AAS 16-239

GLOBAL OPTIMIZATION OF LOW-THRUST INTERPLANETARY TRAJECTORIES SUBJECT TO OPERATIONAL CONSTRAINTS

Jacob A. Englander[*]
Matthew A. Vavrina[†]
David Hinckley[‡]

Low-thrust interplanetary space missions are highly complex and there can be many locally optimal solutions. While several techniques exist to search for globally optimal solutions to low-thrust trajectory design problems, they are typically limited to unconstrained trajectories. The operational design community in turn has largely avoided using such techniques and has primarily focused on accurate constrained local optimization combined with grid searches and intuitive design processes at the expense of efficient exploration of the global design space. This work is an attempt to bridge the gap between the global optimization and operational design communities by presenting a mathematical framework for global optimization of low-thrust trajectories subject to complex constraints including the targeting of planetary landing sites, a solar range constraint to simplify the thermal design of the spacecraft, and a real-world multi-thruster electric propulsion system that must switch thrusters on and off as available power changes over the course of a mission.

INTRODUCTION

Preliminary design of interplanetary missions is frequently labor intensive and computationally expensive. The designer must choose the launch date, flight time, propulsive maneuvers, and possibly a sequence of planetary flybys as well as altitudes and velocity vectors for each of those flybys. Low-thrust missions add an additional degree of complexity over their impulsive counterparts because the designer must also choose a time history of control variables, *i.e.* thrust magnitude and direction for months, if not years, of continuous thrusting. The additional design variables often introduce many families of locally optimal solutions and it is challenging to find the global optimum.

Real-world interplanetary missions are often subject to complex operational constraints. For example, a spacecraft may be required to land on the surface of a planet with both latitude and longitude specified. In some cases the landing might be required to occur at a particular time of local day, or the spacecraft may be required to approach the landing site from a particular azimuth. Such constraints can significantly impact the terminal velocity vector of the spacecraft relative to the planet and therefore have cascading effects throughout the entire trajectory. Other possible constraints include a distance constraint with respect to other bodies in the solar system such as a minimum distance from the sun for thermal reasons or a maximum distance from the Earth for communications reasons.

In addition, the propulsion and power systems of current low-thrust spacecraft add complexity to the mission design process. One particularly challenging example of this is operation of a multi-thruster power-limited propulsion system. Electric thrusters typically have a minimum power at which they may be turned on, a maximum power that may be accommodated by the power processing unit (PPU), and a performance model that describes their function between those bounds. If a spacecraft carries multiple thrusters then

[*] Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center, Member AAS
[†] Aerospace Engineer, a.i. Solutions
[‡] PhD Student, University of Vermont

at various points along the trajectory, when the available power decreases and increases, thrusters must be switched on and off. Further, there is some distance from the sun at which no thrusters may be fired at all. Thruster switching creates a step function in the curves of available thrust and mass flow rate that characterize the propulsion system. Since many low-thrust optimization techniques are gradient based, this step function significantly impacts the robustness of the optimizer and must be accounted for in any robust global optimization technique. It is possible to circumvent this problem by modeling the propulsion system as a single "super engine" with no discontinuities, but as the required fidelity of the analysis increases, this approach ceases to be an option [1].

The above constraints, and many others, must be satisfied by any feasible trajectory design. Historically there have been several techniques to ensure feasibility. One approach is to solve the global unconstrained problem and then use that solution as an initial guess for a constrained local optimization. Unfortunately, the globally optimal solution to the unconstrained problem may not be in the neighborhood of the globally optimal solution to the constrained problem. Another approach is to solve many versions of the unconstrained problem and filter away all of those that do not satisfy the constraints. However, this approach is extremely computationally expensive and will prune away trajectories that only mildly violate the constraints and could easily be made to satisfy them. This scenario often occurs in low-thrust optimization because small changes to the time history of thrust control variables can cause constraints to be satisfied for only a modest cost in propellant and/or flight time.

Several techniques exist to find locally optimal solutions to low-thrust problems. These techniques are generally separable into two categories: direct and indirect methods. Indirect methods are based on the calculus of variations [2]. The trajectory design problem is formulated in terms of the Lagrange multipliers and analytical necessary conditions. The sufficient conditions for optimality are then derived. A boundary value problem is then solved to find the values of the Lagrange multipliers that satisfy the necessary conditions and sufficient conditions. This becomes increasingly difficult as problem complexity increases because the sensitivity of the problem to the Lagrange multipliers also increases.

In direct methods, the decision variables to be optimized are the control values themselves rather than Lagrange multipliers. A nonlinear programming (NLP) problem is then solved to find a set of control variables that satisfies problem-specific control constraints, path constraints, and time constraints while minimizing an objective function. There are many varieties of direct methods, including collocation [3–6], direct parallel shooting [7], and two-point boundary shooting [8, 9]. Because all of the direct methods leverage a NLP problem solver, all can be made to handle operational constraints.

Both indirect and direct methods, because they are dependent on a local solver, require an initial guess and will tend to converge to a solution in the vicinity of the initial guess. If multiple locally optimal solutions exist, then it is useful to find an initial guess in the vicinity of the global optimum. Direct methods tend to be less sensitive to the choice of an initial guess than indirect methods, but this problem must be addressed for both approaches. These initial guesses are usually developed using a low-fidelity model of the low-thrust trajectory. Then, once an initial guess is found, the low-thrust trajectory optimization problem is usually solved in a medium-fidelity model so that many options can be compared. Finally one or more of the medium-fidelity trajectories is chosen to be re-designed in a high fidelity force model. Each of these three design stages is very labor intensive and biased toward the designer's intuition. An automated approach that can efficiently explore the medium-fidelity design space and take into account non-intuitive solutions is therefore desirable.

Several global optimization techniques have been developed for interplanetary low-thrust trajectories. Many researchers have investigated the class of techniques known as "shape-based methods," in which the low-thrust trajectory is approximated as a geometric shape and the control history necessary to follow that trajectory is derived as a function of the shape. Various shape-based methods have expressed low-thrust trajectories as exponential sinusoids [10], inverse polynomials [11], Fourier series [12], or a variety of other shapes [13, 14]. All of these techniques enable rapid generation of approximate low-thrust trajectories when combined either with a grid search or a population based metaheuristic but unfortunately all of them, because they are built for speed rather than fidelity and flexibility, cannot easily handle the operational constraints posed here.

One promising technique that could merge the modeling of operational constraints with the flexibility of global optimization was introduced by Coverstone *et al.* [15] and then further addressed by Vavina *et al.* [16] and Yam *et al.* [17], who expressed low-thrust design problem as a constrained global optimization problem by combining a gradient-based solver with a stochastic search algorithm. All of these works considered only trajectory continuity and control magnitude constraints but are generalizable to the additional constraints considered in this work.

In this work, we demonstrate how several critical mission design constraints, including the targeting of planetary landing trajectories, solar distance, and power-limited multi-thruster spacecraft can be expressed mathematically such that they may be incorporated into a global trajectory optimization framework. Two example missions are presented that exercise these constraints - a low-thrust version of the OSIRIS-REx mission and a notional comet rendezvous. All of the constraints and models presented in this work are incorporated into the Evolutionary Mission Trajectory Generator (EMTG), NASA Goddard's open-source medium-fidelity interplanetary mission design tool [18].

MODELING

Multiple Gravity Assist with Low-Thrust

The trajectory model employed in this work is called multiple gravity assist with low-thrust (MGALT) and is derived from the well-known Sims-Flanagan transcription [8] in which the continuous-thrust trajectory is discretized into many small time steps, and the thrust is approximated as a small impulse occurring at the center of the time step. The trajectory is propagated between control points by solving Kepler's problem [8]. The Sims-Flanagan transcription, when used with a NLP solver such as Sparse Nonlinear Optimizer (SNOPT) and a suitable initial guess, is very fast and robust. It is considered to be a "medium-fidelity" transcription and is used in software packages such as EMTG [18], Mission Analysis Low-Thrust Optimization (MALTO) [19], Gravity Assisted Low-thrust Local Optimization Program (GALLOP) [20], and Parallel Global Multiobjective Optimizer (PaGMO) [21].

In the classical Sims-Flanagan transcription, the optimizer chooses the three components of an impulsive $\Delta\mathbf{v}$ vector at the center of each time-step. In order to improve the robustness of the solver, a modified transcription known as "up-to-unit vector control" is used in this work, where instead of choosing the $\Delta\mathbf{v}$ vector directly the optimizer instead chooses a control 3-vector in $[-1.0, 1.0]$ that is multiplied by the maximum Δv that the spacecraft can produce in that time-step. The magnitude of the control vector is bounded in the range $[0.0, 1.0]$, i.e.,

$$\Delta\mathbf{v}_i = \mathbf{u}_i \Delta v_{max,i}, \|\mathbf{u}_i\| \leq 1.0 \tag{1}$$

where

$$\Delta v_{max,i} = \frac{D n_{available} T_{\max} \left(t_f - t_0\right)}{mN} \tag{2}$$

where D is the thruster duty cycle, $n_{available}$ is the number of available thrusters, $T_{\max}$ is the maximum available thrust from one thruster, t_0 and t_f are the beginning and ending times of the time step, N is the number of time steps in the phase, and m is the mass of the spacecraft at the center of the time step. This modified Sims-Flanagan transcription is used in MALTO, PaGMO, and in this work.

