Orbit Raising Utilizing a Hybrid Low/High-Thrust Bi-Elliptic Transfer Orbit for Generic High-Ratio Circular Orbits

Richard Connor Johnstone*

University of Colorado - Boulder, Boulder, Colorado, USA

Often in astrodynamics and mission design, efficiency in terms of propellant use is the most important factor (as it is often a large cost-driving factor) in the planning of the mission. Usually, an impulsive Hohmann transfer is the simplest and most efficient transfer available. However, in certain cases, such as raising a circular orbit from one value a semi-major axis to another much higher one, a trade-off can be made to increase efficiency by increasing time of flight. The impulsive Bi-Elliptic transfer and continuous low-thrust Spiraling are two methods commonly used to achieve this tradeoff. Through simulation, this paper attempts to explore the possibility of combining the two maneuvers to investigate a new design regime, somewhere in between the time of flight and efficiency ranges of the two maneuvers.

Nomenclature

a	Semi-Major axis, kg
e	Eccentricity
i	Inclination, deg
Ω	Right Ascension of the Ascending Node, deg
ω	Argument of Periapsis, deg
θ^*	True Anomaly, deg
α	Thrust angle wrt to the $\hat{\theta}$ direction (in-plane), deg
β	Thrust angle wrt to the $\hat{\theta}$ direction (out-of-plane), deg
I_{sp}	Engine Specific Impulse
g_0	Acceleration due to gravity on Earth's surface
m	Mass
T	Thrust
p	Semi-latus Rectum
h	Specific angular momentum
ΔV	Change in Velocity ("Delta V")
μ	Gravitation Paramter

I. Introduction

Mission Design is a process driven, when possible, by mathematical optimization, and otherwise by carefully chosen heuristic methods that have been optimized under certain relevant constraints. A few of these methods have been previously mentioned. This paper will focus on the following maneuvers and their usefulness to mission design: the Hohmann transfer, the Bi-Elliptic transfer, and the low-thrust Spiraling transfer.

*Student, Aerospace Engineering

A. Hohmann Transfers

Probably the simplest and most useful orbit transfer in astrodynamics is the Hohmann transfer. It is the most fuel efficient method to transfer between two circular orbits that utilizes only two short thrusting periods. For the purposes of this discussion, the thrusting periods will be assumed to be zero (nearly true for sufficiently high engine thrust). In a typical Hohmann transfer, the spacecraft first thrusts entirely prograde in order to enter an elliptic transfer orbit that has a periapsis at the initial circular orbit radius and an apoapsis at the final circular orbit radius. Then, at apoapsis, the spacecraft thrusts once again in the prograde direction to circularize its orbit. These two thrusts are seen in Figure 1 below.

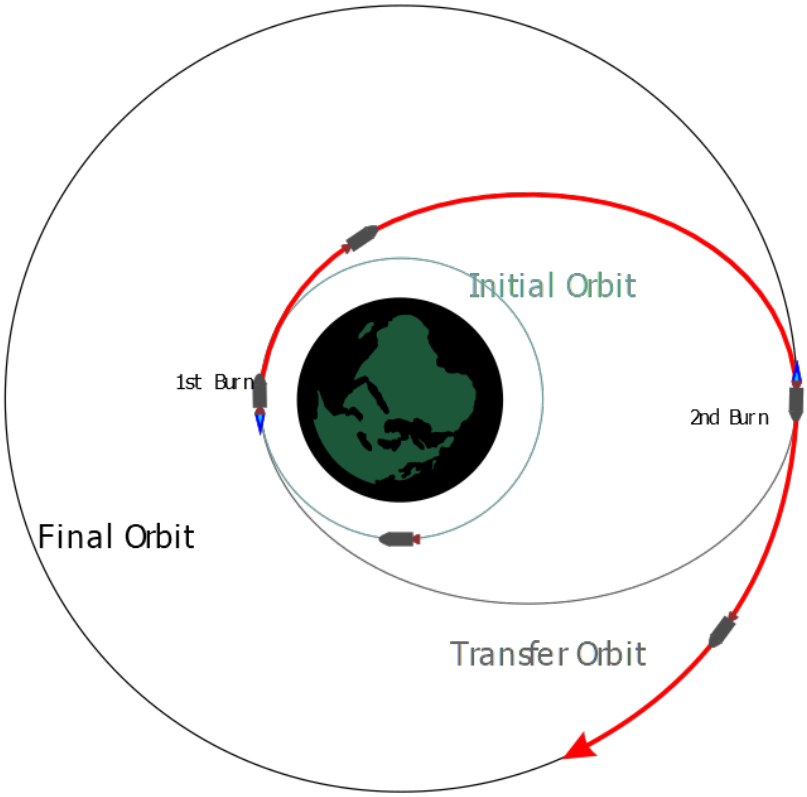

Figure 1. Diagram of a typical Hohmann transfer [1]

In many cases, this is not only the fastest orbital transfer maneuver in terms of flight time, but the most efficient in terms of ΔV. The ΔV for this maneuver can be calculated by subtracting the velocity magnitudes at each point (since all maneuver velocities are in the same direction) and the time of flight (TOF) can be found by the half-period of the transfer orbit:

$$\Delta V_1 = \sqrt{\frac{2\mu}{a_i} - \frac{\mu}{a_t}} - \sqrt{\frac{\mu}{a_i}} \tag{1}$$

$$\Delta V_2 = \sqrt{\frac{\mu}{a_f}} - \sqrt{\frac{2\mu}{a_f} - \frac{\mu}{a_t}} \tag{2}$$

$$\Delta V_{hohmann} = |\Delta V_1| + |\Delta V_2| \tag{3}$$

$$TOF = \pi\sqrt{\frac{a_t^3}{\mu}} \tag{4}$$

Where a_i, a_t, and a_f are the initial, tranfer, and final semi-major axes respectively.

B. Bi-Elliptic Transfers

While the Hohmann transfer is the most efficient two-impulse maneuver to get between two circular orbits, under the circumstance that the two orbits have very different radii, a third maneuver can somewhat improve efficiency at the expense of flight time. This maneuver is called a Bi-Elliptic maneuver and works by raising the semi-major axis in the same manner as the Hohmann maneuver, but passing the target radius and instead increasing the apoapsis to some very large value (usually described as infinity for a best-case scenario, although this is impractical). At the apoapsis of this very eccentric orbit, raising the periapsis to the target radius requires very little ΔV. Upon the craft's return to periapsis, the orbit is then circularized, as before, but this time with a retrograde thrust. See Figure 2:

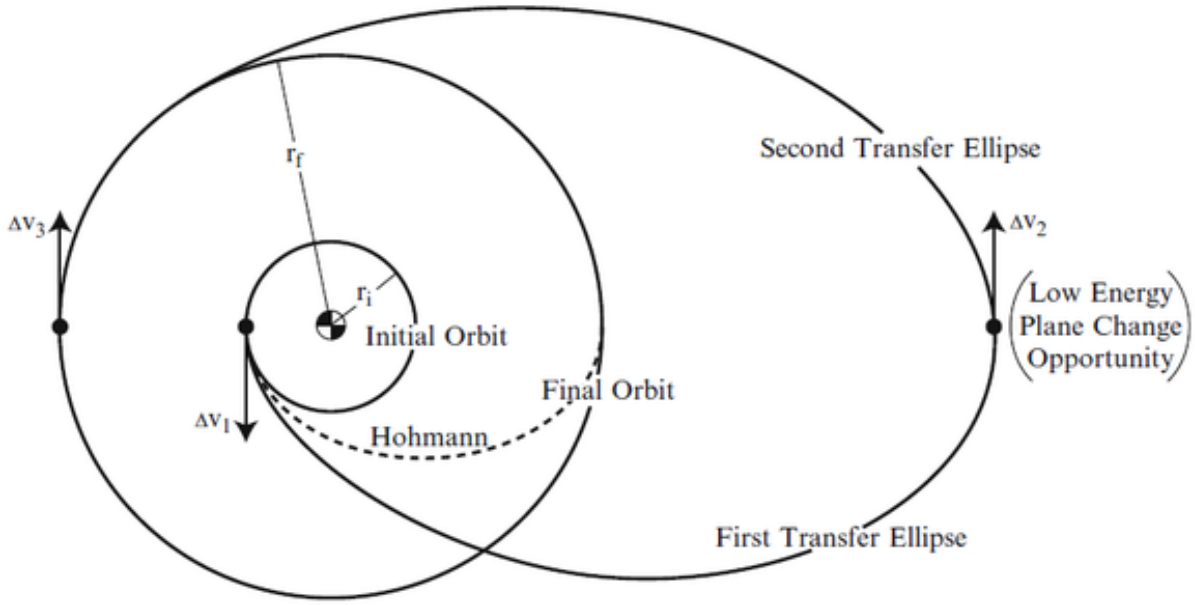

Figure 2. Diagram of a typical Bi-Elliptic transfer

For orbit ratios larger than 11.94 this will be a more fuel-efficient maneuver than the Hohmann transfer. However, since the transfer ellipses are much longer and the transfer isn't complete until both half-orbits are completed, the TOF for this maneuver is much, much longer.

