TARGET SELECTION ALGORITHM FOR MULTIPLE-CAPTURE ORBITAL DEBRIS REMEDIATION SPACECRAFT MISSIONS

William Otis Hudnut

Submitted in Partial Fullfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Approved by:
Dr. Kurt S. Anderson, Ph.D., Chair
Dr. John A. Christian, Ph.D.
Dr. Jason E. Hicken, Ph.D.

Department of Aerospace Engineering
Rensselaer Polytechnic Institute
Troy, New York

[May 2021]
Submitted March 2021

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

This thesis would not have been possible without the support of several individuals.

First, I would like to thank Dr. Anderson for his guidance, support, and assistance in overcoming many of the issues I encountered on the way. Producing this thesis in the environment of the COVID-19 pandemic has not been easy, and his advice and support has made it much easier than I had any reason to expect.

My family's support has been invaluable as well. With the pandemic that has caused so much disruption of my academic plans over the past year, they were a constant pillar of support, without which I almost certainly would not have been able to complete this thesis.

Cameron Mehlman, a fellow student and co-author of our preceding work on the filter-sorter algorithm, provided many helpful suggestions regarding the ΔV calculation algorithm used for this work, and also made improvements to the data collection algorithm used to produce the databases used for this project.

I also want to thank Dr. Hicken for the insights he provided on overcoming the difficulties posed by the problem I chose to try to solve.

ABSTRACT

One major obstacle to successful orbital debris remediation is the determination of which pieces of debris are the most viable targets for capture and de-orbit. The viability of a target is determined by some combination of the debris' risk factor (a combination of its size, composition, and the orbit it occupies), the anticipated resource cost to find and capture the debris, and the underlying probability of successful intercept and capture of that target. The problem of selecting debris for capture by a multi-capture capable spacecraft is fundamentally a traveling salesman problem in which the traveler only has the resources to reach a very limited subset of the available destinations. This problem must be solved, however, since any debris removal spacecraft must be launched into orbit itself, likely producing some debris in its launch. In order to effectively reduce orbital debris populations, space-based solutions must be capable of multiple de-orbits per mission. The solution presented here uses iterative filtering and sorting of existing debris databases to produce lists of up to four targets (the maximum which can be achieved by the OSCaR debris capture CubeSat) which are feasible given the initial conditions of the spacecraft after its launch. A non-Hohmann two-burn node-to-node transfer calculation method was chosen, as it is very close to the energy-optimal transfer between two near-circular inclined orbits. This method is capable of capturing all valid target sets from target lists produced by Hudnut, Mehlman, and Anderson's database filter-sorter algorithm [8], with a run time of approximately 13 seconds on a laptop computer[1].

[1]MSI GS75. Intel i7-9750H processor at 2.60 GHz. Windows 10, MATLAB R2020b.

1. INTRODUCTION

1.1 History and Motivation

The first man-made satellite to enter orbit around Earth without re-entering the atmosphere was the United States' Vanguard I, launched in 1958. Since that day, the number of artificial bodies in Earth's orbit has increased greatly. Around 2000 operational satellites orbit the earth, filling important roles in military intelligence, telecommunications, and scientific missions. However, this number is minuscule compared to the numbers of non-operational satellites, and more importantly, pieces of orbital debris, which populate the Earth's orbit. The ESA estimates that over 100 million pieces of debris orbit the earth, most of which are smaller than 10 cm [4]. Debris, especially items in the 1-10cm size range, pose a critical threat to ongoing launches and space-based operations in Earth orbit. These items travel at hypersonic speeds, up to 7.8 kilometers per second, with potentially much higher object-to-object relative speeds, and are too small and fast for most satellites' on-board sensors to detect in time to evade. A collision at these speeds can severely damage satellites. Even flecks of paint can cause severe damage to sensitive equipment such as cameras, solar panels, or radiators, while an object the size of a pencil carries 300 kilojoules, approximately the energy of a small car traveling at highway speeds.

However, the threat of damage caused by any given piece of orbital debris is actually relatively small, as even low-Earth orbit (LEO) flight regimes comprise vast volumes. The major threat of space debris is caused by a phenomenon called the Kessler cascade or Kessler's syndrome [17]. This phenomenon occurs when space debris reaches a sufficient density that debris items begin to collide at high velocity with other debris items. This causes an exponential growth in the number of debris fragments in orbit, dramatically increasing the risk that any given satellite will suffer debris impact events during its operational lifetime. As a result, large regions of Earth orbit, particularly in LEO, could be rendered unusable by immense fields of nearly-undetectable, hyper-velocity shrapnel, creating an unsurvivable mission environment. In fact, debris tracking efforts have already begun to detect orbital

Portions of this chapter previously appeared as: *A Non-Hohmann Method for Orbital Element Database Pre-Processing. Proceedings of the Small Satellite Conference*, Advanced Concepts I, SSC20-WKI-09. http://digitalcommons.usu.edu/smallsat/2020/all2020/7/.

debris count increasing at rates greater than explained by known impacts and launch debris, indicating that debris-on-debris impacts are occurring and beginning to climb in frequency [17].

1.2 Proposed Solution

The only way to reduce or prevent a cascade in LEO regimes is to remove debris from orbit. Most debris in very low orbits will generally experience sufficient atmospheric drag to de-orbit it rapidly enough that it will not impact another spacecraft. Above an altitude of around 600-700 kilometers, however, the Earth's atmosphere is no longer sufficiently dense to impart sufficient force to de-orbit debris before it becomes likely that it will impact other debris or spacecraft. The solution, therefore, is to find some means to remove debris in these regimes, particularly in the 800-1200 km altitude polar orbit regime, which is both particularly densely populated and has a high proportion of crossing or even retrograde orbits, increasing the probability and damaging effect of collisions [1].

Current proposals are divided into two broad categories, ground-based and space-based approaches. Ground-based approaches generally utilize a large, high-powered laser system to vaporize small debris and ablate portions of large debris to cause deceleration and eventual de-orbit. However, these systems have the drawback of low power efficiency, targeting difficulties, and potential strategic implications, as such systems could also be used as offensive anti-satellite or anti-ballistic missile weapons.

Space-based approaches use a spacecraft to approach, physically capture, and de-orbit one or multiple items of debris. Virtually all space-based systems under development are currently focused on capturing and de-orbiting single debris items or obsolete satellites. OSCaR (Obsolete Spacecraft Capture and Removal), a mission under development by Dr. Kurt Anderson's research group at RPI, is a system comprised of 3-unit CubeSat spacecraft each designed to capture and de-orbit up to four pieces of debris in the 5-50 cm size range using a spring-launched net and electromagnetic drag tether system [1]. Multiple-capture systems are desirable due to the nature of space launches, most of which will introduce at least some debris of their own into orbit. A single-capture system cannot handle this, but a multi-capture system can, not only compensating for the debris of its own launch but also reducing the overall debris population.

1.3 Objective

OSCaR's target set is received from a grounded station where thousands of known pieces of debris (those currently cataloged by NORAD's Space-Track database and the ESA) are taken into consideration, evaluated and the final debris target set is selected. Because of the very limited size of CubeSats, OSCaR has both substantial limits on fuel and nets (each OSCaR is equipped with only 3 to 4 nets depending on the CubeSat configuration). The ephemeris data of cataloged pieces of debris in orbit around the earth is recorded by organizations such as NORAD, and can be obtained in the form of two-line elements [7].

From this ephemeris data, a list of valid targets (those targets which OSCaR is capable of reaching without exhausting its on-board fuel) can be extracted using a database filter-sorter algorithm [8]. However, this list by itself is insufficient. While each individual target on the list must eventually be captured and de-orbited, in order for the mission to be efficient at removing debris, multiple-target missions are desirable.

To determine which multi-target paths are possible given OSCaR's on-board maneuvering capabilities, an iterative method for possible target sets must be used. Starting with the list of valid targets, each must be once again checked against the other possible targets to determine if it is possible, starting from that position, to reach any other debris items. Iterating this process three times will give a full list of all possible 2, 3, and 4-target paths available to OSCaR given its on-board fuel. Once this is done, other criteria, such as positional certainty, remaining fuel, and collision potential of the debris can be assessed to determine which of these multi-target paths is preferable for OSCaR's mission.

2. METHODS

The overall goal of this program is to determine the feasible multi-target paths OSCaR could take, given its initial orbital state. The first step in achieving this is to retrieve the reduced target list provided by the database filter-sorter algorithm [8]. Once this is retrieved, it can be used without needing to refer to the larger database, as any possible secondary, tertiary, or quaternary target must also lie within that set of targets.

