

OPTIMIZATION OF THE LUNAR ICECUBE TRAJECTORY USING STOCHASTIC GLOBAL SEARCH AND MULTI-POINT SHOOTING

Jacob A. Englander*

David C. Folta†

Sun H. Hur-Diaz‡

Lunar IceCube is a 6U cubesat that will launch on NASA’s Artemis 1 mission in 2021. Lunar IceCube will separate from Artemis 1 shortly after trans-lunar injection (TLI) and travel to its science orbit about the moon using its Busek Ion Thruster 3 (BIT-3) propulsion system. This paper describes a technique to rapidly design Lunar IceCube trajectories using the monotonic basin hopping (MBH) stochastic global search algorithm, along with low- and high-fidelity multi-point shooting transcriptions. This technique allows the Lunar IceCube team to rapidly adapt to changing initial conditions, spacecraft properties, and operational constraints.

INTRODUCTION

The Lunar IceCube mission will map the distribution of volatiles on Earth’s moon using an infrared spectrometer. Lunar IceCube will launch with Artemis 1 in 2021 and then travel to the moon using solar-electric propulsion (SEP) [1]. The Lunar IceCube team is led by Morehead State University, with NASA’s Goddard Space Flight Center providing flight dynamics, propulsion systems engineering, and the Broadband InfraRed Compact High-Resolution Exploration Spectrometer (BIRCHES). Busek will provide the Busek Ion Thruster 3 (BIT-3) propulsion system.

Lunar IceCube is a low-cost technology demonstration mission that shares a launch with the Artemis 1 payload as well as twelve other cubesats. As a result, the launch conditions are mostly out of the control of the Lunar IceCube flight dynamics team. Different launch options for Artemis 1 can produce radically different Lunar IceCube trajectories. In addition, operational constraints on the spacecraft are continually evolving. It is therefore very useful to have an automated framework to systematically re-design Lunar IceCube’s trajectory whenever the spacecraft and/or operational constraints change. Such a framework will also be useful in operations whenever it is necessary to redesign due to the consequences of orbit determination (OD) error, maneuver execution error, or fault recovery.

Many researchers have studied the problem of low-energy transfers to the moon [2], and transfers using low-thrust electric propulsion in particular [3–7]. Folta et al. published the first iteration of the Lunar IceCube mission design [8]. Bosanac et al. approached Lunar IceCube using the circular restricted three body problem and differential correction [9]. Pritchett et al. also worked on the Lunar IceCube trajectory using the bi-circular restricted four body problem and a direct collocation solver [10].

*Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center

†Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center

‡Aerospace Engineer, Navigation and Mission Design Branch, NASA Goddard Space Flight Center

This paper discusses a very different approach using stochastic global search methods and multi-point shooting to rapidly and automatically design low-thrust cislunar trajectories. The techniques described here rely heavily on Goddard’s Evolutionary Mission Trajectory Generator (EMTG) optimization tool, which was originally developed for interplanetary trajectory design [11–13]. EMTG uses direct two-point shooting and parallel shooting methods, along with a nonlinear program (NLP) problem solver and the monotonic basin hopping (MBH) stochastic global search algorithm [14, 15] to search for solutions to complex problems without the need for an initial guess. The Python EMTG Automated Trade Study Application (PEATSA) [16] wrapper extends EMTG to trade study analysis, launch period mapping, and other sensitivity analyses such as missed-thrust recovery. With small modifications, EMTG, PEATSA, and the procedures for using them have proven to be effective on the Lunar IceCube problem. This will have implications for the design of future cislunar missions.

The process described in this work has two stages. The first stage is a low-fidelity optimization using a variant of the Sims-Flanagan transcription [17]. The Sims-Flanagan transcription approximates a low-thrust trajectory as a sequence of Keplerian arcs and bounded impulses that approximate the amount of Δv that would be accomplished by thrusting continuously for the length of each arc. Sims-Flanagan is very fast and provides a good initial guess for higher-fidelity optimization when used to design interplanetary missions. However, Sims-Flanagan is not natively suitable for cislunar mission design because it uses a two-body approximation. In this work, a variant of Sims-Flanagan is employed in which natural perturbations, such as n-body gravity and solar radiation pressure (SRP), are integrated into the impulse at the center of each Sims-Flanagan arc. A similar technique was used by Strange et al. to design trajectories to Europa while modeling the gravity of both the Sun and Jupiter [18]. If sufficiently granular arcs are used, then this technique is a good enough approximation to a true n-body integrator that it produces trajectories that may be used as an initial guess for a higher-fidelity optimization. Precise analytical derivatives of the objective function and all of the constraints are provided with respect to all of the decision variables.

The high-fidelity step is done either with a two-point direct shooting technique [19] or with a direct parallel shooting technique based on the work of Enright and Conway [20]. As in the low-fidelity case, analytical derivatives are provided.

PHYSICS MODEL

Low-Fidelity Modeling with Sims-Flanagan and Averaged n-body Perturbations

In this work, both low-fidelity and high-fidelity techniques are used to model thrust arcs. The low-fidelity model, used for initial broad search, is a variant of the well-known Sims-Flanagan transcription [17] in which continuous thrust trajectories are discretized into segments and the thrust is approximated as an impulse at the center of each segment. The impulse is bounded to be no larger than the Δv that the spacecraft could accomplish by thrusting for the length of the segment. The trajectory is propagated forward from the left-hand boundary of the phase and backward from the right-hand boundary of the phase. A set of match points on position, velocity, and mass are imposed at the midpoint. Both the constraints and the control are expressed in Cartesian coordinates, in the International Celestial Reference Frame (ICRF). The only decision variables in this phase type, other than whatever variables are needed to define the boundary conditions, are a set of 3-vectors to describe the control at each time step and one variable to represent that phase time of flight. Each control vector $\mathbf{u}_i$ is constrained to have a magnitude of no more than 1.0.

