Low-Thrust Many-Revolution Trajectory Optimization

by

Jonathan David Aziz

B.S., Aerospace Engineering, Syracuse University, 2013

M.S., Aerospace Engineering Sciences, University of Colorado, 2015

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2018

This thesis entitled:
Low-Thrust Many-Revolution Trajectory Optimization
written by Jonathan David Aziz
has been approved for the Department of Aerospace Engineering Sciences

_____

Daniel J. Scheeres

_____

Jeffrey S. Parker

_____

Jacob A. Englander

_____

Jay W. McMahon

_____

Shalom D. Ruben

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Aziz, Jonathan David (Ph.D., Aerospace Engineering Sciences)

Low-Thrust Many-Revolution Trajectory Optimization

Thesis directed by Professor Daniel J. Scheeres

This dissertation presents a method for optimizing the trajectories of spacecraft that use low-thrust propulsion to maneuver through high counts of orbital revolutions. The proposed method is to discretize the trajectory and control schedule with respect to an orbit anomaly and perform the optimization with differential dynamic programming (DDP). The change of variable from time to orbit anomaly is accomplished by a Sundman transformation to the spacecraft equations of motion.

Sundman transformations to each of the true, mean and eccentric anomalies are leveraged for fuel-optimal geocentric transfers up to 2000 revolutions. The approach is shown to be amenable to the inclusion of perturbations in the dynamic model, specifically aspherical gravity and third-body perturbations, and is improved upon through the use of modified equinoctial elements. An assessment of computational performance shows the importance of parallelization but that a single, multi-core processor is effective. The computational efficiency facilitates the generation of fuel versus time of flight trade-offs within a matter of hours.

Many-revolution trajectories are characteristic of orbit transfers accomplished by solar electric propulsion about planetary bodies. Methods for modeling the effect of solar eclipses on the power available to the spacecraft and constraining eclipse durations are also presented. The logistic sunlight fraction is introduced as a coefficient that scales the computed power available by the fraction of sunlight available. The logistic sunlight fraction and Sundman-transformed DDP are used to analyze transfers from low-Earth orbit to geostationary orbit. The analysis includes a systematic approach to estimating the Pareto front of fuel versus time of flight.

In addition to addressing many-revolution trajectories, this dissertation advances the utility of DDP in the three-body problem. Fuel-optimal transfers are presented in the Earth-Moon circular restricted three-body problem between distant retrograde orbits, between Lyapunov orbits and

between Halo orbits. Those include mechanisms for varying the time of flight and the insertion point onto a target orbit. A multi-phase DDP approach enables initial guesses to be constructed from discontinuous trajectory segments. DDP is shown to leverage the system dynamics to find a heteroclinic connection between Lyapunov orbits, which is facilitated by the multi-phase approach.

Dedication

To Mom and Dad and the rest of you.

Acknowledgements

Contents

Chapter

Bibliography   157

Appendix

...

Tables

Figures

Figure

Chapter 1

Introduction

Highly efficient low-thrust propulsion systems enable mission designers to increase the useful spacecraft mass or delivered payload mass above that from high-thrust engine options. This improvement typically comes at the expense of increased times of flight to mission destinations. For low-thrust trajectories about planetary bodies, an orbital period on the order of hours or days provides inadequate time to impart a substantial change to the orbit within a single revolution, which results in a large number of revolutions that are traversed before reaching the desired state. Determining the optimal control over hundreds or thousands of revolutions poses a sensitive and often unwieldy, high-dimensional optimization problem.

This dissertation provides the theoretical framework, demonstrations and analysis to fill the technology gap in high-fidelity, robust and efficient low-thrust many-revolution trajectory optimization. A summary of existing low-thrust tools and their applicability can be found in Reference [2], although many of those tools have matured through a decade of development and other tools have entered the mix. The 2015 NASA Technology Roadmap states that "current capabilities are not designed to compute multi-revolution spiral trajectories in a full dynamical model, nor can it exploit multi-body dynamics" [3]. A distinction is made between the *multi-* and *many-* prefixes to extend several or tens of revolutions to arbitrarily many. Primary attention has been given to the many-revolution aspect, but the solution has also proved effective in multi-body dynamic models.

1.1   Orbit Transfers

Fundamentals textbooks in the field of astrodynamics detail Tycho Brahe's meticulous observation of planetary motion, Johannes Kepler's deduction of the laws of motion that fit Tycho's data and Isaac Newton's formulation of the universal law of gravitation [4, 5]. Those laws provide a geometric description of orbits and the equations for idealized orbital motion. That idealized motion is named both Keplerian motion and two-body motion after the assumption of two point-masses in mutual gravitational attraction. The most popular descriptors of an orbit are the classical orbital elements, which are also called the Keplerian orbital elements. Those are a, the semi-major axis, e, the eccentricity, i, the inclination, Ω, the right ascension of the ascending node, ω, the argument of periapsis, and ν, the true anomaly. Figure 1.1 identifies the classical orbital elements for an elliptical geocentric (Earth-centered) orbit, except for the eccentricity which identifies how circular an orbit is. Figure 1.1 also identifies periapsis and apoapsis, the lowest and highest points along the orbit, named perigee and apogee for geocentric orbits. The true anomaly is an angular measure of the location of an object within the orbit and advances with time, while a, e, i, Ω and ω are constants that describe the size, shape and orientation of the orbit.
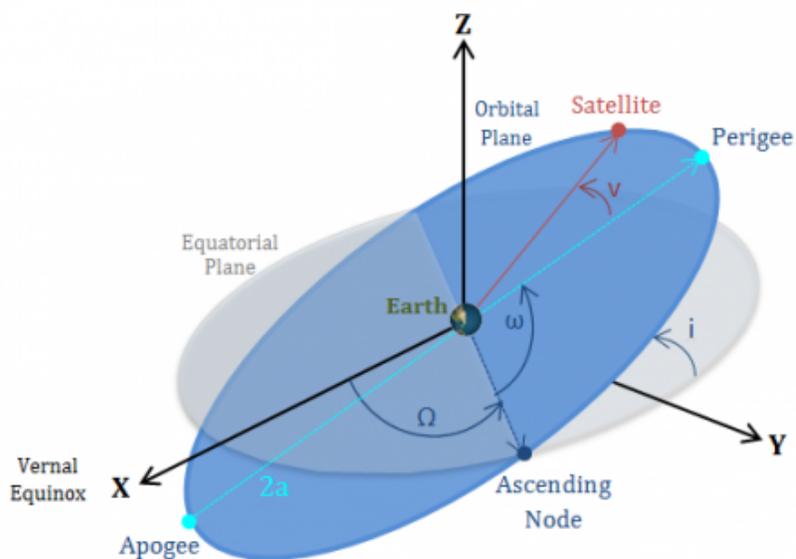
Figure 1.1: Description of a geocentric orbit [1].

Kepler's first law states that the planets move along conic sections, which include the circle, ellipse, parabola and hyperbola. Rectilinear orbits are a special case to be aware of. This dissertation is mainly concerned with elliptical orbits, inclusive of the circle with $e = 0$. Specifically, that concern is how to move a spacecraft from one elliptical orbit to another, i.e. how to change the orbital elements.

The orbital state of a spacecraft may alternatively be described by its position and velocity in Cartesian coordinates. Figure 1.1 shows a Cartesian coordinate system (XYZ). The spacecraft position is a vector with the coordinates x, y and z measured along each respective axis. The components of the spacecraft velocity vector are the time derivatives $\dot{x}$, $\dot{y}$ and $\dot{z}$. Unlike the orbital elements, all of the Cartesian coordinates change with each step along the orbit. The time taken to complete one revolution about the ellipse is the orbital period P, after which the true anomaly will have elapsed $360°$ and x, y, z, $\dot{x}$, $\dot{y}$ and $\dot{z}$ will have returned to their original values.

A preliminary concept in orbit transfer design is to apply an instantaneous change to the velocity components to set the spacecraft on a new path, or trajectory. The instantaneous change to the velocity is called a ΔV (delta-V). Intuitively, a ΔV requires a transfer of energy that most frequently comes from fuel expenditure by a spacecraft propulsion system to produce a thrust force. A notable exception is solar sailing, where the transfer of energy comes from solar photon pressure. Regardless of how the ΔV is to be applied, a common goal is to perform the orbit transfer with as little effort as possible. Trajectory optimization might begin by finding the minimum ΔV necessary to achieve a desired change in the orbital elements.

The Hohmann transfer between two circular orbits is perhaps the most famous minimum-ΔV transfer and is illustrated in Figure 1.2. To move to a higher orbital radius (increase semi-major axis), an initial ΔV_1 is pointed in the velocity direction and is sized to set apoapsis of the transfer orbit to the target circular orbit radius. Upon application of ΔV_1, the original circular orbit state instantaneously becomes the periapsis state of the transfer orbit. The spacecraft travels to the new apoapsis in half of the orbital period of the transfer orbit. Here the apoapsis position of the transfer orbit coincides with a position on the target circular orbit. The corresponding velocities

are different, however, and that difference is the second maneuver ΔV_2 which is again in the velocity direction and circularizes the orbit. The Hohmann transfer can be performed in reverse to move to a smaller circular orbit by redirecting ΔV_1 and ΔV_2 against the velocity vector.
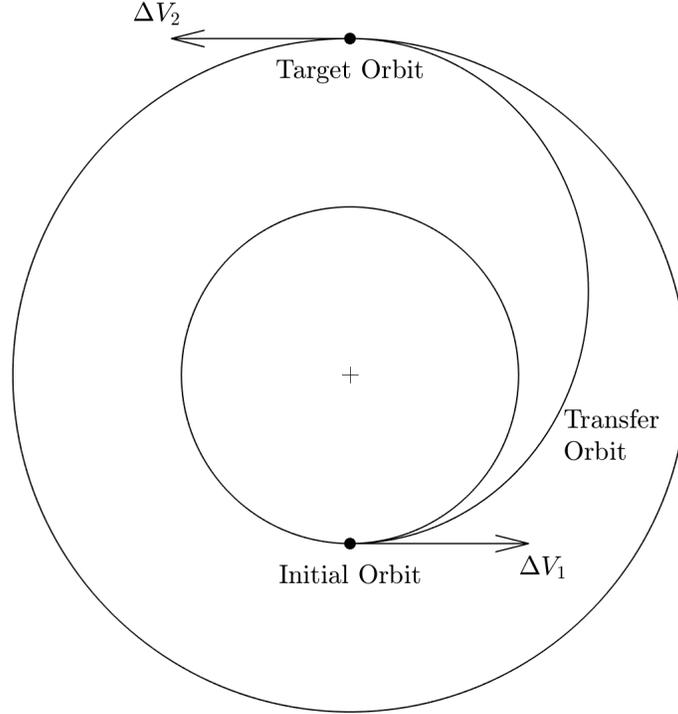
Figure 1.2: An example Hohmann transfer.

The fuel requirement for a given ΔV follows another famous result, Tsiolkovsky's rocket equation,

$$\Delta V = I_{sp} g_0 \ln \frac{m_0}{m_f},\qquad(1.1)$$

where I_{sp} is the specific impulse of the propulsion system, g_0 is the acceleration due to gravity at sea level, m_0 is the initial mass before the ΔV and m_f is the mass after the ΔV. The propellant mass m_p is simply the difference, $m_p = m_0 - m_f$. The specific impulse is a dimensional ratio, standardly in seconds, that measures how effectively the propulsion system uses the propellant mass to produce thrust.

$$I_{sp} = \frac{T}{\dot{m} g_0}\qquad(1.2)$$

In Equation 1.2, T is the thrust and $\dot{m}$ is the mass flow rate. For two engines with the same mass

flow rate, the higher-I_{sp} engine produces more thrust. A restatement of Equation 1.1 is

$$m_f = m_0 \, e^{\frac{-\Delta V}{I_{sp} g_0}},$$ (1.3)

which makes it clearer that for two engines performing the same ΔV, the higher-I_{sp} engine requires less fuel.

Spacecraft design and trajectory design are often performed in tandem. While one might seek to minimize the ΔV requirement for an orbit transfer, one might want to also maximize the ΔV capability of the spacecraft. Design parameters for the latter include the spacecraft wet mass (mass when the spacecraft is fully fueled), propellant mass, thrust, specific impulse and mass flow rate. Equations 1.1 and 1.2 show that increasing I_{sp} via increasing T and/or reducing $\dot{m}$ or increasing the mass ratio m_0/m_f will increase the ΔV capability. Practical design, however, is not as simple as just picking the desired performance specifications. One implication is that for a fixed wet mass, increasing the mass ratio is done by increasing the propellant mass thereby reducing the useful mass for the spacecraft structure, systems and payload. By instead increasing the wet mass, the launch cost to deliver the spacecraft to orbit may increase and the thrust required to yield the same acceleration necessarily increases per Newton's second law, $F = ma$.

Next considering an increased ΔV capability through a high-I_{sp} thruster, a trade-off is encountered between the thrust level and the mass flow rate. Intuitively, a higher throughput of propellant should produce a larger thrust. It is a challenge then to increase the I_{sp} by simultaneously increasing the thrust and reducing the mass flow rate. To date, reducing both the thrust level and the mass flow rate has proved most effective in increasing the ratio $T/\dot{m}$ and consequently increasing the specific impulse.

Low-thrust propulsion then offers an improved ΔV capability, but the premise of an instantaneous ΔV becomes erroneous. The required propellant mass or delivered mass are more appropriate measures of trajectory performance. The replacement for the instantaneous ΔV approximation is named a finite burn. Both high-thrust and low-thrust maneuvers span a measurable duration. For high-thrust, the finite burn duration is short and the maneuver resembles an instantaneous

ΔV. The duration of a low-thrust finite burn may span the entirety of a transfer. An example continuous-thrust transfer is illustrated in Figure 1.3 with thrust vectors shown that are aligned with the velocity direction. This steering approach leads the spacecraft to the target circular orbit radius but with an error in eccentricity. Reaching the target orbit with zero eccentricity and in optimal time requires an oscillating steering angle with increasing amplitude throughout the transfer [6, 7].
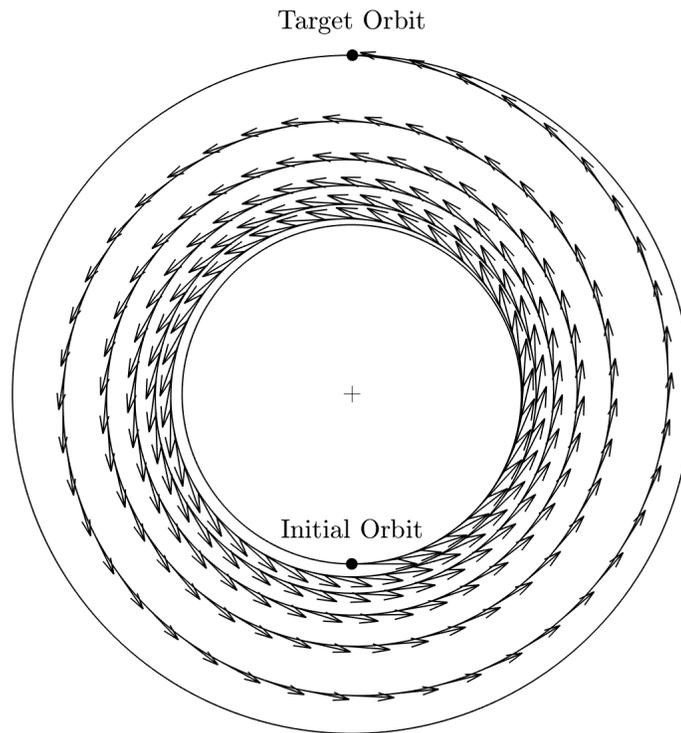
Figure 1.3: An example low-thrust spiral.

While the Hohmann transfer spans one half of one revolution, the example low-thrust transfer in Figure 1.3 spans five and a half revolutions. The necessary finite burn duration surpasses the orbital period so that a spiral trajectory occurs. Even more revolutions would be required with a lower thrust level or with coast arcs, but with the possibility to increase the final mass delivered to the target orbit. At just five and a half revolutions, the task of choosing the thrust optimally

appears tedious by visual inspection of Figure 1.3. How to do so for even longer low-thrust many-revolution transfers, including changes to all of the orbital elements and rendezvous transfers, is the subject of this dissertation.

1.2   Historical Low-Thrust Many-Revolution Solutions

1.2.1   Indirect Methods

Classical approaches to low-thrust many-revolution trajectory optimization employ optimal control theory and are named indirect, Lagrange multiplier or adjoint methods. A pioneering result is Edelbaum's transfer between circular orbits of different semi-major axis and inclination. Edelbaum developed an analytical solution to maximize the inclination change Δi while achieving a specified semi-major axis change Δa between circular orbits [8]. The result was repeated by maximizing the delivered mass over a fixed transfer duration with Δa and Δi specified [9]. Edelbaum assumed the orbit to remain circular throughout the transfer, a constant thrust angle within each revolution, a constant thrust acceleration and two-body dynamics. The circular orbit and constant angle assumptions are reasonably accurate, but constant thrust acceleration is a poor model for a propellant consuming spacecraft and does not allow for coasting. Furthermore, a two-body dynamic model is insufficient for long duration low-thrust missions. Wiesel and Alfano recast the problem to minimize the accumulated ΔV when Δa and Δi are specified [10]. The circular orbit assumption was maintained, but a constant thrust magnitude and mass flow rate were assumed instead of constant thrust acceleration and the thrust angle was allowed to vary. Minimizing ΔV under these conditions similarly minimizes the transfer time and fuel consumption. Wiesel and Alfano solved the resulting two-point boundary value problem numerically to produce a contour map of a Lagrange multiplier as a function of Δa and Δi. Thrust steering for the fast time scale transfer within a single revolution is found analytically after obtaining the Lagrange multiplier from the map. For the long time scale problem, i.e. many-revolution transfers, the Lagrange multiplier is again read from the map, but the thrust angle expression must be approximated numerically.

Kéchichian reformulated Edelbaum's transfer as a minimum-time problem and found a simple analytical expression for the time-varying thrust angle to achieve the desired Δa and Δi [11].

Edelbaum extended his analysis to obtain solutions for small changes to any or all of the classical orbital elements for elliptical orbits, again in fixed time with a continuous thrust acceleration and a two-body dynamic model [12]. The two-point boundary value problem requires numerical solution for the fast time scale transfer, but is reduced to analytical expressions for many-revolution transfers by neglecting periodic terms in the orbital element rates of change due to thrust. Kéchichian obtained the adjoint equations of motion for the minimum-time rendezvous problem in nonsingular equinoctial elements and presented a numerical solution with Newton-Raphson iteration [13]. He suggested an orbit averaging technique to extend this result to long duration rendezvous and followed such an approach to improve Edelbaum's transfer to include changes in the right ascension of the ascending node $\Delta\Omega$ and Earth's J_2 perturbation in the dynamic model [14]. Kéchichian further developed the low-thrust rendezvous in equinoctial elements by considering Earth zonal harmonics up to J_4 [15] and shows analytical but suboptimal approaches for achieving Δa or Δi with eclipses included [16–18].

The indirect problem is *solved* when Lagrange multipliers (adjoints or costates) are found to produce admissible states and controls that extremize the Hamiltonian at every instant in time, while obeying the state and adjoint equations of motion and satisfying boundary and transversality conditions. Hence, Pontryagin's necessary conditions are satisfied. Under significant assumptions, e.g. Edelbaum's transfer, analytic expressions are found that enable quick trajectory computation. Typically, however, values of the Lagrange multipliers must be guessed, evaluated in the equations of motion and corrected. The evaluation and correction is iterated in a numerical procedure until optimality conditions are satisfied. An appropriate initial guess for the Lagrange multipliers is nonintuitive and the resulting trajectory is sensitive to their values. That sensitivity is amplified when the trajectory encompasses many revolutions. The indirect approach is further complicated by the need to reform the Hamiltonian and re-derive the adjoint equations of motion and boundary conditions as different state variables, constraints and dynamics are considered.

1.2.2   Control Laws

In a similar flavor to the indirect approach, control laws use just a few input parameters to describe a rule for maneuvering a spacecraft during a transfer. The indirect solution is an optimal control law that is determined by the Lagrange multipliers, but suboptimal control laws that follow some heuristic can be particularly useful in mission design and even approach optimal results. Control laws prescribe the thrust direction and magnitude as a function of the current state and time or angular position, for example. Spacecraft equations of motion are then evaluated with thrust adhering to the control law until the target orbit state is reached, but if the control law is poorly designed, trajectory computation can proceed indefinitely. Lyapunov feedback control laws are desirable so that convergence can be expected.

Kluever blended the results of optimal control theory for the individual minimum-time changes Δa, Δi and to eccentricity Δe, subject to two-body dynamics [19]. The resulting control law assembles the thrust direction from the weighted optimal thrust directions to change each of those orbital elements. Chang, Chichka and Marsden suggested a Lyapunov function that is the weighted sum of squared errors of the angular momentum and Laplace vectors between the current and target orbit [20]. Minimizing this Lyapunov function at each instance in time yields asymptotic stability for local transfers, i.e. small changes, between Keplerian orbits. Chang et al. suggested transfers through a finite number of intermediate orbits for arbitrary global transfers, e.g. over many revolutions, and presented a method for choosing those intermediate orbits.

Petropoulos derived analytic expressions for the optimal thrust directions and optimal orbit locations for changing each of the classical orbital elements and proposed two control law strategies that include a mechanism for coasting based on the effectivity of the maneuver [21]. This work was further developed as the Q-law, where Q is a candidate Lyapunov function named the proximity quotient [22,23]. Q captures the proximity to the target orbit and best-case time-to-go for achieving the desired change in each orbital element. Thrust directions are chosen to maximize the reduction in Q, but if the rate of reduction is below a user-specified value, the thrust is turned off. Petropoulos

shows a minimum-time Q-law transfer that approaches Edelbaum's result and the minimum-fuel case with coasting over 666 revolutions. Near-optimal results are found for more challenging transfers, including changes to all of the orbital elements except true anomaly, that served to benchmark the Q-law against other low-thrust many-revolution optimization techniques [24].

Heuristic control laws generally yield suboptimal trajectories, but follow a policy that a mission designer deems acceptable. They can be particularly useful to construct an initial guess for a separate optimization procedure or to estimate fuel and time of flight requirements. Furthermore, the control strategy can be detached from the dynamic model. That is to say, a rule for steering can be set forth and employed with an arbitrary set of active forces. Falck, Sjauw and Smith showed that control laws can be effective in high-fidelity planetocentric models [25].

1.2.3 Direct Optimization

While indirect methods solve for the abstract Lagrange multipliers, direct methods seek the physical variables explicitly. A decision vector is formed of control variables, state variables or other variable parameters that collectively describe an entire trajectory. The decision vector could also be the input parameters for a control law as Lee [26] and Kluever [27] both demonstrated. The direct optimization procedure then updates the decision vector iteratively until convergence criteria are satisfied.

Direct optimization of heliocentric low-thrust trajectories is dominated by direct transcription and nonlinear programming (NLP) techniques. Direct transcription transforms the continuous optimal control problem into a discrete approximation [28]. For example, the continuous control $\boldsymbol{u}(t)$ now becomes the sequence $[\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{N-1}]$, and that sequence is the decision vector for the discretized trajectory of N stages, also called grid points, mesh points or nodes. Nonlinear programming generally involves the assembly and inversion of a Hessian matrix that contains the second derivatives and cross partial derivatives of a scalar objective function with respect to the decision vector. The size of the optimization problem grows quadratically with the number of decision variables and proves to be a computational bottleneck when applying nonlinear pro-

gramming to planetocentric low-thrust trajectories that require large decision vectors. Scheel and Conway presented one of the first direct transcription approaches to the many-revolution problem for minimum-time, continuous thrust, planar orbit raising over 100 revolutions [29]. Betts solved the large-scale NLP problem for a variety of geocentric trajectories over several hundred revolutions using collocation and sequential quadratic programming in the Sparse Optimization Suite (SOS) software [30–34]. Reference [32] presents a 578 revolution transfer between low Earth orbits with continuous but throttled thrust and oblate Earth perturbations through J_4. Reference [33] presents a minimum-time lunar transfer with two burn phases of maximum thrust and an intermediate co-ast phase. More recently in Reference [34], Betts modeled coast phases for eclipse conditions and constructed initial guesses for thrust arcs with the control law from Chang et al [20]. An initial guess of the entire trajectory was constructed with a receding horizon algorithm and supplied to SOS for optimization. The equations of motion included solar and lunar gravity perturbations in addition to oblate Earth perturbations through J_4. Transfers are presented from low Earth orbit to geosynchronous orbit in 248 revolutions with 363 alternating burn and coast phases and to a Molniya orbit in 372 revolutions with 693 phases. Both cases maximize the delivered mass.

1.3 Differential Dynamic Programming

A linear scaling of the optimization problem size with the number of control variables is characteristic of differential dynamic programming (DDP) [35]. DDP solves a subproblem for each $\boldsymbol{u}_k$, $k \in [0, N-1]$ in the sequence $[\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{N-1}]$ to optimize a local model of the cost remaining along the trajectory, instead of viewing the decision vector as a whole. This is in stark contrast to other direct methods that update the entire control sequence in the computationally expensive inversion of a large Hessian matrix. If the control vector at each stage is of dimension m, then DDP solves N NLP problems of size m, rather than a single NLP problem of size Nm.

The DDP procedure for updating the nominal control policy is called the *backward sweep* and is motivated by Bellman's Principle of Optimality.

> *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision* [36].

Considering the state that results from applying the nominal control up to stage $k = N - 1$, the sole remaining decision is $\boldsymbol{u}_{N-1}$. Optimization of this final decision is now independent of those preceding it and minimizes the *cost-to-go* that is incurred along this final stage. After performing this optimization and stepping back to stage $k = N-2$, the remaining decisions are $\boldsymbol{u}_{N-2}$ and $\boldsymbol{u}_{N-1}$. The latter is known, however, and only the control at the current stage needs to be determined. An update to the entire control policy is possible by proceeding upstream to the initial state at stage $k = 0$, while optimizing each stage decision along the way.

The subproblem approach of DDP makes it an attractive trajectory optimization algorithm to address the high-dimensional nature of the low-thrust many-revolution problem. DDP is a second-order gradient based method that was first introduced by Mayne [37] and applied to the orbit transfer problem by Gershwin and Jacobson shortly thereafter [38]. Jacobson and Mayne formally present derivations and applications of the algorithm for discrete and continuous systems, bang-bang control problems, and stochastic systems in their aptly named text [35]. It wasn't until Whiffen introduced the Static/Dynamic Optimal Control (SDC) [39] algorithm that DDP truly earned its place in space mission design. SDC inserts spacecraft design optimization into DDP and drives the optimization engine in the Mystic Low-Thrust Trajectory Design and Visualization Software [40, 41]. Mystic can be viewed as state-of-the-art for the design of low-thrust many-revolution trajectories, best exemplified by its use in NASA's Dawn mission [42]. Despite the favorability of DDP for large scale optimization, runtime is still expected to grow with problem size. Mystic can instead leverage control laws when the DDP runtime becomes impractical [40]. Colombo used DDP to design low-thrust trajectories to asteroids [43]. Both Whiffen and Colombo noted the capability of DDP to find unplanned planetary flybys. Lantoine and Russell introduced Hybrid Differential Dynamic Programming (HDDP) [44–48] as a DDP variant that makes the most computationally expensive step suitable for parallelization. The approximation of low-thrust

by impulsive maneuvers separated by Keplerian arcs reduces intensive numerical computations to analytical expressions that make HDDP a fast preliminary mission design tool. Pellegrini and Russell added a number of contributions, including the pairing of HDDP with a quasi-Newton based method for obtaining second derivatives and the Multiple-Shooting Differential Dynamic Programming (MSDDP) algorithm [49–51]. Concurrent with the timeline of this dissertation, Ozaki applied DDP to optimize trajectories for robustness to modeling and maneuver execution errors [52–55].

1.4   Dissertation Overview

This introductory chapter began with a problem statement for low-thrust many-revolution trajectory optimization. A technology gap was identified that limits the exploitation of low-thrust many-revolution trajectories in space mission design. The concept of an orbit transfer was addressed with the case for using low-thrust propulsion. A literature review of historical approaches to the 'many-rev' problem was divided into indirect, control law and direct optimization methods. A formal statement of the general spacecraft trajectory optimization problem and a thorough survey of related numerical methods is available in Reference [56]. Differential dynamic programming was identified as a candidate approach for its amenability to large scale optimization. The seminal text [35] is a valuable resource to service a wider breadth of trajectory optimization problems beyond the niche of this dissertation.

Next, Chapter 2 covers the dynamic modeling that is leveraged throughout the dissertation. That begins with ideal Keplerian motion, how that is affected by low-thrust propulsion and how model fidelity is improved by including other perturbing accelerations. Different descriptions of control variables are introduced for commanding a low-thrust propulsion system. The modified equinoctial elements are detailed as an alternative to the classical orbital elements. Lastly, the circular restricted three-body problem is described.

Chapter 3 is a summary of the DDP algorithm. It builds off of the presentation in Reference [46] but leverages tensor notation and expanded use of an augmented state vector. Related

presentations can be found in References [57,58]. Implementation of the algorithm was first valida-

ted in Reference [59], which reproduced an Earth-Mars rendezvous transfer from Reference [47] and

paired DDP with monotonic basin hopping for stochastic global search. The Earth-Mars rendezvous

validation and stochastic search are presented to conclude the chapter.

The research milestone that made DDP effective for many-revolution trajectory optimization

was to change the independent variable of the spacecraft equations of motion from time to an orbit

anomaly via the Sundman transformation [57,60]. Chapter 4 details how the Sundman transforma-

tion enters the DDP algorithm and provides example transfers from geostationary transfer orbits to

geostationary orbit. Sundman transformations to each of the true, mean and eccentric anomalies

are compared and computational performance is assessed. Improvements to the initial approach in

Cartesian coordinates are shown by using modified equinoctial elements [61].

Chapter 5 presents a method for including the effects of solar eclipsing in the trajectory

optimization process. The sunlight fraction is introduced as a coefficient that scales the computed

power available from an arbitrary power model to only make use of the percentage of sunlight that

is available to the spacecraft solar panels. Inserting the eclipse geometry into a logistic function

constitutes a smoothed eclipse model that is twice-differentiable and suitable for use in DDP or

any other gradient-based optimization method. Chapter 5 is an extension of Reference [62] which

revisited the LEO to GEO transfer from Reference [34]. The analysis includes a methodical appro-

ach to sketch out the Pareto front of fuel versus time of flight and an analytic approximation to

the intersection of an orbit with the penumbral cone.

Additional capabilities of the Sundman-transformed DDP approach are demonstrated in

Chapter 6 through the use of penalty functions to enforce path inequality constraints. A first

example constrains the apoapsis and periapsis radii for a benchmark low-thrust many-revolution

orbit transfer and adds DDP to the comparison study in Reference [24]. In a second example, the

analytic form for the time spent in a cylindrical shadow by a Keplerian orbit is used as a metric to

constrain the maximum eclipse duration for a Mars orbit transfer.

Chapter 7 applies DDP to transfers in the Earth-Moon circular restricted three-body pro-

blem [58]. Formulations are presented for the flight time and a moving target. The latter is used to access all points along a desired periodic orbit as valid terminal states. Multi-phase DDP is applied to discontinuous initial guesses that combine segments of different periodic orbits. The penalty method for path inequality constraints from Chapter 6 is used to enforce a minimum-radius constraint. Numerical examples include multi-revolution transfers between distant retrograde orbits, a heteroclinic transfer between planar Lyapunov orbits and transfers between L1 and L2 Halo orbits.

Chapter 8 summarizes the contributions of this dissertation and discusses open areas for future work.

Chapter 2

Low-Thrust Spaceflight Dynamics

The trajectory of a spacecraft is the time history of its position in space. Where the space-craft goes next depends on its velocity and how its velocity changes depends on its acceleration. Accelerations are governed by the natural dynamics, i.e. gravitational accelerations, and may also be influenced by external forces such as a low-thrust propulsion system. Dynamic models are used to compute the accelerations and vary from low to high-fidelity based on how close the computed accelerations are to the truth. That includes the modeling of power and propulsion subsystems. The measure of fidelity is also married to the numerical procedure for propagating the spacecraft state through time. Generally speaking, increasing the model fidelity increases the computational burden. This chapter presents the dynamic models that are used in numerical examples throughout the dissertation.

2.1   Perturbed Keplerian Motion

A low-fidelity modeling approach is to simply add a thrust force to the two-body equations of motion. The two bodies are a spacecraft with negligible mass and a point mass central body. Without thrust, the trajectory can be described by Kepler's laws of planetary motion. Model fidelity may then be improved by including additional perturbations such as aspherical gravity perturbations and gravity from other massive bodies, and by computing the thrust available from a power model.

The perturbed Keplerian dynamics are first presented for a Cartesian representation of the

spacecraft inertial position and velocity.

$$\boldsymbol{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T, \tag{2.1a}$$

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T. \tag{2.1b}$$

The notational convention reserves boldface for vectors, and the absence of boldface indicates the scalar magnitude, e.g. $r = \|\boldsymbol{r}\|$ and $v = \|\boldsymbol{v}\|$. The Cartesian accelerations for perturbed Keplerian motion are

$$\ddot{x} = -\frac{\mu}{r^3}x + \frac{T_a}{m}u_x + \sum_{i=0}^{n_p-1} \ddot{x}_{p_i}, \tag{2.2a}$$

$$\ddot{y} = -\frac{\mu}{r^3}y + \frac{T_a}{m}u_y + \sum_{i=0}^{n_p-1} \ddot{y}_{p_i}, \tag{2.2b}$$

$$\ddot{z} = -\frac{\mu}{r^3}z + \frac{T_a}{m}u_z + \sum_{i=0}^{n_p-1} \ddot{z}_{p_i}, \tag{2.2c}$$

or in vector form,

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \frac{T_a}{m}\boldsymbol{u} + \sum_{i=0}^{n_p-1} \boldsymbol{\delta}_i. \tag{2.3}$$

The leading terms on the right-hand side of Equations 2.2 and 2.3 are the two-body accelerations, where μ is the gravitational parameter of the central body.

Control of the low-thrust propulsion system is described by the up-to-unit control vector $\boldsymbol{u}$.

$$\boldsymbol{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T \tag{2.4}$$

The control is bounded $u \in [0, 1]$ and is the throttle of the thrust available T_a. The thrust available is similarly bounded by the maximum thrust $T_a \in [0, T_{max}]$. The thrust control is mapped to an acceleration using the spacecraft mass m and Newton's second law. The final terms in Equations 2.2 and 2.3 sum the n_p perturbations that have been added to the model.

$$\boldsymbol{\delta}_i = \begin{bmatrix} \ddot{x}_{p_i} & \ddot{y}_{p_i} & \ddot{z}_{p_i} \end{bmatrix}^T \tag{2.5}$$

The differential equation for the spacecraft mass is the mass flow rate and allows the mass to be updated alongside the Cartesian states as propellant is exhausted.

$$\dot{m} = u\frac{T_a}{I_{sp}g_0} \tag{2.6}$$

In Equation 2.6, I_{sp} is the specific impulse of the propulsion system and g_0 is the acceleration due to gravity at sea level. This thesis considers fixed I_{sp} systems, but is applicable to variable specific impulse (VSI) systems. Mass flow rate may also be modeled at a higher fidelity with a polynomial function of the power available.

2.1.1   Power Modeling

The thrust available is a function of the power available to the propulsion system. A number of power models have been implemented, the simplest of which assumes sufficient power to provide constant thrust available at the maximum value, $T_a = T_{max}$. Otherwise, the thrust available is determined by a user-specified duty cycle D, engine efficiency η and power available P_a.

$$T_a = \frac{2D\eta P_a}{I_{sp}g_0} \tag{2.7}$$

The next degree of fidelity is obtained with an inverse-square solar power model, where power available via spacecraft solar panels decays with the square of heliocentric distance.

$$P_a = \frac{AU^2}{r_{sc/\odot}^2}P_0 \tag{2.8}$$

In Equation 2.8, P_0 is the reference power at one astronomical unit AU. Subscripts are organized so that $\boldsymbol{r}_{sc/\odot}$ is the position of the spacecraft with respect to the Sun, though in this instance the direction is arbitrary. Higher-order polynomials and time dependency may be used for added fidelity. Furthermore, η and I_{sp} are generally not constant, but functions of the input power. Real-world thrusters have discrete throttle points that give T and $\dot{m}$ directly for a discrete input power setting. Accurately capturing the true operating points with a power model is essential for the low-thrust trajectory optimization process to be meaningful [63].

2.1.2   Control Variables

Gradient-based optimization like differential dynamic programming requires derivatives of the mass flow rate with respect to the Cartesian control variables, but those are singular for coast arcs.

This can be resolved by including a "fudge" factor ϵ_T in the norm of the control magnitude [64], which is effectively a mass leak.

$$u = \sqrt{u_x^2 + u_y^2 + u_z^2 + \epsilon_T} \tag{2.9}$$

It is often more effective to use a spherical control representation in a coordinate system that changes with the motion of the spacecraft. Spherical thrust control is defined by magnitude u, yaw angle α and pitch angle β, where the angles are defined relative to the radial-transverse-normal (RSW) frame. RSW basis vectors and the rotation to the inertial frame are defined by

$$\begin{bmatrix} \hat{r} & \hat{s} & \hat{w} \end{bmatrix} = \begin{bmatrix} \dfrac{r}{r} & \dfrac{(r \times v) \times r}{\|(r \times v) \times r\|} & \dfrac{r \times v}{\|r \times v\|} \end{bmatrix}. \tag{2.10}$$

Thrust vector components are then

$$\begin{bmatrix} u_r \\ u_s \\ u_w \end{bmatrix} = \begin{bmatrix} u \sin\alpha \cos\beta \\ u \cos\alpha \cos\beta \\ u \sin\beta \end{bmatrix}, \quad \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \hat{r} & \hat{s} & \hat{w} \end{bmatrix} \begin{bmatrix} u_r \\ u_s \\ u_w \end{bmatrix}, \tag{2.11}$$

so that the pitch angle is measured from the orbit plane about the radial direction and the yaw angle is measured from the transverse direction about the angular momentum $\hat{h} = \hat{w}$. No concern is given for angle wrapping. In fact, computation exhibits favorable performance when the angles are unbounded.

An alternative reference frame is the velocity-normal-binormal (VNB) frame defined as

$$\begin{bmatrix} \hat{v} & \hat{n} & \hat{b} \end{bmatrix} = \begin{bmatrix} \dfrac{v}{v} & \dfrac{r \times v}{\|r \times v\|} & \dfrac{v \times (r \times v)}{\|v \times (r \times v)\|} \end{bmatrix}. \tag{2.12}$$

VNB control components are

$$\begin{bmatrix} u_v \\ u_n \\ u_b \end{bmatrix} = \begin{bmatrix} u \cos\alpha \cos\beta \\ u \sin\beta \\ u \sin\alpha \cos\beta \end{bmatrix}, \quad \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \hat{v} & \hat{n} & \hat{b} \end{bmatrix} \begin{bmatrix} u_v \\ u_n \\ u_b \end{bmatrix}, \tag{2.13}$$

so that the pitch angle is measured from the orbit plane about the binormal direction and the yaw angle is measured from the velocity direction about the angular momentum $\hat{h} = \hat{n} = \hat{w}$.

With the RSW and VNB spherical control representations, derivatives of the mass flow rate are taken with respect to u, α and β, not with respect to components of u. Singularities of the mass flow rate derivative are avoided and no mass leak is required.

2.2   Modified Equinoctial Elements

Orbit element sets are often preferable to a Cartesian description of the spacecraft state. Walker, Ireland, and Owens introduced the modified equinoctial elements as a set of orbital elements that are suitable for perturbation analysis of all types of orbits [65]. The equations of motion are nonsingular except in the case of $180°$ inclination, which is easily avoided by redefining the reference plane. The five *slow variables* describe the shape and orientation of the orbital plane, and are constant for two-body motion or change slowly in the presence of perturbations such as low-thrust. The *fast variable* identifies a location within the orbit and is chosen as the true longitude.

$$p = a\left(1 - e^2\right) \tag{2.14a}$$

$$f = e\cos(\omega + \Omega) \tag{2.14b}$$

$$g = e\sin(\omega + \Omega) \tag{2.14c}$$

$$h = \tan\left(\frac{i}{2}\right)\cos\Omega \tag{2.14d}$$

$$k = \tan\left(\frac{i}{2}\right)\sin\Omega \tag{2.14e}$$

$$L = \omega + \Omega + \nu \tag{2.14f}$$

Equation 2.14 states the modified equinoctial elements in terms of the classical orbital elements, where a is the semi-major axis, e is the eccentricity, ω is the argument of periapsis, Ω is the right ascension of the ascending node, i is the inclination and ν is the true anomaly. The equations of motion are stated following Betts' presentation in Reference [34]. First, several auxiliary terms are defined.

$$q = 1 + f\cos L + g\sin L \tag{2.15a}$$

$$r = \frac{p}{q} \tag{2.15b}$$

$$\alpha^2 = h^2 - k^2 \tag{2.15c}$$

$$\mathcal{X} = \sqrt{h^2 + k^2} \tag{2.15d}$$

$$s^2 = 1 + \mathcal{X}^2 \tag{2.15e}$$

The equations of motion are then stated as,

$$\dot{\boldsymbol{x}} = A(\boldsymbol{x})\boldsymbol{\Delta} + \boldsymbol{b}, \tag{2.16}$$

where

$$\boldsymbol{x} = \begin{bmatrix} p & f & g & h & k & L \end{bmatrix}^T, \tag{2.17}$$

$$A = \begin{bmatrix} 0 & \frac{2p}{q}\sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}}\sin L & \sqrt{\frac{p}{\mu}}\frac{1}{q}\left[(q+1)\cos L + f\right] & -\sqrt{\frac{p}{\mu}}\frac{g}{q}\left(h\sin L - k\cos L\right) \\ -\sqrt{\frac{p}{\mu}}\cos L & \sqrt{\frac{p}{\mu}}\frac{1}{q}\left[(q+1)\sin L + g\right] & \sqrt{\frac{p}{\mu}}\frac{f}{q}\left(h\sin L - k\cos L\right) \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\cos L}{2q} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\sin L}{2q} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{1}{q}\left(h\sin L - k\cos L\right) \end{bmatrix}, \tag{2.18}$$

$$\boldsymbol{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p}\left(\frac{q}{p}\right)^2 \end{bmatrix}^T, \tag{2.19}$$

and $\boldsymbol{\Delta}$ are perturbing accelerations in the radial-transverse-normal (RSW) frame. RSW basis vectors and thus the rotation to the inertial frame are defined by,

$$Q = \begin{bmatrix} \hat{\boldsymbol{r}} & \hat{\boldsymbol{s}} & \hat{\boldsymbol{w}} \end{bmatrix}, \tag{2.20}$$

which is a restatement of Equation 2.11. For perturbations $\boldsymbol{\delta}$ in the inertial frame, the necessary transformation is

$$\boldsymbol{\Delta} = Q^T \boldsymbol{\delta}. \tag{2.21}$$

The Cartesian position and velocity in terms of the modified equinoctial elements are

$$\boldsymbol{r} = \begin{bmatrix} \frac{r}{s^2}\left[(1+\alpha^2)\cos L + 2hk\sin L\right] \\ \frac{r}{s^2}\left[(1-\alpha^2)\sin L + 2hk\cos L\right] \\ \frac{2r}{s^2}\left(h\sin L - k\cos L\right) \end{bmatrix} \tag{2.22a}$$

$$\boldsymbol{v} = \begin{bmatrix} -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left[(1+\alpha^2)\left(\sin L + g\right) - 2hk\left(\cos L + f\right)\right] \\ -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left[(\alpha^2-1)\left(\cos L + f\right) + 2hk\left(\sin L + g\right)\right] \\ \frac{2}{s^2}\sqrt{\frac{\mu}{p}}\left[h\left(\cos L + f\right) + k\left(\sin L + g\right)\right] \end{bmatrix}. \tag{2.22b}$$

In the absence of perturbations, the slow variables are constant and only the true longitude changes. The true longitude rates of change assembled through Equations 2.16 to 2.19 identify a Sundman transformation [66],

$$dt = \left[\sqrt{\frac{\mu}{p}} \frac{q}{h \sin L - k \cos L} \boldsymbol{\Delta}_w + \frac{1}{\sqrt{\mu p}} \left(\frac{p}{q} \right)^2 \right] dL, \tag{2.23}$$

where the perturbation term reflects changes to ω and Ω, as subscript w implies the $\hat{\boldsymbol{w}}$ component of $\boldsymbol{\Delta}$. The second summed term is the Sundman transformation to true anomaly,

$$dt = \frac{1}{\sqrt{\mu p}} \left(\frac{p}{q} \right)^2 d\nu. \tag{2.24}$$

2.3   The Circular Restricted Three-Body Problem

A simplified model for spacecraft motion in the presence of two massive bodies is the circular restricted three-body problem (CRTBP). In the CRTBP, the spacecraft mass is assumed to be negligible in comparison to that of the primary and secondary massive bodies, with masses m_1 and m_2, respectively. The bodies are named such that $m_1 > m_2$. The system is defined by the parameter μ that is the ratio of the secondary mass to the total system mass.

$$\mu = \frac{m_2}{m_1 + m_2} \tag{2.25}$$

Next, it is assumed that the massive bodies obey Keplerian motion in circular orbits about their barycenter. Then it is useful to model the spacecraft dynamics in the synodic frame that follows the rotation of the system.

2.3.1   Spacecraft State and Dynamics

The spacecraft state is chosen as a Cartesian representation of the spacecraft position and velocity in the synodic frame and the spacecraft mass.

$$\boldsymbol{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T, \quad \boldsymbol{v} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{r}^T & \boldsymbol{v}^T & m \end{bmatrix}^T \tag{2.26}$$

The position vector $\boldsymbol{r}$ is measured from the system barycenter. The massive bodies are fixed to the x-axis with the primary in the negative x-direction and secondary in the positive x-direction. The synodic frame rotates about the barycenter with the system angular velocity,

$$\boldsymbol{\omega} = \sqrt{\frac{\mu_1 + \mu_2}{R^3}}\,\hat{\boldsymbol{z}}, \tag{2.27}$$

where μ_1 and μ_2 are the gravitational parameters of the massive bodies that are separated by a distance R. The distance from the barycenter to the primary is $-\mu R$ and the distance from the barycenter to the secondary is $(1-\mu)R$. These distances are the shift in the x-coordinate for the spacecraft position vector from each body.

$$\boldsymbol{r}_1 = \begin{bmatrix} x + \mu R & y & z \end{bmatrix}^T, \quad \boldsymbol{r}_2 = \begin{bmatrix} x - (1-\mu)R & y & z \end{bmatrix}^T \tag{2.28}$$

For spherical thrust control in the CRTBP, an alternative RSW frame is suggested with the radial direction taken from the secondary. This alternative is to facilitate trajectory design in the vicinity of the secondary. For trajectory design throughout the entire state space, measuring the radial direction from the barycenter or primary might prove to be more robust. The updated RSW basis vectors and the rotation to the synodic frame are defined by

$$\begin{bmatrix} \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{s}} & \hat{\boldsymbol{w}} \end{bmatrix} = \begin{bmatrix} \dfrac{\boldsymbol{r}_2}{r_2} & \dfrac{(\boldsymbol{r}_2 \times \boldsymbol{v}) \times \boldsymbol{r}_2}{\|(\boldsymbol{r}_2 \times \boldsymbol{v}) \times \boldsymbol{r}_2\|} & \dfrac{\boldsymbol{r}_2 \times \boldsymbol{v}}{\|\boldsymbol{r}_2 \times \boldsymbol{v}\|} \end{bmatrix}. \tag{2.29}$$

The mapping from control variables to Cartesian thrust vector components is given by

$$\begin{bmatrix} u_r \\ u_s \\ u_w \end{bmatrix} = \begin{bmatrix} u \sin\alpha \cos\beta \\ u \cos\alpha \cos\beta \\ u \sin\beta \end{bmatrix}, \quad \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{r}}_2 & \hat{\boldsymbol{s}} & \hat{\boldsymbol{w}} \end{bmatrix} \begin{bmatrix} u_r \\ u_s \\ u_w \end{bmatrix}. \tag{2.30}$$

It is possible for $\boldsymbol{r}_2$ to be collinear with the velocity $\boldsymbol{v}$ so that the RSW frame is undefined. In those cases the Cartesian thrust vector components are used as the control variables as in Equation 2.4.

A constant power model is assumed with $T_a = T_{max}$. The CRTBP accelerations including

thrust perturbations may then be computed.

$$\ddot{x} = 2\omega\dot{y} + \omega^2 x - \frac{\mu_1}{r_1^3}(x + \mu R) - \frac{\mu_2}{r_2^3}[x - (1 - \mu)R] + \frac{T_a}{m}u_x, \tag{2.31a}$$

$$\ddot{y} = -2\omega\dot{x} + \omega^2 y - \frac{\mu_1}{r_1^3}y - \frac{\mu_2}{r_2^3}y + \frac{T_a}{m}u_y, \tag{2.31b}$$

$$\ddot{z} = -\frac{\mu_1}{r_1^3}z - \frac{\mu_2}{r_2^3}z + \frac{T_a}{m}u_z. \tag{2.31c}$$

It is common to normalize the CRTBP so that the system mass, angular velocity and distance between the primaries are all equal to one. A distance unit DU, time unit TU, mass unit MU and force unit FU are defined to nondimensionalize all quantities.

$$DU = R \tag{2.32a}$$

$$TU = \omega^{-1} \tag{2.32b}$$

$$MU = m_0 \tag{2.32c}$$

$$FU = MU\frac{DU}{TU^2} \tag{2.32d}$$

Note that the definition of the mass unit as the initial spacecraft mass m_0 conflicts with a system mass equal to one. However, as the thrust perturbations are uncoupled from the CRTBP accelerations, mass and thrust can be scaled separately using MU and FU. Then the nondimensional mass and thrust take on reasonable values that are favorable for optimization performance while still producing correctly scaled accelerations. The nondimensional CRTBP equations of motion with thrust become

$$\ddot{x} = 2\dot{y} + x - \frac{1 - \mu}{r_1^3}(x + \mu) - \frac{\mu}{r_2^3}(x - 1 - \mu) + \frac{T_a}{m}u_x, \tag{2.33a}$$

$$\ddot{y} = -2\dot{x} + y - \frac{1 - \mu}{r_1^3}y - \frac{\mu}{r_2^3}y + \frac{T_a}{m}u_y, \tag{2.33b}$$

$$\ddot{z} = -\frac{1 - \mu}{r_1^3}z - \frac{\mu}{r_2^3}z + \frac{T_a}{m}u_z. \tag{2.33c}$$

2.3.2   The Jacobi Constant

The CRTBP equations of motion may be expressed using the gradient of a pseudo-potential U, which is a function of position and the system parameter.

$$U = \frac{1}{2}\left(x^2 + y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2.34}$$

In the absence of perturbations, the equations of motion are stated as,

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x}, \tag{2.35a}$$

$$\ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y}, \tag{2.35b}$$

$$\ddot{z} = \frac{\partial U}{\partial z}. \tag{2.35c}$$

The time derivative of the pseudo-potential can be stated using the chain rule,

$$\frac{\partial U}{\partial t} = \left(\frac{\partial U}{\partial \boldsymbol{r}}\right)^T \boldsymbol{v}, \tag{2.36}$$

which is the the dot product of the vector form of the right-hand side of Equation 2.35 with the velocity. Similarly doing so with the left-hand side yields

$$\frac{\partial U}{\partial t} = \dot{\boldsymbol{v}}^T \boldsymbol{v}. \tag{2.37}$$

Integrating Equation 2.37 results in a constant of integration named the Jacobi constant C.

$$C = 2U - v^2 \tag{2.38}$$

The Jacobi constant is an integral of motion in the CRTBP. It is constant for ballistic trajectories but changes with perturbations such as low-thrust propulsion. Unperturbed motion is limited to regions where the value of the Jacobi constant is attainable. Permissible regions are bounded by zero-velocity curves where Equation 2.38 reduces to $C = 2U$. Figure 2.1 sketches out zero-velocity curves in the Earth-Moon CRTBP described in Table 2.1. Forbidden regions beyond zero-velocity curves require imaginary velocity components to maintain the Jacobi constant.
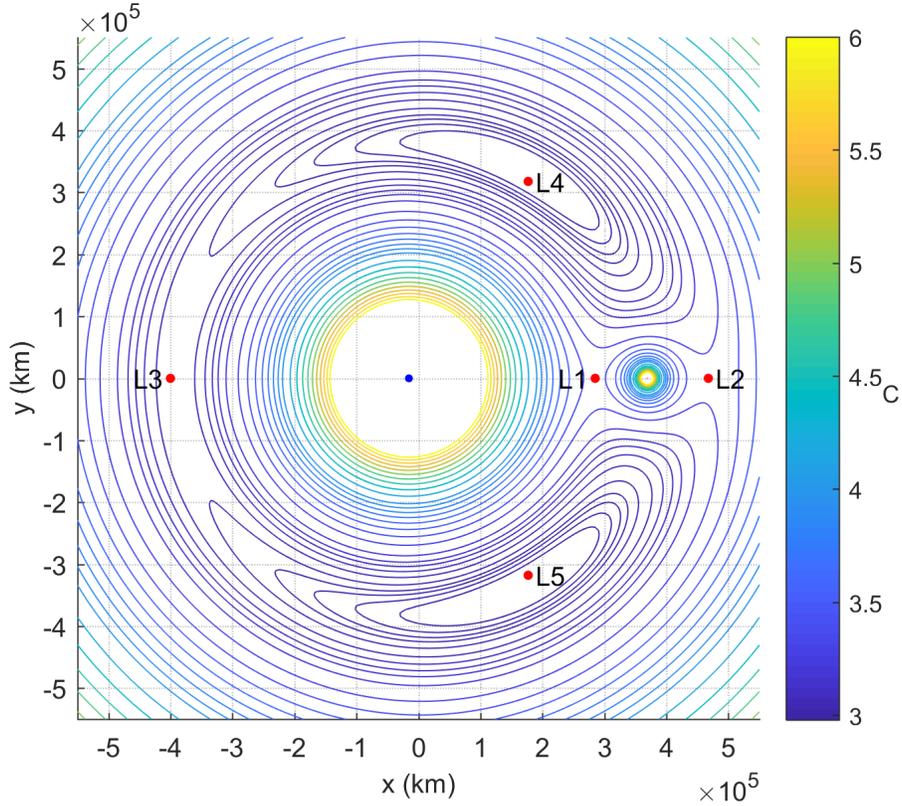
Figure 2.1: Zero-velocity curves in the xy-plane of the Earth-Moon CRTBP synodic frame are colored according to the Jacobi constant. The five Lagrange points are labeled and a spherical Earth is plotted to scale. Zero-velocity curves with $C > 6$ were removed.

Table 2.1: Earth-Moon CRTBP parameters.

μ	0.012004715741012
R	384747.962856037 km
ω^{-1}	375727.551633535 sec

2.3.3 Lagrange Points

Figure 2.1 also identifies five libration points, or Lagrange points, named L1 through L5. These are the five equilibrium points of the CRTBP, where gravitational and centripetal accelerations balance. Objects with zero velocity in the synodic frame will remain fixed at these locations in the synodic frame. Zero-velocity curves provide visual insight into the stability of equilibrium at the Lagrange points. The co-linear Lagrange points L1, L2 and L3 are located at saddle points. Equilibrium is unstable and a small perturbation can lead to rapid departure from this region. Fi-

gure 2.1 shows closed zero-velocity curves nearby the triangular Lagrange points L4 and L5, which are stable for $\mu < 0.0385$.

Chapter 3

Differential Dynamic Programming

This chapter is a restatement of the DDP algorithm presented by Lantoine and Russell in Reference [46]. Departures from their presentation include expanded use of the augmented state vector and introduction of tensor notation. Additionally, algorithmic options related to the current body of work are identified.