The spacecraft state is propagated forward from the first endpoint (*e.g.* planet) in each phase and backward from the second endpoint. The trajectory is propagated by solving Kepler's equation and the spacecraft mass is propagated by assuming a constant mass flow rate across the each time-step. The specific Kepler propagator algorithm used here is a Laguerre-Conway method [22, 23]. A set of nonlinear constraints are applied to ensure continuity in the center of the phase,

$$\mathbf{s}_{mf} - \mathbf{s}_{mb} = \left[\begin{array}{ccccccc} \Delta x & \Delta y & \Delta z & \Delta v_x & \Delta v_y & \Delta v_z & \Delta m \end{array} \right] = \epsilon \tag{3}$$

where ϵ is a very small tolerance value.

The optimizer also chooses the initial and final velocity vectors for each phase. If a phase begins with a launch, the magnitude of the initial velocity vector is used with a launch vehicle model to determine the

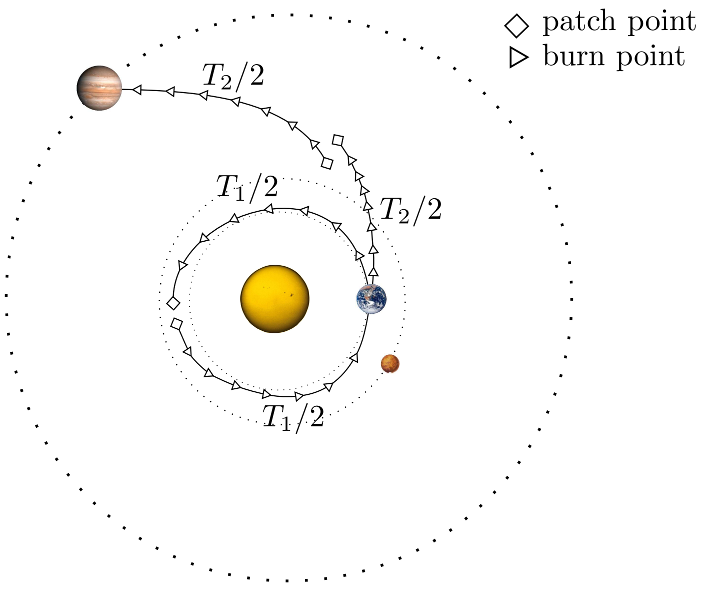


Figure 1: An Example Trajectory Using the Sims-Flanagan Transcription

initial mass of the spacecraft as described later in this work. If a phase begins with a planetary flyby, two nonlinear constraints are applied to ensure that the flyby is feasible. First, the incoming and outgoing velocity magnitudes with respect to the planet must be equal,

$$v_\infty^+ - v_\infty^- = 0 \tag{4}$$

where v_∞^- and v_∞^+ are the velocity magnitudes before and after the flyby, respectively. Second, the spacecraft may not fly closer to the planet than some user-specified minimum flyby distance:

$$\frac{\mu_{planet}}{v_\infty^2}\left[\frac{1}{\sin(\frac{\delta}{2})} - 1\right] - (r_{planet} + h_{safe}) \geq 0 \tag{5}$$

where

$$\delta = \arccos\left[\frac{\mathbf{v}_\infty^- \cdot \mathbf{v}_\infty^+}{\left(v_\infty^-\right)^2}\right] \tag{6}$$

Here μ_{planet} is the gravitational parameter of the planet, r_{planet} is the radius of the planet, δ is the flyby turn angle, and h_{safe} is the user-defined minimum altitude.

Figure 1 is a diagram of a simple low-thrust mission to Jupiter with one Earth flyby using the MGALT model. The continuity constraints are deliberately left unsatisfied in the diagram to illustrate where they must be applied.

There are four significant advantages to using the Sims-Flanagan transcription. First, the optimal objective function value for a Sims-Flanagan trajectory design is usually very close to the optimal cost value for a higher-fidelity version of the same trajectory. Second, a low-thrust trajectory generated using the Sims-Flanagan transcription makes a very good initial guess for a higher-fidelity trajectory design. Third, the Sims-Flanagan transcription is very fast because it does not require numerical integration of differential equations. Fourth, the convergence of an NLP solver solving a Sims-Flanagan problem is very robust to poor initial guesses, making it ideal for an automated design approach. When analytical derivatives for the constraints with respect to the decision variables are supplied as is the case in this work, speed and robustness are even better than described above.

Operational Constraints

We will now consider several operational constraints that a mission design may have to obey. Often these constraints are omitted in the initial broad search for trajectories and then either applied in a second round of optimization or used as filters to post-process the unconstrained results. However there are flaws to these approaches. First, there is no guarantee that the optimal solution to the constrained problem is in the vicinity of the solution to the unconstrained problem and so this method may introduce "false positives" for which much time is spent searching an area of the decision space where there is no constrained solution. Conversely, the method of solving only the unconstrained problem and then pruning away *a posteriori* solutions that violate the operational constraints can have the unintended consequence of pruning away solutions that are so nearly feasible that a constrained optimization process could easily render them acceptable. This circumstance can result in "false negatives," *i.e.* cases where the solver algorithm finds no feasible solutions. Both approaches can bias the search toward regions of the decision space that may not contain the desired solution.

In this work, the operational constraints are imposed directly in the global search problem so that no re-optimization or filtering is necessary.

Distance Constraint

The first operational constraint considered here is a minimum and/or maximum value imposed on the distance between the spacecraft and bodies in the solar system, *i.e.*,

$$d_{LB} \leq r_{s/c-body} \leq d_{UB} \tag{7}$$

where d_{LB} and d_{UB} are defined by the analyst for each problem and for each body. For example, the spacecraft may be constrained to never get too close to the sun for thermal reasons, or may not be allowed to fly farther away from the Earth than some maximum distance for communications reasons. These constraints occur often in real-world mission design, especially in low-thrust design where the desire to prevent the spacecraft from growing too hot is in conflict with the availability of more power and therefore more efficient propulsion closer to the sun.

The distance constraint is very easy to pose in the optimization problem because it requires only looking up the position of the relevant solar system bodies at each time-step in the trajectory. However it is quite computationally expensive for two reasons. The first reason is that each ephemeris lookup requires a call an ephemeris database, which is quite slow. The second reason is that computing analytical derivatives of the distance constraint requires recursive multiplications of the state transition matrices (STMs) and maneuver transition matricies (MTMs) along the trajectory. These computations are omitted from this paper in the interest of brevity but may be found in the open-source code associated with this work [18]. In a large problem with many time-steps, over 50% of the execution time for the trajectory optimization is consumed by the derivative calculation code for the distance constraint.

Planetary Landing Constraint

The next set of constraints describe atmospheric entry and landing in a patched-conic framework. For atmospheric entry, the user specifies the latitude of the atmospheric entry interface L_{EI}, the flight path angle γ_{EI} relative to a notional tangential velocity vector, and the radius (i.e. distance from the center of the body) of the entry r_{EI}. If the target is a body without an atmosphere, then L_{EI}, γ_{EI}, and r_{EI} may be used to define the position of an intersection with the body's surface instead. These values, along with the incoming $\mathbf{v}_\infty$, are used to compute the position and velocity vectors at interface $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$. Various constraints may then be applied to those vectors as described later in this section.