In classic Sims-Flanagan, the trajectory is propagated between impulses using a Kepler solver. This allows for very fast computation speed at the expense of lower modeling fidelity. This approach is suitable for interplanetary problems where two-body physics is dominant. However the two-body approximation is not appropriate for cislunar trajectory design. We therefore augment the Sims-Flanagan transcription by averaging the n-body perturbations across each time segment and incorporating them into the impulse as shown in Equation 3. Here Δv_i is the impulse at the i th time

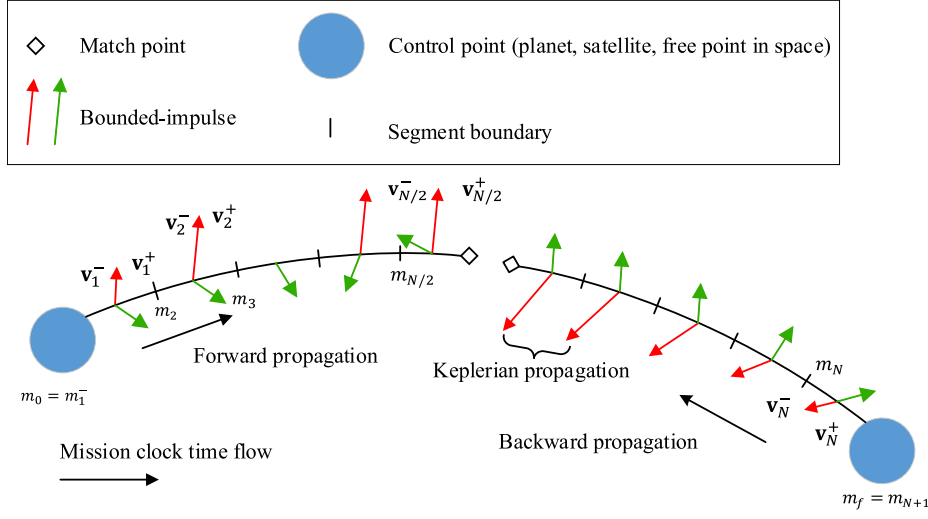


Figure 1: Diagram of a Sims-Flanagan phase with n-body gravity averaged into the impulse. The red vectors represent the Δv due to the thrust perturbation and the green vectors represent the Δv due to the natural perturbation.

segment, D is the thruster duty cycle, T_i is the available thrust, m_i is the mass of the spacecraft, μ_b is the gravitational parameter of the b^{th} body, and $\mathbf{r}_{body,s/c}$ is the vector from the b^{th} body to the spacecraft, where b denotes which body is being modeled. Analytical derivatives of the entire Sims-Flanagan perturbation are provided to the optimizer [12, 13].

$$\Delta v_i = \Delta v_{i,thrust} + \Delta v_{i,gravity} \quad (1)$$

$$\Delta v_{i,thrust} = \frac{\mathbf{u}DT_i\Delta t_i}{m_i} \quad (2)$$

$$\Delta v_{i,gravity} = \Delta t_i \sum_{b=1}^n \frac{-\mu_b \mathbf{r}_{body,s/c}}{r_{body,s/c}^3} \quad (3)$$

Figure 1 depicts a Sims-Flanagan phase with averaged n-body perturbations incorporated into the impulses. In this work, the gravity of the Earth, the Moon, and the Sun is included in the model. Depending on the phase of the mission, the central body may be either the Earth or the Moon and the other two are incorporated into the impulse.

High-Fidelity Modeling with Explicit Integration

After an initial solution is found using the low-fidelity model for thrust arcs, the trajectory is re-designed using a higher fidelity model. There are two different high fidelity models used in this work: a two-point shooting model and a parallel shooting model. In both cases the low-fidelity solution is used as an initial guess.

Two-Point Shooting The first - and simplest - high fidelity thrust arc model used in this work is a two-point shooting transcription that it is identical to the low-fidelity model except that instead of modeling the thrust and natural perturbations with bounded impulses and Kepler propagation, the trajectory is propagated with an 8th order Runge-Kutta integrator and an acceleration model that includes both the thrust and n-body gravity perturbations. Following the method of Pelligrini et al. [21], a fixed-step integrator is used instead of an adaptive step integrator so that we can provide the most accurate possible partial derivatives of the propagated state with respect to the initial

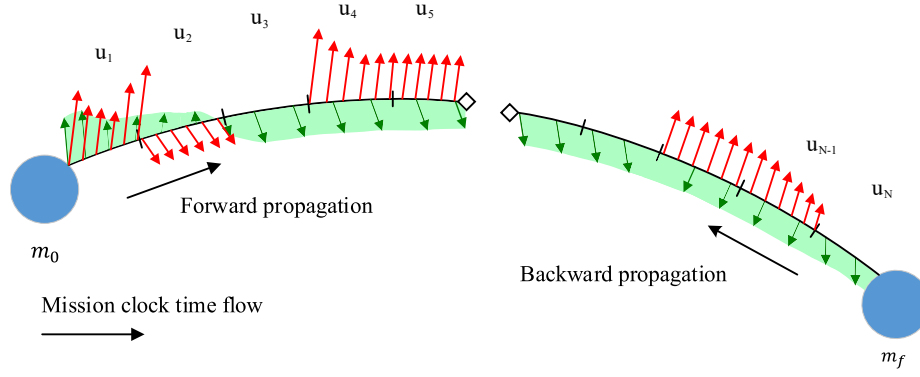


Figure 2: Diagram of a two-point shooting phase with n-body gravity and thrust perturbations. The red vectors represent the thrust vector and the green vectors represent the gravity perturbation.

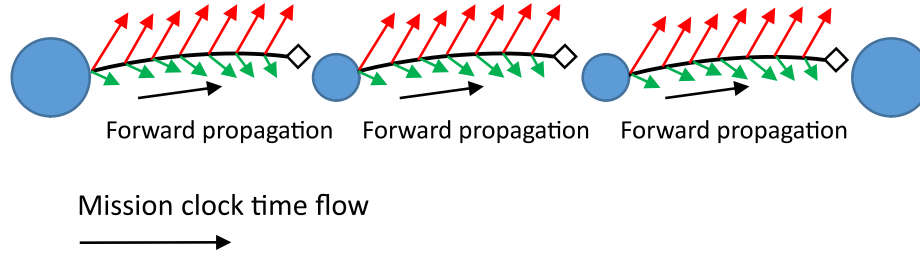


Figure 3: Diagram of a parallel shooting phase with n-body gravity and thrust perturbations. The red vectors represent the thrust vector and the green vectors represent the gravity perturbation.

state, control, and time variables. Because the decision variables and constraints are identical to the low-fidelity model, a solution for one may be easily used as an initial guess for the other. Figure 2 shows a thrust phase using two-point shooting.

Parallel Shooting While the high-fidelity two-point shooting transcription is robust and convenient because it shares a decision variable and constraints with the low-fidelity two-point shooting, it does have drawbacks. The most significant one is the difficulty of posing path constraints, as distance from the Earth or Moon, or constraints on where the thrust vector is allowed to point.

