

High-Fidelity Low-Thrust Trajectory Determination Research and Analysis

Senior Project

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Aerospace Engineering

by

Tyler Hill

June, 2012

High-Fidelity Low-Thrust Trajectory Determination Research and Analysis

Tyler Hill¹

California Polytechnic State University San Luis Obispo, 1 Grand Avenue, San Luis Obispo, CA, 93405

This document discusses a numerical analysis method for low thrust trajectory propagation known as the proximity quotient or Q-Law. The process uses a Lyapunov feedback control law developed by Petropoulos^[1] to propagate trajectories of spacecraft by minimizing the user defined function at the target orbit. A simplified propagator is created from the core mechanics of this method in MATLAB and tested in several user defined cases to demonstrate its capabilities. Several anomalies arose in test cases where variations in eccentricity, inclination, right ascension of the ascending node, and argument of perigee were specified. Solutions to these anomalies are discussed and include development of a coasting mechanic and a new method for thruster angle selection.

Nomenclature

α	= azimuthal angle (rad)
β	= elevation angle (rad)
θ	= true anomaly (rad)
ω	= argument of perigee (rad)
$\dot{\omega}$	= rate of change of argument of perigee (rad/s)
Ω	= right ascension of the ascending node (rad)
$\dot{\Omega}$	= rate of change of right ascension of the ascending node (rad/s)
a	= semi-major axis (km)
\dot{a}	= rate of change of semi-major axis (km/s)
e	= eccentricity
\dot{e}	= rate of change of eccentricity (sec ⁻¹)
inc	= inclination (rad)
\dot{inc}	= rate of change of inclination (rad/s)
Q	= proximity quotient
\dot{Q}	= time rate of change of proximity quotient
W	= weight
oe	= orbital element
\dot{oe}	= time rate of change of orbital element

Subscripts

h	= direction of angular momentum
i	= current time step
$i+1$	= next time step
p	= periapse
$pmin$	= minimum periapse
r	= radial direction
T	= target
θ	= circumferential direction

I. Introduction

Electric propulsion is any propulsion system which uses electricity to assist or accelerate propellant out of the vehicle to generate thrust. These engines produce very little thrust, require large amounts of power, and are relatively heavy and complex. However, the high specific impulse (Isp) is the largest draw for this system as very high ΔV requirements can be met using low fuel mass. The concept of electric propulsion was first recognized by Robert Goddard as early as 1906. Since then, the popularity of this method of propulsion has skyrocketed. As of late 1990 the number of spacecraft using electric propulsion reached the triple-digit mark, so the importance of understanding the governing principles and methods behind low thrust trajectory propagation cannot be underestimated. The contingency with using a low thrust engine is the need to run it for large periods of time, usually for a majority of the transfer orbit, meaning the mass, thruster angle, and (in reality) the transfer orbit itself are constantly changing. Even without including the various perturbations to the orbit, and assuming a standard inverse square gravity field, low thrust transfers are challenging to design due to the number of revolutions around a central body and determination of thruster directions and arc locations. It is a case of a large amount of variables that do not converge nicely with the discrete techniques used for chemical propulsion. To handle these types of scenarios several analytic optimization methods have been developed to determine the optimal orbit, as defined by the mission designer, either directly or indirectly. An unintended consequence of this effort is the vast number of methods now in existence. To add to the already complex problem of knowing which method to use, the implementation of these methods is far from friendly. To get around these complications, some attention has been focused on methods employing heuristic control laws. The advantage of these types of methods is the speed at which it can generate solutions; the drawback is the trajectory will be non-optimal. It is for this reason most numeric optimization programs will use a heuristic solution as a first guess. An example of this is the program MYSTIC which was used to determine the orbits for NASA's Jupiter Icy Moons Orbiter (JIMO) and DAWN^[2].

