FEASIBILITY OF LOW THRUST TRAJECTORY OPTIMIZATION APPLICATIONS TO DEBRIS REMOVAL MISSION DESIGN

Jason A. Reiter,¹ Andrew M. Goodyear,¹ Davide Conte,¹ Jason M. Everett ²

The density of debris in Low Earth Orbit makes operating a spacecraft more difficult with the addition of every new satellite. Kessler proposed a scenario in which the density becomes high such that collisions between objects cascade and cause further collisions. Inspired by the 9th Global Trajectory Optimization Competition, a mission is theorized that employs low-thrust propulsion to optimally rendezvous with and deorbit debris to prevent such a scenario from ever occurring. A beam search clustering method was used to select a series of individual missions that maximize the number of debris pieces removed while minimizing the fuel cost. However, it was found that such a mission is likely to be infeasible due to the J2 perturbation effects and limitations of low-thrust optimization technology.

INTRODUCTION

The Kessler effect is a scenario in which the density of objects in Low Earth Orbit (LEO) is high enough that collisions between objects cascade, generating debris that increase the likelihood of further collisions. To prevent this type of scenario, or at least delay its onset, actions must be taken to reduce the number of debris pieces in LEO. The 9th Global Trajectory Optimization Competition (GTOC 9), as designed by the European Space Agency’s Advanced Concepts Team, proposed a hypothetical scenario in which the removal of 123 debris pieces from sun-synchronous orbit would prevent the Kessler effect from permanently compromising LEO.¹ ² Building off of that concept further, a problem was approached with the idea of using low-thrust propulsion and expanding the debris search to the entire sun-synchronous orbit regime (downloaded from space-track.org) while accounting for perturbation effects due to J2 oblateness to demonstrate the validity of the proposal in a real-world scenario.

The solution was developed by writing and running an optimization procedure in both MATLAB and the C++ programming languages. The procedure consists of three steps: Phase I (Raiden), Phase II (The Butcher), and Phase III (The Boss). Phase I of the program determines the order in which the individual debris pieces are visited and takes the first steps to narrow the search space for each transfer by using what is referred to as a beam search clustering method, as explained in Section 2. Phase II of the program takes the transfer sequence output from Phase I

¹ Graduate Research Assistant, Astrodynamics Research Group of Penn State, The Pennsylvania State University, 229 Hammond Building, University Park, PA 16802
² Undergraduate Student, Department of Aerospace Engineering, The Pennsylvania State University, 229 Hammond Building, University Park, PA 16802
and implements a Particle Swarm Optimization (PSO) for impulsive transfer trajectories to estimate the departure and arrival epochs and states for transfers (using the only the two-body model) between debris objects to ensure that the transfer will converge to a valid, near-optimal solution (see Section 3 for further details on The Butcher). Phase III is then initiated, where the departure and arrival epochs and states from Phase II are used as inputs for a feedback control-based low-thrust trajectory optimization tool. The program structure is summarized in Figure 1.

**Figure 1. Program Flowchart.**

Each function used is shown in Fig. 1 as an individual block. The red colored function block indicates the lower-fidelity, impulsive solutions to the problem, the blue colored function block is the medium-fidelity, impulsive solution, and the green colored function block is the medium-fidelity, low-thrust solution. The search space is progressively narrowed as more fidelity is determined for the solution. A visual representation of this search space is captured in Fig. 2.

After the methods for these approaches are detailed, results are presented comparing the cost differences of the impulsive and low-thrust approaches. An example mission is then considered to demonstrate the validity of the low-thrust approach in de-orbiting actual sun-synchronous debris. The feasibility of such an approach is discussed in detail to highlight the potential difficulties in applying low-thrust trajectory optimization to debris-removal mission design.
Beam search is a heuristic algorithm that builds search trees with a restricted number of states at each level, referred to as beam width. By limiting beam width, beam search can minimize complexity and memory usage, which is beneficial for problems with a large search space. A basic understanding of this algorithm is shown in Figure 3.