To mitigate this drawback, a direct parallel shooting method may be used instead of two-point shooting. In direct parallel shooting, proposed by Enright and Conway [22], a thrust phase is discretized into a set of sub-phases, each of which has its own set of encoded left-hand states (position, velocity, mass) and a set of constraints to ensure that the left-hand encoded boundary of the $(i+1)$ th sub-phase matches the propagated right-hand boundary of the i th sub-phase. Just as in high-fidelity two-point shooting, the trajectory is propagated using an 8th order Runge-Kutta integrator and n-body gravity.

Figure 3 depicts a phase using the high-fidelity parallel shooting transcription. Path constraints may be imposed at the boundaries between sub-phases. This advantage comes at the cost of a larger NLP problem, albeit one that is more sparse. The sub-phase boundary states and constraints can be in Cartesian coordinates, classical orbit elements (COE), modified equinoctial elements (MEE), or spherical coordinates.

Coast Phase

There are times in the mission where the spacecraft performs a pre-planned coast. This is distinct from when the optimizer determines that it is optimal to not thrust during a thrust opportunity.

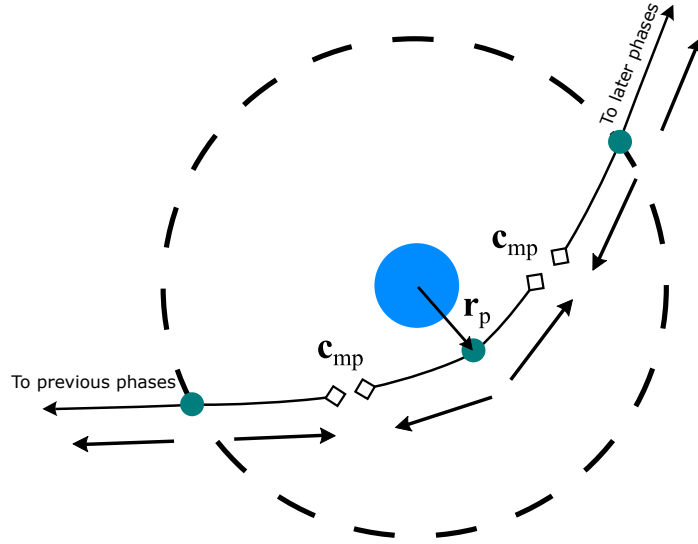


Figure 4: Diagram of a lunar gravity assist, modeled as two two-point shooting coast phases.

These explicit coasts are modeled as a two-point shooting problem with no thrust perturbation. The trajectory may be propagated with either a Kepler solver or the 8th order Runge-Kutta integrator, as appropriate.

Lunar Gravity Assist

A few days after launch, Lunar IceCube performs a lunar gravity assist. This is modeled as two coast phases. The first coast phase begins on the edge of a 50,000 km sphere surrounding the moon and ends at perilune. The perilune state is encoded using either spherical or b-plane coordinates. The second coast phase begins at perilune and ends at the edge of the same 50,000 km sphere. An altitude constraint may be imposed at the perilune point. Figure 4 shows how the gravity assist is constructed. The trajectory may be propagated either with a Kepler solver or the 8th order Runge-Kutta fixed-step integrator. If the integrator is used, then small time steps (60 seconds) are used within 10% of the phase flight time of perilune and larger time steps (600 seconds) are used for the rest of the gravity assist trajectory.

Propulsion Model

Lunar IceCube carries a Busek Ion Thruster 3 (BIT-3) propulsion system. BIT-3 is an electric propulsion system for cubesats that uses solid iodine propellant [23]. BIT-3 have several operational modes. Lunar IceCube operates in a higher power mode prior to the LGA and a lower power mode after the LGA. The details of the BIT-3 operational modes are intentionally omitted from this paper to protect the manufacturer's intellectual property.

Structure of the Optimization Problem

The constructs described above are tied together into a single optimization problem by means of boundary points. The boundaries into and out of the Moon's sphere of influence (SOI) are represented in spherical coordinates. The perilune point of the required LGA is also represented in spherical coordinates. The target orbit may be represented in any coordinate system; in this work we use COE. The boundaries between thrust and coast arcs that do not fit into these categories are encoded in Cartesian coordinates. Figure 5 is a diagram of the structure of the optimization problem.

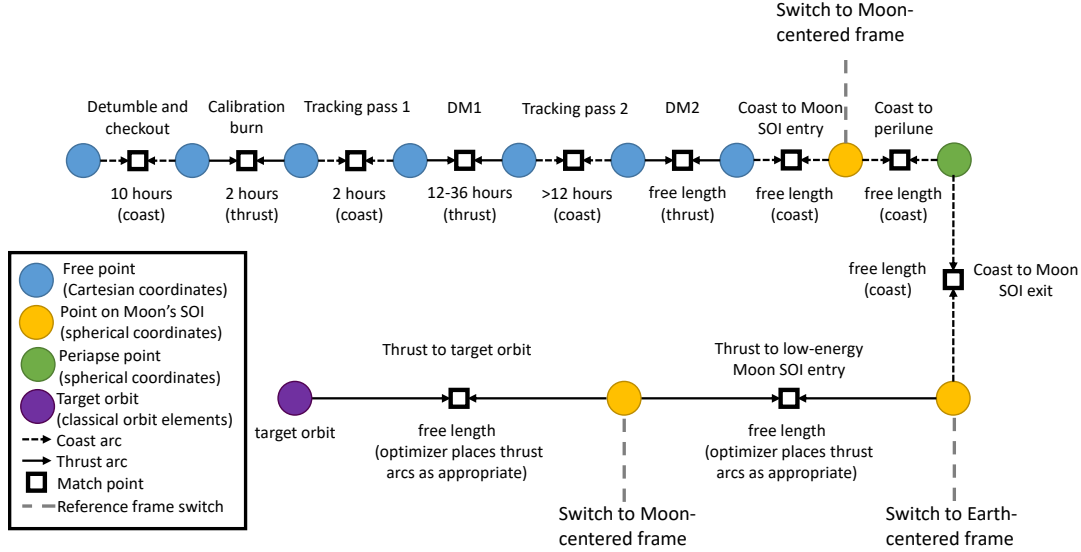


Figure 5: Diagram of the optimization problem for the initial point solution and trade studies, valid for both low-fidelity and high-fidelity models.

