AAS 17-253

LOW-THRUST MANY-REVOLUTION TRAJECTORY OPTIMIZATION VIA DIFFERENTIAL DYNAMIC PROGRAMMING AND A SUNDMAN TRANSFORMATION

Jonathan D. Aziz,[*] Jeffrey S. Parker,[†] Daniel J. Scheeres[‡] and Jacob A. Englander[§]

Low-thrust trajectories about planetary bodies characteristically span a high count of orbital revolutions. Directing the thrust vector over many revolutions presents a challenging optimization problem for any conventional strategy. This paper demonstrates the tractability of low-thrust trajectory optimization about planetary bodies by applying a Sundman transformation to change the independent variable of the spacecraft equations of motion to the eccentric anomaly and performing the optimization with differential dynamic programming. Fuel-optimal geocentric transfers are shown in excess of 1000 revolutions while subject to Earth's J_2 perturbation and lunar gravity.

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 and 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.

Indirect Solutions to Low-Thrust Many-Revolution Orbit Transfers

Classical approaches employ optimal control theory, named the indirect, Lagrange multiplier, or adjoint method, beginning with 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 given semi-major axis change, Δa, between two circular orbits [1]. The result was repeated by maximizing the delivered mass over a fixed transfer duration with Δa and Δi specified [2]. 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 velocity change, ΔV, when Δa and Δi are specified [3]. 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.

[*] Ph.D. Student, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309.

[†] Adjunct Professor, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309.

[‡] Distinguished Professor, A. Richard Seebass Endowed Chair, Colorado Center for Astrodynamics Research, University of Colorado, Boulder, CO 80309.

[§] Aerospace Engineer, NASA/GSFC, Code 595, 8800 Greenbelt Rd, Greenbelt, MD 20771, USA.

Wiesel and Alfano solved the resulting two point boundary value problem numerically to produce a contour map of a constant 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 [4].

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 continuous thrust acceleration and a two-body dynamic model [5]. 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 non-singular equinoctial elements, and presents a numerical solution with Newton-Raphson iteration [6]. 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 [7]. Kéchichian further developed the low-thrust rendezvous in equinoctial elements by considering Earth zonal harmonics up to J_4 [8], and shows analytical but suboptimal approaches for changing Δa or Δi with eclipses included [9, 10].

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. 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 non-intuitive 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.

Low-Thrust 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 determined by the Lagrange multipliers, but sub-optimal 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 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 chosen, trajectory computation can proceed indefinitely. Lyapunov feedback control laws are desirable so that convergence can be expected.

Kluever blends the results of optimal control theory for the individual minimum-time changes Δa, Δi and to eccentricity, Δe, subject to two-body dynamics [11]. The control law assembles the thrust direction from the weighted optimal thrust directions to change each orbital element. Chang, Chichka, and Marsden suggest a Lyapunov function that is the weighted sum of squared errors of the angular momentum and Laplace vectors between the current and target orbit [12]. Minimizing this Lyapunov function at each instance in time yields asymptotic stability for local transfers between Keplerian orbits. Chang et al. suggest transfers through a finite number of intermediate orbits for arbitrary global transfers, e.g. over many revolutions, and present a method for choosing those intermediate orbits.

Petropolous derived analytic expressions for the optimal thrust directions and optimal orbit locations for changing each of the classical orbital elements to propose two control law strategies that include a mechanism for coasting based on the effectivity of the maneuver [13]. This work was further developed as the Q-law, where Q is a candidate Lyapunov function named the proximity quotient [14]. 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. Petropolous 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 with inclination change included and for a transfer about Vesta, and successful computation of a challenging Molniya-type transfer.

Heuristic control laws generally yield sub-optimal 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, for example. 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.

Direct Optimization of Low-Thrust Trajectories

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. 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 [15]. 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 or 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 programming to planetocentric low-thrust trajectories that require a large decision vector. Nonetheless, Betts solved the large-scale NLP problem for geocentric trajectories over several hundred revolutions using collocation and sequential quadratic programming in the Sparse Optimization Suite (SOS) software [16–20]. Betts presents a 578 revolution transfer between low Earth orbits with continuous but throttled thrust and oblate Earth perturbations through J_4 [18] and a minimum time lunar transfer with two burn phases of maximum thrust and an intermediate coast phase [19]. More recently, Betts modeled coast phases for eclipse conditions and constructed initial guesses for thrust arcs with the control law from Chang et al. An initial guess of the entire trajectory was constructed with a receding horizon algorithm and supplied to SOS for optimization [20]. Furthermore, the equations of motion considered oblate Earth perturbations through J_4, Sun and lunar gravity perturbations, and true longitude as the independent variable. Transfers 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 maximized the delivered mass.

A linear scaling of the optimization problem size with number of control variables is characteristic of differential dynamic programming (DDP) [21]. 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 methods that update the entire control sequence in the computationally expensive matrix inverse of a large Hessian. 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* [22].

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 current state-of-the-art technology for the optimal design of low-thrust planetary trajectories is the Mystic Low-Thrust Trajectory Design and Visualization Software [23]. Mystic has best demonstrated its capabilities with the success of NASA's Dawn mission [24]. After an interplanetary leg that included a Mars gravity assist, Dawn completed its first planetary segment about the asteroid Vesta and will end its mission in a second planetary segment at the asteroid Ceres, where it currently operates. Mystic's optimization engine is built around the Static/Dynamic Optimal Control algorithm, a DDP approach developed by Whiffen [25]. Despite the favorability of DDP for large scale optimization, computation time limits Mystic to about 250 revolutions for optimized trajectories before switching to the Q-law [26]. Lantoine and Russell introduced Hybrid Differential Dynamic Programming (HDDP) [27–29], a DDP variant that makes the most computationally expensive step suitable for parallelization.

DDP and a Sundman Transformation

This work advances the capability of differential dynamic programming applied to low-thrust spaceflight by changing the independent variable of the equations of motion. The Sundman transformation [30] is a general change of variables from time to a function of orbital radius, and effectively regularizes the step size of numerical integration [31]. Selecting the new independent variable as the eccentric anomaly is found to add numeric stability when mapping objective function sensitivities along a trajectory, making efficient optimization possible over hundreds of revolutions and thousands of control variables. Yam, Izzo, and Biscani previously applied a Sundman transformation to the Sim-Flanagan transcription to optimize interplanetary trajectories [32, 33]. Pellegrini, Russell, and Vivek show accuracy and efficiency gains for propagation with the Sundman transformation in the Stark and Kepler models [34].

Presentation begins with the state and dynamic model used to compute example trajectories. Next, the generalized Sundman transformation is provided with a derivation for the change of variable to eccentric anomaly. The optimization problem is then posed, and the procedure for computing trajectories is detailed. Implementation closely follows the procedure outlined by the HDDP algorithm [27], so presentation of the algorithm is withheld. Results for the Sundman-transformed DDP approach follow in the form of an example orbit transfer solved for different cases of increasing complexity.

PROBLEM FORMULATION

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

Spacecraft State and Dynamics

The spacecraft state is chosen as a Cartesian representation of the spacecraft inertial position and velocity.

$$\boldsymbol{r} = [x \quad y \quad z]^T, \quad \boldsymbol{v} = [\dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{1}$$

Henceforth, boldface is reserved for column vectors and an overhead dot denotes the time derivative. Implementation of the HDDP algorithm makes use of an augmented state vector that includes time t, mass m, and thrust control variables T, α, and β.

$$\boldsymbol{X} = [t \quad x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad m \quad T \quad \alpha \quad \beta]^T \tag{2}$$

Table 1: Dynamic Model Parameters.