3.1   Notation

The DDP algorithm consists of scalar and vector equations that include operations on matrices and third-order tensors. Careful bookkeeping is required to manage the notational difficulties. In the following presentation, boldface is reserved for column vectors. Matrices and tensors are otherwise identified in the text. Subscripts may be used for indexing or to denote the derivative. For example,

$$J_x = \frac{\partial J}{\partial \boldsymbol{x}} \tag{3.1}$$

is the derivative of the cost J with respect to state vector $\boldsymbol{x}$. With an additional subscript, $J_{x,k}$ is the gradient evaluated at stage k, but J_k is just the scalar quantity at stage k. If $\boldsymbol{x}$ is an $n \times 1$ vector, then J_x is also an $n \times 1$ vector. Note however, the absence of boldface. The symmetric $n \times n$ Hessian matrix of second derivatives is then,

$$J_{xx} = \frac{\partial^2 J}{\partial \boldsymbol{x}^2} \ . \tag{3.2}$$

Cross partial derivatives are organized so that J_{ux} is an $m \times n$ matrix for $\boldsymbol{u}$ size $m \times 1$. Derivatives of vectors are similarly defined, where $\boldsymbol{f}_x$ is $p \times n$ and $\boldsymbol{f}_{ux}$ is a rank-3 tensor of size $p \times m \times n$, for $\boldsymbol{f}$ size $p \times 1$, for example.

3.1.1   Tensor Notation

Tensor notation prevents ambiguities of mathematical operations between tensors, matrices, vectors, and scalars. The adopted convention uses superscripts as indices. The rank of an object is implied by the number of indices. For example, $\boldsymbol{x}^i$ is the i-th element of state vector $\boldsymbol{x}$. First and second derivatives of the state dynamics are stated as

$$A^{i,a} = \frac{\partial \dot{\boldsymbol{x}}^i}{\partial \boldsymbol{x}^a},$$ (3.3a)

$$A^{i,ab} = \frac{\partial^2 \dot{\boldsymbol{x}}^i}{\partial \boldsymbol{x}^a \partial \boldsymbol{x}^b}.$$ (3.3b)

$A^{i,a}$ is the (i, a) entry of the dynamics matrix and is the derivative of the i-th element of state dynamics vector $\dot{\boldsymbol{x}}$ taken with respect to the a-th element of $\boldsymbol{x}$. $A^{i,ab}$ is the (i, a, b) entry of the rank three dynamics tensor that is the cross partial derivative, or second derivative if $a = b$, of the i-th element of $\dot{\boldsymbol{x}}$ with respect to the a-th and b-th elements of $\boldsymbol{x}$.

Following Einstein notation, multiplication is performed by summing over repeated γ indices. The familiar differential equation for the state transition matrix, $\dot{\Phi} = A\Phi$, is the product of two matrices and otherwise written as

$$\dot{\Phi}^{i,a} = A^{i,\gamma_1} \Phi^{\gamma_1,a}.$$ (3.4)

The differential equation for the second-order state transition tensor is

$$\dot{\Phi}^{i,ab} = A^{i,\gamma_1} \Phi^{\gamma_1,ab} + A^{i,\gamma_1 \gamma_2} \Phi^{\gamma_1,a} \Phi^{\gamma_2,b}.$$ (3.5)

The presence of multiple repeated indices implies multiple summations, e.g. for $\boldsymbol{x}$ with n state variables, $A^{i,\gamma_1 \gamma_2} \Phi^{\gamma_1,a} \Phi^{\gamma_2,b}$ is a double summation over $\gamma_1 = 0, 1, ..., n$, $\gamma_2 = 0, 1, ..., n$.

3.2   Introduction

DDP seeks the minimum of a cost functional $J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, t)$, where $\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}, \boldsymbol{w}, t)$ is a state trajectory, $\boldsymbol{u}(t)$ is a control schedule, and $\boldsymbol{w}$ is a constant parameter vector. Parameters influence the initial state through the initial function, $\boldsymbol{x}_0 = \boldsymbol{\Gamma}(\boldsymbol{w})$. Nominal controls $\overline{\boldsymbol{u}}(t)$ and parameters $\overline{\boldsymbol{w}}$ are suggested as an initial guess and produce a nominal state trajectory $\overline{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}_0, \overline{\boldsymbol{u}}, \overline{\boldsymbol{w}}, t)$. The control and parameters are iteratively updated by applying $\delta\boldsymbol{u}(t)$ and $\delta\boldsymbol{w}$ until optimality conditions are satisfied. It is the $\delta\boldsymbol{u}$ and $\delta\boldsymbol{w}$ that are the optimization variables such that the trial control is $\boldsymbol{u} = \overline{\boldsymbol{u}} + \delta\boldsymbol{u}$, and the trial parameters are $\boldsymbol{w} = \overline{\boldsymbol{w}} + \delta\boldsymbol{w}$. The new state that results from the updated control is similarly described by a deviation from the nominal trajectory, $\boldsymbol{x} = \overline{\boldsymbol{x}} + \delta\boldsymbol{x}$.

3.3   Forward Pass

Evaluating $\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}_0, \overline{\boldsymbol{u}} + \delta\boldsymbol{u}, \overline{\boldsymbol{w}} + \delta\boldsymbol{w}, t)$ with trial controls and parameters constitutes a *forward pass*, and is the trajectory computation step of the algorithm. For the first iteration, $\overline{\boldsymbol{u}}$ is the initial guess of controls and $\delta\boldsymbol{u} = \boldsymbol{0}$. Similarly, $\overline{\boldsymbol{w}}$ is the initial guess of parameters and $\delta\boldsymbol{w} = \boldsymbol{0}$. This work considers a discrete problem formulation where a trajectory is described by M phases and each phase i is divided into N_i stages. Control variables can be updated at the beginning of each stage, and parameters at the beginning of each phase. Subscripts are organized so that $\boldsymbol{x}_{i,k}$ is the state at stage k of phase i. The forward pass is then the sequence of function evaluations,

$$\boldsymbol{x}_{i,k+1} = \boldsymbol{f}_{i,k}(\boldsymbol{x}_{i,k}, \boldsymbol{u}_{i,k}, \boldsymbol{w}_i),$$

$$\boldsymbol{x}_{i,0} = \boldsymbol{\Gamma}_i(\boldsymbol{w}_i),$$

$$k = 0, 1, ..., N_i - 1,$$

$$i = 0, ..., M - 1.$$

(3.6)

The transition function $\boldsymbol{f}$ dictates how the state evolves between stages, and might obey a system of linear, nonlinear, or differential equations, and is not necessarily deterministic. DDP is applicable to all of these systems in both continuous and discrete form [35]. Stages in DDP represent sampling of continuous variables, so that the transition function is the integral of the equations of motion.

The relevant transition function is

$$\boldsymbol{x}_{i,k+1} = \boldsymbol{x}_{i,k} + \int\limits_{t_{i,k}}^{t_{i,k+1}} \dot{\boldsymbol{x}}_{i,k}\left(\boldsymbol{x}, \boldsymbol{u}_{i,k}, \boldsymbol{w}_i, t\right) dt, \tag{3.7}$$

where the overhead dot denotes the time derivative, $\boldsymbol{x}$ is a dependent variable and $\boldsymbol{u}_{i,k}$ and $\boldsymbol{w}_i$ are specified for the stage and phase.

The cost can be determined upon reaching the final stage of the final phase, N_M. Nominal states, controls and parameters are updated for successful iterations, so that $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})$ becomes the new $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{w}})$. If optimality conditions are satisfied, then the procedure is finished. Otherwise, control and parameter updates are computed in a *backward sweep*.

3.4   Augmented Lagrangian Method

The standard DDP formulation adjoins terminal constraints to the original cost function using a constant vector of Lagrange multipliers [35]. The additional use of a scalar penalty parameter places additional weight on terminal constraint violations. Here a penalty matrix is used so that the additional weight on each constraint may be treated individually.

$$\widetilde{\varphi}_i = \varphi_i + \boldsymbol{\lambda}_i^T \boldsymbol{\psi}_i + \boldsymbol{\psi}_i^T \Sigma_i \boldsymbol{\psi}_i \tag{3.8}$$

Equation 3.8 is the augmented Lagrangian cost function of phase i, and the objective function is the total cost summed over all M phases.

$$J = \sum_{i=0}^{M-1} \left[\sum_{j=0}^{N_i} (L_{i,j}) + \widetilde{\varphi}_i \right] \tag{3.9}$$

Each phase cost incurs the sum of stage cost functions $L_{i,j}(\boldsymbol{x}_{i,j}, \boldsymbol{u}_{i,j}, \boldsymbol{w}_i)$. The original objective to be minimized at each phase is $\varphi_i(\boldsymbol{x}_{i,N_i}, \boldsymbol{w}_i, \boldsymbol{x}_{i+1,0}, \boldsymbol{w}_{i+1})$. Terminal constraints are now generally phase constraints in the multi-phase formulation,

$$\boldsymbol{\psi}_i\left(\boldsymbol{x}_{i,N_i}, \boldsymbol{w}_i, \boldsymbol{x}_{i+1,0}, \boldsymbol{w}_{i+1}\right) = 0. \tag{3.10}$$

Multipliers $\boldsymbol{\lambda}_i$ are initialized as $\overline{\boldsymbol{\lambda}}_i = \boldsymbol{0}$ and updated at every iteration by $\boldsymbol{\lambda}_i = \overline{\boldsymbol{\lambda}}_i + \delta\boldsymbol{\lambda}_i$ to push the trajectory toward feasibility. Penalty matrix Σ places additional weight on constraint violations and

serves to initialize a quadratic cost function. In contrast to previous approaches that continually increase the penalty weight [46, 59] Σ is held constant for all iterations. In practice, the entries of Σ are tuned after observing how the iterates progress toward feasibility. For example, an initial attempt to optimize a trajectory might begin with Σ as a scalar multiple of the identity matrix so that each constraint is weighted equally. If iterates show little progress toward satisfying a particular constraint, its associated Σ entry could be increased and the process restarted. Similarly, if the algorithm appears to prioritize a constraint without working to satisfy the other constraints or to improve the objective, the Σ entry for that prioritized constraint could be reduced. The initial guess of zero-valued multipliers is not a requirement and may also be viewed as tuning parameters.

3.5   Backward Sweep

The sequential application of transition functions in the forward pass in Equation 3.6 and the divisibility of the augmented Lagrangian cost function in Equation 3.9 allow the dynamic programming principle (Bellman's principle of optimality in Section 1.3) to apply backwards across all phases and stages. For simplicity, phase subscripts are dropped and the backward sweep is presented for an arbitrary phase of N stages before the inter-phase steps are addressed.

3.5.1   Stage Subproblems

The backward sweep solves the sequence of subproblems that minimize the cost-to-go from stage $k = N - 1, N - 2, ..., 0$.

$$J_k^* = \min_{\delta \boldsymbol{u}_k} [J_k] \tag{3.11}$$

The solution to Equation 3.11 is the optimal control update $\delta \boldsymbol{u}_k^*$. By assuming that the cost function can be approximated by a second-order Taylor series expansion about the nominal states,

controls, and multipliers, a prediction for $\delta \boldsymbol{u}_k^*$ is readily available.

$$
\begin{aligned}
J_k(\overline{\boldsymbol{x}}_k + \delta \boldsymbol{x}_k, \overline{\boldsymbol{u}}_k + \delta \boldsymbol{u}_k, \overline{\boldsymbol{w}} + \delta \boldsymbol{w}, \overline{\boldsymbol{\lambda}} + \delta \boldsymbol{\lambda}) \approx\ & ER_{k+1} + J_k(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{u}}_k, \overline{\boldsymbol{w}}, \overline{\boldsymbol{\lambda}}) \\
& + J_{x,k}^T \delta \boldsymbol{x}_k + J_{u,k}^T \delta \boldsymbol{u}_k + J_{w,k}^T \delta \boldsymbol{w} + J_{\lambda,k}^T \delta \boldsymbol{\lambda} \\
& + \tfrac{1}{2} \delta \boldsymbol{x}_k^T J_{xx,k} \delta \boldsymbol{x}_k + \tfrac{1}{2} \delta \boldsymbol{u}_k^T J_{uu,k} \delta \boldsymbol{u}_k + \tfrac{1}{2} \delta \boldsymbol{w}^T J_{ww,k} \delta \boldsymbol{w} + \tfrac{1}{2} \delta \boldsymbol{\lambda}^T J_{\lambda\lambda,k} \delta \boldsymbol{\lambda} \\
& + \delta \boldsymbol{x}_k^T J_{xu,k} \delta \boldsymbol{u}_k + \delta \boldsymbol{x}_k^T J_{xw,k} \delta \boldsymbol{w} + \delta \boldsymbol{x}_k^T J_{x\lambda,k} \delta \boldsymbol{\lambda} \\
& + \delta \boldsymbol{u}_k^T J_{uw,k} \delta \boldsymbol{w} + \delta \boldsymbol{u}_k^T J_{u\lambda,k} \delta \boldsymbol{\lambda} + \delta \boldsymbol{w}^T J_{w\lambda,k} \delta \boldsymbol{\lambda}
\end{aligned}
\tag{3.12}
$$

The expected reduction ER_{k+1} accounts for the predicted change in cost that accumulates with the solution to each subproblem. The left-hand side of Equation 3.12 represents the cost-to-go along a neighboring trajectory that differs from the nominal cost-to-go by δJ_k.

$$
\delta J_k = J_k(\overline{\boldsymbol{x}}_k + \delta \boldsymbol{x}_k, \overline{\boldsymbol{u}}_k + \delta \boldsymbol{u}_k, \overline{\boldsymbol{w}} + \delta \boldsymbol{w}, \overline{\boldsymbol{\lambda}} + \delta \boldsymbol{\lambda}) - J_k(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{u}}_k, \overline{\boldsymbol{w}}, \overline{\boldsymbol{\lambda}})
\tag{3.13}
$$

The quadratic model is then restated as

$$
\begin{aligned}
\delta J_k \approx\ & ER_{k+1} + J_{x,k}^T \delta \boldsymbol{x}_k + J_{u,k}^T \delta \boldsymbol{u}_k + J_{w,k}^T \delta \boldsymbol{w} + J_{\lambda,k}^T \delta \boldsymbol{\lambda} \\
& + \tfrac{1}{2} \delta \boldsymbol{x}_k^T J_{xx,k} \delta \boldsymbol{x}_k + \tfrac{1}{2} \delta \boldsymbol{u}_k^T J_{uu,k} \delta \boldsymbol{u}_k + \tfrac{1}{2} \delta \boldsymbol{w}^T J_{ww,k} \delta \boldsymbol{w} + \tfrac{1}{2} \delta \boldsymbol{\lambda}^T J_{\lambda\lambda,k} \delta \boldsymbol{\lambda} \\
& + \delta \boldsymbol{x}_k^T J_{xu,k} \delta \boldsymbol{u}_k + \delta \boldsymbol{x}_k^T J_{xw,k} \delta \boldsymbol{w} + \delta \boldsymbol{x}_k^T J_{x\lambda,k} \delta \boldsymbol{\lambda} \\
& + \delta \boldsymbol{u}_k^T J_{uw,k} \delta \boldsymbol{w} + \delta \boldsymbol{u}_k^T J_{u\lambda,k} \delta \boldsymbol{\lambda} + \delta \boldsymbol{w}^T J_{w\lambda,k} \delta \boldsymbol{\lambda}\ .
\end{aligned}
\tag{3.14}
$$

The minimizing control update is found by taking the derivative of Equation 3.14 with respect to $\delta \boldsymbol{u}_k$ and setting it equal to zero. The result is an unconstrained feedback control law,

$$
\begin{aligned}
\delta \boldsymbol{u}_k^* &= A_k + B_k \delta \boldsymbol{x}_k + C_k \delta \boldsymbol{w} + D_k \delta \boldsymbol{\lambda} \\
A_k &= -J_{uu,k}^{-1} J_{u,k} \\
B_k &= -J_{uu,k}^{-1} J_{ux,k} \\
C_k &= -J_{uu,k}^{-1} J_{uw,k} \\
D_k &= -J_{uu,k}^{-1} J_{u\lambda,k}
\end{aligned}
\tag{3.15}
$$

The feed-forward terms A_k and feedback gains B_k, C_k and D_k are stored to assemble $\delta \boldsymbol{u}_k^*$ during the forward pass. For a quadratic cost function and linear system, the model and minimization

are exact and DDP exhibits one-step convergence. Algorithmic options from HDDP permit the practical application of Equation 3.15, namely the enforcement of control bounds by a null-space method [47]. A trust-region method restricts the size of A_k, $\delta \boldsymbol{w}$ and $\delta \boldsymbol{\lambda}$ so that the resulting $\delta \boldsymbol{x}_k$ remains within the valid region of the quadratic model [67]. Furthermore, the trust-region method ensures that $J_{uu,k}$ is positive definite so that $\delta \boldsymbol{u}_k^*$ is a descent direction.

Derivatives present in the new control law are obtained by recognizing that the cost-to-go is the sum of the local cost and the optimal cost-to-go from the next stage.

$$J_k = L_k + J_{k+1}^* \tag{3.16}$$

The required derivatives are the Taylor coefficients in Equation 3.14, that is the expansion of the left-hand side of Equation 3.16. Their counterparts from expanding the right-hand side of Equation 3.16 should be equivalent.

$$
\begin{aligned}
\delta L_k \approx{} & L_{x,k}^T \delta \boldsymbol{x}_k + L_{u,k}^T \delta \boldsymbol{u}_k + L_{w,k}^T \delta \boldsymbol{w} + \frac{1}{2} \delta \boldsymbol{x}_k^T L_{xx,k} \delta \boldsymbol{x}_k + \frac{1}{2} \delta \boldsymbol{u}_k^T L_{uu,k} \delta \boldsymbol{u}_k \\
& + \frac{1}{2} \delta \boldsymbol{w}^T L_{ww,k} \delta \boldsymbol{w} + \delta \boldsymbol{x}_k^T L_{xu,k} \delta \boldsymbol{u}_k + \delta \boldsymbol{x}_k^T L_{xw,k} \delta \boldsymbol{w} + \delta \boldsymbol{u}_k^T L_{uw,k} \delta \boldsymbol{w}
\end{aligned}
\tag{3.17a}
$$

$$
\begin{aligned}
\delta J_{k+1}^* \approx{} & ER_{k+1} + J_{x,k+1}^{*T} \delta \boldsymbol{x}_{k+1} + J_{w,k+1}^{*T} \delta \boldsymbol{w} + J_{\lambda,k+1}^{*T} \delta \boldsymbol{\lambda} + \frac{1}{2} \delta \boldsymbol{x}_{k+1}^T J_{xx,k+1}^* \delta \boldsymbol{x}_{k+1} \\
& + \frac{1}{2} \delta \boldsymbol{w}^T J_{ww,k+1}^* \delta \boldsymbol{w} + \frac{1}{2} \delta \boldsymbol{\lambda}^T J_{\lambda\lambda,k+1}^* \delta \boldsymbol{\lambda} + \delta \boldsymbol{x}_{k+1}^T J_{xw,k+1}^* \delta \boldsymbol{w} \\
& + \delta \boldsymbol{x}_{k+1}^T J_{x\lambda,k+1}^* \delta \boldsymbol{\lambda} + \delta \boldsymbol{w}^T J_{w\lambda,k+1}^* \delta \boldsymbol{\lambda}
\end{aligned}
\tag{3.17b}
$$

Derivatives in Equation 3.17a are obtained directly while those in Equation 3.17b are known from the preceding subproblem. Equating Taylor coefficients of like order requires an expression for $\delta \boldsymbol{x}_{k+1}$ as a function of $\delta \boldsymbol{x}_k$, $\delta \boldsymbol{u}_k$, $\delta \boldsymbol{w}$ and $\delta \boldsymbol{\lambda}$. By definition, that relationship is

$$\delta \boldsymbol{x}_{k+1} = \boldsymbol{f}(\overline{\boldsymbol{x}}_k + \delta \boldsymbol{x}_k, \overline{\boldsymbol{u}}_k + \delta \boldsymbol{u}_k, \overline{\boldsymbol{w}} + \delta \boldsymbol{w}, t_k) - \boldsymbol{f}(\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{u}}_k, \overline{\boldsymbol{w}}, t_k). \tag{3.18}$$

3.5.2 Second-Order Dynamics

Equation 3.18 suggests the need to compute a neighboring trajectory alongside the new control schedule during the backward sweep. This is avoided by using a quadratic approximation

of the dynamics, again obtained by a second-order Taylor series expansion.

$$\delta \boldsymbol{x}_{k+1}^i \approx \boldsymbol{f}_{x,k}^{i,\gamma_1} \delta \boldsymbol{x}_k^{\gamma_1} + \boldsymbol{f}_{u,k}^{i,\gamma_1} \delta \boldsymbol{u}_k^{\gamma_1} + \boldsymbol{f}_w^{i,\gamma_1} \delta \boldsymbol{w}^{\gamma_1} + \frac{1}{2} \boldsymbol{f}_{xx,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{x}_k^{\gamma_1} \delta \boldsymbol{x}_k^{\gamma_2} + \frac{1}{2} \boldsymbol{f}_{uu,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{u}_k^{\gamma_1} \delta \boldsymbol{u}_k^{\gamma_2}$$
$$+ \frac{1}{2} \boldsymbol{f}_{ww,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{w}^{\gamma_1} \delta \boldsymbol{w}^{\gamma_2} + \boldsymbol{f}_{xu,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{x}_k^{\gamma_1} \delta \boldsymbol{u}_k^{\gamma_2} + \boldsymbol{f}_{xw,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{x}_k^{\gamma_1} \delta \boldsymbol{w}^{\gamma_2} + \boldsymbol{f}_{uw,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{u}_k^{\gamma_1} \delta \boldsymbol{w}^{\gamma_2} \qquad (3.19)$$

Equation 3.19 is simplified by considering an augmented state vector $\boldsymbol{X}^T = [\boldsymbol{x}^T \ \boldsymbol{u}^T \ \boldsymbol{w}^T]$ and transition function $\boldsymbol{F}^T = [\boldsymbol{f}^T \ \boldsymbol{0} \ \boldsymbol{0}]$.

$$\delta \boldsymbol{X}_{k+1}^i \approx \boldsymbol{F}_{X,k}^{i,\gamma_1} \delta \boldsymbol{X}_k^{\gamma_1} + \frac{1}{2} \boldsymbol{F}_{XX,k}^{i,\gamma_1\gamma_2} \delta \boldsymbol{X}_k^{\gamma_1} \delta \boldsymbol{X}_k^{\gamma_2} \qquad (3.20)$$

Taylor coefficients for the approximate state deviation are known as the first-order state transition matrix (STM) and second-order state transition tensor (STT).

$$\Phi^{i,a}(t_k, t_{k+1}) = \frac{\partial \boldsymbol{X}_{k+1}^i}{\partial \boldsymbol{X}_k^a} = \boldsymbol{F}_{X,k}^{i,a} \qquad (3.21a)$$

$$\Phi^{i,ab}(t_k, t_{k+1}) = \frac{\partial^2 \boldsymbol{X}_{k+1}^i}{\partial \boldsymbol{X}_k^a \partial \boldsymbol{X}_k^b} = \boldsymbol{F}_{XX,k}^{i,ab} \qquad (3.21b)$$

The STMs (referring to both the STMs and STTs) between each stage obey the following differential equations with initial conditions $\Phi(t_k, t_k)^{i,a} = \delta^{i,a}$, the Kronecker delta or identity matrix, and $\Phi^{i,ab}(t_k, t_k) = 0$,

$$\dot{\Phi}^{i,a} = A^{i,\gamma_1} \Phi^{\gamma_1,a} \qquad (3.22a)$$

$$\dot{\Phi}^{i,ab} = A^{i,\gamma_1} \Phi^{\gamma_1,ab} + A^{i,\gamma_1\gamma_2} \Phi^{\gamma_1,a} \Phi^{\gamma_2,b} \qquad (3.22b)$$

where $A^{i,a}$ and $A^{i,ab}$ are the dynamics matrix and tensor.

$$A^{i,a} = \frac{\partial \dot{\boldsymbol{X}}^i}{\partial \boldsymbol{X}^a} \qquad (3.23a)$$

$$A^{i,ab} = \frac{\partial^2 \dot{\boldsymbol{X}}^i}{\partial \boldsymbol{X}^a \partial \boldsymbol{X}^b} \qquad (3.23b)$$

Equation 3.20 is now restated in terms of STMs, with arguments removed so that the implied mapping is from t_k to t_{k+1}.

$$\delta \boldsymbol{X}_{k+1}^i = \Phi^{i,\gamma_1} \delta \boldsymbol{X}_k^{\gamma_1} + \frac{1}{2} \Phi^{i,\gamma_1\gamma_2} \delta \boldsymbol{X}_k^{\gamma_1} \delta \boldsymbol{X}_k^{\gamma_2} \qquad (3.24)$$

3.5.3   Stage Cost-To-Go Derivatives

Before making use of the STMs to obtain the stage cost-to-go derivatives, the quadratic expansions of the cost-to-go are restated in terms of the augmented state vector.

$$\delta J_k \approx \ ER_{k+1} + J_{X,k}^T \delta \boldsymbol{X}_k + J_{\lambda,k}^T \delta \boldsymbol{\lambda} + \frac{1}{2} \delta \boldsymbol{X}_k^T J_{XX,k} \delta \boldsymbol{X}_k + \frac{1}{2} \delta \boldsymbol{\lambda}^T J_{\lambda\lambda,k} \delta \boldsymbol{\lambda} + \delta \boldsymbol{X}_k^T J_{X\lambda,k} \delta \boldsymbol{\lambda} \quad (3.25a)$$

$$\delta L_k \approx L_{X,k}^T \delta \boldsymbol{X}_k + \frac{1}{2} \delta \boldsymbol{X}_k^T L_{XX,k} \delta \boldsymbol{X}_k \tag{3.25b}$$

$$\delta J_{k+1}^* \approx ER_{k+1} + J_{X,k+1}^{*T} \Phi \delta \boldsymbol{X}_k + \frac{1}{2} J_{X,k+1}^{*\gamma_1} \Phi^{\gamma_1,\gamma_2\gamma_3} \delta \boldsymbol{X}_k^{\gamma_2} \delta \boldsymbol{X}_k^{\gamma_3} + J_{\lambda,k+1}^{*T} \delta \boldsymbol{\lambda}$$
$$+ \frac{1}{2} \delta \boldsymbol{X}_k^T \Phi^T J_{XX,k+1}^* \Phi \delta \boldsymbol{X}_k + \frac{1}{2} \delta \boldsymbol{\lambda}^T J_{\lambda\lambda,k+1}^* \delta \boldsymbol{\lambda} + \delta \boldsymbol{X}_k^T \Phi^T J_{X\lambda,k+1}^* \delta \boldsymbol{\lambda} \tag{3.25c}$$

Recall that the control sensitivities of J_{k+1}^* are zero. Substituting Equation 3.24 into Equation 3.17b yields third and fourth-order terms in $\delta \boldsymbol{X}_k$ that have been ignored in Equation 3.25c. That truncation is inconsequential as Taylor coefficients are only being matched to second-order. Doing so finally yields the stage cost-to-go derivatives with respect to the augmented state and multipliers,

$$J_{X,k} = L_{X,k} + \Phi^T J_{X,k+1}^*, \tag{3.26a}$$

$$J_{\lambda,k} = J_{\lambda,k+1}^*, \tag{3.26b}$$

$$J_{XX,k}^{i,a} = L_{XX,k}^{i,a} + J_{XX,k+1}^{*\gamma_1,\gamma_2} \Phi^{\gamma_1,i} \Phi^{\gamma_2,a} + J_{X,k+1}^{*\gamma_1} \Phi^{\gamma_1,ia}, \tag{3.26c}$$

$$J_{\lambda\lambda,k} = J_{\lambda\lambda,k+1}^*, \tag{3.26d}$$

$$J_{X\lambda,k} = \Phi^T J_{X\lambda,k+1}^*, \tag{3.26e}$$

from which the submatrices are acquired for Equation 3.15.

3.5.4   Stage Update Equations

Before proceeding upstream from stage k to $k - 1$, the stage update equations predict the effects of the updated control. The expected reduction and derivatives of the optimal cost-to-go are obtained by inserting the optimal control law into Equation 3.14.

$$ER_k = ER_{k+1} + J_{u,k}^T A_k + \frac{1}{2} A_k^T J_{uu,k} A_k \tag{3.27a}$$

$$J_{x,k}^{*T} = J_{x,k}^T + J_{u,k}^T B_k + A_k^T J_{uu,k} B_k + A_k^T J_{ux,k} \tag{3.27b}$$

$$J_{w,k}^{*T} = J_{w,k}^T + J_{u,k}^T C_k + A_k^T J_{uu,k} C_k + A_k^T J_{uw,k} \tag{3.27c}$$

$$J_{\lambda,k}^{*T} = J_{\lambda,k}^T + J_{u,k}^T D_k + A_k^T J_{uu,k} D_k + A_k^T J_{u\lambda,k} \tag{3.27d}$$

$$J_{xx,k}^* = J_{xx,k} + B_k^T J_{uu,k} B_k + B_k^T J_{ux,k} + J_{ux,k}^T B_k \tag{3.27e}$$

$$J_{ww,k}^* = J_{ww,k} + C_k^T J_{uu,k} C_k + C_k^T J_{uw,k} + J_{uw,k}^T C_k \tag{3.27f}$$

$$J_{\lambda\lambda,k}^* = J_{\lambda\lambda,k} + D_k^T J_{uu,k} D_k + D_k^T J_{u\lambda,k} + J_{u\lambda,k}^T D_k \tag{3.27g}$$

$$J_{xw,k}^* = J_{xw,k} + B_k^T J_{uu,k} C_k + B_k^T J_{uw,k} + J_{ux,k}^T C_k \tag{3.27h}$$

$$J_{x\lambda,k}^* = J_{x\lambda,k} + B_k^T J_{uu,k} D_k + B_k^T J_{u\lambda,k} + J_{ux,k}^T D_k \tag{3.27i}$$

$$J_{w\lambda,k}^* = J_{w\lambda,k} + C_k^T J_{uu,k} D_k + C_k^T J_{u\lambda,k} + J_{uw,k}^T D_k \tag{3.27j}$$

The stage update equations require initial values to solve the first subproblem at stage $k = N - 1$. For multi-phase trajectories, those initial conditions are obtained from the preceding phase minimization. For single-phase trajectories, or the final phase, these values are straightforward to obtain as there is no subproblem at the final state. The optimal and nominal costs are equivalent, $J_N^* = \widetilde{\varphi}$. Derivatives of the optimal cost-to-go are derivatives of the augmented Lagrangian, and there is no expected reduction, $ER_N = 0$.

3.5.5   Inter-phase Subproblems

After the final stage subproblem is solved at stage $k = 0$, the sensitivity of the cost-to-go is known with respect to the multipliers and parameters for that phase. The inter-phase subproblem of phase i computes the optimal updates $\delta \boldsymbol{w}_i^*$ and $\delta \boldsymbol{\lambda}_i^*$ before proceeding with the stage subproblems of phase $i-1$. The inter-phase cost-to-go $J_{i,0}$, however, considers states, parameters, and multipliers for both phase i and phase $i - 1$, so the quadratic model in Equation 3.14 is incomplete. Adopting HDDP phase notation renames the variables $\boldsymbol{x}_+ = \boldsymbol{x}_{i,0}$, $\boldsymbol{w}_+ = \boldsymbol{w}_i$, $\boldsymbol{\lambda}_+ = \boldsymbol{\lambda}_i$, $\boldsymbol{x}_- = \boldsymbol{x}_{i-1,N_i}$,

$\boldsymbol{w}_- = \boldsymbol{w}_{i-1}$, $\boldsymbol{\lambda}_- = \boldsymbol{\lambda}_{i-1}$. The quadratic expansion of $J_{i,0}(\boldsymbol{x}_+, \boldsymbol{w}_+, \boldsymbol{\lambda}_+, \boldsymbol{x}_-, \boldsymbol{w}_-, \boldsymbol{\lambda}_-)$ becomes

$$
\begin{aligned}
\delta J_{i,0} \approx\ & ER_{i,0} + J_{x_+}^T \delta \boldsymbol{x}_+ + J_{w_+}^T \delta \boldsymbol{w}_+ + J_{\lambda_+}^T \delta \boldsymbol{\lambda}_+ + J_{x_-}^T \delta \boldsymbol{x}_- + J_{w_-}^T \delta \boldsymbol{w}_- + J_{\lambda_-}^T \delta \boldsymbol{\lambda}_- \\
& + \frac{1}{2} \delta \boldsymbol{x}_+^T J_{x_+ x_+} \delta \boldsymbol{x}_+ + \frac{1}{2} \delta \boldsymbol{w}_+^T J_{w_+ w_+} \delta \boldsymbol{w}_+ + \frac{1}{2} \delta \boldsymbol{\lambda}_+^T J_{\lambda_+ \lambda_+} \delta \boldsymbol{\lambda}_+ + \frac{1}{2} \delta \boldsymbol{x}_-^T J_{x_- x_-} \delta \boldsymbol{x}_- \\
& + \frac{1}{2} \delta \boldsymbol{w}_-^T J_{w_- w_-} \delta \boldsymbol{w}_- + \frac{1}{2} \delta \boldsymbol{\lambda}_-^T J_{\lambda_- \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{x}_+^T J_{x_+ w_+} \delta \boldsymbol{w}_+ + \delta \boldsymbol{x}_+^T J_{x_+ \lambda_+} \delta \boldsymbol{\lambda}_+ \\
& + \delta \boldsymbol{x}_+^T J_{x_+ x_-} \delta \boldsymbol{x}_- + \delta \boldsymbol{x}_+^T J_{x_+ w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{x}_+^T J_{x_+ \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{w}_+^T J_{w_+ \lambda_+} \delta \boldsymbol{\lambda}_+ \\
& + \delta \boldsymbol{w}_+^T J_{w_+ x_-} \delta \boldsymbol{x}_- + \delta \boldsymbol{w}_+^T J_{w_+ w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{w}_+^T J_{w_+ \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{x}_-^T J_{x_- w_-} \delta \boldsymbol{w}_- \\
& + \delta \boldsymbol{x}_-^T J_{x_- \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{w}_-^T J_{w_- \lambda_-} \delta \boldsymbol{\lambda}_- \ .
\end{aligned}
\tag{3.28}
$$

Coefficients in Equation 3.28 are found from the optimal cost-to-go of phase i and the augmented Lagrangian of phase $i - 1$. There are no cross partial derivatives of $\boldsymbol{\lambda}_+$ with $\boldsymbol{x}_-$, $\boldsymbol{w}_-$, or $\boldsymbol{\lambda}_-$ as $\boldsymbol{\lambda}_+$ affects only $\widetilde{\varphi}_i$ while $\boldsymbol{x}_-$, $\boldsymbol{w}_-$, and $\boldsymbol{\lambda}_-$ affect only $\widetilde{\varphi}_{i-1}$.

$$
\begin{aligned}
& J_{x_+} = J_{x,0}^* + \widetilde{\varphi}_{x_+}, \quad J_{w_+} = J_{w,0}^* + \widetilde{\varphi}_{w_+}, \quad J_{\lambda_+} = J_{\lambda,0}^*, \quad J_{x_-} = \widetilde{\varphi}_{x_-}, \quad J_{w_-} = \widetilde{\varphi}_{w_-}, \quad J_{\lambda_-} = \widetilde{\varphi}_{\lambda_-}, \\[4pt]
& J_{x_+ x_+} = J_{xx,0}^* + \widetilde{\varphi}_{x_+ x_+}, \quad J_{w_+ w_+} = J_{ww,0}^* + \widetilde{\varphi}_{w_+ w_+}, \quad J_{\lambda_+ \lambda_+} = J_{\lambda\lambda,0}^*, \\[4pt]
& J_{x_- x_-} = \widetilde{\varphi}_{x_- x_-}, \quad J_{w_- w_-} = \widetilde{\varphi}_{w_- w_-}, \quad J_{\lambda_- \lambda_-} = 0, \quad J_{x_+ w_+} = J_{xw,0}^* + \widetilde{\varphi}_{x_+ w_+}, \quad J_{x_+ \lambda_+} = J_{x\lambda,0}^* \\[4pt]
& J_{x_+ x_-} = \widetilde{\varphi}_{x_+ x_-}, \quad J_{x_+ w_-} = \widetilde{\varphi}_{x_+ w_-}, \quad J_{x_+ \lambda_-} = \widetilde{\varphi}_{x_+ \lambda_-}, \quad J_{w_+ \lambda_+} = J_{w\lambda,0}^*, \quad J_{w_+ x_-} = \widetilde{\varphi}_{w_+ x_-}, \\[4pt]
& J_{w_+ w_-} = \widetilde{\varphi}_{w_+ w_-}, \quad J_{w_+ \lambda_-} = \widetilde{\varphi}_{w_+ \lambda_-}, \quad J_{x_- w_-} = \widetilde{\varphi}_{x_- w_-}, \quad J_{x_- \lambda_-} = \widetilde{\varphi}_{x_- \lambda_-}, \quad J_{w_- \lambda_-} = \widetilde{\varphi}_{w_- \lambda_-}.
\end{aligned}
\tag{3.29}
$$

At this point in the stage subproblem derivation, a quadratic approximation of the transition function eliminated the dependency on $\delta \boldsymbol{x}_{k+1}$. Here the dependency on $\delta \boldsymbol{x}_+$ is similarly eliminated by a quadratic approximation of the initial function.

$$
\delta \boldsymbol{x}_+^i = \boldsymbol{\Gamma}_w^{i,\gamma_1} \delta \boldsymbol{w}_+^{\gamma_1} + \frac{1}{2} \boldsymbol{\Gamma}_{ww}^{i,\gamma_1 \gamma_2} \delta \boldsymbol{w}_+^{\gamma_1} \delta \boldsymbol{w}_+^{\gamma_2}
\tag{3.30}
$$

Since $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{w}_+)$ the derivatives $\boldsymbol{\Gamma}_w$ and $\boldsymbol{\Gamma}_{ww}$ are with respect to $\boldsymbol{w}_+$ and double subscripts are avoided. The second-order model in Equation 3.28 now becomes,

$$
\begin{aligned}
\delta J_{i,0} \approx\ & ER_{i,0} + \widetilde{J}_{w_+}^T \delta \boldsymbol{w}_+ + J_{\lambda_+}^T \delta \boldsymbol{\lambda}_+ + J_{x_-}^T \delta \boldsymbol{x}_- + J_{w_-}^T \delta \boldsymbol{w}_- + J_{\lambda_-}^T \delta \boldsymbol{\lambda}_- \\
& + \frac{1}{2}\delta \boldsymbol{w}_+^T \widetilde{J}_{w_+ w_+} \delta \boldsymbol{w}_+ + \frac{1}{2}\delta \boldsymbol{\lambda}_+^T J_{\lambda_+ \lambda_+} \delta \boldsymbol{\lambda}_+ + \frac{1}{2}\delta \boldsymbol{x}_-^T J_{x_- x_-} \delta \boldsymbol{x}_- + \frac{1}{2}\delta \boldsymbol{w}_-^T J_{w_- w_-} \delta \boldsymbol{w}_- \\
& + \delta \boldsymbol{w}_+^T \widetilde{J}_{w_+ \lambda_+} \delta \boldsymbol{\lambda}_+ + \delta \boldsymbol{w}_+^T \widetilde{J}_{w_+ x_-} \delta \boldsymbol{x}_- + \delta \boldsymbol{w}_+^T \widetilde{J}_{w_+ w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{w}_+^T \widetilde{J}_{w_+ \lambda_-} \delta \boldsymbol{\lambda}_- \\
& + \delta \boldsymbol{x}_-^T J_{x_- w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{x}_-^T J_{x_- \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{w}_-^T J_{w_- \lambda_-} \delta \boldsymbol{\lambda}_-,
\end{aligned}
\tag{3.31}
$$

where the coefficients have been redefined to combine terms of like order in $\delta \boldsymbol{w}_+$.

$$
\widetilde{J}_{w_+} = J_{w_+} + \boldsymbol{\Gamma}_w^T J_{x_+}
\tag{3.32a}
$$

$$
\widetilde{J}_{w_+ w_+}^{i,a} = J_{w_+ w_+}^{i,a} + \boldsymbol{\Gamma}_{ww}^{i,\gamma_1 a} J_{x_+}^{\gamma_1} + J_{x_+ x_+}^{\gamma_1,\gamma_2} \boldsymbol{\Gamma}_w^{\gamma_1,a} \boldsymbol{\Gamma}_w^{\gamma_1,b} + J_{x_+ w_+}^{i,\gamma_1} \boldsymbol{\Gamma}_w^{\gamma_1 a} + \boldsymbol{\Gamma}_w^{i,\gamma_1} J_{x_+ w_+}^{\gamma_1 a}
\tag{3.32b}
$$

$$
\widetilde{J}_{w_+ \lambda_+} = J_{w_+ \lambda_+} + \boldsymbol{\Gamma}_w^T J_{x_+ w_+}
\tag{3.32c}
$$

$$
\widetilde{J}_{w_+ x_-} = J_{w_+ x_-} + \boldsymbol{\Gamma}_w^T J_{x_+ x_-}
\tag{3.32d}
$$

$$
\widetilde{J}_{w_+ w_-} = J_{w_+ w_-} + \boldsymbol{\Gamma}_w^T J_{x_+ w_-}
\tag{3.32e}
$$

$$
\widetilde{J}_{w_+ \lambda_-} = J_{w_+ \lambda_-} + \boldsymbol{\Gamma}_w^T J_{x_+ \lambda_-}
\tag{3.32f}
$$

3.5.6   Multiplier Update

HDDP decouples the multiplier and parameter updates and first obtains the optimal multiplier update in the same manner as the optimal control update at each stage, by setting the derivative of the quadratic model with respect to the multiplier update equal to zero.

$$
\delta \boldsymbol{\lambda}_+ = A_{\lambda_+} + C_{\lambda_+} \delta \boldsymbol{w}_+
$$

$$
A_{\lambda_+} = -J_{\lambda_+ \lambda_+}^{-1} J_{\lambda_+}
\tag{3.33}
$$

$$
C_{\lambda_+} = -J_{\lambda_+ \lambda_+}^{-1} J_{\lambda_+ w_+}
$$

The trust-region method is again employed to perform the multiplier update in Equation 3.33. Now however, $J_{\lambda_+ \lambda_+}^*$ is negative-definite, so the update $\delta \boldsymbol{\lambda}_+$ is a maximizer and the expected reduction

increases. In preparation for the parameter update, the quadratic model is updated by substituting the multiplier update into Equation 3.31.

$$\delta J_{i,0} \approx ER_{i,0} + \widehat{J}_{w_+}^T \delta \boldsymbol{w}_+ + J_{x_-}^T \delta \boldsymbol{x}_- + J_{w_-}^T \delta \boldsymbol{w}_- + J_{\lambda_-}^T \delta \boldsymbol{\lambda}_- + \frac{1}{2} \delta \boldsymbol{w}_+^T \widehat{J}_{w_+ w_+} \delta \boldsymbol{w}_+$$
$$+ \frac{1}{2} \delta \boldsymbol{x}_-^T J_{x_- x_-} \delta \boldsymbol{x}_- + \frac{1}{2} \delta \boldsymbol{w}_-^T J_{w_- w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{w}_+^T \widehat{J}_{w_+ x_-} \delta \boldsymbol{x}_- + \delta \boldsymbol{w}_+^T \widehat{J}_{w_+ w_-} \delta \boldsymbol{w}_- \qquad (3.34)$$
$$+ \delta \boldsymbol{w}_+^T \widehat{J}_{w_+ \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{x}_-^T J_{x_- w_-} \delta \boldsymbol{w}_- + \delta \boldsymbol{x}_-^T J_{x_- \lambda_-} \delta \boldsymbol{\lambda}_- + \delta \boldsymbol{w}_-^T J_{w_- \lambda_-} \delta \boldsymbol{\lambda}_-$$

The expected reduction and coefficients for Equation 3.34 are updated to reflect the multiplier increment.

$$ER_{i,0} = ER_{i,0} + J_{\lambda_+}^T A_{\lambda_+} + \frac{1}{2} A_{\lambda_+}^T J_{\lambda_+ \lambda_+} A_{\lambda_+} \qquad (3.35a)$$

$$\widehat{J}_{w_+}^T = \widetilde{J}_{w_+}^T + J_{\lambda_+}^T C_{\lambda_+} + A_{\lambda_+}^T J_{\lambda_+ \lambda_+} C_{\lambda_+} + A_{\lambda_+}^T J_{\lambda_+ w_+} \qquad (3.35b)$$

$$\widehat{J}_{w_+ w_+} = \widetilde{J}_{w_+ w_+} + C_{\lambda_+}^T J_{w_+ w_+} C_{\lambda_+} + C_{\lambda_+}^T J_{\lambda_+ w_+} + J_{w_+ \lambda_+} C_{\lambda_+} \qquad (3.35c)$$

$$\widehat{J}_{w_+ x_-} = \widetilde{J}_{w_+ x_-} \qquad (3.35d)$$

$$\widehat{J}_{w_+ w_-} = \widetilde{J}_{w_+ w_-} \qquad (3.35e)$$

$$\widehat{J}_{w_+ \lambda_-} = \widetilde{J}_{w_+ \lambda_-} \qquad (3.35f)$$

3.5.7 Parameter Update

The parameter update is now available by once again minimizing the quadratic model of the cost-to-go that is represented by Equation 3.34.

$$\delta \boldsymbol{w}_+ = A_{w_+} + B_{w_+} \delta \boldsymbol{x}_- + C_{w_+} \delta \boldsymbol{w}_- + D_{w_+} \delta \boldsymbol{\lambda}_-$$
$$A_{w_+} = -\widehat{J}_{w_+ w_+}^{-1} J_{w_+}$$
$$B_{w_+} = -\widehat{J}_{w_+ w_+}^{-1} J_{w_+ x_-} \qquad (3.36)$$
$$C_{w_+} = -\widehat{J}_{w_+ w_+}^{-1} J_{w_+ w_-}$$
$$D_{w_+} = -\widehat{J}_{w_+ w_+}^{-1} J_{w_+ \lambda_-}$$

The trust-region method is again utilized for the parameter update. The expected reduction and optimal cost-to-go are updated to include the effects of the parameter update.

$$ER_{i,0} = ER_{i,0} + \widehat{J}_{w_+}^T A_{w_+} + \frac{1}{2} A_{w_+}^T \widehat{J}_{w_+ w_+} A_{w_+} \tag{3.37a}$$

$$J_{x_-}^{*T} = J_{x_-}^T + \widehat{J}_{w_+}^T B_{w_+} + A_{w_+}^T \widehat{J}_{w_+ w_+} B_{w_+} + A_{w_+}^T \widehat{J}_{w_+ x_-} \tag{3.37b}$$

$$J_{w_-}^{*T} = J_{w_-}^T + \widehat{J}_{w_+}^T C_{w_+} + A_{w_+}^T \widehat{J}_{w_+ w_+} C_{w_+} + A_{w_+}^T \widehat{J}_{w_+ w_-} \tag{3.37c}$$

$$J_{\lambda_-}^{*T} = J_{\lambda_-}^T + \widehat{J}_{w_+}^T D_{w_+} + A_{w_+}^T \widehat{J}_{w_+ w_+} D_{w_+} + A_{w_+}^T \widehat{J}_{w_+ \lambda_-} \tag{3.37d}$$

$$J_{x_- x_-}^* = J_{x_- x_-} + B_{w_+}^T \widehat{J}_{w_+ w_+} B_{w_+} + B_{w_+}^T \widehat{J}_{w_+ x_-} + \widehat{J}_{w_+ x_-}^T B_{w_+} \tag{3.37e}$$

$$J_{w_- w_-}^* = J_{w_- w_-} + C_{w_+}^T \widehat{J}_{w_+ w_+} C_{w_+} + C_{w_+}^T \widehat{J}_{w_+ w_-} + \widehat{J}_{w_+ w_-}^T C_{w_+} \tag{3.37f}$$

$$J_{\lambda_- \lambda_-}^* = J_{\lambda_- \lambda_-} + D_{w_+}^T \widehat{J}_{w_+ w_+} D_{w_+} + D_{w_+}^T \widehat{J}_{w_+ \lambda_-} + \widehat{J}_{w_+ \lambda_-}^T D_{w_+} \tag{3.37g}$$

$$J_{x_- w_-}^* = J_{x_- w_-} + B_{w_+}^T \widehat{J}_{w_+ w_+} C_{w_+} + B_{w_+}^T \widehat{J}_{w_+ w_-} + \widehat{J}_{w_+ x_-}^T C_{w_+} \tag{3.37h}$$

$$J_{x_- \lambda_-}^* = J_{x_- \lambda_-} + B_{w_+}^T \widehat{J}_{w_+ w_+} D_{w_+} + B_{w_+}^T \widehat{J}_{w_+ \lambda_-} + \widehat{J}_{w_+ x_-}^T D_{w_+} \tag{3.37i}$$

$$J_{w_- \lambda_-}^* = J_{w_- \lambda_-} + C_{w_+}^T \widehat{J}_{w_+ w_+} D_{w_+} + C_{w_+}^T \widehat{J}_{w_+ \lambda_-} + \widehat{J}_{w_+ w_-}^T D_{w_+} \tag{3.37j}$$

The phase update equations in Equation 3.37 complete the inter-phase subproblem. The backward sweep proceeds with the stage subproblems of phase $i - 1$ and continues through every stage of each phase.

Sections 3.5.5 to 3.5.7 are equally valid for single phase trajectories. In that case the contributions from $\boldsymbol{x}_-$, $\boldsymbol{w}_-$, and $\boldsymbol{\lambda}_-$ can be ignored. The phase update equations are then unnecessary, but the expected reduction should still be incremented. Similarly, these results apply for trajectories absent of static parameters $\boldsymbol{w}_+$ or $\boldsymbol{w}_-$. By removing those contributions, the inter-phase subproblem terminates with the multiplier update.

3.6 Trust-Region Quadratic Subproblem

The unconstrained control update in Equation 3.15 is likely to step beyond the valid region of the quadratic approximation, as are the multiplier update in Equation 3.33 and parameter update in Equation 3.36. An unconstrained backward sweep and subsequent application of the new control

law is likely to lead to divergence or infeasible iterates. The updates also require the Hessians to be invertible, with $J_{uu,k}$ and $\widehat{J}_{w_+w_+}$ positive definite, and $J_{\lambda_+\lambda_+}$ negative definite. HDDP overcomes these challenges by solving a trust-region quadratic subproblem (TRQP) at each stage and inter-phase subproblem. Now to compute each stage control update, for example, $\delta\boldsymbol{u}_k$ is required to lie within the trust-region radius Δ.

$$\min_{\delta\boldsymbol{u}_k} \left[J_{u,k}^T \delta\boldsymbol{u}_k + \frac{1}{2}\delta\boldsymbol{u}_k^T J_{uu,k} \delta\boldsymbol{u}_k \right]$$

$$\text{s.t. } \|D\delta\boldsymbol{u}_k\| \leq \Delta \tag{3.38}$$

The subproblem posed in Equation 3.38 is named TRQP (J_u, J_{uu}, Δ). The inter-phase subproblems are TRQP $(J_{w_+}, J_{w_+w_+}, \Delta)$ and TRQP $(-J_{\lambda_+}, -J_{\lambda_+\lambda_+}, \Delta)$. Different variables, stages, and phases may be assigned their own Δ. The methods of Reference [67] have proved robust in solving this subproblem in DDP. However, DDP performance is sensitive to the selection of a scaling matrix D that determines the shape of the trust-region. Often, setting D to the identity matrix is sufficient. When components of the gradient and Hessian vary by orders of magnitude, a robust heuristic for selecting D becomes necessary. Reference [47] suggests several scaling methods and provides a performance comparison.

3.7   Iteration

Quadratic models of the cost function and dynamics are inexact for higher-order and nonlinear systems. Iterations of the forward pass and backward sweep are necessary to proceed from an infeasible or suboptimal reference trajectory to the local minimum. The reduction ratio

$$\rho = \frac{\delta J}{ER_{0,0}} \tag{3.39}$$

serves as an acceptance criterion for new iterates. If $\rho \approx 1$, then the quadratic model is good and the reduction in cost is as predicted. The iterate is accepted and a larger trust-region is allowed. Otherwise the model is poor, the iterate is rejected and the trust-region is reduced for the next

backward sweep.

$$\Delta_{p+1} = \begin{cases} \min\left[(1+\kappa)\,\Delta_p, \Delta_{\max}\right], & \text{if } \rho \in [1-\epsilon_1, 1+\epsilon_1] \\ \\ \max\left[(1-\kappa)\,\Delta_p, \Delta_{\min}\right], & \text{otherwise} \end{cases} \tag{3.40}$$

For the trust-region update in Equation 3.40, p is the iteration counter, κ and ϵ_1 are an additional set of tuning parameters, as is the allowable trust-region range $[\Delta_{\min}, \Delta_{\max}]$.

If $ER_{0,0}$ is zero after the backward sweep, or less than some optimality tolerance, the optimization has reached a stationary point of the cost function with respect to controls, parameters and multipliers. This point is a minimum if all of the Hessians $J_{uu,k}$ and $\widehat{J}_{w_+ w_+}$ are positive definite and $J_{\lambda_+ \lambda_+}$ are negative definite. The algorithm converges upon reaching a minimum while satisfying terminal constraints to within a feasibility tolerance.

3.8 DDP and Indirect Optimization

In summarizing the connection drawn between HDDP and Pontryagin's Principle in Reference [46], a single phase trajectory of N stages is considered. The discrete Hamiltonian at each stage is defined as the sum of the local cost L and the scalar product of time-varying co-states $\boldsymbol{p}$ with the transition function.

$$H_k = L_k + \boldsymbol{p}_k^T \boldsymbol{f}_k \tag{3.41}$$

Disregarding the ALM cost function for the moment, terminal constraints on the state variables are enforced by $\boldsymbol{\psi}(\boldsymbol{x}_N) = \boldsymbol{0}$ and with constant multipliers $\boldsymbol{\lambda}$ and final cost φ define the terminal function.

$$\Psi(\boldsymbol{x}_N) = \varphi(\boldsymbol{x}_N) + \boldsymbol{\lambda}^T \boldsymbol{\psi}(\boldsymbol{x}_N) \tag{3.42}$$

Necessary conditions for an optimal control trajectory arise from the Euler-Lagrange theorem.

$$\frac{\partial H_k^*}{\partial \boldsymbol{u}_k} = \boldsymbol{0} \tag{3.43a}$$

$$\dot{\boldsymbol{x}}_k^T = \frac{\partial H_k^*}{\partial \boldsymbol{p}_k} \tag{3.43b}$$

$$\dot{\boldsymbol{p}}_k^T = -\frac{\partial H_k^*}{\partial \boldsymbol{x}_k} \tag{3.43c}$$

$$\boldsymbol{p}_N^T = \frac{\partial \Psi^*}{\partial \boldsymbol{x}_N} \tag{3.43d}$$

$$\boldsymbol{p}_0^T = -\frac{\partial \Psi^*}{\partial \boldsymbol{x}_0} \tag{3.43e}$$

As with previous notations of the optimal cost-to-go, the superscript asterisk denotes an optimal value. An optimal trajectory terminates with $\boldsymbol{\psi} = \boldsymbol{0}$ and the ALM cost and terminal function are equivalent, $\tilde{\varphi}^* = \Psi^*$. Next consider the sensitivity of those cost functions to the terminal state.

$$\frac{\partial \tilde{\varphi}^*}{\partial \boldsymbol{x}_N} = \frac{\partial \varphi}{\partial \boldsymbol{x}_N} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{\psi}}{\boldsymbol{x}_N} = \frac{\partial \Psi^*}{\partial \boldsymbol{x}_N} \tag{3.44}$$

This result suggests that the final values of the co-states can be found by minimizing the ALM cost, even by direct methods such as DDP.

$$\frac{\partial \tilde{\varphi}^*}{\partial \boldsymbol{x}_N} = \boldsymbol{p}_N^T \tag{3.45}$$

The same argument applies to the initial values.

$$\frac{\partial \tilde{\varphi}^*}{\partial \boldsymbol{x}_0} = -\boldsymbol{p}_0^T \tag{3.46}$$

3.9   Example Earth-Mars Rendezvous

Implementation of the DDP algorithm was validated by reproducing an Earth-Mars rendezvous transfer from Reference [47]. However, here the control is continuous low-thrust in the radial, transverse and normal directions, as opposed to approximating the low-thrust by impulsive maneuvers separated by Keplerian arcs in HDDP. The two-body dynamics with continuous low-thrust are consistent with the finite-burn low-thrust model (FBLT) in the Evolutionary Mission Trajectory Generator (EMTG), which was used for additional validation [68].