$$\Delta V_1 = \sqrt{\frac{2\mu}{a_i} - \frac{\mu}{a_{t,1}}} - \sqrt{\frac{\mu}{a_i}} \tag{5}$$

$$\Delta V_2 = \sqrt{\frac{2\mu}{a_f} - \frac{\mu}{a_{t,2}}} - \sqrt{\frac{2\mu}{a_i} - \frac{\mu}{a_{t,1}}} \tag{6}$$

$$\Delta V_3 = \sqrt{\frac{\mu}{a_f}} - \sqrt{\frac{2\mu}{a_f} - \frac{\mu}{a_{t,2}}} \tag{7}$$

$$\Delta V_{Bi-Elliptic} = |\Delta V_1| + |\Delta V_2| + |\Delta V_3| \tag{8}$$

$$TOF = \pi \left(\sqrt{\frac{a_{t,1}^3}{\mu}} + \sqrt{\frac{a_{t,2}^3}{\mu}} \right) \tag{9}$$

C. Continuous Low-Thrust Propulsion

Our assumption so far is that thrust is sufficiently high that orbit transfers can happen nearly instantaneously. This is generally true for chemical rocket propulsion, the most common propulsion method today. However, one of the primary goals of astrodynamics and propulsion research is to increase propellant efficiency. To calculate that, one must turn to the Tsiolkovsky Rocket Equation:[2]

$$\Delta V = I_{sp} g_0 \ln \left(\frac{m_0}{m_f} \right) \tag{10}$$

Where, for simplicity's sake, we will take $I_{sp} = \frac{T}{\dot{m} g_0}$ to be a constant for a given engine. From a practical standpoint, though, T often increases with increases in $\dot{m}$ and generally it has been found that the easiest way to increase specific impulse is by increasing the ratio $T/\dot{m}$ by decreasing T. Therefore, the most simple method for increasing the specific impulse of an engine is to create an engine that has a very low thrust and a low mass flow rate. Obviously, though, if the thrust is decreased then our assumption that a single, impulsive thrust can change our orbital parameters significantly will no longer hold. This has led to a branch of research in the concept of continuous, low-thrust orbital transfers, where the spacecraft is constantly (at least for a significant portion of the orbit) thrusting using a highly efficient low-thrust engine.

1. *Spiraling Transfers*

These engines are generally so weak that the target orbit cannot be achieved over the course of a single orbital period of thrusting. Therefore, the term used to describe an increase in semi-major axis using this type of engine is a Spiraling transfer. In general, the fuel consumption is much lower and TOF much higher for these transfers in comparison to chemical propulsion, far moreso than for the general Bi-Elliptic transfer in comparison to the Hohmann transfer. However, these values are more difficult to compute because of the long thrusting periods. The method used in this paper to model these values will be covered in Section III.

D. Problem Statement

Typically, for the purpose of high-level, early-stage design assumptions, we can think of these maneuvers (for high ratio orbit raising) as taking up certain regimes in the fuel efficiency vs. time of flight space where the Hohmann transfer is the least efficient and fastest, the Bi-Elliptic transfer is slightly more efficient and slower, and the Spiraling low-thrust transfer is significantly more efficient but much slower. The question that this paper will explore is whether a regime can be achieved in between the Bi-Elliptic and Spiraling regimes in terms of efficiency and TOF. The approach is to equip a spacecraft with two different engines. First, the spacecraft will use all of its chemical propellant to impulsively transfer to the first transfer orbit of a typical Bi-Elliptic manuever. Immediately after, the craft will begin thrusting with its low-thrust engine in order to spiral back to the target circular orbit.

In general, low-thrust maneuvers are determined via a control law. While in certain cases an exact mathematical optimization problem can be solved, the solution may not provide much in the way of physical insight for the mission designer. Therefore, heuristic control laws have been designed to provide actual

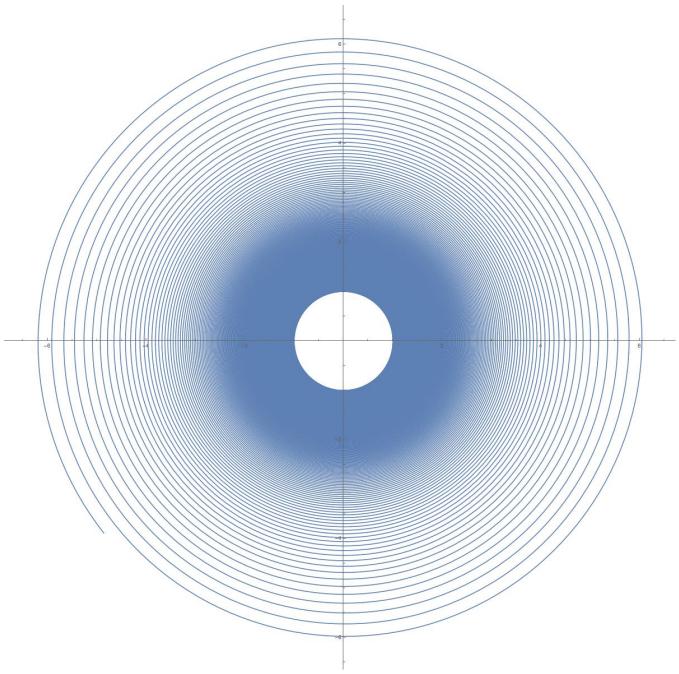

Figure 3. Diagram of a typical Spiral orbit raising

instructions for the spacecraft that come close to approximating the exactly optimized solution. Many of these control laws are available[34] and this paper uses the Proximity Quotient or Q-Law, which will be described in greater detail in Section III.

This paper will test four sample cases in order to gain insight into a more general orbit raising concept. Four sample spacecrafts with the same dry mass will start in a circular Low Earth Orbit (LEO) at $a = 6471\,\text{km}$ and will target a final circular orbit of $a = 100\,000\,\text{km}$, utilizing a Hohmann, Bi-Elliptic, Spiraling, and combination Bi-Elliptic/Spiraling transfer respectively. The propellant used (and the cost of the propellant) as well as the time of flight will be compared for each case.

II. Problem Overview

As discussed, we would like to explore the various regimes of the orbital transfers for a generic high-ratio orbit raising case. The goal is to determine whether or not the combination approach will allow us a new design regime in terms of cost and flight time.

A. Assumptions

The first and most important item to discuss in regards to achieving this result would be the assumptions that we must recognize in order to calculate the final solutions and determine whether they are satisfactory.

Our largest assumption will be that the the gravitational effects in the problem can be adequately described by the assumptions of the two body problem. Any perturbations due to third bodies, oblateness, etc. won't depend upon the path or method of thrust with two exception: first, for certain cases of very high orbit ratios, it is possible, particularly for the low-thrust cases, that the craft would come into close proximity with the Moon or far enough away that the Sun has a significant effect on the motion of the craft. Therefore, we should be careful to not select orbits on the order of the orbit of the Moon or larger, or to take those effects into consideration if we do consider those cases. Second, the effects of solar radiation pressure will have a stronger effect on the crafts with longer flight times. For the example in this paper, we ignore the effects of the Moon, but for a higher fidelity model, one would choose a slightly smaller Bi-Elliptic transfer radius in order to avoid those issues or plan the orbit around these constraints.

Secondly, we assume a constant specific impulse for the engine, independent of time. This should be a reasonably close approximation of the actual case, although in real life specific impulse does change slightly

with time and mass expulsion.