A fast-paced beam search method capable of dividing the given timeline into a sequence of missions was used. This method, referred to as Raiden, was implemented to define the sequence by linking together debris based on the right ascension of ascending node (RAAN) and inclination of each orbit. Each mission was constructed such that no individual transfer exceeded a maximum lane change magnitude while attempting to distribute the required propellant mass evenly between missions. Raiden is structured to continue to link pieces of debris together until either a maximum number of objects per mission is achieved, or the estimated propellant usage exceeds the allowable estimated propellant usage per mission. The departure and arrival epochs in between each piece of debris were chosen such that the transfer is performed when the two debris pieces are near their closest approach and the transfer could be completed with plenty of time to spare given the required plane change.
The beam search algorithm was applied on both the debris-to-debris scope, and the mission-to-mission scope. The starting epoch of each mission was probed randomly throughout the allowable mission timeline for a set amount of iterations and, at each probed starting epoch, Raiden was used to find the locally next best missions and store them in a custom beam search map structure designed to branch from in future iterations. After the stopping criteria is met that terminates the debris-level beam search algorithm, the mission-level algorithm branches from all locally optimal missions to find the next best set of missions. Each branch of the mission-level beam search algorithm is partnered with a custom time cell structure that contains information about available time intervals for future missions based on the previous missions in that specific branch. The only stopping criteria of the mission-level beam search algorithm is the remaining number of debris available to be linked in a mission, based on the locations of the debris throughout the allowable time intervals.

1. Initialization
   (a) Let $\omega$ be the beam width.
   (b) Set $B = \{B_0\}$ and $B_\omega = \emptyset$, where $B$ is the set of nodes to be investigated, and $B_\omega$ the set of nodes branched out of the nodes in $B$.
   (c) If an initial feasible solution is available, set $z^*$ to its objective function value; otherwise, set $z^* = \infty$.

2. Iterative step
   (a) Choose the first node $\mu \in B$; Out of $\mu$, create as many branches as the problem allows with each branch obtained by appending to the partial solution associated with $\mu$ the variable corresponding to the next level of the tree, and insert the created nodes (i.e., the offsprings of $\mu$) into $B_\omega$.
   (b) If a node $\mu$ of $B_\omega$ is a leaf, then
      i. compute its objective function value $z_\mu$;
      ii. if $z_\mu < z^*$, update $z^*$ and the incumbent solution;
      iii. remove $\mu$ from $B_\omega$.
   (c) Assess the potential of each node $\mu'$ of $B_\omega$ using an evaluation operator (which yields an upper bound on the value of the objective function for any solution containing the partial solution associated with $\mu'$).
   (d) Rank the nodes of $B_\omega$ in a non-increasing order of their values.
   (e) Insert the min$\{\omega, |B_\omega|\}$ best nodes of $B_\omega$ into $B$; and set $B_\omega = \emptyset$.

3. Stopping condition
   If $B = \emptyset$, stop; otherwise, goto the iterative step.

Figure 3. Standard beam search algorithm.\textsuperscript{3}
TRAJECTORY OPTIMIZATION

In this section, the trajectory optimization methods used to compute the individual orbital transfers between debris pieces, referred to as The Butcher and The Boss, are presented. A Keplerian approximation (solution to Lambert’s problem), The Butcher, was used with a particle swarm algorithm to narrow down the search space and provide inputs for The Boss, a low-thrust optimization tool.

The Butcher (Impulsive Trajectory Optimization)

Given the computationally expensive process involved in optimizing trajectories where J2 perturbations are considered, it was necessary to find a way to decrease the search space by utilizing a lower-fidelity approximation to each orbital transfer. It was determined that the solution to the Keplerian Lambert’s problem was a “good enough” estimate to be passed on to the optimizer that would take into account J2 effects. In order to determine such a search space, a Particle Swarm Optimization (PSO) technique was applied. Given the desired debris ID numbers and the window available for the transfer as determined by the beam search algorithm, the available departure epoch in the transfer window and the transfer time (based on the remaining time available to complete the transfer) were used as particles in the optimization. The subsequent cost for each particle at each iteration was found by using the Keplerian Lambert’s solution to determine the ∆v cost of the transfer. This optimization method resulted in a single departure epoch and transfer time that would minimize the ∆v cost for the transfer under purely Keplerian dynamics.

Low-Thrust Optimizer (Low-Thrust Trajectory Optimization)

Though the results from The Butcher are considered accurate for an impulsive maneuver, a second tool is necessary to solve for a low-thrust maneuver between the debris pieces. A look-forward feedback control method was applied for optimizing the low-thrust maneuver using inputs from the impulsive optimizer.