SOLVER

Nonlinear Programming

The optimization of the problems in this work may be formulated as NLP problems. The optimizer solves a problem of the form:

$$\begin{aligned}
 &\text{Minimize } f(\mathbf{x}) \\
 &\text{Subject to:} \\
 &\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub} \\
 &\mathbf{c}(\mathbf{x}) \leq \mathbf{0} \\
 &\mathbf{A}\mathbf{x} \leq \mathbf{0}
 \end{aligned} \tag{4}$$

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

The problems in this work consist of hundreds of variables and tens to hundreds of constraints. Such problems are best solved with a sparse NLP solver such as Sparse Nonlinear OPTimizer (SNOPT) [24]. SNOPT uses a sparse sequential quadratic programming (SQP) method and benefits greatly from precise knowledge of the problem Jacobian, i.e., the matrix of partial derivatives of the objective function and constraints with respect to the decision variables. EMTG provides analytical expressions for all of the necessary partial derivatives, leading to improved convergence vs. using numerically approximated derivatives [12, 13, 25]. SNOPT, like all NLP solvers, requires an initial guess of the solution and tends to converge to a solution in the neighborhood of that initial guess. The next section discusses EMTG’s fully automated method for generating initial guesses.

Monotonic Basin Hopping

Many researchers have studied stochastic search methods that do not require an initial guess [11–13, 26–30]. EMTG is designed to operate without user oversight, and so relies heavily upon stochastic search. The particular stochastic search method used in this work is monotonic basin hopping (MBH).

MBH [15] is an algorithm for searching for the best solutions to problems with many local optima.

Many problems, including those described in this work, are structured such that individual locally optimal “basins” cluster together, where the distance in the decision space from one local optima to the next in a given cluster may be traversed in a short “hop.” A problem may have several such clusters. 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 [11, 26, 31].

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. The version of MBH used in EMTG has two parameters: the stopping criterion and the type of random step used to generate the perturbed decision vector $\mathbf{x}'$. 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 usually generates small steps that allow MBH to exploit the local cluster around the current best solution. However, some of the steps generated by the bi-directional Pareto distribution are much larger, in some cases spanning the entire decision space. These larger steps allow MBH to explore the full decision space. This approach has been shown to be robust on complex interplanetary trajectory design problems [14].

MBH may be started either from a uniform-randomly chosen point in the decision space or from an initial guess derived from a previous problem. As long as the two problems are sufficiently similar, the latter approach is more efficient than starting from randomness. This property is exploited in the following section.

Python EMTG Automated Trade Study Application (PEATSA)

EMTG is most often run in a large-scale search mode, in which there are tens-to-thousands of individual cases that must be optimized. One example, as described later in this work, is the case of mapping a launch period to generate an optimal solution at one-day, or finer, intervals. Another example is the case of re-optimizing the mission-to-go for thousands of Monte Carlo samples. While it is possible to perform an independent global search on each problem posed to EMTG using NLP and MBH as described above, it is greatly advantageous to exploit similarities between the cases and share information between EMTG runs.

In this work, we use EMTG in tandem with the PEATSA wrapper script developed by Knittel et al. [16]. The user selects one or more variables by which solutions may be compared to each other, and the difference between the value(s) of these variable(s) define a “seed distance.” PEATSA harvests the results from a set of EMTG runs, and determines based on seed distance which EMTG solutions should be used to seed their neighbors. The EMTG cases are then re-run with each MBH process with the initial guesses locations chosen by PEATSA. After a short period of time (minutes), the EMTG runs are terminated and the seed-sharing process is repeated. This continues until either solutions to the population of cases cease to improve, the solutions all meet some user-defined threshold, or the analyst chooses to end the study.

CONCEPT OF OPERATIONS

This section describes the concept of operations (conops) that Lunar IceCube will fly. EMTG models and optimizes the entire conops sequence in a single optimization problem.

Deployment to LGA

Lunar IceCube deploys from Artemis 1 at “bus stop 1,” shortly after launch. Deployment states vary from day to day. However a consistent conops is used for all launch opportunities from deployment until the LGA.

The trajectory from launch to LGA is modeled as a set of explicit thrust and coast arcs. While the thrust arc transcriptions used in this work typically allow the optimizer to choose control of “up

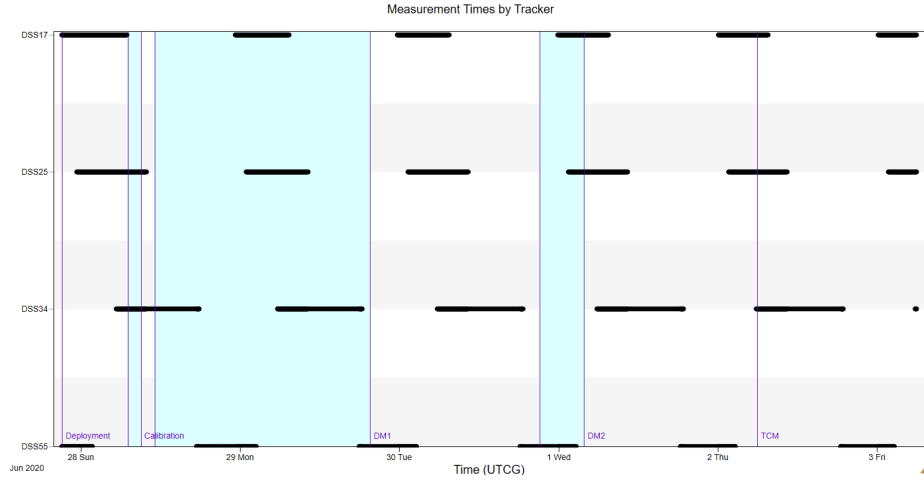


Figure 6: Time line of the first phase of the mission, from deployment to LGA entry. Light blue blocks indicate maneuvers and white blocks indicate coast arcs. The black horizontal lines indicate visibility to Deep Space Network (DSN) ground stations. This particular time line is associated with a June, 2020 launch opportunity.

to unit” magnitude and therefore allow partial thrust arcs, we constrain the control vector in the pre-LGA phase to exactly unit magnitude to force full thrust. The sequence does not change when switching from low-fidelity modeling to high-fidelity modeling. The central body for the first phase of the mission is the Earth, and the Sun and Moon are modeled as perturbing bodies.