Thirdly we assume that, for the mission at hand, the chosen value of empty mass would be sufficient. Obviously, this would depend upon the mission parameters, but should at least allow us to make some generalizations for future study.

We also assume that the Q-Law control is sufficiently close to the optimized case for orbit transfer. The results of this simulation have been compared to the paper in which Q-Law was introduced[5] and that paper compares the values to the optimal case. The results are reasonably close to optimal.

B. Applications

The most obvious application for this type of transfer would be that of an orbit raising from LEO to Geostationary Earth Orbit (GEO). This orbit profile is often used for many different types of satellite applications. For any given mission profile the time of flight would be an important variable to consider and if this approach can improve cost over a Bi-Elliptic transfer without the time of flight cost of traditional Spiraling, then it would prove to be a useful tool for mission designers.

One important consideration in this approach is the design complexity of utilizing two different engines. However, I believe that this is less complex than one would intuit. Staging is a very common practice in spacecraft propulsion and one could consider the case of a typical two-stage to LEO rocket such as the Falcon 9. It would be reasonably easy to, for a sufficiently small payload, reach a final second stage orbit equal to the first transfer orbit of a Bi-Elliptic transfer utilizing only the second stage of the main rocket itself. The payload, containing only a low-thrust engine, could then be released to complete the Spiraling maneuver. This would require no more design complexity than a typical low-thrust satellite (though perhaps a cost trade-off with the launch provider to achieve the more energetic orbit).

Another option for the application of this approach is in achieving a large, circular orbit around an extraterrestrial body. A hyperbolic flyby could approximate the inital transfer orbit, then requiring very little fuel to achieve the circularized orbit via low-thrust methods. The hyperbolic flyby could be achieved by an impulsive rocket and the circularization could be done via low-thrust.

C. Limitations

As previously mentioned, the largest limitation to this approach is the presence of third bodies, namely moons. Spiraling orbits at high radii will sweep a large area around the planet, potentially causing issues when passing near other orbiting bodies. Other than that, most of the limitations to this method are practical in nature. The expectation is that these results will only be useful for a specific regime of efficiency and flight time, so that must be taken into consideration when designing this type of transfer.

It should be noted at this point, what a "good" simulation result would be. First, our expectation that the three typical transfers occupy different regimes in efficiency/TOF space should hold. We should also see the combination maneuver fit nicely in between the Bi-Elliptic and Spiraling regimes. Lastly, an even better result would be that there are even more significant cost savings for a given TOF regime than is expected by merely interpolating between the two surrounding cases.

III. Technical Approach

A. Dynamical Model

For the purposes of this simulation, we will model only the Earth and the spacecraft. Naturally, this leads to a two body problem, the assumptions of which were outlined in Section II. Three reference frames are considered for this model. Initial states and all states handled by the solver will be in an Earth-Centered-Earth-Fixed frame, using the $\hat{X}$, $\hat{Y}$, and $\hat{Z}$ direction convention. Also used in calculation of the thrust direction is the radial/along-track/cross-track frame using the convention $\hat{R}$, $\hat{\theta}$, $\hat{H}$. The final and most important frame used is the Keplerian orbital elements a,e,i,Ω,ω,θ^*, which will be used to calculate the Proximity Quotient.

B. Orbital Parameters

It will be important for us to understand the dynamics of the orbital elements. In particular, it will be useful to know their time derivatives for the calculation of Q. These can be determined analytically and are given

as:

$$\frac{da}{dt} = \frac{2a^2}{h}\left(e\sin\theta^* f_r + \frac{p}{r}f_\theta\right) \tag{11}$$

$$\frac{de}{dt} = \frac{1}{h}\left(p\sin\theta^* f_r + \left((p+r)\cos\theta^* + re\right)f_\theta\right) \tag{12}$$

$$\frac{di}{dt} = \frac{r\cos\left(\theta^*+\omega\right)f_h}{h} \tag{13}$$

$$\frac{d\Omega}{dt} = \frac{r\sin\left(\theta^*+\omega\right)f_h}{h\sin i} \tag{14}$$

$$\frac{d\omega}{dt} = \frac{1}{eh}\left(-p\cos\theta^* f_r + (p+r)\sin\theta^* f_\theta\right) - \frac{r\sin\left(\theta^*+\omega\right)\cos i f_h}{h\sin i} \tag{15}$$

And we will ignore the time derivative of true anomaly as it is not a commonly targeted orbital parameter, although Q-Law does allow for it. It should also be noted that the thrust components are:

$$f_r = f\cos\beta\sin\alpha \tag{16}$$

$$f_\theta = f\cos\beta\cos\alpha \tag{17}$$

$$f_h = f\sin\beta \tag{18}$$

At every point in our simulation, we will also want to compare to the best case scenario. Therefore it is important to know the optimal time derivatives for each of the orbital parameters in terms of both thrust angles and the true anomaly along the osculating orbit. This paper won't derive the values, but they can be found in Petropoulos.[5]

$$\dot{a}_{xx} = 2f\sqrt{\frac{a^3\left(1+e\right)}{\mu\left(1-e\right)}} \tag{19}$$

$$\theta^*_{a,xx} = 0 \tag{20}$$

$$\dot{e}_{xx} = \frac{2pf}{h} \tag{21}$$

$$\theta^*_{e,xx} = \pi \tag{22}$$

$$\dot{i}_{xx} = \frac{pf}{h\left(\sqrt{1-e^2\sin^2\omega} - e|\cos\omega|\right)} \tag{23}$$

$$\sin\theta^*_{i,xx} = -\left(e\cos\omega - \mathrm{sgn}^*\left(\cos\omega\right)\sqrt{1-e^2\sin^2\omega}\right)\sin\omega \tag{24}$$

$$\cos\theta^*_{i,xx} = -e\sin^2\omega - |\cos\omega|\sqrt{1-e^2\sin^2\omega} \tag{25}$$

$$\dot{\Omega}_{xx} = \frac{pf}{h\sin i\left(\sqrt{1-e^2\cos^2\omega} - e|\sin\omega|\right)} \tag{26}$$

$$\sin\theta^*_{\Omega,xx} = \left(e\sin\omega - \mathrm{sgn}^*\left(\sin\omega\right)\sqrt{1-e^2\cos^2\omega}\right)\cos\omega \tag{27}$$

$$\cos\theta^*_{\Omega,xx} = -e\cos^2\omega - |\sin\omega|\sqrt{1-e^2\cos^2\omega} \tag{28}$$

$$\dot{\omega}_{xx} = \frac{f}{eh}\sqrt{p^2\cos^2\theta^*_{\omega,xx} + \left(p+r\right)^2\sin^2\theta^*_{\omega,xx}} \tag{29}$$

$$\cos\theta^*_{\omega,xx} = \left(\frac{1-e^2}{e^3} + \sqrt{\frac{1}{4}\left(\frac{1-e^2}{e^3}\right)^2 + \frac{1}{27}}\right)^{\frac{1}{3}} - \left(-\frac{1-e^2}{e^3} + \sqrt{\frac{1}{4}\left(\frac{1-e^2}{e^3}\right)^2 + \frac{1}{27}}\right)^{\frac{1}{3}} - \frac{1}{e} \tag{30}$$

Where the function sgn^* is defined as:

$$\mathrm{sgn}^* x = \begin{cases} \mathrm{sgn}\, x & \text{if } x \neq 0 \\ \pm 1 & \text{if } x = 0 \end{cases} \tag{31}$$

C. Proximity Quotient

Now that we have our foundational equations for the orbital parameter dynamics, we can define our Q as:

$$Q = (1 + W_P P) \sum_{\text{œ}} W_{\text{œ}} S_{\text{œ}} \left(\frac{d\left(\text{œ}, \text{œ}_T\right)}{\dot{\text{œ}}_{xx}} \right)^2 \qquad \text{for œ} = a, e, i, \Omega, \omega \tag{32}$$

One can think of Q as a sort of "least squares" cost function from the current set of orbital parameters to the target parameters. This allows for a single, scalar value that can tell the algorithm exactly how far away from the target it is. Q will always be positive and will be equal to 0 when the target orbit is reached. This is actually a refined version of the Proximity Quotient,[6] which includes a couple of useful modifications:

P is a penalty function used to prevent the orbit from decaying to a periapsis that is too low at any given point where:

$$P = \exp\left(k \left(1 - \frac{r_p}{r_{p,min}} \right) \right) \tag{33}$$

Where the constant scalar k is chosen to determine the strength of the effect (chosen as 1 for these cases). $S_{\text{œ}}$ is a scaling function that helps determine the relative strength of a in comparison to the other orbital elements and ensures convergence.

$$S_{\text{œ}} = \begin{cases} \left(1 + \left(\frac{a - a_T}{m a_T} \right)^n \right)^{1/r} & \text{for œ} = a \\ 1 & \text{for œ} = e, i, \Omega, \omega \end{cases} \tag{34}$$

with scalar values $m = 3$, $n = 4$, and $r = 2$ used nominally. The final difference from the original Q-Law is the distance function:

$$d\left(\text{œ}, \text{œ}_T\right) = \begin{cases} \text{œ} - \text{œ}_T & \text{for œ} = a, e, i \\ \cos^{-1}\left(\cos\left(\text{œ} - \text{œ}_T\right)\right) & \text{for œ} = \Omega, \omega \end{cases} \tag{35}$$

With the principal value of $\cos^{-1}$ used.