A 1000 kg spacecraft with a 0.5 N, 2000 second I_{sp} thruster was set to depart Earth on April 10, 2007. A fixed time of flight of 348.79 days sets the arrival date to March 23, 2008. The augmented state vector consists of the three position components, three velocity components, spacecraft mass, and three spherical thrust control variables, $\boldsymbol{X}^T = [\boldsymbol{r}^T \ \boldsymbol{v}^T \ m \ \boldsymbol{u}^T]$. The single-phase trajectory is discretized into 40 stages of equal time and numerical integration was performed with fixed-step eighth-order Runge-Kutta (RK8) formulae from Dormand and Prince [69]. The augmented Lagrangian is formed to maximize the final mass with a penalty on the position and velocity errors at Mars arrival.

$$
\phi = -m_f
$$
$$
\psi = \begin{bmatrix} \boldsymbol{r}_f - \boldsymbol{r}_M \\ \boldsymbol{v}_f - \boldsymbol{v}_M \end{bmatrix}
\tag{3.47}
$$

Subscript f indicates a final value and M indicates the state of Mars. The resulting transfer is shown in Figure 3.1a with corresponding thrust profile in Figure 3.1b. The trajectory was also computed in the EMTG – FBLT model for validation. DDP obtains the expected bang-bang control profile, but differs slightly from EMTG at the switching points. Table 3.1 shows how final mass increases with the improved accuracy of a continuous low-thrust model, which is particular to this example and not a generalization. Lagrange multipliers from the continuous low-thrust solution and the reference HDDP solution in Table 3.2 differ but are seemingly related.

Table 3.1: Comparison of spacecraft final mass between Earth-Mars rendezvous solution methods.

HDDP	$m_f = 598.66$ kg
DDP – FBLT	$m_f = 603.29$ kg
EMTG – FBLT	$m_f = 603.45$ kg

Table 3.2: Comparison of Lagrange multipliers between Earth-Mars rendezvous solution methods.

HDDP	$\boldsymbol{\lambda} = \begin{bmatrix} 0.5095 & -1.2700 & -0.2665 & 0.1178 & 2.0701 & 0.1340 \end{bmatrix}^T$
DDP – FBLT	$\boldsymbol{\lambda} = \begin{bmatrix} 0.4326 & -1.0663 & -0.2380 & 0.0727 & 1.7121 & 0.1064 \end{bmatrix}^T$
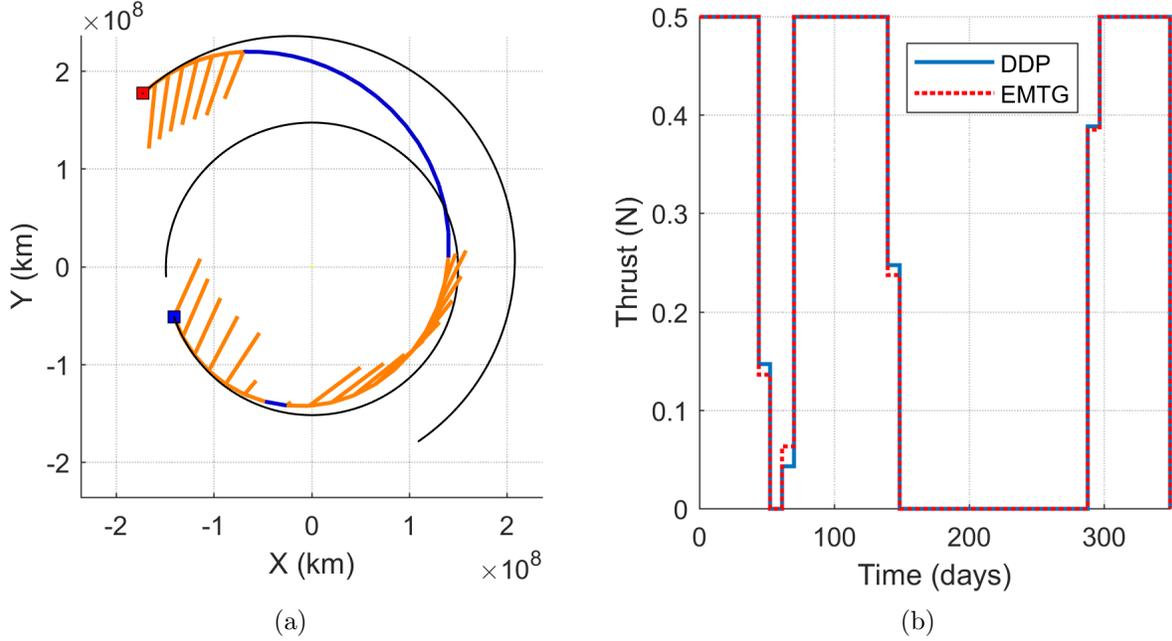
Figure 3.1: (a) Ecliptic view of an example Earth-Mars rendezvous transfer with thrust vectors shown. (b) Thrust profiles from the FBLT implementation of DDP and the EMTG–FBLT model.

3.9.1   Variable Departure and Arrival Times

Trajectory performance is expected to vary with the times of departure and arrival, especially when planetary encounters are involved. Mission event dates may be determined by a parametric study where an optimal trajectory is computed for each combination in a range of departure dates and arrival dates. Other design variables can be included. The trade study allows informed mission design when the ranges of parameter values are sampled with sufficient resolution. A nominal trajectory can be selected from the grid of solutions, but the specified parameter values are unlikely to match the optimal values. This limitation can be resolved in DDP by including the parameters as continuous variables to be solved for in the parameter vector $\boldsymbol{w}$. For variable departure and arrival times, $\boldsymbol{w} = [t_0 \ t_f]^T$.

To begin the backward sweep and for inter-phase subproblems, the derivatives $\widetilde{\varphi}_{w_+}$, $\widetilde{\varphi}_{w_+ w_+}$, $\widetilde{\varphi}_{x_+ w_+}$ and $\widetilde{\varphi}_{w_+ \lambda_+}$ in Equation 3.29 are taken with respect to the parameter vector of time variables. Furthermore, the sensitivity of the states at all stages to the initial and final times must be captured

in the STMs. Those derivatives depend on how the increments δt_0 and δt_f are applied. In this implementation, evenly distributing the time increments among all stages affects the discretization mesh in an accordion-like fashion.

$$\frac{\partial t_k}{\partial t_0} = 1 - \frac{k}{N} \tag{3.48a}$$

$$\frac{\partial t_k}{\partial t_f} = \frac{k}{N} \tag{3.48b}$$

In Equation 3.48, k is the stage index beginning from zero and N is the number of stages. The final stage is held fixed when the initial time is changed, and vice versa. STM entries may be obtained by chaining the local time derivatives with Equation 3.48. STM subscripts are introduced to note the derivative of the downstream state with respect to the subscript, e.g. $\Phi_{w,k} = \partial \boldsymbol{X}_{k+1}/\partial w$.

$$\Phi_{t_0,k} = \dot{\boldsymbol{x}}_{k+1} \frac{\partial t_{k+1}}{\partial t_0} \tag{3.49a}$$

$$\Phi_{t_0 t_0,k} = \ddot{\boldsymbol{x}}_{k+1} \left(\frac{\partial t_{k+1}}{\partial t_0} \right)^2 \tag{3.49b}$$

$$\Phi^{i,a}_{x t_0,k} = \Phi^{i,a\gamma_1}_{xx,k} \dot{\boldsymbol{x}}^{\gamma_1}_k \frac{\partial t_k}{\partial t_0} + \dot{\Phi}^{i,a}_{x,k} \left(\frac{\partial t_{k+1}}{\partial t_0} - \frac{\partial t_k}{\partial t_0} \right) \tag{3.49c}$$

$$\Phi_{t_f,k} = \dot{\boldsymbol{x}}_{k+1} \frac{\partial t_{k+1}}{\partial t_f} \tag{3.49d}$$

$$\Phi_{t_f t_f,k} = \ddot{\boldsymbol{x}}_{k+1} \left(\frac{\partial t_{k+1}}{\partial t_f} \right)^2 \tag{3.49e}$$

$$\Phi^{i,a}_{x t_f,k} = \Phi^{i,a\gamma_1}_{xx,k} \dot{\boldsymbol{x}}^{\gamma_1}_k \frac{\partial t_k}{\partial t_f} + \dot{\Phi}^{i,a}_{x,k} \left(\frac{\partial t_{k+1}}{\partial t_f} - \frac{\partial t_k}{\partial t_f} \right) \tag{3.49f}$$

$$\Phi_{t_0 t_f,k} = \ddot{\boldsymbol{x}}_{k+1} \left(\frac{\partial t_{k+1}}{\partial t_0} \right) \left(\frac{\partial t_{k+1}}{\partial t_f} \right) \tag{3.49g}$$

Derivatives of the augmented Lagrangian with respect to the final time are found by chaining the sensitivity of $\widetilde{\varphi}$ with respect to $\boldsymbol{x} = \boldsymbol{x}_N = \boldsymbol{x}_f$ and the time derivative $\dot{\boldsymbol{x}}$. The initial time

derivative vanishes following $\partial t_N / \partial t_0 = 0$ in Equation 3.48.

$$\widetilde{\varphi}_{t_f} = \widetilde{\varphi}_{x_+} \dot{\boldsymbol{x}} \tag{3.50a}$$

$$\widetilde{\varphi}_{t_f t_f} = \dot{\boldsymbol{x}}^T \widetilde{\varphi}_{x_+ x_+} \dot{\boldsymbol{x}} + \widetilde{\varphi}_{x_+} \ddot{\boldsymbol{x}} \tag{3.50b}$$

$$\widetilde{\varphi}_{x_+ t_f} = \widetilde{\varphi}_{x_+ x_+} \dot{\boldsymbol{x}} \tag{3.50c}$$

$$\widetilde{\varphi}_{t_f \lambda_+} = \left(\widetilde{\varphi}_{\lambda_+ x_+} \dot{\boldsymbol{x}} \right)^T \tag{3.50d}$$

The addition of variable departure and arrival times failed to improve the delivered mass from the fixed time of flight Earth-Mars rendezvous example. Optimal departure and arrival times were preselected. DDP iterates only exhibit local improvements to the nominal trajectory within a region determined by the validity of the quadratic approximation of the *cost-to-go*. After converging on a solution it is difficult to claim anything more than local optimality, as is the case for any nonlinear optimization algorithm. Pairing trajectory optimization with a global search method allows for the selection of the best design from many locally optimal solutions and a reduced computational burden from a parametric study.

3.9.2 Monotonic Basin Hopping

Monotonic basin hopping (MBH) is a global optimization heuristic for exploring the entire solution space by exploiting local optima to find improvements in nearby solutions. MBH has been successfully coupled with NLP solvers by the European Space Agency's Advanced Concepts Team [70] and in EMTG [71]. Pseudocode in Algorithm 1 details the procedure, where $\boldsymbol{x}$ is a vector of decision variables. The two-loop process requires a robust trajectory optimizer as the inner loop to converge from initial guesses that are generated in an outer loop. The robustness of DDP to poor initial guesses makes DDP a suitable inner-loop for MBH. That is demonstrated here for the Earth-Mars rendezvous with variable departure and arrival times.

First, an initial guess is provided for DDP to generate a nominal solution. Instead of a random point $\boldsymbol{x}$, a ballistic initial guess results in the first $\boldsymbol{x}_{current}$ which is the fixed time of flight

---

Algorithm 1 Monotonic Basin Hopping (MBH)

---

generate random point $\mathbf{x}$
run NLP solver to find point $\mathbf{x}^*$ using initial guess $\mathbf{x}$
$\mathbf{x}_{current} = \mathbf{x}^*$
if $\mathbf{x}^*$ is a feasible point **then**
    save $\mathbf{x}^*$ to archive
end if
while not hit stop criterion **do**
    generate $\mathbf{x}'$ by randomly perturbing $\mathbf{x}_{current}$
    for each time of flight variable t_i in $\mathbf{x}'$ **do**
        if $rand\,(0,1) < \rho_{time-hop}$ **then**
            shift t_i forward or backward one time period
        end if
    end for
    run NLP solver to find locally optimal point $\mathbf{x}^*$ from $\mathbf{x}'$
    if $\mathbf{x}^*$ is feasible **and** $J\,(\mathbf{x}^*) < J\,(\mathbf{x}_{current})$ **then**
        $\mathbf{x}_{current} = \mathbf{x}^*$
        save $\mathbf{x}^*$ to archive
    else if $\mathbf{x}^*$ is infeasible **and** $\|\boldsymbol{\psi}\,(\mathbf{x}^*)\| < \|\boldsymbol{\psi}\,(\mathbf{x}_{current})\|$
        $\mathbf{x}_{current} = \mathbf{x}^*$
    end if
end while
return best $\mathbf{x}^*$ in archive

---

solution in Figure 3.1. MBH then proceeds until meeting a prescribed stopping criteria such as computation time or an iteration limit. For this example, MBH was terminated after one hour.

The MBH iteration, or hop, begins by introducing random perturbations to the decision variables of the nominal solution. Then, the inner-loop is reinitialized. Small perturbations are expected to fall within the quadratic trust-region so that DDP should return the previous nominal trajectory. Hops to different local optima can be encouraged by drawing the random perturbations from long-tailed Cauchy and Pareto distributions as suggested in Reference [71]. The results presented here correspond to Pareto distribution sampling. Perturbations were applied to the time variables only. Control variables and Lagrange multipliers were carried through each hop.

Introduction of mission event dates as decision variables adds many local optima to the trajectory optimization problem. Furthermore, if the initial and final orbits lie in different orbital planes, as with the Earth and Mars, then opportunities across different synodic periods are not equivalent. The parameter $\rho_{time-hop}$ sets a percent likelihood that a hop shifts the time variables to a different period. If a sample from the standard uniform distribution is below that value, the time variables are shifted forward or backward one time period. Selection of the time period is problem dependent and should be set as some fraction or multiple of orbital or synodic period to explore different families of trajectories. A threshold of $\rho_{time-hop} = 0.2$ was set to hop one synodic period between departure opportunities in 2007 and 2009.

Figure 3.2 shows the progress of MBH through one hour of runtime for the Earth-Mars rendezvous. Each point corresponds to a successful hop, where DDP converged to an improved local optimum. The process is stochastic, so repeated trials should produce different histories of successful MBH iterations. It is possible to not make any progress in the allotted time but also to find the global optimum in the first hop. The current result shows minor improvements in the first few hops until the first hop in synodic period leads to a significant improvement in the delivered mass. Steady progress is maintained even through additional synodic hops that only offer minor gains. Large changes are possible without the synodic hop, as evidenced with two large jumps within 2009 departures before the delivered mass levels off to negligible growth.
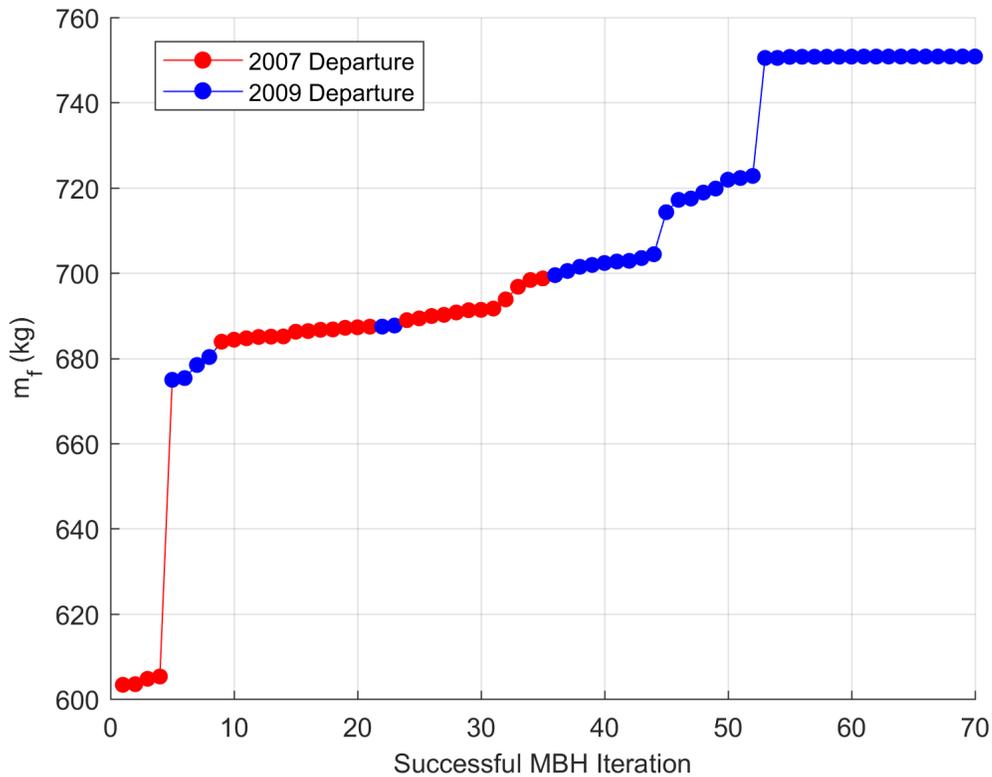
Figure 3.2: Progress of MBH iterations with indication of hops in synodic period for the Earth-Mars rendezvous with time variables.

Figure 3.3a shows the final trajectory and Figure 3.3b shows the corresponding thrust profile. The initial time has advanced by 775.4 days to May 24, 2009 and the final time has advanced by 917.60 days to September 27, 2010. Time of flight has increased to 491 days alongside the improved delivered mass of 750.82 kg. Thrusting does not start immediately, but the trajectory begins with a 49 day coast. Departure may then occur on July 12, 2009 with a time of flight of 442 days. Fuel-optimization should yield a departure maneuver at maximum throttle, not at an intermediate thrust level as shown. This can be resolved by increasing the resolution of stages so that stage times t_k occur close to the optimal on/off control switches for the departure maneuver.
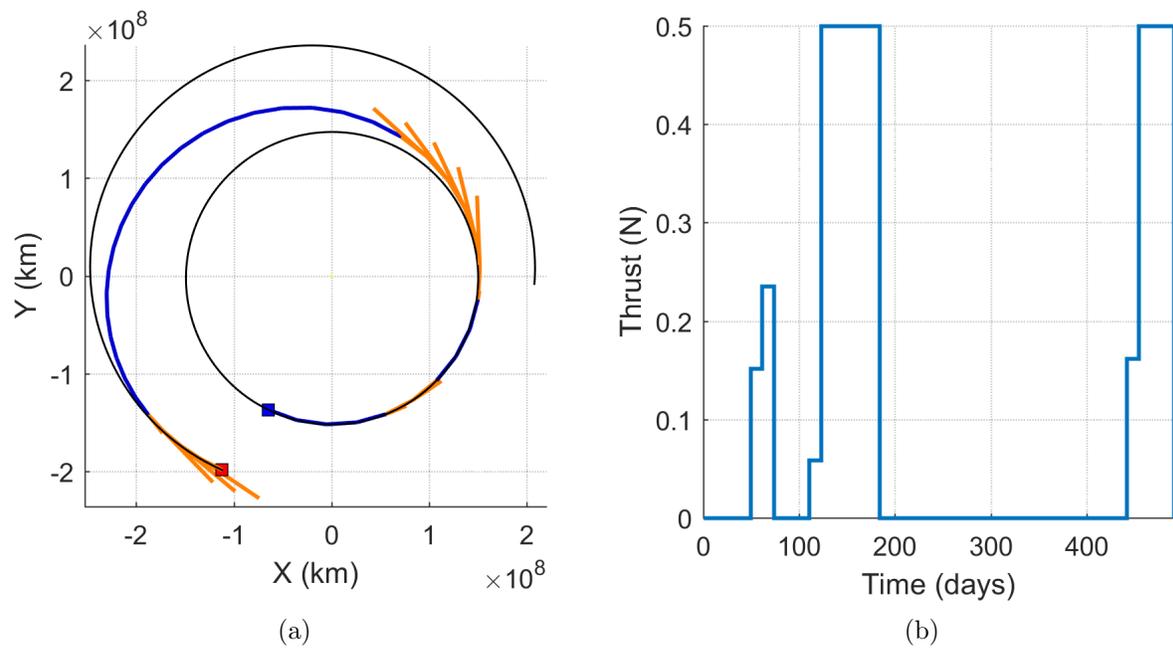
Figure 3.3: (a) Ecliptic view and (b) thrust profile of the final Earth-Mars rendezvous transfer after monotonic basin hopping.

Chapter 4

Sundman-Transformed Differential Dynamic Programming

The main contribution of this thesis is a method for the optimization of low-thrust many-revolution spacecraft trajectories. The method is to apply a Sundman transformation to change the independent variable of the spacecraft equations of motion to an orbit anomaly and perform the optimization with DDP. The Sundman transformation is a general change of variable from time to a function of orbital radius and effectively regulates the step size of numerical integration [66, 72]. Motivation for the change of variable stems from the preference for fixed-step integration for speed and optimizer performance. The advantage is notable for elliptic orbits, where evenly spaced time steps undersample the trajectory through periapsis and oversample through apoapsis. Undersampled periapsis passes induce fixed-step integration error that compounds as the number of revolutions accumulates. Automatic step size adjustment provided by the Sundman transformation protects the validity of fixed-step integration while avoiding the pitfalls associated with error control in a variable-step integrator [73]. Berry and Healy assessed numerical integration accuracy and computational speed to establish an eccentricity threshold at which the transformation proves more effective than time-integration [74]. Pellegrini, Russell and Vittaldev showed accuracy and efficiency gains for propagation with the Sundman transformation in the Stark and Kepler models [75]. Yam, Izzo and Biscani previously applied a Sundman transformation to a continuous-thrust extension of the Sims-Flanagan transcription to optimize interplanetary trajectories [76, 77].

This chapter presents Sundman-transformed DDP and numerical examples. The DDP algorithm does not change, but the dynamics in the forward pass and STM computation step are

reformed for different independent variables. Fuel-optimal geocentric transfers are computed with the transfer duration extended up to 2000 revolutions. The flexibility of the approach to higher fidelity dynamics is shown with Earth's J_2 perturbation and lunar gravity included for a 500 revolution transfer. Sundman transformations to the true, mean and eccentric anomalies are investigated and computational performance is summarized.

4.1   The Sundman Transformation

In regularizing the equations of motion to solve the three-body problem, Karl Sundman introduced a change of independent variable from time t to the new independent variable τ [66].

$$dt = c_n r^n d\tau \tag{4.1}$$

Time and the new independent variable are related by a function of the orbital radius, r. The real number n and coefficient c_n may be selected so that τ represents an orbit anomaly. For $n = 0, 1, 2$, the independent variables are the mean anomaly M, eccentric anomaly E and true anomaly ν. Those transformations are given by

$$dt = \sqrt{\frac{a^3}{\mu}}\, dM, \tag{4.2a}$$

$$dt = \sqrt{\frac{a}{\mu}}\, r dE, \tag{4.2b}$$

$$dt = \frac{r^2}{h} d\nu, \tag{4.2c}$$

where a is the semi-major axis, μ is the gravitational parameter of the central body, and h is the angular momentum magnitude.

4.1.1   Sundman-Transformed Dynamics

Transforming the time-dependent equations of motion simply requires multiplication by the functional relationship between the two independent variables.

$$\mathring{\boldsymbol{X}} = \frac{d\boldsymbol{X}}{d\tau} = \left(\frac{d\boldsymbol{X}}{dt}\right)\frac{dt}{d\tau} = \dot{\boldsymbol{X}} c_n r^n \tag{4.3}$$

In Equation 4.3, $\boldsymbol{X}$ represents a state vector, the overhead dot denotes a time derivative and the overhead ring denotes a derivative with respect to τ. Time-dependent equations of motion $\dot{\boldsymbol{X}}$ are typically propagated for a prescribed duration of time from the state at an initial epoch. Now, however, propagation is specified for a duration of τ. The relevant transition function for the Sundman-transformed dynamics in Equation 4.3 is

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \int_{\tau_k}^{\tau_{k+1}} \mathring{\boldsymbol{X}}_k(\tau)\, d\tau. \tag{4.4}$$

When τ is an orbit anomaly, the number of revolutions N_{rev} may be specified so that the τ duration is $2\pi N_{rev}$. The elapsed time is unknown *a priori* but may be tracked by including t in the state vector and numerically integrating $dt/d\tau$.

4.1.2   The Sundman-Transformed State Transition Matrix and Tensor

As with the state dynamics transformation, the dynamics matrix and tensor need to be transformed to reflect the change of independent variable.

$$\Lambda^{i,a} = \frac{\partial \mathring{\boldsymbol{X}}^i}{\partial \boldsymbol{X}^a} \tag{4.5a}$$

$$\Lambda^{i,ab} = \frac{\partial^2 \mathring{\boldsymbol{X}}^i}{\partial \boldsymbol{X}^a \partial \boldsymbol{X}^b} \tag{4.5b}$$

The new dynamics matrix and tensor are obtained first with respect to time then their transformed counterparts in Equations 4.5a and 4.5b are obtained by extensive application of the chain rule. First, the general Sundman transformation is redefined along with its first and second derivatives with respect to the state vector.

$$\eta = dt/d\tau = c_n r^n \tag{4.6}$$

$$\eta_X{}^i = \frac{\partial \eta}{\partial \boldsymbol{X}^i} \tag{4.7}$$

$$\eta_{XX}{}^{i,a} = \frac{\partial^2 \eta}{\partial \boldsymbol{X}^i \partial \boldsymbol{X}^a} \tag{4.8}$$

The variable η is not to be confused with the propulsion system engine efficiency in Equation 2.7. After assembling Equations 3.23a and 3.23b, the chain rule completes the transformation,

$$\Lambda^{i,a} = A^{i,a}\eta + \dot{\boldsymbol{X}}^i \eta_X{}^a, \tag{4.9a}$$

$$\Lambda^{i,ab} = A^{i,ab}\eta + A^{i,a}\eta_X{}^b + A^{i,b}\eta_X{}^a + \dot{\boldsymbol{X}}^i \eta_{XX}{}^{a,b}, \tag{4.9b}$$

and the transformed differential equations for the STMs become

$$\mathring{\Phi}^{i,a} = \Lambda^{i,\gamma_1}\Phi^{\gamma_1,a}, \tag{4.10a}$$

$$\mathring{\Phi}^{i,ab} = \Lambda^{i,\gamma_1}\Phi^{\gamma_1,ab} + \Lambda^{i,\gamma_1\gamma_2}\Phi^{\gamma_1,a}\Phi^{\gamma_2,b}. \tag{4.10b}$$

4.2 GTO to GEO Transfer in Cartesian Coordinates

Fuel-optimal low-thrust transfers from a geostationary transfer orbit (GTO) to geostationary orbit (GEO) were computed to demonstrate the efficacy of the Sundman-transformed DDP approach.

4.2.1 Spacecraft State and Dynamics

The spacecraft state is given by a Cartesian representation of the spacecraft inertial position and velocity. Implementation and notational complexities of DDP can be mitigated to some degree through the use of an augmented state vector of both states and controls. Time is additionally included as a state variable for use with the Sundman transformation. The augmented state vector includes time, position, velocity, mass m, and thrust control variables u, α and β.

$$\boldsymbol{X} = \begin{bmatrix} t & x & y & z & \dot{x} & \dot{y} & \dot{z} & m & u & \alpha & \beta \end{bmatrix}^T \tag{4.11}$$

Spherical thrust control is defined using the RSW frame in Equation 2.11. Spacecraft dynamics consider geocentric two-body motion perturbed by thrust, J_2 and lunar gravity,

$$\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}_\oplus + \dot{\boldsymbol{X}}_T + \dot{\boldsymbol{X}}_{J_2} + \dot{\boldsymbol{X}}_{\mathbb{C}}, \tag{4.12}$$

where $\dot{X}_\oplus$ is the two-body motion due to point mass Earth gravity, $\dot{X}_T$ is the thrust acceleration and mass flow rate for a constant power model, $\dot{X}_{J_2}$ is Earth's J_2 perturbation, and $\dot{X}_{\mathbb{C}}$ is the point mass lunar gravity perturbation. The Sundman transformation is made after assembling the complete equations of motion with respect to time.

$$\mathring{X} = \left(\dot{X}_\oplus + \dot{X}_T + \dot{X}_{J_2} + \dot{X}_{\mathbb{C}} \right) \eta \tag{4.13}$$

The time derivatives are defined by

$$\dot{X}_\oplus = \begin{bmatrix} 1 & \dot{x} & \dot{y} & \dot{z} & -\dfrac{\mu_\oplus}{r^3}x & -\dfrac{\mu_\oplus}{r^3}y & -\dfrac{\mu_\oplus}{r^3}z & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.14a}$$

$$\dot{X}_T = \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{T_a}{m}u_x & \dfrac{T_a}{m}u_y & \dfrac{T_a}{m}u_z & -\dfrac{T_a}{I_{sp}g_0}u & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.14b}$$

$$\dot{X}_{J_2} = -\dfrac{3J_2\mu_\oplus R_\oplus^2}{2r^5} \begin{bmatrix} 0 & 0 & 0 & 0 & x\left(1 - 5\dfrac{z^2}{r^2}\right) & y\left(1 - 5\dfrac{z^2}{r^2}\right) & z\left(3 - 5\dfrac{z^2}{r^2}\right) & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{4.14c}$$

$$\dot{X}_{\mathbb{C}} = -\mu_{\mathbb{C}} \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{x - x_{\mathbb{C}}}{\|r - r_{\mathbb{C}}\|^3} + \dfrac{x_{\mathbb{C}}}{r_{\mathbb{C}}^3} & \dfrac{y - y_{\mathbb{C}}}{\|r - r_{\mathbb{C}}\|^3} + \dfrac{y_{\mathbb{C}}}{r_{\mathbb{C}}^3} & \dfrac{z - z_{\mathbb{C}}}{\|r - r_{\mathbb{C}}\|^3} + \dfrac{z_{\mathbb{C}}}{r_{\mathbb{C}}^3} & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{4.14d}$$

Gravitational parameters for the Earth and the Moon are $\mu_\oplus$ and $\mu_{\mathbb{C}}$, respectively. The J_2 perturbation is owed to the Earth's oblateness and is a function of the Earth's equatorial radius $R_\oplus$. Table 4.1 lists these dynamic model constants. Including the lunar perturbation requires the Moon's inertial position with respect to the Earth,

$$r_{\mathbb{C}} = \begin{bmatrix} x_{\mathbb{C}} & y_{\mathbb{C}} & z_{\mathbb{C}} \end{bmatrix}^T, \tag{4.15}$$

that is assumed to evolve according to geocentric two-body motion. The Moon's state is initialized with the orbital elements listed in Table 4.2.

Table 4.1: Dynamic model parameters for the example GTO to GEO transfer in Cartesian coordinates.

$\mu_\oplus$	398600.44 km^3/s^2	$\mu_{\mathbb{C}}$	4904.928372 km^3/s^2
$R_\oplus$	6378.136 km	g_0	0.00980665 km/s^2
J_2	0.0010826265		

Table 4.2: The Moon's Earth-centered ICRF orbital elements at 01 Jan 2000 00:00:00.0 TDB.

a	381218.68756119 km	Ω	12.23324045°
e	0.06476694	ω	60.78357549°
i	20.94024252°	M	140.74025588°

4.2.2 Augmented Lagrangian Cost Function

Final conditions for GEO are described in the terminal constraint function ψ, so that $\psi(\boldsymbol{X}_f) = 0$ for a feasible solution. The cost in Equation 3.9 also includes an objective function defined so that the final mass is maximized and does not include local stage costs L_k.

$$\phi = -m_f \tag{4.16a}$$

$$\psi = \begin{bmatrix} \|\boldsymbol{r}_f\| - r_{target} \\ \|\boldsymbol{v}_f\| - v_{target} \\ \boldsymbol{r}_f \cdot \boldsymbol{v}_f \\ z_f \\ \dot{z}_f \end{bmatrix} \tag{4.16b}$$

$$\Sigma = I_{5\times5} \tag{4.16c}$$

The terminal constraint function is satisfied upon arrival at GEO distance with circular orbital velocity, zero flight path angle and zero inclination. The arrival longitude is unconstrained. The penalty matrix is set as the identity matrix so that all constraints are weighted equally. A scaled feasibility tolerance requires $\|\psi\| < 1 \times 10^{-8}$ and an optimality tolerance requires $ER_0 < 1 \times 10^{-9}$. Scaling improves the numerical behavior of both trajectory computation and optimization but adds to the set of tuning parameters. Here a distance unit DU, time unit TU, force unit FU and mass unit MU are set as

$$\begin{aligned} DU &= r_{target}, \\ TU &= 10\sqrt{DU^3/\mu_\oplus}, \\ FU &= T_{max}, \\ MU &= FU\,TU^2/\,DU, \end{aligned} \tag{4.17}$$

where the distance unit is approximately GEO radius, the time unit has been scaled by an additional factor of 10 and the force unit is the maximum thrust.

4.2.3   Trajectory Computation

Propagating a trajectory from the initial state requires effective discretization and a numerical method to evaluate the transition function between stages. Trajectories are discretized to 100 stages per revolution and were propagated with an RK8 integrator. Stages are equally spaced in the independent variable. Each stage offers an opportunity to update the thrust control variables that are held constant across an integration step. A fixed integration step accumulates $\Delta\tau = \tau_{k+1} - \tau_k = 2\pi/100$. The initial guess for all stage control variables is zero, so the first iteration considers a ballistic orbit in GTO for a prescribed number of revolutions, N_{rev}. The fixed transfer duration in orbit anomaly is $2\pi N_{rev}$ and there are $300 N_{rev}$ control variables to solve for.

4.2.4   STM Computation

The DDP implementation is programmed in C++ and compiled on the RMACC Summit supercomputer [78]. Results were generated on a single node that contains two Intel Xeon E5-2680 v3, 2.50 GHz CPUs with 12 cores each. STM computations were distributed in parallel across all 24 cores with OpenMP [79]. All other steps of the algorithm run serially. To permit parallelization, STMs are obtained separately from attempted trajectories with trial controls. When an iterate is accepted as the new nominal trajectory, Equation 4.4 is augmented with the STM differential equations in Equation 4.10. Each $\boldsymbol{X}_k$ is known, so integrations from any stage k to $k+1$ can be performed separately and in parallel, instead of serially from $k = 0$ to $k = N - 1$. The states at integrator substeps are not saved, so the 11 state differential equations are recomputed with the $11^2 + 11^3$ STM differential equations.

4.2.5 Case B Transfer Results

Petropoulos et al. [24] compared direct and indirect methods and the Q-law over several cases of many-revolution transfers, including a GTO to GEO example named Case B. Now Sundman-transformed DDP is added to the comparison. Table 4.3 describes the initial and target orbits, with slight differences between Reference [24] and the DDP transfer. Eccentricity and inclination targets were set to zero for use of Equation 4.16. The dynamics consider two-body motion and thrust, but not J_2 and lunar perturbations, so the equal reduction to initial and target inclination is inconsequential to the comparison. The DDP transfer requires more effort to completely circularize the final orbit, but the dynamics use a slightly smaller gravitational parameter for the Earth. The effects are small and offsetting with regard to fuel expenditure and time of flight, but have no qualitative implications. The scaled feasibility tolerance corresponds to a 0.42165 m position requirement and .003075 mm/s velocity requirement.

Table 4.3: Initial and target orbits for the Case B transfer.

Orbit	a (km)	e	i (deg)	T_{max} (N)	m_0 (kg)	I_{sp} (sec)	$\mu_\oplus$ (km^3/s^2)
Initial [24]	24505.9	0.725	7.05	0.350	2000	2000	398600.49
Target [24]	42165	0.001	0.05				
Initial (DDP)	24505.9	0.725	7.0	0.350	2000	2000	398600.44
Target (DDP)	42165	0	0				

Transfers were computed across a range of N_{rev} and with Sundman transformations to each of the true, mean and eccentric anomalies. Attempts with time as the independent variable were unsuccessful. The result is a Pareto front of propellant mass versus N_{rev} as shown in Figure 4.1a. That is not identically the propellant mass versus time of flight trade-off. The target orbit might be reached before the final revolution is completed, resulting in a terminal coast arc that adds extra flight time. Nonetheless, the Pareto front of propellant mass versus time of flight shown in Figure 4.1b is consistent with Petropoulos' result.

The intuitive limiting cases are continuous thrust for the minimum time of flight transfer that reduces to impulsive apoapsis maneuvers for infinite flight time. The trend is evidenced
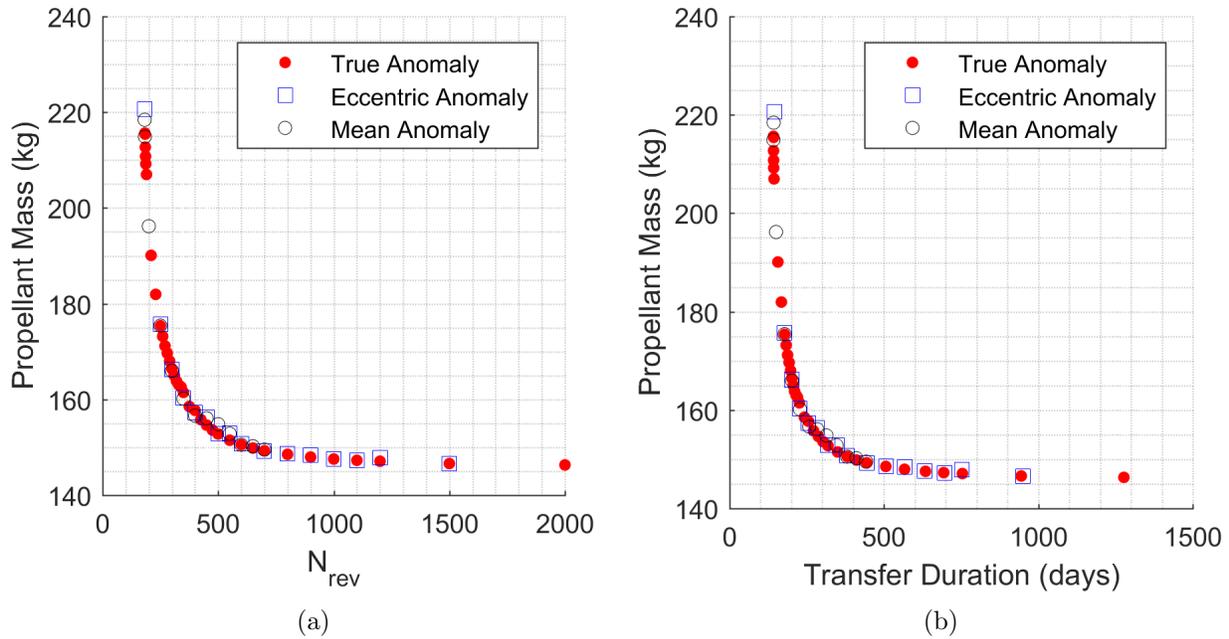
Figure 4.1: Case B trade-off of propellant mass versus time of flight and number of revolutions.

in Figure 4.2, where coast arcs (colored blue) grow in duration and number, while thrust arcs (colored orange) reduce to small maneuvers centered on their optimal locations.Thrust arcs are colored orange and coast arcs are colored blue. The first and final revolutions are overlaid in black. Figure 4.3 shows the thrust magnitude history for the 500 revolution transfer as evidence of the expected bang-bang control structure for the fuel-optimal transfer. It also proves worthwhile for the shorter duration transfers to raise apoapsis beyond GEO radius for more efficient inclination change. The change in maneuver strategy as time of flight increases is shown in Figure 4.4, with apsis radii distances and inclination provided through the duration of the 183 and 500 revolution true anomaly transfers. The 183 revolution true anomaly transfer spans 141.95 days and uses 215.64 kg of propellant, but falls short of finding the minimum time solution with a number of perigee centered coast arcs. On the other end, the Sundman-transformed DDP approach is able to extend the transfer duration out to 2000 revolutions for a 1276.83 days time of flight requiring 146.32 kg of propellant. There are limited returns on extending the flight time, as the 1000 revolution transfer

shown only increases the propellant use up to 147.55 kg for 634.35 days time of flight.

The accuracy of each DDP solution was checked by variable-step RK8(7) integration with a relative error tolerance of 10^{-11} [69]. The resulting constraint violations are shown in Figure 4.5. Solutions with mean anomaly as the independent variable prove unreliable, with actual constraint violations growing six orders of magnitude from what was viewed as a feasible trajectory with fixed-step integration. As with fixed-step time integration, mean anomaly integration incurs errors through large steps around periapsis. Mean anomaly integration, however, automatically adapts the step size as the orbit changes during the transfer, whereas time integration steps remain fixed. An adaptive-mesh technique might enable successful time integration, but that is essentially the point of the Sundman transformation. DDP attempts with mean anomaly failed when N_{rev} was increased from 700 to 800. The true anomaly and eccentric anomaly prove more effective in regulating the step size. Most eccentric anomaly trajectories fall just outside of feasibility with constraint violations $10^{-8} < \|\boldsymbol{\psi}\| < 10^{-7}$, except for the 1000 and 1500 revolution cases that remain feasible. Every true anomaly trajectory remains feasible. The superiority of true anomaly was first realized for a coarse range of N_{rev} before proceeding exclusively with true anomaly through a finer resolution of N_{rev}. Data points are absent for the 200, 220 and 240 revolution true anomaly transfers and 200 and 650 revolution eccentric anomaly transfer. For these cases, the algorithm found a stationary point that was just outside of feasibility, again in the region $10^{-8} < \|\boldsymbol{\psi}\| < 10^{-7}$. Given the range of feasible trajectories, this might be resolved by adjusting the many tuning parameters. Optimization of the 2000 revolution eccentric anomaly transfer was cut off after 48 hours of runtime on iteration 1556.

Computational performance is profiled in Figures 4.6 and 4.7. The quickest solution time was 20 minutes and 40 seconds to compute the 184 revolution true anomaly case in 186 iterations. The lowest number of iterations to convergence was 181 for the 187 revolution true anomaly case and required 21 minutes and 1 second. Generally, both runtime and number of iterations grow with the number of revolutions. The behavior is unpredictable, especially with excessive transfer duration, where the runtime and iterations might jump up or down by a factor of two or more. Runtimes for the forward pass and STM subroutines however, do have a predictable linear growth.
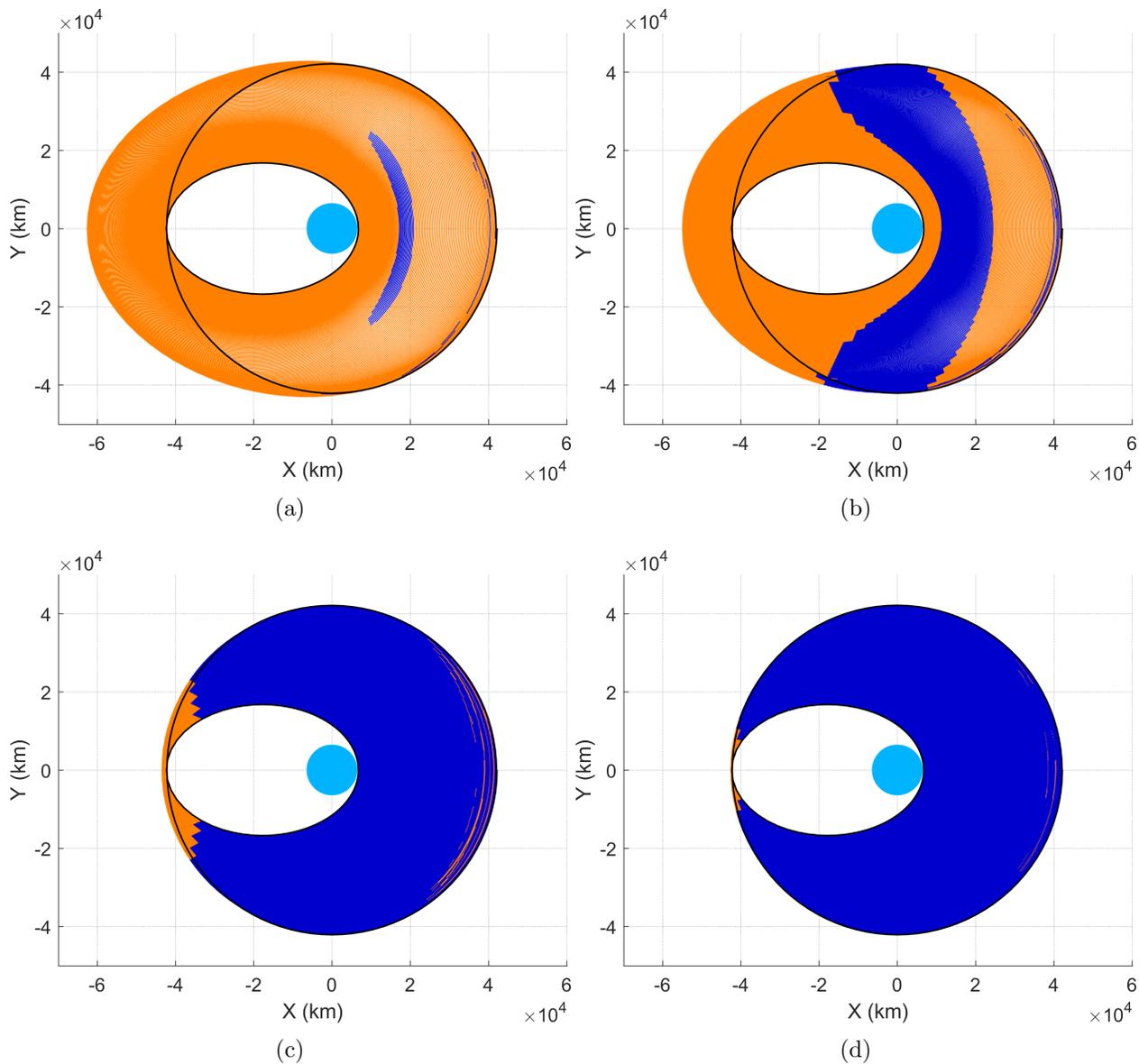
Figure 4.2: Equatorial projections of Case B transfers for (a) 183, (b) 210, (c) 500 and (d) 1000 revolutions with true anomaly as the independent variable.
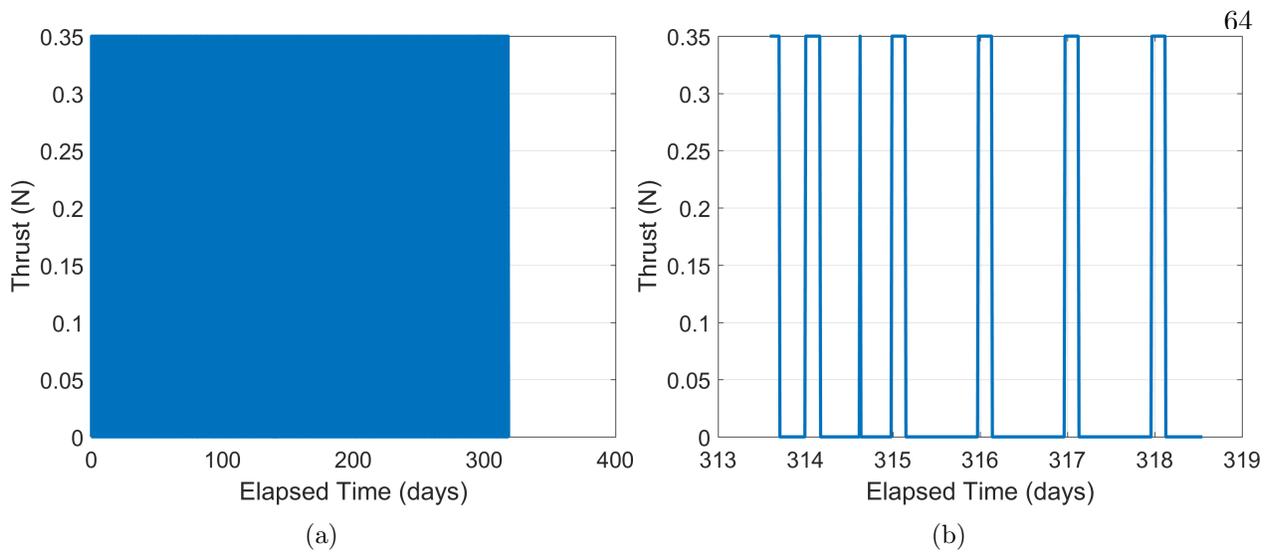
Figure 4.3: Thrust magnitude for the 500 revolution true anomaly transfer is shown for (a) the entire transfer and (b) zoomed in on the last few revolutions.

The listed times correspond to the first iteration that is ballistic propagation in GTO. Computing the STMs is the most computationally intensive step in the DDP algorithm. The importance of parallelization is highlighted when comparing the parallel STM elapsed real time to the total CPU time across all cores, and realizing that the STM step would slow by an order of magnitude in the current configuration.

4.2.6   Case B with Perturbations

Next, the robustness of the approach is tested by introducing J_2 and lunar gravity perturbations to the 500 revolution Case B transfers with both eccentric and true anomalies as independent variables. Perturbations are introduced one at a time, so that the new cases are J_2-perturbed and J_2 and Moon-perturbed transfers.

Equatorial projections of the new transfers in Figure 4.8 show the pronounced effect of the J_2-induced periapsis drift over a long transfer duration. The effect of lunar gravity is less noticeable. Three-dimensional views of the two-body and J_2 and Moon-perturbed true anomaly transfers are provided in Figure 4.9. Trajectory performance is compared with the two-body case in Table 4.4,
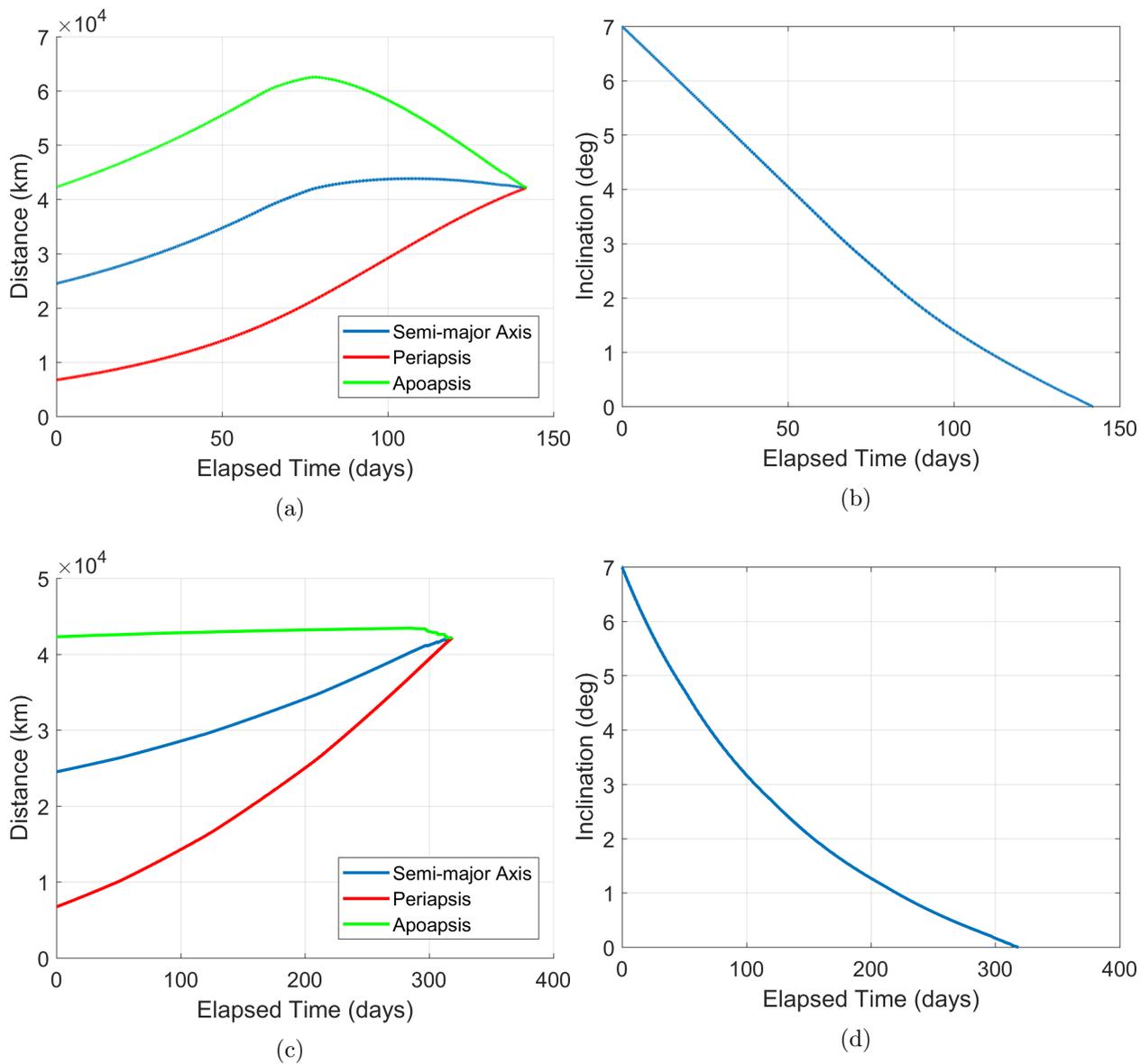
Figure 4.4: (a) Apsis radii and (b) inclination history for the 183 revolution true anomaly transfer and (c) apsis radii and (d) inclination history for the 500 revolution true anomaly transfer.
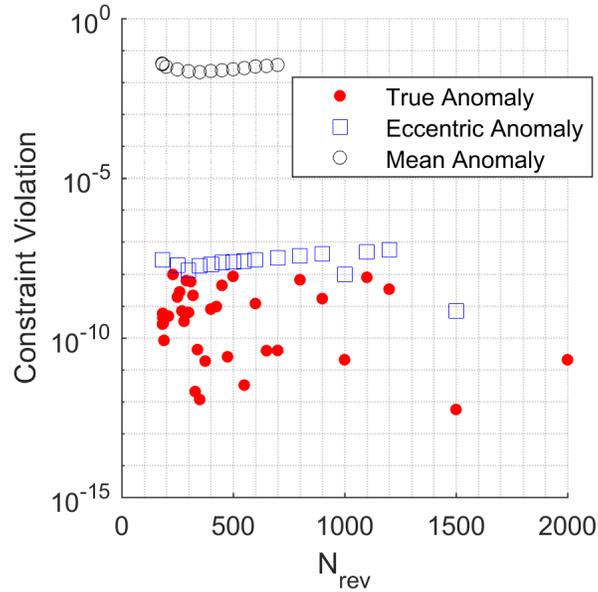
Figure 4.5: Constraint violations after fixed-step integration solutions were recomputed with variable-step integration and a relative error tolerance of 10^{-11}.