Reiter et al. presents a feedback control law (restated in Table 1) in which the optimal angle to increase each classical orbital element (COE) is weighted based on the COE’s relative distance from its end state. The weighted angles are then combined so that the spacecraft is thrusting in the optimal direction for all desired COEs to reach their end states simultaneously. Using this control law as a basis, changes are made to maximize the effect of the controller. A threshold is applied to account for conflicting optimal thrusting directions. The optimal thrusting angle is updated at each time step and the thrusting direction are compared to the user-designated cut-off thresholds. If the thresholds are not met, thrusting does not occur.

To save computation time (compared to MATLAB’s built-in integrators) and avoid convergence tolerance issues, the COE differential equations presented in Reference 5 are combined with Euler’s method to propagate the orbital states. The differential equations for the semimajor axis, eccentricity, inclination, right ascension of the ascending node, and argument of perigee are
The thrust direction angles are solved for each COE separately using the equations in Table 1. These are the in-plane ($\alpha$) and out-of-plane ($\beta$) thrusting angles, both measured from the circumferential direction in the RCN (radial-circumferential-normal) reference frame, for the maximum instantaneous time-rate of change of each orbital element, regardless of their effect on the other orbital elements.

**Table 1. Optimal in-plane ($\alpha$) and out-of-plane ($\beta$) thrust angles for the maximum instantaneous change of each orbital element.**

<table>
<thead>
<tr>
<th>Classic Orbital Element</th>
<th>In-Plane Angle (rad)</th>
<th>Out-of-Plane Angle (rad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-major axis ($a$)</td>
<td>$\alpha = \tan^{-1}\left(\frac{e \sin v}{1 + e \cos v}\right)$</td>
<td>$\beta = 0$</td>
</tr>
<tr>
<td>Eccentricity ($e$)</td>
<td>$\alpha = \tan^{-1}\left(\frac{\sin v}{\cos v - \cos E}\right)$</td>
<td>$\beta = 0$</td>
</tr>
<tr>
<td>Inclination ($i$)</td>
<td>$\alpha = 0$</td>
<td>$\beta = sgn(\cos u)\frac{\pi}{2}$</td>
</tr>
<tr>
<td>RAAN ($\Omega$)</td>
<td>$\alpha = 0$</td>
<td>$\beta = sgn(\sin u)\frac{\pi}{2}$</td>
</tr>
<tr>
<td>Argument of Perigee ($\omega$)</td>
<td>$\alpha = \tan^{-1}\left(\frac{1 + e \cos v}{2 + e \cos v} \cot v\right)$</td>
<td>$\beta = \tan^{-1}\left(\frac{e \cot i \sin u}{\sin(\alpha - v)(1 + e \cos v - \cos \alpha \sin v)}\right)$</td>
</tr>
</tbody>
</table>

When more than one COE is changed in the orbital maneuver, the separate angles are combined together to make one optimal vector of angles $\alpha$ and $\beta$. To do this, the angles must first be transformed into the body-fixed RCN (radial-circumferential-normal) reference frame. This reference is such that the radial component is aligned with the radial unit vector positive in the zenith direction, the normal component is aligned with the osculating angular momentum vector positive in the direction of the cross product $\mathbf{R} \times \mathbf{\hat{v}}$, and the circumferential component is normal to the radius vector in the orbital plane and completes the right-handed triad of unit vectors. The transformation from $\alpha$ and $\beta$ to RCN can be calculated for each COE.
A weighting is then applied to this vector based on the relative value for the COE compared to its starting state and desired final state to ensure that all orbital elements converge simultaneously,

$$ w_{position} = \frac{z_f - z}{|z_f - z_0|} $$

where \( z_f \) is the desired final value, \( z \) is the instantaneous osculating value, and \( z_0 \) is the starting value for each COE being controlled. The weight is then multiplied by the optimal thrusting vector for the corresponding COE and the resulting vectors for all controlled COEs are added together to get a single optimal instantaneous thrusting angle,

$$ T = \sum_z w_{position} T_{COE} $$

where \( \Sigma_z \) signifies the summation of the term for all controlled COEs. This vector is then transformed back into angles \( \alpha \) (the inverse tangent of the radial magnitude of the angle \( T_R \)) divided by the circumferential magnitude \( T_c \) and \( \beta \) (the inverse tangent of the normal magnitude of the angle \( T_N \)) divided by the hypotenuse of the triangle created by connecting the normal and circumferential vectors \( \sqrt{T_R^2 + T_c^2} \), which are inserted into Eqs. (1-5) to propagate the transfer at each time step.