The entry/landing interface constraints are applied in the reference frame of the body, so the first step is to rotate $\mathbf{v}_\infty$ from the International Celestial Reference Frame (ICRF), which is the calculation frame for the interplanetary trajectory, to a body-centered inertial (BCI) frame that has $\hat{\mathbf{z}}$ aligned with the body's spin pole, and $\hat{\mathbf{x}}$ aligned with the first point in the constellation Ares. Note that if the target body is the Earth then ICRF and BCI are the same frame.

$$\mathbf{v}_{\infty-BCI} = \mathbf{R}_{ICRF-to-BCI}\mathbf{v}_{\infty-ICRF} \tag{8}$$

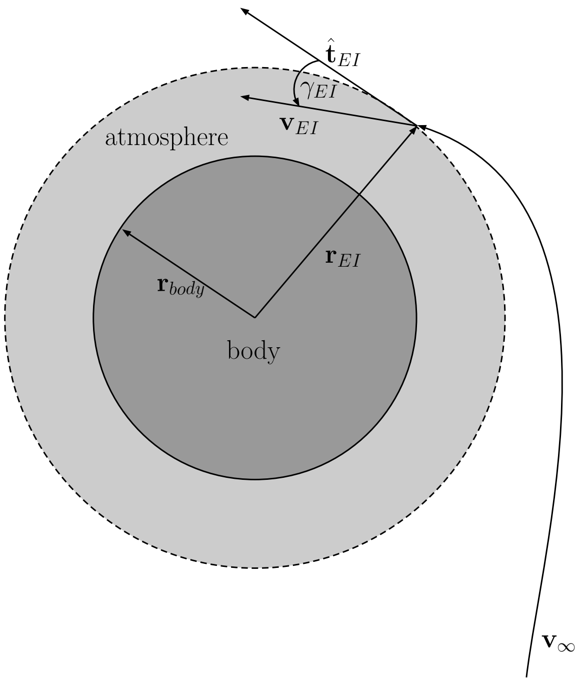


Figure 2: Definition of r_{EI} and γ_{EI}

where $\mathbf{R}_{ICRF-to-BCI}$ the rotation matrix from the ICRF to the body's equatorial frame as defined by the IAU working group on cartographic coordinates [24]. The next step is to calculate the periapse radius of the inbound hyperbola,

$$r_p = \left(\frac{\mu_{body}}{v_\infty^2}\right)\left(-1.0 + \left(1.0 + \frac{v_\infty^2 r_{EI}}{\mu_{body}}\cos^2\gamma_{EI}\left(2.0 + \frac{v_\infty^2 r_{EI}}{\mu body}\right)\right)^{1/2}\right) \tag{9}$$

where r_{EI} is the interface radius and γ_{EI} is the flight path angle as described in Figure 2.

The next step is to determine β_{EI}, the range angle between the entry/landing interface and the incoming hyperbolic asymptote. β_{EI}, described in Figure 3, defines the position of the interface point on the approach hyperbola and is key to finding the position and velocity vectors $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$. First the true anomalies of the hyperbolic asymptote ν_∞ and the entry/landing interface point ν_{EI} are computed,

$$\nu_\infty = -\arccos\left(-1.0/\left(1.0 + \frac{v_\infty^2 r_p}{\mu_{body}}\right)\right) \tag{10}$$

$$\nu_{EI} = \arccos\left(\frac{r_p}{r_{EI}}\left(2.0 + \frac{r_p v_\infty^2}{\mu_{body}}\right) - 1.0\right) / \left(1.0 + \frac{r_p v_\infty^2}{\mu_{body}}\right) \tag{11}$$

Finally, the range angle β_{EI} may be computed as:

$$\beta_{EI} = -\nu_\infty - \nu_{EI} \tag{12}$$

Once β_{EI} is known, it is possible to compute $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$. We compute $\mathbf{r}_{EI}$ first. There are two possible locations for $\mathbf{r}_{EI}$, a prograde solution $\mathbf{r}_{EI-prograde}$ and a retrograde solution $\mathbf{r}_{EI-retrograde}$. Both of these points are defined by the intersection of plane that passes through the user-defined interface latitude L_{EI} and the circle of accessible entry points defined by the geometry of the inbound hyperbola as described in Figure 5. The latitude plane is parallel to the body equatorial plane at a distance z_L as shown in Figure 4,

$$z_L = r_{EI}\sin L \tag{13}$$

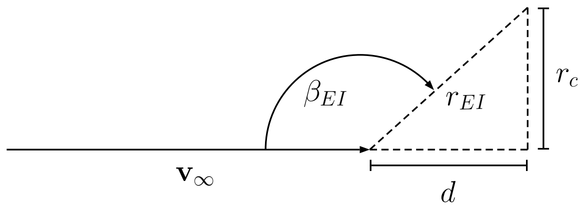


Figure 3: Definition of β_{EI}

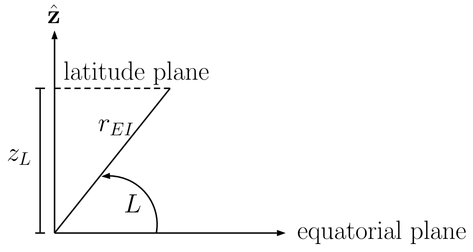


Figure 4: Definition of the latitude plane

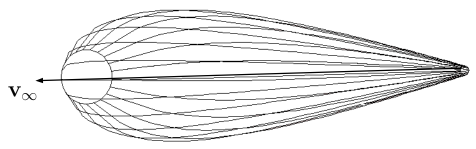


Figure 5: Circle of accessible points defined by v_∞ and examples of possible hyperbolic paths. Only two of the continuum of paths shown here intersect the latitude plane.

Before finding either $\mathbf{r}_{EI-prograde}$ or $\mathbf{r}_{EI-retrograde}$, it is necessary to find the vector $\mathbf{q}$ from the origin that defines the point midway between $\mathbf{r}_{EI-prograde}$ and $\mathbf{r}_{EI-retrograde}$. Conveniently, the projection of $\mathbf{q}$ into the body equatorial frame is aligned with the equatorial projection of $\mathbf{v}_\infty$ (EPV). It is therefore possible to define $\mathbf{q}$ in the plane defined by the body spin pole vector $\hat{\mathbf{z}}$ and $\widehat{\mathbf{EPV}}$. The first step in this process is to find the vector $\mathbf{d}$, the projection of $\mathbf{q}$ onto $\hat{\mathbf{v}}_\infty$ as shown in Figure 6.

$$\mathbf{d} = d\hat{\mathbf{v}}_\infty \tag{14}$$

$$d = r_{EI}\cos(\pi - \beta_{EI}) \tag{15}$$

The vector from $\mathbf{d}$ to $\mathbf{q}$ may then be defined,

$$\mathbf{g} = m(\mathbf{n} \times \mathbf{d}) \tag{16}$$

$$\mathbf{n} = \mathbf{v}_\infty \times \hat{\mathbf{z}} \tag{17}$$

where m is defined such that the $\hat{\mathbf{z}}$ component of $\mathbf{q}$ is equal to z_L, *i.e.*,

$$m = \frac{z_L - d_z}{n_x d_y - d_x n_y} \tag{18}$$

Once $\mathbf{d}$ and $\mathbf{g}$ are known, $\mathbf{q}$ may be computed as

$$\mathbf{q} = \mathbf{d} + \mathbf{g} \tag{19}$$

The candidate entry points $\mathbf{r}_{EI-prograde}$ and $\mathbf{r}_{EI-retrograde}$ are defined relative to $\mathbf{q}$ and are constructed by translating half a chord length of the circle of accessible entry points at a distance g from d and aligned normal to the $\hat{\mathbf{z}}$-$\hat{\mathbf{v}}_\infty$ plane, *i.e.* along $\hat{n}$ as described in Figure 7. The position of the entry points $\mathbf{r}_{EI-prograde}$ and $\mathbf{r}_{EI-retrograde}$ is given relative to $\mathbf{g}$ as $\mathbf{f}$,

$$\mathbf{f} = f\hat{\mathbf{n}} \tag{20}$$

$$f = \left(r_c^2 + g^2\right)^{1/2} \tag{21}$$

$$r_c = r_{EI}\sin(\pi - \beta_{EI}) \tag{22}$$

$\mathbf{r}_{EI-prograde}$ and $\mathbf{r}_{EI-retrograde}$ may then be defined as,

$$\mathbf{r}_{EI-prograde} = \mathbf{q} + \mathbf{f} \tag{23}$$

$$\mathbf{r}_{EI-retrograde} = \mathbf{q} - \mathbf{f} \tag{24}$$

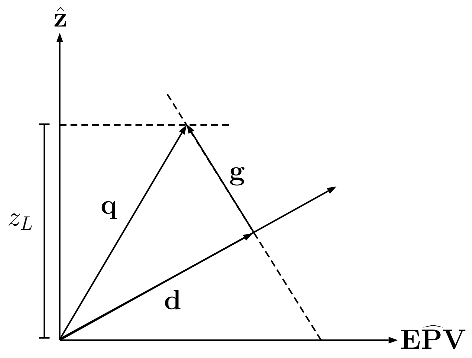


Figure 6: Definition of the vectors $\mathbf{d}$, $\mathbf{g}$, and $\mathbf{q}$

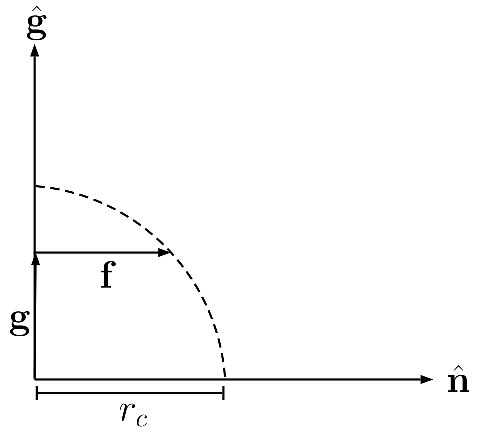


Figure 7: Definition of $\mathbf{f}$ in the $\hat{\mathbf{g}} - \hat{\mathbf{n}}$ plane

In practice the prograde solution is often chosen because the velocity vector of the spacecraft relative to the rotating atmosphere or body surface is lower. It is therefore reasonable to set $\mathbf{r}_{EI}$ equal to $\mathbf{r}_{EI-prograde}$ for the remainder of this analysis.