With this reduced target list, an iterative search is performed to determine which initial targets permit multi-target paths. From each target in the initial list, a further computation of the ΔV cost to access each other item in the debris list is performed. To reduce the computation cost of this iterative search, an analytic ΔV cost calculation algorithm is used. If any further targets are accessible from these later targets, the search is repeated from those targets using the same method. This search is repeated on every target in the list to ensure that no potentially valid paths are missed.

With the multi-target path list determined, a value can be applied to each multi-target path. Essentially, the path calculation acts as a constraint on the optimization problem of determining OSCaR's target set for any given mission. The optimization itself is over a discrete domain, the list of possible paths. The path valuation function can be made highly flexible to reflect any given mission's preferences for what debris is to be deorbited.

2.1 Pre-processing of TLE Data

The filter-sorter algorithm is discussed in great detail in *A Non-Hohmann Method for Orbital Element Database Pre-Processing* [8], and so its treatment here will be in general terms. In essence, the process performed by the filter-sorter algorithm uses three steps to convert a database's two-line element data into a feasible target set containing the important orbital element data. The first step is to process the data files to determine necessary parameters for ΔV calculation. The second step is to calculate the ΔV necessary to reach a given target. The third is to package the data into a format suitable for use by the optimizer.

Portions of this chapter previously appeared as: *A Non-Hohmann Method for Orbital Element Database Pre-Processing. Proceedings of the Small Satellite Conference*, Advanced Concepts I, SSC20-WKI-09. http://digitalcommons.usu.edu/smallsat/2020/all2020/7/.

The first step is relatively simple. The current dominant format for orbital ephemeris data is the two-line element (TLE) standard, which already contains the orbital parameters necessary to characterize a spacecraft's orbital state. This data can then be used to calculate the ΔV cost to access each piece of debris, completing the second step. Finally, the filter strips out any items which have access costs in excess of the ΔV budget specified by the user in the set-up of the algorithm.

Once the algorithm has run its calculation and filtered the results, it returns three files. The first contains the TLE data of the accessible debris and information about the transfer orbit used to access it. The second contains the classical orbit parameters of the accessible debris orbits and the remaining ΔV after accessing that orbit. The third saves the initial state of the spacecraft input during the setup of the algorithm.

2.2 Derivation of the ΔV Calculation Algorithm

In most cases for a non-Hohmann orbital transfer, time-optimal transfers are preferred. These transfer maneuvers are usually calculated by a method which iteratively solves Lambert's problem [16]. However, while these solutions are good for exact intercepts between spacecraft, they are computationally intensive due to their iterative nature. Therefore, an analytical method of trajectory and ΔV calculation was derived in order to save computational time.

The following derivation is a method to calculate the ΔV required for a two-burn non-Hohmann transfer between two inclined elliptical orbits, called orbits 1 and 2, where these orbits have arbitrarily oriented apse lines. The departure burn occurs at a point on the line of nodes on the initial orbit, and the insertion burn occurs at a point on the line of nodes on the other orbit, nominally the final orbit, opposing the initial point.

Beginning with the orbit elements for orbits 1 $(a_1, e_1, \Omega_1, i_1, \omega_1, \theta_1)$ and 2 $(a_2, e_2, \Omega_2, i_2, \omega_2, \theta_2)$; where (for $j = 1, 2$) a_j is the orbital semimajor axis of the trajectory, e_j is the orbital eccentricity, Ω_j is the right of ascension of the ascending node, i_j is the orbital inclination with respect to the equatorial plane, ω_j is the argument of periapsis, and θ_j is the actual anomaly; the orbital elements of the transfer orbit T $(a_t, e_t, \Omega_t, i_t, \omega_t, \theta_t)$ must be calculated in order to determine the ΔV cost to place OSCaR in the target debris orbit.

From the orbital elements of orbits 1 and 2, the direction cosine matrices $[{}^N\mathbf{C}^{PF_1}]$ and $[{}^N\mathbf{C}^{PF_2}]$ which relate the orientations of the perifocal reference frames of orbits 1 and

2 respectively to the inertial reference frame N, with its basis vectors $\hat{n_1}, \hat{n_2}, \hat{n_3}$ can be calculated.

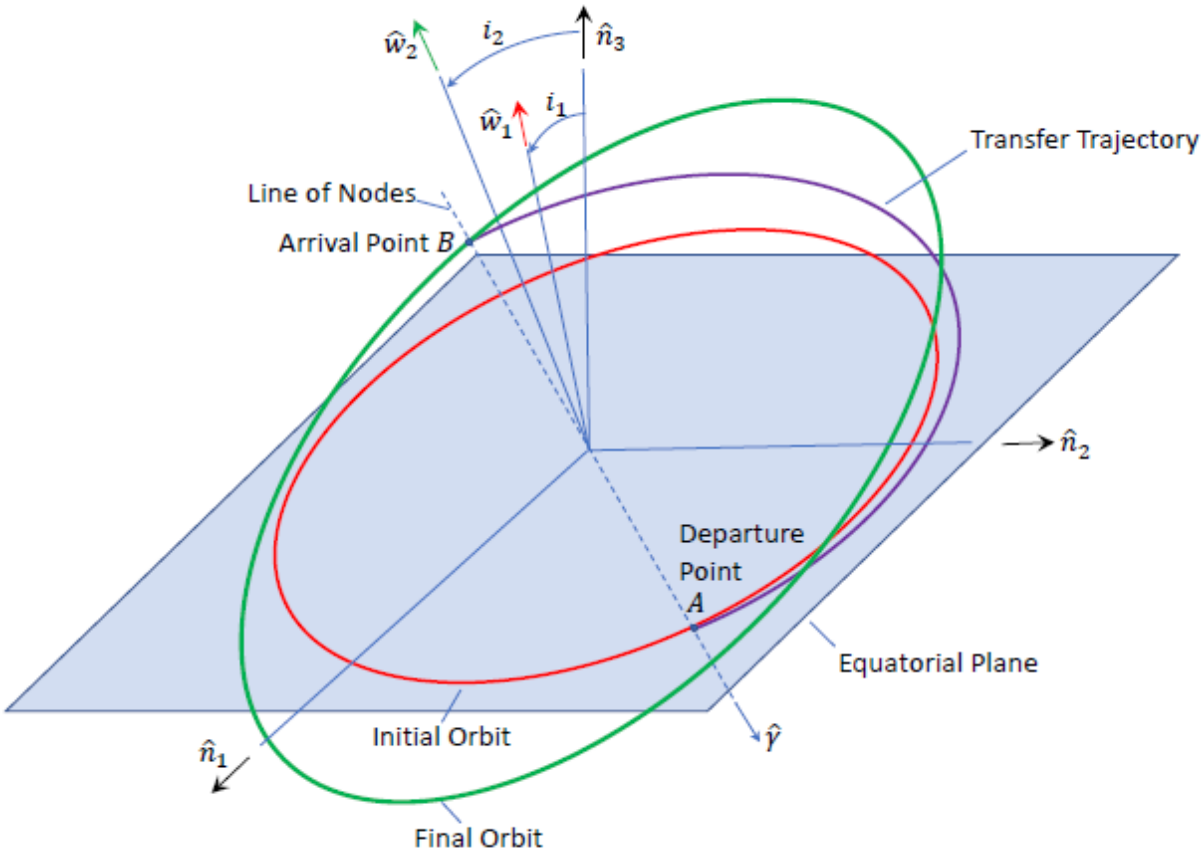

Figure 2.1: Node-to-node transfer between inclined elliptic orbits

The unit vector of the line of nodes $\hat{\gamma}$ is given by

$$\hat{\gamma} = \frac{\hat{w}_1 \times \hat{w}_2}{\hat{w}_1 \times \hat{w}_2}, \tag{2.1}$$

as shown in Figure 2.1, where $\hat{w}_1$, $\hat{w}_2$ are the unit vectors fixed in the inertial reference frame parallel to the orbital angular momentum vectors of orbits 1 and 2 respectively. The matrix representations of these vectors in the N reference frame correspond to the third columns of $[{}^N\mathbf{C}^{PF_1}]$ and $[{}^N\mathbf{C}^{PF_2}]$, respectively.