A cut-off threshold is then applied based on directionality. For certain orbital maneuvers, the combination of COEs being adjusted means that, at times, they will conflict in their optimal thrusting directions. When the conflict occurs, the resulting thrusting vector will not be ideal for reaching the desired final state because it harms the progress of one element at the expense of another. To avoid this, a threshold is applied to ensure that thrusting only occurs when it is ideal to do so. The directionality value is found from the instantaneous weighted thrusting angle for each COE by normalizing the total of all angle vectors and dividing that by the sum of each angle vector normalized.

$$ \frac{\|\Sigma_{\theta} T\|}{\Sigma_{\theta} \|T\|} < \text{directionality} $$

This results in a value between zero and one. Unlike the efficiency threshold, however, choosing a directionality threshold value (as it is an input to the tool) is not straight-forward. The optimal directionality threshold varies from case to case and, in some cases, there is such thing as a bad input. For example, if the threshold is set above 0.7071, calculated from Eq. (13) assuming that the two angle vectors are [0,1] and [1,0], thrusting for perpendicular vectors of equal length cannot occur. This means that, with a threshold above 0.7071, it would be impossible to increase both inclination and semimajor axis/eccentricity at the same time since there is a 90° angle between the two vectors. However, if only in-plane thrusting is required, this would not be an issue and values about 0.7071 could be chosen. The same method can be applied to show how the directionality threshold serves its purpose. If two COEs require thrusting in opposite
directions (assuming two angle vectors of \([0.8,0]\) and \([-0.5,0]\)), the directionality value becomes small (0.2308) and easily caught by a cut-off threshold.

After applying the control law and modifying the thrust locations using the cutoff threshold, the solutions found fit closely to results from Q-law, nonlinear programming, and indirect optimization techniques under similar conditions. However, for near-circular orbits, this optimization method fails when attempting to target specific eccentricity or argument of perigee values. Also, just like in Q-law, this optimizer is unable to target the fast-variable – in this case true anomaly.

RESULTS

Applying the approach detailed above to debris removal mission design proved extraordinarily difficult. To highlight this, the results of the impulsive solution (presented as the ARGoPS solution to the 9th Global Trajectory Optimization Competition) and low-thrust solution are compared using identical sets of simulated data. The low-thrust approach is then explored further using actual sun-synchronous debris data. The limitations of such an approach are then discussed.

Impulsive Solution (Simulated Debris)

The impulsive solution, as detailed in Reference 2, varies from the low-thrust solution in that \textit{The Boss} is replaced by a program named \textit{Sniper Wolf}. After \textit{The Butcher} has refined the search space even further, \textit{Sniper Wolf} takes over and implements another search for transfer trajectories using Particle Swarm Optimization (PSO) to find and calculate the transfers between debris objects using the full J2-affected dynamics to ensure that the transfer will converge to a valid and suboptimal solution. In order to find the exact necessary \(\Delta v\) and time-of-flight between two given debris for a range of departure and arrival dates (accounting for J2 effects), a multi-objective PSO was implemented. \textit{Sniper Wolf} is tasked with finding valid transfer solutions that satisfy the relative position and velocity constraints for rendezvous (100 meters and 1 m/s, respectively) while minimizing the total \(\Delta v\) needed to accomplish such a maneuver.

After running \textit{Raiden}, the top-level optimizer, multiple times using the beam search method described in Section 2, the most optimal solution that was found, \textit{The Fury}, consisted of 20 missions with a total of 103 transfers between debris pieces. The timeline of the mission is shown in Figure 4, where each segment corresponds to each individual mission (as defined by the number above it) launched as part of \textit{The Fury}. The first mission in chronological order, Mission 20, starts on MDJ 23476.38. A summary of all of the missions and transfers \(\Delta v\) and propellant mass is given in Table 2. The average required \(\Delta v\) and transfer time values per transfer for each mission are shown in Figure 5.
The intention in the mission planning was to attempt to gather as many debris pieces as possible while keeping fuel requirements within the mass restrictions defined by the problem. Figure 5 shows that this objective was achieved with success for larger missions, such as Missions 1 through 8. These larger missions maintained higher efficacy in using the fuel that was allocated for their duration. However, as large missions are created by sorting debris pieces into similar orbital planes, the likelihood of being able to group the rest of the debris similarly decreases. There is a trend that appears, where an increase in the ratio of fuel cost per debris captured manifests as the number of debris available decreases to a small number (such as a single transfer). Some missions consisted of lower quantities of debris removed with larger fuel masses, such as Missions 17 through 20. Although this is an undesirable effect, it is inherently a part of the problem.