$\mu_{\oplus}$	398600.44 km^3/s^2	T_{max}	0.25 N
$\mu_{\leftmoon}$	4904.928372 km^3/s^2	I_{sp}	1950 s
J_2	0.0010826265	g_0	0.00980665 km/s^2
$R_{\oplus}$	6378.136 km		

Spherical thrust control is defined by magnitude T, yaw angle α, and pitch angle β, where the angles are defined relative to the radial-transverse-normal (RSW) frame. RSW basis vectors and thus the rotation to the inertial frame are defined by,

$$[\hat{\boldsymbol{r}} \quad \hat{\boldsymbol{s}} \quad \hat{\boldsymbol{w}}] = \left[\frac{\boldsymbol{r}}{r} \quad \frac{(\boldsymbol{r}\times\boldsymbol{v})\times\boldsymbol{r}}{\|(\boldsymbol{r}\times\boldsymbol{v})\times\boldsymbol{r}\|} \quad \frac{\boldsymbol{r}\times\boldsymbol{v}}{\|\boldsymbol{r}\times\boldsymbol{v}\|}\right]. \tag{3}$$

Thrust vector components are then

$$\begin{bmatrix} T_r \\ T_s \\ T_w \end{bmatrix} = \begin{bmatrix} T\sin\alpha\cos\beta \\ T\cos\alpha\cos\beta \\ T\sin\beta \end{bmatrix}, \quad \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = [\hat{\boldsymbol{r}} \quad \hat{\boldsymbol{s}} \quad \hat{\boldsymbol{w}}] \begin{bmatrix} T_r \\ T_s \\ T_w \end{bmatrix}, \tag{4}$$

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. No concern is given for angle wrapping. In fact, computation exhibits favorable performance when the angles are unbounded. Spacecraft dynamics consider geocentric two-body motion perturbed by thrust, J_2, and lunar gravity,

$$\dot{\boldsymbol{X}} = \dot{\boldsymbol{X}}_{\oplus} + \dot{\boldsymbol{X}}_{T} + \dot{\boldsymbol{X}}_{J_2} + \dot{\boldsymbol{X}}_{\leftmoon}, \tag{5}$$

where $\dot{\boldsymbol{X}}_{\oplus}$ is the two-body motion due to point mass Earth gravity, $\dot{\boldsymbol{X}}_T$ is the thrust acceleration and mass flow rate, $\dot{\boldsymbol{X}}_{J_2}$ is Earth's J_2 perturbation, and $\dot{\boldsymbol{X}}_{\leftmoon}$ is the point mass lunar gravity perturbation, defined by,

$$\dot{\boldsymbol{X}}_{\oplus} = \begin{bmatrix} 1 & \dot{x} & \dot{y} & \dot{z} & -\frac{\mu_{\oplus}}{r^3}x & -\frac{\mu_{\oplus}}{r^3}y & -\frac{\mu_{\oplus}}{r^3}z & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{6a}$$

$$\dot{\boldsymbol{X}}_{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{T_x}{m} & \frac{T_y}{m} & \frac{T_z}{m} & -\frac{T}{I_{sp}g_0} & 0 & 0 & 0 \end{bmatrix}^T, \tag{6b}$$

$$\dot{\boldsymbol{X}}_{J_2} = -\frac{3J_2\mu_{\oplus}R_{\oplus}^2}{2r^5} \begin{bmatrix} 0 & 0 & 0 & 0 & x(1-5\frac{z^2}{r^2}) & y(1-5\frac{z^2}{r^2}) & z(3-5\frac{z^2}{r^2}) & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{6c}$$

$$\dot{\boldsymbol{X}}_{\leftmoon} = -\mu_{\leftmoon} \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{x-x_{\leftmoon}}{\|\boldsymbol{r}-\boldsymbol{r}_{\leftmoon}\|^3} + \frac{x_{\leftmoon}}{r_{\leftmoon}^3} & \frac{y-y_{\leftmoon}}{\|\boldsymbol{r}-\boldsymbol{r}_{\leftmoon}\|^3} + \frac{y_{\leftmoon}}{r_{\leftmoon}^3} & \frac{z-z_{\leftmoon}}{\|\boldsymbol{r}-\boldsymbol{r}_{\leftmoon}\|^3} + \frac{z_{\leftmoon}}{r_{\leftmoon}^3} & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{6d}$$

Gravitational parameters for the Earth and the Moon are $\mu_{\oplus}$ and $\mu_{\leftmoon}$, respectively. A constant power model is assumed with $T \in [0, T_{max}]$. Thrust magnitude constraints are enforced by a null-space method [28]. Mass flow rate is inversely proportional to the specific impulse, I_{sp}, and acceleration due to gravity at sea level g_0. The J_2 perturbation is owed to the Earth's oblateness, and is a function of the Earth's equatorial radius $R_{\oplus}$. Table 1 lists these dynamic model constants. Including the lunar perturbation requires the Moon's inertial position with respect to the Earth,

$$\boldsymbol{r}_{\leftmoon} = [x_{\leftmoon} \quad y_{\leftmoon} \quad z_{\leftmoon}]^T, \tag{7}$$

that is assumed to evolve according to geocentric two-body motion. The Moon's state is initialized with the orbital elements listed in Table 2, where ω and M are introduced as the argument of periapsis and mean anomaly.

Table 2: The Moon's Earth-Centered ICRF Orbital Elements at 01 Jan 2000 00:00:00.0 TDB.

a	381218.68756119191 km	Ω	12.23324045627363°
e	0.06476694126942699	ω	60.7835754956735°
i	20.94024252661913°	M	140.7402558848975°

A 2000 kg spacecraft is initialized at the J2000 epoch on the x-axis with the position and velocity selected for an example GTO with approximately 300 km perigee altitude and 28.5° inclination.

$$\boldsymbol{X}_0 = \begin{bmatrix} t_0 \\ x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \\ m_0 \\ T_0 \\ \alpha_0 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \text{01 Jan 2000 12:00:00.0 TDB} \\ \text{6678.1363 km} \\ 0 \\ 0 \\ 0 \\ \text{8.92130624172 km/s} \\ \text{4.84387407216 km/s} \\ \text{2000 kg} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

Equation 8 is the spacecraft initial state vector. The initial time is listed as the J2000 epoch, but state variable t is numerically integrated to track the relative time from $t_0 = 0$. An initial guess for the controls must be provided along the entire trajectory. Those controls are identically zero at all times as shown for the initial state.

Sundman Transformation

In regularizing the equations of motion to solve the three-body problem, Karl Sundman introduced a change of independent variable from time to the new independent variable τ [30].

$$dt = c_n r^n d\tau \tag{9}$$

Time and the new independent variable are related by a function of the orbital radius. The real number n and coefficient c_n may be selected so that τ represents an orbit angle. Transforming the time-dependent equations of motion simply requires a multiplication by the functional relationship between the two independent variables.

$$\overset{\circ}{\boldsymbol{X}} = \left(\frac{d}{dt}\boldsymbol{X}\right)\frac{dt}{d\tau} = \dot{\boldsymbol{X}} c_n r^n \tag{10}$$

Time-dependent equations of motion are typically propagated for a prescribed duration from the state at an initial epoch. Now however, propagation is specified for a duration of τ and the elapsed time is unknown *a priori*. Time may be tracked by including it in the state vector.

The Sundman transformation to eccentric anomaly is found by differentiating Kepler's equation,

$$t = \sqrt{a^3/\mu}\,(E - e\sin E)\ , \tag{11}$$

$$dt = \sqrt{a^3/\mu}\,(1 - e\cos E)dE = c_n r^n d\tau\ , \tag{12}$$

where E is the eccentric anomaly. Considering $n = 1$ and making use of the relationship $r = a(1 - e\cos E)$ leads to a solution for c_1.

$$c_1 = (1/r)\sqrt{a^3/\mu}\,(1 - e\cos E) = (1/r)\sqrt{a^3/\mu}\,(r/a) = \sqrt{a/\mu} \tag{13}$$

$$dt = \sqrt{a/\mu}\, r dE \tag{14}$$

Sundman Transformed Dynamics

The Sundman transformation is made after assembling the complete equations of motion with respect to time.

$$\mathring{\boldsymbol{X}} = (\dot{\boldsymbol{X}}_{\oplus} + \dot{\boldsymbol{X}}_{T} + \dot{\boldsymbol{X}}_{J_2} + \dot{\boldsymbol{X}}_{\leftmoon})\sqrt{a/\mu_{\oplus}}\, r \tag{15}$$

HDDP makes use of the first-order state transition matrix (STM) and second-order state transition tensor (STT) to map cost function derivatives between stages. Differential equations for the STMs (referring to both the STMs and STTs) rely on the dynamics matrix and tensor,

$$A^{i,j} = \frac{\partial \dot{\boldsymbol{X}}^{i}}{\partial \boldsymbol{X}^{j}}\ , \tag{16a}$$

$$A^{i,jk} = \frac{\partial^{2} \dot{\boldsymbol{X}}^{i}}{\partial \boldsymbol{X}^{j} \partial \boldsymbol{X}^{k}}\ , \tag{16b}$$

where tensor notation has been adopted to avoid ambiguities. The dynamics matrix and tensor similarly need to be transformed to reflect the change of independent variable.

$$\mathit{\Lambda}^{i,j} = \frac{\partial \mathring{\boldsymbol{X}}^{i}}{\partial \boldsymbol{X}^{j}} \tag{17a}$$

$$\mathit{\Lambda}^{i,jk} = \frac{\partial^{2} \mathring{\boldsymbol{X}}^{i}}{\partial \boldsymbol{X}^{j} \partial \boldsymbol{X}^{k}} \tag{17b}$$

The new dynamics matrix and tensor in Equations (17a) and (17b) are similarly obtained first with respect to time and then the change of variables is performed 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.

$$\eta = dt/d\tau = c_n r^n \tag{18}$$

$$\eta_{X}{}^{i} = \frac{\partial \eta}{\partial \boldsymbol{X}^{i}} \tag{19}$$

$$\eta_{XX}{}^{i,j} = \frac{\partial^{2} \eta}{\partial \boldsymbol{X}^{i} \partial \boldsymbol{X}^{j}} \tag{20}$$

After assembling Equations (16a) and (16b), the chain rule completes the transformation.

$$\mathit{\Lambda}^{i,j} = A^{i,j}\eta + \dot{\boldsymbol{X}}^{i}\eta_{X}{}^{j} \tag{21a}$$

$$\mathit{\Lambda}^{i,jk} = A^{i,jk}\eta + A^{i,j}\eta_{X}{}^{k} + A^{i,k}\eta_{X}{}^{j} + \dot{\boldsymbol{X}}^{i}\eta_{XX}{}^{j,k} \tag{21b}$$

Equations (18) to (21) are stated generally for any change of independent variable, but τ is selected as the eccentric anomaly for the computed examples.