Once $\mathbf{r}_{EI}$ is found, $\mathbf{v}_{EI}$ may be constructed as

$$\mathbf{v}_{EI} = v_{EI}\hat{\mathbf{v}}_{EI} \tag{25}$$

The magnitude v_{EI} is easily found from the vis-viva equation,

$$v_{EI} = \left(\mu_{body}\left(\frac{2}{r_{EI}} - \frac{1}{a}\right)\right)^{1/2} \tag{26}$$

where for a hyperbola,

$$a = -\frac{\mu_{body}}{v_\infty^2} \tag{27}$$

The direction of interface velocity $\hat{\mathbf{v}}_{EI}$ may be found first by finding the unit vector $\hat{\mathbf{t}}_{EI}$ orthogonal to $\mathbf{r}_{EI}$ in the plane of the inbound hyperbola and then rotating about the orbital angular momentum unit vector $\hat{h}$ by the interface flight path angle γ_{EI} as described in Figure 2,

$$\hat{\mathbf{t}}_{EI} = \hat{\mathbf{h}} \times \hat{\mathbf{v}}_\infty \tag{28}$$

where

$$\hat{\mathbf{h}} = \hat{\mathbf{v}}_\infty \times \hat{\mathbf{r}}_{EI} \tag{29}$$

Finally $\hat{\mathbf{v}}_{EI}$ is computed by rotation,

$$\hat{\mathbf{v}}_{EI} = \mathbf{R}\left(\hat{\mathbf{h}}, -\gamma_{EI}\right)\hat{\mathbf{t}}_{EI} \tag{30}$$

where $\mathbf{R}\left(\hat{\mathbf{h}}, -\gamma_{EI}\right)$ is the rotation matrix about $\hat{\mathbf{h}}$ by angle $-\gamma_{EI}$.

The construction of $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$ has been validated by computing several test entry interface states and integrating the equations of motion backward in time in the high-fidelity modeling tool General Mission Analysis Toolkit (GMAT) [25] to verify that the correct incoming hyperbolic $\mathbf{v}_\infty$ is obtained.

Entry/Landing Azimuth Constraint It is sometimes desirable to constrain the entry/landing azimuth $AZ_{arrival}$, *i.e.* the angle from the body's spin pole to the arrival hyperbola at the moment of interface as shown in Figure 8. $AZ_{arrival}$ is computed from declination of the v_∞ vector at interface plant (DAP) and the inclination of the arrival hyperbola,

$$AZ_{arrival} = \arcsin\left(\cos i_{interface}/\cos L\right) \tag{31}$$

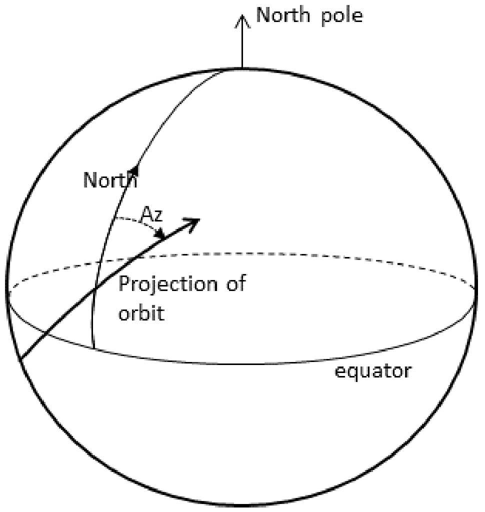


Figure 8: Definition of $AZ_{arrival}$

where $i_{interface}$ is the inclination of the arrival hyperbola,

$$\cos i_{interface} = \cos\theta_B \cos DAP \tag{32}$$

and θ_B is the B-plane clock angle,

$$\theta_B = \arccos\left(\frac{(\sin L - \cos(DAP + \pi/2)\cos\beta_{EI})}{\sin(DAP + \pi/2)\sin\beta_{EI}}\right) - \frac{\pi}{2} \tag{33}$$

and DAP is,

$$DAP = \arcsin(v_{\infty-z}/v_\infty) \tag{34}$$

Entry/Landing Illumination Constraint Some missions may require that the entry/landing event occur at an interface point that is illuminated by the sun. This constraint is evaluated by computing the solar phase angle β_{sun} between the interface position vector $\mathbf{r}_{EI}$ and the vector from the body center to the sun $\mathbf{r}_{body-\odot}$ in the BCI frame. The body-sun vector is first computed,

$$\mathbf{v}_{\infty-BCI} = -\mathbf{R}_{ICRF-to-BCI}\mathbf{r}_{\odot-body} \tag{35}$$

where $\mathbf{r}_{\odot-body}$ is the position of the body relative to the sun in the ICRF and $\mathbf{R}_{ICRF-to-BCI}$ is computed using Ref. [24]. The solar phase angle is then computed as

$$\beta_{sun} = \arccos\frac{\mathbf{r}_{EI}\cdot\mathbf{v}_{\infty-BCI}}{r_{EI}v_{\infty-BCI}} \tag{36}$$

If $\beta_{sun} \leq 90°$ then the interface point is illuminated. However the user may choose to set a more stringent constraint on β_{sun}.

Entry/Landing Longitude Constraint While a value for entry interface latitude is specified as an assumption in the model, the interface longitude is a function of the incoming asymptote and the epoch of encounter. The longitude may be constrained by first finding the longitude of the actual interface point and then comparing it to the desired longitude. First, the interface point $\mathbf{r_{EI}}$ must be transformed from BCI to body-centered fixed (BCF) coordinates,

$$\mathbf{r}_{EI-BCF} = \mathbf{R}_{BCI-to-BCF}\mathbf{r}_{EI-BCI} \tag{37}$$

where $\mathbf{R}_{BCI-to-BCF}$ is the rotation matrix from BCI to BCF and varies as a function of epoch. The longitude may then be computed,

$$LON = \arctan\frac{r_{EI-BCF-y}}{r_{EI-BCF-x}} \tag{38}$$

The entry longitude constraint may then be formulated as,

$$c_{longitude} = LON_{actual} - LON_{desired} \tag{39}$$

Entry/Landing Surface Velocity Constraint Once $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$ are computed in Equations 24 and 25, it is possible to compute the relative velocity between the spacecraft at interface and the rotating atmosphere. This is useful because a maximum atmosphere entry velocity is usually given as a requirement for sample return vehicles. First it is necessary to compute the velocity vector of the atmosphere at the given interface altitude and latitude,

$$\mathbf{v}_{atm} = (\hat{r}_{EI} \times \hat{z})\, v_{atm} \tag{40}$$

where v_{atm} is the magnitude of the atmosphere's velocity,

$$v_{atm} = d_{spin-axis}\dot{W} \tag{41}$$

$\dot{W}$ is the rotation rate of the planet [24] and $d_{spin-axis}$ is the distance between the interface point and the planet's spin axis at the given altitude and latitude,

$$d_{spin-axis} = r_{EI} \cos L \tag{42}$$

The relative velocity between the spacecraft and the atmosphere may then be computed,

$$\mathbf{v}_{entry-relative} = \mathbf{v}_{EI} - \mathbf{v}_{atm} \tag{43}$$

The relative velocity magnitude $v_{entry-relative}$ may then be constrained as necessary.

Entry/Landing Feasibility The entry interface constraints described above are very helpful in global optimization of sample-return trajectories but suffer from the limitation that $\mathbf{f}$ is defined only when the revolved hyperbola intersects the latitude circle, *i.e.*,

$$g \leq r_c \tag{44}$$

When this is not the case, $\mathbf{f}$ is undefined and therefore so are $\mathbf{r}_{EI}$ and $\mathbf{v}_{EI}$. A penalty function is therefore added to each of the entry constraints above that pushes the optimizer towards an entry asymptote whose revolved hyperbola intersects the latitude circle and disappears smoothly into the other other constraints when this condition is satisfied. If $g > r_c - 1$ then the penalty function is computed,

$$g_{penalty} = \frac{(g - r_c + 1)^2}{r_c^2} \tag{45}$$

and $\mathbf{g}$ is rescaled as,

$$\mathbf{g}_{scaled} = \frac{r_c - 1}{g}\mathbf{g} \tag{46}$$

$\mathbf{g}_{scaled}$ is then used in place of $\mathbf{g}$ for the computation of $\mathbf{f}$ and the rest of the interface model. $g_{penalty}$ is then added to each of the constraints in the NLP problem to push the solver back into the region where 44 holds. This method is found to be quite robust and makes it possible for the entry interface model described here to work effectively in a global search during that a wide variety of incoming velocity asymptotes might be considered. However, unlike for the other constraints used in this work, analytical derivatives of the entry constraints are not supplied to the optimizer. This is left for future work.