D. Derivative of Q

Now that we know what Q is, we need an algorithmic way to drive Q to zero. This is accomplished by determining the time derivative of Q, which depends upon the thrust angle:

$$\frac{dQ}{dt} = \sum_{\text{œ}} \frac{dQ}{d\text{œ}} \frac{d\text{œ}}{dt} \tag{36}$$

where the time derivatives of our orbital elements come from Eqns 11-15 and the derivatives of Q with respect to the orbital elements must be determined numerically (via finite differences method). This value can then be minimized using the `fminsearchbnd`[7] function available online for Matlab to find the thrust angles that minimizes $\frac{dQ}{dt}$.

E. Optimal Derivatives and effectivity

The only piece missing now from an effective algorithm to simulate a near-optimal thrust profile is a way to determine at which points along the orbit to thrust or not. This can be determined by defining an effectivity for the driving of Q. Namely, when the ratio of the optimal time derivative of Q with respect to thrust angles to the optimal time derivative of Q with respect to thrust angles AND true anomaly is above a certain threshold (dubbed effectivity), then the algorithm tells the engines to fire. The optimal time derivative of Q with respect to thrust angle and true anomaly is found by replacing the time derivatives of the orbital elements with the optimal time derivatives (from Eqns 19-30) in Eqn 36. This effectivity will be a value between 0 and 1 and the cutoff can be chosen anywhere in that range. A higher effectivity cutoff will be more fuel-efficient, but the transfer will take more revolutions to complete. A lower effectivity cutoff will have a shorter time of flight but use more propellant.

F. Algorithmic Approach

After determining the relevant values for the Proximity Quotient, an algorithm can be developed. First, the spacecraft constants are set. For the impulsive maneuvers, a thrust and specific impulse equivalent to the SpaceX Merlin 1D engine were chosen and for the low-thrust craft, the values were chosen to be equivalent to 2 NASA NEXT engines (like those used on the Dawn spacecraft). This ensures experimental relevance. Then, the target orbital elements were chosen, as decribed above and in the next Section, as well as the weights for our Q (chosen so that a and e were controlled and i, Ω, and ω were left free. Initial conditions were set depending upon which case was to be simulated and the the solver (`ode45` from Matlab) options were chosen. My solver used a Relative Tolerance of $1e-10$ and an absolute tolerance of $[1e-3, 1e-3, 1e-3, 1e-7, 1e-7, 1e-7, 1e-4]$. Frankly, this is a looser tolerance than would be preferred. However because the Q and Q derivative functions get called so many times at every call to the ode solver, the simulation runs rather slowly in comparison to natural orbital motion and involved so many revolutions (hundreds, in some cases) that these were chosen to keep the simulations under a few hours. The solver was also set up to stop when Q reached a certain cutoff value (0.01 in this case).

The Equations of Motion for this system is where the algorithm gets complicated. The EOM function is set up to do the following:

- Calculate the orbital elements from the state information: $[x, y, z, \dot{x}, \dot{y}, \dot{z}, m]$
  (appending m to the state vector allows for easy calculation of state decay)
- Find the optimal thrust angles for the given state to minimize the time derivative of Q
- Determine effectivity by dividing the minimal Q with the optimal Q over the osculating orbit
- If effectivity is below cutoff:
  - Convert the thrust values to the frame of the state vector
  - Provide the time derivative of the state vector as:

    $[\dot{x}, \dot{y}, \dot{z}, -\frac{\mu}{r}x, -\frac{\mu}{r}y, -\frac{\mu}{r}z, 0]$
- If effectivity is above cutoff:
  - Convert the thrust values to the frame of the state vector
  - Provide the time derivative of the state vector as:

    $[\dot{x}, \dot{y}, \dot{z}, -\frac{\mu}{r}x + \frac{f_x}{m}, -\frac{\mu}{r}y + \frac{f_y}{m}, -\frac{\mu}{r}z + \frac{f_z}{m}, -\frac{f}{I_{sp}g_0}]$

IV. Results and Analysis

Then either the algorithm outlined in Section III (for the low-thrust case) or an analytical calculation of m_f and TOF was implemented with the following initial and target conditions (Figure 4):

	Thrust kN	Specific Impulse secs	Initial a (km)	Initial e	Initial mass (kg)	Final a (km)	Final e	Final mass (kg)
Hohmann	914	311	6471	0	1139.2	100000	0	286.75
Bi-Elliptic	914	311	6471	0	1115.4	100000	0	286.75
Spiral	4.72E-04	4190	6471	0.001**	331.45	100000	0.01**	286.88
Combination	4.72E-04	4190	203235.5	0.96816	300*, 845.79	100000	0.01**	286.74

* Value of mass after first burn in Bi-Elliptic manuever

** Values chosen not equal to zero to avoid machine errors

Figure 4. The inital conditions were chosen so that the final masses (dry) would be approximately equal

And the completion of these simulations yielded the following in Figure 5. Since the combination maneuver used two different propellants, the estimated cost of RP-1 ($112/kg)[8] and Xenon ($850/kg)[9] are used as a more effective comparison measure between the maneuvers.

	Hohmann	Bi-Elliptic	Spiral	Combination
Propellant Used (kg)	852.45	828.65	44.57	500, 13.26
Propellant Price Est. $	$95,474.40	$92,808.80	$37,884.50	$67,271.00
Time of Flight (days)	0.7074	12.48	57.35	38.77

Figure 5. Table of results

A. Hohmann

For the Hohmann transfer, we calculated a propellant usage of 852.45 kg, for a total cost of $95474. The time of flight was about 0.7074 days. This matches our expectations and will serve as a good baseline for the remainder of the results.

B. Bi-Elliptic

For the Bi-Elliptic transfer, we calculated a propellant usage of 828.65 kg, for a total cost of $92809 and the time of flight was about 12.5 days. This is, expectedly, a slight gain in efficiency over the Hohmann transfer but a cost in time of flight of almost two weeks. I would expect that for this maneuver, the cost savings may not generally be worth the loss in speed unless TOF is highly devalued. This does match our expectations and will be used as an effective baseline to compare the other two maneuvers with.

C. Spiraling

Our Spiraling transfer had some very significant cost savings. Even factoring in the added price per kilogram of Xenon, the propellant used was only 44.57 kg for a total cost of $37885, a savings of more than $50000 over the Bi-Elliptic transfer. However, the flight time is also much longer at 57 days.

Charts 6-8 give a good sense of how this maneuver was controlled. Upon analysis it becomes clear that the Spiraling trajectory is almost exactly as expected. For the majority of the beginning of the transfer the thrust was primarily in the $\hat{\theta}$ direction, with some oscillation. This allowed the craft to spiral outward until the Q-Law determined that the eccentricity needed to be corrected. The craft then spent a number of revolutions thrusting only when effective (see the mass plot) so that the orbit would circularize efficiently. This matches up with our expectations of what a Spiraling maneuver should look like and is a good sign of confidence for our Proximity Quotient Control Law.