Table 2. Summary of all of the missions of the sequence *The Fury.*

<table>
<thead>
<tr>
<th>Mission Number</th>
<th>Number of Transfers</th>
<th>( \Delta v ) [m/s]</th>
<th>Propellant Mass [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>3978.62</td>
<td>4951.58</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>3929.44</td>
<td>4881.75</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>3316.76</td>
<td>3599.01</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>3923.04</td>
<td>4764.82</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>3408.66</td>
<td>3776.52</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>3984.92</td>
<td>4841.94</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2258.99</td>
<td>2027.75</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>2765.71</td>
<td>2685.57</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>2313.49</td>
<td>2102.76</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>2785.08</td>
<td>2715.98</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>2508.40</td>
<td>2339.24</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>2506.96</td>
<td>2308.16</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>2787.16</td>
<td>2701.60</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>1467.72</td>
<td>1140.72</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1251.12</td>
<td>941.42</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>1384.10</td>
<td>1049.31</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>2373.11</td>
<td>2106.17</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1479.99</td>
<td>1134.23</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>2861.90</td>
<td>2759.22</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>2479.13</td>
<td>2239.80</td>
</tr>
</tbody>
</table>
Figure 4. The Fury – Mission Timeline.
In order to remove larger quantities of debris in less missions, the conditions that determine how similar orbital planes are classified for groups of debris needed to be relaxed. The relaxation of these conditions can result in costly transfers that are high in fuel consumption. The higher fuel consumption at times would fall outside of the bounds of the mass capability set forth by the problem statement. The alternative approach was to take a larger number of missions in order to be capable of transferring between all debris pieces without exceeding the mass restrictions that govern the problem. The required mass ended up being lower than the maximum carrying capability available at times, but this was useful in lowering the cost function for specific missions directly.

Generally, the missions consisting of longer average times of flight between objects also required less fuel, as seen in Figure 5. However, outliers like Mission 3 exist that do not fit that pattern. This likely occurred because not only are variation in semimajor axis and eccentricity not accounted for when determining the mission sequence, but the optimization methods used are not guaranteed to find global minima.

The impulsive approach results from ARGoPS’ solution to the 9th GTOC can be used as a basis for determining the validity of applying low-thrust to an identical problem.
Low-Thrust Solution (Simulated Debris)

To demonstrate the usefulness of applying low-thrust technology rather than relying on an impulsive engine, the first mission of the impulsive solution sequence was solved using the low-thrust optimization approach described above. Modeled after the Hall thruster designed for the Asteroid Redirect Mission (ARM), each thruster was assumed to have an ISP of 3000 seconds and a thrust of about 0.5 N.\(^6\) The wet mass of the spacecraft (based on the number of debris to be de-orbited) is used to determine the number of thrusters required for the spacecraft to be able to traverse LEO under the influence of J2 effects.

Applying the low-thrust trajectory optimization described above, it was found that the directionality threshold resulted in the thruster being turned off for a large portion of each orbit. Figure 8 shows an example of a single transfer, where the portion in red indicates the thruster is on and the portion in blue indicates the thruster is off.

![Low-Thrust Transfer Example](image)

Figure 8: Low-Thrust Transfer Example

Assuming that the same 11 transfers as that of the first mission in *The Fury* are completed, this mission requires 20 Hall thrusters and 2270 kg of fuel, compared to almost 5000 kg of fuel required using impulsive thrusters. However, it should be noted that, not only is a phasing maneuver still required to target the fast variable (true anomaly), but given the small force produced by low-thrust engines, it proved difficult to successfully target orbital elements affected by the oblateness perturbations. For this solution, the target orbit was set constant and a note was made that this is not a truly accurate representation. Though it would require additional fuel past
the 2270 kg calculated, the majority of the required plane-change maneuvers (by-far the driving factor in fuel requirements) has already been taken into account.

