they were all expected to satisfy the first order necessary conditions (FONC) of Equations 3.1 and 3.2 or else be deemed to have failed to converge. Once filtered, each dimension in acceleration space was plotted in a "radial plot" to highlight the various classifications that were derived from a qualitative analysis of the state and control variables and their first time derivatives. Finally, a method to extend the applicability of even the relatively simple database computed here was presented that involved the inclusion of "free orbits" associated with finite subsets of each individual local solution found in the original database. Such solutions should in principle satisfy the FONC if their parent trajectory also satisfies the FONC, thereby providing a

postulated set of new local solutions between specific orbits determined by the boundary conditions on the subset of a trajectory taken in this fashion.

Bibliography

[1] A. V. Arefiev and B. N. Breizman. Theoretical components of the vasimr plasma propulsion concept. *Physics of plasmas*, 11:2942, 2004.

[2] L. T. Biegler and V. M. Zavala. Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization. *Computers & Chemical Engineering*, 33(3):575–582, 2009.

[3] J. R. Brophy, M. G. Marcucci, G. B. Ganapathi, C. E. Garner, M. D. Henry, B. Nakazono, and D. Noon. The ion propulsion system for dawn. *AIAA Paper*, 4542, 2003.

[4] E. Y. Choueiri. A critical history of electric propulsion: the first 50 years (1906-1956). *Journal of Propulsion and Power*, 20(2):193–203, 2004.

[5] V. Coverstone-Carroll, J. Hartmann, and W. Mason. Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering*, 186(24):387 – 402, 2000.

[6] D. Feng and B. Krogh. Acceleration-constrained time-optimal control in dimensions. *Automatic Control, IEEE Transactions on*, 31(10):955–958, 1986.

[7] H. P. Geering. *Optimal control with engineering applications*. Springer, 2007.

[8] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[9] T. Haberkorn, P. Martinon, and J. Gergaud. Low thrust minimum-fuel orbital transfer: A homotopic approach. *Journal of Guidance, Control, and Dynamics*, 27(6):1046–1060, 2004.

[10] J. A. Kechichian. Optimal low-thrust rendezvous using equinoctial orbit elements. *Acta Astronautica*, 38(1):1 – 14, 1996.

[11] W. R. Kerslake and L. R. Ignaczak. Development and flight history of sert 2 spacecraft. In *Joint Propulsion Conference and Exhibit*, volume 1, pages 6–8, 1992.

[12] X. López Hellín. Study of earth-to-mars transfers with low-thrust propulsion. 2012.

[13] M. A. Patterson and A. V. Rao. Gpops- ii: A matlab software for solving multiple-phase optimal control problems using hp–adaptive gaussian quadrature collocation methods and sparse nonlinear programming. 2013.

[14] R. Ragsac. Study of electric propulsion for manned mars missions. *Journal of Spacecraft and Rockets*, 4(4):462–468, 1967.

[15] J. K. Whiting. *Orbital Transfer Trajectory Optimization.* PhD thesis, Massachusetts Institute of Technology, 2004.

[16] C. H. Yam, T. T. McConaghy, K. J. Chen, and J. M. Longuski. Preliminary design of nuclear electric propulsion missions to the outer planets. In *AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper*, volume 5393, pages 16–19, 2004.