Defining $\theta_{\gamma 1}$ and $\theta_{\gamma 2}$ as the angles between the first two perifocal basis vectors (the vector representations of which in the N basis form first columns of $[{}^N\mathbf{C}^{PF_1}]$ and $[{}^N\mathbf{C}^{PF_2}]$ respectively) and $\hat{\gamma}$, it can be shown that the two terminal radii are $r_1(\theta_{\gamma 1})$ and $r_2(\theta_{\gamma 2})$, using

the formula

$$r_j(\theta) = \frac{a_j(1-e_j^2)}{1+e_j\cos\theta_{\gamma j}}, \tag{2.2}$$

using appropriate values of a, e, θ.

Since orbital velocities, and therefore required ΔVs for turning maneuvers, are higher at lower orbits, the method assumes that the ΔV at the lower terminal radius is performed entirely along the flight path of the spacecraft at that point, and that all turning required for the transfer is performed at the higher terminal radius. This will adequately constrain the problem of solving for three parameters $a_t, e_t, \theta_{\gamma t}$ of the transfer orbit. It should be noted that in all situations during the implementation of this algorithm in code, the inverse tangent operation was carried out using the atan2 function to avoid ambiguities associated with the normal inverse tangent operation.

To solve for these three unknowns, the following three equations are used (assuming $r_1 < r_2$, although the equations can be modified to function using the reverse assumption):

$$K_1 = \frac{a_t(1-e_t^2)}{1+e_t\cos\theta_{\gamma t}} = \frac{a_1(1-e_1^2)}{1+e_1\cos\theta_{\gamma 1}} = r_1(\theta_{\gamma 1}), \tag{2.3}$$

$$K_2 = \frac{a_t(1-e_t^2)}{1-e_t\cos\theta_{\gamma t}} = \frac{a_2(1-e_2^2)}{1+e_2\cos\theta_{\gamma 2}} = r_2(\theta_{\gamma 2}), \tag{2.4}$$

$$\alpha_t = \tan^{-1}\frac{e_t\sin\theta_{\gamma t}}{1+e_t\cos\theta_{\gamma t}} = \tan^{-1}\frac{e_1\sin\theta_{\gamma 1}}{1+e_1\cos\theta_{\gamma 1}} = \alpha_1. \tag{2.5}$$

K_1 and K_2 are placeholder symbols for the known values of $r_1(\theta_{\gamma 1})$ and $r_2(\theta_{\gamma 2})$, while α_t is the flight path angle of the transfer orbit at the point of departure, A, which is defined as equal to α_1, the flight path angle of the original orbit at the point of departure.

The following is a rearrangement of equation (2.5) which will be useful in the solution, specifically,

$$\alpha_t + \alpha_t e_t\cos\theta_{\gamma t} = e_t\sin\theta_{\gamma t}. \tag{2.6}$$

Equation (2.3) can be rearranged to acquire an expression for a_t as

$$a_t = \frac{K_1(1+e_t\cos\theta_{\gamma t})}{1-e_t^2}. \tag{2.7}$$

Substituting this into equation (2.4) yields

$$K_2(1 - e_t \cos\theta_{\gamma t}) = K_1(1 + e_t \cos\theta_{\gamma t}), \tag{2.8}$$

which can be rearranged into an expression for e_t as

$$e_t = \frac{K_2 - K_1}{(K_2 + K_1)\cos\theta_{\gamma t}}. \tag{2.9}$$

Substitution of the expression for e_t into equation (2.6) gives

$$\alpha_t + \alpha_t\left(\frac{(K_2 - K_1)\cos\theta_{\gamma t}}{(K_2 + K_1)\cos\theta_{\gamma t}}\right) = \frac{(K_2 - K_1)\sin\theta_{\gamma t}}{(K_2 + K_1)\cos\theta_{\gamma t}},$$

$$\alpha_t\left(1 + \frac{K_2 - K_1}{K_2 + K_1}\right) = \frac{K_2 - K_1}{K_2 + K_1}\tan\theta_{\gamma t},$$

$$\alpha_t\left(\frac{K_2 + K_1}{K_2 - K_1} + 1\right) = \tan\theta_{\gamma t} \longrightarrow$$

$$\tan\theta_{\gamma t} = \frac{\alpha_t(2K_2)}{K_2 - K_1} \longrightarrow$$

$$\theta_{\gamma t} = \tan^{-1}\left(\frac{\alpha_t(2K_2)}{K_2 - K_1}\right). \tag{2.10}$$

Now that an expression has been found to calculate $\theta_{\gamma t}$ from known quantities K_1, K_2, α_t, the value calculated for $\theta_{\gamma t}$ can be used in equation (2.9), and these values for $\theta_{\gamma t}$ and e_t are used in equation (2.5) to get a_t.

Using the equations for orbital angular momentum, radial, and tangential velocity, specifically

$$h = a(1 - e^2), \tag{2.11}$$

$$v_r = \frac{\mu}{h} e \sin\theta, \& \tag{2.12}$$

$$v_\theta = \frac{\mu}{h}(1 + e\cos\theta). \tag{2.13}$$

v_{1rA}, $v_{1\theta A}$, the radial and tangential velocities of orbit 1 at departure point A; v_{trA}, $v_{t\theta A}$, the radial and tangential velocities of the transfer orbit at departure point A; $v_{t\theta B}$, v_{trB}, the radial and tangential velocities of the transfer orbit at arrival point B; and v_{2rA}, $v_{2\theta B}$, the radial and tangential velocities of orbit 2 at arrival point B can all be calculated.

Since $v_1^A \cdot v_t^A = v_{1A} v_{tA}$ is known because the initial burn is constrained to be in the direction of the original velocity vector at point A, this can be used to find

$$\frac{v_1^A}{v_{1A}} v_{tA} = v_t^A, \tag{2.14}$$

and since ${}^N r_1^a = {}^N r_t^A$, ${}^N v_t^A$ can then be found using the following manipulation

$$\{r_1^A\}_N = [{}^N\mathbf{C}^{PF_1}]\{r_1^A\}_{PF_1}, \tag{2.15}$$

$$\{v_1^A\}_N = [{}^N\mathbf{C}^{PF_1}]\{v_1^A\}_{PF_1}, \tag{2.16}$$

$${}^N v_t^A = \frac{{}^N v_1^A}{v_{1A}} v_{tA}. \tag{2.17}$$

With ${}^N r_t^A$, ${}^N v_t^A$, $[{}^N\mathbf{C}^{PF_t}]$ can be found using

$${}^N h_t = {}^N r_t^A \times {}^N v_t^A, \tag{2.18}$$

$${}^N e_t = \frac{{}^N v_t^A \times {}^N h_t}{\mu} - \frac{{}^N r_t^A}{r_{tA}}. \tag{2.19}$$

With these vector quantities, the perifocal frame unit vectors and the direction cosine matrix of the transfer orbit can be quickly assembled:

$$\hat{p}_t = \frac{{}^N e_t}{e_t}, \quad \hat{w}_t = \frac{{}^N h_t}{h_t}, \quad \hat{q}_t = \hat{w}_t \times \hat{p}_t,$$

$$[{}^N\mathbf{C}^{PF_t}] = [\hat{p}_t, \hat{q}_t, \hat{w}_t],$$

which can be used to get

$$\{v_t^B\}_N = [{}^N\mathbf{C}^{PF_t}]\{v_t^B\}_{PF_t}. \tag{2.20}$$

The following relation is also found

$$\{v_2^B\}_N = [{}^N\mathbf{C}^{PF_2}]\{v_2^B\}_{PF_2}. \tag{2.21}$$

$\Delta V_1 = {}^N v_t^A - {}^N v_1^A$, and $\Delta V_2 = {}^N v_2^B - {}^N v_t^B$. The sum of these two values, ΔV_1 and ΔV_2, is the total cost of the transfer maneuver, ΔV_{tot}.

The above method was implemented as a function in MATLAB R2020b, and used to

estimate the individual ΔV to transfer from a specified initial orbit to each orbit in a target data set.

2.3 Path Valuation and Selection

In order to select one of the the multi-target paths as the one which OSCaR will undertake, some selection criteria must be applied to the paths. Various selection criteria are possible, and can be applied, removed, or adjusted to reflect mission objectives. This section will discuss them as two categories. The first is criteria which impact probability of mission success. These criteria must be involved in any assessment of the value of a target path. The second category is those criteria which indicate the importance or potential threat of individual debris items on the path.

2.3.1 Criteria Influencing Mission Performance

The first category of criteria that any path selection algorithm must consider is those criteria which impact OSCaR's ability to perform its mission, the capture and de-orbit of the debris items on its selected path. Of these, the most important are the remaining ΔV budget on-board OSCaR, and the positional certainty with which each debris item can be located.