D. Combination Maneuver

The Combined Bi-Elliptic/Spiraling maneuver had a propellant usage of 500 kg RP1 and 13.26 kg Xenon. This ultimately costs approximately $67271, landing the cost for the maneuver in between the Bi-Elliptic and Spiraling maneuvers. With a flight time of 38.77 days, the combination also sits in between those two regimes very nicely in terms of flight time.

Analysis of Figures 9-11 yields some very interesting insights. The Q Law seems to have split the maneuver into 4 sections. Initially, while the craft is traveling immediately after the beginning of the transfer, the control law chooses to actually raise the semi-major axis in order to lower the eccentricity. This seems counter-intuitive, but actually improves the results, since semi-major axis will prove to be easier to control than eccentricity for this particular orbital radius (for more on the principle of "overshoot" as it relates to low-thrust Q Law transfers, see Petropoulos[5]). Afterwards, there is a period of coasting, due to low effectivity. Then, as the craft approached periapsis along this new, higher orbit, it is able to thrust in a direction that allows it to decrease both semi-major axis and eccentricity. From there, the crafts makes a few revolutions, thrusting when efficient as before in the Spiraling approach, in order to circularize the orbit

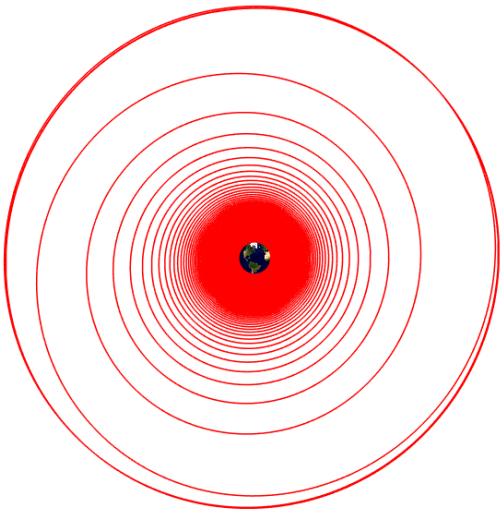

Figure 6. The trajectory of the Spiraling transfer maneuver

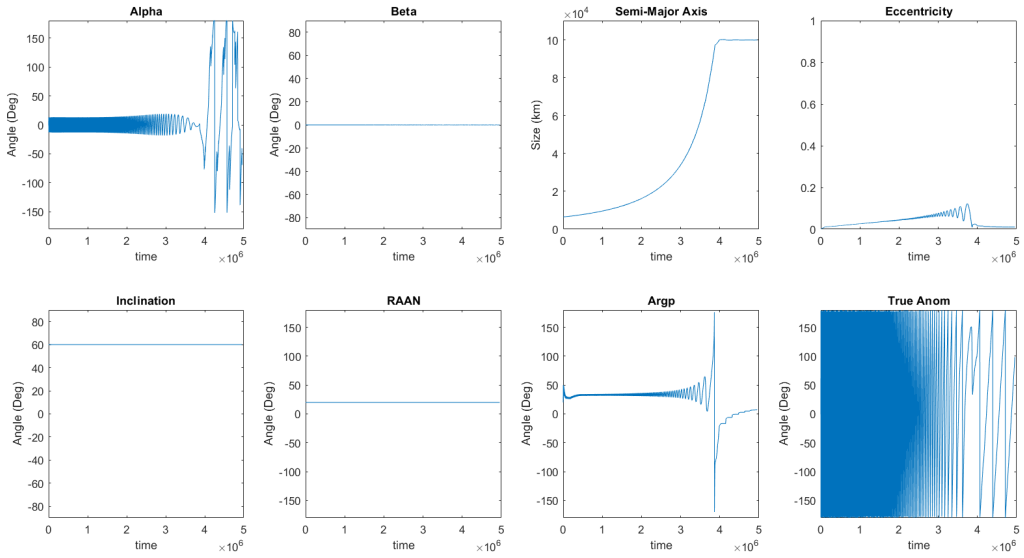

Figure 7. The orbital element plots of the Spiraling transfer maneuver

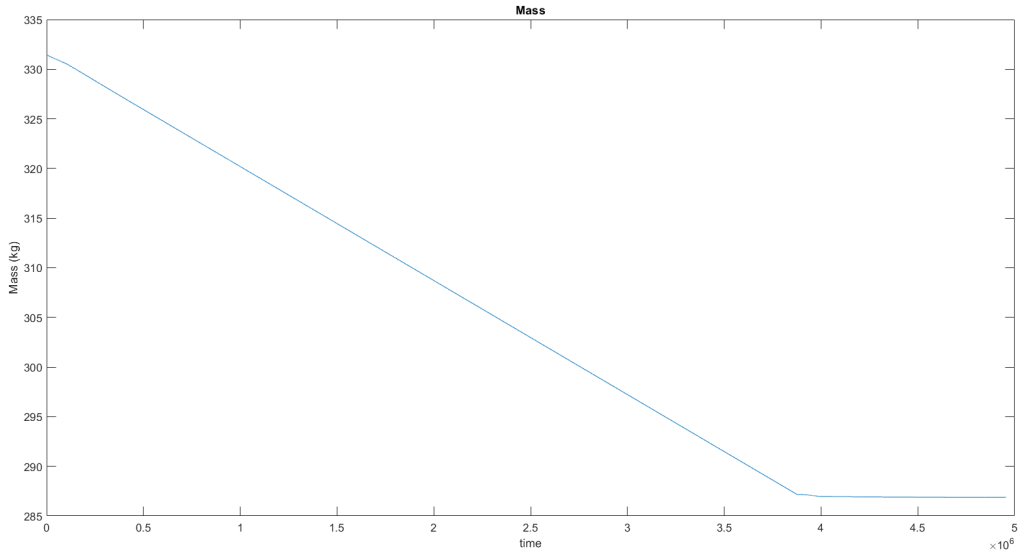

Figure 8. The mass plot of the Spiraling transfer maneuver

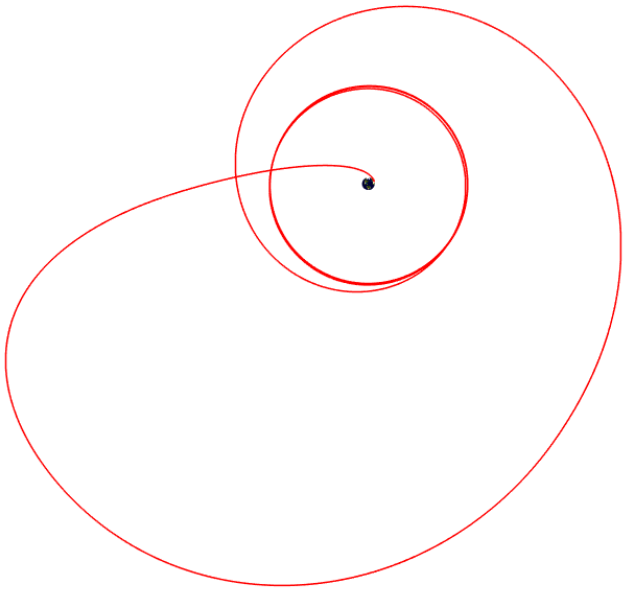

Figure 9. The trajectory of the Combination transfer maneuver

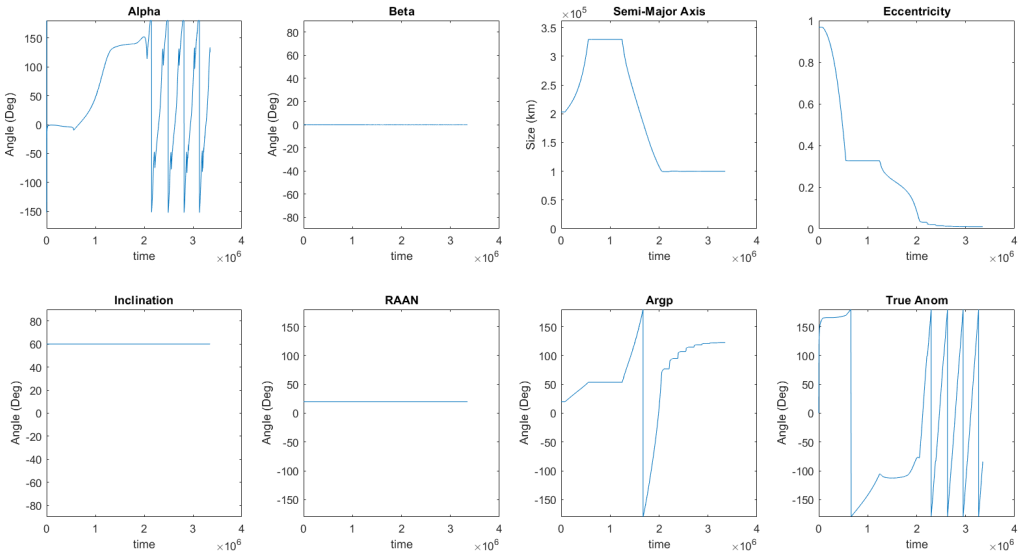

Figure 10. The orbital element plots of the Combination transfer maneuver

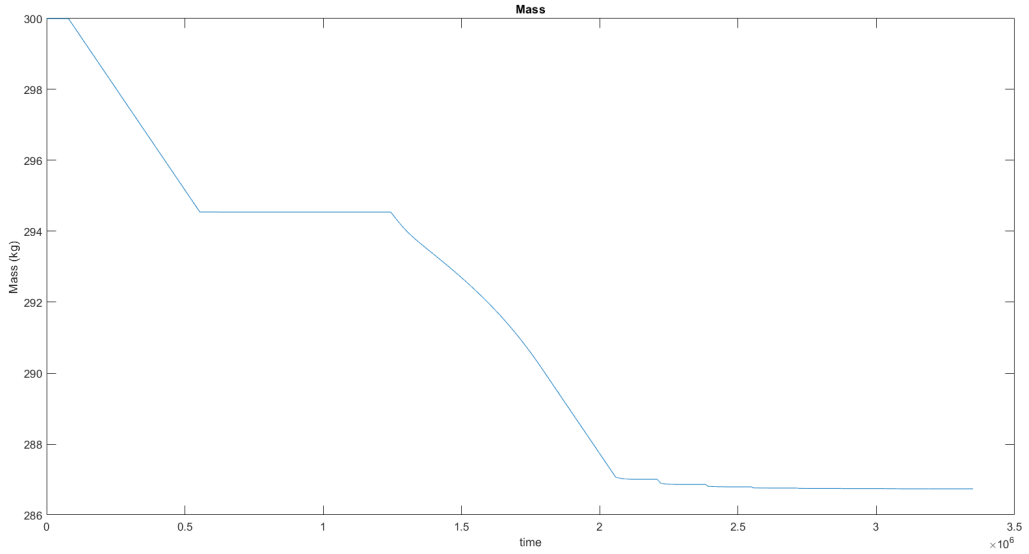

Figure 11. The mass plot of the Combination transfer maneuver