Spacecraft Hardware Modeling

Solar electric propulsion (SEP) systems produce varying amounts of thrust and consume propellant at varying rates depending on the available power and therefore on the position of the spacecraft in the solar system. This effect must be modeled in an effective low-thrust global optimization technique. In this work we will focus on models of real-world thrusters. Typically trajectory design engineers are supplied with polynomials defining the thrust and mass flow rate of a given thruster, *i.e.*,

$$T = a_T P_{eff}^4 + b_T P_{eff}^3 + c_T P_{eff}^2 + d_T P_{eff} + e_T \tag{47}$$

$$\dot{m}_{max} = a_F P_{eff}^4 + b_F P_{eff}^3 + c_F P_{eff}^2 + d_F P_{eff} + e_F \tag{48}$$

where P_{eff} is the power available to each thruster,

$$P_{eff} = P/N_{active} \tag{49}$$

and N_{active} is the number of thrusters firing at any point in time.

Equations 48 are valid over a range $[P_{min}, P_{max}]$ where P_{min} represents the minimum amount of power necessary to turn on the thrusterś PPU at the lowest setting and P_{max} represents the maximum amount of power that the PPU can safely accommodate.

The available power P is the difference between the power generated by the spacecraft $P_{generated}$ and the power required to operate the spacecraft bus $P_{s/c}$,

$$P = (1 - \delta_{power})\left(P_{generated} - P_{s/c}\right) \tag{50}$$

where δ_{power} is propulsion power margin.

In this work, the power delivered by a solar array is given by [16]:

$$P_{generated} = \frac{P_0}{r^2}\left(\frac{\gamma_0 + \gamma_1/r + \gamma_2/r^2}{1 + \gamma_3 r + \gamma_4 r^2}\right) \tag{51}$$

where the γ_i are user-defined solar panel coefficients, r is the distance between the Sun and the spacecraft in Astronomical Unit (AU) and P_0 is the "base power" delivered by the array at 1 AU. P_0 is in turn a function of the time since launch,

$$P_0 = P_{0-BOL}\left(1 - \tau\right)^t \tag{52}$$

where P_{0-BOL} is the base power delivered by the array at 1 AU on the day of launch, τ is the decay rate of the solar arrays measured as a percentage per year, and t is the time since launch in years. Equation (52) may also be used to model the decay of an radioisotope thermal generator (RTG) or advanced Stirling radiosotope generator (ASRG) power system.

The power required by the spacecraft bus $P_{s/c}$ is modeled as a polynomial,

$$P_{s/c} = a_{s/c} + b_{s/c}/r + c_{s/c}/r^2 \tag{53}$$

where $a_{s/c}$, $b_{s/c}$, and $c_{s/c}$ are chosen by the user.

The most interesting case is when $P_{eff} > P_{max}$ or $P_{eff} < P_{min}$, and therefore thrusters must be switched on or off. If $P_{eff} > P_{max}$ then either an additional thruster must be switched on, or if no other thrusters are available, P_{eff} must be clipped to P_{max}. If $P_{eff} < P_{min}$ then a thruster must be switched off. A discontinuity exists in Equation 48 at the boundaries where $P_{eff} = P_{max}$ or $P_{eff} = P_{min}$. This discontinuity is very confusing to gradient-based optimizers, especially in the case where there is not enough power to turn on any thrusters at all. It is desirable to smooth the power and propulsion models and remove the discontinuity. McConaghy [1] proposed smoothing the propulsion model using the "smoothstep" technique from the field of computer graphics. However a different approach is used in this work.

Heaviside [26] defined the unit step function as instantaneously taking half value at the point of transition. It is then possible to approximate the step function using the logistics function,

$$H(x) = \lim_{k\to\infty} \frac{1}{1 + \exp(-2kx)} \tag{54}$$

Equation 54 is continuously differentiable and therefore eliminates the problems that a gradient-based solver would have with a regular step function. In the context of multi-thruster switching, we define a set of Heaviside step functions $H_i(P)$,

$$H_i(P) = \frac{1}{1 + \exp(-2k(P - P_i^*))} \tag{55}$$

where each Heaviside step function $H_i(P)$ defines the switch state of the ith thruster and k defines the sharpness of the transition. The larger the value of k, the closer $H_i(P)$ approximates the Heaviside step

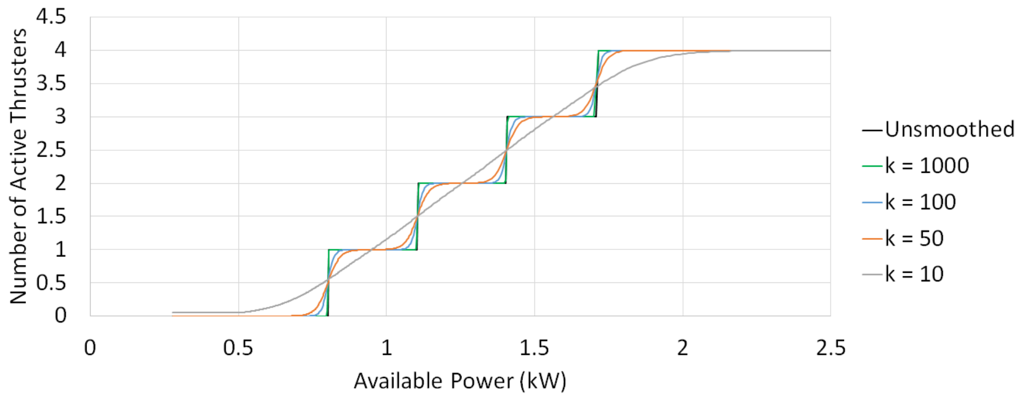


Figure 9: N_{active} vs Available Power for various values of k

function. However, while the derivatives $H_i'(P)$ increase as k increases, they remain finite and $H_i(P)$ remains continuous. We find that the optimizer behaves best when the derivatives are reasonably small (*i.e.* small k) but we also find that if k is too small then the thruster model will frequently report a fractional non-integer $H_i(P)$. N_{active} may then be defined as,

$$N_{active} = \sum_{i=1}^{N} H_i(P) \tag{56}$$

To fully define N_{active}, it is necessary to define the transition powers P_i at which a thruster would be switched on and off. In this work it is assumed that as few thrusters as possible are used for a given available power and so each P_i is an integer multiples of P_{max} except for P_1, which is equal to P_{min}. Alternatively if as many thrusters as possible are activated, then each P_i would then be an integer multiple of P_{min}. It is also possible to define other switching laws such as maximum thrust or maximum specific impulse (I_{sp}), in which case one would need to compute the P_i where those merit functions change as a function of number of thrusters.

Figure 9 shows the number of active thrusters for a notional system with four BPT-4000 thrusters, each with $P_{min} = 0.302kW$. The spacecraft's solar array can provide 10 kW at 1 AU and the spacecraft bus requires 0.5 kW at all times. Several values of k are shown, each a different compromise between smoothness and accuracy. Figure 10 shows the same propulsion system but this time with axes of distance from the Sun and available thrust, zoomed in to the region between 2 and 4 AU where the thruster transitions occur. One can see that the green line, representing $k = 1000$, closely approximates the unsmoothed black line that does not have continuous derivatives. However the smoothing method described here removes that discontinuity and significantly improves the robustness of the solver. A value of $k = 100$ is recommended as a good compromise between well-behaved derivatives and accuracy.

OPTIMIZATION

Nonlinear Programming

The optimization of the MGALT problem may be formulated as nonlinear programming (NLP) problems. NLP problems explicitly model nonlinear constraints. The optimizer solves a problem of the form:

$$\begin{aligned}&\text{Minimize } f(\mathbf{x})\\&\text{Subject to:}\\&\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}\\&\mathbf{c}(\mathbf{x}) \leq \mathbf{0}\\&A\mathbf{x} \leq \mathbf{0}\end{aligned} \tag{57}$$

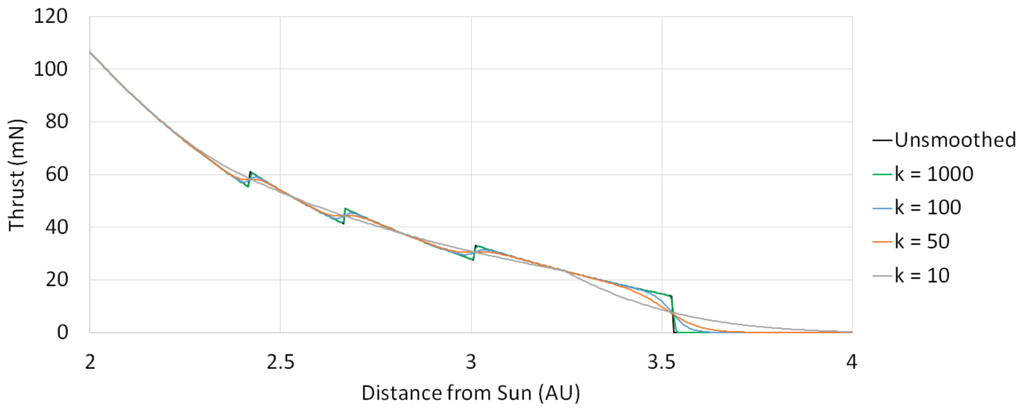


Figure 10: Thrust vs Distance from Sun for various values of k

where $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are the lower and upper bounds on the decision vector, $c(\mathbf{x})$ is a vector of nonlinear constraint functions, and A is a matrix describing any linear constraints (*i.e.* time constraints).