(a)

(b)
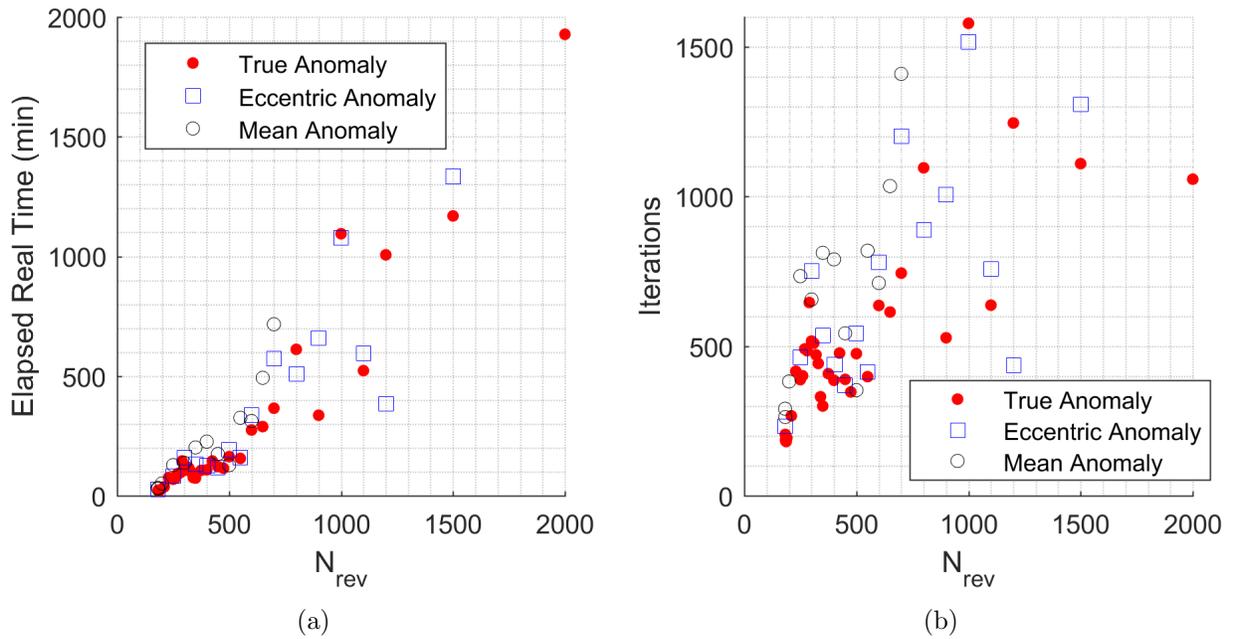
Figure 4.6: (a) Elapsed real time and (b) number of iterations to convergence for Case B transfers.

where m_p is the propellant mass. The new cases are unable to leverage perturbations to improve
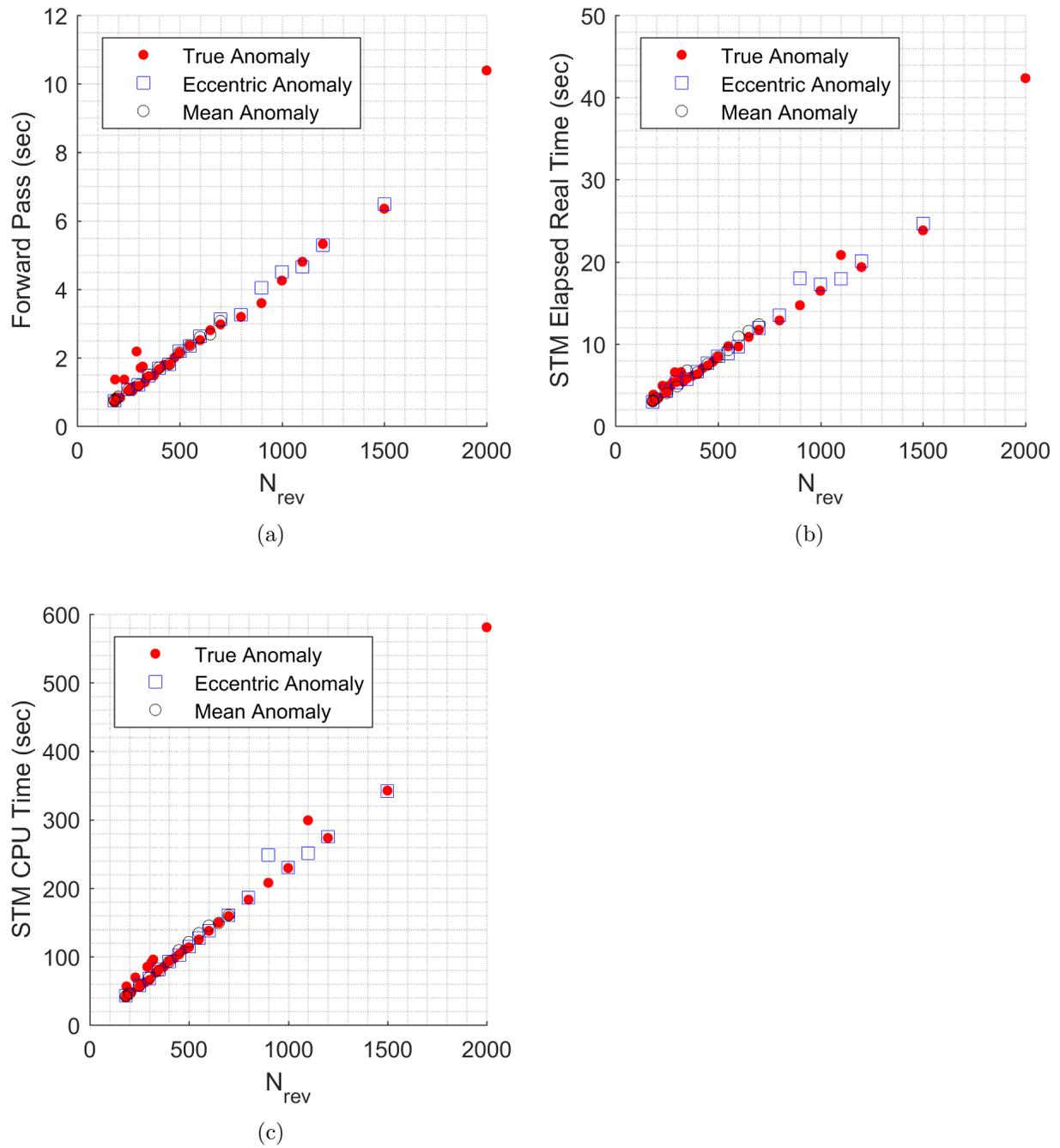
Figure 4.7: Elapsed real time for (a) forward pass and (b) STM subroutines and (c) the total CPU time for the STM subroutine for Case B transfers.

upon the two-body result, but propellant mass and time of flight reach similar values. Solution accuracy when checked with RK8(7) integration remains consistent with the two-body results and is summarized in Table 4.5. Computational performance is summarized in Table 4.6. These measures collectively add the significant result that Sundman-transformed DDP can accommodate perturbations to yield a reliable solution without detriment to the computational effort. That is not for free, as seen by the effect on subroutine runtimes in Table 4.7. Including J_2 is an inexpensive addition relative to the cost of computing the lunar perturbation and the necessary derivatives with respect to the spacecraft state. Table 4.7 adds the comparison to the elapsed real time of a serial STM step, as opposed to the parallel STM step total CPU time in Figure 4.7c, and shows an order of magnitude speed improvement owed to parallelization.

Table 4.4: 500 revolution Case B transfer results.

Perturbations	True Anomaly		Eccentric Anomaly	
	m_p (kg)	t_f (days)	m_p (kg)	t_f (days)
None	152.77	318.55	152.83	317.67
J_2	157.65	319.05	159.01	314.17
J_2 and Moon	156.93	320.08	159.20	313.93

Table 4.5: Constraint violations after variable-step integration of 500 revolution Case B fixed-step solutions.

Perturbations	True Anomaly	Eccentric Anomaly
none	8.3368×10^{-9}	2.3907×10^{-8}
J_2	1.7207×10^{-10}	2.4244×10^{-8}
J_2 and Moon	1.0832×10^{-11}	2.5892×10^{-8}

4.3 Sundman-Transformed DDP in Modified Equinoctial Elements

The preceding sections applied a Sundman transformation to DDP to enable efficient optimization of low-thrust many-revolution trajectories. Appropriately choosing the transformation allows the integrator to automatically take smaller steps at small orbital radii and larger steps at
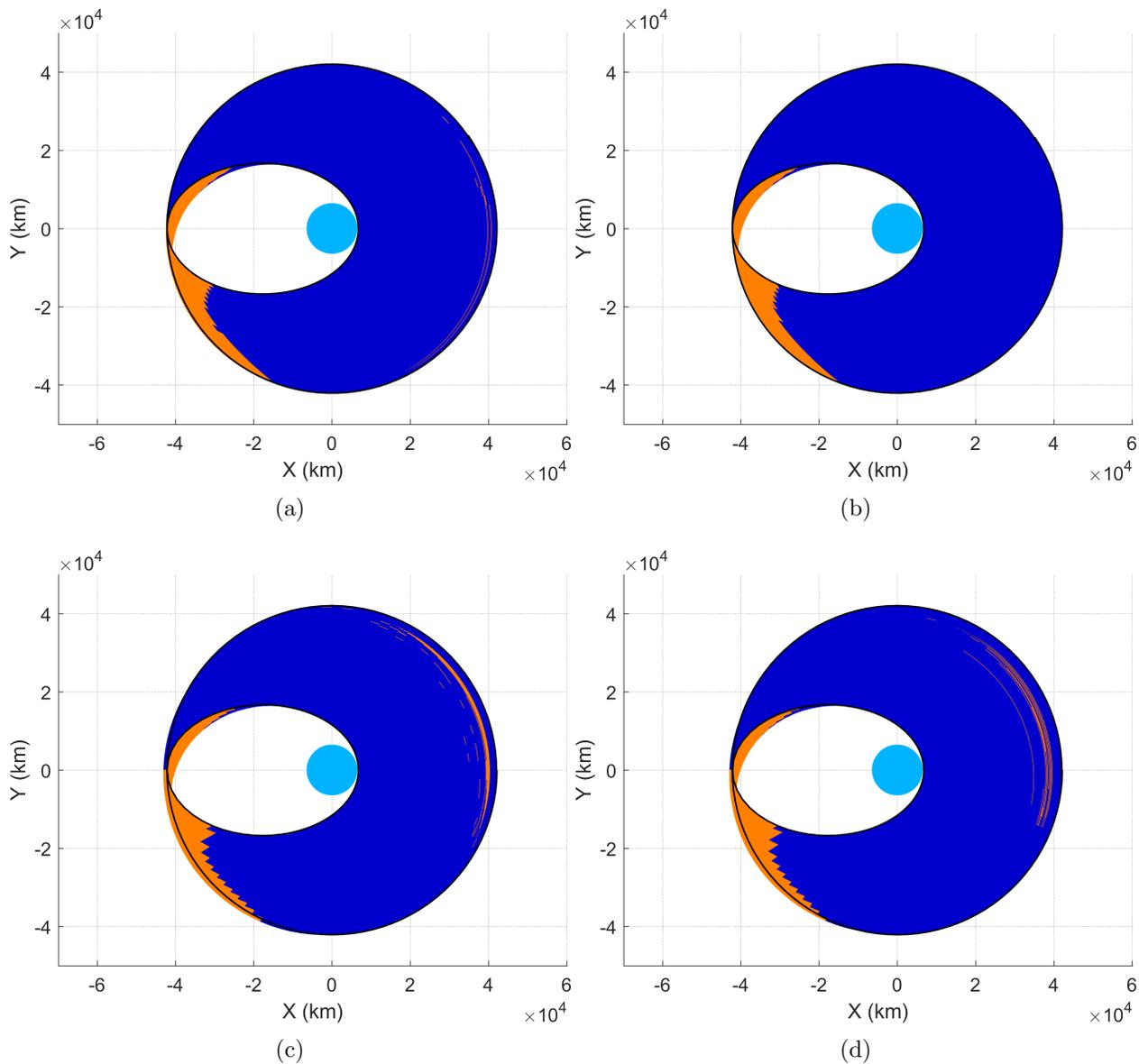
Figure 4.8: Equatorial projections of Case B eccentric anomaly transfers with (a) J_2 and (b) J_2 and Moon perturbations and true anomaly transfers with (c) J_2 and (d) J_2 and Moon perturbations.

(a)

(b)
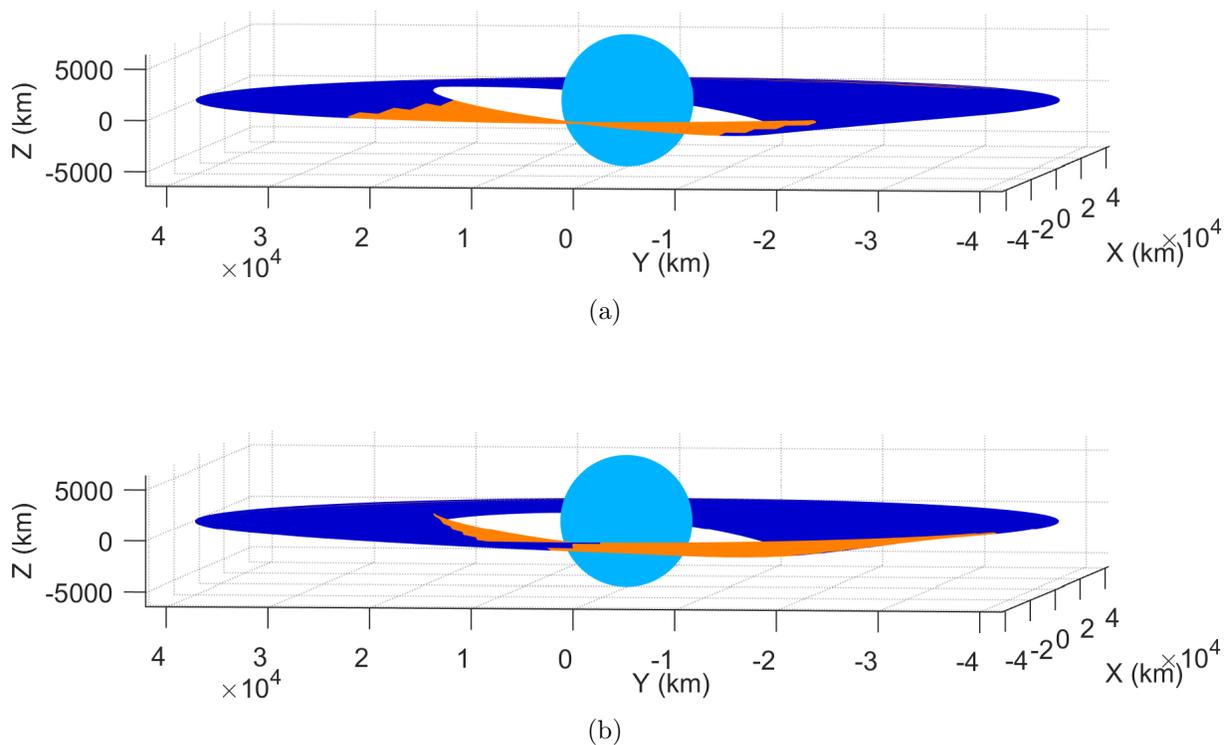
Figure 4.9: Three dimensional views of the Case B (a) two-body and (b) J_2 and Moon-perturbed true anomaly transfers.

large orbital radii. Fine discretization requirements are mitigated to some degree by the transformation, as oversampling of the trajectory is easily avoided while accuracy is maintained at lower radii. However, the rapid dynamics of a Cartesian state representation also drove a fine discretization requirement. Accurate integration can be accomplished in fewer steps through use of slow changing state variables, i.e. orbital elements. The remainder of this chapter details how the modified equinoctial orbital element set enters the Sundman-transformed DDP approach. The GTO to GEO transfer from Reference [60] is revisited to show the benefit of using modified equinoctial elements.

Table 4.6: Computational performance for 500 revolution Case B transfers.

	Perturbations	Iterations	Elapsed Real Time (min)
True Anomaly	None	475	163
	J_2	489	166
	J_2 and Moon	596	237
Eccentric Anomaly	None	542	193
	J_2	524	200
	J_2 and Moon	371	151

4.3.1   Sundman Transformations in Modified Equinoctial Elements

The Sundman transformations to each of the true, mean and eccentric anomalies in terms of the modified equinoctial elements are

$$dt = \frac{1}{\sqrt{\mu p}} \left(\frac{p}{q}\right)^2 d\nu, \tag{4.18a}$$

$$dt = \frac{1}{\sqrt{\mu}} \left(\frac{p}{1 - f^2 - g^2}\right)^{3/2} dM, \tag{4.18b}$$

$$dt = \frac{1}{\sqrt{\mu}} \frac{p}{q} \sqrt{\frac{p}{1 - f^2 - g^2}} \, dE. \tag{4.18c}$$

4.3.2   The Augmented Modified Equinoctial Element State Vector

The augmented Cartesian (*IJK*) state vector is redefined with a new subscript notation and the augmented modified equinoctial element (*MEE*) state vector is introduced as

$$\boldsymbol{X}_{IJK} = \begin{bmatrix} t & x & y & z & \dot{x} & \dot{y} & \dot{z} & m & T & \alpha & \beta \end{bmatrix}^T, \tag{4.19a}$$

$$\boldsymbol{X}_{MEE} = \begin{bmatrix} t & p & f & g & h & k & L & m & T & \alpha & \beta \end{bmatrix}^T. \tag{4.19b}$$

The equations of motion are redefined in terms of the augmented state vector,

$$\dot{\boldsymbol{X}}_{MEE} = A\boldsymbol{\Delta}_{\text{Thrust}} + \dot{m}T_a + AQ^T\boldsymbol{\delta}_p + \boldsymbol{b}, \tag{4.20}$$

Table 4.7: Subroutine elapsed real time in seconds for 500 revolution Case B transfers.

	Perturbations	Forward Pass	Parallel STMs	Serial STMs
True Anomaly	none	2.10	8.10	85.76
	J_2	2.19	8.34	86.80
	J_2 and Moon	2.69	8.54	87.26
Eccentric Anomaly	none	2.18	8.13	86.69
	J_2	2.23	8.20	87.22
	J_2 and Moon	2.68	8.52	88.62

where

$$
A = \begin{bmatrix}
0 & 0 & 0 \\
0 & \dfrac{2p}{q}\sqrt{\dfrac{p}{\mu}} & 0 \\
\sqrt{\dfrac{p}{\mu}}\sin L & \sqrt{\dfrac{p}{\mu}}\dfrac{1}{q}[(q+1)\cos L + f] & -\sqrt{\dfrac{p}{\mu}}\dfrac{g}{q}(h\sin L - k\cos L) \\
-\sqrt{\dfrac{p}{\mu}}\cos L & \sqrt{\dfrac{p}{\mu}}\dfrac{1}{q}[(q+1)\sin L + g] & \sqrt{\dfrac{p}{\mu}}\dfrac{f}{q}(h\sin L - k\cos L) \\
0 & 0 & \sqrt{\dfrac{p}{\mu}}\dfrac{s^2\cos L}{2q} \\
0 & 0 & \sqrt{\dfrac{p}{\mu}}\dfrac{s^2\sin L}{2q} \\
0 & 0 & \sqrt{\dfrac{p}{\mu}}\dfrac{1}{q}(h\sin L - k\cos L) \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}, \qquad (4.21)
$$

$$
\dot{\boldsymbol{m}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\dfrac{T}{I_{sp}g_0} & 0 & 0 & 0 \end{bmatrix}^T, \qquad (4.22)
$$

$$
\boldsymbol{b} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p}\left(\dfrac{q}{p}\right)^2 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \qquad (4.23)
$$

$$
\boldsymbol{\delta}_p = \sum_{i=0}^{n_p} \boldsymbol{\delta}_{p_i}. \qquad (4.24)
$$

Now $\boldsymbol{b}$ captures the two-body dynamics, $\dot{\boldsymbol{m}}$ is the mass flow rate, $T \in [0,1]$ is the throttle, T_a is the thrust available computed from a power model and A transforms the contributions of thrust and the summation of n_p perturbations formed in the inertial frame.

Quadratic expansions of the cost-to-go are mapped through the backward sweep by the state transition matrix and second-order state transition tensor. Both are henceforth referred to

as the STMs. The STMs are the first and second derivatives of Equation 4.20 with respect to Equation 4.19b and are detailed in the appendix.

4.4 GTO to GEO Transfer in Modified Equinoctial Elements

The GTO to GEO transfer from Reference [60] first maximized the delivered mass in 450.5 revolutions subject to two-body dynamics. The Sundman transformation to eccentric anomaly was utilized. Model fidelity was improved to consider Earth's J_2 perturbation, and again with J_2 and lunar gravity. A final case increased the transfer duration to 1000.5 revolutions while including the J_2 and lunar gravity perturbations.

4.4.1 Boundary Conditions

Initial conditions for GTO are prescribed in terms of both the augmented *IJK* and *MEE* state vectors.

$$
\boldsymbol{X}_{IJK,0} = \begin{bmatrix} t_0 \\ x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \\ m_0 \\ T_0 \\ \alpha_0 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \text{01 Jan 2000 12:00:00.0 TDB} \\ 6678.1363 \text{ km} \\ 0 \\ 0 \\ 0 \\ 8.92130624 \text{ km/s} \\ 4.84387407 \text{ km/s} \\ 2000 \text{ kg} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.25a}
$$

$$
\boldsymbol{X}_{MEE,0} = \begin{bmatrix} t_0 \\ p_0 \\ f_0 \\ g_0 \\ h_0 \\ k_0 \\ L_0 \\ m_0 \\ T_0 \\ \alpha_0 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \text{01 Jan 2000 12:00:00.0 TDB} \\ 11530.089201 \text{ km} \\ 0.72654295 \\ 0 \\ 0.25396764 \\ 0 \\ 0 \\ 2000 \text{ kg} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.25b}
$$

The transfer begins at a 300 km perigee altitude on the x-axis with an inclination of $28.5°$ and apogee radius of 42164.169972 km. The 2000 kg spacecraft's propulsion system operates at a constant specific impulse of $I_{sp} = 1950$ sec with maximum thrust $T_{max} = 0.25$ N.

4.4.2   Augmented Lagrangian Cost Function

$$\boldsymbol{\psi}_{IJK} = \begin{bmatrix} \|\boldsymbol{r}_f\| - 42164.169972 \text{ km} \\ \|\boldsymbol{v}_f\| - 3.07466 \text{ km/s} \\ \boldsymbol{r}_f \cdot \boldsymbol{v}_f \\ z_f \\ \dot{z}_f \end{bmatrix}, \quad \boldsymbol{\psi}_{MEE} = \begin{bmatrix} p_f - 42164.169972 \text{ km} \\ f_f \\ g_f \\ h_f \\ k_f \end{bmatrix} \tag{4.26a}$$

$$\Sigma_{IJK} = diag(100, 10, 1, 1, 1), \quad \Sigma_{MEE} = diag(10, 10, 10, 10, 10) \tag{4.26b}$$

Terminal constraints for $\boldsymbol{X}_{IJK,f}$ prescribe a final GEO radius and velocity magnitude and zero flight path angle, position and velocity in the equatorial plane. Terminal constraints for $\boldsymbol{X}_{MEE,f}$ are stated simply as values for each component, with the final longitude free. Weights and scaling parameters were tuned by starting several trial runs to see that the initial iterations proceeded successfully. In effort to improve the numerical behavior for both trajectory computation and optimization, quantities are non-dimensionalized and scaled with a distance unit DU, time unit TU, force unit FU, and mass unit MU.

$$DU = 42164.0 \text{ km} \tag{4.27a}$$

$$TU = 10 \sqrt{DU^3/\mu_\oplus} \tag{4.27b}$$

$$FU = 0.25 \text{ N} \tag{4.27c}$$

$$MU = FU\,TU^2/DU \tag{4.27d}$$

Feasibility and optimality tolerances were set to 10^{-8} and 10^{-9}, respectively. Equations 4.16 and 4.27 and the feasibility tolerance sets the necessary position accuracy at 0.42164 m and velocitiy accuracy to .003075 mm/s.

4.4.3 Numerical Setup

Transfers are fixed in duration to a number of revolutions N_{rev} and were computed with a fixed-step, eighth-order Dormand-Prince method [69]. With eccentric anomaly as the independent variable, equations of motion were numerically integrated to $2\pi N_{\text{rev}}$. Trajectories are discretized into a number of stages per revolution that are equally spaced in eccentric anomaly. Examples computed with the *IJK* state vector are discretized to 100 stages per revolution and named with the resolution in the subscript, IJK_{100}. Previous IJK_{100} trajectories from Reference [60] were recomputed to reflect the current hardware and software configurations. *MEE* examples are discretized to both 100 and 24 stages per revolution, i.e. MEE_{100} and MEE_{24}.

The *MEE* state equations in Equation 4.20 and the STM differential equations that use Equations A.12 and A.13 require additional overhead when compared to those for the Cartesian state, and a numerical integration step is computationally more intensive. The improved accuracy of the *MEE* equations, however, permits a looser discretization of the trajectory, which can in turn offer a computational speedup. These assertions are quantified in Tables 4.8 to 4.10. With an equivalent resolution of 100 stages per revolution, MEE_{100} proves more demanding than IJK_{100} for both the forward pass and STM computation. *MEE* discretization was reduced for the J_2 and Moonperturbed 1000.5 revolution case until speedup in both the forward pass and STM computation was realized. This occurred when discretization was reduced to 24 stages per revolution. Table 4.10 shows that MEE_{24} suffers a slight but not detrimental loss in accuracy when compared to IJK_{100} for this scenario.

Table 4.9 compares the fixed-step, eighth-order integration error in position, velocity and time after one revolution in GTO subject to two-body dynamics. Those errors are with regard to the state obtained after variable-step integration with a seventh-order relative error tolerance of 10^{-15}. In addition to the time variable, only the true longitude varies for *MEE* through an integration step. For *IJK*, however, all position and velocity components evolve. Consequently, MEE_{100} shows improved accuracy by orders of magnitude over IJK_{100}. Table 4.10 summarizes the

Table 4.8: Subroutine runtimes in seconds for Cartesian states and modified equinoctial elements with different numbers of stages per revolution.

	Perturbations	N_{rev}	Forward Pass	Parallel STMs	Serial STMs
IJK_{100}	none	450.5	1.92	7.46	104.03
	J_2	450.5	2.00	7.44	103.30
	J_2 and Lunar Gravity	450.5	2.40	7.59	105.68
	J_2 and Lunar Gravity	1000.5	5.41	17.23	238.29
MEE_{100}	none	450.5	2.46	19.48	150.83
	J_2	450.5	2.52	30.01	331.22
	J_2 and Lunar Gravity	450.5	2.91	31.37	342.96
	J_2 and Lunar Gravity	1000.5	6.53	74.31	754.25
MEE_{24}	none	450.5	0.60	4.88	36.35
	J_2	450.5	0.60	7.74	78.69
	J_2 and Lunar Gravity	450.5	0.70	7.56	83.58
	J_2 and Lunar Gravity	1000.5	1.47	17.00	187.13

repeat study for the J_2 and Moon-perturbed case, with similar results.

The DDP implementation is a C++ program compiled on the RMACC Summit supercomputer [78]. Results were generated on a single node that contains two Intel Xeon E5-2680 v3, 2.50 GHz CPUs with 12 cores each. STM computations were distributed in parallel across all 24 cores with OpenMP [79]. Table 4.8 lists the computation time in seconds for single function calls of the forward pass and STM subroutines for each example trajectory. The forward pass is an entire trajectory computation with a new control profile and must be done serially. After an iterate is accepted as the new nominal trajectory, the states at all stages are already known from the forward pass and STMs may be computed in parallel. The significance of parallelization in DDP is evidenced by the substantially slower runtimes for serial STM computations listed alongside the parallel STM runtimes that are experienced in practice.

4.4.4   GTO to GEO Results

Table 4.11 summarizes the GTO to GEO transfers by providing the final masses and times of flight. When comparing these values, there are two important features to keep in mind. First,

Table 4.9: Fixed-step integration errors through one revolution in GTO with two-body dynamics.

	Position Error (km)	Velocity Error (km/s)	Time Error (sec)
IJK_{100}	5.6944×10^{-7}	3.0947×10^{-10}	-1.9470×10^{-6}
MEE_{100}	5.9314×10^{-12}	5.2222×10^{-15}	ϵ
MEE_{24}	1.1581×10^{-6}	1.0196×10^{-9}	7.1531×10^{-7}

Table 4.10: Fixed-step integration errors through one revolution in GTO with J_2 and lunar gravity.

	Position Error (km)	Velocity Error (km/s)	Time Error (sec)
IJK_{100}	5.7255×10^{-7}	3.1171×10^{-10}	-1.9394×10^{-6}
MEE_{100}	1.8263×10^{-11}	1.2615×10^{-14}	8.7311×10^{-11}
MEE_{24}	1.4067×10^{-6}	1.2474×10^{-9}	-9.3065×10^{-6}

IJK_{100}, MEE_{100} and MEE_{24} trajectories are computed with different accuracies. Identical controls applied to each representation produce different trajectories as was shown in Tables 4.9 and 4.10, where the controls were zero thrust. Second, the optimization path, or how the iterates evolve, should also be expected to vary with each representation. Converged trajectories might fall into different local minima. Nonetheless, it is encouraging to see the different cases reach similar final mass and time of flight results. MEE_{100} delivers slightly less mass than IJK_{100} across the board, but that is trivial as MEE_{100} was already proven orders of magnitude more accurate. What is not trivial, is the agreement between MEE_{100} and the reduced MEE_{24} results.

Table 4.12 provides the number of iterations required to reach convergence and the run-time. Comparing subroutine runtimes in Table 4.8 suggests that MEE_{100} should warrant the most computational effort to produce a solution. The number of iterations to convergence shows the significant result that MEE proves to be more favorable for this optimization procedure than IJK. MEE iterations are more successful, and fewer are required to reach convergence. Even though a single iteration proved computationally demanding, less computation time was required overall. Not surprisingly, MEE_{24} offers additional speedup, but without an obvious effect on the iteration count.

Converged values of the IJK_{100} Lagrange multipliers associated with the $\boldsymbol{\psi}_{IJK}$ constraints are

Table 4.11: GTO to GEO results for Cartesian and modified equinoctial element comparison transfers.

Perturbations	N_{rev}	IJK_{100}		MEE_{100}		MEE_{24}	
		m_f (kg)	t_f (days)	m_f (kg)	t_f (days)	m_f (kg)	t_f (days)
None	450.5	1789.21	330.17	1788.79	329.73	1788.76	330.18
J_2	450.5	1768.86	354.09	1768.64	351.59	1768.27	351.51
J_2 and Moon	450.5	1770.18	354.69	1769.92	356.25	1769.42	350.55
J_2 and Moon	1000.5	1794.15	592.40	1785.02	547.10	1783.80	554.76

Table 4.12: Computational performance for Cartesian and modified equinoctial element comparison transfers.

Perturbations	N_{rev}	IJK_{100}		MEE_{100}	
		Iterations	Runtime (min)	Iterations	Runtime (min)
None	450.5	700	233	219	116
J_2	450.5	854	265	314	218
J_2 and Moon	450.5	738	244	334	239
J_2 and Moon	1000.5	1252	906	477	746
				MEE_{24}	
None	450.5			253	32
J_2	450.5			355	60
J_2 and Moon	450.5			286	50
J_2 and Moon	1000.5			344	132

listed in Table 4.13. Lagrange multipliers for MEE_{100} and MEE_{24} are instead associated with the ψ_{MEE} constraints and are listed separately in Table 4.14. The close agreement between the MEE_{100} and MEE_{24} multipliers further justifies the utility of the reduced discretization to obtain a reliable solution quickly. While IJK and MEE multipliers have no relation, it is useful to compare the constraints ψ_{IJK} and ψ_{MEE} in Equation 4.26a. MEE offers the benefit of being able to prescribe the constraints directly as final values of state variables. This possibly contributes to the improved numerical performance exhibited with MEE, but certainly facilitates obtaining analytical derivatives of the constraints. Except for the out of plane penalties, ψ_{IJK} instead contains functions of state variables. Choosing those functions for GEO or other terminal constraints affects attainability and

numerical properties of derivatives that in turn affect the optimizer performance. Of course, *MEE* constraints are not exempt from this and may be formed as complicated functions, and a simpler *IJK* constraint could be stated exclusively as a Cartesian state rendezvous.

Table 4.13: Lagrange multipliers for IJK_{100} GTO to GEO transfers.

Perturbations	N_{rev}	λ_0	λ_1	λ_2	λ_3	λ_4
None	450.5	-1.1533	-0.2658	4.6335×10^{-5}	0.0027	-0.1385
J_2	450.5	-0.8382	-0.2066	-0.0231	-2.0782	-0.0221
J_2 and Moon	450.5	-0.8295	-0.2054	-0.0192	-2.0713	-0.0226
J_2 and Moon	1000.5	-0.9445	-0.2291	-0.0809	0.5737	0.0998

Table 4.14: Lagrange multipliers for *MEE* GTO to GEO transfers.

	Perturbations	N_{rev}	λ_0	λ_1	λ_2	λ_3	λ_4
MEE_{100}	None	450.5	0.2281	1.5540	-1.7250×10^{-4}	2.7992	-0.0101
	J_2	450.5	0.2227	1.2314	0.3355	2.3185	-3.4978
	J_2 and Moon	450.5	0.2299	1.2152	0.3397	2.2490	-3.4914
	J_2 and Moon	1000.5	1.1063	0.4016	2.7975	0.1326	-2.9303
MEE_{24}	None	450.5	0.2001	1.5271	-6.9611×10^{-4}	2.7834	-0.0121
	J_2	450.5	0.2242	1.2395	0.3219	2.3298	-3.5316
	J_2 and Moon	450.5	0.2305	1.2468	0.3257	2.2908	-3.5875
	J_2 and Moon	1000.5	1.1120	0.5883	2.7642	0.4000	-3.0477

Equatorial projections of each trajectory in Figure 4.10 add visual verification that the different state representations lead to similar transfers. Markers are placed at the initial and final state, and GEO is noted with a dotted line. The color scheme is maintained, with orange indicating thrust arcs and blue indicating coast arcs. Three-dimensional views of the MEE_{100} transfers are shown in Figure 4.11. The three-dimensional view reveals thrust arcs that are otherwise aliased in the equatorial projection. The effect is most pronounced for the 1000.5 revolution transfer. Figures 4.12 to 4.14 trace the semi-major axis, apsis radii, eccentricity and inclination through the duration of the two-body MEE_{100} transfer, and the 450.5 and 1000.5 revolution MEE_{100} transfers with J_2 and lunar perturbations.

The shorter 450.5 revolution transfers can be described by three maneuver phases for both

the two-body and perturbed cases. First, continuous thrust raises both perigee and apogee while reducing eccentricity and inclination. Apogee notably increases above GEO distance for more efficient inclination change. Second, shorter thrust arcs are centered on apogee with coast arcs spanning most of each revolution. The rate of apogee raising decreases with the loss of thrusting through perigee. Eccentricity and inclination changes also slow down, while perigee raising is accelerated. The final revolutions contain both apogee and perigee-centered maneuvers.

Four different maneuver phases characterize the MEE_{100} 1000.5 revolution transfers, with a different strategy than the 450.5 revolution transfers. Initial maneuvering is almost exclusively purposed for inclination change. An approximately 90 day coast arc constitutes the second phase. Next, both eccentricity reduction via perigee raising and inclination reduction occur. A small increment to apogee radius aids the inclination change. As before, perigee thrust arcs return in the final phase to complete the transfer. The IJK_{100} 1000.5 revolution transfer appears to fall in a transition between the two strategies, with significant apogee raising occurring in place of the 90 day coast arc.

4.5   Conclusion

The pairing of differential dynamic programming and the Sundman transformation has been presented as a viable approach to the low-thrust many-revolution spacecraft trajectory optimization problem. The utility of this method has been demonstrated by the fuel-optimization of transfers from geostationary transfer orbit to geostationary orbit with an implementation of the differential dynamic programming algorithm and transformations to the true, mean and eccentric anomalies. The resulting Pareto front of propellant mass versus time of flight is consistent with those of other methods. Beyond just reproducing a past result, the method is demonstrably efficient and amenable to perturbations. Choosing the true anomaly for this study proved most effective and enabled the direct optimization of up to 600,000 variables for the 2000 revolution transfer.

The size of the optimization problem and computation time can be significantly reduced by replacing the Cartesian representation of the spacecraft with an orbital element set. This was

demonstrated by using modified equinoctial elements. First and second derivatives of the corresponding dynamics require meticulous attention when increased fidelity is desired. That procedure has been detailed in a manner that allows arbitrary perturbations or spacecraft power models, pending the availability of necessary derivatives. Modified equinoctial elements require additional overhead in this implementation, and it was first identified that trajectory computation would be slower until the discretization was reduced to 24 stages per revolution. Significant speed improvement via parallelization was also shown.

Delivered masses, times of flight and trajectory visualizations validate the comparisons made and show the robustness of Sundman-transformed DDP to different state representations. More significantly, modified equinoctial elements proved more effective than the Cartesian state during optimization by reaching convergence in fewer iterations with improved runtime. Also significant is that nearly identical solutions were found when discretization was reduced from 100 stages per revolution to 24, so that reliable solutions may be obtained even faster.
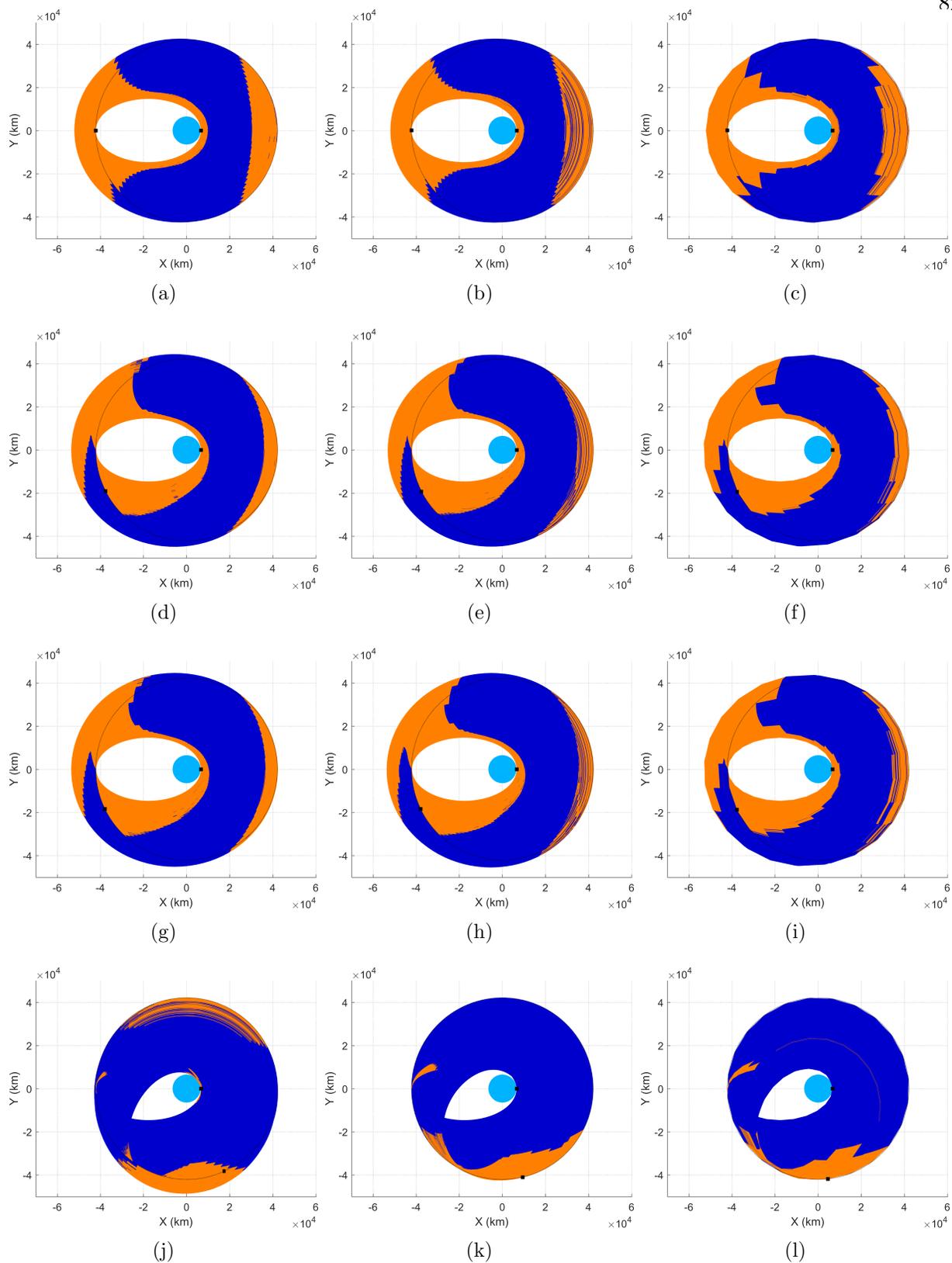
Figure 4.10: Equatorial projections of GTO to GEO transfers are organized by column (a) IJK_{100}, (b) MEE_{100}, (c) MEE_{24}, and row (a) two-body, (d) J_2, (g) J_2 and Moon, (j) 1000.5 revolutions.
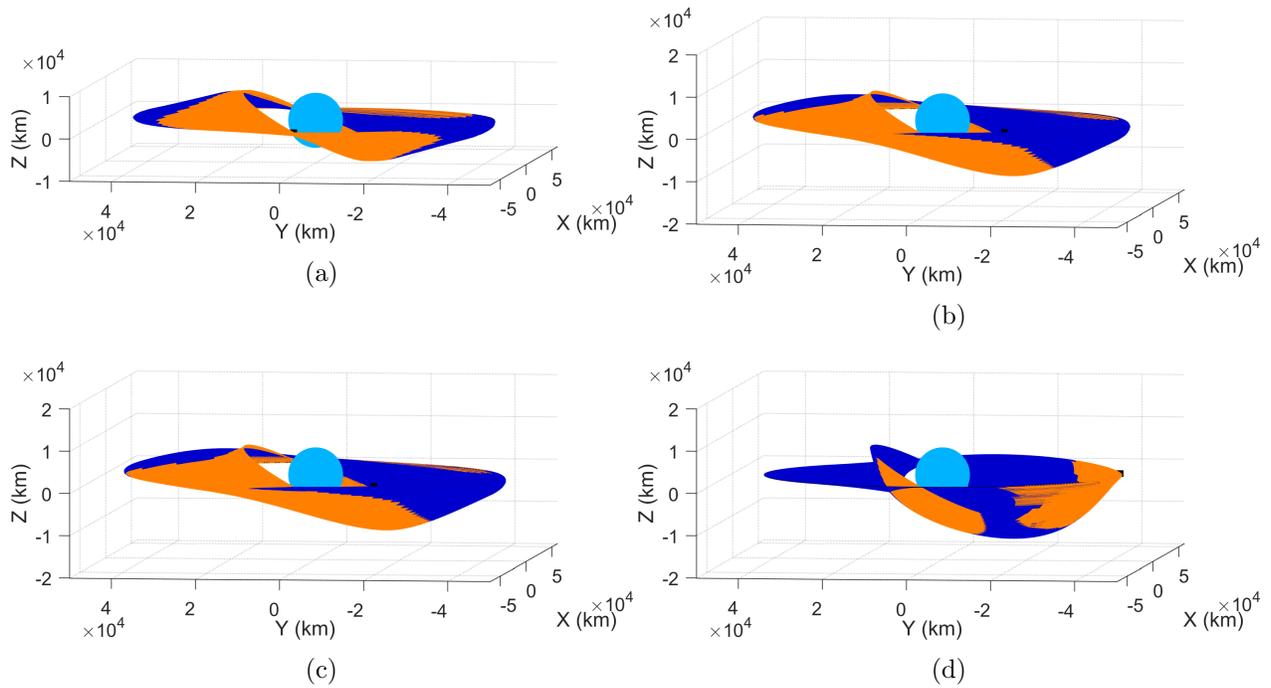
Figure 4.11: Three-dimensional views are provided for the MEE_{100} (a) two-body, (b) J_2, (c) J_2 and Moon and (d) 1000.5 revolution transfers.
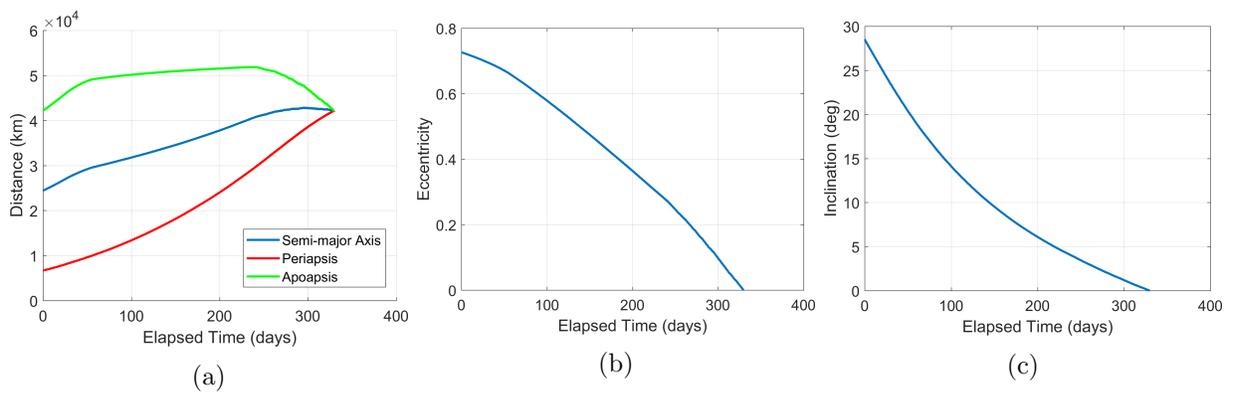
Figure 4.12: Time profiles of (a) semi-major axis, periapsis and apoapsis distance, (b) eccentricity and (c) inclination for the 450.5 revolution MEE_{100} two-body transfer.
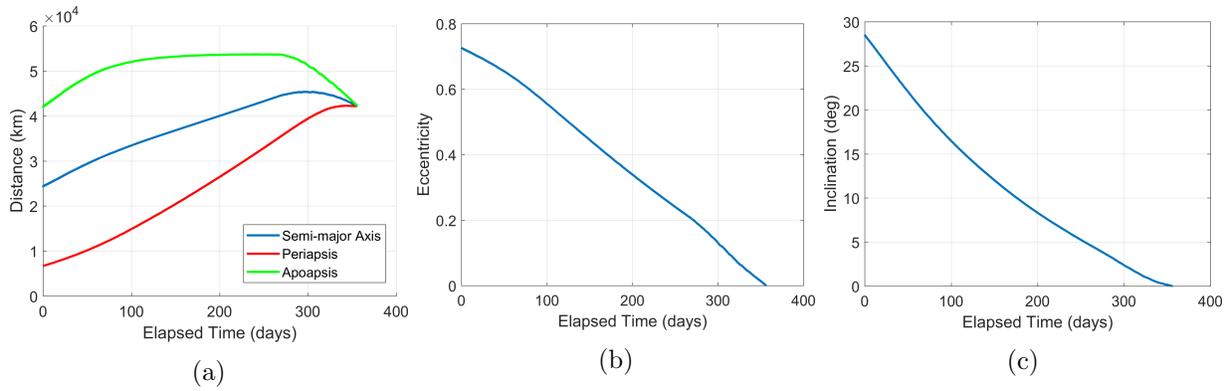
Figure 4.13: Time profiles of (a) semi-major axis, periapsis and apoapsis distance, (b) eccentricity and (c) inclination for the 450.5 revolution MEE_{100} transfer with J_2 and lunar perturbations.
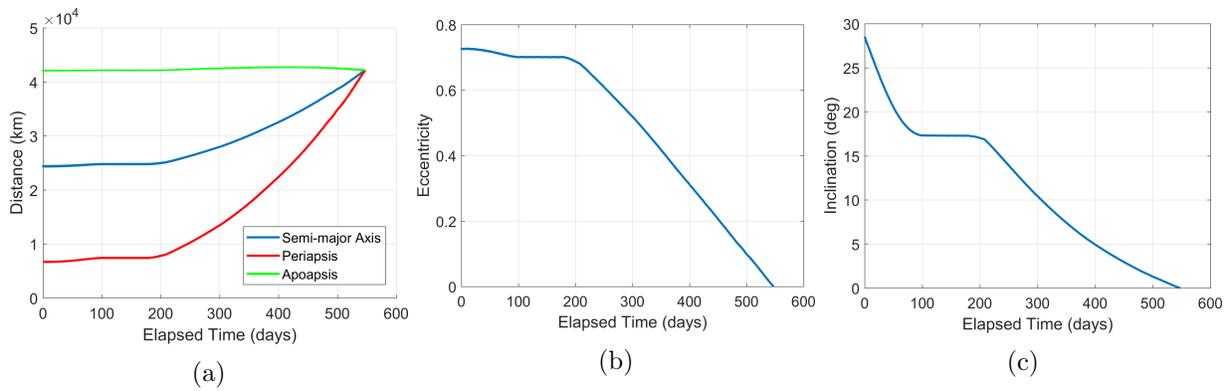
Figure 4.14: Time profiles of (a) semi-major axis, periapsis and apoapsis distance, (b) eccentricity and (c) inclination for the 1000.5 revolution MEE_{100} transfer with J_2 and lunar perturbations.

Chapter 5

A Smoothed Eclipse Model

Solar electric propulsion (SEP) is the dominant design option for employing low-thrust propulsion on a space mission. Spacecraft solar arrays power the SEP system but are subject to blackout periods during solar eclipse conditions. Discontinuity in power available to the spacecraft should be accounted for in trajectory optimization, but gradient-based methods require a differentiable power model. This chapter presents a power modeling technique that smooths the power available through eclipse entry and exit. The approach is to scale any user-supplied power model by the fraction of sunlight that is visible to the spacecraft, named the sunlight fraction γ. For the inverse-square power model in Equation 2.8, the power available is

$$P_a = \gamma \frac{AU^2}{r_{sc/\odot}^2} P_0. \tag{5.1}$$

The sunlight fraction is a discontinuous function of the possible eclipse conditions of 100% sunlight and total, partial or annular eclipse. The hassle of checking eclipse conditions and transitioning between cases is removed by considering the sunlight fraction as a Heaviside step function between 100% sunlight and 0% within the penumbral cone. The discontinuous step can then be smoothed with a logistic function. Similar approaches have seen previous success when applied to the discontinuity in discrete number of thrusters available [80] and discrete throttle settings [63].

5.1 Conical Eclipse Geometry

By assuming spherical shapes of the Sun and occulting body and neglecting atmospheric effects, the eclipse geometry can be represented with overlapping circular discs [81]. Figure 5.1

sketches the eclipse geometry when the edges of the solar disc and occulting planetary disc are perceived to be coincident by a spacecraft. The true measure of the solar radius is $R_\odot$ and that of
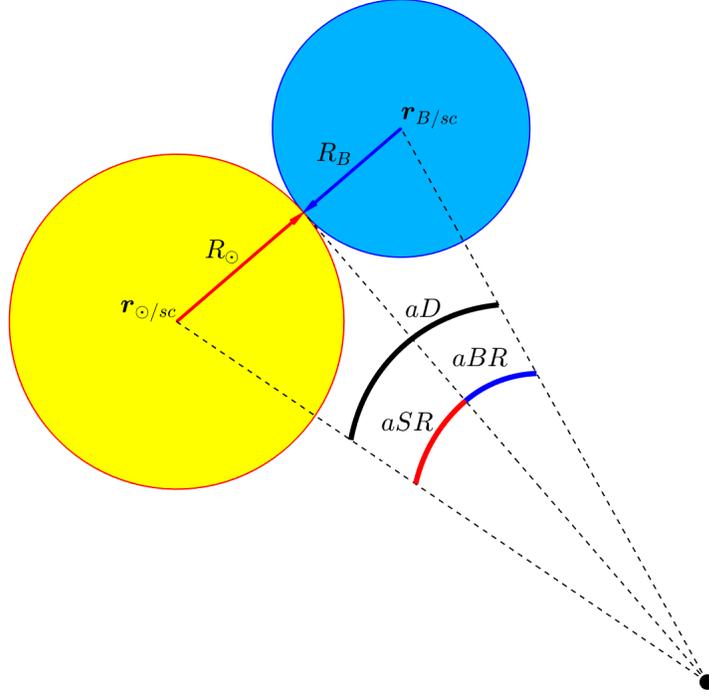
Figure 5.1: Solar and planetary discs as viewed by a spacecraft on the threshold of an eclipse event. Geometric features relevant to the eclipse model are labeled.

the occulting body is R_B. Relative position vectors are defined so that

$$\boldsymbol{r}_{\odot/sc} = \boldsymbol{r}_\odot - \boldsymbol{r}_{sc} \tag{5.2}$$

is the position of the Sun with respect to the spacecraft and

$$\boldsymbol{r}_{B/sc} = \boldsymbol{r}_B - \boldsymbol{r}_{sc} \tag{5.3}$$

is the position of the occulting body with respect to the spacecraft. The apparent Solar radius aSR and apparent body radius aBR are the angles subtended by their respective radii.

$$\sin(aSR) = \frac{R_\odot}{\|\boldsymbol{r}_{\odot/sc}\|} \tag{5.4}$$

$$\sin(aBR) = \frac{R_B}{\|\boldsymbol{r}_{B/sc}\|} \tag{5.5}$$

The astute reader will notice that Equations 5.4 and 5.5 are in slight error from the correct geometric

relations are $\sin(aSR/2) = R_\odot/(2r_{\odot/sc})$ and $\sin(aBR/2) = R_B/(2r_{B/sc})$ since the tangent points

from the spacecraft to either body are separated by less than the respective body's diameter. The

apparent distance aD is the angle between the two bodies as viewed by the spacecraft.

$$\cos(aD) = \frac{\boldsymbol{r}_{B/sc}^T \boldsymbol{r}_{\odot/sc}}{\|\boldsymbol{r}_{B/sc}\|\|\boldsymbol{r}_{\odot/sc}\|} \tag{5.6}$$

By inspection of Figure 5.1, an eclipse occurs when the sum of the apparent radii exceeds the

apparent distance between the two bodies,

$$aSR + aBR > aD \ . \tag{5.7}$$

When the inequality holds and $\boldsymbol{r}_{B/sc} < \boldsymbol{r}_{\odot/sc}$, additional criteria determine which part of the

occulting body's shadow the spacecraft is positioned in.

- Umbra: $aD \leq aBR - aSR$ and $aSR \leq aBR$

- Penumbra: $aD > |aBR - aSR|$

- Antumbra: $aD \leq aBR - aSR$ and $aSR > aBR$

The eclipse is total in umbra, partial in penumbra and annular in antumbra.

5.2   Smoothed Sunlight Fraction

The sunlight fraction is a discontinuous function of the geometry of the spacecraft, Sun and

occulting body positions.

$$\gamma = \begin{cases} 0, & aD \leq aBR - aSR, \ aSR \leq aBR \text{ and } aSR + aBR > aD \\ (0,1), & aD > |aBR - aSR| \text{ and } aSR + aBR > aD \\ 1, & aSR + aBR \leq aD \end{cases} \tag{5.8}$$

The intermediate case is itself a discontinuous function that depends on if the spacecraft is in

penumbra or antumbra. Additional relations from the overlapping disc geometry allow for the

computation of the sunlight fraction in each of these cases [81]. Those are disregarded in the smoothed eclipse model by imposing a zero-thrust constraint for both partial and total eclipse. The sunlight fraction is then a step function between one and zero and relies only on Equations 5.4 to 5.6 for the necessary geometry. With the Heaviside definition of a step function, the sunlight fraction is half-valued at the eclipse transition.

$$
H(x) = \begin{cases} 0, & x < x^* \\ 0.5, & x = x^* \\ 1, & x > x^* \end{cases} \tag{5.9}
$$

$$
\gamma_H = \begin{cases} 0, & aSR + aBR > aD \\ 0.5, & aSR + aBR = aD \\ 1, & aSR + aBR < aD \end{cases} \tag{5.10}
$$

Equation 5.9 is the unit Heaviside step function that transitions at x^* and is applied to the eclipse transition to develop the Heaviside sunlight fraction in Equation 5.10. The Heaviside step function alone does not resolve the issue of a discontinuous power model, but it can be approximated by a logistic function with smooth derivatives that are favorable for gradient-based optimization.

$$
L(x) = \frac{1}{1 + \exp\left[-c_s\left(x - x^*\right)\right]} \tag{5.11}
$$

$$
\gamma_L = \frac{1}{1 + \exp\left\{-c_s\left[aD - c_t\left(aSR + aBR\right)\right]\right\}} \tag{5.12}
$$

Equation 5.11 is the unit logistic function that is a smooth approximation to the unit Heaviside step function. Equation 5.12 is the logistic sunlight fraction and constitutes the smoothed eclipse model.