Table 3 details the time-of-flight, ΔV, and fuel mass required for each of the 11 transfers. It can be noted that the time-of-flight can vary wildly depending on the characteristics of the transfer being completed. Closer examination shows that, as expected, the time-of-flight and fuel mass vary linearly, but the time-of-flight and ΔV relationship becomes less linear as the time-of-flight increases due to the presence of engine shutoffs. Further optimization of the departure epoch and time-of-flight would likely lead to more continuity across the transfers and even lower fuel mass costs. Comparing the propellant mass required to that of the impulsive mission shows that the propellant mass savings are significant except for two transfers. This is likely due to the fact that The Butcher does not produce truly optimal departure and arrival epochs, especially for low-thrust transfers.

<table>
<thead>
<tr>
<th>Transfer Number</th>
<th>TOF (days)</th>
<th>ΔV (km/s)</th>
<th>Propellant Mass - Low Thrust (kg)</th>
<th>Propellant Mass - Impulsive (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38.56</td>
<td>7.27</td>
<td>1093.79</td>
<td>722.07</td>
</tr>
<tr>
<td>2</td>
<td>1.20</td>
<td>0.25</td>
<td>32.62</td>
<td>833.77</td>
</tr>
<tr>
<td>3</td>
<td>7.48</td>
<td>1.59</td>
<td>204.22</td>
<td>551.56</td>
</tr>
<tr>
<td>4</td>
<td>7.78</td>
<td>1.83</td>
<td>220.66</td>
<td>396.75</td>
</tr>
<tr>
<td>5</td>
<td>2.76</td>
<td>0.55</td>
<td>63.95</td>
<td>403.02</td>
</tr>
<tr>
<td>6</td>
<td>4.32</td>
<td>0.88</td>
<td>99.98</td>
<td>640.75</td>
</tr>
<tr>
<td>7</td>
<td>10.86</td>
<td>2.81</td>
<td>298.97</td>
<td>369.70</td>
</tr>
<tr>
<td>8</td>
<td>2.32</td>
<td>0.55</td>
<td>55.01</td>
<td>274.78</td>
</tr>
<tr>
<td>9</td>
<td>0.48</td>
<td>0.13</td>
<td>12.43</td>
<td>117.77</td>
</tr>
<tr>
<td>10</td>
<td>23.72</td>
<td>7.67</td>
<td>669.77</td>
<td>116.27</td>
</tr>
<tr>
<td>11</td>
<td>0.86</td>
<td>0.25</td>
<td>19.13</td>
<td>403.00</td>
</tr>
<tr>
<td>Total</td>
<td>100.34</td>
<td>23.78</td>
<td>2770.53</td>
<td>4829.44</td>
</tr>
</tbody>
</table>

Low-Thrust Solution (Actual Debris)

The same approach was used to determine the requirements for removing actual pieces of debris from sun-synchronous orbit. The two-line element (TLE) data was collected from space-track.org and all catalogued debris pieces in LEO were included in the beam search, Raiden. The beam search results suggested that the real debris pieces could be collected with nearly as much ease as the simulated pieces and that, given a 15-year search space, all 296 catalogued pieces of debris could be collected in just 29 missions. Running the beam search with similar constraints to that placed for the GTOC results, a 12-transfer mission was chosen for analysis.

When optimizing the transfers between real pieces of debris, it was found that an even higher thrust was required for convergence. A Thrust of 12.43 N was used, equivalent to 25 ARM Hall
thrusters. With this thrust, the 12 transfers were completed with 2961 kg of fuel over 82.51 days. It is useful to note that the first transfer required the most amount of fuel by far. This is likely due to the high initial mass of the spacecraft and the direction of the RAAN change. Efforts to decrease the dry mass of the spacecraft would go a long way in decreasing the total fuel required as the remaining 11 transfers cost less than another 200-kg combined. Again, further optimization of the departure epoch and time-of-flight would likely lead to more continuity across the transfers and even lower fuel mass costs as The Butcher does not produce truly optimal departure and arrival epochs, especially for low-thrust transfers.