Augmented Lagrangian Method

The standard DDP formulation adjoins terminal constraints $\boldsymbol{\psi}(\boldsymbol{X}_f) = 0$ to the original cost function using a constant vector of Lagrange multipliers. HDDP selects a cost function based on the augmented Lagrangian method where 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 can be treated individually.

$$J = \phi + \boldsymbol{\lambda}^{T}\boldsymbol{\psi} + \boldsymbol{\psi}^{T}\mathit{\Sigma}\boldsymbol{\psi} \tag{22}$$

The first term $\phi(\boldsymbol{X}_f)$ is the original objective to be minimized, where $\boldsymbol{X}_f$ is the final value of the state vector. Multipliers $\boldsymbol{\lambda}$ are initialized as the zero vector and updated at every iteration to push the trajectory toward feasibility. Penalty matrix $\mathit{\Sigma}$ places additional weight on constraint violations and serves to initialize a quadratic cost function space. In contrast to previous approaches that continually increase the penalty weight [27, 35], 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, 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.

Final conditions for GEO are described in the terminal constraint function $\boldsymbol{\psi}$, so that $\boldsymbol{\psi}(\boldsymbol{X}_f) = 0$ for a feasible solution. The objective is to arrive at GEO after expending the minimum amount of propellant.

$$\phi = -m_f \tag{23a}$$

$$\boldsymbol{\psi} = \begin{bmatrix} \|\boldsymbol{r}_f\| - 42164.169972 \text{ km} \\ \|\boldsymbol{v}_f\| - 3.07466008566 \text{ km/s} \\ \boldsymbol{r}_f \cdot \boldsymbol{v}_f \\ z_f \\ \dot{z}_f \end{bmatrix} \tag{23b}$$

$$\Sigma = diag(\sigma_0, \sigma_1, \sigma_3, \sigma_4) \tag{23c}$$

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 places additional weight on the final position and velocity magnitude constraints, and its entries are later specified for each computed example. A scaled feasibility tolerance requires $\|\boldsymbol{\psi}\| < 1 \times 10^{-6}$ and an optimality tolerance requires $ER_0 < 1 \times 10^{-7}$, where ER_0 is the expected reduction of the objective function obtained in the backward sweep [27]. Then the constraint violations are satisfactorily small and the backward sweep sees a stationary point of the cost function with respect to controls and multipliers. Scaling improves the numerical stability of the procedures 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

$$\begin{aligned} DU &= 42164.0 \text{ km} \\ TU &= 10\sqrt{DU^3/\mu_\oplus} \\ FU &= 0.25 \text{ N} \\ MU &= FU\,TU^2/DU \end{aligned} , \tag{24}$$

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. The scaled feasibility tolerance corresponds to a 42.164 m position requirement and 0.3075 mm/s velocity requirement.

Trajectory Computation

HDDP considers a discrete form of DDP where a trajectory can be described by any number of phases, with each phase described by a number of stages. This work considers single phase trajectories of N stages. The trajectory computation step is named the *forward pass*, and is the sequence of function evaluations,

$$\boldsymbol{X}_{k+1} = \boldsymbol{F}(\boldsymbol{X}_k),\ k = 0, 1, ..., N - 1. \tag{25}$$

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 [21]. The relevant transition function is the integral of Equation 15 between stages.

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \int_{E_k}^{E_{k+1}} \mathring{\boldsymbol{X}}_k \, dE \tag{26}$$

Propagating a trajectory from the initial state requires effective discretization and a numerical integration scheme. Equation 26 is integrated with a fixed-step fifth-order Dormand-Prince method [36]. The trajectory is described by 100 stages per revolution that are equally spaced in eccentric anomaly. There are then 300 control variables for every revolution. Each stage offers an opportunity to update the thrust control variables that are held constant across an integration step.

A fixed integration step accumulates $\Delta E = E_{k+1} - E_k = 2\pi/100$. Having set the initial guess for all stage control variables to zero, the first iteration considers a ballistic orbit in GTO for a prescribed number of revolutions, N_{rev}. The fixed transfer duration in eccentric anomaly is $2\pi N_{\text{rev}}$.

STM Computation

Results were generated on a Dual Intel Xeon quad-core 2GHz, 24GB memory Linux machine. STM computations were distributed in parallel across 8 cores with OpenMP [37]. 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 26 is augmented with the STM differential equations.

$$\mathring{\Phi}^{i,j} = \Lambda^{i,a}\Phi^{a,j} \tag{27a}$$

$$\mathring{\Phi}^{i,jk} = \Lambda^{i,a}\Phi^{a,jk} + \Lambda^{i,ab}\Phi^{a,j}\Phi^{b,k} \tag{27b}$$

STMs are initialized as $\Phi^{i,j} = \delta^{i,j}$ and $\Phi^{i,jk} = 0$. Equation 27 is stated in terms of the Sundman transformation, but all that changes from the time-dependent case is the notation. 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 $k = N - 1$.

RESULTS

Four example trajectories are computed from initial conditions in Equation 8, and with HDDP employed to minimize the augmented Lagrangian described in Equations (22) and (23). First, only two-body dynamics are considered, so that Equation 15 ignores the J_2 and lunar gravity perturbations. The J_2 perturbation is introduced in the second case, while the third and fourth cases include both perturbations. The transfer duration is fixed at $N_{\text{rev}} = 450.5$ for the first three cases, and $N_{\text{rev}} = 1000.5$ for the final case. The penalty matrix is initially tuned to $\Sigma = diag(100, 10, 1, 1, 1)$.

Case 1: 450.5 Revolutions With 2-Body Dynamics

The first example spans 450.5 revolutions, yielding 135,150 control variables to compute. In the 2-body dynamic model, the spacecraft acceleration is due to point-mass Earth gravity and thrust. The choice of $N_{\text{rev}} = 450.5$ is somewhat arbitrary but allows quick computation of a many-revolution trajectory with a large number of control switches between thrust and coast arcs. Prescribing the extra half-revolution fixes the terminal state of the initial guess to be at apogee on the negative x-axis.

Figure 1 provides a two-dimensional view of the resulting transfer in the equatorial plane and the steering profile for both thrust angles. After 450.5 revolutions, 315.75 days have elapsed and the spacecraft arrives in GEO with a final mass of 1759.1754 kg. The eccentricity vector and line of nodes remain coincident throughout the transfer, yielding symmetry in the pitch and yaw steering to remove the eccentricity and inclination. Apogee raising early in the transfer is leveraged for a cheaper inclination change. Both Figures 1 and 2 show how a long initial maneuver is followed by the bang-bang thrust profile expected for a fuel-optimal transfer. Yaw steering is symmetric about the transverse direction and switches sign at the apsides. Pitch steering to change the inclination is favored around apogee. Both pitch and yaw are zero-valued for coast arcs, which is not arbitrary, because this sets the nominal value for a control update during iteration. Given the definitions of α and β in Equation 4, if an iterate attempts to turn the thrust on, that initial thrust increment must step from the positive transverse and/or normal directions.

Figure 3 shows the performance of HDDP from the ballistic initial guess through 86 iterations to convergence in 54 minutes of computer runtime. The algorithm approaches the final solution within the first half of the total iterations. Remaining efforts then refine the solution to meet convergence criteria. Final values for the multipliers are listed in Table 3. Constraints on the flight path angle and z-coordinate are satisfied by the initial guess. The associated multipliers λ_2 and λ_3 only see minor updates from their initial values of zero.

Evolution of the constraint violations and multipliers through each iteration is provided in Figure 4. The initial guess begins with small errors in the radial position, flight path angle and z-coordinate that have been introduced by numerical integration. Errors are primarily incurred during the many low altitude perigee passages of the ballistic 450.5 revolutions in GTO. The step size regularization achieved by changing variables to the eccentric anomaly works to mitigate this effect. Integration with respect to time would require more steps around perigee to maintain accuracy, further increasing the problem size. When thrusting commences through the iterations, those perigee passages are at higher altitudes and integration accuracy increases. Large initial constraint violations in velocity magnitude and z-velocity are owed to the initial orbit size and inclination.