Remaining ΔV is an important criterion to assess because this represents the remaining maneuvering capability of OSCaR. Simply entering the orbit of the targeted debris item is insufficient to capture it. While careful timing of OSCaR's maneuvers can reduce or eliminate the need for phasing maneuvers to intercept the debris, it is still necessary to perform maneuvers to precisely locate the debris and engage it with OSCaR's net-tether de-orbit system. These maneuvers require the expenditure of maneuvering resources, so paths which capture a given number of targets while expending less fuel are preferred, as they leave more margin for these precise intercept maneuvers, and also allow OSCaR to expend fuel to accelerate the deorbit of the debris.

The positional certainty associated with each debris target is also a very important criterion to assess. However, at this time, it is the most difficult criterion to assess for each piece of debris. The reason it is so important to assess is linked to OSCaR's mission profile. Once OSCaR is co-orbital with its target, it maneuvers to a distance of approximately 20 km from the object, and attempts to locate it with onboard sensors. These sensors, at an

estimated range of 10 km from the target, have a combined field of view which permits it to locate targets which have perturbed up to 1 km in any transverse direction. If the target is not in fact located within that 2x2x20 km ellipsoid whose long axis lies on the direction of flight, OSCaR will have to perform further maneuvers to determine its location, or may in fact not be able to locate the target at all. This could result in a failure to capture that debris target.

However, assessing the positional certainty associated with a given debris target is very difficult. While modeling orbital perturbations for an object with known properties is a straightforward, albeit computationally expensive, proposition, the major difficulty associated with this criterion is that the current TLE data standard does not contain any data on the size or composition of the debris. This makes it nigh-impossible to build an accurate model for projected area or mass of the object, meaning that at the moment, only very crude estimations of probable orbital perturbations for debris targets can be made.

2.3.2 Criteria Influencing Target Value

The value for each individual debris target should exert a weaker influence on the selection algorithm than the mission critical criteria for the path. However, they can be useful for allowing launch providers and mission designers to assert control over the exact path chosen, even if it may have lower (but still acceptable) values in the mission performance categories. The criteria which influence target value can be split into two broad categories: target size, and the probability that the target will collide with other debris or spacecraft in the near future.

Target size is, much like positional certainty, impossible to assess from TLE data alone. However, if it can be ascertained, it can be used as a metric for target value. First, an object with a larger projected area will have a higher probability of collision with other debris or spacecraft, simply because it sweeps a larger volume of space. Secondly, a larger or more massive target presents a greater threat if it were to impact another object. If that object were to be a functional spacecraft, a large impactor could completely destroy it, while a small impactor might only damage systems. Additionally, the large impactor will generate more and larger fragments if it breaks up on impact, as well as being more likely to produce large quantities of fragmentation from the object it impacts.

Probability of target collision is actually more easily assessed. For example, the Space

Track database provided by NORAD also provides a tool which can be used to assess the probability of collision of a given orbital body with any orbital body in their database. If it is desirable and practical to do so, each item in the target list could be so assessed using that tool, or a similar tool could be developed to perform that function.

3. RESULTS

The target selection algorithm is able to run to completion without errors or incorrect calculation of values for all tested cases using real TLE data from the NORAD database. The minimum ΔV cost for a two-target path in the target sets assessed was 246.8 meters per second. The minimum ΔV cost found for a three-target path was 448.7 meters per second. The minimum ΔV cost found for a four-target path was 763.8 meters per second. The target sets used to produce these values were generated using an initial orbit at an inclination of 98.6°, a semi-major axis of 7178 km, an eccentricity of 0.02, a right ascension of 180°, and an argument of perigee of 230°. This initial orbit had been previously determined to be in a regime of particularly high debris densities [1] [8].

3.1 Analysis of a Sample Mission

To demonstrate the essential functionality of the optimizer, a sample mission analysis is performed below. The initial mission definition was selected to place a multi-capture spacecraft in an initial state. For this example the initial state was an Earth-centered orbit with a semi-major axis of 7,178 km, eccentricity of 0.02, right ascension of 180°, inclination of 98.6°, and argument of perigee of 230°. The spacecraft was assumed to have an initial ΔV budget of 500 meters per second, selected as an optimistic but realistic estimate for near-future CubeSat propulsion modules. Running the filter-sorter algorithm [8] gives a 93-item list of accessible targets. The launch was assumed to take place on May 26th, 2019 at midnight.

Using this 93-item list as the input, the target selection algorithm found 4 possible two-target paths and 1 possible three-target path. No four-target paths were feasible with the ΔV budget given from the initial conditions specified.

Once these paths were determined, the path valuation algorithm assessed the potential value of each path. The path valuation algorithm used for this example considered as metrics the remaining ΔV budget, the positional certainty, the out-of-plane angle of the debris, and the inherent value of debris capture. Using a series of weights to ensure broadly equivalent scales for these metrics, each path's value was constructed by assessing the value of each debris item on the path, aggregating these values, and then adding a value for the remaining

ΔV for the total path. Each path's value was independently assessed, and the list of paths was then sorted from highest to lowest aggregate value.

Remaining ΔV budget was assessed for value over the entire path. This was to ensure that an accurate estimate could be made of the remaining budget for intercept maneuvers. Positional certainty was assessed by propagating estimated solar radiation pressure over the time since last observation of the debris in the database. As mentioned before, this propagation is unreliable since the database does not contain data on size and composition of the debris, but it functions as a rough estimation. The out-of-plane angle of the debris was used as a heuristic for target value, as it is a rough metric for how much energy that debris item would have if it were to collide with other debris in the target set. Finally, a factor was added to represent the fact that de-orbiting any individual piece of debris is better than not de-orbiting any debris. This had the secondary effect of strongly emphasizing the value of higher target-count paths. The table of valued paths is displayed in Table 3.1.

Table 3.1: The `valuedPaths` array output by the target selection algorithm

Index 1	Index 2	Index 3	Index 4	ΔV remaining	ΔV value	Targets value	Total value
28344	30837	29806	0	51.331	0.10228	7.1165	7.2188
28344	30837	0	0	253.28	0.69538	5.3298	6.0252
31854	4774	0	0	160.62	0.37951	4.4833	4.8629
31926	32343	0	0	41.063	0.07979	2.1803	2.2601
30837	37690	0	0	141.38	0.32480	-0.02881	0.2960

The first four columns of the table contain the debris indices for the given path, and the fifth column contains the ΔV remaining after completion of the path, in meters per second. The sixth column contains the value assessed for the ΔV after weighting. The seventh column contains the sum of the values assigned to each piece of debris on the path, and the eighth column contains the sum of this and the ΔV value. For further explanation of how these values are assessed, see Appendix A.2.

To understand the data presented in this table, each path can be inspected. From this, a further understanding of the merits and demerits of each path can be assessed from the function of the optimizer.

First, it must be noted that the two-target precursor path for the three-target path is included. This is deliberate. As can be seen, the two-target path saves over 250 m/s of ΔV, and only has a reduction in total targets value of 1.8. In some cases, such as if either target on the two-target path were too large to effectively de-orbit using only the electromagnetic

tethers on OSCaR, or if one of the targets were a time-critical target requiring the expenditure of additional ΔV to capture before it impacted a functioning satellite, it might be desirable to pursue the smaller target path.

The three-target path is the highest-valued path despite its relatively low remaining ΔV score. This is because of the scoring for capturing an additional target. However, the algorithm assigns a base value of 2.0 to each target. The relative change in value between this path and its precursor (the second most valuable path on the list, interestingly) is only about 1.8. This decrease in relative value of the final piece indicates that it likely has a relatively high degree of positional uncertainty, and that pursuing it may not succeed.

The lowest-valued path on the list also demonstrates this. Since we know that the base value of each target is 2.0 (arbitrarily assigned to represent the inherent value of removing any debris from orbit), the targets are each losing an average of 2.0 points due to positional uncertainty. This is sufficient to place it at the bottom of the list, despite it having 100 m/s of ΔV advantage over the second-worst target path. Note, however, that the second target path is also losing value due to positional uncertainty. With a default targets value of 4.0 for a two-target path, 6.0 for a three-target path, and 8.0 for a four-target path, any path with values lower than these should be immediately suspicious. Paths below these thresholds (under the current optimizer) risk the debris not being in a volume where OSCaR can localize and capture it.