Figure 6 depicts the conops from deployment to LGA entry. This phase of the mission begins at the deployment state and ends on the edge of a 50,000 km sphere surrounding the Moon. Lunar IceCube coasts for the first 10 hours of free flight, during which time detumble, spacecraft checkout, and orbit determination are performed. The next event is a two-hour calibration burn. This burn is used primarily to characterize the performance of the thruster, but we allow the optimizer to choose the thrust direction so that it can be useful in setting up the LGA. The calibration burn is followed by a 2 hour coast arc in which we conduct a tracking pass, which in turn informs an OD analysis and maneuver reconstruction to see how the thruster performed.

The next event is deterministic maneuver (DM) 1, which is constrained to last between 12 and 36 hours as determined by the optimizer. DM1 happens too soon after the calibration burn to allow for re-planning, but the results of the calibration can still be used to predict the spacecraft’s state after DM1 ends. DM1 is followed by another coast period of at least 12 hours, which is sufficient for an OD, re-optimization, and maneuver planning cycle. DM2 then begins, which lasts as long as the optimizer deems necessary. Finally, the spacecraft coasts until it reaches 50,000 km from the moon, ending the first phase of the mission.

Lunar Gravity Assist

The next mission phase is a lunar gravity assist. The central body is the Moon and no thrusting is allowed. This phase begins and ends on a 50,000 km sphere surrounding the Moon.

LGA to Low-Energy SOI Entry

After the LGA, conops varies depending on the deployment state and epoch, as well as the desired arrival epoch and final orbit at the moon. The trajectory from LGA exit to low-energy entrance ($v_\infty < 500m/s$) into a 50,000 km surrounding the moon. Because the sequence of thrust and coast arcs is not known a priori, this phase is initially modeled as a Sims-Flanagan phase with 100 thrust

segments. The optimizer may choose to fully thrust, partially thrust, or coast during each of these thrust segments. A 90% duty cycle is used to allow time for tracking. This duty cycle will eventually be explicitly modeled but in this work it is “averaged” by multiplying the commanded thrust and mass flow rate by 0.9.

After an initial solution is found using Sims-Flanagan, the entire trajectory is re-optimized using two point high-fidelity direct shooting. Partial thrust segments ($||\mathbf{u}|| < 1.0$) are still allowed after the LGA in this second optimization. Finally, the trajectory is re-phased with explicit thrust and coast arcs based on the solution to the second analysis, and then solved a third and final time. The resulting trajectory most closely models what the spacecraft will actually fly, and the thrust arcs may be used as an initial guess for a navigation tool.

If path constraints are needed then there is an optional fourth step in which the trajectory is re-optimized using the parallel shooting transcription. Regardless of which transcription is used, the central body for this phase is the Earth, and the Sun and Moon are modeled as perturbing bodies.

Moon SOI entry to Capture

The final phase modeled in this work is the descent for the Moon’s sphere of influence to a polar capture orbit with a semi-major axis of 38,250 km and an eccentricity of 0.875. Right ascension of the ascending node, argument of periapse, true anomaly, and flight time are free parameters. This very roughly matches the size and shape of a near-rectilinear halo orbit (NRHO). We chose this approach because, as the spacecraft’s orbit shrinks, there are many revolutions about the moon that require many thrust steps in order to let the inertial thrust direction vary, and so the size of the NLP problem becomes very large. We have also conducted experiments where Lunar IceCube rendezvous with the Lunar Gateway orbit [32]. Both versions of “intermediate orbit” provide a convenient and reasonably stable interface between the first half of the mission in cislunar space and the second half of the mission which takes place at the Moon.

This phase is modeled the same way at the LGA-to-SOI-entry phase except that the central body is the Moon. The same sequence of successively higher-fidelity transcriptions, from low-fidelity Sims-Flanagan to high-fidelity two-point or parallel shooting is used. Interestingly, Sims-Flanagan does not work very well when trying to rendezvous with the Lunar Gateway’s ephemeris file because the final time segment is too sensitive to be modeled as an impulse. However the higher fidelity transcriptions work fine.

Spiral to Science Orbit

After arrival in the intermediate elliptical orbit, Lunar IceCube will spiral down to a lower orbit where it will conduct science operations. The spiral and science phases of the mission are not considered in the optimization problem described here but rather are solved offline in Systems Tool Kit (STK) [33]. All Δv and propellant numbers quoted in this paper are up until the beginning of the spiral.

RESULTS

This section describes a particular case study that was done for a Lunar IceCube deployment on September 29th, 2021. The Artemis 1 launch date is currently in flux, and this particular deployment state is a representative “synthetic” deployment state created by the authors for the purpose of practicing the design process. Table 1 lists the initial conditions for the optimization runs described in this work. As new deployment states become available, the process described here will be repeated.

Parameter	Value
Epoch (JD TDB)	2459487.01900687
Epoch (GD TDB)	29 Sep 2021 00:27:22.194
x (km)	-30371.264669375927
y (km)	32098.932746719394
z (km)	26700.529858331356
vx (km/s)	-3.1422233925483196
vy (km/s)	1.3998239088421296
vz (km/s)	1.2475855654584151
mass (kg)	14.0

Table 1: Example deployment state in Earth-centered ICRF

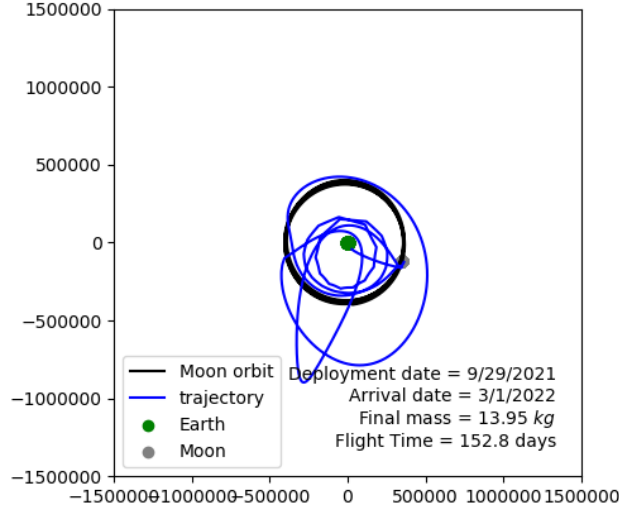


Figure 7: Example Lunar IceCube trajectory using Sims-Flanagan with averaged n-body gravity.

Initial Point Solution

The trajectory was first solved using MBH and no initial guess except for the specified initial state, using the low-fidelity models. We found that it was useful to solve the problem to LGA perilune first, and then add on the rest of the mission. In this optimization the arrival date was left free. The objective function for this and all other runs in this work is to maximize final mass.