Iteration history of the radial position constraint ψ_0 is evidence of the motivation for its increased penalty weight $\sigma_0 = 100$, the first entry in Σ. The algorithm favored sacrificing feasibility in radial position while working to satisfy the other constraints, so much as to hinder advancements toward convergence in earlier trials with a smaller penalty weight. The oscillatory behavior of the z-velocity constraint ψ_4 suggests that its penalty weight could also be increased to improve the rate of convergence. The other constraints appear well-tuned, as they are directed quickly and uniformly toward feasibility. The multipliers also deviate far from their final values before convergence. It is possible that a smaller trust-region to restrict the multiplier update at each iteration might further improve performance. These are empirical observations and largely speculative with regard to performance, but provide insight into how the algorithm might be tuned.

Case 2: 450.5 Revolutions With J_2 Perturbation

Next, the gravitational perturbation of Earth's oblateness is included by introducing the J_2 spherical harmonic term to the dynamic model. J_2 causes a secular precession in the right ascension of the ascending node and rotation in the argument of perigee. At 28.5° inclination, the argument of perigee advances in the direction of spacecraft motion. Now the ballistic initial guess no longer terminates on the equator at GEO radius.

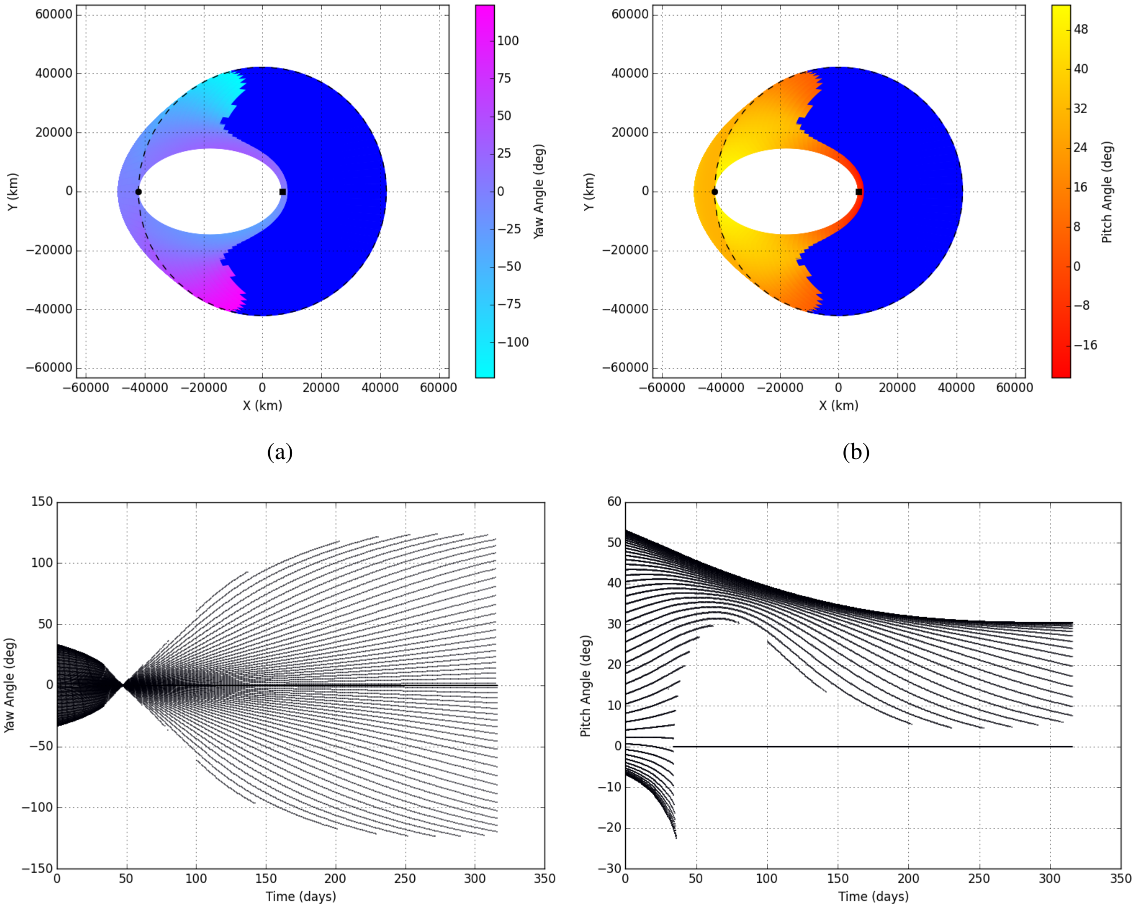
The trajectory and steering profile for the J_2-perturbed transfer are depicted in Figure 5. Compensating for the J_2 perturbation leads to an increased propellant mass requirement so that the final mass in GEO is reduced to 1737.1949 kg. Time of flight increases to 342.13 days. The final inertial longitude defined by $\tan\theta = y/x$ is unconstrained and drifts from 180° in the 2-body model to 199.8728° with the J_2 perturbation. The number of iterations, computer runtime, final mass, time of flight, and final longitude are summarized in Table 4 for each example trajectory.

Thrust pitch and yaw steering exhibit similar behavior to the 2-body case, but the symmetries are warped following perturbations to the node and eccentricity vectors. Algorithmic performance is also similar to the 2-body case, where the solution is approached early on and much of the effort is then refining the control to meet convergence criteria. The number of iterations required increases to 110 in 61 minutes of runtime. Multipliers take on a different set of values, largely reflected in the additional effort to satisfy the flight path angle and z-coordinate constraints.

Case 3: 450.5 Revolutions With J_2 and Lunar Gravity

Fidelity of the dynamic model is further improved by including the lunar gravity perturbation. The orbital state of the Moon is obtained at each integration step (and substep) by analytical Kepler propagation of the state in Table 2 to the current epoch, obtained from the the elapsed time t in the state vector. Note the 12 hour offset in the Moon's reference state and t_0.

Figure 6 shows both the resulting transfer and the Moon's trajectory in a three-dimensional view. The transfer depicted in Figure 7 is similar to the J_2 case. Nonetheless, the lunar gravity perturbation is leveraged



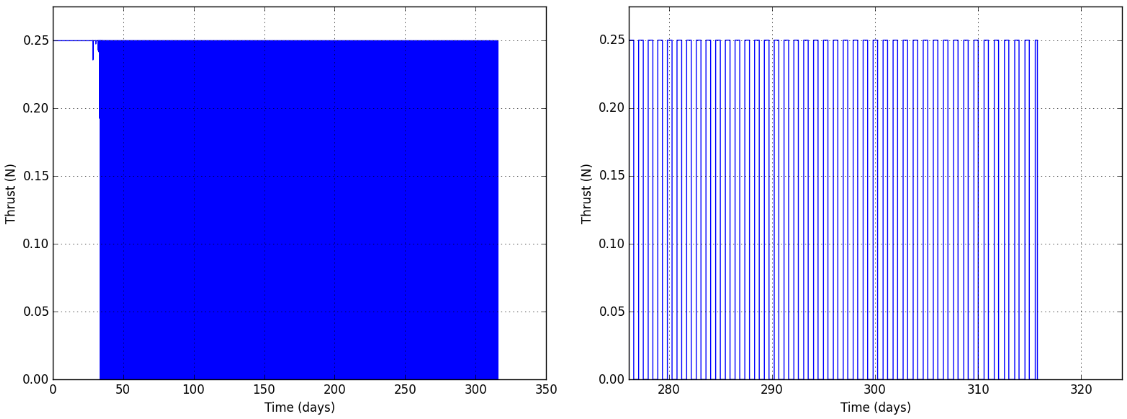
(a)

(b)

(c)

(d)

Figure 1: An equatorial projection of the 450.5 revolution transfer from GTO to GEO with 2-body dynamics is colored by (a) yaw angle contours and (b) pitch angle contours with coast arcs in blue. Markers are placed at the initial and final states and a dashed line marks the target GEO orbit. (c) Thrust yaw angle and (d) pitch angle are shown for the duration of the transfer.



(a)

(b)

Figure 2: Thrust magnitude through the 450.5 revolution transfer from GTO to GEO with 2-body dynamics for (a) the full mission duration and (b) a zoom on the final days to emphasize the bang-bang control structure.