and achieve the final target with more precison. It is worth noting that with a lower effectivity cutoff, one could probably cut this maneuver time almost in half and reduce the maneuver time of the Spiral approach by about 25%, at the expense of some extra fuel.

V. Summary and Conclusions

The results of the various simulations reveal that the combination Bi-Elliptic/Spiraling transfer maneuver does indeed exist in a regime almost directly in between that of the two maneuvers. This could prove to be a very useful tool for mission designers in the case that a high orbit-raising is needed, under the condition that flight time is not the most important characteristic, but traditional low-thrust techniques would still be considered too slow. The technique does involve a bit more complexity in terms of propulsive design, but as discussed earlier, those problems could mostly be mitigated. Combination Bi-Elliptic/Spiraling maneuvers look to be a promising tool in the mission design tool box.

References

[1]Wikimedia Commons, M., "Orbital Hohmann Transfer," 2011, File: Orbital_Hohmann_Transfer.svg.

[2]Aziz, J. D., *Low-Thrust Many-Revolution Trajectory Optimization*, Ph.D. thesis, University of Colorado at Boulder, 2018.

[3]Whiffen, G. J. and Sims, J. A., "Application of the SDC optimal control algorithm to low-thrust escape and capture including fourth body effects," 2002.

[4]Falck, R. D., Sjauw, W. K., and Smith, D. A., "Comparison of Low-Thrust Control Laws for Applications in Planetocentric Space," *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 2014, p. 3714.

[5]Petropoulos, A. E., "Simple control laws for low-thrust orbit transfers," 2003.

[6]Petropoulos, A. E., "Refinements to the Q-law for the low-thrust orbit transfers," 2005.

[7]"fminsearchbnd," .

[8]Mohr, D., "What is the cost of RP-1 rocket-grade kerosene?" Oct 2018.

[9]Kieckhafer, A. and King, L. B., "Energetics of propellant options for high-power Hall thrusters," *Journal of propulsion and power*, Vol. 23, No. 1, 2007, pp. 21–26.