The category of heuristics discussed in this paper uses Lyapunov feedback control, where a suitable Lyapunov function is defined by the mission designer and minimized at the desired state. The specific function discussed and replicated for this project is called the proximity quotient or Q-Law and was created by Anastassois Petropoulos^[1]. The Q-law is still termed a "candidate" Lyapunov function because it has not been rigorously proved in spite of convergence being observed in all transfers performed thus far^[3]. The function was created based on analytic expressions for maximum rates of change and desired rates of change of each element. The term proximity quotient is coined because these values may be thought of as a measure of the proximity to the target orbit. The propagation itself is carried out by multiplying the time rate of change of the element by the step size and adding it to the previous value for that element, a method similar to variation of parameters. An overview of the algorithm employed in the Q-Law is provided in the next section.

II. The Q-Law Algorithm

This Q-Law function is designed to optimize a trajectory based on fuel economy by maximizing the rate of change of five orbital elements given an initial and final orbit. These five elements are semi-major axis (a), eccentricity (e), inclination (i), argument of perigee (ω), and right ascension of the ascending node (Ω). Since the Q-Law is designed for trajectories between Keplerian orbits and not point-to-point transfers the true anomaly of the osculating transfer orbit is ignored when optimizing the trajectory. The proximity quotient value is calculated using

$$Q = (1 + W_p P) \sum_1^5 W_{oe} S_{oe} \left[\frac{d(oe - oe_T)}{\dot{oe}_{xx}} \right]^2 \quad (1)$$

where oe designates a current orbital parameter term, oe_T designates the target orbital parameter, W_p and W_{oe} are weights assigned to the function and are determined using additional optimizing techniques, S_{oe} is a scalar function,

Aerospace Engineering, California Polytechnic University, Undergraduate Student¹

and δe_{xx} is the maximum of the parameter over true anomaly of the maximum over thrust angle. This value is then stored to monitor the progress of the function as it attempts to reach the target orbit. Additionally, this value can be used in a coasting mechanic determining the effectiveness of thrusting at this moment in time on the osculating orbit but this is ignored since this version of the function involves constant thrust. If these inequalities are strictly followed a situation called thrust jitters, or rapid start/stopping of the thruster, would occur. To mitigate this effect a minimum radius periapsis radius is implemented in the form of

$$P = \exp \left[k \left(1 - \frac{r_p}{r_{pmin}} \right) \right] \quad (2)$$

where k is a scalar, r_p is the osculating periapsis radius in km, and r_{pmin} is the lowest permissible value of r_p which is found using additional optimizing techniques. To ensure convergence to the target orbit the previously mentioned scaling function is used and takes the form

$$S_{oe} = \begin{cases} \left[1 + \left| \frac{a - a_T}{ma_T} \right|^{n/r} \right] & \text{for } oe = a \\ 1 & \text{for } oe = e, i, \omega, \Omega \end{cases} \quad (3)$$

where m, n, and r are scalars, a is the semi-major axis in km, and the subscript T designates the parameter of the target orbit. Additionally, the distance function shown in the proximity quotient equation is found using

$$d(oe - oe_T) = \begin{cases} oe - oe_T & \text{for } oe = a, e, i \\ \cos^{-1}[\cos(oe - oe_T)] & \text{for } oe = \omega, \Omega \end{cases} \quad (4)$$

where oe designates the respective orbit parameter for the current and target orbits. The reason for the specific form of the distance function for ω and Ω is because it provides an angular measurement for the distance between the two points of the current trajectory using the “short way around” the circle since the sign of the derivative will indicate whether it leads or lags the target. Next the value of the time rate of change of the proximity quotient is determined using

$$\dot{Q} = \sum_{i=1}^5 \frac{\partial Q}{\partial oe_i} \delta oe_i \quad (5)$$

where δoe_i are the Gaussian rates of change of each of the orbital elements. These Gaussian rates are determined using

$$\dot{\Omega} = \frac{r \sin(\theta + \omega)}{h \sin i} f_h \quad (7)$$

$$\dot{inc} = \frac{r \cos(\theta + \omega)}{h} f_h \quad (8)$$

$$\dot{\omega} = \frac{1}{eh} [-p \cos \theta f_r + (p + r) \sin \theta f_\theta] - \frac{r \sin(\theta + \omega) \cos i}{h \sin i} f_h \quad (9)$$

$$\dot{a} = \frac{2a^2}{h} \left(e \sin \theta f_r + \frac{p}{r} f_\theta \right) \quad (10)$$