<table>
<thead>
<tr>
<th>Transfer Number</th>
<th>TOF (days)</th>
<th>ΔV (km/s)</th>
<th>Propellant Mass - Low Thrust (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76.79</td>
<td>23.82</td>
<td>2774.07</td>
</tr>
<tr>
<td>2</td>
<td>1.24</td>
<td>0.52</td>
<td>38.77</td>
</tr>
<tr>
<td>3</td>
<td>0.19</td>
<td>0.067</td>
<td>4.99</td>
</tr>
<tr>
<td>4</td>
<td>1.89</td>
<td>0.89</td>
<td>65.14</td>
</tr>
<tr>
<td>5</td>
<td>0.29</td>
<td>0.12</td>
<td>8.76</td>
</tr>
<tr>
<td>6</td>
<td>0.31</td>
<td>0.16</td>
<td>11.13</td>
</tr>
<tr>
<td>7</td>
<td>0.23</td>
<td>0.08</td>
<td>5.79</td>
</tr>
<tr>
<td>8</td>
<td>0.40</td>
<td>0.19</td>
<td>13.43</td>
</tr>
<tr>
<td>9</td>
<td>0.029</td>
<td>0.012</td>
<td>0.87</td>
</tr>
<tr>
<td>10</td>
<td>0.83</td>
<td>0.39</td>
<td>27.65</td>
</tr>
<tr>
<td>11</td>
<td>0.16</td>
<td>0.078</td>
<td>5.46</td>
</tr>
<tr>
<td>12</td>
<td>0.15</td>
<td>0.082</td>
<td>5.65</td>
</tr>
<tr>
<td>Total</td>
<td>82.51</td>
<td>26.41</td>
<td>2961.71</td>
</tr>
</tbody>
</table>

Limitations of Applying Low-Thrust Optimization

As demonstrated by the sheer number of engines necessary to complete the analyzed transfers, applying low-thrust optimization to debris removal mission design has major limitations. The main inhibitor is the force due to J2 perturbations. J2 naturally causes the RAAN value of an object to drift in the increasing direction. For impulsive maneuvers, this helps to save fuel when that is taken into account, but thrusting against the changing RAAN value is still easily
accomplished. This is not the case with low-thrust engines. In order to move such a large spacecraft fast enough to overcome the effects of J2, and then some, on the osculating elements, a large thrust is required. For a numerical comparison, assuming a 5000-kg spacecraft, the acceleration due to J2 is on the order of $1 \times 10^{-5}$ km/s² while the 25 ARM engines produce an acceleration on the order of $1 \times 10^{-6}$ km/s². The advanced engines designed for ARM provide enough acceleration to allow for convergence, but it would require 8 times as many thrusters as the ARM spacecraft itself. This would likely prove to be infeasible based on the power requirements alone.

In addition, even state-of-the-art low-thrust optimization tools would have difficulties solving for the optimal transfer. The issues surrounding circular orbits could be solved by using equinoctial elements but, as of today, no low-thrust optimization tool can target all six orbital elements. In addition, the osculation of the target state means that the states in the optimizer are constantly evolving. Given the strength of the perturbations and the nature of trajectory optimization, this may prove insurmountable in sun-synchronous orbit.

Overall, low-thrust optimization and low-thrust technology both must see an increase in capability for applying low-thrust optimization for this type of mission to be feasible. When that time comes, low-thrust engines will have a distinct advantage over impulsive engines in fuel cost savings.

**CONCLUSION**

An optimization strategy using a beam search clustering method *(Raiden)* was used to group orbital debris with similar orbital planes and RAAN to determine the order and timing of each visit. An impulsive Lambert’s problem solution utilizing Particle Swarm Optimization *(The Butcher)* was applied to narrow down the search space for a high-fidelity low-thrust trajectory optimizer *(The Boss)*, which determined the best departure date and time of flight for each visit. Missions were analyzed for the applicability of low-thrust technology to debris removal mission design using simulated and real sun-synchronous debris data. The results from the actual data demonstrated the potential infeasibility of such a mission. The amount of thrust required to overcome the J2 perturbation effects would likely prove too great to justify applying low-thrust techniques for such a mission. Further analysis is required to conclusively prove this. More computational power would allow for low-thrust trajectory optimization to be performed within the beam search algorithm which may result in more realistic and fuel-minimal transfers. If that proves true, further work is required in the low-thrust hardware and optimization tools to make such a mission a reality.

**ACKNOWLEDGMENTS**

The authors would like to thank the ESA Advanced Concepts Team for their prompt for the 9th Global Trajectory Optimization Competition (GTOC), which inspired this work.¹ We would also like to thank Ghanghoon Paik, Guanwei He, Mollik Nayyar, Matthew Shaw and Jeffrey Small for their work on the GTOC solution that contributed to this paper.²
REFERENCES