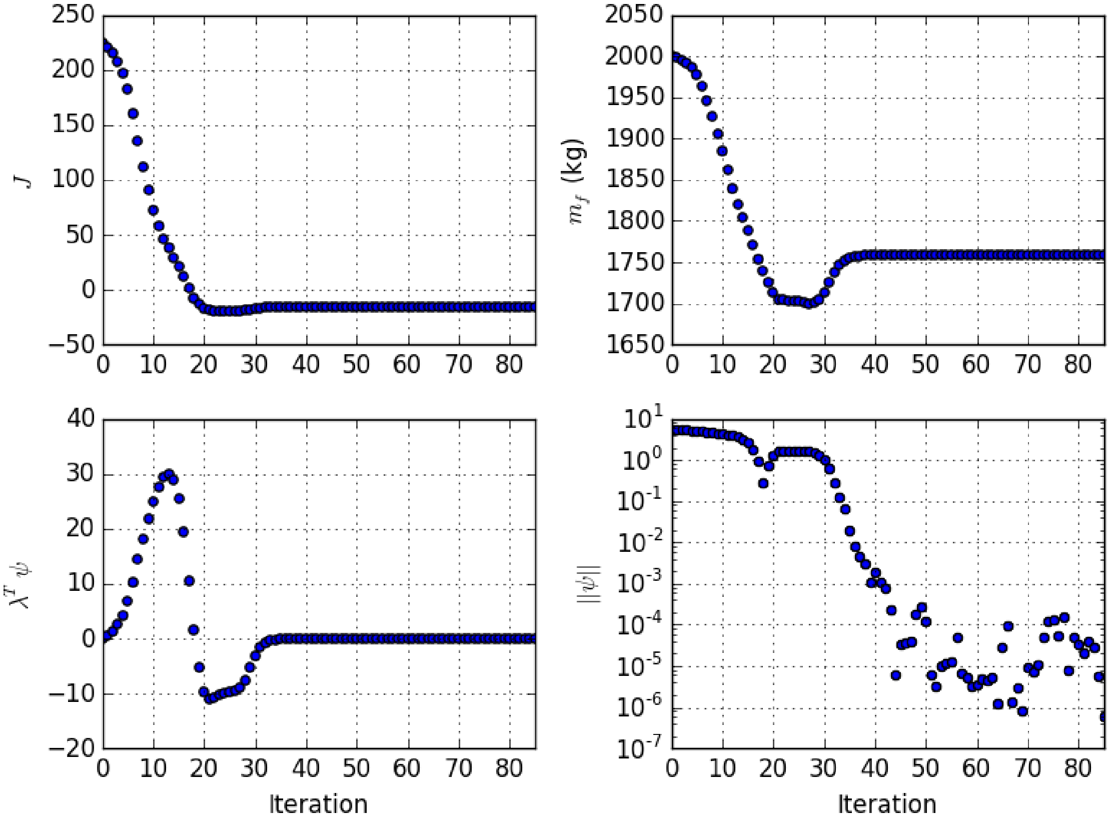


Figure 3: Iteration history of the scaled cost function, the final mass in kilograms, the scaled cost contributions from the multiplier term and the scaled norm of the constraint violations for the 450.5 revolution transfer from GTO to GEO with 2-body dynamics.

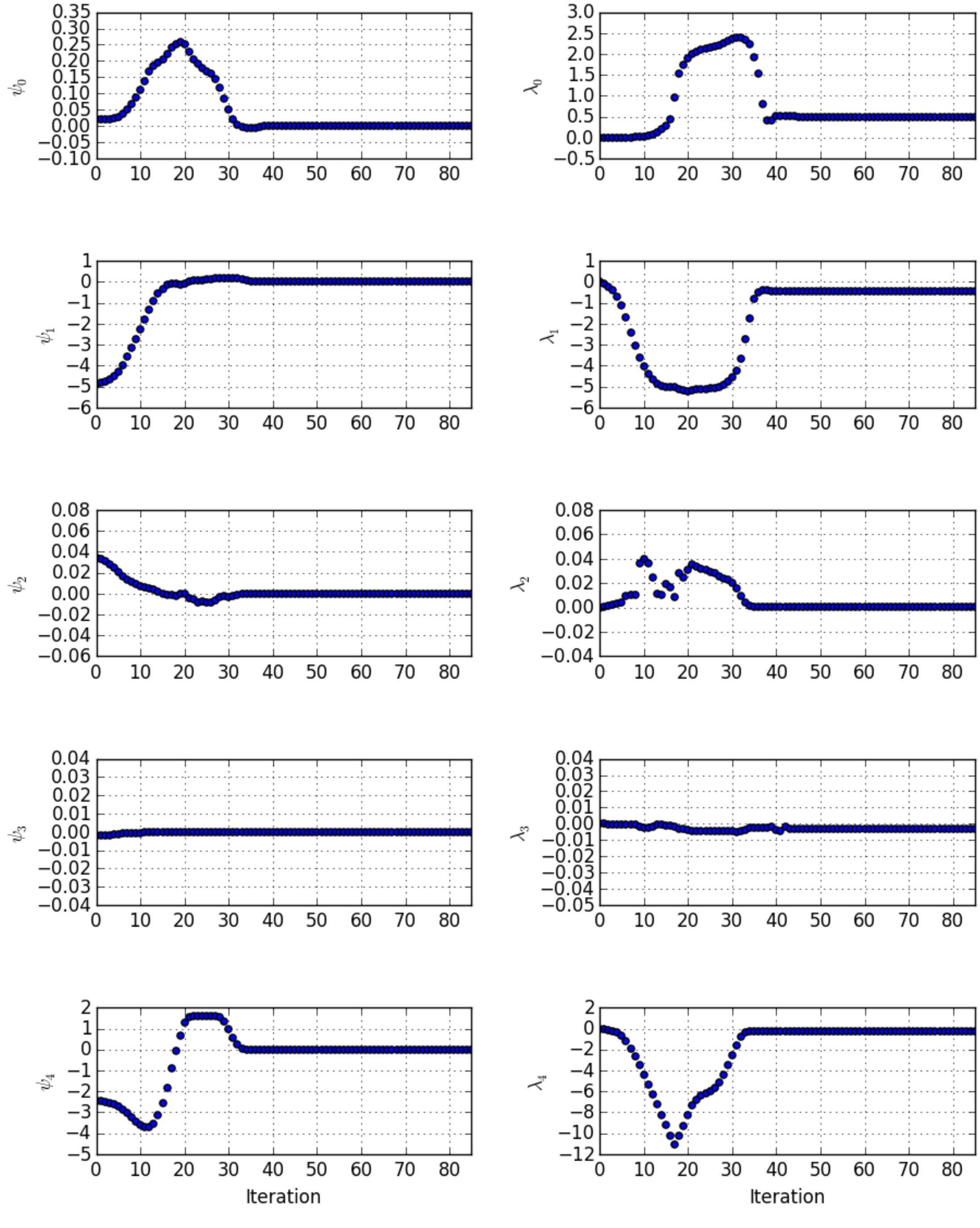
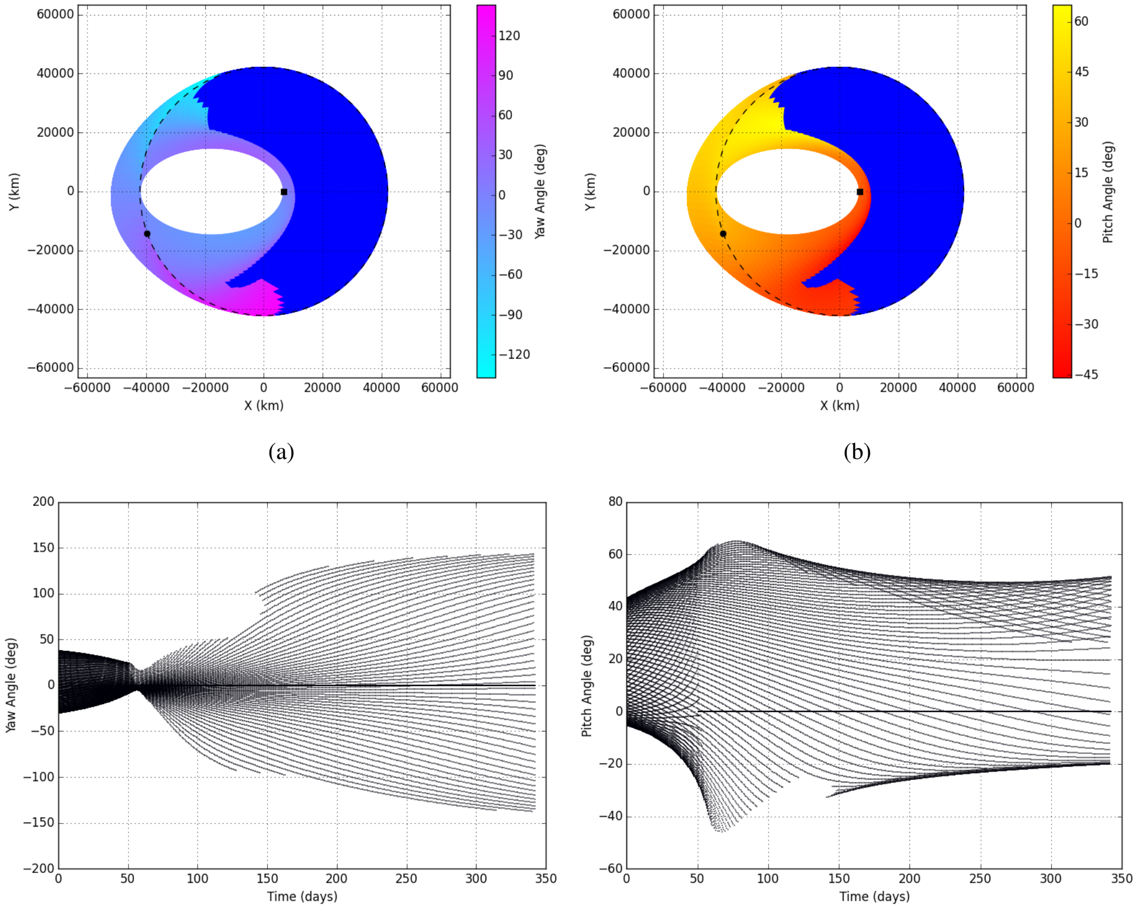