$$\dot{e} = \frac{1}{h} \{ p \sin \theta f_r + [(p+r) \cos \theta + re] f_\theta \} \quad (11)$$

where h is the specific angular momentum in m^2/s , f is the acceleration due to thrust in m/s^2 , p is the semilatus rectum in m , r is the radial distance from the central body, θ is the true anomaly in radians, and a , e , i , ω , and Ω are the parameters mentioned previously. The f values are the thrust accelerations with respect to each of the thrust angles and can be found using

$$f_r = f \cos \beta \sin \alpha \quad (12)$$

$$f_\theta = f \cos \beta \cos \alpha \quad (13)$$

$$f_h = f \sin \beta \quad (14)$$

where f is the thruster acceleration, α is the azimuthal thrust angle, and β is the elevation thrust angle with respect to the osculating orbit's angular momentum. At this point we notice we have not found values for the thrust angles meaning, the current value for the time derivative is a function of these angles. To determine the values of these angles at this point in the orbit the gradient of the time derivative is taken with respect to α and β . The two gradient equations are then set equal to zero and solved. Since this approach involves solving trigonometric functions, multiple solutions will be produced. To determine the desired set of angles, each pair is substituted back into the equation for the time derivative of the proximity quotient and each value is stored in an array. Once all pairs of angles have been entered, the desired pair is located by finding the one that yields the most negative value of \dot{Q} . This is because the purpose of the function is to drive the value of the proximity quotient to zero as fast as possible. After the angles have been determined, they are substituted into the Eq. 12-14 and the rate of change of the elements for this moment in time is determined using Eqn. 7-11. These values are then used with the orbital elements from this current time step to propagate the orbit forward to the next using

$$oe_{i+1} = oe_i + \dot{oe}_i t_{step} \quad (15)$$

where oe_i is the element at the current time step, oe_{i+1} is the calculated value at the next time step, \dot{oe}_i is the element's rate of change, and t_{step} is the time step in seconds. These new orbital elements are then stored under a new variable and used to determine the spacecraft's new state vector. The array of current elements is then used as the new initial state and the algorithm starts again. The Q-Law function will repeat this process until either the osculating elements are within a specified tolerance of the target values or the stop time condition is reached. Normally, only the first termination condition is used however to ensure the function does not get stuck in an infinite loop a limitation is placed on the number of steps it can iterate through. However the addition of this safety measure means the final values of the orbital elements has to be analyzed to determine which termination condition halted the iteration. If the function is stopped because the time constraint is met then the maximum runtime must be extended until the spacecraft is allowed to reach its target orbit.

III. Analysis

To test and troubleshoot the function only one of the five orbital elements was varied at any one time, thus the five test cases shown in Table 1. For each test the final values of each of the elements, the thrust angles, proximity value, trip time, and trajectory appearance will be used to assess the performance of the function. The *Aerospace Engineering, California Polytechnic University, Undergraduate Student*¹

decision to omit the coasting mechanic in this replication of the proximity quotient is kept in mind when analyzing each of the aforementioned parameters to determine if the function is performing as expected. Additionally the values for the spacecraft mass and thrust output are selected to be 1000 kg and 5 mN respectively. These values may or may not reflect actual spacecraft using EP which may contribute to any discrepancy or anomaly observed in the final trajectory. The results of each of the test cases are discussed in the following section.

target element values for
] where the angles are

Target Elements
[20000, 0.1, 0.1, 6, 5]
[20000, 0.8, 0.1, 6, 5]
[20000, 0.1, 0.3, 6, 5]
[20000, 0.1, 0.1, 0.4, 5]
[20000, 0.1, 0.1, 0.1, 0.3]