Almost all low-thrust interplanetary trajectory optimization problems, including those transcribed by the MGALT model, are very large, composed of hundreds or thousands of decision variables and tens or hundreds of constraints. A large-scale NLP solver such as SNOPT [27] is therefore required to solve the problems of interest in an efficient and robust manner. However, SNOPT, like all NLP solvers, requires an initial guess of the solution and will tend to converge to a solution in the neighborhood of that initial guess. The next section will address how the automated method of this work generates this initial guess in a fully automated manner.

Monotonic Basin Hopping

Recent research in low-thrust trajectory optimization has led to the creation of stochastic search methods that do not require an initial guess [9, 16, 17, 28–36]. The stochastic search method used in this work is monotonic basin hopping (MBH).

MBH [37] is an algorithm for finding globally optimal solutions to problems with many local optima. MBH works on the principle that many real-world problems have a structure where individual local optima, or “basins” tend to cluster together into “funnels” where one local optimum is better than the rest. A problem may have several such funnels. MBH was originally developed to solve molecular conformation problems in computational chemistry, but has been demonstrated to be effective on various types of interplanetary trajectory problems [17, 32–34, 38–40]. The pseudocode for MBH is listed in Algorithm 1.

MBH is run until either a specified number of iterations (trial points attempted) or a maximum CPU time is reached, at which point the best solution stored in the archive is returned as final solution. The version of MBH used in this work has two parameters - the stopping criterion and the type of random step used to generate the perturbed points $\mathbf{x}'$. In this work, the random step is drawn from a bi-directional Pareto distribution with the Pareto parameter, α, set to 1.4. The bi-directional Pareto distribution will usually generate small steps that allow MBH to *exploit* the local funnel around the current best solution. However some of the steps generated by the bi-directional Pareto distribution will be much larger, in some cases spanning the entire solution space. These larger steps allow MBH to *explore* the full problem. This approach has been shown to be robust on complex low-thrust problems [36].

The combination of NLP and MBH is well suited to searching for globally optimal solutions to the problems presented in this paper. The hop operator in MBH naturally handles searching for the globally optimal solution (*i.e.* escaping from local optima), and the NLP step is naturally suited to not only satisfy the trajectory continuity constraints inherent in MGALT but also the complex operational constraints discussed in this work.