Figure 4: Iteration history of the scaled constraints and multipliers for the 450.5 revolution transfer from GTO to GEO with 2-body dynamics.



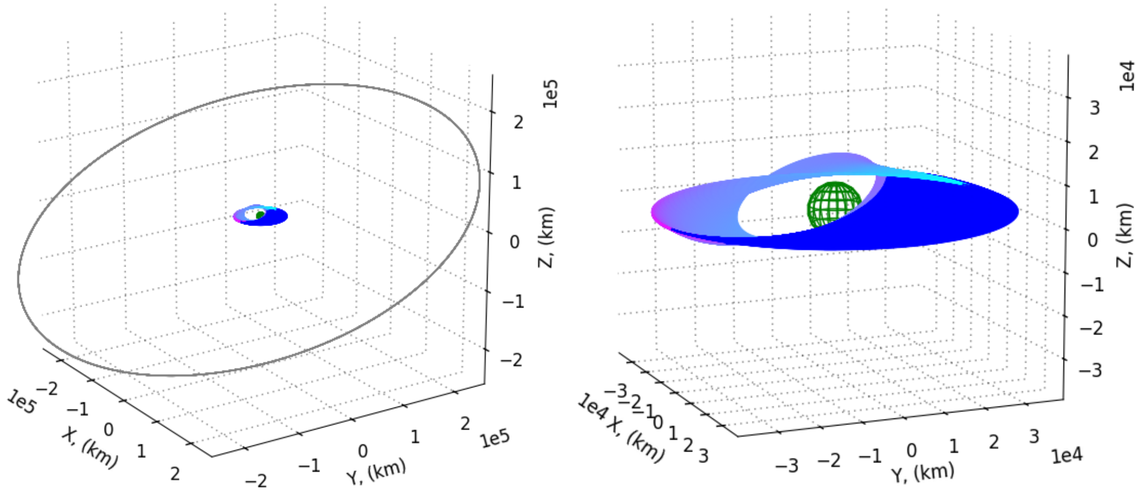
(a)

(b)

(c)

(d)

Figure 5: An equatorial projection of the 450.5 revolution transfer from GTO to GEO with 2-body dynamics and Earth's J_2 perturbation is colored by (a) yaw angle contours and (b) pitch angle contours with coast arcs in blue. Markers are placed at the initial and final states and a dashed line marks the target GEO orbit. (c) Thrust yaw angle and (d) pitch angle are shown for the duration of the transfer.



(a)

(b)

Figure 6: A three dimensional perspective of the 450.5 revolution transfer from GTO to GEO with 2-body dynamics perturbed by Earth J_2 lunar gravity (a) shows the trajectory of the Moon and (b) a closer view, colored according to thrust yaw angle.

to improve the final mass to 1745.2012 kg. The resulting time of flight is 322.63 days and final longitude is 201.0805°. Multipliers are similar to the J_2 case with the exception of λ_0, the multiplier associated with the orbital radius constraint.

An additional 25 iterations from the J_2 case are required to compute the transfer, coincidentally the same increase from the 2-body case to J_2 case. However, computer runtime sees a worse penalty. The increase to 107 minutes is owed largely to computing the Moon's orbital state in the forward pass and its derivatives in the STMs. With time included in the augmented state vector, the associated entries in the dynamics matrix and tensor include the Moon's velocity and acceleration.

Case 4: 1000.5 Revolutions With J_2 and Lunar Gravity

A final example maintains the J_2 and lunar gravity perturbed model, but increases the transfer duration to 1000.5 revolutions. Now there are 300,150 optimization variables. Reducing the penalty matrix entry σ_0, the weight on the orbital radius constraint, was found to improve performance.

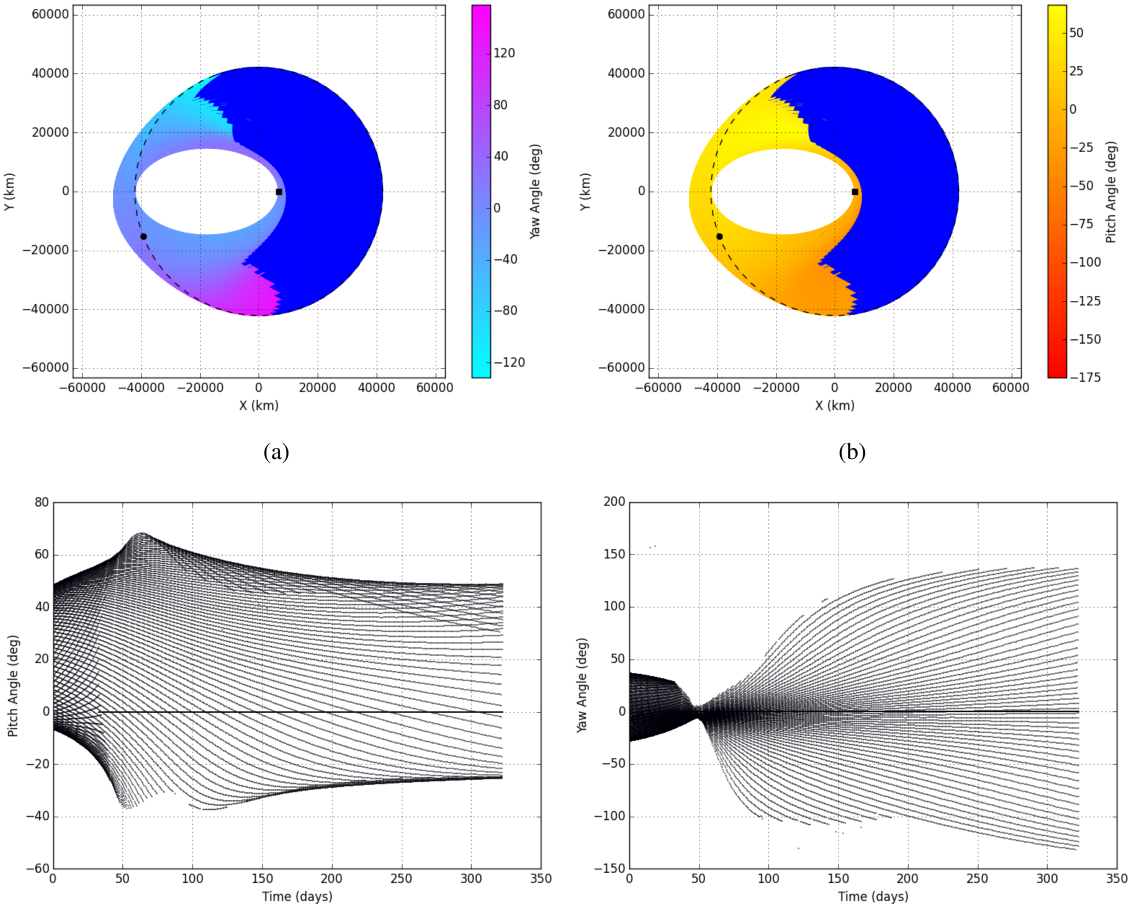
$$\Sigma = diag(10, 10, 1, 1, 1) \tag{28}$$

Perigee thrust arcs that were characteristic of the earlier transfers do not arise. Iterates did not deviate far from the target orbital radius to achieve a cheaper inclination change, so the higher penalty weight was unwarranted.

Time of flight increases with the number of revolutions to 558.86 days. The increased duration allows for more thrust arcs that are condensed around their optimal locations. The resulting final mass increases beyond that of even the 2-body case to 1784.3632 kg. Of course, the increased transfer time also allows for more precession in argument of perigee. The final longitude drifts to 276.7209°.