Figures 5.2a and 5.2b depict the Heaviside sunlight fraction and the logistic sunlight fraction for different (c_s, c_t) values. Sharpness coefficient c_s determines the slope of the curve at the transition point. Transition coefficient c_t scales the total angle of the apparent solar and body radii, and effectively moves the transition point. Selection of a (c_s, c_t) pair might be driven by numerical
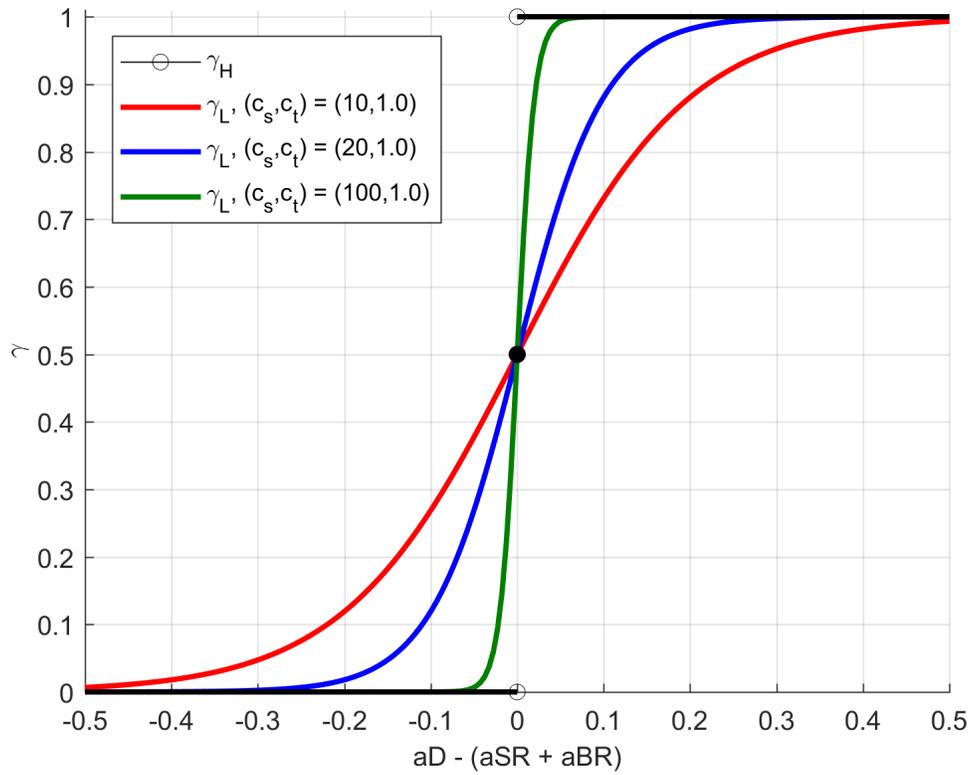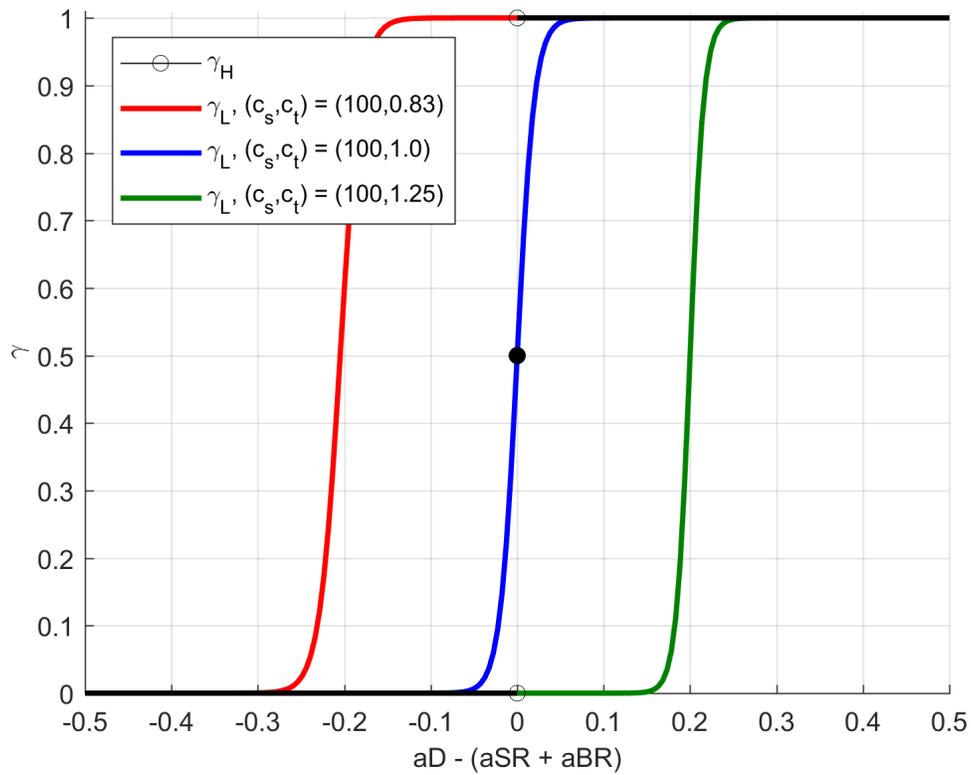
Figure 5.2: The sunlight fraction is represented by a Heaviside step function and logistic functions of (a) different sharpness coefficients and (b) different transition coefficients.

behavior, desired model accuracy or mission operations. An intermediate c_s value that is not too steep or flat is likely to benefit optimizer performance, but at the expense of model accuracy. When model accuracy is a priority, the (c_s, c_t) pair that minimizes the smoothed model error could be selected. In lieu of constant values, (c_s, c_t) could be a function of time or orbital radius, for example, to reflect the changing eclipse geometry during a transfer. From a mission operations perspective, it might be necessary to power down/up the spacecraft at a certain rate and offset from eclipse entry/exit. That rate and offset are prescribed by (c_s, c_t) in the smoothed eclipse model.

5.3   Trajectory Optimization Example

Low-thrust transfers from LEO to GEO characteristically encounter frequent solar eclipses. That rate is expectedly once per revolution unless the eclipse is avoided by favorable geometry. To demonstrate the utility of the smoothed eclipse model, the minimum-fuel LEO to GEO transfer from Reference [34] was reproduced with DDP as the optimization method.

5.3.1   Constant power with smoothed eclipsing

Power available to the spacecraft is modeled by scaling a reference power level by the logistic sunlight fraction.

$$P_a = \gamma_L P_0 \tag{5.13}$$

Higher-fidelity modeling is possible by replacing P_0 with improved power models as described in Section 2.1.1. It is assumed that the spacecraft has gimbaled solar arrays and/or gimbaled thrusters so that the arrays are always wholly effective. Thrust available from the SEP system is then given by Equation 2.7. Table 5.2 lists the necessary spacecraft parameters. Reference [34] considers $T_a = T_{max}$ for burn phases, where the maximum thrust is a constant value that is always attainable. Thus, power modeling can be reduced to a scaling of T_{max} by the logistic sunlight fraction.

$$T_a = \gamma_L T_{max} \tag{5.14}$$

Selection of a (c_s, c_t) pair began by choosing $c_t = 1.0$ so that there is no shift in the eclipse transition. Next, $c_s = 289.78$ was computed to minimize the error between the logistic sunlight fraction and discontinuous eclipse model [81] for a 300 km altitude circular LEO. That value maintained accuracy as the orbital altitude was increased. The Sun and body radii and Sun-body distance drive the optimal c_s value, rather than the spacecraft's planetocentric orbit. The values $(c_s, c_t) = (289.78, 1.0)$ were held fixed for all numerical examples.

5.3.2   Boundary conditions and dynamics

Initial conditions for a 500 km altitude LEO with $28.5°$ inclination are stated in terms of the modified equinoctial elements in Table 5.1. Initial conditions for the target GEO are also listed with the exception of the unconstrained true longitude. Dynamics include solar and lunar third-body perturbations and perturbations from Earth zonal harmonics J_2, J_3 and J_4. Dynamic model parameters are summarized in Table 5.2. The Sun's position, velocity and acceleration are obtained from heliocentric Keplerian motion of the Earth beginning with the osculating orbital elements for the Earth at the reference epoch. The Moon's state is similarly obtained by geocentric Keplerian motion.

Table 5.1: Initial and target orbits in terms of the modified equinoctial elements for the LEO to GEO transfer with eclipsing.

p_0	6878.14 km	p_{target}	42241.095482 km
f_0	0.0	f_{target}	0.0
g_0	0.0	g_{target}	0.0
h_0	-0.253967	h_{target}	0.0
k_0	0.0	k_{target}	0.0
L_0	π		

Reference [34] selected the true longitude as the independent variable and arrived at an optimal final mass of $m_f = 718.79$ kg in $\Delta L = 248$ revolutions and time of flight $t_f = 43.13$ days. For this example, the true anomaly was chosen as the independent variable and the endpoint was fixed at $\Delta \nu = 248$ revolutions. The DDP setup also included the augmented modified equinoctial

Table 5.2: Dynamic model parameters for the LEO to GEO transfer with eclipsing.

$\mu_\oplus$	398600.436380 km^3/s^2	J_2	1.082639×10^{-3}
$\mu_\odot$	132712440018 km^3/s^2	J_3	-2.565×10^{-6}
$\mu_\mathstrut$	4902.798815 km^3/s^2	J_4	-1.608×10^{-6}
$R_\oplus$	6378.14 km	g_0	9.80665×10^{-3} km/s^2
$R_\odot$	695500 km	(c_s, c_t)	$(289.78, 1.0)$
m_0	1000 kg	T_{max}	1.445 N
I_{sp}	1849.347748 sec		
Reference Epoch	2457377.5 Julian Date TDB		

element state vector and discretization set to 60 stages per revolution (MEE_60) evenly spaced in true anomaly. An initial guess of $T = 0$ for all stages set DDP iteration to begin from a ballistic trajectory through 248 revolutions in LEO. RK8 integration was used as the numerical procedure.

5.3.3   Initial results

First, the LEO to GEO trajectory was computed without eclipsing to provide a reference solution and quantify the effects of eclipsing on trajectory performance. The 53.32 day transfer delivers 744.04 kg and is shown in Figure 5.3. The trajectory computed by DDP with the smoothed eclipse model active is shown in Figure 5.4a and the eclipse passages are highlighted in Figure 5.4b. When compared to the soltuion in Reference [34], an improved final mass of 723.29 kg comes at the expense of an increase in time of flight to 44.72 days. True longitude accumulates to 248.48 revolutions.

Eclipse passages interfere with initial thrusting to raise the orbital radius from LEO. Coast arcs in sunlight are driven by the minimum-propellant objective and do not occur until late in the transfer. The significantly shorter flight time with eclipse effects can be attributed to the eclipses occurring at low orbital radii combined with the fixed number of revolutions. More revolutions are required to escape eclipse season, but these are passing at short orbital periods. Once eclipses cease to occur, fewer revolutions of increased orbital period remain for completing the transfer. Without eclipsing, the increased number of higher radius revolutions allows the final maneuvers to be more
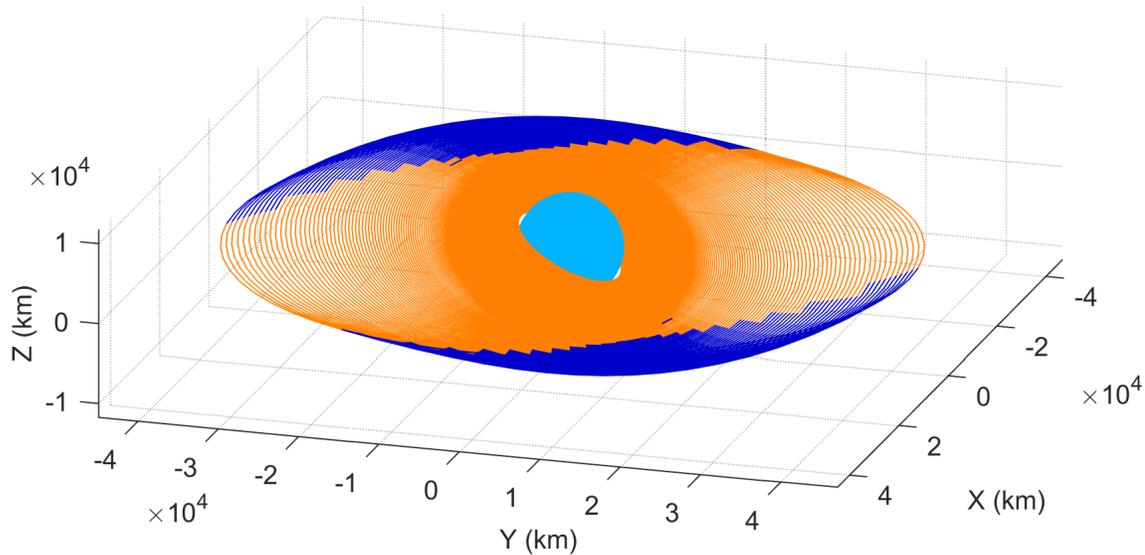
Figure 5.3: The 248 revolution LEO to GEO transfer without eclipse modeling.

closely centered on their optimal locations to improve the delivered mass.

The initial DDP solution and Betts' solution in Reference [34] do not provide a one-to-one comparison, owed to the different flight times. For the DDP solution, delivered mass expectedly increased alongside the time of flight. To better understand the trade-off of propellant mass versus time of flight, optimization was repeated for a number of transfers fixed in duration to a number of revolutions. Results are shown in Figure 5.5, which shows the expected behavior of fuel requirement decreasing with transfer duration. There is a noticeable penalty for transfers completing in one quarter or three quarters of a revolution, thereby losing symmetric maneuvering for inclination change about the line of nodes. The more desirable comparison, however, is propellant mass versus time, rather than revolutions, which is shown in Figure 5.6. The trade study failed to lower the time of flight to that of Reference [34]. The shortest transfer of 44.69 days results from the 249 revolution transfer. Continuing to decrease the number of revolutions only results in increased times of flight. A better initial guess such as that from a control law, rather than a ballistic orbit in LEO, could help extend the Pareto-front into the lower flight time regime. In an attempt to match Reference [34], the 248 revolution solution was used as an initial guess for optimization restarted
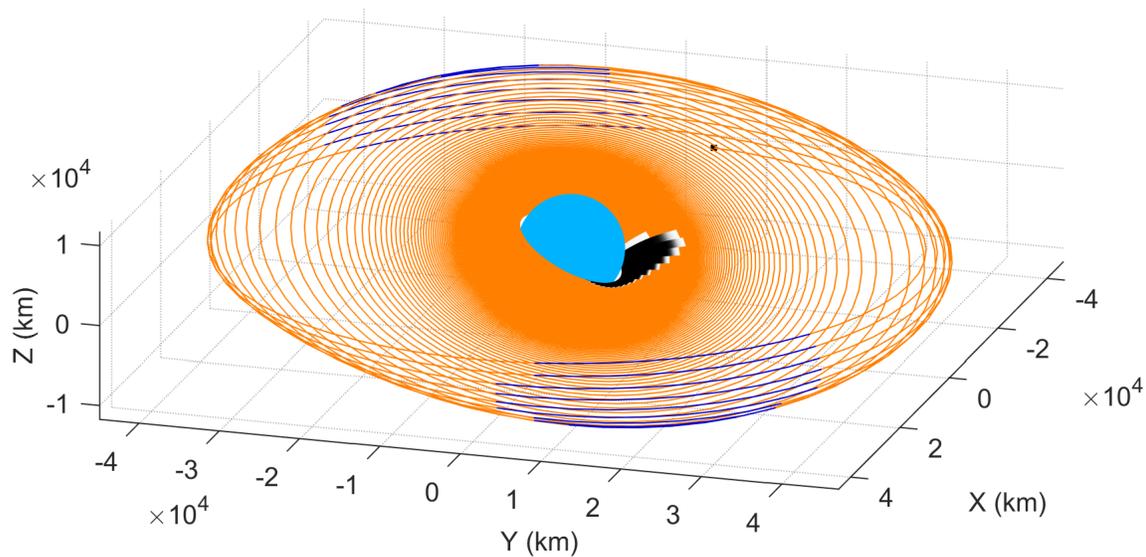
(a)
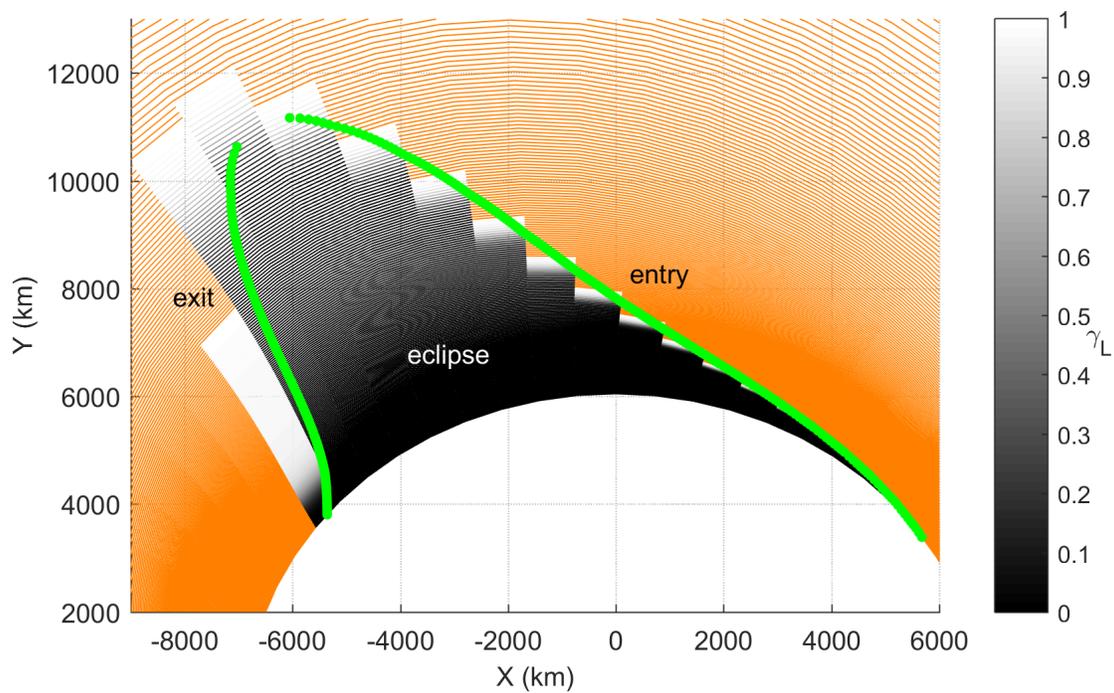
(b)

Figure 5.4: (a) The 248 revolution LEO to GEO trajectory computed with the smoothed eclipse model active is shown in a three-dimensional view. Eclipse arcs are colored in grayscale from black for total eclipse to white for total sunlight. (b) An equatorial projection of the eclipse passages is shown with penumbra entry and exit locations marked.
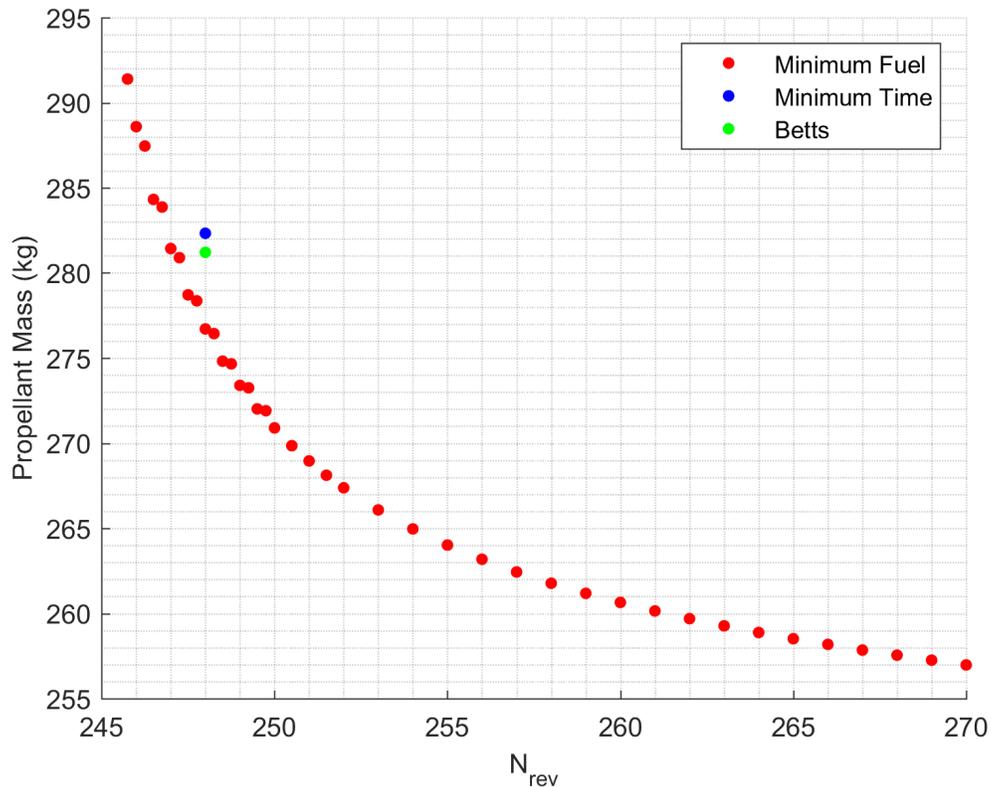
Figure 5.5: Trade-off of fuel versus number of revolutions for the LEO to GEO transfer with eclipsing.

to minimize time instead of fuel. The awkward prescription to minimize time in a fixed number of revolutions successfully approaches the desired mass and time results, with $m_f = 717.76$ kg and $t_f = 43.14$ days. It is plausible that the initial guess from a Lyapunov based control law in Reference [34] approached the minimum-time solution and subsequent optimization passes for fuel optimization found a local minimum only nearby the initial point. Both Betts' solution and the minimum-time DDP solution are marked in each of Figures 5.5 and 5.6.

Intuitively, there should be trajectories to fill the gap between the minimum-time solution and the curve of minimum-fuel solutions. The first attempt at finding these trajectories followed the same approach as the time-minimization of the 248 revolution transfer, by taking the minimum-fuel solutions as initial guesses for restarts of DDP with time of flight as the new objective to minimize. The new curve of minimum-time DDP solutions is shown in Figure 5.7. As the number of revolutions
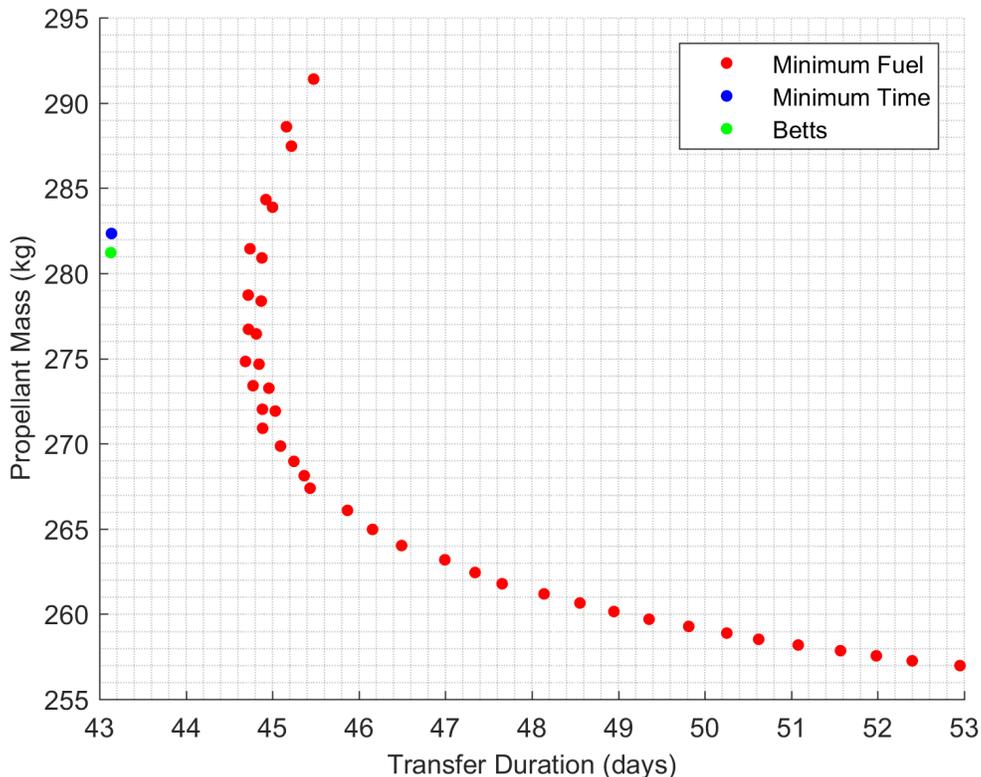
Figure 5.6: Trade-off of fuel versus time of flight for the LEO to GEO transfer with eclipsing.

is decreased from the reference 248 revolution solution, both the propellant requirements and times of flight increase to complete the transfer. The lower number of revolutions is insufficient to perform the inclination change. Additional orbit raising becomes necessary to accomplish more inclination change within a given revolution but with the penalty of added flight time. On the other end, increasing the number of revolutions produces trajectories with both cheaper fuel requirements and shorter times of flight. This trend proceeds down to the lowest time of flight of 41.92 days over 259 revolutions, after which time of flight turns to increasing with the number of revolutions.

In order to find trajectories in the gap between the minimum-fuel and minimum-time curves, a target final time t_{target} was added to the terminal constraint function. Then the value of t_{target} was varied between the minimum time and the flight time of the minimum-fuel solution for a specified number of revolutions. DDP was restarted to minimize fuel at each value of t_{target}, beginning with
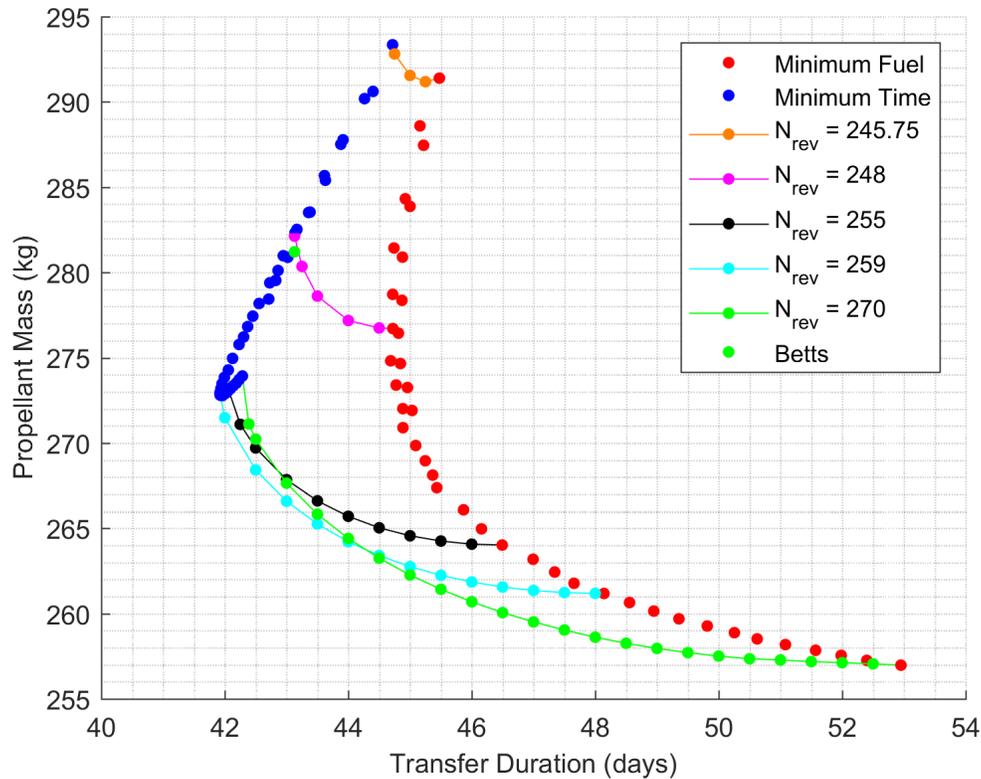
Figure 5.7: Trade-off of fuel versus time of flight for varied objectives for the LEO to GEO transfer with eclipsing.

the minimum-time solution as an initial guess and passing each solution forward as an initial guess for the next t_{target}. These solution curves are included in Figure 5.7. An approximate Pareto front appears with just the two curves of 259 and 270 revolutions. The 259 revolution curve was selected since it begins at the overall minimum-time solution. Curves of fewer revolutions are dominated, e.g. the 255 revolution curve does not intersect the 259 revolution curve to contribute to the Pareto front. After the fold (N_{rev} increasing) in the minimum-time curve, however, solution curves do intersect, so that the fuel-optimal number of revolutions varies with the time of flight. This is promising, so that there is not a fixed 'magic number' of revolutions. Curves between 259 and 270 revolutions necessarily have to intersect the 259 revolution curve and curves with a higher number of revolutions, at least up to 270. If there is a limit, it remains unknown without further computation.

Propellant mass and time of flight for all transfers are shown versus number of revolutions

in Figures 5.8 and 5.9. The fold in the minimum-time curve of Figure 5.7 now appears as leveling off and holding fairly constant as the number of revolutions changes around the overall minimum-time. Curves of constant number of revolutions are more obvious as they appear vertically between the minimum-fuel and minimum-time results. The trajectory spacing is uniform in Figure 5.9 (except for a few choice trajectories) because the times were selected. The spacing is not uniform in Figure 5.8 as it becomes more demanding to approach the minimum time of flight.
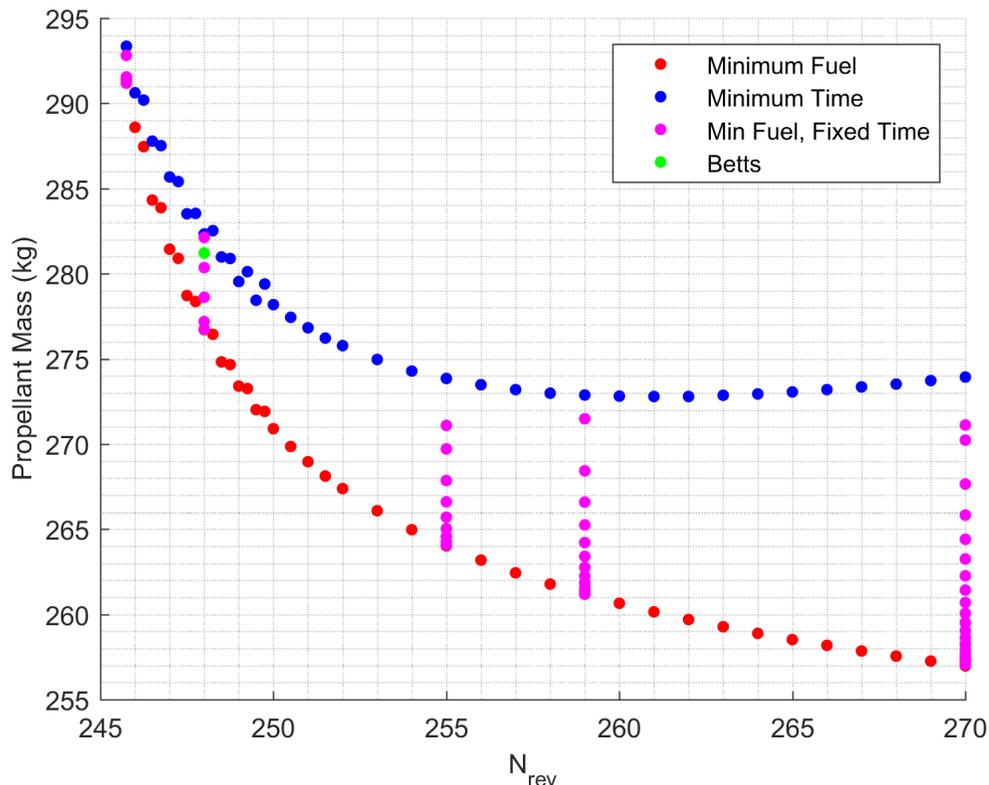
Figure 5.8: Trade-off of fuel versus number of revolutions for varied objectives for the LEO to GEO transfer with eclipsing.

5.4   Penumbra Terminator Detection

While the smoothed eclipse model benefits from an automatic response in power available during eclipse without the need for event detection, a fixed mesh of integration stages will step over the eclipse entry and exit points. The spacecraft effectively sees prolonged sunlight into pe-numbra entry and prolonged eclipse through penumbra exit. The discretization error is illustrated
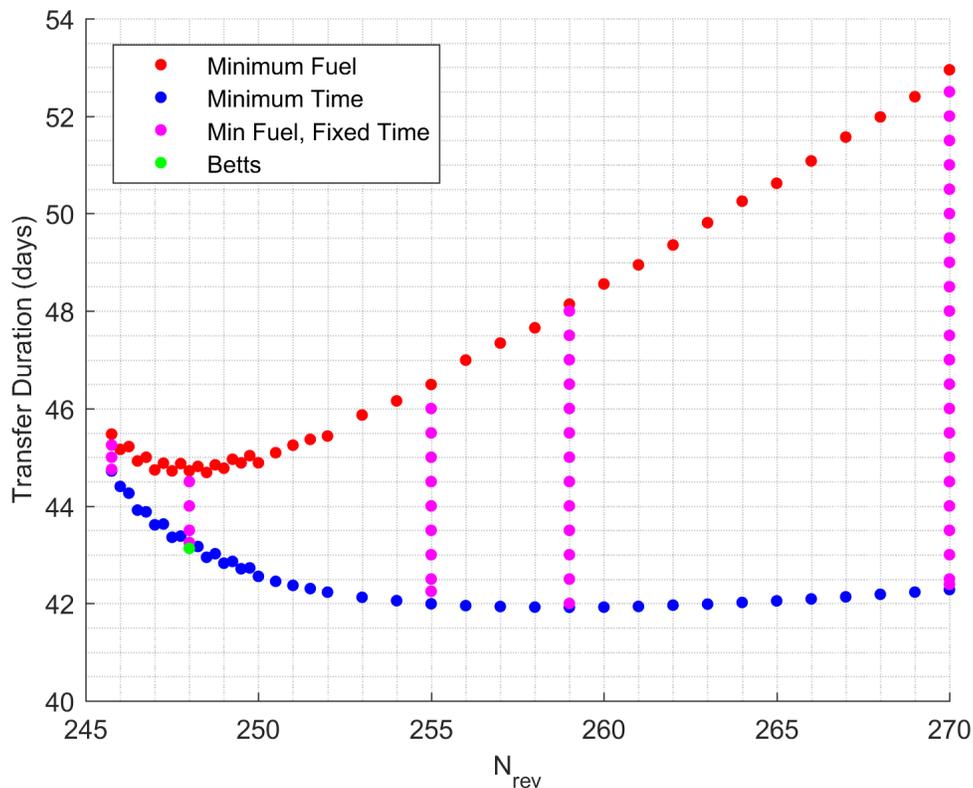
Figure 5.9: Trade-off of time of flight versus number of revolutions for varied objectives for the LEO to GEO transfer with eclipsing.

in Figure 5.4b, where penumbra entry and exit locations have been marked on an equatorial projection of the initial trajectory result. There are several ways to fix this error. One is to use a variable-step integrator, but fixed-step integrators are favorable for speed and optimizer performance. Reference [73] discusses the main pitfalls of variable-step integration. Another way to correct the discretization error is to compensate analytically, which is discussed in the following sections.

5.4.1   Calculation method for penumbra entry and exit

Penumbra entry and exit locations can be estimated as the intersection of the line between two integration stages and the penumbral cone. Calculations in this study do not correct for light time but capture the instantaneous penumbral cone geometry that is illustrated in Figure 5.10. Reference [82] provides a presentation of how to find the intersection of a line and a cone that is
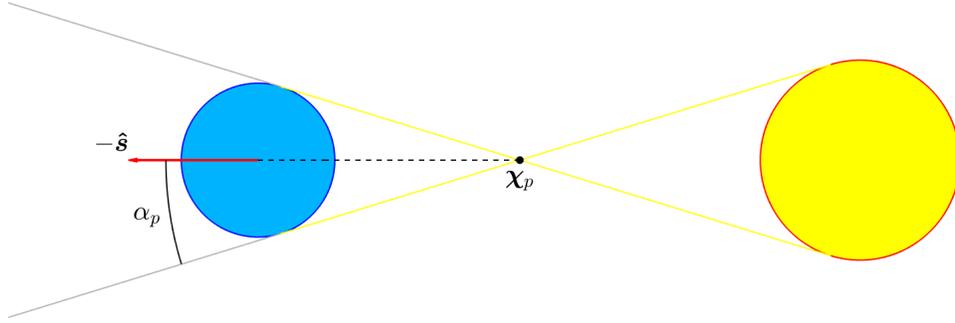
Figure 5.10: Geometry of the penumbral cone.

leveraged here with Sun-body-spacecraft relations.

The origin of the penumbral cone is a function of the solar and occulting body radii and the solar vector.

$$\boldsymbol{\mathcal{X}}_p = \frac{R_B}{R_B + R_\odot} \boldsymbol{s} \tag{5.15}$$

Note that $\boldsymbol{s} = \boldsymbol{r}_{\odot/B}$. The penumbral cone axis is represented by a unit vector opposite the solar vector $-\hat{\boldsymbol{s}}$. The penumbral cone angle is

$$\sin \alpha_p = \frac{R_B}{\|\boldsymbol{\mathcal{X}}_p\|}, \tag{5.16}$$

which is again a minor simplification from $\sin(\alpha_p/2) = R_B/(2\mathcal{X}_p)$. A spacecraft with position vector $\boldsymbol{r}$ is within the penumbral cone if the vector difference $\boldsymbol{r} - \boldsymbol{\mathcal{X}}_p$ forms an angle with the cone axis that is smaller than the cone angle.

$$-(\boldsymbol{r} - \boldsymbol{\mathcal{X}}_p)^T \hat{\boldsymbol{s}} \geq \|(\boldsymbol{r} - \boldsymbol{\mathcal{X}}_p)\| \cos \alpha_p \tag{5.17}$$

Squaring Equation 5.17 yields a quadratic inequality with roots along the penumbra terminator.

$$(\boldsymbol{r} - \boldsymbol{\mathcal{X}}_p)^T (\hat{\boldsymbol{s}}\hat{\boldsymbol{s}}^T - I \cos^2 \alpha_p)(\boldsymbol{r} - \boldsymbol{\mathcal{X}}_p) \geq 0 \tag{5.18}$$

Consider the position vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ just before and after penumbra entry. Inserting the line $\boldsymbol{r}(t) = \boldsymbol{r}_1 + t\boldsymbol{r}_2$ into Equation 5.18 and solving for the roots of $at^2 + 2bt + c = 0$ provides the points of intersection of $\boldsymbol{r}(t)$ and the penumbral cone. After defining $M = (\hat{\boldsymbol{s}}\hat{\boldsymbol{s}}^T - I \cos^2 \alpha_p)$, the

coefficients are

$$a = \boldsymbol{r_2}^T M \boldsymbol{r_2} \tag{5.19a}$$

$$b = \boldsymbol{r_2}^T M (\boldsymbol{r_1} - \boldsymbol{X}_p) \tag{5.19b}$$

$$c = (\boldsymbol{r_1} - \boldsymbol{X}_p)^T M (\boldsymbol{r_1} - \boldsymbol{X}_p). \tag{5.19c}$$

The smaller root $t = (-b \pm \sqrt{b^2 - ac}\,)/a$ estimates the spacecraft penumbra entry, while the larger root extends the line to the far side of the penumbral cone. For position vectors $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ just before and after penumbra exit, a positive root estimates the spacecraft penumbra exit and a negative root is the far side intersection.

5.4.2   Mesh adjustment for penumbra entry and exit

The method for calculation of penumbra entry and exit points can be leveraged for mesh adjustment to place integration stages on the penumbra terminator. First, a generic integration step relies on the current state and the step size.

$$\boldsymbol{X}_2 = \boldsymbol{f}(\boldsymbol{X}_1, \tau_2 - \tau_1) \tag{5.20}$$

State vector $\boldsymbol{X}$ includes the position vector $\boldsymbol{r}$ or orbital elements that determine $\boldsymbol{r}$, while $\boldsymbol{f}$ is a transition function, e.g., the integral of the equations of motion, and τ is the independent variable of integration.

Given $\boldsymbol{r_1}$ and having computed $\boldsymbol{r_2}$, the two positions can be checked for a change in eclipse conditions. The step size $\tau_2 - \tau_1$ should be selected so that the penumbra will not be stepped over entirely. If eclipse entry or exit is detected, the estimated intersection of the trajectory and penumbra terminator is $\boldsymbol{r_e} = \boldsymbol{r}(t_e)$ where t_e is selected as the appropriate root obtained by Equations 5.18 and 5.19. The angle between $\boldsymbol{r_1}$ and $\boldsymbol{r_e}$ is the approximate change in true anomaly to reach the penumbra terminator.

$$\cos(\Delta\nu) = \frac{\boldsymbol{r_e}^T \boldsymbol{r_1}}{\|\boldsymbol{r_e}\|\|\boldsymbol{r_1}\|} \tag{5.21}$$

Then the independent variable at the terminator $\tau_e = \tau(\boldsymbol{X}_1, \tau_1, \Delta\nu)$ can be determined, and two consecutive steps replace Equation 5.20.

$$\boldsymbol{X}_e = \boldsymbol{f}(\boldsymbol{X}_1, \tau_e - \tau_1) \tag{5.22a}$$

$$\boldsymbol{X}_2 = \boldsymbol{f}(\boldsymbol{X}_e, \tau_2 - \tau_e) \tag{5.22b}$$

As an approximation, $\boldsymbol{r}_e$ targets the penumbra terminator but is expected to err on either side. Stepping too far provides a conservative model of the power available, so it may be worthwhile to further increment $\Delta\nu$. This is perhaps a more realistic representation of a spacecraft shutting down early and remaining down for some duration after an eclipse event. In the worst case, a single-step integrator carries the incorrect power available from $\boldsymbol{r}_e$ to $\boldsymbol{r}_2$. Multi-step methods, however, should have integration sub-steps on the correct side of the penumbra terminator to offset this error.

5.5   Results with Penumbra Detection

Accuracy of the initial 248 revolution result was improved by inserting stages at the penumbra entry and exit locations detected by the methods of Section 5.4. DDP was restarted with the previous solution as an initial guess. Power available is again given by the smoothed eclipse model but stages for eclipse passages are forced to coast by setting $T_{max} = 0$. For each revolution with an eclipse passage, the new throttle constraint is assigned for the inserted penumbra entry stage and subsequent stages up to but not including the inserted penumbra exit. While the improved trajectory is similar, its penumbra entry and exit locations correspond to the previous solution. Subsequent iterations insert stages to update the penumbra entry and exit locations to further improve solution accuracy.   The new trajectory is practically indistinguishable aside from the improved resolution at the penumbra terminator. Color contours of the logistic sunlight fraction are not used in Figure 5.12a (as they were in Figure 5.4a). As the smoothed model is active, stages adjacent to the penumbra see γ_L less than but approximately 100%. Mass and time of flight results are summarized in Table 5.3. Iteration 1 is the first attempt without penumbra detection.

Table 5.3: Final mass and time of flight for the original DDP solution and each DDP solution after updates to the penumbra entry and exit locations

Iteration	m_f (kg)	t_f (days)
1	723.29	44.72
2	720.93	44.75
3	720.07	44.72
4	719.30	44.83
5	719.12	44.84

Subsequent iterations have updated the penumbra entry and exit locations, and require a number of DDP iterations to reconverge. Computational performance is summarized in Table 5.4. The computational resources were identical to those used in Chapter 4.

Thrust steering profiles are depicted in Figure 5.11. The in-plane thrust angle is measured from the transverse direction about the angular momentum, and the out-of-plane thrust is measured from the orbit plane about the radial direction. The thrust magnitude profile confirms the expected bang-bang control, though initial coast arcs are owed to eclipsing, while later coast arcs are for fuel optimality. Steering angles indicate how initial steering is devoted to raising the orbital radius. The out-of-plane component increases throughout the transfer as it becomes more fuel efficient to change the inclination at a larger orbital radius.

Table 5.4: The number of iterations and runtime for the original DDP solution and each DDP solution after updates to the penumbra entry and exit locations.

Iteration	DDP Iterations	Runtime (sec)
1	67	1078
2	20	259
3	17	224
4	13	299
5	2	46
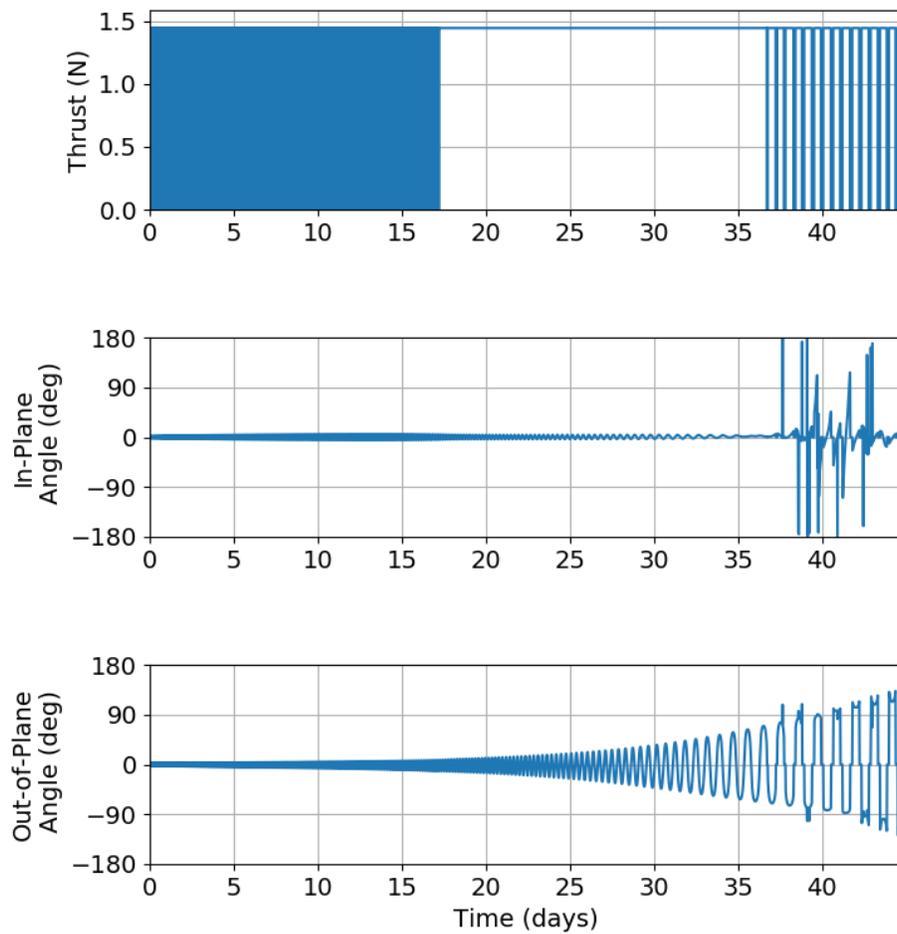
Figure 5.11: Thrust magnitude, in-plane, and out-of-plane thrust angles are shown for the duration of the 248 revolution LEO to GEO transfer after final updates to the penumbra entry and exit locations.

5.6   Conclusion

A smoothed eclipse model has been presented for use in computing optimal trajectories for space missions that employ solar electric propulsion. The logistic sunlight fraction has been introduced as the visible sunlight through the smoothed eclipse transition from total eclipse to total sunlight (and vice versa) by a logistic function. The model is continuously differentiable and suitable for gradient-based optimization methods. As a percentage of the sunlight available, the smoothed

(a)

(b)
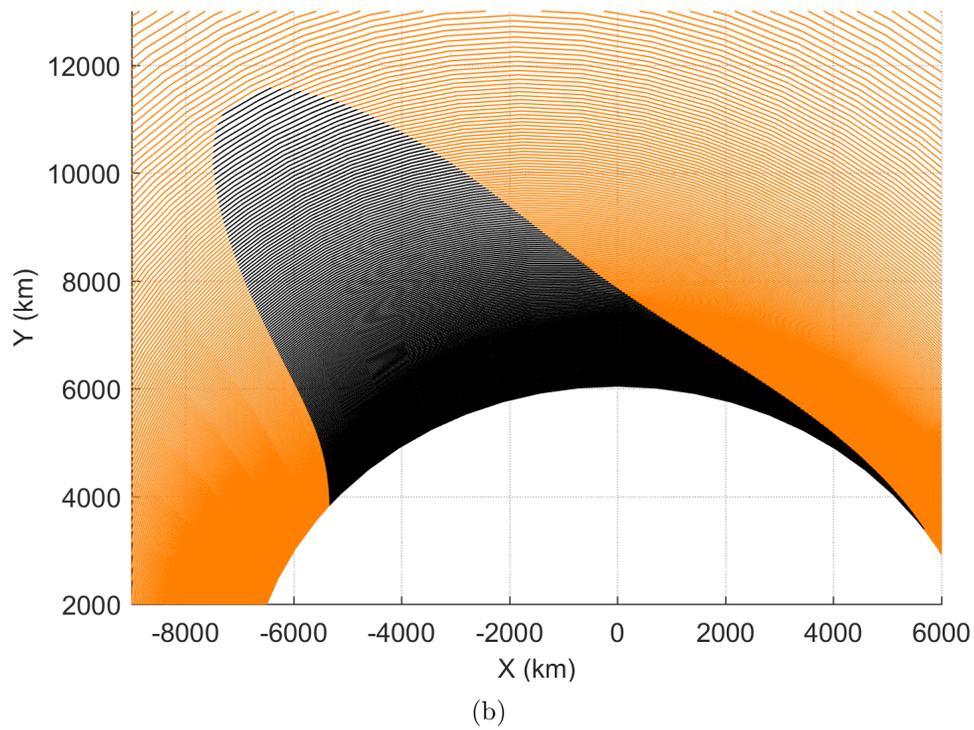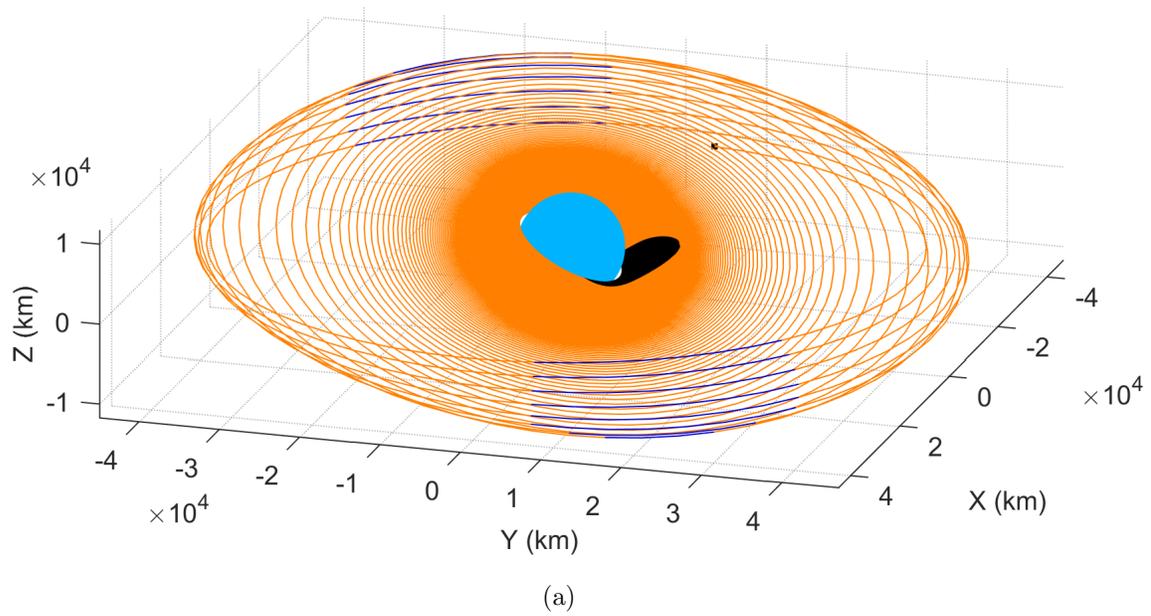
Figure 5.12: (a) The final LEO to GEO transfer with penumbra detection is shown in a three-dimensional view. (b) An equatorial projection of the eclipse passages along the final solution.

eclipse model is a coefficient that can be paired with arbitrary power models. A set of tuning parameters allows for mission designers to adjust the rate at which power available to the spacecraft declines or recovers, and to shift the location of the eclipse discontinuity. An example trajectory computed with DDP and the smoothed eclipse model improves upon the mass delivered for a LEO to GEO trajectory in [34] but with increased flight time. Additional trajectory computations present the trade-off of propellant mass versus transfer duration. That exercise showed capability of the Sundman-transformed DDP approach to compute minimum-time trajectories and do target a specific time of flight. It also showed how initial solutions can be passed forward as initial guesses to fill the solution space. Finally, a refined solution was computed to correct for discretization errors by inserting penumbra entry and exit locations with only a minor adjustment to the delivered mass and time of flight.

Chapter 6

A Penalty Method for Path Inequality Constraints

A spacecraft trajectory might also be subject to path inequality constraints. In the current discrete DDP formulation, the constraint is applied at each stage.

$$\boldsymbol{g}_k(\boldsymbol{X}_k) \leq \boldsymbol{0} \tag{6.1}$$

Equation 6.1 is a vector equation containing any number N_g of stage inequality constraints. Constraining the thrust magnitude to its maximum is a necessary path inequality constraint. If a control update during the backward sweep steps beyond the maximum thrust, it is reduced to the boundary and the full trust-region is used to update the direction. Similarly, if feedback terms bring the thrust beyond the maximum during the forward pass, thrust is again reduced to the boundary. Maximum thrust is a hard constraint that is limited by the physical capability of the propulsion system. Soft constraints are permitted some degree of violation beyond their bounds, so a penalty method is sufficient. If a stage constraint is active, then a penalty cost is assigned to that stage.

$$L_k = \begin{cases} cg_k(\boldsymbol{X}_k)^2, & g_k(\boldsymbol{X}_k) > 0 \\ 0, & g_k(\boldsymbol{X}_k) \leq 0 \end{cases} \tag{6.2}$$

Equation 6.2 assigns a quadratic penalty with weight c. Each constraint may be assigned its own penalty weight and a different penalty function, such as a cubic or exponential cost. The chosen penalty function should be twice differentiable. If an iterate is accepted with active stage constraints, the derivatives $L_{X,k}$ and $L_{XX,k}$ are also evaluated.