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 synodic period
        **end if**
    **end for**
    run NLP solver to find locally optimal point $\mathbf{x}^*$ from $\mathbf{x}'$
    **if** $\mathbf{x}^*$ is feasible **and** $f\,(\mathbf{x}^*) < f\,(\mathbf{x}_{current})$ **then**
        $\mathbf{x}_{current} = \mathbf{x}^*$
        save $\mathbf{x}^*$ to archive
    **else if** $\mathbf{x}^*$ is infeasible **and** $\|c\,(\mathbf{x}^*)\| < \|c\,(\mathbf{x}_{current})\|)$
        $\mathbf{x}_{current} = \mathbf{x}^*$
    **end if**
**end while**
**return** best $\mathbf{x}^*$ in archive
```

RESULTS

Low-Thrust OSIRIS-REx

The first example presented here is a simulation of a low-thrust equivalent of the OSIRIS-REx mission [41], which we call "LowSIRIS-REx." The OSIRIS-REx mission, and also its fictional low-thrust cousin, will launch in 2016 and fly to 101955 Bennu. There it will perform one year of proximity operations culminating in acquisition of a sample, which it will then bring back to the Earth and drop into the Utah Test and Training Range (UTTR). The LowSIRIS-REx concept is therefore an ideal test of the atmosphere interface constraints developed in this work. In particular, the velocity relative to the Earth's atmosphere at interface, $v_{entry-relative}$, is constrained to be no greater than 12.4 km/s. The problem assumptions for the LowSIRIS-REx mission concept are shown in Table 1.

The LowSIRIS-REx problem was run twice, once with the atmospheric interface velocity constraint imposed and once without. In both cases the objective was to maximize the mass returned to Earth. The key results of the two runs are shown in Table 2. Note that the unconstrained mission returns slightly more mass than the constrained mission but at the cost of a much higher entry velocity of 13.06 km/s vs the 12.4 km/s required by the sample return capsule. The outbound journeys of the two missions are identical, but the return journeys are very different as shown in Figure 11. The constrained version of the mission departs Bennu two months earlier and arrives at Earth one week later. The trajectory itself is quite different. Notably there is only one thrust arc in the unconstrained return trajectory and then a long coast before arrival at the Earth, but in the constrained trajectory there are three thrust arcs - one long arc to depart Bennu and then two more short arcs to set up the entry velocity vector. In practical terms this costs only one week of flight time and 40 kg of propellant, but this example clearly shows that if one were to solve the unconstrained problem first and then prune away solutions that do not naturally satisfy the constraint, one might miss the constrained optimal solution that was not far from the unconstrained optimum. By imposing the operational constraint directly onto the global search process, we have removed a potential "false negative" in which the optimal solution might have been lost. However, in this particular example, the constrained solution is sufficiently close to the unconstrained solution that the unconstrained solution could be used as an initial guess for the constrained optimization.

Table 1: Assumptions for the LowSIRIS-REx Mission

Option	Value
Launch window open date	1/1/2016
Launch window close date	1/1/2017
Flight time upper bound	7 years
Arrival condition at Bennu	rendezvous (match position and velocity)
Arrival condition at Earth	intercept with $v_{entry-relative} \leq 12.4 km/s$
Launch vehicle	Atlas V 411
Launch asymptote declination bounds	$[-28.5, 28.5]$ (Kennedy Space Center)
Post-launch coast duration	60 days
Pre-arrival coast duration	90 days
Solar array $P0$ at end of life	15 kW
Solar array coefficients γ_i	$[1, 0, 0, 0, 0]$
Propulsion system	2 NEXT
Duty cycle	90%
Power margin	15%
Number of control steps per phase	40
MBH run time	3600 seconds

Table 2: Comparison of the Constrained and Unconstrained LowSIRIS-REx Solutions

	Unconstrained	Constrained
Launch date	11/4/2016	11/4/2016
Bennu arrival date	7/2/2019	7/2/2019
Bennu departure date	2/12/2021	11/10/2020
Earth arrival date	9/25/2023	10/2/2023
$v_{entry-relative}$ (km/s)	13.06	12.40
Mass at Earth return (kg)	3442	3402

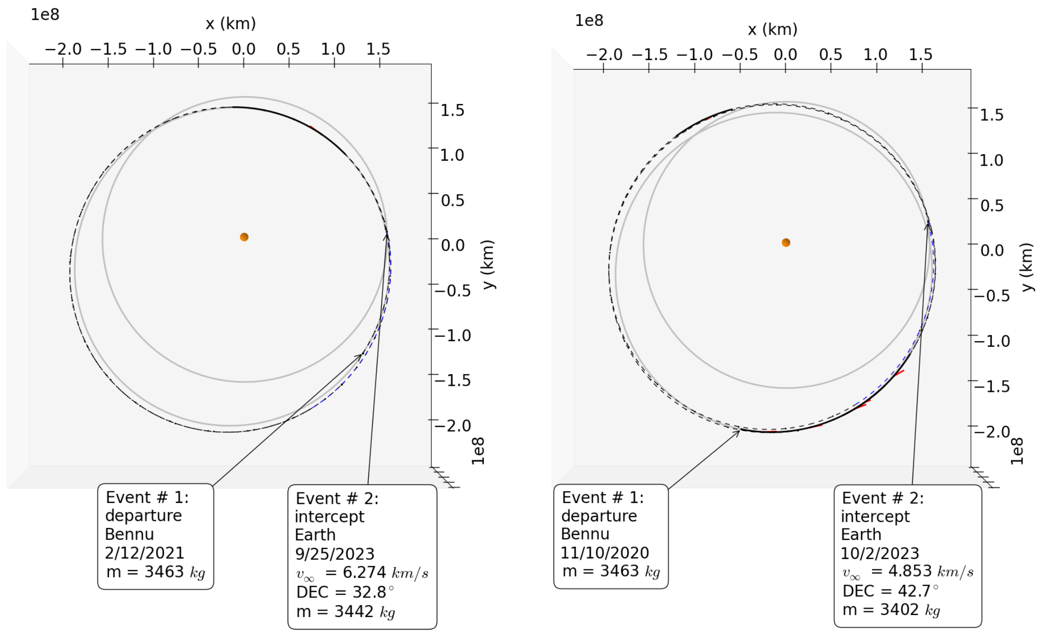


Figure 11: LowSIRIS-REx Earth Return Journey for the Unconstrained (L) and Constrained (R) Versions of the Mission

Table 3: Assumptions for the Comet Rendezvous Mission

Option	Value
Launch window open date	1/1/2020
Launch window close date	1/1/2026
Flight time upper bound	15 years
Arrival condition at 45P	rendezvous (match position and velocity)
Launch vehicle	Falcon 9 v1.1
Launch asymptote declination bounds	$[-28.5, 28.5]$ (Kennedy Space Center)
Post-launch coast duration	60 days
Solar array $P0$ at end of life	15 kW
Solar array coefficients γ_i	$[1, 0, 0, 0, 0]$
Propulsion system	2 NEXT
Duty cycle	90%
Power margin	15%
Number of control steps per phase	200
MBH run time	3600 seconds

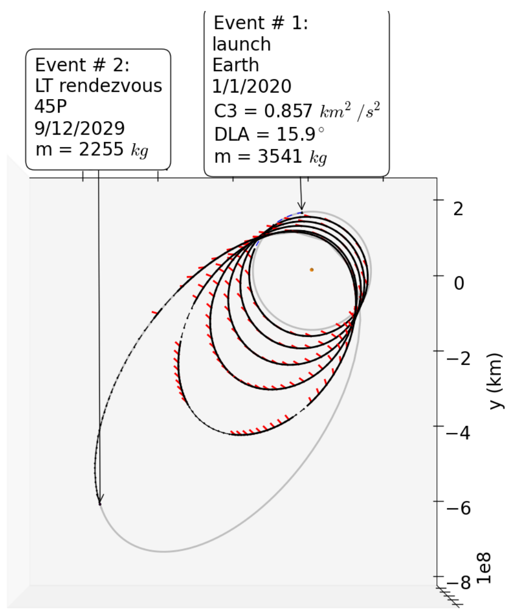
Table 4: Comparison of the Constrained and Unconstrained Comet Rendezvous Solutions

	Unconstrained	Constrained
Launch date	1/1/2020	9/14/2020
45P arrival date	9/12/2029	5/9/2023
Flight time (years)	9.7	2.6
Minimum solar approach distance (AU)	0.5	0.9
C3 (km^2/s^2)	0.9	13.8
Mass at launch (kg)	3541	684
Mass at arrival (kg)	2255	307

Comet Rendezvous

The second example is a rendezvous mission to Comet 45P/Honda-Mrkos-Pajdušáková using the same spacecraft as the LowSIRIS-REx mission but this time launching on a Falcon 9. Comet 45P was chosen because it reaches a perihelion distance of 0.52 AU and an aphelion distance of 5.51 AU, two extremes that would drive spacecraft design. The purpose of this example is to demonstrate how one might design a trajectory for a mission that would only be in the vicinity of 45P when it is sufficiently far from perihelion that the spacecraft would not have to be designed to survive both the extreme heat of 0.52 AU and the extreme cold of 5.51 AU. This may be done by constraining the spacecraft to fly no closer than 0.9 AU to the sun prior to rendezvous. In addition to testing the solar distance constraint, this example is also a good test of the new multi-thruster switching method. Table 3 lists the problem assumptions for the comet rendezvous mission.

The comet rendezvous problem was run twice, both with and without the solar distance constraint. In both cases the objective was to maximize mass delivered to the comet. The Heaviside multi-thruster switching technique was applied in both runs. Earth flybys were considered for both versions of the mission but good results were not found - the best solution found in both cases was a direct flight. The trajectories for the two solutions are shown in Figure 12 and summarized in Table 4. Figure 13 shows distance from the sun, power, and propulsion values as a function of time for both missions. The two solutions are completely different. The unconstrained mission launches to a lower C3 than the constrained mission, has a much longer flight time, and delivers far more mass than the constrained mission. These differences are all because the unconstrained mission is allowed to get closer to the sun, which in turn allows thrusting at a more efficient location on the orbit and also better thrust and I_{sp}. The unconstrained mission slowly pumps periapse down and apoapse up until it matches the orbit of 45P. The constrained mission, in contrast, needs to gain almost all of its orbital energy in one orbit because it cannot lower its periapse until it is already outbound to the rendezvous point.



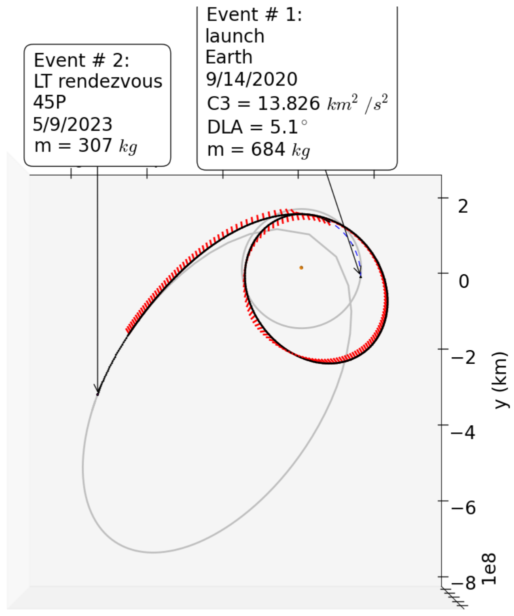


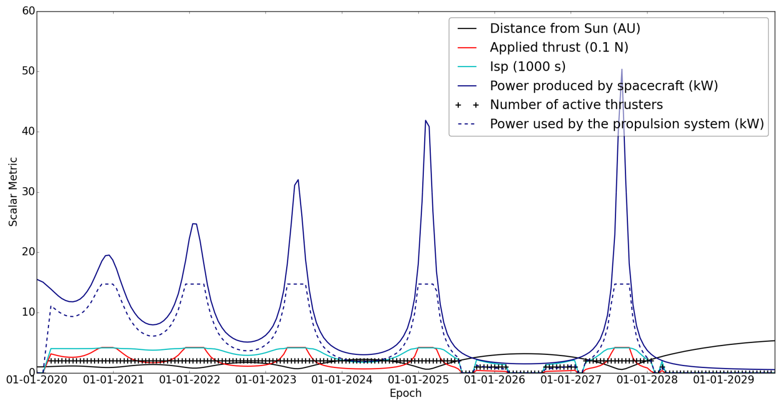
Figure 12: Comet Rendezvous Trajectory for the Unconstrained (L) and Constrained(R) Versions of the Mission

Clearly the unconstrained and constrained solutions are not in the same family. It is not possible to get to one with a local optimizer and an initial guess from the other. Furthermore, the comet rendezvous problem is an example of a "false positive" in which by solving the unconstrained problem one might be lead to believe that a good solution exists to the constrained problem. In actuality the solution to the constrained problem is completely unrelated. Further, while it is mathematically feasible it could not actually be flown because a spacecraft with a 15 kW solar array and two NEXT thrusters would be much larger than 300 kg. This is perhaps an extreme example of a "false positive" but is sufficient to illustrate the usefulness of imposing the operational constraints directly in the global search algorithm. Additionally, this example is a good use case for the Heaviside thruster-switching model because, as shown by the black plus signs in Figure 13, the number of thrusters switches from 0 to 1 to 2 and back again several times.