The inverse case is demonstrated in the second-best path. With an expected targets value of 4.0, we see that the optimizer actually values this path at 5.3298. This indicates that the two targets listed have a very low predicted perturbation value, or have very high out-of-plane value (and therefore relatively high kinetic energy, were they to impact spacecraft or debris near OSCaR's initial orbit). Since the third target in the three-target path adds a sub-nominal value to the total targets value, it may be worth considering the two-target version of the path rather than the three-target version in this case, especially since the three-target path also has much less remaining ΔV for maneuvers that could potentially permit it to capture the debris item with the higher positional uncertainty.

However, all these valuations are predicated on the valuation algorithm chosen for the mission. One modification which has been tested is to remove the inherent valuation of each piece of debris. This puts a much stronger weight on the certainty of capture of each debris item for the valuation algorithm. In this case, as might be expected, it in fact indicates that

the two-target path is better than the three-target path due to the higher average positional certainty along that path.

3.2 Valid Path Counts

Using the aforementioned high debris density initial orbit, the correlation between ΔV budget and number of potential multi-target paths was explored. The algorithm was run with initial ΔV budgets of between 200 and 1000 meters per second, in increments of 50 m/s. This generated the data displayed in Figure 3.1.

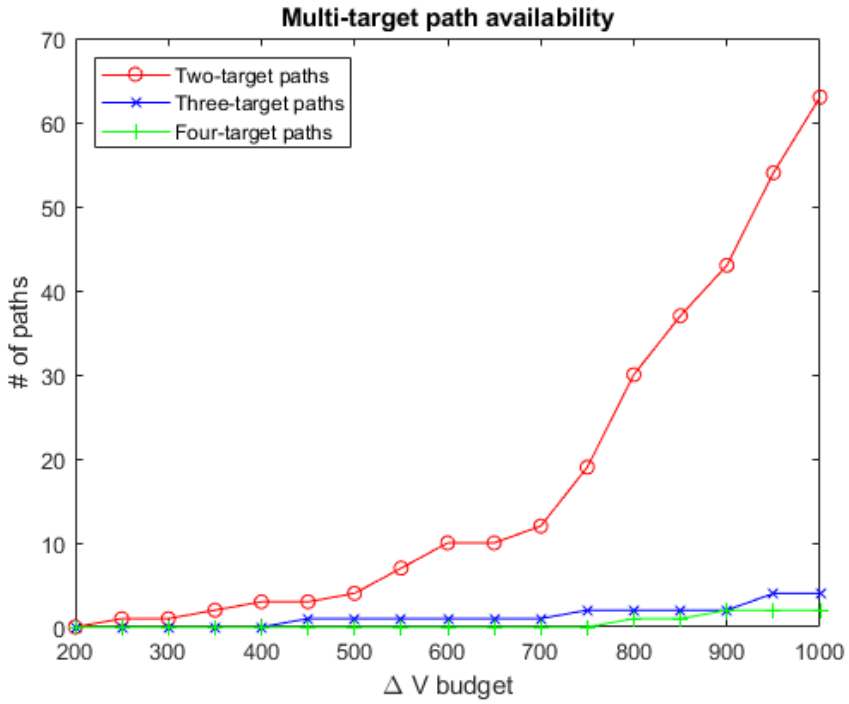

Figure 3.1: Count of available multi-target paths over increasing ΔV budget

These data can be assessed to make several interesting conclusions. First, it is obvious that an attempt to build a multi-capture spacecraft operating in this regime will require a minimum ΔV budget of at least 250 m/s. While it would be theoretically possible to put OSCaR on a perfect intercept path to the first debris target from launch, thereby vastly reducing the cost to capture the initial target, this would only be possible if OSCaR were to be made the primary payload of the launch, or in an extremely convenient launch which happened to be heading almost directly for the initial debris target. Since OSCaR is designed from its inception to be launched as secondary, not primary, payload on other missions, it will be important to use propulsion systems capable of producing at least this much impulse.

Another datum worthy of note is that the count of multi-target paths available to a multi-capture spacecraft, at least in densely populated regions, increases superlinearly with increasing ΔV budget. The superlinear effects begin to appear around the point at which the spacecraft's ΔV budget reaches 550 meters per second for two-target paths. It is likely that the superlinear effects would require much more ΔV than was considered here to appear for three- and four-target paths. The reason that higher-ΔV budgets were not considered is that current propulsion technologies are unlikely to be able to provide such high ΔV budgets in the near future.

3.3 Computational Performance

An evaluation of algorithm run-time and trends may be useful. Using some navigational algorithms, computation time can be a major consideration. However, this algorithm is designed to use an analytical method to calculate trajectory ΔV. This is useful in reducing computation time. While the optimizer does use an integrator to estimate perturbation, the number of times the integrator is run is relatively low, and the function being integrated is simple.

In order to characterize the algorithm's performance, it was run several times with varying numbers of valid targets. As expected, the algorithm does become significantly more computationally intensive with increasing target counts, as the iterative nature of the assessment down each target path significantly increases the number of operations necessary. Average run times were calculated for four ΔV budgets. The characterization was performed using an MSI GS75 laptop, running Windows 10, using the MATLAB R2020b distribution, on an Intel i7-9750H processor. The results are presented in Table 3.2.

Table 3.2: Runtimes for varying target counts associated with given ΔV budgets

ΔV budget ($\frac{m}{s}$)	Target count	Average runtime (s)
250	40	0.1012
500	93	1.359
750	164	14.86
1000	222	79.48

As can be seen, run time does increase non-linearly with target count increases. This is an unavoidable consequence of the iterated path search method. However, it is worth noting that the run time for reasonable near-future CubeSat ΔV budgets was quite manageable. Most current CubeSat propulsion technologies suitable for use in OSCaR produce between 200 and 400 m/s of ΔV, with some optimistic proposals offering between 500 and 600 m/s. For these ΔV budgets, run time was under 2 seconds. Even with highly optimistic proposals, average run time was under 80 seconds. This indicates that the program is suitable for rapid assessment of mission profiles. New factors or data can be added and the reassessment run in sub-minute times for most probable missions. This means that OSCaR missions can be flexible, even after launch has been completed, so long as onboard flight profiles can be updated.

3.4 Case Study

To illustrate the usage of the algorithm further, a case study was made of the SAOCOM-1B launch. SAOCOM-1B is an Argentine Earth observation satellite which was launched on August 30th, 2020. The mission was launched aboard a Falcon 9 Block V, along with two ride-share payloads, GNOMES-1 and Tyvak 0172 [15]. Had OSCaR been mission ready at the time of this launch, it is quite possible that it would have been worth carrying an OSCaR spacecraft along as well. An analysis of the potential mission will be performed here.

For the purposes of this analysis, it will be assumed that OSCaR separated from the Falcon 9 lift vehicle co-orbitally with the SAOCOM-1B satellite. This places it in an orbit at an altitude of 620 km, with an inclination of 97.89°, eccentricity of 0.0001527, right ascension of 235.2°, and argument of perigee of 84.14° [13]. A ΔV budget of 600 m/s will be assumed.

Using this initial condition, data was pre-processed and then passed through the target selection algorithm. Only 6 debris targets were accessible from this orbit with the 600 m/s

ΔV budget, none of which were near enough to each other to make a multi-target path feasible. This clearly indicates to the user that this launch would have been unsuitable for a debris mitigation mission.

However, for a test mission, there was one debris target of particular interest. When the individual single-target paths were passed to the optimizer, NORAD Catalog ID 28481 had an individual target value of 6.7712 points. The only way in which the optimization algorithm could have reached this level of value for a single debris target would be an extremely high positional certainty. Given this, this mission may well have been suitable for a test launch of OSCaR to validate debris rendezvous and capture operations.

This assessment demonstrates one possible issue with OSCaR's mission profile as currently imagined. With the exception of some Chinese satellite launches, this launch was one of the most likely prospects in 2020 for debris collection, as most other launches were to either much lower flight regimes or geostationary orbits [14]. As ride-sharing on Chinese launches in the near future is unlikely, due to political tensions between China and the United States, OSCaR may suffer difficulties in finding launches which can convey it to near the 800-1200 km altitude sun-synchronous orbit regime which contains the highest debris densities [5].

4. DISCUSSION

This target selection algorithm is a useful tool for future multi-capture debris mitigation mission planning. In addition to determining all feasible multi-target trajectories that the spacecraft in question can take, it also provides transparent, cogent, and flexible target selection. It also uses modular ΔV cost calculation and target valuation methods, rendering it flexible in methodology. Should a user wish to alter its method for calculation of target values or ΔV, or use a different data standard, it is easily convertible to use those methods.