Thrust angle profiles in Figure 8 depict a 76 day coast arc that begins 105 days into the transfer, indicating that fixing the transfer to 1000.5 revolutions has over-prescribed the number of revolutions required. Pitch and yaw steering exhibit different behavior on either side of this long coast arc.

The number of iterations required has increased along with the problem size up to 913. Total cost and multiplier contributions $\boldsymbol{\lambda}^T\boldsymbol{\psi}$ evolve similarly to the previous cases, but the mass and feasibility history in



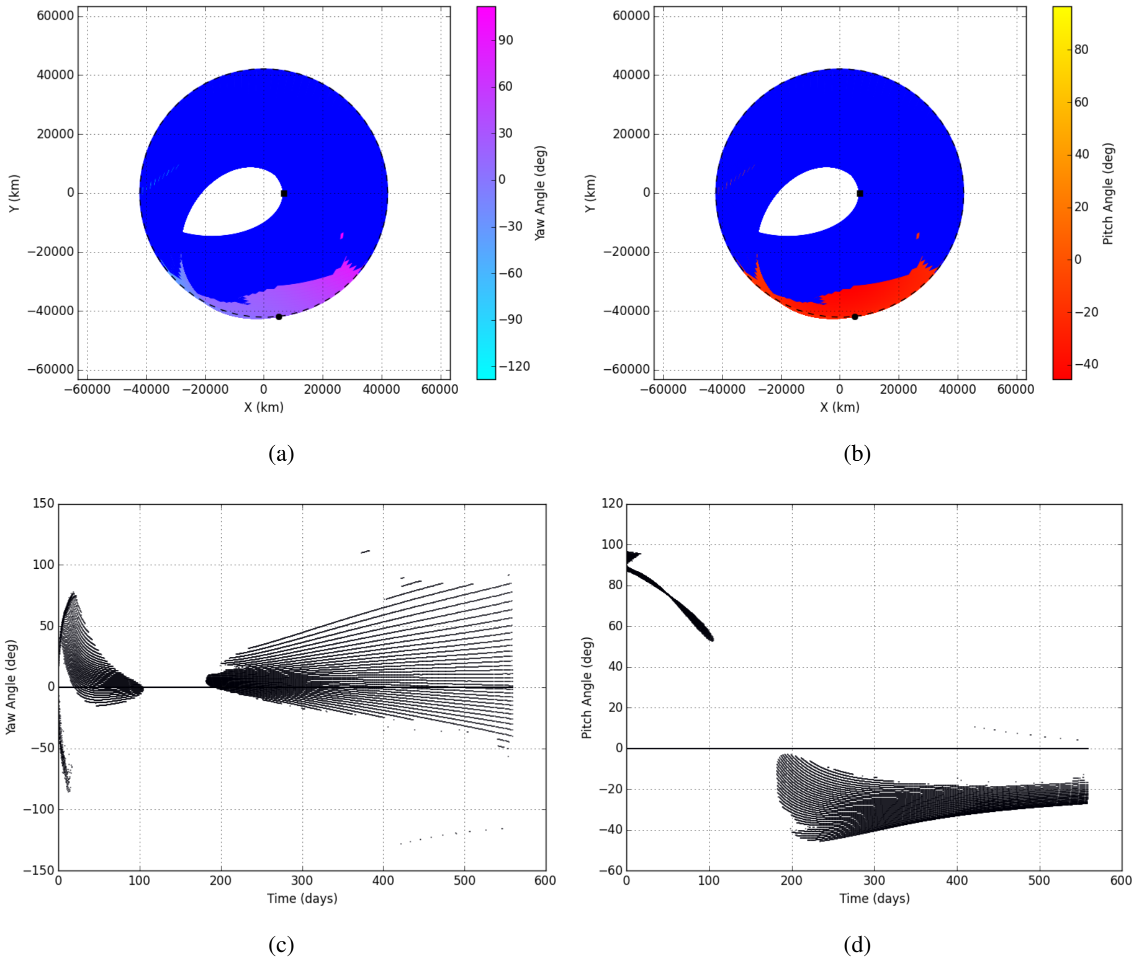
(a)

(b)

(c)

(d)

Figure 7: An equatorial projection of the 450.5 revolution transfer from GTO to GEO with 2-body dynamics perturbed by Earth J_2 and lunar gravity is colored by (a) yaw angle contours and (b) pitch angle contours with coast arcs in blue. Markers are placed at the initial and final states and a dashed line marks the target GEO orbit. (c) Thrust yaw angle and (d) pitch angle are shown for the duration of the transfer.



(a)

(b)

(c)

(d)

Figure 8: An equatorial projection of the 1000.5 revolution transfer from GTO to GEO with 2-body dynamics perturbed by Earth J_2 and lunar gravity is colored by (a) yaw angle contours and (b) pitch angle contours with coast arcs in blue. Markers are placed at the initial and final states and a dashed line marks the target GEO orbit. (c) Thrust yaw angle and (d) pitch angle are shown for the duration of the transfer.

Figure 9 suggest a step toward an intermediate solution that could not meet feasibility requirements.

The increase in computation time with problem size jumps an order of magnitude for the 1000.5 revolution case up to nearly 1 day from runtimes of 1-2 hours for the shorter cases. While the number of control variables grows linearly, there is no guarantee for the effect on convergence properties as the spacecraft trajectory problem is nonlinear and the algorithm is sensitive to tuning. A super-linear but less than quadratic increase in computational effort should be expected.

Computer memory requirements are driven by storage of the state and STMs. At each stage there are 11 state components, 11^2 STM components, and 11^3 STT components. Storing 8-byte values for 100 stages per revolution, across 1000.5 revolutions, requires 1.17 GB. Ignoring the possible savings from symmetry and sparsity patterns, the 3×3 control Hessians for each stage require just 7.2 MB, whereas a single Hessian for all 300,150 control variables would require 720.72 GB.

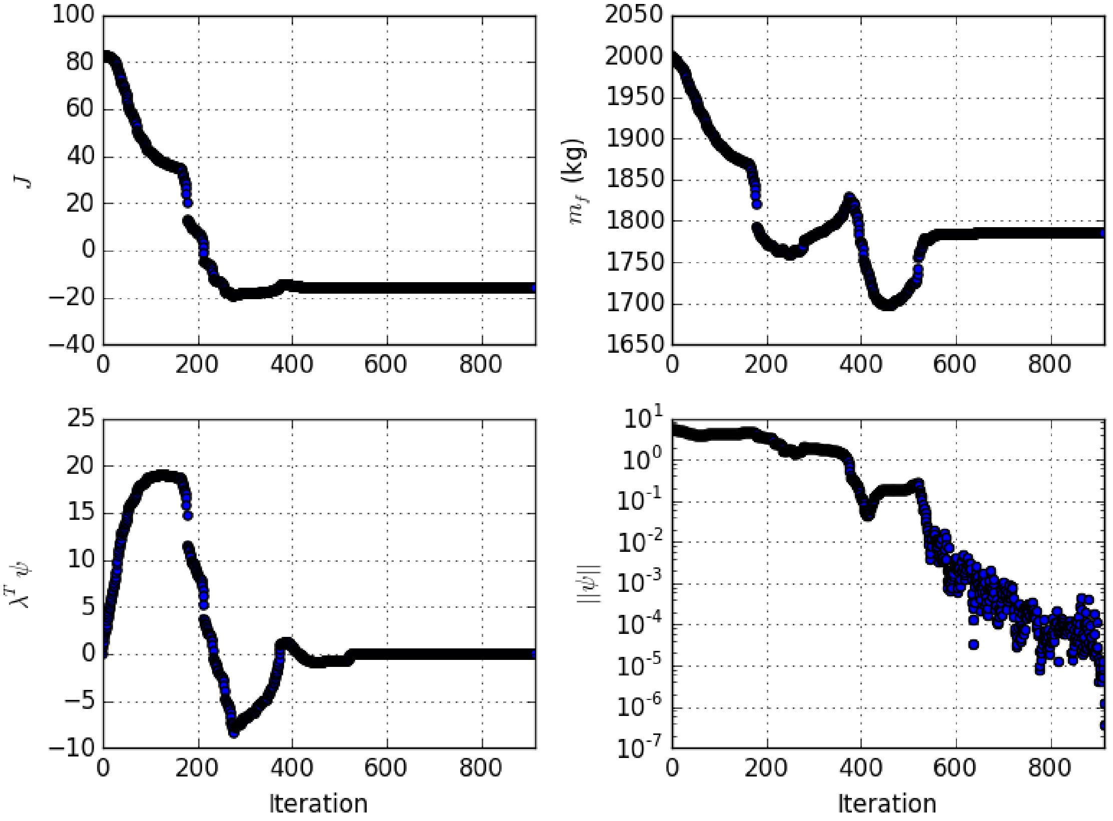


Figure 9: Iteration history of the scaled cost function, the final mass in kilograms, the scaled cost contributions from the multiplier term and the scaled norm of the constraint violations for the 1000.5 revolution transfer from GTO to GEO with 2-body dynamics perturbed by Earth J_2 and lunar gravity.