Appendix

```
format shortG

%Spacecraft Info
f = 2*0.236e-3 ; %kN
Isp = 4190 ; %secs
m = 331.45 ; %kg

%Constants
mu = 3.986004415e5 ; %km^3/s^2

%Target Orbital Elements (Ignore for Now)
oe_t = [ 100000, 0.01, 60*pi/180, 20*pi/180, 20*pi/180 ] ;
w = [ 1, 1, 0, 0, 0 ] ;

%Initial Conditions
oe = [ 6471, 0.001, 60*pi/180, 20*pi/180, 20*pi/180, 0 , 0] ;
x0 = [dcm(oe_to_rth(oe,mu),oe); m] ;
Q_rec = [] ;
alpha_rec = [] ;
beta_rec = [] ;
oe_rec = [] ;

%Solver
options = odeset('Stats','off',...
                 'RelTol',1e-10,...
                 'AbsTol',[1e-3,1e-3,1e-3,1e-6,1e-6,1e-6,1e-4],...
                 'Events',@(t,x)stopEvent(t,x,w,oe_t,f,mu)) ;
[t, x, te, xe, ie] = ode45(@(t,x) EOM(t,x,mu,w,oe_t,f,Isp), ...
                           [0 50*365*24*3600], x0, options) ;

%Plotting
figure('Color', 'k');
plot3(x(:,1),x(:,2),x(:,3), 'color','r','LineWidth',0.5) ;
hold on;
set(gca, 'NextPlot','add', 'Visible','off');
axis equal;
axis auto;
[xe, ye, ze] = ellipsoid(0, 0, 0, 6371, 6371, 6371, 180);
globe = surf(xe, ye, -ze, 'FaceColor', 'none', 'EdgeColor', 0.5*[1 1 1]);
if ~isempty([])
    hgx = hgtransform;
    set(hgx,'Matrix', makehgtform('zrotate',[]));
    set(globe,'Parent',hgx);
end
cdata = imread('http://upload.wikimedia.↙
org/wikipedia/commons/thumb/c/cd/Land_ocean_ice_2048.jpg/1024px-Land_ocean_ice_2048.↙
jpg');
set(globe, 'FaceColor', 'texturemap', 'CData', cdata,...
    'FaceAlpha', 1, 'EdgeColor', 'none');
```

```
hold off
axis vis3d

function [x_dot] = EOM(t,x,mu,w,oe_t,f,Isp)
% Equations of Motion for lab
grav = -mu/norm(x(1:3))^3 ;
oe = get_oe(x,mu) ;
angs = [0,0] ;
[angs,Q_dot] = fminsearchbnd(@(angs)dQdt(angs,oe,w,oe_t,f,mu),angs,[-pi,-pi/2],[pi, ↙
pi/2]) ;
alpha = angs(1) ;
beta = angs(2) ;
Q_val = Q(oe,w,oe_t,f,mu) ;
eff = Q_dot/dQdt_xx(oe,w,oe_t,f,mu) ;

statement = 'Q = ' + string(Q_val) + ', t = ' + string(t) + ', a = '...
    + string(oe(1)) + ', e = ' + string(oe(2)) + ', ta = ' + string(oe(6)*180/pi) +↙
'\n' ;
fprintf(statement)

if eff > 0.5
    dv = dcm([f*cos(beta)*sin(alpha),f*cos(beta)*cos(alpha),f*sin(beta)],oe) ;
    x_dot = [ x(4); x(5); x(6); grav*x(1) + dv(1)/x(7); grav*x(2) + dv(2)/x(7); ↙
grav*x(3) + dv(3)/x(7); -(f*1000)/(Isp*9.81)] ;
else
    dv = [0,0,0] ;
    x_dot = [ x(4); x(5); x(6); grav*x(1) + dv(1)/x(7); grav*x(2) + dv(2)/x(7); ↙
grav*x(3) + dv(3)/x(7); 0] ;
end

end

function [value,isterminal,direction] = stopEvent(t,x,w,oe_t,f,mu)
oe = get_oe(x,mu) ;
value = Q(oe,w,oe_t,f,mu) - 0.05 ;
isterminal = 1 ;
direction = 0 ;
end
```

```
function [Q_val] = Q(oe,weight,oe_t,f,mu)
%Determines the value of Q for a given set of orbital parameters

a = oe(1) ;
e = oe(2) ;
i = oe(3) ;
O = oe(4) ;
w = oe(5) ;
ta = oe(6) ;

a_t = oe_t(1) ;
e_t = oe_t(2) ;
i_t = oe_t(3) ;
O_t = oe_t(4) ;
w_t = oe_t(5) ;

p = a*(1-e^2) ;
h = sqrt(mu*p) ;
r = p/(1+e*cos(ta)) ;

a_d = 2*f*sqrt((a^3*(1+e))/(mu*(1-e))) ;
a_w = (1+((a-a_t)/(3*a_t))^4)^(1/2) ;
a_c = weight(1)*a_w*((a-a_t)/a_d)^2 ;

e_d = 2*p*f/h ;
e_c = weight(2)*((e-e_t)/e_d)^2 ;

i_d = (p*f)/(h*(sqrt(1-e^2*sin(w)^2)-e*abs(cos(w)))) ;
i_c = weight(3)*((i-i_t)/i_d)^2 ;

%ignoring RAAN and ARGP for now

peri_weight = 1 + 1*exp(1*(1-(a*(1-e)/6478.1363))) ;

Q_val = peri_weight*(a_c + e_c + i_c) ;

end
```

```
function [dQdt] = dQdt(angs,oe,weights,oe_t,f,mu)
%UComputes the time derivative of Q for a given orbit, and thrust angles
alpha = angs(1) ;
beta = angs(2) ;

a = oe(1) ;
e = oe(2) ;
i = oe(3) ;
O = oe(4) ;
w = oe(5) ;
ta = oe(6) ;

p = a*(1-e^2) ;
h = sqrt(mu*p) ;
r = p/(1+e*cos(ta)) ;

f_r = f*cos(beta)*sin(alpha) ;
f_t = f*cos(beta)*cos(alpha) ;
f_h = f*sin(beta) ;

delt = 1e-10 ;

Q_val = Q([a,e,i,O,w,ta],weights,oe_t,f,mu) ;

dadt = (2*a^2/h)*(e*sin(ta)*f_r + (p/r)*f_t) ;
dQda = (Q([a*(1+delt),e,i,O,w,ta],weights,oe_t,f,mu) - Q_val)/(a*delt) ;

dedt = (1/h)*(p*sin(ta)*f_r + ((p+r)*cos(ta) + r*e)*f_t) ;
dQde = (Q([a,e*(1+delt),i,O,w,ta],weights,oe_t,f,mu) - Q_val)/(e*delt) ;

didt = r*cos(ta+w)*f_h/h ;
dQdi = (Q([a,e,i*(1+delt),O,w,ta],weights,oe_t,f,mu) - Q_val)/(i*delt) ;

dOdt = r*sin(ta + w)*f_h/(h*sin(i)) ;
dQdO = (Q([a,e,i,O*(1+delt),w,ta],weights,oe_t,f,mu) - Q_val)/(O*delt) ;

dwdt = (1/(e*h))*(-p*cos(ta)*f_r + (p+r)*sin(ta)*f_t) - r*sin(ta+w)*cos(i)*f_h/(h*sin ↙
(i)) ;
dQdw = (Q([a,e,i,O,w*(1+delt),ta],weights,oe_t,f,mu) - Q_val)/(w*delt) ;

dQdt = dadt*dQda + dedt*dQde + didt*dQdi + dOdt*dQdO + dwdt*dQdw ;

end
```

```
function [dQdt] = dQdt_xx(oe,weights,oe_t,f,mu)
%Determines the optimum "minimum dQdt" over the values of true anom

a = oe(1) ;
e = oe(2) ;
i = oe(3) ;
O = oe(4) ;
w = oe(5) ;
ta = oe(6) ;

a_t = oe_t(1) ;
e_t = oe_t(2) ;

p = a*(1-e^2) ;
h = sqrt(mu*p) ;
r = p/(1+e*cos(ta)) ;

delt = 1e-10 ;

dadt = 2*f*sqrt((a^3*(1+e))/mu*(1-e)) ;
if a > a_t
    dadt = -dadt ;
end
dQda = (Q([a*(1+delt),e,i,O,w,0],weights,oe_t,f,mu) - Q([a,e,i,O,w,0],weights,oe_t,f,
mu))/(a*delt) ;

dedt = 2*p*f/h ;
if e > e_t
    dedt = -dedt ;
end
dQde = (Q([a,e*(1+delt),i,O,w,pi],weights,oe_t,f,mu) - Q([a,e,i,O,w,pi],weights,oe_t,
f,mu))/(e*delt) ;

didt = p*f/(h*(sqrt(1-e^2*sin(w)^2) - e*abs(cos(w)))) ;
%ignoring ta for best i determination for now...
dQdi = (Q([a,e,i*(1+delt),O,w,ta],weights,oe_t,f,mu) - Q([a,e,i,O,w,ta],weights,oe_t,
f,mu))/(i*delt) ;

dOdt = p*f/(h*sin(i)*(sqrt(1-e^2*cos(w)^2)-e*abs(sin(w)))) ;
%also ignoring ta for best O determination for now...
dQdO = (Q([a,e,i,O*(1+delt),w,ta],weights,oe_t,f,mu) - Q([a,e,i,O,w,ta],weights,oe_t,
f,mu))/(O*delt) ;

cos_theta_xx = ((1-e^2)/(2*e^3)+sqrt((1/4)*((1-e^2)/e^3)^2+1/27))^(1/3) ...
    - (-(1-e^2)/(2*e^3)+sqrt((1/4)*((1-e^2)/e^3)^2+1/27))^(1/3) - 1/e ;
theta_xx = acos(cos_theta_xx) ;
r_xx = p/(1+e*cos(theta_xx)) ;
w_d_xxi = (f/(e*h))*sqrt(p^2*cos(theta_xx)^2+(p+r_xx)^2*sin(theta_xx)^2) ;
w_d_xxo = dOdt*abs(cos(i)) ;
```

```
dwdt = (w_d_xxi+0.01*w_d_xxo)/(1+0.01) ;
dQdw = (Q([a,e,i,O,w*(1+delt),theta_xx],weights,oe_t,f,mu) - Q([a,e,i,O,w,theta_xx], ↙