The only common element in both versions of the comet rendezvous mission is that in both cases the spacecraft appears to coast for a while immediately before rendezvous. This "terminal coast" lasts 18 months in the unconstrained mission and 5 months in the constrained mission. This coast appears because SNOPT cannot tell the difference between coasting next to the comet and arrival at the comet unless the analyst specifies a reason for it to do so - either an upper bound on the flight time that is strict enough to affect the solution or a time component in the objective function - neither of which are present in this example. Furthermore in both cases the coasting occurs when the spacecraft passes 3 AU, at which point there is no longer enough power to operate a single NEXT thruster. The Heaviside throttle switching model does allow SNOPT to compute derivatives of the match point constraints with respect to the arrival state and therefore improves solver performance, but it does not help SNOPT tell the difference between coasting and parking. Fortunately this phenomenon, while graphically unpleasant, does not affect the solution.

CONCLUSION

This work demonstrates the practicality and utility of including operational constraints in an automated global search process when designing low-thrust interplanetary trajectories. Such constraints may be applied to the NLP step of the MBH global search as described in this work. By including constraints such as atmosphere interface conditions and minimum solar distance, one can avoid both the "false negatives" and "false positives" that can occur when solving the unconstrained version of a problem first and then using it as an initial guess for a constrained optimization. While these constraints do add computational expense,



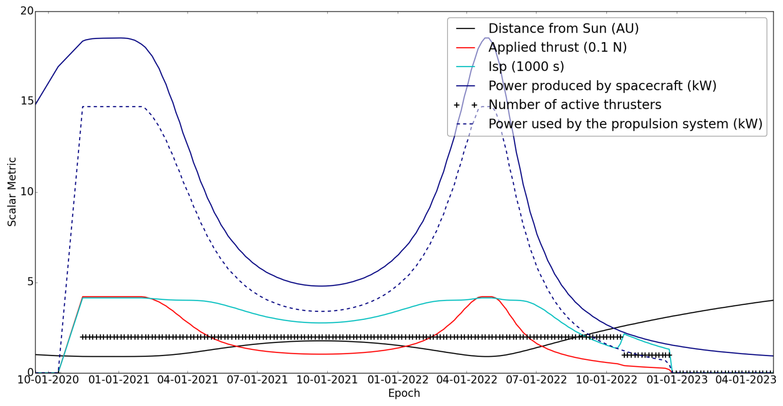


Figure 13: Distance from the Sun, Power, and Propulsion Values as a Function of Date for the Unconstrained (L) and Constrained (R) Comet Rendezvous Mission

they are compatible with an automated global search method and can help an analyst find better solutions. It would be possible to improve the performance of the solver when the atmosphere entry constraint is active if the derivatives of the entry constraints were specified analytically. The constraints presented here are just two of many that no doubt will be of interest to other researchers and mission planners.

In addition, the method of smooth thrusters switching using Heaviside's definition of the step function enables more robust convergence of the gradient-based local optimization component of the MBH global search algorithm. This allows for smooth derivatives of the trajectory design problem even in cases such as the comet rendezvous example where the number of thrusters must change over the course of the mission. In general the Heaviside step function allows a gradient-based optimizer to make discrete choices without causing a bifurcation in the solution space. There are no doubt many other applications for this technique.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Goddard Space Flight Center IRAD program and Vermont Space Grant, which paid for this work, Donald Ellison and Professor Bruce Conway at the University of Illinois for their work on analytical derivatives for the MGALT problem, and Jeremy Knittel at Goddard for his help in testing. We would also like to thank the planetary science community for giving us so many interesting problems to solve.

REFERENCES

[1] T. T. McConaghy, *Design and optimization of interplanetary spacecraft trajectories*. PhD dissertation, School of Aeronautics and Astronautics, Purdue University, 2004.

[2] A. Bryson and Y. Ho, *Applied Optimal Control*. Taylor and Francis, 1975.

[3] P. J. Enright and B. A. Conway, "Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981 – 985.

[4] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592 – 599.

[5] A. L. Herman and B. A. Conway, "Optimal, low-thrust, Earth-moon orbit transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 141 – 147.

[6] S. Tang and B. A. Conway, "Optimization of low-thrust interplanetary trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 599 – 604. Low thrust interplanetary trajectories;Collocation;Two body gravitational models;Planetocentric coordinates;Heliocentric coordinates;Earth to Mars transfer;Euler Lagrange equation;Spiral orbits;.

[7] P. Enright and B. Conway, "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 15, July-August 1992, pp. 994–1002.

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

[9] 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, Santa Fe, NM*, January 2014.

[10] A. Petropoulos and J. Longuski, "Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories," *Journal of Spacecraft and Rockets*, Vol. 41, September-October 2004, pp. 787–796.

[11] B. Wall and B. Conway, "Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 95–101.

[12] O. Abdelklalik and E. Taheri, "Approximate On-Off Low-Thrust Space Trajectories Using Fourier Series," *AIAA Journal of Spacecraft and Rockets*, Vol. 49, No. 5, 2012, pp. 962–965.

[13] D. Novak and M. Vasile, "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2010, pp. 128–147.

[14] B. Wall and D. Novak, "A 3D Shape-Based Approximation Method for Low-Thrust Trajectory Design," *AAS Astrodynamics Specialist Conference, Girdwood, AK*, July 2011.

[15] B. Woo, V. Coverstone, and M. Cupples, "Low-thrust trajectory optimization procedure for gravity-assist, outer-planet missions," *Journal of Spacecraft and Rockets*, Vol. 43, 2006, pp. 121–129.

[16] M. Vavrina and K. Howell, "Global Low Thrust Trajectory Optimization through Hybridization of a Genetic Algorithm and a Direct Method," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, August 18-21 2008.

[17] C. 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*, Vol. 225, 2011, pp. 1243–1251.

[18] J. Englander, M. Vavrina, D. Ellison, J. Knittel, D. Hinckley, R. Beeson, A. Ghosh, and B. Conway, “EMTG (Evolutionary Mission Trajectory Generator),” `https://sourceforge.net/projects/emtg/`.

[19] J. Sims, P. Finlayson, E. Rinderle, M. Vavrina, and T. Kowalkowski, “Implementation of a low-thrust trajectory optimization algorithm for preliminary design,” *AIAA/AAS Astrodynamics Specialist Conference*, August 2006.

[20] T. T. McConaghy, *GALLOP Version 4.5 User’s Guide*. School of Aeronautics and Astronautics, Purdue University, 2005.

[21] “PaGMO (Parallel Global Multiobjective Optimizer),” `http://pagmo.sourceforge.net/pagmo/index.html`.

[22] J. Prussing and B. Conway, *Orbital Mechanics*. Oxford University Press, New York, 1993.

[23] B. A. Conway, “An Improved Method due to Laguerre for the Solution of Kepler’s Equation,” *Celestial Mechanics*, Vol. 39, No. 2, 1986, pp. 199–211.

[24] B. A. Archinal, M. F. A’Hearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. K. Seidelmann, P. Stooke, D. J. Tholen, P. C. Thomas, and I. P. Williams, “Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2009,” *Celestial Mechanics and Dynamical Astronomy*, 2010.

[25] “General Mission Analysis Toolkit (GMAT),” `http://gmatcentral.org/display/GW/GMAT+Wiki+Home`.

[26] R. Bracewell, ed., *The Fourier Transform & Its Applications, 3rd ed.* McGraw-Hill Science/Engineering/Math, 2000.

[27] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Rev.*, Vol. 47, No. 1, 2005, pp. 99–131.

[28] G. Raowulf and V. Coverstone, “Near-Optimal Low-Thrust Orbit Transfers Generated by a Genetic Algorithm,” *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, 1996, pp. 859–862.

[29] V. Coverstone-Carroll, “Near-Optimal Low-Thrust Trajectories via Micro-Genetic Algorithms,” *Journal of Guidance Control and Dynamics*, Vol. 20, 1997, pp. 196–198.

[30] V. Coverstone-Carroll, J. Hartmann, and W. Mason, “Optimal multi-objective low-thrust spacecraft trajectories,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 24, 2000, pp. 387 – 402.

[31] J. Englander, B. Conway, and T. Williams, “Automated Mission Planning via Evolutionary Algorithms,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887.

[32] J. A. Englander, B. A. Conway, and T. Williams, “Automated Interplanetary Mission Planning,” *AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN*, August 2012.

[33] J. A. Englander, *Automated Trajectory Planning for Multiple-Flyby Interplanetary Missions*. PhD thesis, University of Illinois at Urbana-Champaign, April 2013.

[34] D. H. Ellison, J. A. Englander, and B. A. Conway, “Robust Global Optimzation of Low-Thrust, Multiple-Flyby Trajectories,” *AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC*, August 2013.

[35] D. H. Ellison, J. A. Englander, M. T. Ozimek, and B. A. Conway, “Analytical Partial Derivative Calculation of the Sims-Flanagan Transcription Match Point Constraints,” *AAS/AIAA Space-Flight Mechanics Meeting, Santa Fe, NM*, January 2014.

[36] 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.

[37] R. Leary, “Global optimization on funneling landscapes,” *Journal of Global Optimization*, Vol. 18, December 2000, pp. 367–383.

[38] M. Vasile, E. Minisci, and M. Locatelli, “Analysis of Some Global Optimization Algorithms for Space Trajectory Design,” *Journal of Spacecraft and Rockets*, Vol. 47, March-April 2010, pp. 334–344.

[39] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, “A global optimization method for the design of space trajectories,” *Computational Optimization and Applications*, Vol. 48, April 2011, pp. 635–652.

[40] J. Englander, M. Vavrina, R. Merill, and M. Qu, “Mars, Phobos, and Deimos Sample Return Enabled by ARRM Alternative Trade Study Spacecraft,” *AIAA/AAS Astrodynamics Specialist Conference*, August 2014.

[41] “OSIRIS-REx,” 2015. `http://www.asteroidmission.org/`.