Table 3: Multipliers for Example Cases.

Perturbations	N_{rev}	λ_0	λ_1	λ_2	λ_3	λ_4
None	450.5	0.4874	-0.4205	5.4191 $\times 10^{-4}$	-2.8552 $\times 10^{-3}$	-0.2470
J_2	450.5	3.2181	-0.2713	-0.1157	-4.2530	-0.1756
J_2 and Lunar Gravity	450.5	0.9317	-0.2790	-0.0752	-2.9954	-0.1191
J_2 and Lunar Gravity	1000.5	-0.3183	-0.3045	-0.0874	-0.0153	0.1486

Table 4: Summary of GTO to GEO Results.

Perturbations	N_{rev}	Iterations	Runtime (minutes)	m_f (kg)	t_f (days)	θ (deg)
None	450.5	86	54	1759.1754	315.75	180.0
J_2	450.5	111	61	1737.1949	342.13	199.8728
J_2 and Lunar Gravity	450.5	136	107	1745.3012	322.63	201.0805
J_2 and Lunar Gravity	1000.5	913	1359	1784.3632	558.86	276.7209

CONCLUSION

This paper presents an approach to the low-thrust many-revolution spacecraft trajectory optimization problem. Differential dynamic programming is well-suited for high-dimensional problems and the hybrid differential dynamic programming algorithm has been implemented as the optimization procedure. Posing the spacecraft trajectory problem with a dependency on eccentric anomaly instead of time is accomplished via the Sundman transformation. Application of HDDP to the Sundman-transformed problem exhibits practical performance for challenging trajectories that are otherwise intractable for common approaches. The utility of this method has been demonstrated by a set of transfers from geostationary transfer orbit to geosynchronous orbit in 450.5 revolutions for dynamic models of increasing fidelity and finally for a 1000.5 revolution transfer.

The trajectories that have been presented should be viewed as local solutions for the given problem setup. A large set of tuning parameters must be selected. Different sets of scaling factors, penalty weights, and implementation in general will affect how the iterates advance toward different local optima. Discretization could be refined or relaxed, and the control more or less sophisticated.

This approach is amenable to different Sundman tranformations, thrust representations, and dynamic models. One can simply "plug in" the choice of each, given that the required first and second derivatives are available. Real-world mission design is then practical with models of a true propulsion system and true dynamics.

ACKNOWLEDGMENT

This work was supported by a NASA Space Technology Research Fellowship.

REFERENCES

[1] T. Edelbaum, "Propulsion Requirements for Controllable Satellites," *ARS Journal*, Vol. 31, August 1961, pp. 1079–1089.

[2] T. Edelbaum, "Theory of Maxima and Minima," *Optimization Techniques, With Applications to Aerospace Systems*, 1962.

[3] W. E. Wiesel and S. Alfano, "Optimal Many-Revolution Orbit Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 8, 1985, pp. 155–157.

[4] J. A. Kéchichian, "Reformulation of Edelbaum's Low-Thrust Transfer Problem Using Optimal Control Theory," *Journal of Guidance, Control, and Dynamics*, Vol. 20, September 1997, pp. 988–994.

[5] T. N. Edelbaum, "Optimum Low-Thrust Rendezvous and Station Keeping," *AIAA Journal*, Vol. 2, No. 7, 1964, pp. 1196–1201.

[6] J. A. Kéchichian, "Optimal Low-Thrust Rendezvous Using Equinoctial Orbit Elements," *Acta Astronautica*, Vol. 38, No. 1, 1996, pp. 1–14.

[7] J. A. Kéchichian, "Optimal Low-Thrust Transfer in General Circular Orbit Using Analytic Averaging of the System Dynamics," *The Journal of the Astronautical Sciences*, Vol. 57, January-June 2009, pp. 369–392.

[8] J. A. Kéchichian, "Inclusion of Higher Order Harmonics in the Modeling of Optimal Low-Thrust Orbit Transfer," *The Journal of the Astronautical Sciences*, Vol. 56, January-March 2008, pp. 41–70.

[9] J. A. Kéchichian, "Orbit Raising with Low-Thrust Tangential Acceleration in Presence of Earth Shadow," *Journal of Spacecraft and Rockets*, Vol. 35, July-August 1998, pp. 516–525.

[10] J. A. Kéchichian, "Low-Thrust Inclination Control in Presence of Earth Shadow," *Journal of Spacecraft and Rockets*, Vol. 35, July-August 1998, pp. 526–532.

[11] C. A. Kluever, "Simple Guidance Scheme for Low-Thrust Orbit Transfers," *Journal of Guidance, Control, and Dynamics*, Vol. 21, November 1998, pp. 1015–1017.

[12] D. E. Chang, D. F. Chichka, and J. E. Marsden, "Lyapunov-Based Transfer Between Elliptic Keplerian Orbits," *Discrete and Continuous Dynamical Systems-Series B*, Vol. 2, February 2002, pp. 57–67.

[13] A. E. Petropoulos, "Simple Control Laws for Low-Thrust Orbit Transfers," *AAS/AIAA Astrodynamics Specialist Conference*, August 2003.

[14] A. E. Petropoulos, "Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August 2004.

[15] J. T. Betts, "Practical Methods for Optimal Control and Estimation using Nonlinear Programming," *Society for Industrial and Applied Mathematics*, 2010.

[16] J. T. Betts, "Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Optimal control, International Series of Numerical Mathematics* (R. Bulirsch, A. Miele, J. Stoer, and K. Well, eds.), Vol. 117, Birkhäuser Basel, 1993.

[17] J. T. Betts, "Sparse Optimization Suite, SOS, User's Guide, Release 2015.11," http://www.appliedmathematicalanalysis.com/downloads/sosdoc.pdf. [Online; accessed November-2016].

[18] J. T. Betts, "Very Low-Thrust Trajectory Optimization Using a Direct SQP Method," *Journal of Computational and Applied Mathematics*, Vol. 120, August 2000, pp. 27–40.

[19] J. T. Betts and S. O. Erb, "Optimal Low Thrust Trajectories to the Moon," *SIAM Journal on Applied Dynamical Systems*, Vol. 2, May 2003, pp. 144–170.

[20] J. T. Betts, "Optimal Low Thrust Orbit Transfers with Eclipsing," *Optimal Control Applications and Methods*, Vol. 36, February 2014.

[21] D. H. Jacobson and D. Q. Mayne, "Differential Dynamic Programming," *American Elsevier Publishing Company, Inc.*, 1970.

[22] R. E. Bellman, "Dynamic Programming," *Princeton University Press*, 1957.

[23] N. T. T. Program, "Mystic Low-Thrust Trajectory Design and Visualization Software," https://software.nasa.gov/software/NPO-43666-1. [Online; accessed October-2016].

[24] 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*, Vol. 58, April 2006, pp. 605–616.

[25] G. J. Whiffen, "Static/Dynamic Control for Optimizing a Useful Objective," *No. Patent 6496741*, December 2002.

[26] 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*, August 2006.

[27] 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*, Vol. 154, No. 2, 2012, pp. 382–417.

[28] 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*, Vol. 154, No. 2, 2012, pp. 418–442.

[29] G. Lantoine and R. P. Russell, "A Methodology for Robust Optimization of Low-Thrust Trajectories in Multi-Body Environments," *Ph.D. Thesis*, 2010.

[30] K. Sundman, "Memoire sur le probleme des trois corps," *Acta Math*, Vol. 36, 1913, pp. 105–179, doi:10.1007/BF02422379.

[31] G. Jannon and V. R. Bond, "The Elliptic Anomaly," *NASA Technicai Memorandum 58228*, April 1980.

[32] C. H. Yam, D. D. Lorenzo, and D. Izzo, "Towards a High Fidelity Direct Transcription Method for Optimisation of Low-Thrust Trajectories," *International Conference on Astrodynamics Tools and Techniques - ICATT*, 2010.

[33] J. A. Sims and S. N. Flanagan, "Preliminary Design of Low-Thrust Interplanetary Missions," *AAS/AIAA Astrodynamics Specialist Conference*, August 1999.

[34] 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*, Vol. 118, April 2014, pp. 355–378.

[35] J. D. Aziz, J. S. Parker, and J. A. Englander, "Hybrid Differential Dynamic Programming With Stochastic Search," *AAS/AIAA Space Flight Mechanics Meeting*, February 2016.

[36] P. Prince and J. Dormand, "High order embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, Vol. 7, Issue 1, March 1981, pp. 67–75.

[37] O. A. R. Board, "OpenMP Application Program Interface Version 3.0," May 2008.