The algorithm is currently designed to return mission profiles for spacecraft capable of capturing up to four debris items. It can be modified to permit targeting for fewer captures, by deactivating (through commenting out) the iterative loops, or to permit targeting for more captures by adding further iterative checks for additional targets. However, as this algorithm is designed for use in the context of the OSCaR mission, this four-target capability is the design goal.

This algorithm is not intended to return actual maneuvering data for use in OSCaR missions. Once it is used to select the preferred target path for the OSCaR mission, the maneuver timing, duration, attitude, and other data still will have to be generated for use in the actual mission.

4.1 Known Issues and Potential Solutions

The algorithm's current design is suitable for the mission context it was created for. However, some conditions or alterations to the mission profile may cause the algorithm to be sub-optimal for use. In those cases, the algorithm should either be altered, or potentially replaced with a different approach.

The first situation to be considered, and the most likely near-future major alteration that the algorithm might require, would be in the case of a propulsion module switch on the OSCaR spacecraft. Currently, OSCaR's design calls for a green monopropellant propulsion module. However, current designs for such propulsion modules which are suitable for use in a 3U CubeSat suffer from highly limited ΔV budgets. Developments in CubeSat power systems since this specification was made may make other propulsion systems, particularly iodine ion thrusters, attractive for use in OSCaR. These systems will offer a much higher

ΔV budget, but cannot be modeled as providing an impulsive transfer. This means that the current ΔV calculation will not return correct access costs for targets.

The solution for this would be to create a new ΔV calculation algorithm. The algorithm would necessarily be more computationally expensive, as long-burn maneuver models require integration of thrust over time to calculate the maneuvers being made and are not tractable for analytic solutions, but would be easy to interface with the optimizer, simply replacing the current impulsive-transfer calculator.

Another likely near-future issue that the algorithm might encounter would be a change in database used to find target data. This issue is more likely to occur than a propulsion module switch, but the solution is even simpler. The inputs to the target selection algorithm are straightforward requirements for the classical orbit parameters of each acceptable debris item, plus some timing data. All of these should be accessible from any useful orbital debris database, as most if not all use the TLE data storage standard.

In these cases, all that will need to be done is some minor alterations to the algorithm's input processing to ensure that the correct data is being sent to the correct functions within the program. In fact, it is likely that any such data processing could be done in the front-end, within the filter-sorter program, without requiring any alterations to the target selection algorithm itself.

The final major issue that could be encountered is a structural issue with the algorithm. Due to its iterative nature, as seen in Section 3.3, its computational resource cost grows dramatically with increasing target counts. None of the tested target counts were necessarily prohibitively expensive, as all cases tested (up to 222 targets) had run times under 80 seconds. However, in the future, for OSCaR to remain useful as a multi-capture system, it will need to be able to engage large target sets. This could occur either by dramatic increases in CubeSat propulsion capability, which are unlikely, or by improvement in the databases used for target selection. This is much more likely.

The debris databases used for testing this algorithm contained only on the order of 10,000 items. However, current databases track relatively little debris. The European Space Agency estimates that there are millions of pieces of space debris in the 1-20 cm size range OSCaR is designed to target. If orbital debris remediation becomes a more significant issue, and organizations begin to dedicate greater efforts to tracking and identifying debris, the database populations will increase, which will increase the number of known debris locations

that OSCaR can access from a given orbit, and hopefully dramatically increase target counts.

It will, of course, be up to future mission planners to determine what computational times are acceptable for this initial target selection algorithm. If it is primarily being used for mission planning, in non-time-critical situations, even very long run times may be acceptable. If, on the other hand, it is being used for rapid evaluations on orbit, more emphasis will need to be placed on its speed of execution.

If it is determined that the current algorithm is overly slow for the application it is being used in, the recommended solution is to attempt to use the data generated by this algorithm to train a neural network model. Neural nets are likely to be effective in solving the problems presented by the selection of debris targets, as they have been in effectively playing chess, Go, and similar games, and for a similar reason. The objective of finding the "best" multi-target trajectory is quite similar to planning moves ahead in chess, as the spacecraft should only take certain energy- or time-optimal paths from target to target. Unfortunately, the author did not identify neural networking's suitability for this problem in time to adequately investigate the potential effectiveness of this method.

4.2 Evaluation of Future Prospects

The debris capture and removal mission that this target selection algorithm is intended to facilitate is critical for the continued military, commercial, and scientific utilization of near-Earth orbital space. OSCaR in particular represents a highly useful and feasible solution for this mission, as it is designed from the ground up as a secondary ride-share payload for other missions, reducing launch costs and allowing launch providers the ability to demonstrate their commitment to good space citizenship.

However, even cursory analysis of OSCaR's mission profile using this target selection algorithm indicates that OSCaR's approach has some significant flaws in the current environment surrounding the debris remediation mission. The usage of a 3U CubeSat, while attractive in many ways, inherently limits onboard system capability and fuel budgets, which exacerbates some issues caused by the current approach to debris tracking. Additionally, even this algorithm's utility is limited by the debris database format and sizes.

The current orbital ephemeris data standard, the two-line element system, has some inherent flaws in its usefulness for debris tracking. First, it notably lacks any significant size or mass categorization, which is crucial for accurate modeling of orbital perturbations and

also for determination of target values. The only information that the two-line element model provides is the estimated BSTAR coefficient of the debris. While this can be useful in orbital modeling, it is far from sufficient for target evaluation. Additionally, the TLE data format has very limited precision. While it is useful for tracking large satellites, when tracking small satellites or debris items, the positional precision may not be sufficiently granular to accurately represent the location of the satellite. In order to make the OSCaR mission more feasible, a new data standard providing further sizing, mass, and possibly albedo data, as well as greater precision in its positional data, would be highly desirable.

Additionally, current target databases are startlingly sparse, with only about 26000 total items currently being tracked out of an estimated 900,000+ [4], and primarily track operational and defunct spacecraft and relatively large debris. This leads to the issues identified in Section 3.2, where it was found that in order to utilize OSCaR's fourth capture capability, the CubeSat's propulsion system would have to acheive an unrealistically high 800 m/s ΔV budget. If greater efforts are made in the near future to identify and track more and smaller debris targets, ride-shared launches with realistic ΔV budgets are more likely to be able to utilize the full capacity of OSCaR and similar multi-capture spacecraft.

OSCaR's design as a 3U CubeSat is also a limiting factor on mission capability. It places fairly strict limitations on onboard sensing capability, ΔV budget, and mission lifetime, which will need to be worked around. Especially with the current relatively sparse debris data, it may be difficult to find worthwhile debris remediation missions which can be performed on ride-shared launches.

However, even with the current limitations associated with TLE data standards, it would still be valuable for OSCaR to be ride-shared along with other launches. Some debris should be accessible on many or even most higher-altitude LEO launches, and while the preferred multi-target launches may not be possible, any effort to remove persistent space debris is preferable to no effort made. As mentioned previously, collision-based debris growth is already being observed, and efforts to reduce the population will push the inflection point of the dreaded exponential growth pattern further into the future, creating more time for better debris tracking and remediation methods to be developed.

4.3 Conclusions

The target selection algorithm presented here is a useful and necessary part of OSCaR mission planning, and can be adapted to conform to future alterations in mission profile, spacecraft design, or mission environment. Used for mission operations planning, it will produce real, achievable target paths in a reasonable time frame. The valuation algorithm will allow mission planners to select a target path which is not only feasible but also contains valuable debris items with a high probability of capture.

REFERENCES

[1] J. Baker, T. Canter, B. Gollin, C. Kirchner, J. Lyons, J. Roman,and J. Shen, "OSCAR II Critical Design Report." Rensselaer Polytechnic Institute, 2016. [Print]

[2] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics* (2nd ed.) Reston, VA: American Institute of Aeronautics and Astronautics, 1999. [Print]

[3] V. A. Chobotov, H. K. Karrenberg, C. Chao, J. Y. Miyamoto, and T.J. Lang. *Orbital Mechanics* (V. A. Chobotov, Ed.) Washington, DC: American Inst. of Aeronautics and Astronautics, 1991. [Print]

[4] European Space Agency, "ESA Meteoroid and Space Debris Terrestrial Environment Reference". Accessed on May 20, 2020. Available: https://sdup.esoc.esa.int/.

[5] European Space Agency, "Space Debris by the Numbers". Accessed 12 May 2020. Available: https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers.

[6] Gooding, R. H. "A Procedure for the Solution of Lambert's Orbital Boundary Value Problem." Celestial Mechanics and Dynamical Astronomy, 48, 145-165, 1990.