Reference [51] covers a more formal approach to path inequality constraints that involves solving for a Lagrange multiplier for each active constraint. Multipliers and path constraint violations are collectively gathered into the augmented Lagrangian like terminal constraints but with similar logic to the penalty method. Multipliers may be assigned for all stages and assigned values or nulled as constraints are activated or deactived, so that the stages with active constraints on the converged solution do not need to be known *a priori*. This approach may or may not add a large set of new variables to solve for. Constraining the flyby altitude through a gravity assist, for example, might only add a single multiplier at the stage of closest approach. This stage of closest approach might change between DDP iterations so that multipliers across a few stages are routinely updated en route to convergence. The stages with active constraints do not need to be known a priori. do not needAs an extreme example, an additional Lagrange multiplier could be necessary for every stage to constrain the thrust magnitude. Reference [83] attaches both a Lagrange multiplier and a slack variable to the control constraint at each stage, which proved effective for a small number of stages. Pellegrini points out that the slack variables can be handled implicitly without additional computation [51].

Penalty functions were used in Reference [39] to constrain the thrust magnitude and launch date. The launch date is a static parameter. The penalty method has broad application to control and static parameters, path constraints and terminal constraints and is simple to implement. Forming $g_k(\boldsymbol{X}_k)$ may require some ingenuity, namely to arrive at a twice-differentiable form. If analytic derivatives prove to be elusive, then a finite-differencing method might suffice. The ease of implementation and null effect on optimization problem size makes the penalty method attractive for low-thrust many-revolution trajectory optimization. In this chapter, the penalty method is leveraged to constrain apsis radii for a benchmark orbit transfer and to constrain the maximum duration in eclipse.

6.1   Case E Transfer

In Reference [24], low-thrust many-revolution trajectory optimization techniques were tested through orbit transfers of varying complexity, named Case A through Case E. The transfers were first introduced in Reference [22] to demonstrate the Q-law. Case B is a GTO to GEO transfer and was studied in Chapter 4 and Reference [57] to introduce and verify the Sundman-transformed DDP approach. Case E is the most challenging of the benchmark orbit transfers due to large rotations of the angular momentum and eccentricity vectors and is defined in Table 6.1. The boundary conditions were converted to the modified equinoctial elements in Table 6.2 for the current example and the true anomaly was chosen as the independent variable.

Table 6.1: Case E transfer setup.

Orbit	a (km)	e	i (deg)	ω (deg)	Ω (deg)
Initial	24505.9	0.725	0.06	0.0	0.0
Target	26500	0.7	116.0	270.0	180.0

T_{max} (N)	m_0 (kg)	I_{sp} (sec)	$\mu_\oplus$ (km^3/s^2)
2.0	2000	2000	398600.49

Table 6.2: Case E initial and target modified equinoctial elements.

Orbit	p (km)	f	g	h	k	L
Initial	11624.9863	0.725	0.0	5.2359$\times 10^{-4}$	0.0	0.0
Target	13515	0.0	0.7	-1.6003	0.0	

The dynamic model includes point-mass Earth gravity and thrust, i.e. the two-body model is used. The numerical setup is unchanged except for setting the discretization to 60 stages per revolution, MEE_{60}. The number of revolutions was varied to trace the Pareto front of fuel versus time of flight. The procedure began with ballistic initial guesses but with the initial out-of-plane steering set to $\beta_k = -\text{sgn}\left[\cos\left(\omega + \nu\right)\right]\pi/2$ for all stages, which maximizes the rate of decreasing inclination $di/dt < 0$ with respect to β. The initial guess also maximizes $d\Omega/dt$ for half of each revolution, as $\beta_k = \text{sgn}\left[\sin\left(\omega + \nu\right)\right]\pi/2$ maximizes $d\Omega/dt > 0$. Thus, the decreasing rate $d\Omega/dt < 0$

is maximized for the other half of each orbit. Reference [84] summarizes the optimal steering angles to maximize the instantaneous change in each orbital element, which could be used for constructing other initial guesses.

6.1.1 Apsis Constraints

Periapsis throughout the transfer is constrained to an altitude of 120 km. Without the periapsis constraint, DDP became prone to collapsing to the singularity $r = 0$ or converging to a trajectory that dipped below the Earth's surface. On the other end, apoapsis is unconstrained in the Case E definition but it proved useful to enforce a maximum apoapsis constraint to guide DDP toward favorable results. Without the apoapsis constraint, some trajectories would drift to an excessive orbital radius to perform an inexpensive plane change, but doing so ultimately led to local minima above the Pareto front of fuel expenditure versus time of flight. For other trajectory optimization exercises, orbit raising sometimes proceeded to reach escape trajectories. This became a failure mode for each of the Sundman transformations in Equation 4.18. The automatic step-size regulation provided by the Sundman transformation takes integration steps of increasing size on a departure hyperbola, growing like r^2 for the true anomaly transformation, for example. Adding the apoapsis constraint serves as a safeguard but the maximum allowed value becomes a tuning parameter. Various constraint values are discussed in the results. Both apsis constraints are applied to the osculating state at each stage with a quadratic penalty function. The relevant apsis constraints are

$$g_{0,k}(\boldsymbol{X}_k) = r_{per,min} - r_{per,k} \leq 0, \tag{6.3a}$$

$$g_{1,k}(\boldsymbol{X}_k) = r_{apo,k} - r_{apo,max} \leq 0, \tag{6.3b}$$

where $r_{per,min}$ is the minimum periapsis radius, $r_{per,k}$ and $r_{apo,k}$ are the periapsis and apoapsis radii at stage k and $r_{apo,max}$ is the maximum apoapsis radius. The penalty cost is assigned as described

in Equation 6.2 but summed over each of N_g constraints.

$$L_{i,k} = \begin{cases} c_i g_{i,k}(\boldsymbol{X}_k)^2, & g_{i,k}(\boldsymbol{X}_k) > 0 \\ \\ 0, & g_{i,k}(\boldsymbol{X}_k) \leq 0 \end{cases} \tag{6.4a}$$

$$L_k = \sum_{i=0}^{N_g-1} L_{i,k} \tag{6.4b}$$

Each stage incurs a local cost L_k that is the sum of penalty costs $L_{i,k}$ associated with each constraint $g_{i,k}$. Each constraint can be assigned its own weight c_i. That weight could be assigned to each stage as $c_{i,k}$ but was left uniform across all stages for simplicity, with $c_0 = 10^6$ and $c_1 = 10^3$. The first and second derivatives of $L_{i,k}$ with respect to $\boldsymbol{X}_k$ are evaluated and summed following the logic of Equation 6.4 when a DDP iteration is successful.

6.1.2 Augmented Lagrangian Cost Function

Once again the objective is to maximize the final mass. The terminal constraint function targets the modified equinoctial elements in Table 6.2 with equal weights on the quadratic cost and the final true longitude free. The augmented Lagrangian is redefined to include the local costs of all N_k stages associated with inequality constraint penalty functions.

$$J = \phi + \boldsymbol{\lambda}^T \boldsymbol{\psi} + \boldsymbol{\psi}^T \Sigma \boldsymbol{\psi} + \sum_{k=0}^{N_k-1} L_k \tag{6.5a}$$

$$\phi = -m_f \tag{6.5b}$$

$$\boldsymbol{\psi}_{MEE} = \begin{bmatrix} p_f - p_{target} \\ f_f \\ g_f - g_{target} \\ h_f - h_{target} \\ k_f \end{bmatrix} \tag{6.5c}$$

$$\Sigma = 10\, I_{5\times 5} \tag{6.5d}$$

6.1.3 Case E Transfer Results

The first set of computations left the apoapsis distance unconstrained and set the number of revolutions from 70 to 300 in steps of 10. While the 70 revolution transfer converged in just 29

minutes, optimization was terminated for transfers above 180 revolutions once the runtime reached 24 hours. The computational setup was identical to that in Chapter 4. The longest runtime of the converged transfers was the 150 revolution transfer at 11 hours and 34 minutes. The iteration counts for the 70 and 150 revolution transfers were 515 and 3806, respectively. The propellant mass and time of flight for the successful transfers are plotted along with the results from [24] in Figure 6.1.
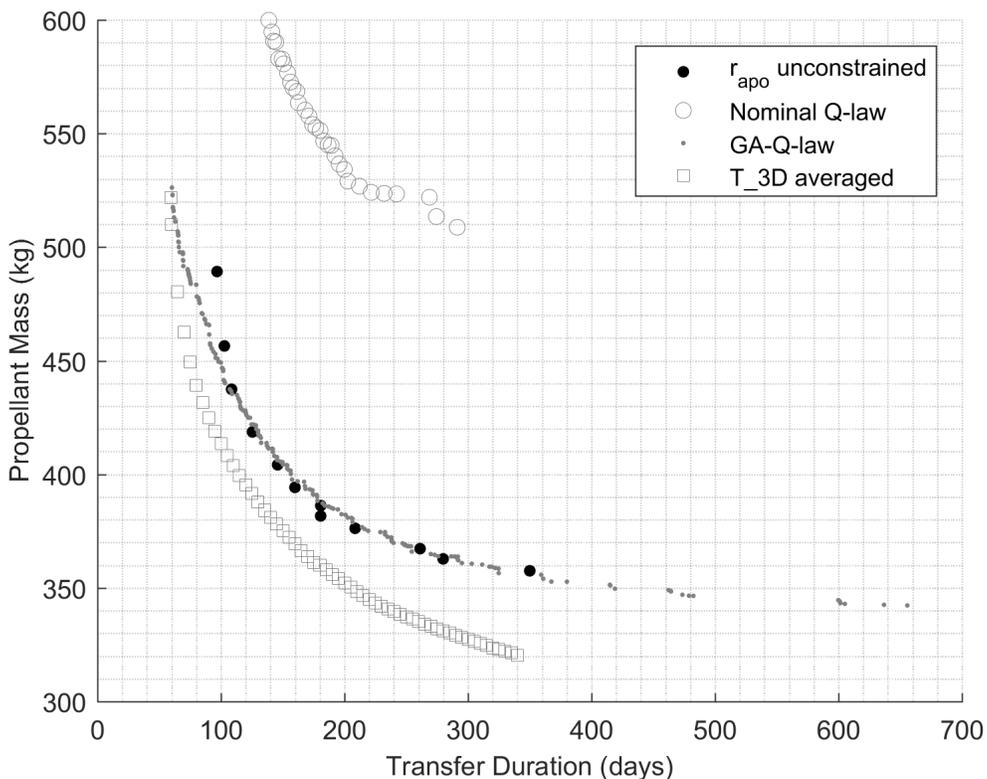
Figure 6.1: Case E trade-off of fuel versus time of flight with apoapsis unconstrained.

Figure 6.1 includes results from the Q-law with a nominal set of tuning parameters, the improved transfers from tuning the Q-law with a genetic algorithm (GA-Q-law) and transfers with averaged dynamics using T_3D, an indirect optimizer [85, 86]. The Q-law uses exact dynamics, so the improved GA-Q-law results offer the best comparison. The DDP-computed propellant mass requirements at the shortest times of flight rise above the expected Pareto front. For times of flight

around 150 to 200 days, DDP outperforms GA-Q-law before returning to the same front at longer times of flight. Petropoulos notes that the CPU runtimes to produce the Case E trade-off are on the order of a minute for the Q-law, tens of minutes for T_3D and tens of hours for GA-Q-law.

Next, the maximum-apoapsis constraint was activated with $r_{apo,max}$ set to 60, 70, 80, 90, 100, 150 and 200×10^3 km. Again, the number of revolutions was set from 70 to 300 in steps of 10, except for $r_{apo,max} = 60 \times 10^3$ km which was included for only 70 to 160 revolutions. Optimization began from a ballistic initial guess for each solution. The fuel versus time of flight results are gathered for all of the transfers in Figure 6.2. There are DDP trajectories falling below the GA-Q-law Pareto front across the range of times of flight. For each apoapsis constraint value, an elbow in the DDP curve outperforms GA-Q-law curve at intermediate values of time of flight. Then there are two intersections so that GA-Q-law finds better solutions than DDP at the shortest and longest times of flight. These intersections move to shorter flight times as the apoapsis constraint is decreased.

The DDP solutions were filtered to produce the Pareto front in Figure 6.3. There is increased overlap at the shorter flight times between solutions with different apoapsis constraint values. The trend is clear, however, that the increasing apoapsis constraint dominates with increasing flight time. There are noticeable inflections with each increment to $r_{apo,max}$ that could be removed by computing more trajectories with a finer resolution of $r_{apo,max}$.

The effect of varying the apoapsis constraint value becomes more evident when viewing the trade-off of propellant mass versus number of revolutions in Figure 6.4 and the trade-off of transfer duration versus number of revolutions in Figure 6.5. For a fixed number of revolutions, the tighter constraint increases the fuel requirement and decreases the time of flight. For $r_{apo,max} = 60 \times 10^3$ km, 70 and 80 revolutions were insufficient to complete the transfer, as were 70 revolutions for $r_{apo,max} = 70 \times 10^3$ km. These failures were recognized as DDP iterating on an infeasible solution without continuing to make progress. For $r_{apo,max} = 150 \times 10^3$ km, the constraint is inactive for transfers of 80, 90, 100 and 140 revolutions and for $r_{apo,max} = 200 \times 10^3$ km, the constraint is inactive for transfers ranging 70 to 170 revolutions. Those solutions are identical to the unconstrained solutions. Optimization was again terminated at 24 hours of runtime for the 270
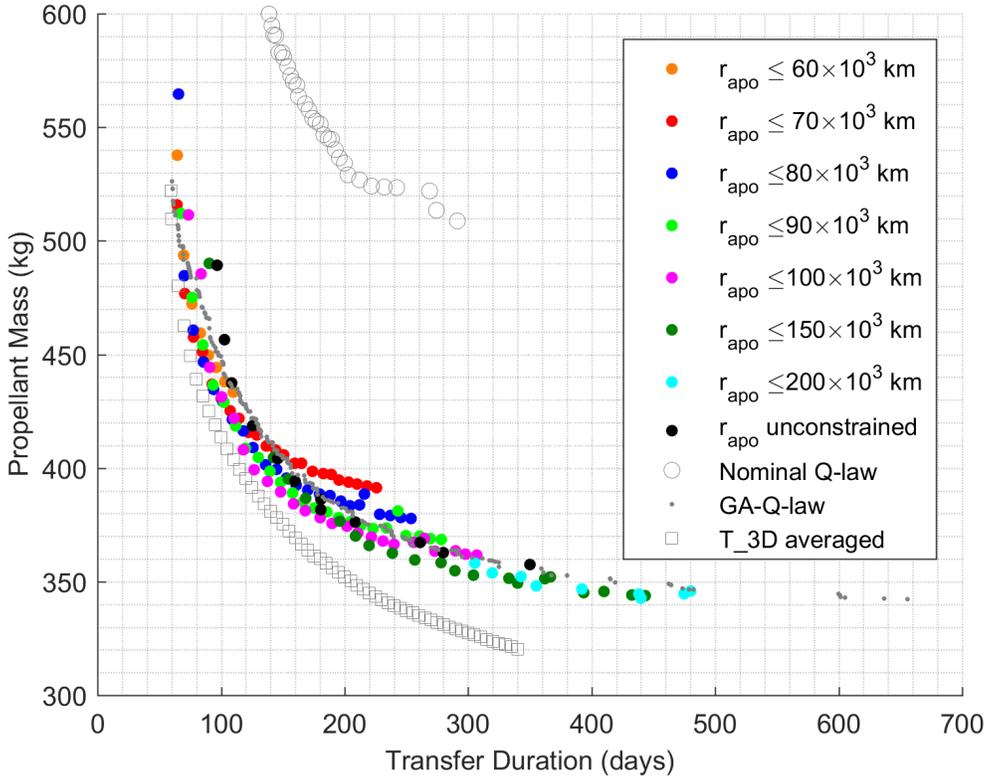
Figure 6.2: Case E trade-off of fuel versus time of flight for all of the DDP solutions.

to 300 revolution cases with $r_{apo,max} = 200 \times 10^3$ km. The solution process is not perfect every time, as evidenced by fluctuations in the trade-off curves in all of Figures 6.1 to 6.5. The affect of the apoapsis constraint on convergence time and number of iterations is unpredictable, while the affect of number of revolutions is more so. Computational effort ranged from 163 iterations in 11 minutes for the 90 revolution $r_{apo,max} = 60 \times 10^3$ km transfer to 5309 iterations in 17 hours and 31 minutes for the 250 revolution $r_{apo,max} = 200 \times 10^3$ km transfer.

Trajectory plots are shown for the 80 revolution $r_{apo,max} = 70 \times 10^3$ km and 300 revolution $r_{apo,max} = 100 \times 10^3$ km transfers in Figures 6.6 and 6.7. Markers are placed at the initial and final states, which are the periapsis states of the initial and target orbits. Time histories of r_{apo}, r_{per}, i, Ω and ω are shown in Figure 6.8, which most notably show r_{apo} on the constraint boundaries and that large changes in Ω and ω can be accomplished early in the transfer when the orbit is nearly
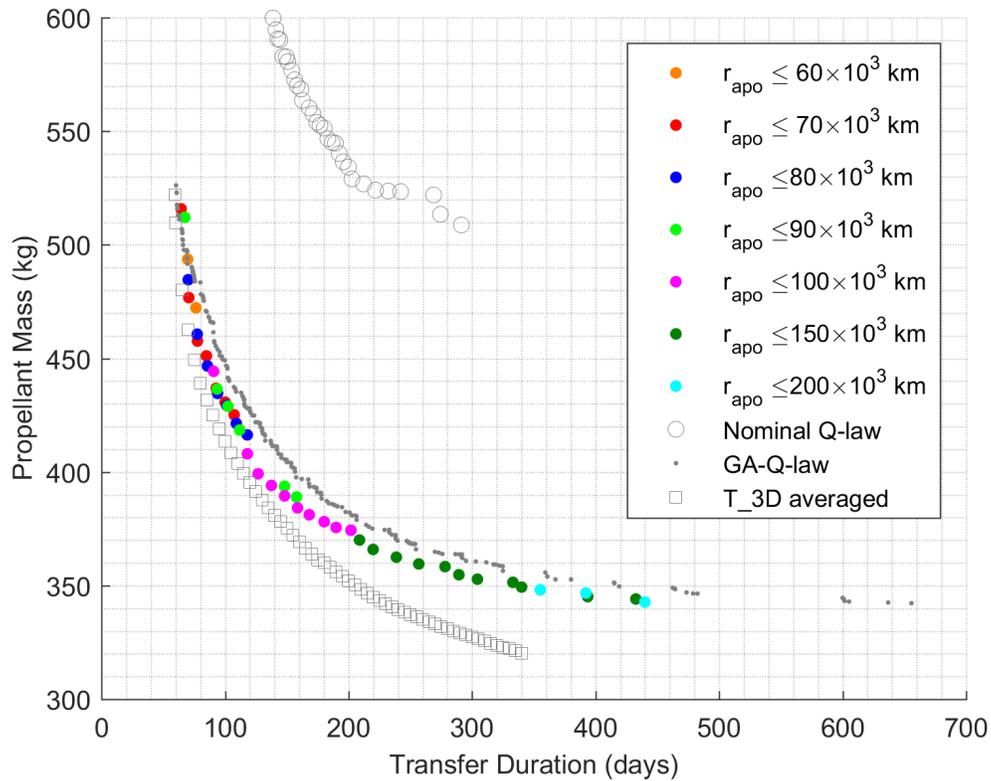
Figure 6.3: Case E Pareto front of fuel versus time of flight.

equatorial. While Figure 6.6 shows the shortest time of flight DDP solution, there is still significant

coasting visible, suggesting that a much shorter time of flight is possible. Indeed, Figures 6.2 and 6.3

show several GA-Q-law and T_3D solutions with shorter times of flight. As more revolutions and

consequently more time of flight are available to complete the transfer, thrust arcs reduce to small

durations and much of the transfer is spent coasting between these more fuel-efficient maneuvers.

6.2   Maximum Eclipse Duration

The motivation to constrain the maximum eclipse duration arose from mission design for a

thermally limited spacecraft. If the spacecraft were to spend too much time in eclipse, then certain

system components would fail. The concern is for the duration of any individual eclipse, not the

sum total from all eclipses along a trajectory. Furthermore, this design challenge is not limited to
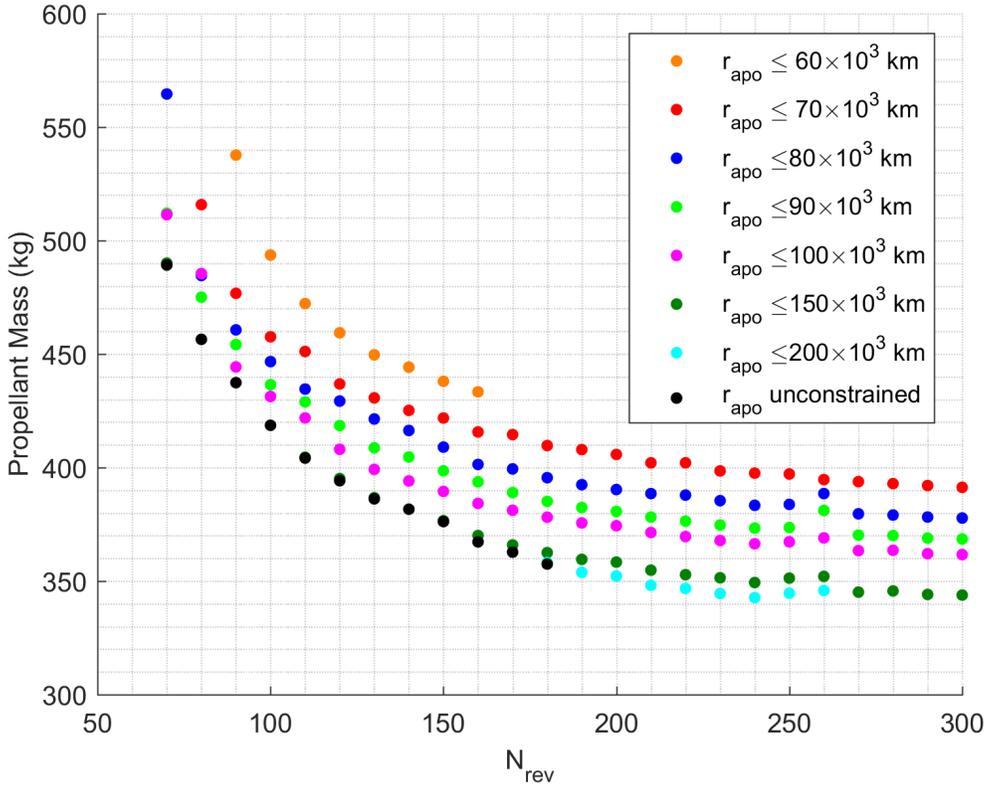
Figure 6.4: Case E trade-off of fuel versus number of revolutions.

low-thrust propulsion nor many-revolution trajectories.

The relevant form of Equation 6.1 is

$$g_k(\boldsymbol{X}_k) = \Delta t_{eclipse,k} - \Delta t_{eclipse,max} \leq 0, \tag{6.6}$$

where $\Delta t_{eclipse,k}$ is the duration of the eclipse in which stage k falls, and $\Delta t_{eclipse,max}$ is the maximum allowed duration of any individual eclipse. Most frequently, $\Delta t_{eclipse,k} = 0$. As in Chapter 5, the eclipse condition is considered binary, so that $\Delta t_{eclipse,k}$ accumulates during partial eclipse. If successive stages k, $k+1$,... fall in the same eclipse passage, then each stage incurs its own cost associated with $g_k(\boldsymbol{X}_k)$, $g_{k+1}(\boldsymbol{X}_{k+1})$ and so on.

The smoothed eclipse model in Chapter 5 automatically identifies when $\Delta t_{eclipse,k} \neq 0$. However, an output of the eclipse duration does not accompany the computed percent sunlight. To enforce Equation 6.6 effectively, an analytic and twice-differentiable equation for $\Delta t_{eclipse,k}$ is
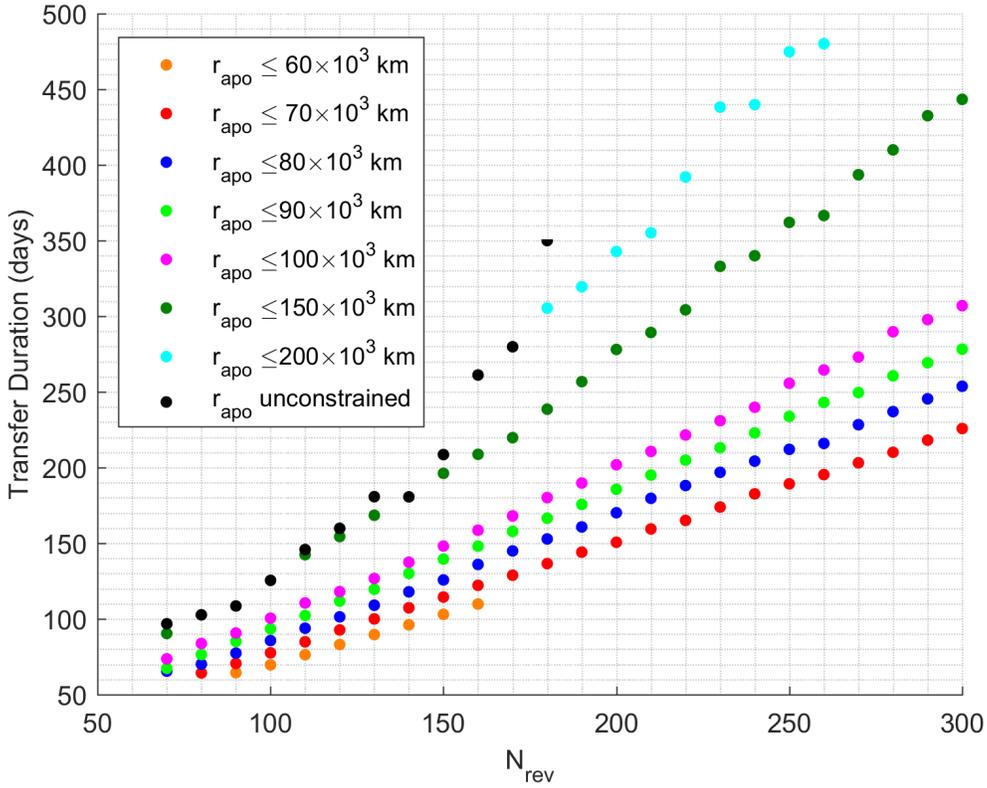
Figure 6.5: Case E trade-off of time of flight versus number of revolutions.

desired. Chapter 5 considers conical eclipse geometry and non-Keplerian motion. The time of flight between penumbra entry and exit must be computed with a numerical routine. An example approach numerically integrates the trajectory until a penumbra crossing is detected and uses the bisection method to adjust the final step size to place the final state on the edge of the penumbral cone. That procedure would need to be repeated for each state variable to obtain the finite-difference derivatives and become overly cumbersome.

Keplerian motion yields the famous time of flight equation, but finding intersections of the orbit and penumbral cone requires an iterative procedure [87, 88]. The assumption of Keplerian motion is not a far reach when thrusting is disallowed through eclipse and the passage is a short arc segment. Furthermore, an exact measure of $\Delta t_{eclipse,k}$ is not a requirement. The calculation simply needs to be good enough and that level of sufficiency is problem dependent. With an understanding
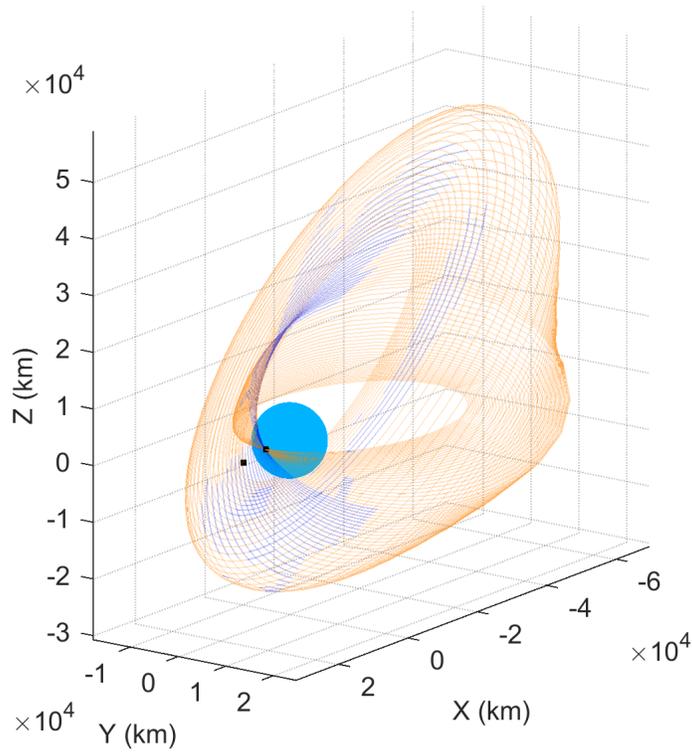
Figure 6.6: The 80 revolution Case E transfer constrained to $r_{apo,max} = 70 \times 10^3$ km.

of how a modeled value of $\Delta t_{eclipse,k}$ is over or under conservative, the penalty weight and constraint value can be tuned to meet mission design requirements. In lieu of conical eclipse geometry, a simpler cylindrical shadow model is used to reach a twice-differentiable $\Delta t_{eclipse,k}$. When also assuming Keplerian motion and the Sun's position to be fixed through an eclipse passage, shadow entry and exit locations can be obtained as the roots of a quartic equation. The time of flight equation can then be applied across the shadow transit to give $\Delta t_{eclipse,k}$. Reference [88] illustrates the potential error for a worst case scenario where the orbit skims the shadow cylinder. Then the cylindrical model doesn't show any time in eclipse although there could be significant time spent in the penumbral cone.
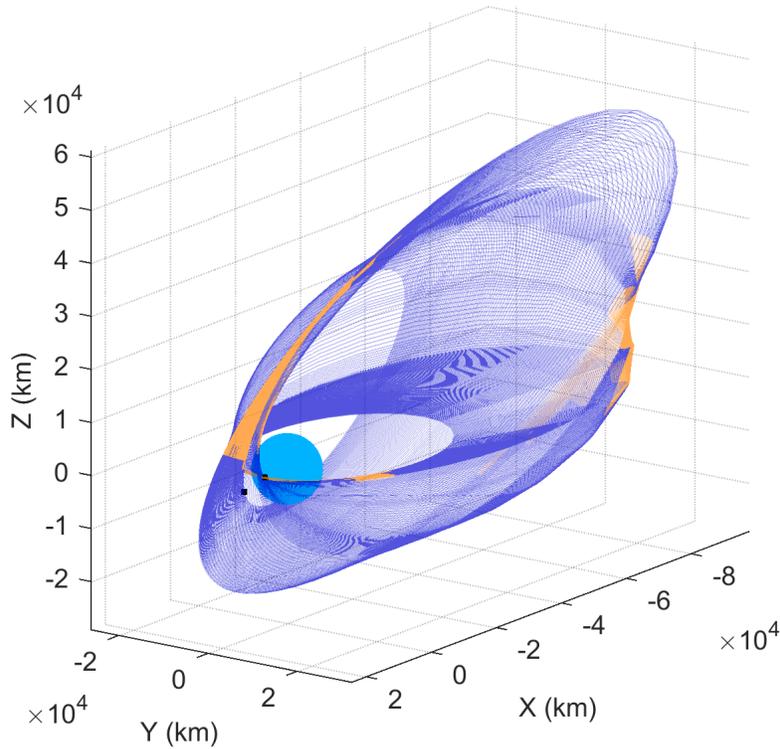
Figure 6.7: The 300 revolution Case E transfer constrained to $r_{apo,max} = 100 \times 10^3$ km.

6.2.1 Cylindrical Shadow Geometry

A cylindrical shadow results from the assumption that the Sun is infinitely far away. Light rays shine antiparallel to the Sun vector $\boldsymbol{s}$ from the central body so that a cylindrical shadow forms with infinite length and a radius equal to that of the body, R_B generally or $R_\oplus$ for the Earth. Figure 6.9 is a redrawing from Reference [5] that illustrates the Earth shadow cylinder and a spacecraft orbit. A spacecraft position $\boldsymbol{r}$ is shown during an eclipse passage with marked entry and exit points. The angle ζ is measured between the Sun vector and spacecraft position.

$$\cos \zeta = \hat{\boldsymbol{s}}^T \hat{\boldsymbol{r}} \qquad (6.7)$$

If $\zeta < 90°$ then the spacecraft is closer to the Sun than the Earth is and there is no need to proceed further in checking for eclipse. If $\zeta > 90°$, then an eclipse occurs if $r \sin \zeta < R_\oplus$. This logic could be used in the same manner as the conical eclipse transition in Chapter 5 to produce a cylindrical
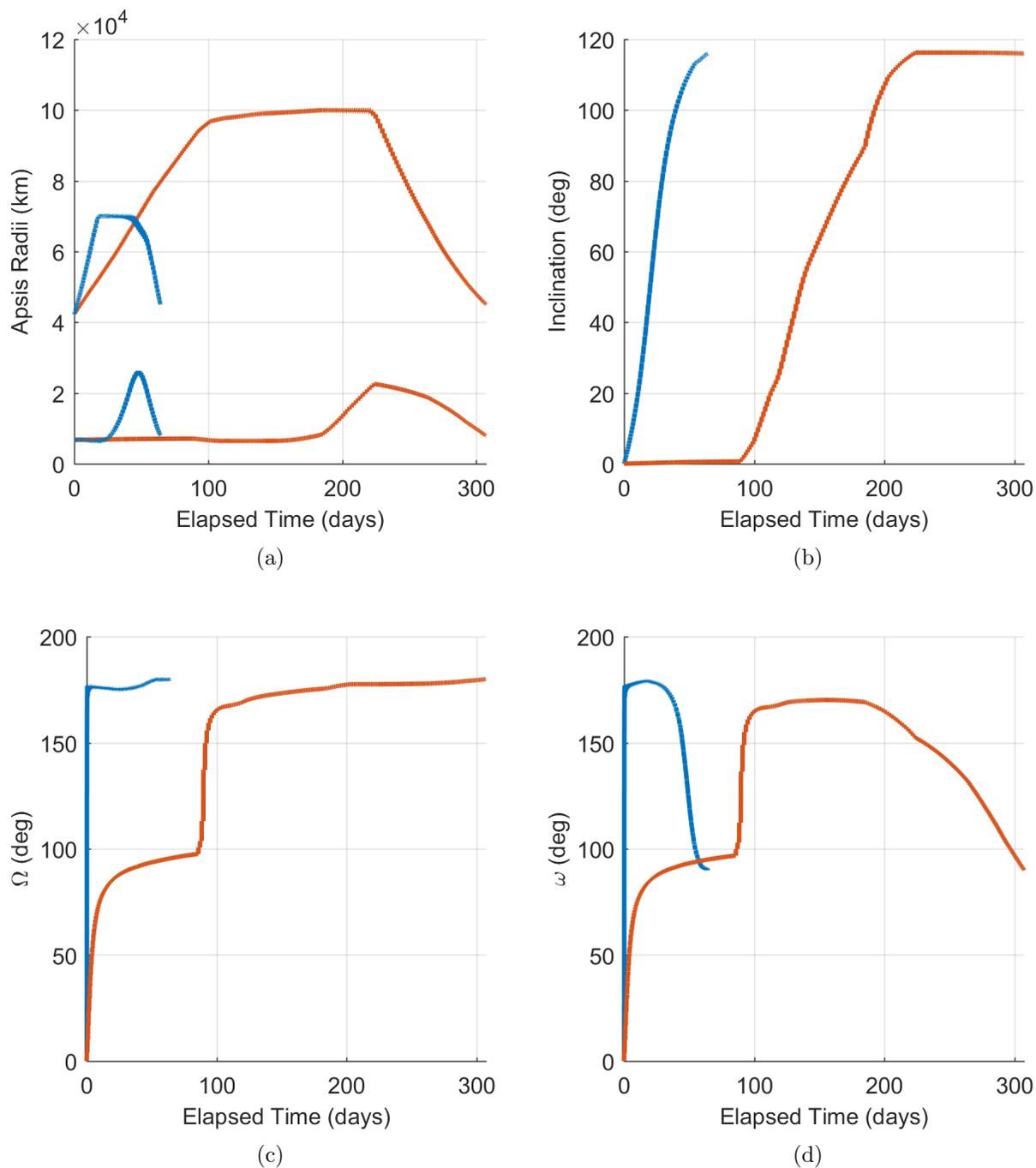
Figure 6.8: (a) Apsis radii, (b) inclination, (c) right ascension of the ascending node and (d) argument of periapsis for 80 revolution $r_{apo,max} = 70 \times 10^3$ km and 300 revolution $r_{apo,max} = 100 \times 10^3$ km Case E transfers.

logistic sunlight fraction.
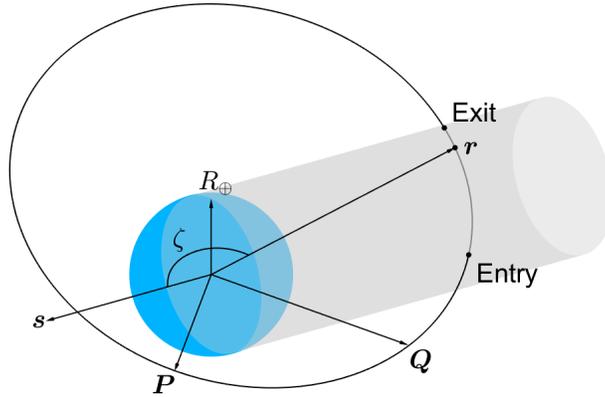
Figure 6.9: An example spacecraft orbit is shown intersecting a cylindrical Earth shadow.

The current cylindrical shadow analysis makes use of the perifocal coordinate system (PQW), where $\boldsymbol{P}$ is aligned with the eccentricity vector and points toward periapsis, $\boldsymbol{W}$ (not pictured) is aligned with the angular momentum and $\boldsymbol{Q}$ completes the right-handed frame to align with the position at a true anomaly of $90°$.

$$\boldsymbol{P} = \begin{bmatrix} \cos\omega\cos\Omega - \sin\omega\sin\Omega\cos i \\ \cos\omega\sin\Omega + \sin\omega\cos\Omega\cos i \\ \sin\omega\sin i \end{bmatrix} \tag{6.8a}$$

$$\boldsymbol{Q} = \begin{bmatrix} -\sin\omega\cos\Omega - \cos\omega\sin\Omega\cos i \\ -\sin\omega\sin\Omega + \cos\omega\cos\Omega\cos i \\ \cos\omega\sin i \end{bmatrix} \tag{6.8b}$$

As the spacecraft moves through the shadow cylinder, the central body is advancing in its heliocentric orbit. Thus, $\boldsymbol{s}$ is changing and the shadow cylinder should move with it. The planetocentric motion of the spacecraft occurs on a shorter timescale than the body's heliocentric motion. It is then a mild assumption to hold $\boldsymbol{s}$ fixed for a given eclipse passage.

6.2.2   Spacecraft Beta Angle

An important quantity for the study of spacecraft eclipse is the angle between the spacecraft's planetocentric orbital plane and the Sun, named the beta angle $\beta_\odot$. An expression for $\beta_\odot$ in

Reference [5] uses the right ascension $\alpha_\odot$ and declination $\delta_\odot$ of the Sun.

$$\sin\beta_\odot = \cos\delta_\odot \sin i \sin(\Omega - \alpha_\odot) + \sin\delta_\odot \cos i \qquad (6.9)$$

The solar right ascension and declination are given by

$$\tan\alpha_\odot = \frac{y_\odot}{x_\odot}, \qquad (6.10a)$$

$$\tan\beta_\odot = \frac{z_\odot}{\sqrt{x_\odot^2 + y_\odot^2}}. \qquad (6.10b)$$

Vallado [5] also gives an expression for the eclipse duration for a circular orbit at a given beta angle,

$$\Delta t_{eclipse,circ} = \cos^{-1}\left(\frac{\sqrt{1 - \left(\frac{R_B}{r}\right)^2}}{\cos\beta_\odot}\right)\frac{P}{\pi}, \qquad (6.11)$$

where P is the orbital period,

$$P = 2\pi\sqrt{\frac{a^3}{\mu}}. \qquad (6.12)$$

Equation 6.11 could be used to evaluate $\Delta t_{eclipse,k} = \Delta t_{eclipse,circ}$ in the constraint function Equation 6.6. For elliptical orbits the measure is conservative for periapsis eclipses but underestimates the duration of an apoapsis eclipse.

6.2.3   Eclipse Maps

Mission orbit selection and transfer analysis can be facilitated by exploiting Equation 6.11. Varying the orbital radius and beta angle in Equation 6.11 maps out the possible eclipse durations for circular orbits. Eclipse maps for Earth and Mars are shown in Figures 6.10a and 6.10b for orbits ranging from an altitude of 100 km to a radius of 100,000 km. At lower altitudes, eclipses are encountered through a wide range of beta angles, but the orbital velocities are high and the eclipses are short in duration. That range of susceptible beta angles decreases with increasing altitude, but so does orbital velocity. Thus, eclipse durations increase. Since the radius of the shadow cylinder is fixed, the fraction of an orbital period spent in eclipse decreases with radius, despite the increasing eclipse duration.
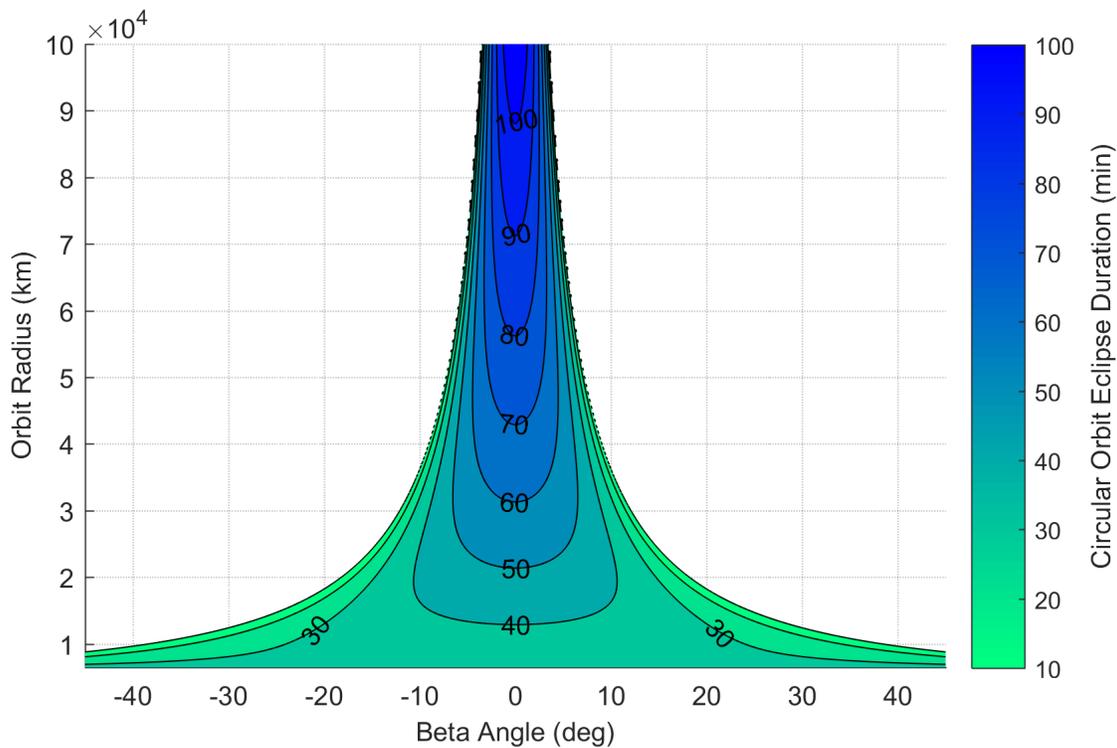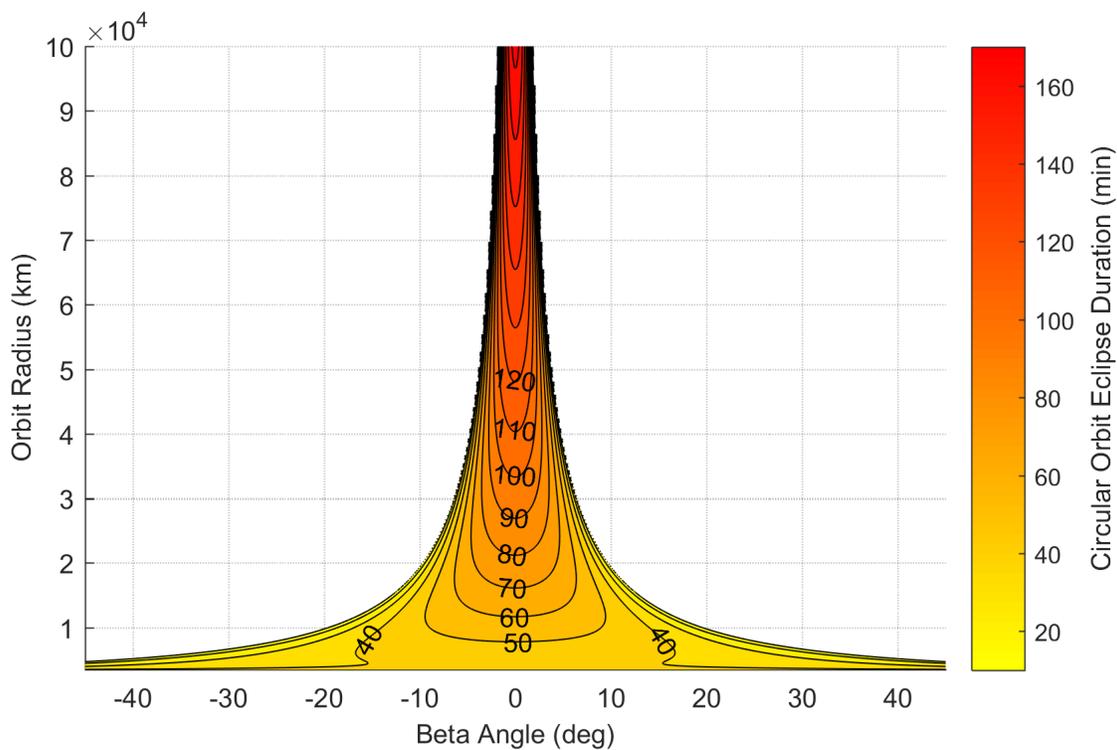
Figure 6.10: (a) An Earth eclipse map and (b) a Mars eclipse map show contours of eclipse duration for circular orbits of various radii and beta angles in a cylindrical shadow model.

6.2.4   The Shadow Quartic

An obvious but not so useful expression for the eclipse duration is the difference between the shadow exit time t_{exit} and entry time t_{entry}.

$$\Delta t_{eclipse} = t_{exit} - t_{entry} \tag{6.13}$$

The Keplerian motion assumption permits a restatement of $\Delta t_{eclipse}$ using the time of flight equation,

$$\Delta t_{eclipse} = \frac{M_{exit} - M_{entry}}{n}, \tag{6.14}$$

where M_{exit} and M_{entry} are the mean anomalies at shadow exit and entry and n is the mean motion.

$$n = \sqrt{\frac{\mu}{a^3}} \tag{6.15}$$

References [5, 89–91] present a quartic equation that is satisfied by the true anomalies at entry and exit ν_{entry} and ν_{exit}. Those can be converted to eccentric anomalies E_{entry} and E_{exit} which in turn yield M_{entry} and M_{exit} for use in Equation 6.14. This presentation closely follows Vallado's [5] though it is noted that Edelbaum's [90] presentation uses an equinoctial element set. The latter has been implemented in the Solar Electric Control Knob Setting Program by Optimal Trajectories (SECKSPOT), which includes a delay in thruster start-up after exiting eclipse [92, 93].

The true anomaly is introduced to the shadow analysis by restating the spacecraft position in the perifocal frame.

$$\boldsymbol{r} = r \cos \nu \boldsymbol{P} + r \sin \nu \boldsymbol{Q} \tag{6.16}$$

Equations 6.7 and 6.16 yield an expression for the Sun-body-spacecraft angle as a function of the true anomaly.

$$\cos \zeta = \cos \nu \hat{\boldsymbol{s}}^T \boldsymbol{P} + \sin \nu \hat{\boldsymbol{s}}^T \boldsymbol{Q} \tag{6.17}$$

The notation is simplified by defining,

$$\beta_1 = \hat{\boldsymbol{s}}^T \boldsymbol{P}, \tag{6.18a}$$

$$\beta_2 = \hat{\boldsymbol{s}}^T \boldsymbol{Q}, \tag{6.18b}$$

so that Equation 6.17 becomes

$$\cos \zeta = \beta_1 \cos \nu + \beta_2 \sin \nu. \tag{6.19}$$

Vallado points out that elliptical orbits can only encounter eclipse if

$$1 - \left(\frac{R_B}{a(1-e)}\right)^2 < \beta_1^2 < 1 - \left(\frac{R_B}{a(1+e)}\right)^2. \tag{6.20}$$

Using Equation 6.19 in the intersection of the eclipse and the shadow cylinder $r \cos \zeta = R_B$ gives an expression that is satisfied by ν_{entry} and ν_{exit}, but first the position magnitude is replaced by a function of true anomaly, the trajectory equation.

$$r = \frac{p}{1 + e \cos \nu} \tag{6.21}$$

A shadow equation is then obtained by using Equations 6.19 and 6.21 in $r \cos \zeta = R_B$, squaring both sides and rearranging terms.

$$S = R_B^2(1 + e \cos \nu)^2 + p^2(\beta_1 \cos \nu + \beta_2 \sin \nu)^2 - p^2 \tag{6.22}$$

The shadow quartic results from converting to exclusively $\cos \nu$ terms,

$$S = \alpha_1 \cos^4 \nu + \alpha_2 \cos^3 \nu + \alpha_3 \cos^2 \nu + \alpha_4 \cos \nu + \alpha_5, \tag{6.23}$$

where

$$\alpha = \frac{R_B}{p}, \tag{6.24a}$$

$$\alpha_1 = \alpha^4 e^4 - 2\alpha^2 e^2(\beta_2^2 - \beta_1^2) + (\beta_1^2 + \beta_2^2)^2, \tag{6.24b}$$

$$\alpha_2 = 4\alpha^4 e^3 - 4\alpha^2 e(\beta_2^2 - \beta_1^2), \tag{6.24c}$$

$$\alpha_3 = 6\alpha^4 e^2 - 2\alpha^2(\beta_2^2 - \beta_1^2) - 2\alpha^2 e^2(1 - \beta_2^2) + 2(\beta_2^2 - \beta_1^2)(1 - \beta_2^2) - 4\beta_2^2\beta_1^2, \tag{6.24d}$$

$$\alpha_4 = 4\alpha^4 e - 4\alpha^2 e(1 - \beta_2^2), \tag{6.24e}$$

$$\alpha_5 = \alpha^4 - 2\alpha^2(1 - \beta_2^2) + (1 - \beta_2^2)^2. \tag{6.24f}$$

The shadow quartic is satisfied for values of $\cos \nu$ that produce $S = 0$. Those roots can be obtained analytically, but recovering the true anomaly requires checking both $\cos(\pm\nu)$. Eight

possible values of the true anomaly must be reduced to the correct ν_{entry} and ν_{exit}. The first check requires $\zeta > 90°$ or equivalently $\cos\zeta < 0$. Second, $\sin\nu \neq \sin(-\nu)$ so $S = 0$ can be checked in Equation 6.22. Whether the true anomaly corresponds to an entry or exit can be determined by the sign of $\partial S/\partial\nu$, where S is increasing for ν_{entry} and decreasing for ν_{exit}. Then the eccentric anomalies are obtained using

$$\sin E = \frac{\sin\nu \sqrt{1 - e^2}}{1 + e\cos\nu}, \tag{6.25a}$$

$$\cos E = \frac{e + \cos\nu}{1 + e\cos\nu}, \tag{6.25b}$$

$$E = \text{atan2}(\sin E, \cos E). \tag{6.25c}$$

Finally, M_{entry} and M_{exit} are given by

$$M = E - e\sin E, \tag{6.26}$$

and $\Delta t_{eclipse}$ may be calculated with Equation 6.14.

6.2.5   Evaluating the Eclipse Duration and its Derivatives

The maximum eclipse duration constraint is checked during the cost function evaluation. Equation 6.14 is evaluated at each stage k using the current state to give an osculating eclipse duration $\Delta t_{eclipse,k} = \Delta t_{eclipse}(\boldsymbol{X}_k)$. The first step in that routine is to check the eclipse requirements $\zeta > 90°$ and $r\sin\zeta < R_B$. If there is no eclipse, then $\Delta t_{eclipse,k} = 0$. Otherwise, the routine proceeds with the steps of Section 6.2.4. The derivatives $L_{X,k}$ and $L_{XX,k}$, thus $\partial \Delta t_{eclipse,k}/\partial \boldsymbol{X}_k$ and $\partial^2 \Delta t_{eclipse,k}/\partial \boldsymbol{X}_k^2$ are computed for successful iterations and active constraints $\Delta t_{eclipse,k} > \Delta t_{eclipse,max}$. Application of the chain rule through each of the steps in Section 6.2.4 has proved to be an efficient approach. The process should include the derivatives of the PQW frame with respect to $\boldsymbol{X}$, the time derivative of $\boldsymbol{s}$ and logical statements to carry forward the derivatives corresponding to the correct roots of the shadow quartic. The last statement similarly applies to a cubic equation that is solved within the quartic solution.

The constraint has been implemented with the modified equinoctial element state representation. A singularity in the derivatives arises for circular orbits, where $e = \sqrt{f^2 + g^2} = 0$.

$$\begin{bmatrix} \dfrac{\partial e}{\partial f} & \dfrac{\partial e}{\partial g} \end{bmatrix} = \begin{bmatrix} \dfrac{f}{\sqrt{f^2 + g^2}} & \dfrac{g}{\sqrt{f^2 + g^2}} \end{bmatrix} \tag{6.27}$$

The derivatives of eccentricity are not taken explicitly since e is eliminated when substituting in the modified equinoctial elements, but that does not prevent division by $\sqrt{f^2 + g^2}$. At the same time, the eccentricity vector and $\boldsymbol{P}$ require redefinition. These challenges can be mitigated by forcing a small eccentricity magnitude or by using Equation 6.11 when a circular orbit is detected.

6.2.6   Eclipse-Constrained Mars Orbit Transfer

An example mission to Mars requires a specific mass, orbital plane, arrival epoch and zero eccentricity at the altitude of Deimos. The trajectory optimization problem is posed backward in time to compute the minimum starting mass at a 100,000 km circular orbit as a plausible state shortly after interplanetary arrival with $C_3 = 0$ at the sphere of influence. First, no concern is given for the maximum eclipse duration, but eclipsing is included in the power model. Then, a maximum eclipse duration is enforced by the penalty method and cylindrical shadow model.

The transfer setup is described in Tables 6.3 and 6.4. The initial orbit plane is specified with $i = 92.7°$ and $\Omega = 30.0°$. Target values of h, k and L are unconstrained. Dynamics include the J_2 perturbation, although the effect is small in this high-altitude regime. The sharpness coefficient $c_s = 100$ for use in the smoothed eclipse model from Chapter 5 was simply selected without analysis as a large enough value to provide a near immediate step between 100% and 0% power. The reference epoch sets the initial time t_0 at midnight January 1, 2024. The transfer was initially fixed to 40 revolutions ($N_{rev} = 40$) and discretized to 60 stages per revolution with modified equinoctial elements (MEE_{60}).

Since the equations of motion are being evaluated backward in time, mass accumulates during thrusting. The objective is to minimize that accumulation from a fixed starting mass. The terminal constraint function measures the errors in p_f, f_f and g_f which are all weighted equally. These

Table 6.3: Initial and target orbits for the eclipse-constrained Mars orbit transfer.

p_0	23460.0 km	p_{target}	1×10^5 km
f_0	0.0	f_{target}	0.0
g_0	0.0	g_{target}	0.0
h_0	0.907828		
k_0	0.524135		
L_0	0.0		

Table 6.4: Dynamic model parameters for the eclipse-constrained Mars orbit transfer.