The resulting trajectory arrives on March 1st, 2021, and requires a Δv of 67.4 m/s. The spacecraft expends 45.9 g of propellant. Figure 7 depicts the low-fidelity trajectory in the geocentric solar ecliptic (GSE) frame. Note that the Sims-Flanagan transcription, combined with averaged perturbations from the Sun and Moon, results in some rather strange looking trajectories when the spacecraft is between the Earth and the Moon. Nonetheless, this low-fidelity trajectory was a sufficiently good initial guess for a high-fidelity optimization.

The solution to the low-fidelity problem was then used as an initial guess for a second optimization run using the high-fidelity two-point shooting model. The resulting trajectory is shown in Figure 8. Note that the high-fidelity trajectory requires significantly more propellant. This trajectory performs 456 m/s of Δv and consumes 346 g of propellant. The difference in propellant is caused by properly

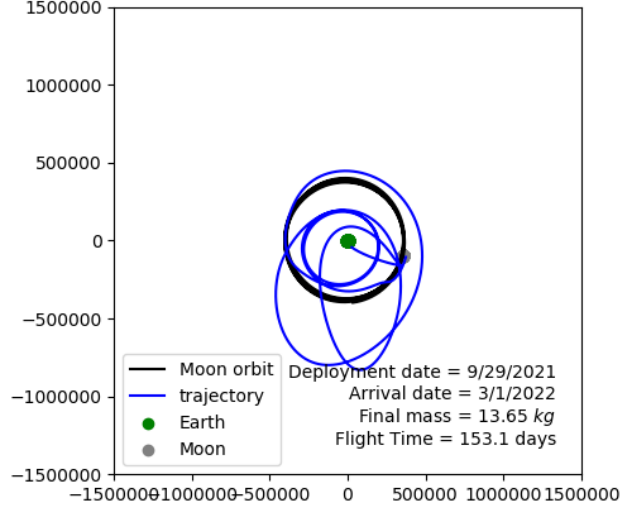


Figure 8: Example Lunar IceCube trajectory using high-fidelity two-point shooting, seeded from the low-fidelity solution.

integrating the force model instead of approximating it as a set of impulses. The low-fidelity model gives too large of a Δv due to gravitational perturbations for time segments whose center is either close to the moon or far from the Earth where the Sun is dominant, and in fact the optimizer tends to choose trajectories that do so. However, as shown in the next section, the most important thing is that the low-fidelity method was adequate to construct an initial guess at all, as even the high-fidelity point solution is only the seed for a broad search.

High-Fidelity Arrival Epoch Trade Study

So far, all of the results shown in this work are for a single deployment state and a single arrival date at the Moon. Furthermore, the time-free high-fidelity run converged to the same arrival date as the time-free low fidelity run. We next decided to perform a trade study across a year's worth of arrival dates. This was done by using the PEATSA process, in which the only trade parameter is the arrival date. Each PEATSA subproblem (PEATSA_{dough}) is constrained to arrive on a different date in 2022. After baking each set of cases in the PEATSAoven to produce a set of solutions (PEATSA_{crusts}), PEATSA shares information between neighboring PEATSA_{doughs} and re-bakes.

After 138 iterations, PEATSA returned the data set shown in Figure 9. This process was quite time consuming and ran for several days while the authors attended to other tasks. This plot continually improves, and some of the poorer days are likely just stuck in locally optimal solutions and would improve with further iterations. There are several blocks of arrival dates that deliver more than 13.9 kg of final mass. We will examine one of these cases in further detail in the next section.

High-Fidelity Point Solution

Figure 10 depicts a trajectory from the arrival date trade study that arrives on May 22nd, 2022. It is immediately apparent that despite being drawn from a PEATSA run seeded from the March 2nd, 2022 solution, this one is completely different. This trajectory requires a Δv of 112.86 m/s and consumes 86.1 g of propellant.

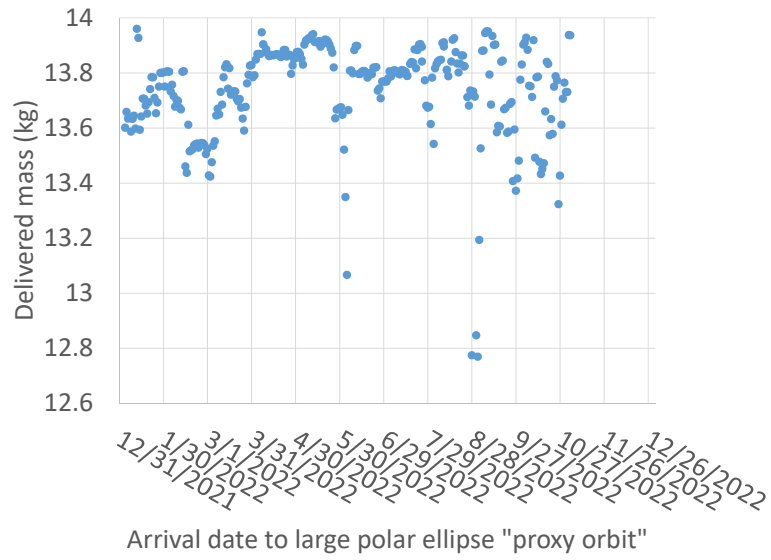


Figure 9: Trade study of final mass vs arrival date at the Moon, in high fidelity.

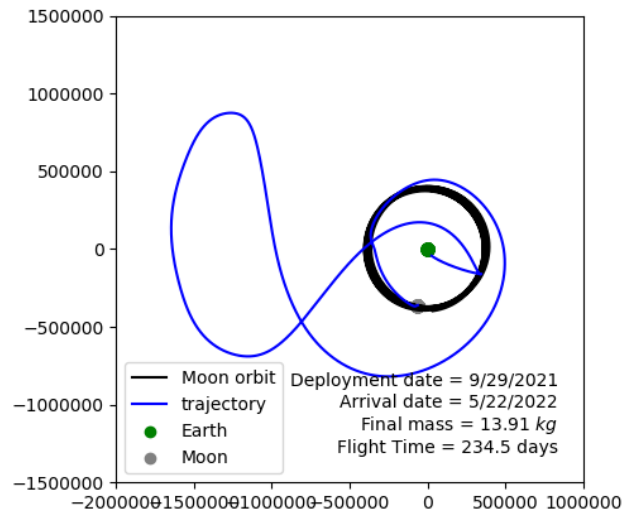


Figure 10: A selected trajectory from the arrival date trade study, in which the optimizer was allowed to choose full or partial thrust anywhere in the LGA-to-capture and capture phases.