[7] P. Hoddinott, "Tracking of Space Debris from Publically Available Data." Master's thesis. Rensselaer Polytechnic Institute, Troy, NY, 2018. [Print]

[8] W. O. Hudnut, C. P. Mehlman, and K. S. Anderson. "A Non-Hohmann Method for Orbital Element Database Pre-Processing." *Proceedings of the Small Satellite Conference*, Advanced Concepts I, SSC20-WKI-09, 2020.

[9] G. Leitmann, et al. *Optimization Techniques With Applications to Aerospace Systems* (G. Leitmann, Ed.) London: Academic Press, 1962. [Print]

[10] J. P. Marec. "Transferts Optimaux Entre Orbits Elliptiques Proches". Doctoral thesis, Faculty of Sciences of Paris, 1967 (NASA, Trans.) Accessed on June 19, 2020. [Print]

[11] M. M. Peet, "MAE 462 Lecture 8: Impulsive Orbital Maneuvers." Lecture. Accessed on 31 May 2019. Available: http://control.asu.edu/Classes/MAE462/462Lecture08.pdf.

[12] M. M. Peet, "MAE 462 Lecture 9: Bi-elliptics and Out-of-Plane Maneuvers." Lecture. Accessed May 31, 2019. Available: http://control.asu.edu/Classes/MAE462/462Lecture09.pdf.

[13] SAIC, "Two-Line Element Bulk Database". Accessed 25 September 2020. Available: http://www.space-track.org.

[14] European Space Agency, "Satellite Missions Directory". Accessed 16 February 2021. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/.

[15] European Space Agency, "Satellite Missions Directory". Accessed 16 February 2021. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/s/saocom.

[16] F. Sun and N. X. Vinh, "Lambertian Invariance and Application to the Problem of Optimal Fixed-Time Impulsive Orbital Transfer". *Acta Astronautica*, 10(5-6), 319-330, 1983.

[17] J. Corbett. Micrometeoroids and Orbital Debris (MMOD). Accessed Available: https://www.nasa.gov/centers/wstf/site_tour/remote_hypervelocity_test_laboratory/micrometeoroid_and_orbital_debris.html.

APPENDIX A
CODE OPERATIONS DESCRIPTION

This appendix will discuss the MATLAB functions and scripts which implement the target selection, from pre-processing to final valuation. It will describe the inputs and outputs required for each to run successfully, and discuss any known failure modes.

The program consists of two MATLAB scripts, `Data_Presort_301`, which is the main script for the filter-sorter database pre-processing tool, and `Target_Selection_200`, which is the main script for the target selection algorithm. To evaluate a potential mission profile for OSCaR, these must be run sequentially, with `Data_Presort_301` run first, and then `Target_Selection_200` run using the outputs provided by the pre-sorter.

A.1 `Data_Presort_301`: Operation and Dependencies

`Data_Presort_301` is the database pre-processing tool used to provide a feasible target list for the target selection algorithm. It operates by using a ΔV calculator function to calculate the fuel cost to transfer from a user-specified initial orbit to each possible target orbit in a data set specified by the user. Once it has calculated the ΔV cost to access each possible target, it produces a reduced list of only those target orbits which are feasible with the user-specified ΔV budget.

The inputs to `Data_Presort_301` are provided by the user by editing the script. The user must provide both the initial state and the database to be filtered. The initial state is specified at line 57-63, and must include the semi-major axis in meters, the eccentricity, the right of ascension of the ascending node in radians, the inclination in radians, the argument of perigee in radians, and the true anomaly in radians of the initial orbit, and the allocated ΔV budget in meters per second.

The database should be stored as a .mat file. It is loaded at line 68, using MATLAB's `load` command. If it is not stored in the working directory for the file, the directory path must be specified. Currently, the program is designed to use data files provided by Philip Hoddinott's database access tool [7]. These will contain a date of file creation and a double array named `tle_final`. This array will be an n-by-eleven array, where n is the number of debris-sized targets found in the database, containing selected orbital data extracted from

the TLEs in the database.

Each row in `tle_final` contains eleven pieces of data associated with an individual debris target. The first column will contain the NORAD catalog ID number of the item. The second column contains the epoch time at which the TLE data of the debris was last updated. The third column is the inclination, in degrees, of the debris orbit. The fourth column is the right of ascension of the ascending node of the orbit in degrees. The fifth column is the eccentricity of the orbit. The sixth column is the argument of perigee of the orbit. The seventh and eighth columns are the mean anomaly in degrees and mean motion in revolutions per day of the debris. The ninth column is the period of revolution in seconds. The tenth and eleventh columns are the semi-major and semi-minor axes of the debris orbit.

Other database formats can be used so long as the classical orbital parameters of the debris can be readily extracted from them. This may require changing some indices in the data set sort section of the script so that the correct data is being passed to the ΔV calculator.

Once the list of accessible targets has been produced, the script will package three sets of data and save them as .mat files. One will be the list of possible targets with full data, including ΔV cost to access and some orbital information such as node of departure and whether the transfer was from a higher orbit to a lower orbit or vice versa, along with all data provided in the database format used. The second is a list of possible targets with the catalog ID number, the classical orbit parameters, and the remaining ΔV after orbital transfer. This reduced list can be handier for some operations in target selection. The third is the saved initial state of the spacecraft. It will also output in the command window a statement that the database has been successfully loaded and the number of targets in the database before sorting the database, and a statement of number of valid targets found and the run time once the sort has been completed.

`Data_Presort_301` requires an external ΔV calculator to find transfer ΔV costs. It is currently written to use the `nH_delta_V` analytic non-Hohmann two-impulse calculation algorithm. Requirements, inputs, and outputs of this calculator will be discussed in Section A.3, as it is also used by the target selection algorithm.

A.2 `Target_Selection_200`: Operation and Dependencies

As this script's function as the target selection algorithm is discussed extensively in this thesis already, this section will focus on clarifying details of its input requirements and dependencies. For an in-depth discussion of its outputs and method of solution, read Section 2.3 and Chapter 3.

`Target_Selection_200` requires as inputs the three data files saved to the workspace by `Data_Presort_301`, and a user-specified time of initiation of maneuvers. The format for the time is provided as a comment in the script, and should match the time formatting used in the database. It requires two external functions, a ΔV calculator and a path valuation function. It is currently set up to use the `nH_delta_V` ΔV calculator and the `PathValues` path valuation function.

A.2.1 `PathValues` and `pieceValues`

The `PathValues` function requires as inputs an array containing the catalog ID numbers of the debris on each valid path, the initial time for mission start, the full database information associated with all debris on each path, and the initial state of the spacecraft. It extracts the debris data associated with each path, and passes that data, along with the mission start time and spacecraft state, to the `pieceValues` function. When the piece values are returned for each debris item on the path, the function sums those values for each path and adds the logarithm of the remaining ΔV to add value for saving maneuvering capability. The logarithm is here used to scale the value of the ΔV to the same scale as the weights from `pieceValues`, since there will be diminishing returns from retaining huge amounts of ΔV for intercepts.

The `pieceValues` function assigns a value to each debris item on its path. It uses the `epochTimeDelta` and `perturbation` functions to determine the magnitude of perturbation expected for the debris, and uses the logarithm of 1000 meters divided by the expected perturbation in meters as the weight associated with perturbation. This produces a positive value for debris that has an expected perturbation less than 1000 meters and a negative value for debris with an expected perturbation of over 1000 meters. It additionally adds a factor to account for how much relative inclination the debris has with respect to the spacecraft's initial orbit. This, as discussed previously, is a simple heuristic for probability of collision with other debris in the spacecraft's range.

`PathValues` returns essentially the same array as `Target_Selection_200`. `pieceValues` returns an n-by-one array containing the values of each piece of debris whose data was passed to it in the array of debris data created in `PathValues`.

`PathValues` requires as a dependency the `pieceValues` function. `pieceValues` requires as dependencies the `epochTimeDelta` and `perturbation` functions. The target lists array passed to `PathValues` must contain only valid catalog ID numbers in the first one to four columns, and the last column containing the remaining ΔV for the path.

A.3 `nH_delta_V`: Operation and Dependencies

The ΔV calculation algorithm is the backbone of the constraints applied to the target selection algorithm, and is also a critical part of the database pre-processing algorithm. Accordingly, it will be discussed in greater detail here.