weights,oe_t,f,mu))/(w*delt) ;

dQdt = dadt*dQda + dedt*dQde + didt*dQdi + dOdt*dQdO + dwdt*dQdw ;

end
```

```
oe = [] ;
for i=1:size(x,1)
    if i==1
        waitbar(i/size(x,1), 'oe') ;
    end
    oe = [oe ; get_oe(x(i,:),mu), t(i) ] ;
    waitbar(i/size(x,1))
end

angs = [] ;
for i=1:size(x,1)
    if i==1
        waitbar(i/size(x,1), 'angs') ;
    end
    angs = [angs ; fminsearchbnd(@(angles)dQdt(angles,oe(i,:),w,oe_t,f,mu),[0,0],[-
pi,-pi/2],[pi,pi/2]), t(i) ] ;
    waitbar(i/size(x,1))
end

%Plotting
figure('Color', 'k');
plot3(x(:,1),x(:,2),x(:,3), 'color','r','LineWidth',1) ;
hold on;
set(gca, 'NextPlot','add', 'Visible','off');
axis equal;
axis auto;
[xe, ye, ze] = ellipsoid(0, 0, 0, 6371, 6371, 6371, 180);
globe = surf(xe, ye, -ze, 'FaceColor', 'none', 'EdgeColor', 0.5*[1 1 1]);
if ~isempty([])
    hgx = hgtransform;
    set(hgx,'Matrix', makehgtform('zrotate',[]));
    set(globe,'Parent',hgx);
end
cdata = imread('http://upload.wikimedia.
org/wikipedia/commons/thumb/c/cd/Land_ocean_ice_2048.jpg/1024px-Land_ocean_ice_2048.
jpg');
set(globe, 'FaceColor', 'texturemap', 'CData', cdata, 'FaceAlpha', 1, 'EdgeColor',
'none');
hold off
axis vis3d

figure() ;
subplot(2,4,1) ;
plot(angs(:,3),angs(:,1)*180/pi) ;
title('Alpha') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-180,180]) ;
subplot(2,4,2) ;
```

```
plot(angs(:,3),angs(:,2)*180/pi) ;
title('Beta') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-90,90]) ;
subplot(2,4,3) ;
plot(oe(:,7),oe(:,1)) ;
title('Semi-Major Axis') ;
ylabel('Size (km)') ;
xlabel('time') ;
ylim([0,max(oe(:,1))*1.1]) ;
subplot(2,4,4) ;
plot(oe(:,7),oe(:,2)) ;
title('Eccentricity') ;
xlabel('time') ;
ylim([0,1]) ;
subplot(2,4,5) ;
plot(oe(:,7),oe(:,3)*180/pi) ;
title('Inclination') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-90,90]) ;
subplot(2,4,6) ;
plot(oe(:,7),oe(:,4)*180/pi) ;
title('RAAN') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-180,180]) ;
subplot(2,4,7) ;
plot(oe(:,7),oe(:,5)*180/pi) ;
title('Argp') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-180,180]) ;
subplot(2,4,8) ;
plot(oe(:,7),oe(:,6)*180/pi) ;
title('True Anom') ;
ylabel('Angle (Deg)') ;
xlabel('time') ;
ylim([-180,180]) ;

figure()
plot(t,x(:,7))
title('Mass')
ylabel('Mass (kg)')
xlabel('time')

% a_con = linspace(6000,150000,100) ;
% e_con = linspace(0,0.1,100) ;
```

```
% Q_con = [] ;
% for i=1:length(a_con)
%     for j=1:length(e_con)
%         Q_con(i,j) = Q([a_con(i),e_con(j),2,2,2,2],w,oe_t,f,mu) ;
%     end
% end
% figure() ;
% contour(a_con,e_con,Q_con) ;
% plot(oe_rec(:,1),oe_rec(:,2)) ;

disp(x(size(x,1),size(x,2)))
disp(min(oe_rec(:,2)))
```

```
%Calculating the impulsive Hohmann and Bi-Elliptic Thrusts

%Merlin 1D
f = 914 ; %kN
Isp = 311 ; %secs

%Planetary
mu = 3.986004415e5 ; %km^3/s^2

%Initial Conditions
a1 = 6471 ;
oe1 = [ a1, 0.001, 60*pi/180, 20*pi/180, 20*pi/180, 0 ] ;
state1 = dcm(oe_to_rth(oe1,mu),oe1) ;
a_t = 100000 ;
rat = 4 ;
m_f = 286.75 ; %kg
% m_i = 845.79 ; %kg

%Hohmann
a2_h = (6471+a_t)/2 ;
oe2_h = [ a2_h, 1-a1/a2_h, 60*pi/180, 20*pi/180, 20*pi/180, 0 ] ;
state2_h = dcm(oe_to_rth(oe2_h,mu),oe2_h) ;
dv1_h = sqrt(2*mu/a1 - mu/a2_h) - sqrt(mu/a1) ;

oe3_h = [ a_t, 0.001, 60*pi/180, 20*pi/180, 20*pi/180, pi ] ;
state3_h = dcm(oe_to_rth(oe3_h,mu),oe3_h) ;
dv2_h = sqrt(mu/a_t) - sqrt(2*mu/(a_t) - mu/a2_h) ;

dv_h = abs(dv1_h) + abs(dv2_h) ;
m_i_h = m_f*exp((dv_h*1000)/(Isp*9.81))
tof_h = pi*sqrt(a2_h^3/mu)/(24*3600) ; %days


%Bi-elliptic
a2_b = (a1+rat*a_t)/2 ;
oe2_b = [ a2_b, 1-a1/a2_b, 60*pi/180, 20*pi/180, 20*pi/180, 0 ] ;
state2_b = dcm(oe_to_rth(oe2_b,mu),oe2_b) ;
dv1_b = sqrt(2*mu/a1 - mu/a2_b) - sqrt(mu/a1) ;

a3_b = (a_t+rat*a_t)/2 ;
oe3_b = [ a3_b, 1-a_t/a3_b, 60*pi/180, 20*pi/180, 20*pi/180, pi ] ;
state3_b = dcm(oe_to_rth(oe3_b,mu),oe3_b) ;
dv2_b = sqrt(2*mu/(rat*a_t) - mu/a3_b) - sqrt(2*mu/(rat*a_t) - mu/a2_b) ;

oe4_b = [ a_t, 0.001, 60*pi/180, 20*pi/180, 20*pi/180, 0 ] ;
state4_b = dcm(oe_to_rth(oe4_b,mu),oe4_b) ;
dv3_b = sqrt(mu/a_t) - sqrt(2*mu/a_t - mu/a3_b) ;

dv_b = abs(dv1_b) + abs(dv2_b) + abs(dv3_b) ;
```

```
m_i_b = m_f*exp((dv_b*1000)/(Isp*9.81))
tof_b = pi*(sqrt(a2_b^3/mu)+sqrt(a3_b^3/mu))/(24*3600) ; %days
```

```
function [xyz_vec] = dcm(state,oe)
%Converts from (R,Theta,H) frame to (X,Y,Z) frame

raan = oe(4) ;
th = oe(5)+oe(6) ;
i = oe(3) ;

cr = cos(raan) ;
sr = sin(raan) ;
ct = cos(th) ;
st = sin(th) ;
ci = cos(i) ;
si = sin(i) ;
mat = [  cr*ct-sr*ci*st,    -cr*st-sr*ci*ct,    sr*si   ;
         sr*ct+cr*ci*st,    -sr*st+cr*ci*ct,    -cr*si  ;
         si*st,             si*ct,              ci      ] ;

if max(size(state)) == 6
    mat = kron(eye(2),mat) ;
end

if size(state,1) == 1
    state = state' ;
end

xyz_vec = mat*state ;

end
```

```
function [oe] = get_oe(state,mu)
%This function gets the orbital elements from cartesian coordinates
r_xyz = state(1:3) ;
v_xyz = state(4:6) ;
r = norm(r_xyz) ;
v = norm(v_xyz) ;

h_xyz = cross(r_xyz,v_xyz) ;
h = norm(h_xyz) ;
E = 0.5*v^2 - mu/r ;

a = -mu/(2*E) ;

e = sqrt(1+(2*h^2*E)/mu^2) ;
e_xyz = cross(v_xyz,h_xyz)/mu - r_xyz/r ;

i = acos(h_xyz(3)/h) ;

n_xyz = cross([0,0,1],h_xyz) ;
n = norm(n_xyz) ;
if dot(n_xyz,[0,1,0]) > 0
    O = real(acos(dot(n_xyz,[1,0,0])/n)) ;
else
    O = -real(acos(dot(n_xyz,[1,0,0])/n)) ;
end

if dot(e_xyz,[0,0,1]) > 0
    w = real(acos(dot(n_xyz,e_xyz)/(n*e))) ;
else
    w = -real(acos(dot(n_xyz,e_xyz)/(n*e))) ;
end

if dot(r_xyz,v_xyz) > 0
    ta = real(acos(dot(r_xyz,e_xyz)/(r*e))) ;
else
    ta = -real(acos(dot(r_xyz,e_xyz)/(r*e))) ;
end

oe = [a,e,i,O,w,ta] ;

end
```

```
function [rth] = oe_to_rth(oe,mu)
% Takes orbital elements and spits out the position and velocity in
% (R,Theta, H)
a = oe(1) ;
e = oe(2) ;
ta = oe(6) ;

p = a*(1-e^2) ;
h = sqrt(mu*p) ;
r = p/(1+e*cos(ta)) ;

rth = [ r; 0; 0; (mu/h)*e*sin(ta); (mu/h)*(1+e*cos(ta)); 0] ;
end
```