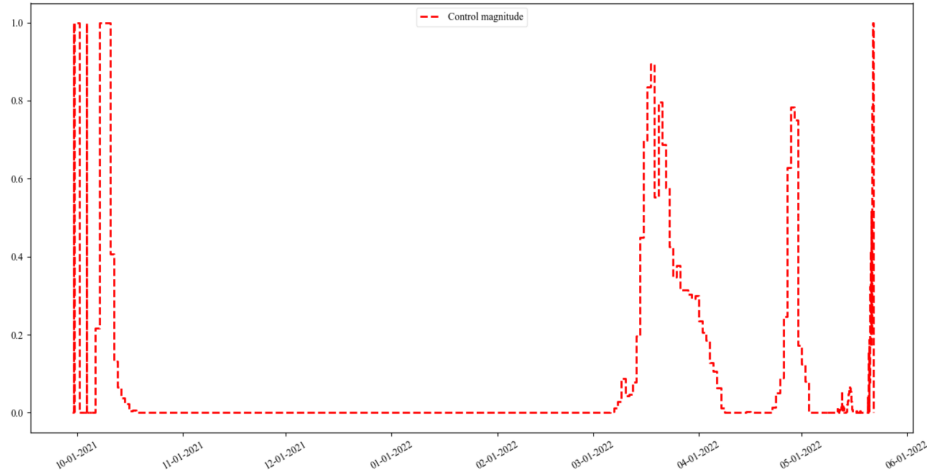


Figure 11: Control magnitude as a function of date, for the trajectory shown in Figure 10.

However, we are not yet done. As shown in Figure 11, the optimizer was allowed to choose full or partial thrust at any point during the LGA-to-capture and capture phases. In operations, the thruster may only be on or off. The final step is to re-phase the problem with discrete thrust and coast arcs, using a script that parses the EMTG output file and creates a new input file in which thrust arcs begin and end whenever the commanded thrust crosses 50%. The start epoch and length of each maneuver are allowed to vary, as is the control direction.

The “discrete thrust arc” problem is then re-optimized one last time in EMTG using only the NLP solver, without MBH. SNOPT converges in less than 60 seconds to the trajectory and control magnitude plots shown in Figures 12 and 13, respectively. The resulting trajectory requires 128.0 m/s and consumes 97.6 g of propellant. It is possible that further optimization with monotonic basin hopping might reduce this further. This trajectory may be used to generate targets for operational maneuver planning in STK, which serves as Lunar IceCube’s navigation tool.

CONCLUSION

Direct two-point and parallel shooting, combined with NLP and MBH have proven to be an effective combination of techniques for the optimizing the Lunar IceCube mission. This set of techniques is robust to changes in deployment conditions, spacecraft properties, conops, and other mission constraints. Trajectories created using the EMTG tool are then used as an initial guess for forward-targeted maneuver planning in AGI’s Systems Toolkit (STK) [33]. This approach may also be useful for other cislunar design problems, such as the Deep Space Gateway and/or the Human Landing System programs.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Lunar IceCube project for funding this work. Jacob Englander would also like to thank Donald Ellison at NASA Goddard Space Flight Center and Professor Bruce Conway at the University of Illinois for their help over many years in developing the trajectory transcriptions and solvers used in this work. Donald Ellison also created several of the figures that are used to describe the transcriptions. Jeremy Knittel at KinetX Aerospace created PEATSA, and Arnold Englander at the University of New Hampshire created the version of MBH that is used in this work.

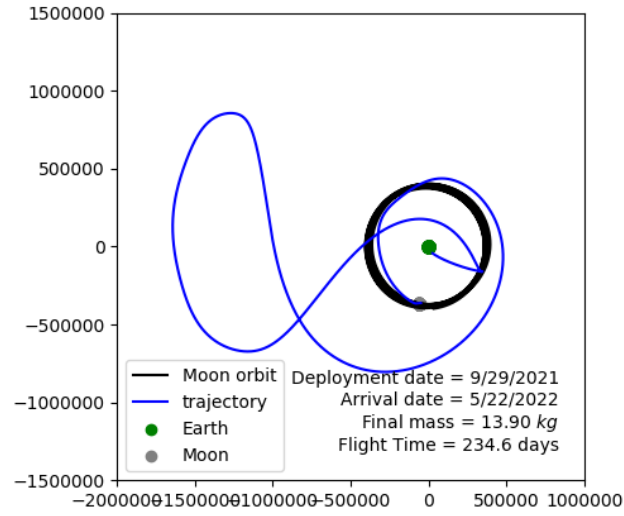


Figure 12: The May 22nd, 2022 arrival date trajectory with discrete thrust and coast arcs.

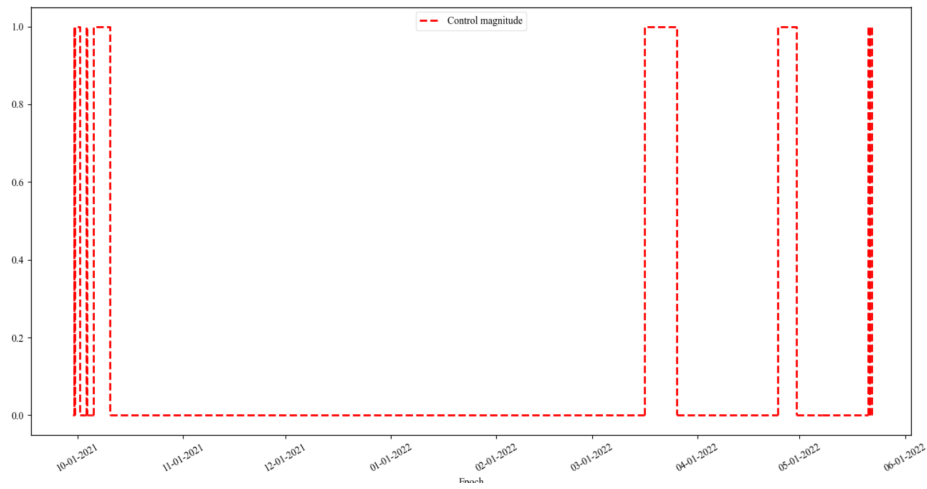


Figure 13: Control magnitude as a function of date, for the May 22nd, 2022 arrival date trajectory with discrete thrust and coast arcs.