μ_δ	42828.3 km^3/s^2	J_2	1.96045×10^{-3}
R_δ	3389.9 km	(c_s, c_t)	$(100.0, 1.0)$
$R_\odot$	695500 km	T_{max}	0.2907 mN
m_0	3000 kg	I_{sp}	3940 sec
Reference Epoch		2460310.5 Julian Date TDB	

components of the augmented Lagrangian cost function are summarized in Equation 6.28.

$$\phi = m_f \tag{6.28a}$$

$$\psi = \begin{bmatrix} p_f - p_{target} \\ f_f \\ g_f \end{bmatrix} \tag{6.28b}$$

$$\Sigma = I_{3\times3} \tag{6.28c}$$

The unconstrained trajectory is shown in Figure 6.11. Viewing the trajectory backward in time, initial continuous thrusting eventually gives way to a sequence of burns through periapsis and coast arcs through apoapsis. Thrusting to raise apoapsis dominates the middle of the transfer but the maneuver scheme switches toward the end of the transfer, with coasting through periapsis and thrusting through apoapsis to circularize the orbit. Eclipses and a reduction in the logistic sunlight fraction are shown to interfere with thrusting toward the ends of the periapsis burn sequence.

The transfer is depicted on an eclipse map in Figure 6.12. The view is bounded from the initial $r_0 = 23460.0$ km and $\beta_0 = -33.48°$ to the final $r_f = 100 \times 10^3$ km and $\beta_f = 23.59°$. The first three of the preceding values are specified while the final beta angle is an output. The
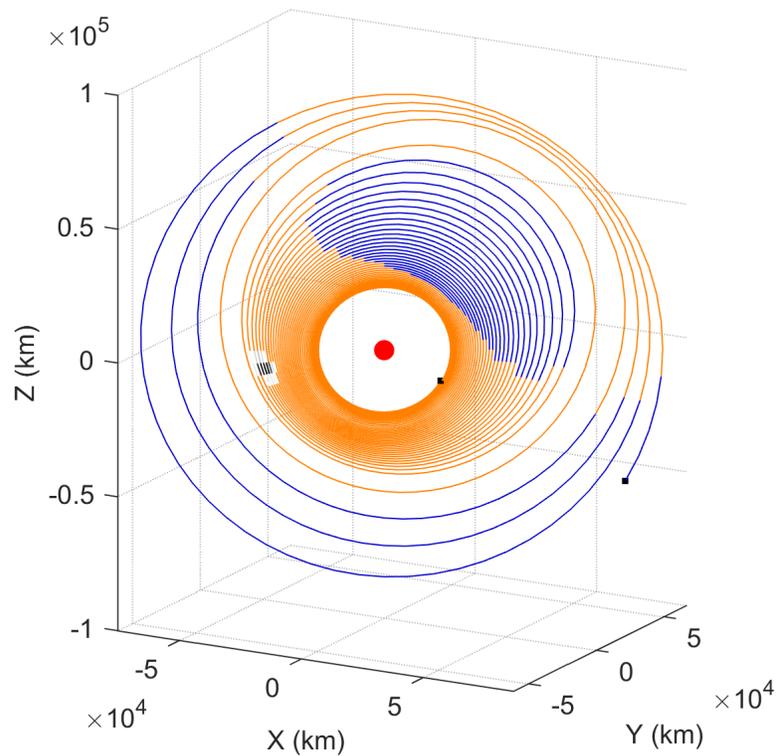
Figure 6.11: The Mars orbit transfer with the smoothed eclipse model active and unconstrained eclipse durations.

trajectory moves backward in time from left to right and the sinusoidal behavior of the radius is indicative of eccentricity being increased and then removed during the transfer. The transfer spans 125.6 days adds 53.93 kg of propellant. Eclipse season occurs in the middle of the transfer. Square markers indicate five eclipse events that were identified by checking the logistic sunlight fraction $\gamma_L \leq 0.5$. The first eclipse enters the penumbral cone but not the shadow cylinder, while the remaining eclipses pass through the shadow cylinder with durations of 87.8, 104.4, 102.5 and 75.1 minutes. Each actual eclipse duration is shorter than what appears on the eclipse map because it occurs near periapsis of a larger orbit and the spacecraft is moving faster through eclipse than a smaller circular orbit at that radius. Eclipse passages near apoapsis would be longer in duration than what is shown on the eclipse map.

The maximum eclipse duration constraint was tested by reducing the constraint value until
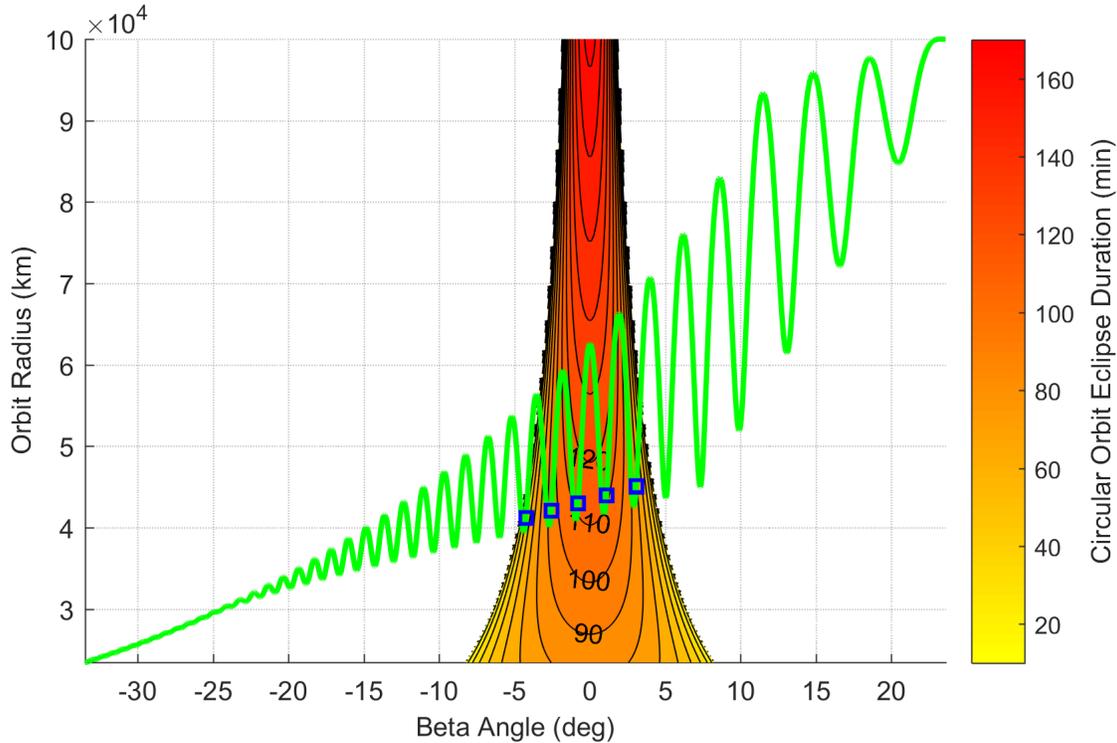
Figure 6.12: An eclipse map of the unconstrained Mars orbit transfer.

convergence failed. First, the penalty weight was fixed to $c = 1000$ and DDP converged with a constraint as low as $\Delta t_{eclipse,max} = 94$ minutes before a reduction to 93 minutes failed. A trajectory view and eclipse map of the constrained transfer are shown in Figures 6.13 and 6.14. Both visualizations show how the eccentricity vector is oriented for eclipses to occur during passes through periapsis so that the duration is minimized. Figure 6.14 also shows, however, that passing through eclipse season at lower orbital radii adds another eclipse passage to the transfer. The first and last eclipses pass through the penumbral cone but not the shadow cylinder, where the middle eclipses pass in 84.7, 94.0, 94.0 and 84.5 minutes. The time of flight decreases from the unconstrained case to 117.5 days while the fuel requirement increases to 54.1 kg. The trade-off in fuel versus time of flight is encountered once again, but now the maximum eclipse duration becomes a third objective to consider. The present method could be used to create a Pareto surface of fuel versus time of flight versus eclipse duration as an extension of the previously discussed fuel versus time of flight

Pareto curves. That was pursued to some degree by extending the number of revolutions to 41 and 42, which allowed tighter constraints on the eclipse duration of 93 and 92 minutes. The resulting times of flight and fuel requirements are 126.2 days and 53.85 kg for $N_{rev} = 41$ and 123.9 days and 53.89 kg for $N_{rev} = 42$. There is little additional qualitative information beyond that already gathered for the constrained $N_{rev} = 40$ case in Figures 6.13 and 6.14.
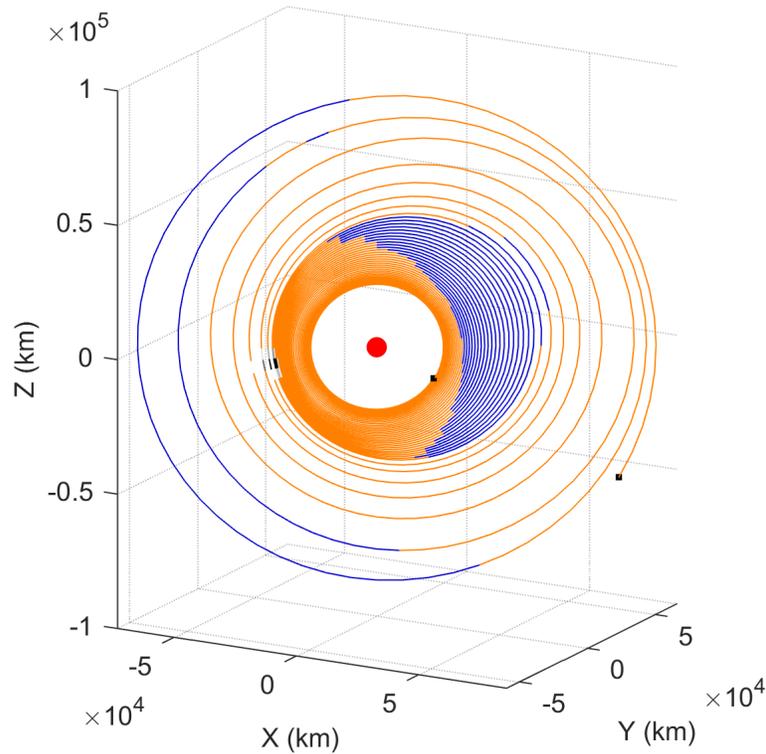
Figure 6.13: The Mars orbit transfer with $\Delta t_{eclipse,max} = 94$ minutes.

A second example changes the specified orbital plane to $\Omega = 10°$ by setting $h_0 = 1.032344$ and $k_0 = 0.182030$. From these conditions the unconstrained transfer encounters one eclipse spanning 150.0 minutes. The time of flight is 124.3 days and the fuel requirement is 53.915 kg. Eclipse season is passed at a large orbital radius where only a narrow range of beta angles are susceptible to eclipse. The eclipse map shows a steep gradient in eclipse duration across beta angles at large radii, indicating that a large change to the eclipse duration can be accomplished with just a small
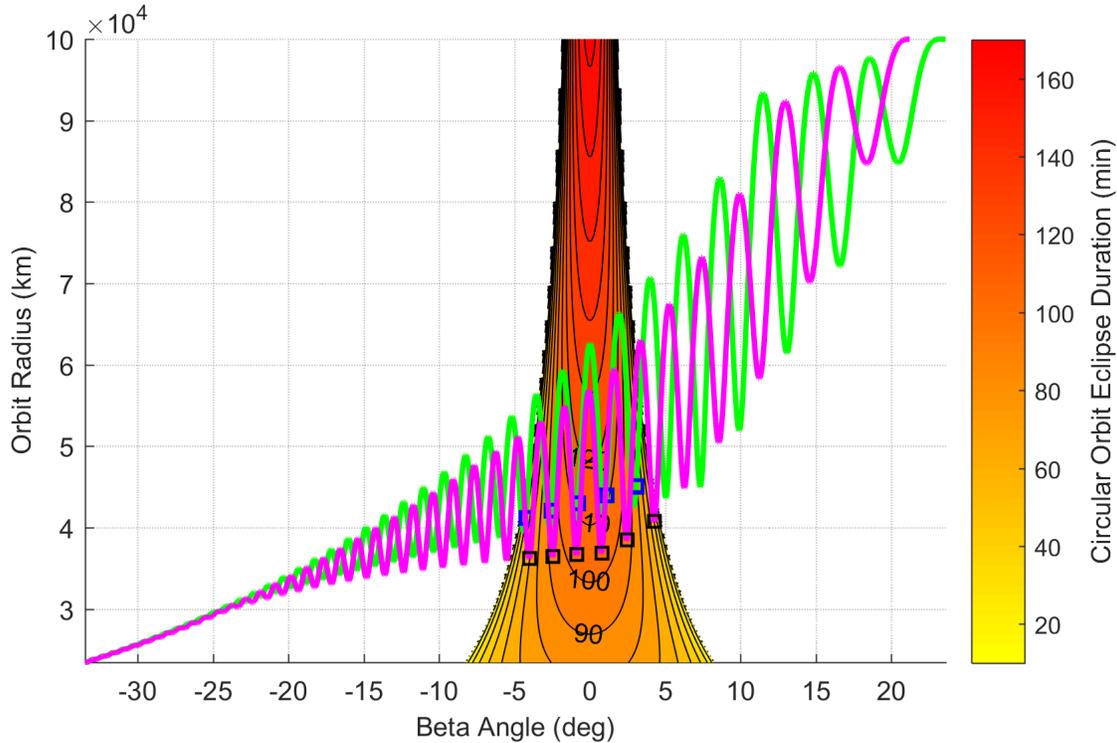
Figure 6.14: An eclipse map of both the Mars orbit transfer with $\Delta t_{eclipse,max} = 94$ minutes and the unconstrained transfer

change to the transfer. The increased sensitivity, however, is the likely cause for slow convergence (3324 iterations) to reduce the eclipse duration to 90.0 minutes. That adjustment is accomplished by re-phasing the eccentricity change which increases the time of flight to 126.4 days. The fuel requirement sees less impact and increases to 52.924 kg. Figure 6.15 shows an eclipse map of the unconstrained transfer in green and constrained transfer in magenta with eclipse events marked with blue squares. Activating the constraint pushes the eclipse to the edge of the eclipse contours, almost out of eclipse season entirely.

6.3   Conclusion

This chapter shows the effectiveness of a penalty method to enforce path inequality constraints at discrete stages in the Sundman-transformed DDP approach. First, apsis constraints made DDP effective in producing the Pareto-front of fuel versus time of flight for the benchmark Q-law Case E
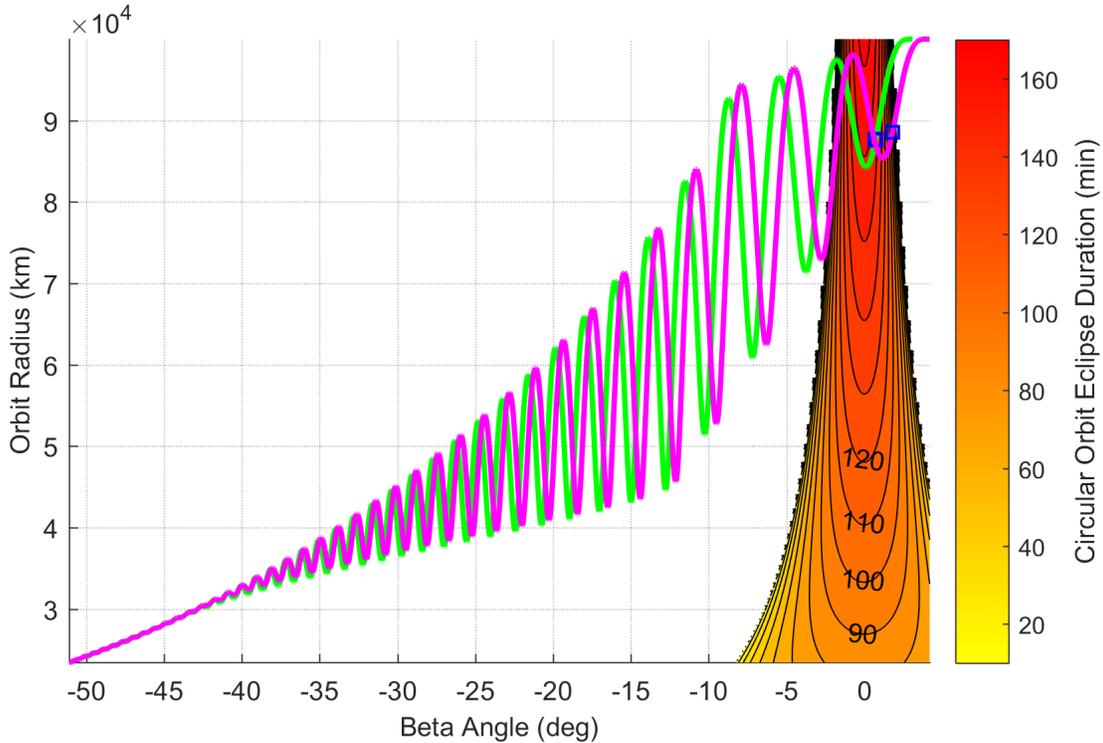
Figure 6.15: An eclipse map of the Mars orbit transfer with a single eclipse of 150.0 minutes that was reduced to $\Delta t_{eclipse,max} = 90$ minutes.

orbit transfer. In general, fixing the number of revolutions for transfers that require large plane changes may lead DDP to solutions with excessive times of flight when the plane change is performed at a large orbital distance. Using the penalty method to constrain the apoapsis radius can effectively constrain DDP to find solutions with times of flight that are practical for real mission design. The second part of this chapter presents a method to constrain the eclipse duration by using a penalty cost assigned to the Keplerian time of flight through a cylindrical shadow. The maximum eclipse duration constraint successfully lowers the encountered eclipse durations of initially unconstrained transfers. Constraining the eclipse duration without first obtaining an unconstrained solution as an initial guess was less successful. This difficulty can in part be attributed to the uncontrollable geometry of the central body and the Sun, whereas the apsis constraints could totally be satisfied by the control decisions. If the mission design allows a variable orbital plane and timing then the respective variables could be selected to mitigate eclipsing or avoid eclipse seasons altogether.

Chapter 7

Differential Dynamic Programming in the Three-Body Problem

This chapter presents the application of DDP to low-thrust trajectory optimization in the three-body problem along with supporting techniques. In Chapter 4, DDP was applied to geocentric transfers with lunar gravity included as a third-body perturbation, and additionally with solar gravity in Chapter 5. SDC has previously been applied to low-thrust escape and capture trajectories with three-body and four-body dynamics [94, 95]. The ability to include n-body perturbations is important for high-fidelity modeling, but preliminary mission design can often afford to sacrifice accuracy in favor of rapid development. Simplified models of the three-body problem facilitate the study of equilibria, periodic orbits, and controlled trajectories through these interesting regions. In Reference [51], Multiple-Shooting Differential Dynamic Programming (MDDP) was used to compute a ballistic periodic orbit and the controls to maintain periodicity of a continuously perturbed trajectory. Both examples were modeled in the CRTBP.

The techniques presented in this chapter are applicable to arbitrary force modeling, but demonstrated with computed examples in the CRTBP model detailed in Chapter 2. Furthermore, the formulations presented should be amenable to other trajectory optimization methods. The DDP algorithm and CRTBP model together provide the tools to begin computing optimal transfers, but enhancements are suggested in the following sections. In order to avoid gravitational singularities and for the practical concern of surface impact, the penalty method is leveraged to enforce minimum-radius constraints. An additional time variable is introduced to propagate a target state forward or backward from an initial state to access any point along a desired orbit as a valid

terminal condition. Finally, these techniques are demonstrated in the Earth-Moon CRTBP with fuel-optimal transfers between distant retrograde orbits (DROs), between planar Lyapunov orbits and between Halo orbits.

7.1   Minimum-Radius Constraint

Trajectory optimization in the CRTBP should avoid the singularities $r_1 = 0$ and $r_2 = 0$. The final solution should have all $r_{1,k}$ and $r_{2,k}$ exceed the radii of the primary and secondary, or exceed a minimum altitude. Discretization should be fine enough to avoid violations between stages k and $k+1$. Constraints for both distances from the primary $r_{1,min}$ and secondary $r_{2,min}$ can be enforced simultaneously.

$$g_{1,k}\left(\boldsymbol{X}_k\right) = r_{1,min} - r_{1,k} \leq 0 \qquad (7.1\text{a})$$

$$g_{2,k}\left(\boldsymbol{X}_k\right) = r_{2,min} - r_{2,k} \leq 0 \qquad (7.1\text{b})$$

Each constraint follows the logic of Equation 6.2 and the stage penalties are summed, e.g. $L_k = L_{1,k} + L_{2,k}$, as are the derivatives.

7.2   Time Variables

Boundary conditions are described in the constraint functions $\boldsymbol{\psi}_i$, so that $\boldsymbol{\psi}_i(\boldsymbol{x}_{i,N_i}, \boldsymbol{x}_{i+1,0}) = \boldsymbol{0}$ for a feasible trajectory. For phase-free Keplerian orbit transfers where any true anomaly is acceptable, $\boldsymbol{\psi}_i$ may be the desired orbit elements. When the target state $\boldsymbol{x}_t$ is a periodic orbit or manifold in the three-body problem, there is not a complete set of constant orbital elements as in the Keplerian case. There is no uniquely valid target, but an entire continuum of acceptable endpoints. These states become accessible by parameterizing the admissible target states $\boldsymbol{x}_t(\tau)$.

$$\boldsymbol{\psi}_{M-1} = \boldsymbol{x}_f(t_f) - \boldsymbol{x}_t(\tau) \qquad (7.2)$$

Subscript f identifies the final state reached at the final stage of the final phase, $\boldsymbol{x}_f = \boldsymbol{x}_{M-1,N_{M-1}}$. There is no mass target in $\boldsymbol{x}_t$, so the mass term is truncated in $\boldsymbol{x}_f$ to take the difference.

The optimization problem now includes solving for the static parameter $w = \tau$, or if the flight time is also allowed to vary, the parameter vector $\boldsymbol{w} = [t_f \; \tau]^T$. Possibilities for the parameterization include defining τ as an angular phase variable [96] or as a time variable [97]. Here a moving target $\boldsymbol{x}_t(\overline{\tau})$ is specified that propagates for the time updates $\delta\tau$ in the unperturbed equations of motion, Equation 2.31 or Equation 2.33 with $T = 0$. The effort required to update $\boldsymbol{x}(\overline{\tau} + \delta\tau)$ can be reduced with a curve fit of the target orbit [96, 97].

As in the example Earth-Mars rendezvous inChapter 3, the time of flight derivatives of the augmented Lagrangian are found by chaining the sensitivity of $\widetilde{\varphi}$ with respect to $\boldsymbol{x} = \boldsymbol{x}_f$ and the time derivative $\dot{\boldsymbol{x}}$. Similarly, the τ-derivatives are formed by chaining the derivatives of $\widetilde{\varphi}$ with respect to $\boldsymbol{x}_t$ and the time derivative $\dot{\boldsymbol{x}}_t$.

$$\widetilde{\varphi}_{w_+} = \begin{bmatrix} \widetilde{\varphi}_{x_+}\dot{\boldsymbol{x}} & \widetilde{\varphi}_{x_t}\dot{\boldsymbol{x}}_t \end{bmatrix}^T \tag{7.3a}$$

$$\widetilde{\varphi}_{w_+ w_+} = \begin{bmatrix} \dot{\boldsymbol{x}}^T\widetilde{\varphi}_{x_+ x_+}\dot{\boldsymbol{x}} + \widetilde{\varphi}_{x_+}\ddot{\boldsymbol{x}} & \dot{\boldsymbol{x}}^T\widetilde{\varphi}_{x_+ x_t}\dot{\boldsymbol{x}}_t \\ \dot{\boldsymbol{x}}_t^T\widetilde{\varphi}_{x_t x_+}\dot{\boldsymbol{x}} & \dot{\boldsymbol{x}}_t^T\widetilde{\varphi}_{x_t x_t}\dot{\boldsymbol{x}}_t + \widetilde{\varphi}_{x_t}\ddot{\boldsymbol{x}}_t \end{bmatrix} \tag{7.3b}$$

$$\widetilde{\varphi}_{x_+ w_+} = \begin{bmatrix} \widetilde{\varphi}_{x_+ x_+}\dot{\boldsymbol{x}} & \widetilde{\varphi}_{x_+ x_t}\dot{\boldsymbol{x}}_t \end{bmatrix} \tag{7.3c}$$

$$\widetilde{\varphi}_{w_+ \lambda_+} = \begin{bmatrix} \widetilde{\varphi}_{\lambda_+ x_+}\dot{\boldsymbol{x}} & \widetilde{\varphi}_{\lambda_+ x_t}\dot{\boldsymbol{x}}_t \end{bmatrix}^T \tag{7.3d}$$

Furthermore, the sensitivity of upstream states to the final time must be captured in the STMs. The effect of the final time is captured in Equation 3.49. There is no effect from τ on upstream states.

7.3   Transfers in the CRTBP

Numerical examples are provided here that apply the preceding DDP techniques to trajectory optimization in the Earth-Moon CRTBP. Constants of the Earth-Moon CRTBP are listed in Table 2.1. Single and two-phase trajectories are presented between DROs, planar Lyapunov orbits and Halo orbits.

The augmented Lagrangian is defined so that the final mass is maximized and the terminal

constraint function is satisfied by arrival at the target Cartesian position and velocity.

$$\varphi = -m_f \tag{7.4a}$$

$$\boldsymbol{\psi} = \boldsymbol{x}_f - \boldsymbol{x}_t \tag{7.4b}$$

$$\Sigma = I_{6\times 6} \tag{7.4c}$$

The two-phase examples are set up so that an initial guess can be constructed from two different trajectories with a state discontinuity at the inter-phase juncture. The parameters of the second phase include the initial state of the second phase, and the initial function is simply

$$\boldsymbol{x}_{1,0} = \boldsymbol{\Gamma}(\boldsymbol{w}_1) = \boldsymbol{w}_1. \tag{7.5}$$

Time variables may also be included in $\boldsymbol{w}_1$, in which case Equation 7.5 would require appropriate indexing to extract either time or state variables. Since $\boldsymbol{x}_+ = \boldsymbol{w}_+$, Equations 3.30 to 3.32 reduce to $\tilde{J} = J$ so that the cost-to-go derivatives with respect to the initial state are not doubly counted. The two-phase augmented Lagrangian includes an inter-phase constraint function that requires continuity in position, velocity and mass.

$$\varphi_1 = -m_f \tag{7.6a}$$

$$\varphi_0 = 0 \tag{7.6b}$$

$$\boldsymbol{\psi}_1 = \boldsymbol{x}_f - \boldsymbol{x}_t \tag{7.6c}$$

$$\boldsymbol{\psi}_0 = \boldsymbol{x}_{0,N_0} - \boldsymbol{x}_{1,0} \tag{7.6d}$$

$$\Sigma_1 = I_{6\times 6} \tag{7.6e}$$

$$\Sigma_0 = I_{7\times 7} \tag{7.6f}$$

Each stage offers an opportunity to update the thrust control variables that are held constant across an integration step. Trust-regions for the spherical control variables are scaled to normalize

the thrust magnitude by its maximum and allow sufficiently large steps in the steering angles.

$$D = diag\left(\frac{1}{T_{max}^2}, \ \frac{1}{\pi}, \ \frac{1}{\pi}\right) \tag{7.7}$$

Trust-regions for the Cartesian control components are uniformly scaled.

$$D = diag\left(\frac{1}{T_{max}^2}, \ \frac{1}{T_{max}^2}, \ \frac{1}{T_{max}^2}\right) \tag{7.8}$$

Trust-regions for multipliers and parameters are not scaled, i.e. D is the identity matrix, unless otherwise noted.

All transfers were computed with fixed-step RK8 integration. Feasibility and optimality tolerances were set to require $\|\psi_i\| < 1 \times 10^{-7}$ and $ER_{0,0} < 1 \times 10^{-8}$, respectively. The examples are small optimization problems in comparison to previous DDP demonstrations that leveraged high-performance computing environments as in Chapters 4 and 5 [57, 61, 62]. Computation was performed on a personal laptop with an Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz with four processors. Runtimes ranged from less than one minute to twelve minutes.

7.3.1   Single-Revolution DRO Transfer

The first example is a single-phase, fixed endpoint and fixed time of flight transfer between DROs. The transfer begins and ends on the x-axis with y decreasing at lunar distances of approximately 70,000 and 120,000 km and periods of 13.4 and 21.6 days. First, the transfer duration was fixed to the average of these periods, 17.5 days, evenly divided among 80 stages and $T_{max} = 0.25$ N was used to produce a one-revolution transfer. Then, the trajectory was recomputed with variable t_f and τ. The updates δt_f were distributed equally among all stages. Optimization began from a ballistic initial guess in both cases. The DRO transfer setup is described in Table 7.1.

The resulting fixed time and variable time, single-revolution DRO transfers are shown in Figure 7.2. The coordinate system has been re-centered on the Moon for plotting. Thrust arcs and vectors are colored orange and coast arcs are colored blue. Markers are placed at the initial and final states on the respective DROs. The transfers begin on the interior DRO and move clockwise, or

Table 7.1: DRO transfer setup.

Orbit	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$	Period (days)
Initial	1.171359	0.0	0.0	0.0	-0.489458	0.0	13.4
Target	1.301844	0.0	0.0	0.0	-0.642177	0.0	21.6

T_{max} (N)	m_0 (kg)	I_{sp} (sec)
0.25, 0.15,	2000	1950
0.05, 0.02		
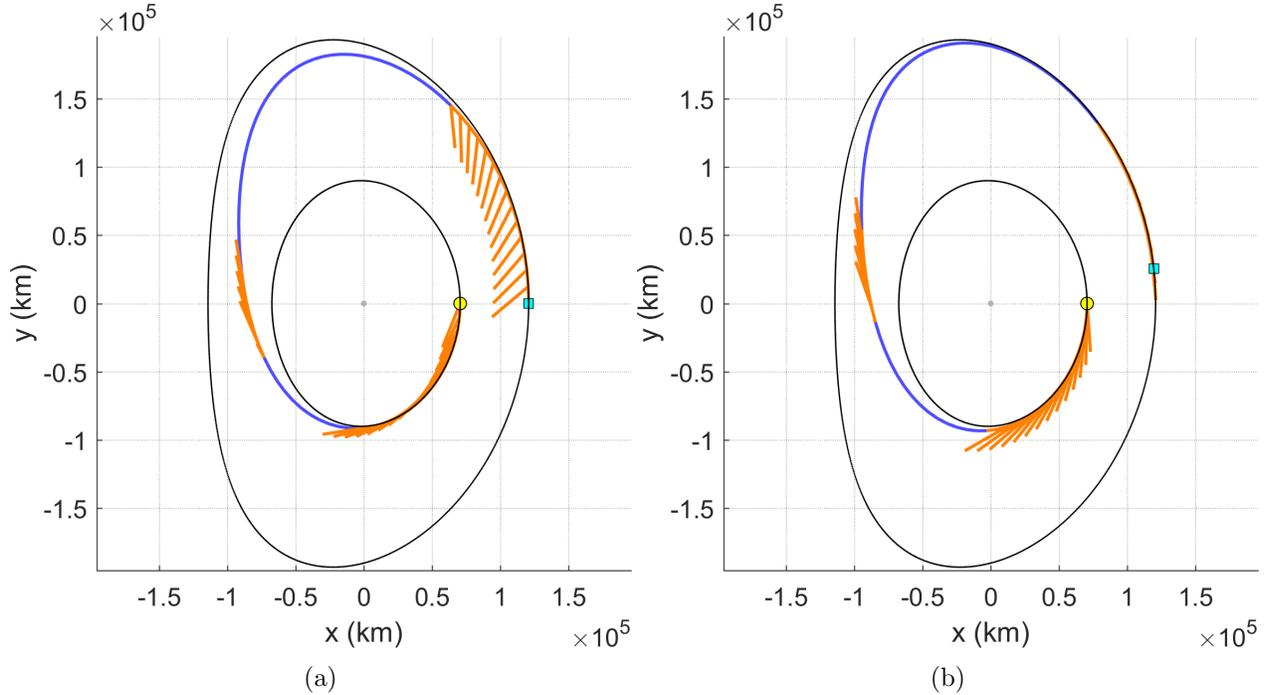
Figure 7.1: Single-revolution DRO transfers with (a) fixed t_f and τ and (b) variable t_f and τ.

retrograde about the Moon, to the larger DRO. A three-burn maneuver structure appears in both cases, but with different rotations of the in-plane control α for the departure and arrival maneuvers. For the variable time case, the second maneuver occurs later, after which the DRO is approached sooner and the arrival maneuver is aligned with the velocity direction. The fixed time of flight transfer reaches the DRO with a final mass of 1991.54 kg. The variable time transfer increases the duration from 17.5 to 17.6 days and increases the delivered mass to 1992.78 kg. Arrival on the DRO is earlier in phase, by $\tau = -0.45$ days. After additional coasting, the original target would

be reached in a total flight time of 18.05 days.

7.3.2 Multi-Revolution DRO Transfers

Multi-revolution transfers were produced by reducing T_{max} as listed in Table 7.1 and increasing the transfer duration. The time of flight and number of stages were set to factors of 2, 5 and 10 times the original values, and produced 2, 5 and 12-revolution transfers. The resulting DRO transfers are illustrated in Figures 7.2a to 7.2c and summarized in Table 7.2. With added flight time, the multi-revolution iterates were slow to depart from a ballistic initial guess. Continuous thrusting in the transverse direction at 10% throttle served as the initial guess for the two-revolution transfer and was increased to 50% throttle for the initial guesses over 5 and 12 revolutions. The converged solutions use sequences of near-tangential burns through the x-axis crossings. Petropoulos [96] found similar thrusting structure for an 11-revolution DRO transfer in the Jupiter-Europa system. The thrust profile of the 12-revolution transfer in Figure 7.2d demonstrates the bang-bang control structure. The multi-revolution transfers exhibit comparable performance to the single-revolution transfer, but this is because maximum thrust was reduced while flight time was increased. If the maximum thrust was instead held fixed as the transfer duration was increased, the deliverable mass would increase and the thrust duration would decrease.

Table 7.2: Fixed time of flight DRO transfer results.

t_f (days)	T_{max} (N)	m_f (kg)
17.5	0.25	1991.54
35.0	0.15	1993.25
87.5	0.05	1993.18
175.0	0.02	1993.04

7.3.3 Multi-Phase Single-Revolution DRO Transfer

The fixed time one-revolution DRO transfer is revisited, but now with the multi-phase DDP formulation. The single-phase setup is simply divided in half, using two phases of 40 stages and
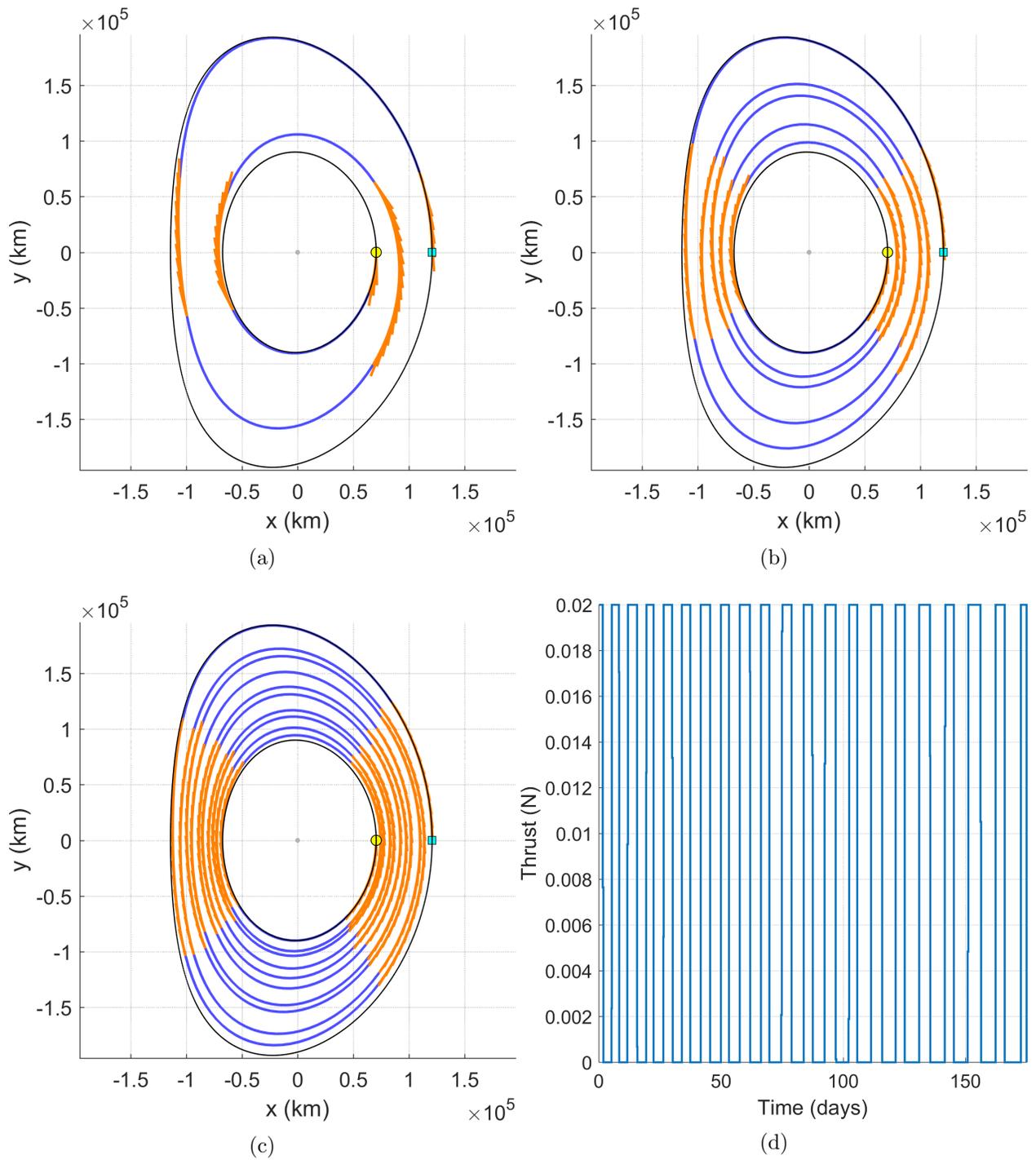
Figure 7.2: DRO transfers spanning (a) 2, (b) 5 and (c) 12 revolutions. (d) Thrust profile for the 12 revolution DRO transfer.

phase flight times of 8.75 days. The initial guess for the first phase is a ballistic arc on the initial DRO from the previous initial conditions. The initial guess for the second phase is a ballistic arc on the target DRO. The initial guess of parameters is $\boldsymbol{w}_1 = \boldsymbol{x}_t$, i.e. the second phase begins on the target state and propagates past it for the entire phase duration. This is purposefully a poor initial guess so that $\boldsymbol{w}_1$ must move substantially. The initial guess for the mass at the start of phase two begins with $m_{1,0} = m_0$. Figure 7.3a shows the initial guess and the first 12 iterations moving the
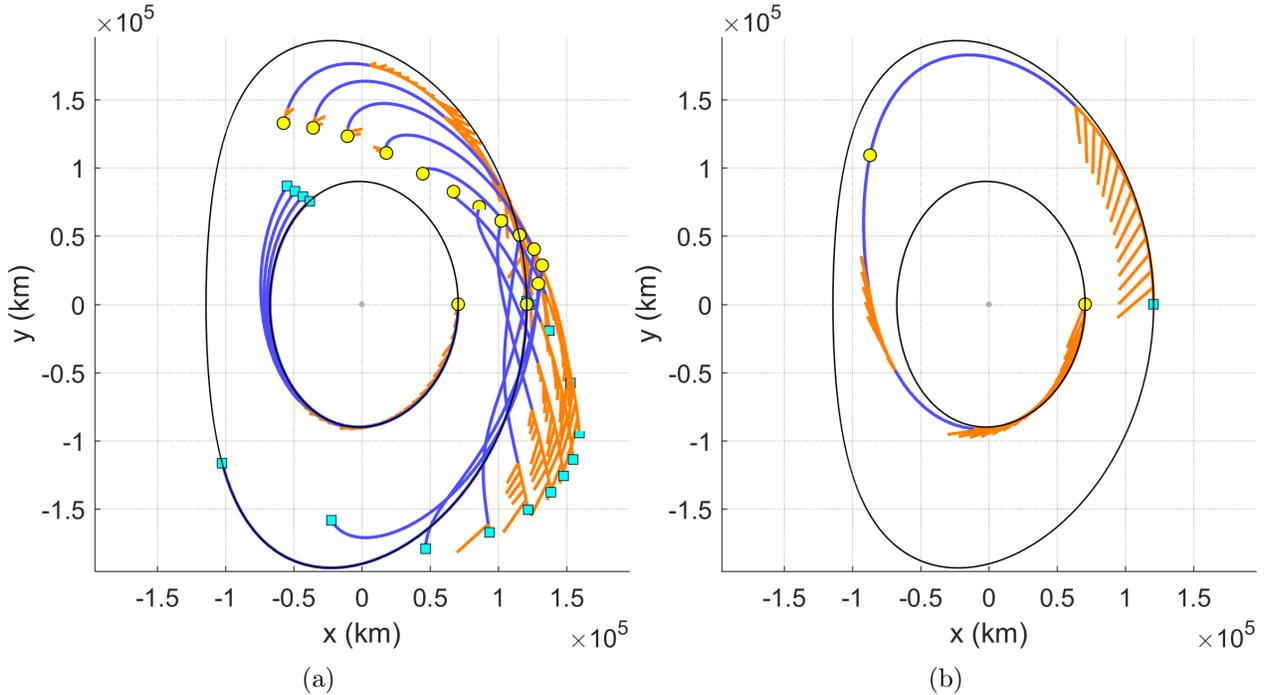
Figure 7.3: (a) Initial guess and first 12 iterations and (b) final solution of the two-phase one-revolution DRO transfer.

start of the second phase through the state space. It wasn't until the tenth iteration that the first phase departed the interior DRO. The final solution is shown in Figure 7.3b. The result is the now familiar transfer structure that matches the single-phase delivered mass of 1991.54 kg.

7.3.4   Planar Lyapunov Heteroclinic Transfer

The L1 and L2 planar Lyapunov orbits listed in Table 7.4 have the same Jacobi constant and share heteroclinic connections that may be used as a ballistic transfer between them. A robust

optimization approach should be able to find such free transfers. Of course, some perturbation is required to depart the initial orbit along its unstable manifold. Arrival along the stable manifold of the target orbit is asymptotic, so a second small maneuver may be used to reach the target orbit in finite time.

Table 7.3: L1 and L2 Lyapunov orbits with the same Jacobi constant.

Orbit	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$	Period (days)	C
L1	0.811446	0.0	0.0	0.0	0.264541	0.0	12.9	3.1241
L2	1.19	0.0	0.0	0.0	-0.234023	0.0	15.2	3.1241

Before tasking DDP to find the low-thrust transfer, a heteroclinic connection was obtained by conventional means [98,99]. Perturbations along the unstable manifold of the L1 Lyapunov orbit were propagated forward to the surface of section defined as the yz-plane centered on the Moon. Perturbations along the stable manifold of the L2 Lyapunov orbit were propagated backward to this surface of section. Intersections of the manifolds with the surface of section produce a Poincaré map. The manifolds and Poincaré map are shown in Figure 7.4.

For this planar Lyapunov example, heteroclinic connections appear as intersections of the stable and unstable manifolds on the Poincaré map. The out of plane components z and $\dot{z}$ are zero everywhere and the manifold x-coordinates are equal on the surface of section. Once y and $\dot{y}$ are matched on the Poincaré map, $\dot{x}$ is also matched because the manifolds have the same Jacobi constant. Figure 7.4 shows two heteroclinic connections that pierce the surface of section at lunar distances of 6,726 and 48,777 km. The higher altitude pass was chosen for an example transfer computed with DDP.

To better exhibit the utility of the DDP approach, the initial guess was distorted from the free transfer itself. The two-phase setup was used with phase one on the L1 Lyapunov orbit and phase two on the L2 Lyapunov orbit as shown in Figure 7.5a. Initial states for each phase are the endpoints of the heteroclinic transfer before perturbations onto each manifold. So, like the multi-phase DRO example, the second phase begins on the very state it is asked to target. Flight times
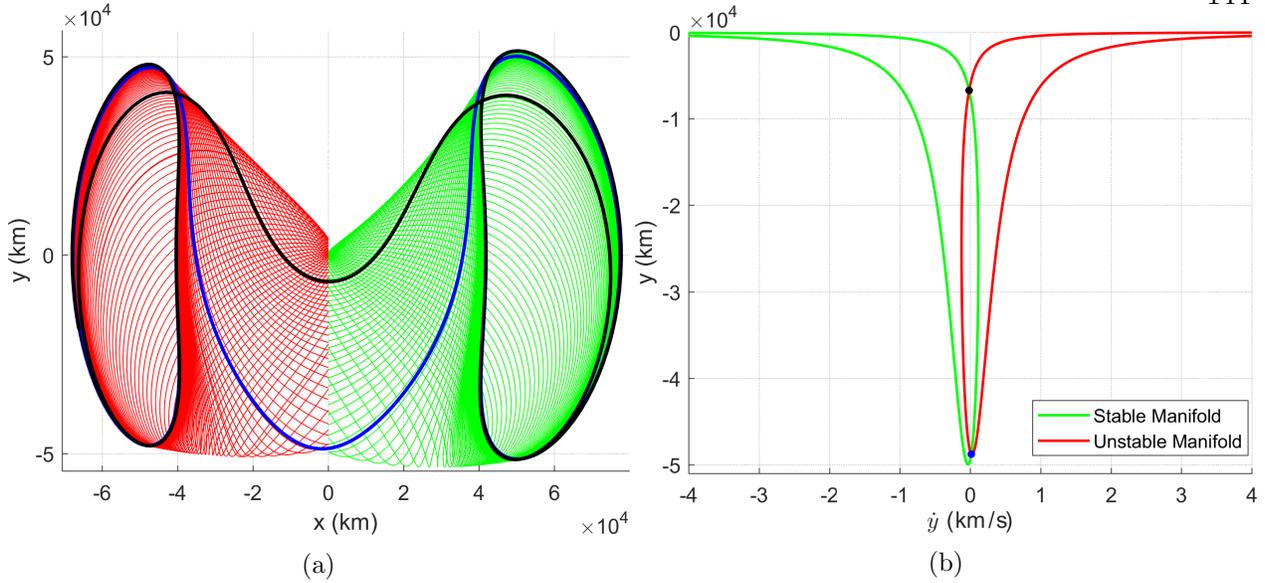
Figure 7.4: (a) L1 Lyapunov unstable manifolds and L2 Lyapunov stable manifolds with two heteroclinic connections highlighted. (b) The Poincaré map shows two intersections of the stable and unstable manifolds on the surface of section.

for each phase Δt_0 and Δt_1 are those spanned from the perturbed states to the surface of section and were divided into 80 stages per phase. Yellow markers in Figure 7.5a indicate the beginning of each phase that spans just under one and a half periods of the respective orbit and terminate with cyan markers. The direction of motion is clockwise.

The position with respect to the Moon $\boldsymbol{r}_2$ and the velocity $\boldsymbol{v}$ align twice per revolution in both the initial and target Lyapunov orbits, so the Cartesian control in Equations 2.4 and 2.9 was selected with $\epsilon_T = 10^{-8}$. As recognized in Reference [64], DDP convergence rates also improve with a larger mass leak parameter ϵ_T. That is likely because very little thrusting occurs in this example, which makes it difficult to track the derivatives of mass flow rate with respect to controls. Accuracy of the solution was improved by restarting the optimization with a reduced $\epsilon_T = 10^{-12}$ and then $\epsilon_T = 10^{-16}$ with each preceding solution as an initial guess. The final trajectory is shown in Figure 7.5b. The nearly free transfer delivers 1999.997 kg in 41.5 days. The transfer is superimposed on the manifold trajectories in Figure 7.5c to show consistency with the heteroclinic connection found in Figure 7.4a. The effect of varying ϵ_T is evident in the thrust profiles that are

Table 7.4: Lyapunov orbit transfer setup.

Orbit	x	y	$\dot{x}$	$\dot{y}$	Period (days)	C
Initial	0.868113	0.1243646	0.080688	-0.022968	12.9	3.1241
Target	1.151358	0.115260	0.107474	-0.116568	15.2	3.1241

T_{max} (N)	m_0 (kg)	I_{sp} (sec)	Δt_0 (days)	Δt_1 (days)
1.5	2000	1950	18.8	22.7

shown in Figure 7.5d. Minimizing the fuel cost simultaneously minimizes the ℓ_1-norm of thrust magnitude and produces a bang-bang control profile. The smaller ϵ_T is more accurate and performs as expected, though the departure and arrival maneuvers are supplemented with two even smaller burns. Fictitiously increasing the thrust magnitude influences a control profile that is characteristic of ℓ_2-norm minimization, or a minimum-energy trajectory, where the thrust magnitude slowly ramps up and down between lower peak magnitudes than the bang-bang control case.

7.3.5   L2 to L1 Halo Orbit Transfer

The next set of examples are transfers from an L2 Halo orbit to an L1 Halo orbit with a different Jacobi constant. Transfers depart and arrive at maximum z-amplitudes and the time of flight is fixed. Initial and final conditions are described in Table 7.5. The spacecraft mass and specific impulse are unchanged from previous examples, but the maximum thrust is increased to 1.5 N. The transfer duration is taken as the average of the Halo orbit periods, 12.7 days, distributed evenly among 120 stages and the initial guess is a ballistic trajectory. First, the distance from the Moon was unconstrained. Then, a minimum radius $r_{2,min} = 20,000$ km was enforced with a weight $c = 1000$ on the quadratic penalty, and subsequently increased to 25,000, 30,000 and 35,000 km, with preceding solutions used as the initial guess. The unconstrained transfer is shown in Figure 7.6. Distance from the Moon, r_2, is shown for all of the transfers in Figure 7.7a. Planar views of the radius-constrained Halo orbit transfers are shown in Figures 7.7b to 7.7d. Thrust vectors have been removed for clarity. The actual minimum radius and the final mass for each transfer is listed in Table 7.6. When the constraint is first enacted, DDP reaches an improved local
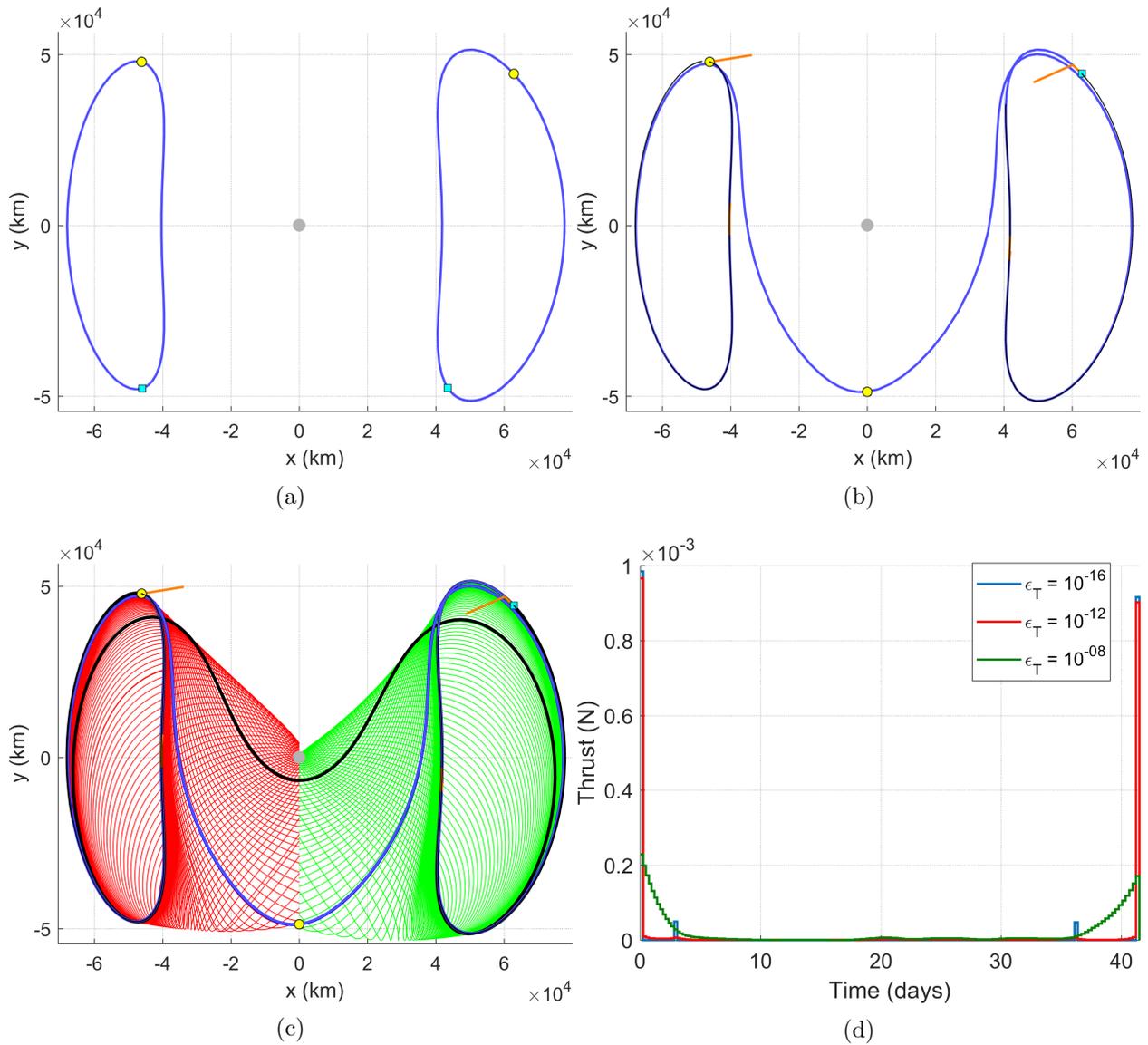
Figure 7.5: (a) Initial guess for the two-phase Lyapunov orbit transfer. (b) The converged hetero-clinic transfer. (c) The transfer is superimposed on manifold trajectories to the surface of section. (d) The thrust profile shows small maneuvers to complete the nearly ballistic transfer and the effect of different sized mass leaks.

Table 7.5: Halo orbit transfer setup.

Orbit	x	y	z	$\dot{x}$	$\dot{y}$	$\dot{z}$	Period (days)	C
Initial	1.16	0.0	-0.122697	0.0	-0.207128	0.0	14.2	3.0931
Target	0.85	0.0	0.173890	0.0	0.262114	0.0	11.2	3.0090

T_{max} (N)	m_0 (kg)	I_{sp} (sec)
1.5	2000	1950

minimum. A close lunar flyby aids the transfer, so the required propellant increases accordingly when the minimum radius is increased further. Initially, the transfer requires arrival and departure maneuvers and additional maneuvers on either side of the flyby. There are again flybys at 20,000 and 25,000 km minimum radii, but above that the transfers move along the constraint boundary for an extended duration. Lastly, with the minimum radius increased to 35,000 km, the transfer is approaching the need to thrust for the entire duration, and increasing the constraint further would soon render the transfer infeasible for the given flight time. The actual minimum radii demonstrate the trade-off made with a penalty method, where some violation of the constraint is allowed in favor of improving the objective. Larger violations were found as the radius constraint became more demanding, The most aggressive constraint of 35,000 km was too aggressive, so a lower weight of $c = 100$ was used to reach convergence but permitted a large constraint violation.