The derivation for the mathematical method used to calculate the ΔV cost of transferring from one orbit to another by this calculator is presented in Section 2.2 of this paper. This method has a few noteworthy characteristics. First, it is an analytical method, meaning that it does not require expensive iterations to calculate the ΔV cost of the transfer. This means that even though the target selection algorithm may be expensive to run, the calculation of ΔV accounts for relatively little of the computation time of the algorithm.

It is also an optimal solution for the case which accounts for most capture maneuvers which OSCaR will be performing. Since the spacecraft's ΔV budget is so low, all the orbits it will be transferring between will be constrained to low eccentricity and relative inclination. This also happens to well characterize the orbits of most debris in low-Earth orbit, as most spacecraft, excepting some military reconnaissance satellites, tend to occupy near-circular orbits in certain orbital regimes, and therefore the debris generated by those launches and/or collisions between those spacecraft tends to remain near those regimes. It is worth noting that for some orbital transfers, particularly transferring between highly elliptical orbits, this transfer is definitely sub-optimal. However, these transfers also tend to have very high ΔV costs, even for their optimal transfers, and so are unlikely to be valid targets for OSCaR in any case.

`nH_delta_V` requires two input vectors, one each for its initial and target orbit. These vectors shoudl contain the classical orbit parameters for the initial and target orbits: the semimajor axis in meters; the eccentricity; and the right ascension, inclination, argument of

perigee, and true anomaly in radians. Given these two input vectors, the code then calculates two possible transfer paths, one each from the ascending node and descending node of the orbit plane intersections from the initial orbit. The algorithm selects the transfer with the lower ΔV cost, and returns that ΔV cost and several parameters associated with the transfer orbit's characteristics. These parameters include an index to indicate whether the departure was from the positive or negative line of nodes direction (1 for positive, 2 for negative), an index to indicate whether the initial orbit was the location of the off-tangential burn (1 for no, 2 for yes), and the true anomaly of the arrival node on the target orbit.

`nH_delta_V` can encounter some numerical errors due to the construction of the method. In order to avoid those, which are usually associated with extremely small relative inclinations or extremely small eccentricities, the code includes several checks to ensure that if those variations are too small, the relevant values are set to those which reduce the code to a Hohmann transfer orbit, which is the limiting case of this method for zero relative inclination and zero eccentricity. In those cases, the method will not quite return the correct ΔV, but since in those cases the relevant values would have to be very small, on the order of machine precision, the loss of accuracy for those cases is acceptably small.

APPENDIX B
DEBRIS DISTRIBUTION CHARACTERISTICS

In the near future of OSCaR mission planning, it is apparent from the analyses performed in Chapter 3 that it will be difficult to find valid multi-target paths. In addition, it is evident that the number of valid multi-target paths that can be found from available data is directly correlated to the density of debris in the area of LEO that OSCaR launches to. Accordingly, some study of debris distribution in low-Earth orbit should be made prior to mission planning to ensure that multi-target paths are available in the planned mission regime.

This potential issue has been recognized for some time during the OSCaR conceptualization and design phase. The 2016 OSCaR critical design review performed some study of orbital debris densities. The team found that certain areas of near-Earth orbit contain significantly higher debris densities than others. Particularly notable for high debris density was the volume around an altitude of 800 km, with an inclination of approximately 95-100° [1]. They produced a plot of debris spatial densities which is reproduced below in Figure B.1.

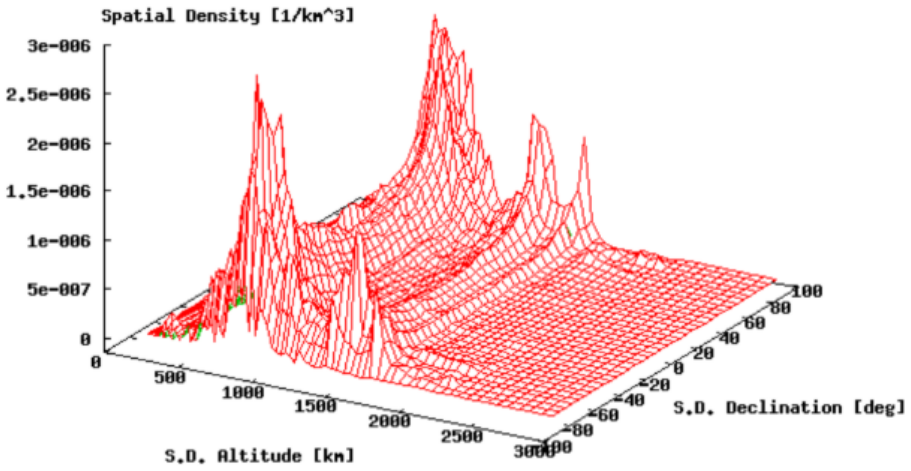

Figure B.1: Debris densities in low Earth orbit with respect to altitude and inclination [1]

Portions of this chapter previously appeared as: *A Non-Hohmann Method for Orbital Element Database Pre-Processing. Proceedings of the Small Satellite Conference*, Advanced Concepts I, SSC20-WKI-09. http://digitalcommons.usu.edu/smallsat/2020/all2020/7/.

This figure can be used to provide some intuition regarding spatial debris densities. Hudnut, Mehlman, and Anderson, in their 2020 paper on debris database pre-processing, used the tool they generated (at the time, the `Master_Analysis_300` script, which would become the `Data_Presort_301` script after further refinement) to perform some further investigation of actual accessible target counts in some near-Earth orbit regimes [8].

Most orbital regimes, as expected, contained very little debris that was being actively tracked. However, as was also expected, the orbital regimes corresponding with the "spikes" on the density graph returned useful and relatively high debris counts. When the filter-sorter algorithm was run with an initial orbit with near-zero eccentricity, an inclination of 98.6°, semi-major axis of 7.287×10^6 meters, maximum ΔV of 450 m/s and varying right ascension of ascending node between 0° and 180° with step size of 1°, the following results were obtained.

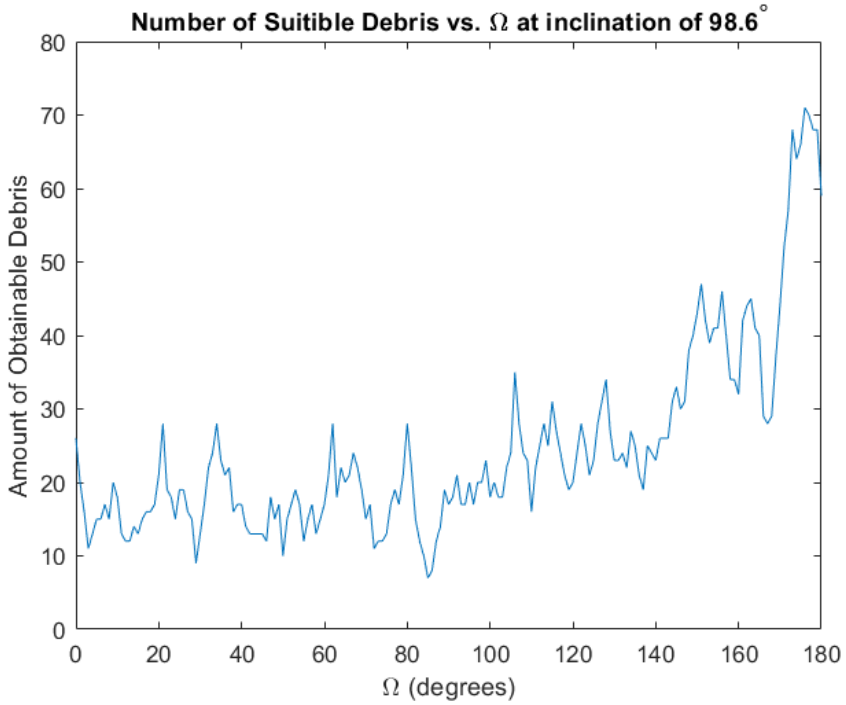

Figure B.2: Debris counts at fixed inclination with varying RAAN [8]

At the time the data which produced the results in Figure B.2 were obtained, there was a major spike in debris incidence at a right of ascension of approximately 180 degrees. It is likely that this debris spike is due to the presence of debris from either the Iridium 33/Cosmos-3 collision or the 2007 Chines ASAT test. It is worth noting that in data sets

captured on later dates, due to the high inclination and therefore relatively high precession rate due to J2 effects, this debris spike had moved. It is therefore recommended that these studies be performed with the data set to be used for a given mission, especially if near-polar orbit regimes are to be targeted. As these regimes tend to contain a majority of dangerous debris, such studies would be a best practice for all future OSCaR launches.