REFERENCES

- [1] B. K. Malphrus, K. Z. Brown, J. Garcia, C. Conner, J. Kruth, M. S. Combs, N. Fite, S. McNeil, P. Clark, K. Angkasa, D. Folta, C. Brambora, P. Mason, S. Hur-Diaz, J. Nakamura, A. Martinez, M. Tsaye, M. Grubb, and C. Cutright, “Lunar IceCube: Pioneering Technologies for Interplanetary Small Satellite Exploration,” 70th International Astronautical Congress, October 2019.
- [2] J. S. Parker and R. L. Anderson, *Low-Energy Lunar Trajectory Design*, DESCANSO Deep Space Communication and Navigation Series. Hoboken, New Jersey: John Wiley and Sons, June 2014.
- [3] C. Martin and B. Conway, “Optimal Low-Thrust Trajectories Using Stable Manifolds,” *Spacecraft Trajectory Optimization* (B. Conway, ed.), Cambridge University Press, 2010.
- [4] M. Ozimek and K. Howell, “Low-Thrust Transfers in the Earth–Moon System, Including Applications to Libration Point Orbits,” *AIAA Journal of Guidance Control and Dynamics*, Vol. 33, 03 2010, pp. 533–549, 10.2514/1.43179.
- [5] J. Shannon, M. Ozimek, J. Atchison, and C. Hartzell, “Rapid Design and Exploration of High-Fidelity Low-Thrust Transfers to the Moon,” 03 2020.
- [6] C. A. Kluever and B. L. Pierson, “Optimal low-thrust three-dimensional Earth-moon trajectories,” *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, 1995, pp. 830–837, 10.2514/3.21466.
- [7] J. T. Betts and S. O. Erb, “Optimal Low Thrust Trajectories to the Moon,” *SIAM Journal on Applied Dynamical Systems*, Vol. 2, No. 2, 2003, pp. 144–170, 10.1137/S1111111102409080.
- [8] D. Folta, N. Bosanac, A. Cox, and K. Howell, “The Lunar IceCube Mission Challenge: Attaining Science Orbit Parameters from a Constrained Approach Trajectory,” 02 2017.
- [9] N. Bosanac, A. D. Cox, K. C. Howell, and D. C. Folta, “Trajectory design for a cislunar CubeSat leveraging dynamical systems techniques: The Lunar IceCube mission,” *Acta Astronautica*, Vol. 144, Mar. 2018, pp. 283–296, 10.1016/j.actaastro.2017.12.025.
- [10] R. Pritchett, K. C. Howell, and D. C. Folta, “Low-Thrust Trajectory Design for a Cislunar CubeSat Leveraging Structures from the Bicircular Restricted Four-Body Problem,” 70th International Astronautical Congress, October 2019.
- [11] J. Englander and B. Conway, “An Automated Solution of the Low-Thrust Interplanetary Trajectory Problem,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1, 2017, pp. 15–27.
- [12] D. H. Ellison, B. A. Conway, J. A. Englander, and M. T. Ozimek, “Analytic Gradient Computation for Bounded-Impulse Trajectory Models Using Two-Sided Shooting,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 7, 2018, pp. 1449–1462, 10.2514/1.G003077.
- [13] D. H. Ellison, B. A. Conway, J. A. Englander, and M. T. Ozimek, “Application and Analysis of Bounded-Impulse Trajectory Models with Analytic Gradients,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 8, 2018, pp. 1700–1714, 10.2514/1.G003078.
- [14] 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, ISSFD, May 2014.
- [15] R. H. Leary, “Global Optimization on Funneling Landscapes,” *Journal of Global Optimization*, Vol. 18, No. 4, 2000, pp. 367–383, 10.1023/A:1026500301312.
- [16] J. Knittel, K. Hughes, J. Englander, and B. Sarli, “Automated Sensitivity Analysis of Interplanetary Trajectories for Optimal Mission Design,” International Symposium on Space Flight Dynamics, Matsuyama, Japan, June 2017.
- [17] J. A. Sims and S. N. Flanagan, “Preliminary Design of Low-Thrust Interplanetary Missions,” AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, Paper AAS 99-338, August 1999.
- [18] N. Strange, D. Landau, R. Hofer, J. S. Snyder, T. Randolph, S. Campagnola, J. Szabo, and B. Pote, “Solar Electric Propulsion Gravity-Assist Tours For Jupiter Missions,” AAS/AIAA Astrodynamics Specialist Conference, August 2012.
- [19] 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, AAS, January 2014.
- [20] 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.

- [21] E. Pellegrini and R. Russell, “On the Computation and Accuracy of Trajectory State Transition Matrices,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, 09 2016, pp. 2485–2499, 10.2514/1.G001920.
- [22] 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.
- [23] “BIT-3 RF Ion Thruster,” June 2020. http://busek.com/index_htm_files/70010819F.pdf.
- [24] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM J. Optim.*, Vol. 12, jan 2002, pp. 979–1006, 10.1137/s1052623499350013.
- [25] D. H. Ellison, Robust Preliminary Design for Multiple Gravity Assist Spacecraft Trajectories. PhD thesis, May 2018.
- [26] C. H. Yam, D. D. Lorenzo, and D. Izzo, “Low-thrust trajectory design as a constrained global optimization problem,” Vol. 225, SAGE Publications, aug 2011, pp. 1243–1251, 10.1177/0954410011401686.
- [27] V. Coverstone-Carroll, J. Hartmann, and W. Mason, “Optimal multi-objective low-thrust spacecraft trajectories,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, jun 2000, pp. 387–402, 10.1016/s0045-7825(99)00393-x.
- [28] 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*, AIAA, aug 2008, 10.2514/6.2008-6614.
- [29] J. A. Englander, B. A. Conway, and T. Williams, “Automated Mission Planning via Evolutionary Algorithms,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, nov 2012, pp. 1878–1887, 10.2514/1.54101.
- [30] M. Vavrina, J. Englander, and D. Ellison, “Global Optimization of N-Manuever, High-Thrust Trajectories Using Direct Multiple Shooting,” *AAS/AIAA Space Flight Mechanics Meeting*, February 2016.
- [31] 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, jun 2009, pp. 635–652, 10.1007/s10589-009-9261-6.
- [32] D. Lee, “White Paper: Gateway Destination Orbit Model: A Continuous 15 Year NRHO Reference Trajectory,” tech. rep., NASA, 2019.
- [33] “Systems Tool Kit,” <http://agi.com/products/engineering-tools>.