Table 7.6: Halo orbit transfer results for different lunar distance constraints.

$r_{2,min}$ (km)	c	$\min(r_{2,k})$ (km)	m_f (kg)
None	None	18,808	1964.25
20,000	1000	19,996	1964.87
25,000	1000	24,985	1962.31
30,000	1000	29,976	1956.59
35,000	100	33,733	1923.28
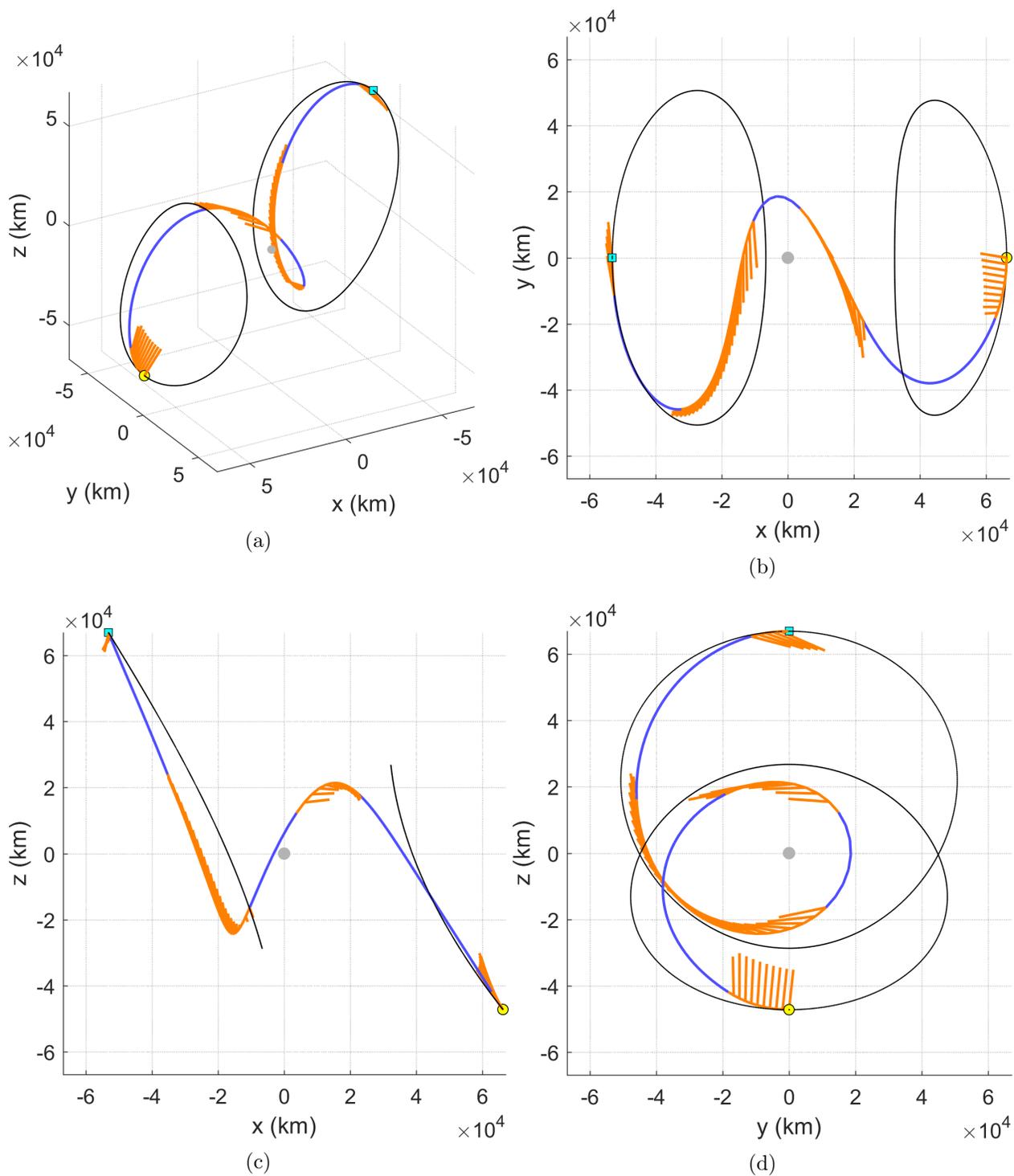
Figure 7.6: (a) Three-dimensional view and (b-d) planar projections of the Halo orbit transfer without a minimum-radius constraint.
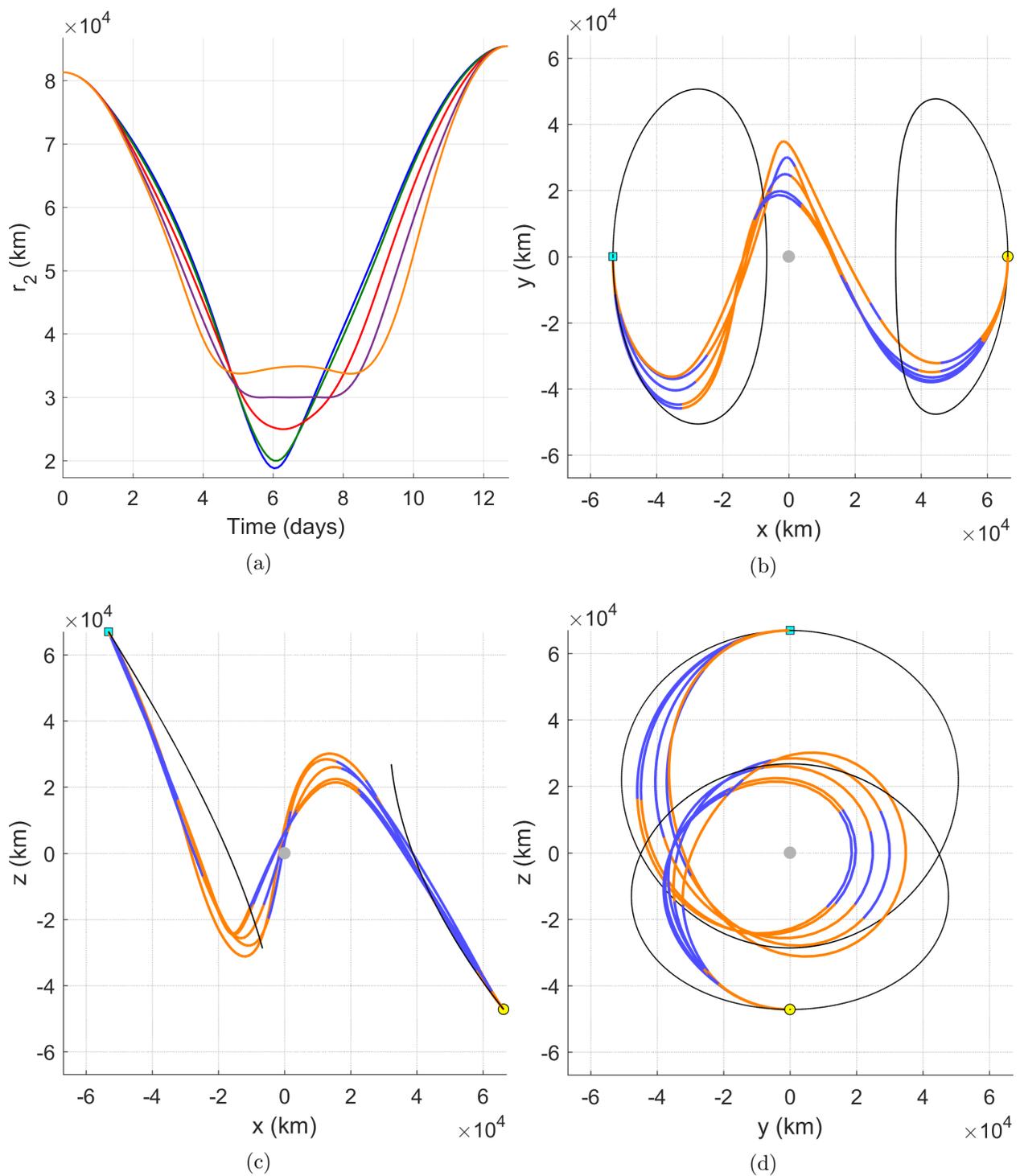
Figure 7.7: (a) Distance from the Moon for each of the radius-constrained Halo orbit transfers. (b-d) Planar projections of Halo orbit transfers with increasing minimum radius from the Moon.

7.3.6   Multi-Phase L2 to L1 Halo Orbit Transfer

The final example demonstrates the use of time variables in the multi-phase formulation. The Halo orbit transfer was recomputed with double the flight time and number of stages but now divided into two phases. Each phase is comprised of 120 stages and each phase takes half of the total flight time. The initial guess for the first phase is a ballistic segment on the L2 Halo orbit and the initial guess for the second phase is a ballistic segment on the L1 Halo with $\boldsymbol{x}_{1,0} = \boldsymbol{x}_t$. Parameters are the times of flight for each phase, τ for the second phase and the initial state for the second phase.

$$w_0 = t_{0,N_0} \tag{7.9a}$$

$$\boldsymbol{w}_1 = \begin{bmatrix} t_f & \tau & \boldsymbol{x}_{1,0}^T \end{bmatrix}^T \tag{7.9b}$$

Trust-region scaling for the mass parameter, i.e. the corresponding entry on the diagonal of D, was reduced by an order of magnitude. This facilitated convergence by allowing larger steps to update $m_{1,0}$ at each iteration.

Figure 7.8a shows the initial guess for the two-phase Halo orbit transfer and Figure 7.8b shows the converged solution. The distance from the Moon and planar projections are shown in Figure 7.9. The first phase resembles the single-phase Halo orbit transfer with but uses a higher altitude flyby and reaches a lower z-amplitude. The phase discontinuity is resolved during a coast arc near this apex. Phase two is then another revolution that approaches the target Halo orbit with three more maneuvers. Insertion onto the target Halo orbit occurs later than the fixed time example. The time variables moved by $\delta t_{0,N_0} = 1.7$ days, $\delta t_f = -0.1$ days and $\delta \tau = 0.8$ days. A final mass of 1970.62 kg reaches the target Halo orbit in 27.0 days, which expectedly improves upon the shorter transfer.

7.4   Conclusion

DDP has been presented as a utility for trajectory optimization in the circular restricted three-body problem. Important considerations for transfer design in the CRTBP are how to allow
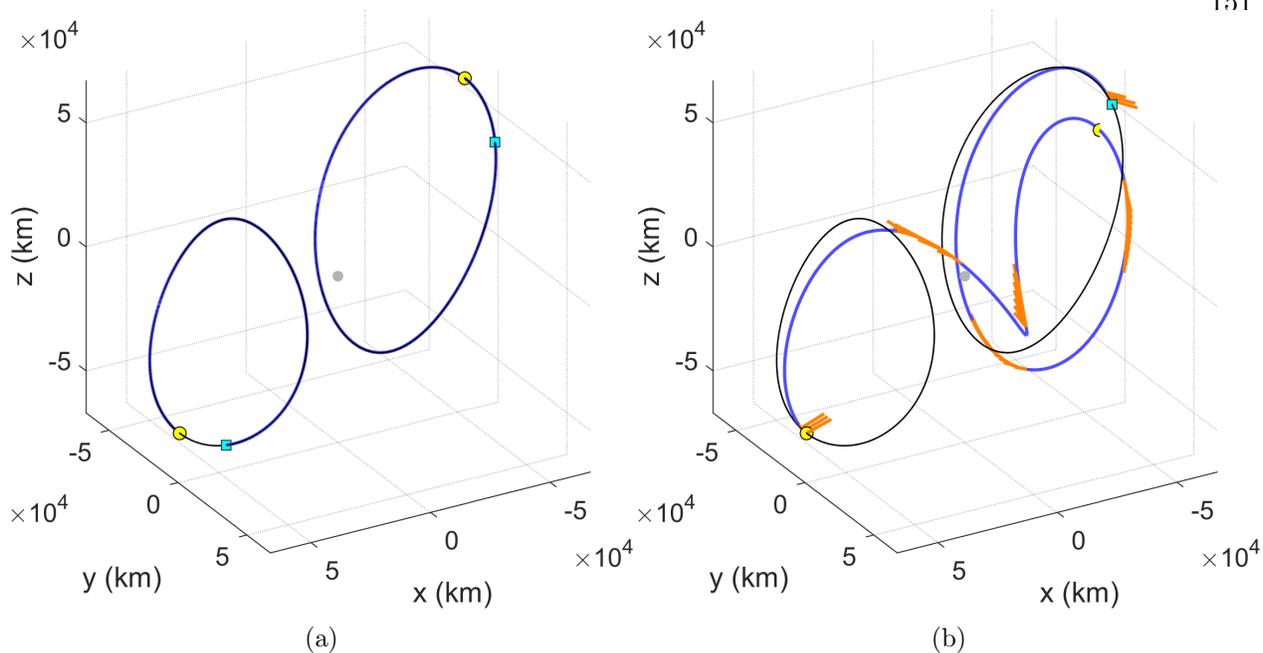
Figure 7.8: (a) Initial guess and (b) final solution of the two-phase Halo orbit transfer.

terminal constraints to accept any valid state on a target orbit, and constraints on the minimum radii from the massive bodies. A moving target can relax the terminal constraint of a single point on a target orbit and a penalty method can enforce path inequality constraints. These techniques were applied to both single and multi-phase transfers between distant retrograde orbits, between planar Lyapunov orbits and between Halo orbits. The multi-phase DDP formulation proved effective for transfer design between periodic orbits in the vicinity of L1 and L2 and successfully leveraged the system dynamics to complete a heteroclinic transfer.
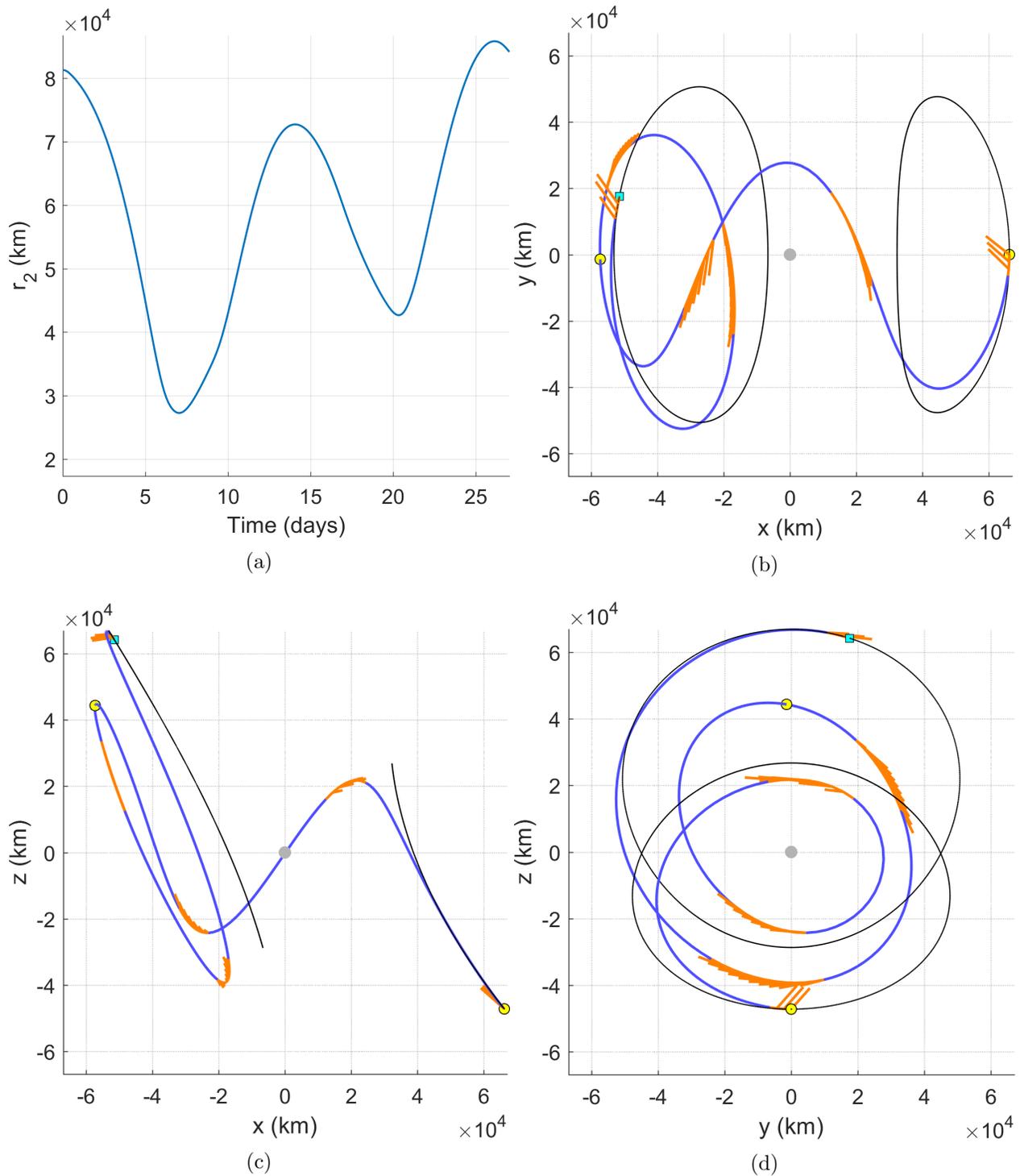
Figure 7.9: (a) The distance from the Moon and (b-d) planar projections of the two-phase Halo orbit transfer.

Chapter 8

Conclusion

This dissertation offers a solution to the low-thrust many-revolution trajectory optimiza-
tion problem. That solution is the pairing of differential dynamic programming and the Sundman
transformation, which has been most effective by using modified equinoctial elements to represent
the spacecraft state. A second significant contribution is the treatment of solar eclipsing in the
trajectory optimization process, which includes both the effect on power available to solar powe-
red spacecraft and the capability to constrain the duration of an eclipse passage. Finally, this
dissertation extends the reach of DDP in multi-body dynamic environments, specifically through
application in the circular restricted three-body problem.

8.1   Summary

Early research efforts invested in the DDP algorithm as an approach to low-thrust many-
revolution trajectory optimization. Inherent bottlenecks associated with control scheduling and
trajectory computation in the time domain were relieved by applying a Sundman transformation
to the spacecraft equations of motion. Sundman transformations to each of the true, mean and
eccentric anomalies permitted the computation of fuel-optimal GTO to GEO transfers in reasonable
runtimes that were frequently less than one hour. Choosing the true anomaly as the independent
variable and the modified equinoctial elements to represent the spacecraft state has been most
effective for example geocentric transfers. Having to specify the number of revolutions can be
a limitation, although the method is efficient enough run a parametric sweep over the number of

revolutions with modest computational resources. Example parametric sweeps illustrated the trade-off of fuel versus time of flight for the Q-law Case B and Case E orbit transfers with performance curves that compete with the benchmark results.

After developing a solution method for low-thrust many-revolution trajectory optimization, the research focus turned toward resolving the challenges of solar eclipsing for this class of orbit transfers. With DDP as the algorithm of choice, a twice-differentiable power model was required. The logistic sunlight fraction was designed to smooth the discontinuity in power available through eclipse and enabled the optimization of a reference LEO to GEO transfer. A parametric study resulted in time-optimal and fuel-optimal curves that are connected by curves of constant number of revolutions. This topology enables a systematic approach to sketching out the Pareto front of fuel versus time of flight. A second mission design concern for solar eclipsing is how to constrain the maximum duration of an eclipse. This complicated constraint was handled by assigning a penalty cost whenever the constraint was violated. As before, a twice-differentiable constraint function was warranted. This was attained by assuming a cylindrical shadow model and Keplerian dynamics through eclipse and was demonstrated by reductions in eclipse durations for orbit transfers about Mars. The penalty method was further utilized for geocentric apsis constraints and lunar distance constraints in the CRTBP.

Later pursuits sought to extend the applicability of DDP to multi-body dynamic models. Transfers in the CRTBP were demonstrated between DROs, Lyapunov orbits and Halo orbits, where DDP has only seen limited attention in the past. The CRTBP served as testbed for variable time of flight, moving target and multi-phase formulations, but these techniques may be applied to other dynamic models. Tasking DDP to compute a heteroclinic connection between Lyapunov orbits served to verify the approach.

8.2   Future Work

A straightforward next step is to attempt orbit transfers with different objectives, dynamics and constraints, or as otherwise stated, to practice mission design with the methods described

in this dissertation. Each example used a fixed-I_{sp} thruster with T_{max} always accessible (except for in eclipse), the initial mass was always fixed and the objective was either to maximize the delivered mass or minimize the time of flight. High-fidelity thruster and power models may instead be used, so long as the necessary derivatives can be supplied to DDP. The initial mass may be activated as a design variable in which case maximizing the delivered mass and minimizing the propellant are different objectives. Arbitrary objectives may be stated, but again require the necessary derivatives. Force modeling stopped at the J_2, J_3 and J_4 spherical harmonic terms and third-body perturbations, but may be extended to high-order gravity fields, drag, solar radiation pressure and additional perturbing bodies, for example. Low-thrust propulsion is not a restriction and may be combined or replaced with high-thrust options or solar sailing. Initial and target orbit selection is another open area for research given that the improved capabilities for low-thrust many-revolution trajectory optimization may open up entirely new mission concepts.

The Sundman transformation is a general change of variable from time to a function of radius. The true, mean and eccentric anomalies were investigated in this dissertation because they are fundamental. Other transformations may be more useful and that is expected to be problem dependent. For example, transfers in the circular restricted three-body problem may leverage $dt = r_1 d\tau$, $dt = r_2 d\tau$ or even a Hybrid Sundman transformation $dt = r_1 r_2 d\tau$ [51]. The elliptic anomaly is another useful independent variable for orbit propagation, with $n = 3/2$ in Equation 4.1 [72].

While the number of revolutions was left fixed for the provided examples, it can be allowed to vary by using Equations 3.48 to 3.50 but by taking derivatives with respect to the Sundman-transformed independent variable, rather than time. In practice, enabling a variable N_{rev} usually resulted in transfers that only deviated within ± 1 revolution from the initial guess, owed to the nature of local optimization. Thus, it was more effective to leave N_{rev} fixed and vary it discretely across the design space. Developing a way to allow N_{rev} to grow or shrink by large amounts that are well-informed and with a high success rate would be another pioneering step in low-thrust many-revolution trajectory optimization.

Parallelization proved to be critical for the efficiency of this approach but that has yet to be exploited beyond more than two multi-core processors. At best, STMs were computed simultaneously between 24 stages but the problem size extended up to 200,000 stages. An increased scale of parallelization using more processors and a message-passing interface (MPI) [100] or graphics processing units (GPU) is worth pursuing for gains in computational efficiency.

The demonstrated capability to include solar eclipse effects made use of simplified eclipse models. A significant effort resulted in useful examples but less effort was expended on trying to break the models and determine the realm of applicability. The sharpness and transition coefficients for the smoothed eclipse model present an open area of study for the stated trade-offs of accuracy, optimizer performance and spacecraft operation. The maximum eclipse duration constraint was exhibited with the initial orbit plane and epoch specified, so that there would actually be eclipses to constrain. However, the initial conditions could be activated as decision variables to satisfy the eclipse constraint. Furthermore, the maximum eclipse duration of a given transfer can become a performance metric to weigh against other objectives and add a dimension to the Pareto surface of favorable trajectories.

The capability of DDP to exploit multi-body dynamic models is promising. In the circular-restricted three-body problem, open areas include transfers between different structures, e.g. DRO to Halo orbit transfers rather than DRO to DRO or Halo to Halo. Many-revolution transfers in the CRTBP could be pursued, for which multi-phase or many-phase DDP should prove useful, rather than simply using two phases as was demonstrated. The associated techniques could be applied to a full-ephemeris model. A prospective exercise includes multi-phase and multi-body Sundman-transformed escape and capture trajectories, where each escape and capture phase is a many-revolution trajectory.

Bibliography

[1] European GNSS Agency. European GNSS Service Centre, Orbital and Technical Parameters. https://www.gsc-europa.eu/system-status/orbital-and-technical-parameters. Accessed February 2018.

[2] L. D. Kos, T. Polsgrove, R. C. Hopkins, D. Thomas, and J. A. Sims. Overview of the Development for a Suite of Low-Thrust Trajectory Analysis Tools. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, August 2006.

[3] NASA. NASA Technology Roadmaps TA 5: Communications, Navigation, and Orbital Debris Tracking and Characterization Systems. http://www.nasa.gov/sites/default/files/atoms/files/2015_nasa_technology_roadmaps_ta_5_communication_and_navigation_final.pdf, July 2015. Accessed February 2018.

[4] R. R. Bate, D. D. Mueller, and J. E. White. Fundamentals of Astrodynamics. Dover Publications, Inc., New York, 1971.

[5] D. A. Vallado. Fundamentals of Astrodynamics and Applications. 4th Edition, Microcosm Press and Springer, Hawthorne, CA, 2013.

[6] A. E. Bryson Jr. and Y. C. Ho. Applied Optimal Control. Ginn and Company, Waltham, MA, 1969.

[7] J. M. Longuski, J. Guzmán, and J. E. Prussing. Optimal Control with Aerospace Applications. Microcosm Press and Springer, El Segundo, CA, 2014.

[8] T. N. Edelbaum. Propulsion Requirements for Controllable Satellites. ARS Journal, 31:1079–1089, August 1961.

[9] T. N. Edelbaum. Theory of Maxima and Minima. Optimization Techniques, With Applications to Aerospace Systems, 1962.

[10] W. E. Wiesel and S. Alfano. Optimal Many-Revolution Orbit Transfer. Journal of Guidance, Control, and Dynamics, 8:155–157, 1985.

[11] J. A. Kéchichian. Reformulation of Edelbaum's Low-Thrust Transfer Problem Using Optimal Control Theory. Journal of Guidance, Control, and Dynamics, 20:988–994, September 1997.

[12] T. N. Edelbaum. Optimum Low-Thrust Rendezvous and Station Keeping. AIAA Journal, 2(7):1196–1201, 1964.

[13] J. A. Kéchichian. Optimal Low-Thrust Rendezvous Using Equinoctial Orbit Elements. <u>Acta Astronautica</u>, 38(1):1–14, 1996.

[14] J. A. Kéchichian. Optimal Low-Thrust Transfer in General Circular Orbit Using Analytic Averaging of the System Dynamics. <u>The Journal of the Astronautical Sciences</u>, 57(1-2):369–392, January-June 2009.

[15] J. A. Kéchichian. Inclusion of Higher Order Harmonics in the Modeling of Optimal Low-Thrust Orbit Transfer. <u>The Journal of the Astronautical Sciences</u>, 56(1):41–70, January-March 2008.

[16] J. A. Kéchichian. Orbit Raising with Low-Thrust Tangential Acceleration in Presence of Earth Shadow. <u>Journal of Spacecraft and Rockets</u>, 35(4):516–525, July-August 1998.

[17] J. A. Kéchichian. Low-Thrust Eccentricity-Constrained Orbit Raising. <u>Journal of Spacecraft and Rockets</u>, 35(3):327–335, May-June 1998.

[18] J. A. Kéchichian. Low-Thrust Inclination Control in Presence of Earth Shadow. <u>Journal of Spacecraft and Rockets</u>, 35(4):526–532, July-August 1998.

[19] C. A. Kluever. Simple Guidance Scheme for Low-Thrust Orbit Transfers. <u>Journal of Guidance, Control, and Dynamics</u>, 21:1015–1017, November 1998.

[20] D. E. Chang, D. F. Chichka, and J. E. Marsden. Lyapunov-Based Transfer Between Elliptic Keplerian Orbits. <u>Discrete and Continuous Dynamical Systems-Series B</u>, 2(1):57–67, February 2002.

[21] A. E. Petropoulos. Simple Control Laws for Low-Thrust Orbit Transfers. <u>AAS/AIAA Astrodynamics Specialist Conference</u>, Big Sky, MT, August 2003.

[22] A. E. Petropoulos. Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting. <u>AIAA/AAS Astrodynamics Specialist Conference and Exhibit</u>, Providence, RI, August 2004.

[23] A. E. Petropoulos and R. P. Russell. Refinements to the Q-law for low-thrust orbit transfers. <u>AAS/AIAA Space Flight Mechanics Conference</u>, Copper Mountain, CO, January 2005.

[24] A. E. Petropoulos, Z. B. Tarzi, G. Lantoine, T. Dargent, and R. Epenoy. Techniques For Designing Many-Revolution, Electric Propulsion Trajectories. <u>AAS/AIAA Space Flight Mechanics Meeting</u>, Santa Fe, NM, January 2014.

[25] R. D. Falck, W. K. Sjauw, and D. A. Smith. Comparison of Low-Thrust Control Laws for Application in Planetocentric Space. <u>50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA Propulsion and Energy Forum</u>, Cleveland, OH, July 2014.

[26] S. Lee, P. von Allmen, W. Fink, A. E. Petropoulos, and R. J. Terrile. Design and Optimization of Low-thrust Orbit Transfers. <u>IEEE Aerospace Conference</u>, Big Sky, MT, March 2005.

[27] C. A. Kluever and S. R. Oleson. Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers. <u>Journal of Spacecraft and Rockets</u>, 35:509–515, July-August 1998.

[28] J. T. Betts. Practical Methods for Optimal Control and Estimation using Nonlinear Programming. 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2010.

[29] W. A. Scheel and B. A. Conway. Optimization of Very-Low-Thrust, Many-Revolution Spacecraft Trajectories. Journal of Guidance, Control, and Dynamics, 17(6):1185–1191, November-December 1994.

[30] J. T. Betts. Trajectory Optimization Using Sparse Sequential Quadratic Programming. In R. Bulirsch, A. Miele, J. Stoer, and K. Well, editors, Optimal control, International Series of Numerical Mathematics, volume 117. Birkhäuser Basel, 1993.

[31] J. T. Betts. Sparse Optimization Suite, SOS, User's Guide, Release 2015.11. http://www.appliedmathematicalanalysis.com/downloads/sosdoc.pdf. Accessed November 2016.

[32] J. T. Betts. Very Low-Thrust Trajectory Optimization Using a Direct SQP Method. Journal of Computational and Applied Mathematics, 120:27–40, August 2000.

[33] J. T. Betts and S. O. Erb. Optimal Low Thrust Trajectories to the Moon. SIAM Journal on Applied Dynamical Systems, 2(2):144–170, May 2003.

[34] J. T. Betts. Optimal Low Thrust Orbit Transfers with Eclipsing. Optimal Control Applications and Methods, 36:218–240, February 2015.

[35] D. H. Jacobson and D. Q. Mayne. Differential Dynamic Programming. American Elsevier Publishing Company, Inc., New York, NY, 1970.

[36] R. E. Bellman. Dynamic Programming. Princeton University Press, Princeton, NJ, 1957.

[37] D. Q. Mayne. A Second-order Gradient Method for Determining Optimal Trajectories of Non-linear Discrete-time Systems. International Journal of Control, 3:85–95, 1966.

[38] S. B. Gershwin and D. H. Jacobson. A Discrete-Time Differential Dynamic Programming Algorithm With Application to Optimal Orbit Transfer. AIAA Journal, 8(9):1616–1626, 1970.

[39] G. J. Whiffen. Static/Dynamic Control for Optimizing a Useful Objective. United States Patent No. 6496741, December 2002.

[40] G. J. Whiffen. Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, August 2006.

[41] NASA Technology Transfer Program. Mystic Low-Thrust Trajectory Design and Visualization Software. https://software.nasa.gov/software/NPO-43666-1. Accessed October 2016.

[42] M. D. Rayman, T. C. Fraschetti, C. A. Raymond, and C. T. Russell. Dawn: A mission in development for exploration of main belt asteroids Vesta and Ceres. Acta Astronautica, 58:605–616, April 2006.

[43] C. Colombo, M. Vasile, and G. Radice. Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming. Celestial Mechanics and Dynamical Astronomy, 105(75):75–112, 2009.

[44] G. Lantoine and R. P. Russell. A Hybrid Differential Dynamic Programming Algorithm for Robust Low-Thrust Optimization. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, August 2008.

[45] G. Lantoine and R. P. Russell. A Fast Second-Order Algorithm for Preliminary Design of Low-Thrust Trajectories. 59th International Astronautical Congress, Glasgow, Scotland, September 2008.

[46] G. Lantoine and R. P. Russell. A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory. Journal of Optimization Theory and Applications, 154(2):382–417, 2012.

[47] G. Lantoine and R. P. Russell. A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application. Journal of Optimization Theory and Applications, 154(2):418–442, 2012.

[48] G. Lantoine. A Methodology for Robust Optimization of Low-Thrust Trajectories in Multi-Body Environments. PhD thesis, Georgia Institute of Technology, 2010.

[49] E. Pellegrini and R. P. Russell. Quasi-Newton Differential Dynamic Programming for Robust Low-Thrust Optimization. AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, August 2012.

[50] E. Pellegrini and R. P. Russell. A Multiple-Shooting Differential Dynamic Programming Algorithm. AAS/AIAA Space Flight Mechanics Meeting, San Antonio, TX, February 2017.

[51] E. Pellegrini and R. P. Russell. Applications of the Multiple-Shooting Differential Dynamic Programming Algorithm with Path and Terminal Constraints. AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, August 2017.

[52] N. Ozaki, S. Campagnola, C. H. Yam, and R. Funase. Differential Dynamic Programming Approach for Robust-Optimal Low-Thrust Trajectory Design Considering Uncertainty. 25th International Symposium on Space Flight Dynamics, Munich, Germany, October 2015.

[53] N. Ozaki, S. Campagnola, R. Funase, and C. H. Yam. Stochastic Differential Dynamic Programming with Unscented Transform for Low-Thrust Trajectory Design. Journal of Guidance, Control, and Dynamics, 41(18), 2018.

[54] N. Ozaki and R. Funase. Tube Stochastic Differential Dynamic Programming for Robust Low-Thrust Trajectory Optimization Problems. AIAA Guidance, Navigation, and Control Conference, Kissimmee FL, January 2018.

[55] N. Ozaki, R. Funase, S. Campagnola, and C. H. Yam. Robust-Optimal Trajectory Design against Disturbance for Solar Sailing Spacecraft. AIAA/AAS Astrodynamics Specialist Conference, Long Beach CA, September 2016.

[56] J. T. Betts. Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance, Control, and Dynamics, 21(2):193–207, March-April 1998.

[57] J. D. Aziz, J. S. Parker, D. J. Scheeres, and J. A. Englander. Low-Thrust Many-Revolution Trajectory Optimization via Differential Dynamic Programming and a Sundman Transformation. The Journal of the Astronautical Sciences, January 2018.

[58] J. D. Aziz, D. J. Scheeres, and G. Lantoine. Differential Dynamic Programming in the Three-Body Problem. AIAA/AAS Space Flight Mechanics Meeting, Kissimmee, FL, January 2018.

[59] J. D. Aziz, J. S. Parker, and J. A. Englander. Hybrid Differential Dynamic Programming With Stochastic Search. AAS/AIAA Space Flight Mechanics Meeting, Napa, CA, February 2016.

[60] J. D. Aziz, J. S. Parker, D. J. Scheeres, and J. A. Englander. Low-Thrust Many-Revolution Trajectory Optimization via Differential Dynamic Programming and a Sundman Transformation. AAS/AIAA Space Flight Mechanics Meeting, San Antonio, TX, February 2017.

[61] J. D. Aziz and D. J. Scheeres. Improvements to Sundman-Transformed HDDP Through Modified Equinoctial Elements. AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, August 2017.

[62] J. D. Aziz, D. J. Scheeres, J. S. Parker, and J. A. Englander. A Smoothed Eclipse Model for Solar Electric Propulsion Trajectory Optimization. 26th International Symposium on Space Flight Dynamics, Matsuyama, Japan, June 2017.

[63] J. M. Knittel, J. A. Englander, M. T. Ozimek, J. A. Atchison, and J. J. Gould. Improved Propulsion Modeling for Low-Thrust Trajectory Optimization. AAS/AIAA Space Flight Mechanics Meeting, San Antonio, TX, February 2017.

[64] T. T. McConaghy and J. M. Longuski. Parameterization Effects on Convergence when Optimizing a Low-Thrust Trajectory with Gravity Assists. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Providence, RI, August 2004.

[65] M. J. H. Walker, B. Ireland, and J. Owens. A Set of Modified Equinoctial Elements. Celestial Mechanics, 36:409–419, August 1985.

[66] K. Sundman. Memoire sur le probleme des trois corps. Acta Math, 36:105–179, 1913.

[67] A. R. Conn, N. I. M. Gould, and P. L. Toint. Trust-Region Methods. Mathematical Programming Society and the Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[68] J. A. Englander, D. H. Ellison, and B. A. Conway. Global Optimization of Low-Thrust Multiple-Flyby Trajectories at Medium and Medium-High Fidelity. AAS/AIAA Space Flight Mechanics Meeting, Sante Fe, NM, January 2014.

[69] P. Prince and J. Dormand. High order embedded Runge-Kutta formulae. Journal of Computational and Applied Mathematics, 7, Issue 1:67–75, March 1981.

[70] C. H. Yam, D. D. Lorenzo, and D. Izzo. Low-thrust trajectory design as a constrained global optimization problem. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 225:1243–1251, November 2011.

[71] J. A. Englander and A. C. Englander. Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization. 24th International Symposium on Space Flight Dynamics, Laurel, MD, May 2014.

[72] G. Janin and V. R. Bond. The Elliptic Anomaly. NASA Technical Memorandum 58228, April 1980.

[73] E. Pellegrini and R. P. Russell. On the Computation and Accuracy of Trajectory State Transition Matrices. Journal of Guidance, Control, and Dynamics, 39(11):2485–2499, November 2016.

[74] M. Berry and L. Healy. The generalized Sundman transformation for propagation of high-eccentricity elliptical orbits. AAS/AIAA Space Flight Mechanics Meeting, San Antonio, TX, January 2002.

[75] E. Pellegrini, R. P. Russell, and V. Vittaldev. F and G Taylor Series Solutions to the Stark and Kepler problems with Sundman Transformations. Celestial Mechanics and Dynamical Astronomy, 118:355–378, April 2014.

[76] C. H. Yam, D. D. Lorenzo, and D. Izzo. Towards a High Fidelity Direct Transcription Method for Optimisation of Low-Thrust Trajectories. 4th International Conference on Astrodynamics Tools and Techniques - ICATT, Madrid, Spain, May 2010.

[77] J. A. Sims and S. N. Flanagan. Preliminary Design of Low-Thrust Interplanetary Missions. AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, August 1999.

[78] J. Anderson, P. J. Burns, D. Milroy, P. Ruprecht, T. Hauser, and H. J. Siegel. Deploying RMACC Summit: An HPC Resource for the Rocky Mountain Region. PEARC17, July 2017.

[79] L. Dagum and R. Menon. OpenMP: An Industry-Standard API for Shared-Memory Programming. IEEE Computational Science and Engineering, 5:46–55, January 1998.

[80] J. A. Englander, M. A. Vavrina, and D. Hinckley. Global Optimization of Low-Thrust Interplanetary Trajectories Subject to Operational Constraints. AAS/AIAA Space Flight Mechanics Meeting, Napa, CA, February 2016.

[81] O. Montenbruck and E. Gill. Satellite Orbits, Models, Methods and Applications. corrected 4th printing, Springer Heidelberg, Dordrecht, Netherlands, 2012.

[82] D. Eberly. Intersection of a Line and a Cone. Geometric Tools, Redmond, WA, December 2014.

[83] P. Patel and D. J. Scheeres. A second-order optimization algorithm using quadric control updates for multistage optimal control problems. Optimal Control Applications and Methods, 30:525–536, April 2009.

[84] A. Ruggiero, P. Pergola, S. Marcuccio, and M. Andrenucci. Low-Thrust Maneuvers for the Efficient Correction of Orbital Elements. 32nd International Electric Propulsion Conference, Wiesbaden, Germany, September 2011.

[85] T. Dargent and V. Martinot. An integrated tool for low thrust optimal control orbit transfers. 18th International Symposium on Space Flight Dynamics, Munich Germany, October 2004.

[86] T. Dargent. Averaging Technique in T-3D an Integrated Tool for Continuous Thrust Optimal Control in Orbit Transfers. AAS/AIAA Spaceflight Mechanics Meeting, Santa Fe, NM, January 2014.

[87] J. D. Kraft. A Solution for Satellite Orbit Time in Umbra and Penumbra with Application to a Lunar Satellite Mission Analysis. NASA Technical Report 19660007948, November 1965.

[88] C. R. O. Longo and S. L. Rickman. Method for the Calculation of Spacecraft Umbra and Penumbra Shadow Terminator Points. NASA Technical Report 19950023025, April 1995.

[89] P. R. Escobal. Methods of Orbit Determination. 2nd Edition, Robert E. Krieger Publishing Company, Huntington, NY, 1976.

[90] T. N. Edelbaum, L. L. Sackett, and H. L. Malchow. Optimal Low Thrust Geocentric Transfer. AIAA 10th Electric Propulsion Conference, Lake Tahoe, NV, October-November 1973.

[91] B. Neta and D. Vallado. On Satellite Umbra/Penumbra Entry and Exit Positions. The Journal of the Astronautical Sciences, 46(1):91–103, January-March 1996.

[92] L. L. Sackett, H. L. Malchow, and T. N. Edelbaum. Solar Electric Geocentric Transfer With Attitude Constraints: Analysis. NASA CR134927, August 1975.

[93] L. L. Sackett, H. L. Malchow, and T. N. Edelbaum. Solar Electric Geocentric Transfer With Attitude Constraints: Manual. NASA CR134927, August 1975.

[94] G. J. Whiffen and J. A. Sims. Application of the SDC Optimal Control Algorithm to Low-Thrust Escape and Capture Trajectory Optimization. AAS/AIAA Space Flight Mechanics Meeting, San Antonio, TX, January 2002.

[95] G. J. Whiffen and J. A. Sims. Application of the SDC optimal control algorithm to low-thrust escape and capture including fourth body effects. 2nd International Symposium on Low Thrust Trajectories, Toulouse, France, June 2002.

[96] A. E. Petropoulos and R. P. Russell. Low-Thrust Transfers using Primer Vector Theory and a Second-Order Penalty Method. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, HI, August 2008.

[97] G. Lantoine and R. P. Russell. Near-Ballistic Halo-to-Halo Transfers between Planetary Moons. Journal of the Astronautical Sciences, 58(3):335–363, 2011.

[98] K. E. Davis, R. L. Anderson, D. J. Scheeres, and G. H. Born. The use of invariant manifolds for transfers between unstable periodic orbits of different energies. Celestial Mechanics and Dynamical Astronomy, 107(4):471–485, August 2010.

[99] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross. Heteroclinic Connections between Periodic Orbits and Resonance Transitions in Celestial Mechanics. Chaos, 10:427–469, June 2000.

[100] MPI: A Message-Passing Interface Standard Version 3.0. University of Tennessee, Knoxville, TN, September 2012.

Appendix A

The Dynamics Matrix and Tensor for the Augmented Modified Equinoctial Element State Vector

In order to compute the STMs for the augmented *MEE* state vector, the dynamics matrix and tensor must be obtained for each term in Equation 4.20. Derivatives of $\boldsymbol{b}$ are available directly, but the preceding terms require careful application of the chain rule. First, the thrust term is restated.

$$A\boldsymbol{\Delta}_{\text{Thrust}} = \frac{A\boldsymbol{u}T_a}{m} \tag{A.1}$$

The thrust contribution to the dynamics matrix is the first derivative of Equation A.1 with respect to the augmented *MEE* state vector.

$$\frac{\partial(A\boldsymbol{u}T_a/m)}{\partial \boldsymbol{X}_{MEE}} = \frac{A\boldsymbol{u}}{m}\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}} + \frac{A}{m}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}_{MEE}}T_a + A\frac{\partial}{\partial \boldsymbol{X}_{MEE}}\left(\frac{1}{m}\right)\boldsymbol{u}T_a + \frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\frac{\boldsymbol{u}T_a}{m} \tag{A.2}$$

Power models frequently use the Cartesian inertial state, so it is convenient to obtain the derivative of the thrust available with respect to the $\boldsymbol{X}_{IJK}$ and then make the transformation to the $\boldsymbol{X}_{MEE}$ sensitivity.

$$\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}} = \frac{\partial T_a}{\partial \boldsymbol{X}_{IJK}}\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}} \tag{A.3a}$$

$$\left(\frac{\partial^2 T_a}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,a} = \left(\frac{\partial T_a}{\partial \boldsymbol{X}_{IJK}}\right)^{\gamma_1}\left(\frac{\partial^2 \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}^2}\right)^{\gamma_1,ia} + \left(\frac{\partial^2 T_a}{\partial \boldsymbol{X}_{IJK}^2}\right)^{\gamma_1,\gamma_2}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,i}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_2,a} \tag{A.3b}$$

Tensor notation has been introduced with the convention that superscripts are indices, and repeated indices of γ are summed over.

With the second derivative of the thrust available now defined with respect to the $\boldsymbol{X}_{MEE}$, the dynamics tensor contributions from thrust can be obtained.

$$\frac{\partial^2 (A\boldsymbol{u}T_a/m)}{\partial \boldsymbol{X}_{MEE}^2} = t_{11} + t_{12} + t_{13} + t_{14}$$

$$+ t_{21} + t_{22} + t_{23} + t_{24}$$

$$+ t_{31} + t_{32} + t_{33} + t_{34} \tag{A.4a}$$

$$+ t_{41} + t_{42} + t_{43} + t_{44}$$

$$t_{11}^{i,ab} = \left(\frac{A}{m}\right)^{i,\gamma_1} \boldsymbol{u}^{\gamma_1} \left(\frac{\partial^2 T_a}{\partial \boldsymbol{X}_{MEE}^2}\right)^{a,b} \tag{A.4b}$$

$$t_{12}^{i,ab} = \left(\frac{A}{m}\right)^{i,\gamma_1} \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,a} \left(\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}}\right)^{b} \tag{A.4c}$$

$$t_{13}^{i,ab} = A^{i,\gamma_1} \boldsymbol{u}^{\gamma_1} \left[\frac{\partial}{\partial \boldsymbol{X}_{MEE}} \left(\frac{1}{m}\right)\right]^{a} \left(\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}}\right)^{b} \tag{A.4d}$$

$$t_{14}^{i,ab} = \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a} \left(\frac{\boldsymbol{u}}{m}\right)^{\gamma_1} \left(\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}}\right)^{b} \tag{A.4e}$$

$$t_{22}^{i,ab} = \left(\frac{A}{m}\right)^{i,\gamma_1} \left(\frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{X}_{MEE}^2}\right)^{\gamma_1,ab} T_a \tag{A.4f}$$

$$t_{23}^{i,ab} = A^{i,\gamma_1} \left[\frac{\partial}{\partial \boldsymbol{X}_{MEE}} \left(\frac{1}{m}\right)\right]^{a} \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,b} T_a \tag{A.4g}$$

$$t_{24}^{i,ab} = \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a} \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,b} \left(\frac{T_a}{m}\right) \tag{A.4h}$$

$$t_{33}^{i,ab} = A^{i,\gamma_1} \boldsymbol{u}^{\gamma_1} \left[\frac{\partial^2}{\partial \boldsymbol{X}_{MEE}^2} \left(\frac{1}{m}\right)\right]^{a,b} T_a \tag{A.4i}$$

$$t_{34}^{i,ab} = \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a} \boldsymbol{u}^{\gamma_1} \left[\frac{\partial}{\partial \boldsymbol{X}_{MEE}} \left(\frac{1}{m}\right)\right]^{b} T_a \tag{A.4j}$$

$$t_{44}^{i,ab} = \left(\frac{\partial^2 A}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,\gamma_1 ab} \boldsymbol{u}^{\gamma_1} \left(\frac{T_a}{m}\right) \tag{A.4k}$$

$$t_{21}^{i,ab} = t_{12}^{i,ba}, \quad t_{31}^{i,ab} = t_{13}^{i,ba}, \quad t_{32}^{i,ab} = t_{23}^{i,ba}$$

$$\tag{A.4l}$$

$$t_{41}^{i,ab} = t_{14}^{i,ba}, \quad t_{42}^{i,ab} = t_{24}^{i,ba}, \quad t_{43}^{i,ab} = t_{34}^{i,ba}$$

Proceeding to the mass flow rate term,

$$\frac{\partial(\dot{m}T_a)}{\partial \boldsymbol{X}_{MEE}} = \dot{m}\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}} + \frac{\partial \dot{m}}{\partial \boldsymbol{X}_{MEE}}T, \tag{A.5a}$$

$$\left(\frac{\partial^2(\dot{m}T_a)}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,ab} = \dot{m}^i\left(\frac{\partial^2 T_a}{\partial \boldsymbol{X}_{MEE}^2}\right)^{a,b} + \left(\frac{\partial \dot{m}}{\partial \boldsymbol{X}_{MEE}}\right)^{i,a}\left(\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}}\right)^b + \left(\frac{\partial \dot{m}}{\partial \boldsymbol{X}_{MEE}}\right)^{i,b}\left(\frac{\partial T_a}{\partial \boldsymbol{X}_{MEE}}\right)^a,$$
$$\tag{A.5b}$$

where the second derivatives of $\dot{m}$ are zero and ignored in presentation.

The effects of perturbations are included by considering the summation of their first and second derivatives in an inertial frame.

$$\frac{\partial \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{IJK}} = \sum_{i=0}^{n_p}\frac{\partial \boldsymbol{\delta}_{p_i}}{\partial \boldsymbol{X}_{IJK}} \tag{A.6a}$$

$$\frac{\partial^2 \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{IJK}^2} = \sum_{i=0}^{n_p}\frac{\partial^2 \boldsymbol{\delta}_{p_i}}{\partial \boldsymbol{X}_{IJK}^2} \tag{A.6b}$$

As previously obtained for the thrust available, the perturbation sensitivities must be found with respect to the augmented *MEE* state vector.

$$\frac{\partial \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{MEE}} = \frac{\partial \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{IJK}}\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}} \tag{A.7a}$$

$$\left(\frac{\partial^2 \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,ab} = \left(\frac{\partial \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{IJK}}\right)^{i,\gamma_1}\left(\frac{\partial^2 \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}^2}\right)^{\gamma_1,ab} + \left(\frac{\partial^2 \boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{IJK}^2}\right)^{i,\gamma_1\gamma_2}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,a}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_2,b}$$
$$\tag{A.7b}$$

Derivatives of the rotation matrix Q^T are also required. Those too are first found inertially.

$$\left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{IJK}}\right)^{i,ab} = \frac{\partial (Q^T)^{i,a}}{\partial \boldsymbol{X}_{IJK}^b} \tag{A.8a}$$

$$\left(\frac{\partial^2 Q^T}{\partial \boldsymbol{X}_{IJK}^2}\right)^{i,abc} = \frac{\partial (Q^T)^{i,a}}{\partial \boldsymbol{X}_{IJK}^b \partial \boldsymbol{X}_{IJK}^c} \tag{A.8b}$$

Next, the derivatives of Q^T are found with respect to the augmented *MEE* state vector.

$$\left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{MEE}}\right)^{i,ab} = \left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{IJK}}\right)^{i,a\gamma_1}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,b} \tag{A.9a}$$

$$\left(\frac{\partial^2 Q^T}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,abc} = \left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{IJK}}\right)^{i,a\gamma_1}\left(\frac{\partial^2 \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}^2}\right)^{\gamma_1,bc} + \left(\frac{\partial^2 Q^T}{\partial \boldsymbol{X}_{IJK}^2}\right)^{i,a\gamma_1\gamma_2}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,b}\left(\frac{\partial \boldsymbol{X}_{IJK}}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_2,c}$$
$$\tag{A.9b}$$

The necessary terms have been defined to assemble the dynamics matrix and tensor contributions from perturbations.

$$\left(\frac{\partial(AQ^T\boldsymbol{\delta}_p)}{\partial \boldsymbol{X}_{MEE}}\right)^{i,a} = \left(AQ^T\frac{\partial\boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{MEE}}\right)^{i,a} + A^{i,\gamma_1}\left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,\gamma_2 a}\boldsymbol{\delta}_p^{\gamma_2} + \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a}(Q^T\boldsymbol{\delta}_p)^{\gamma_1} \quad (A.10)$$

$$\frac{\partial^2(AQ^T\boldsymbol{\delta}_p)}{\partial \boldsymbol{X}_{MEE}^2} = t_{11} + t_{12} + t_{13}$$
$$+ t_{21} + t_{22} + t_{23} \qquad (A.11a)$$
$$+ t_{31} + t_{32} + t_{33}$$

$$t_{11}^{i,ab} = A^{i,\gamma_1}(Q^T)^{\gamma_1,\gamma_2}\left(\frac{\partial^2\boldsymbol{\delta}_p}{\boldsymbol{X}_{MEE}^2}\right)^{\gamma_2,ab} \qquad (A.11b)$$

$$t_{12}^{i,ab} = A^{i,\gamma_1}\left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,\gamma_2 a}\left(\frac{\partial\boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_2,b} \qquad (A.11c)$$

$$t_{13}^{i,ab} = \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a}(Q^T)^{\gamma_1,\gamma_2}\left(\frac{\partial\boldsymbol{\delta}_p}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_2,b} \qquad (A.11d)$$

$$t_{22}^{i,ab} = A^{i,\gamma_1}\left(\frac{\partial^2 Q^T}{\partial \boldsymbol{X}_{MEE}^2}\right)^{\gamma_1,\gamma_2 ab}\boldsymbol{\delta}_p^{\gamma_2} \qquad (A.11e)$$

$$t_{23}^{i,ab} = \left(\frac{\partial A}{\partial \boldsymbol{X}_{MEE}}\right)^{i,\gamma_1 a}\left(\frac{\partial Q^T}{\partial \boldsymbol{X}_{MEE}}\right)^{\gamma_1,\gamma_2 b}\boldsymbol{\delta}_p^{\gamma_2} \qquad (A.11f)$$

$$t_{33}^{i,ab} = \left(\frac{\partial^2 A}{\partial \boldsymbol{X}_{MEE}^2}\right)^{i,\gamma_1 ab}(Q^T)^{\gamma_1,\gamma_2}\boldsymbol{\delta}^{\gamma_2} \qquad (A.11g)$$

$$t_{21}^{i,ab} = t_{12}^{i,ba}, \quad t_{31}^{i,ab} = t_{13}^{i,ba}, \quad t_{32}^{i,ab} = t_{23}^{i,ba} \qquad (A.11h)$$

The complete assembly of the dynamics matrix is the summation of Equations A.2, A.5a and A.7a and derivatives of $\boldsymbol{b}$.

$$\frac{\partial \dot{\boldsymbol{X}}_{MEE}}{\partial \boldsymbol{X}_{MEE}} = \frac{\partial(A\boldsymbol{u}T_a/m)}{\partial \boldsymbol{X}_{MEE}} + \frac{\partial(\dot{m}T_a)}{\partial \boldsymbol{X}_{MEE}} + \frac{\partial(AQ^T\boldsymbol{\delta}_p)}{\partial \boldsymbol{X}_{MEE}} + \frac{\partial\boldsymbol{b}}{\partial \boldsymbol{X}_{MEE}} \qquad (A.12)$$

Finally, the complete assembly of the dynamics tensor is the summation of Equations A.4, A.5b and A.7b and second derivatives of $\boldsymbol{b}$.

$$\frac{\partial^2 \dot{\boldsymbol{X}}_{MEE}}{\partial \boldsymbol{X}_{MEE}^2} = \frac{\partial^2(A\boldsymbol{u}T_a/m)}{\partial \boldsymbol{X}_{MEE}^2} + \frac{\partial^2(\dot{m}T_a)}{\partial \boldsymbol{X}_{MEE}^2} + \frac{\partial^2(AQ^T\boldsymbol{\delta}_p)}{\partial \boldsymbol{X}_{MEE}^2} + \frac{\partial^2\boldsymbol{b}}{\partial \boldsymbol{X}_{MEE}^2} \qquad (A.13)$$