IV. Results & Discussion

A. Test Case 1

The first test case involves increasing the semi-major axis from 20,000 km to 35,000 km while the other elements are kept constant. As seen in Fig. 1 the trajectory is fairly indicative of a spacecraft using electric propulsion, consisting of a fairly tight spiral with the radial spacing between each pass at apoapse slightly larger than at periapse. This indicates the function appears to be obeying Keplerian principles as max force should be applied in the velocity direction at periapse where the ΔV is smaller. The trajectory has an associated flight time of approximately 60.5 hours which seems appropriate for the 12,800 km increase in altitude that is occurring. Comparing the end values of the elements to the target element values it can be seen in Table 2 that the greatest difference is from the eccentricity at about 29%. The slight increase in eccentricity could be due to the function attempting to optimize for fuel efficiency without the coasting mechanic. This is because it is more fuel efficient to raise the semi-major axis through increasing eccentricity causing the apoapse to rise to the appropriate value or even slightly over. Once this step is complete the orbit will then be circularized until it matches the target orbit within an acceptable tolerance^[1]. If the coasting mechanic was implemented the propagated trajectory may show this procedure if the required increase in semi-major axis demanded it. At this point in time however, the function appears to be working close to ideal for this orbit transfer.

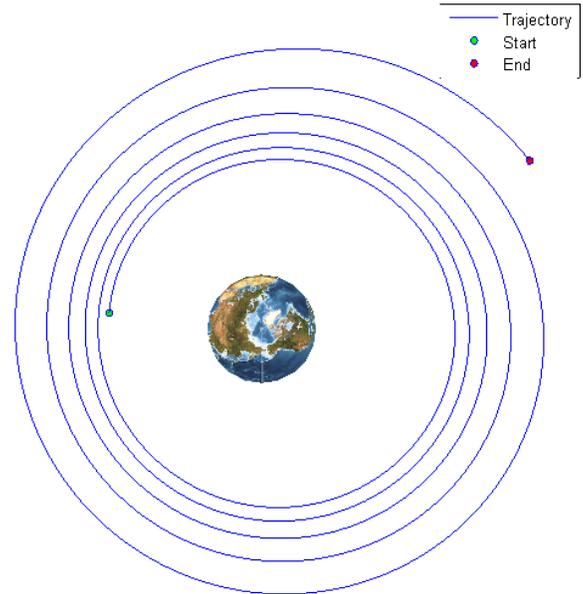


Figure 1. Transfer trajectory for a semi-major axis raise from 20,000 km to 35,000 km.

Table 2. Error values between the functions ending values and target orbit values for each of the five orbital elements.

Element	[a, e, i, ω , Ω]
Error (%)	[1, 28.7, 4.84, 4.65, 0.03]

B. Test Case 2

The second test case involves increasing the eccentricity of the spacecraft's initial orbit as dictated by the values in Table 1. The resulting trajectory, shown in Appendix A, sees a large increase the orbital radius and inclination while moving toward the desired change in eccentricity. To rule out the possibility of not allowing enough time for the complete trajectory to occur, the run time is increased to 20 days and the termination condition *Aerospace Engineering, California Polytechnic University, Undergraduate Student*¹

for the element is removed. The resulting trajectory did not behave as predicted, rather it went through the same inclination change when the eccentricity value had reach its target and stayed there for the duration of the propagation. Further troubleshooting was attempted by adjusting the weights in the proximity quotient to attempt to compensate for this yet the results do not show any significant improvement. The starting and ending values of the semi-major axis are adjusted to determine if this is causing some anomaly in the propagation. However the trajectory this adjustment produces a trajectory with a larger proximity value indicating it is farther from the desired orbit. This indicates there is an issue with the propagator itself and the method which it chooses the direction to apply thrust in. This is intriguing because the previous test case appeared to function close to nominal using the same angle selection method. The plots of the thrust angles over time are shown in Appendix A however the values from Case II do not offer much insight into the issue when compared to Case I. Another possibility is the variation in eccentricity is too high so the test is rerun with a target eccentricity of 0.3 however the results did not improve. Further troubleshooting of the angle selection is required to determine the cause of this anomaly. One persisting theory is that the constant thrust condition is preventing the apogee from being lowered indicating the coasting mechanic is the potential solution. To investigate this the coasting mechanic must be implemented once its functionality is understood.

C. Test Case 3

The third test case involves increasing the inclination of the orbit while holding all the other elements constant as seen in Table 1. The trajectory produced by the function can be seen in Fig. 3 and, again, exhibits several qualities of a low thrust orbit. As expected the inclination change is gradual, requiring several revolutions before the

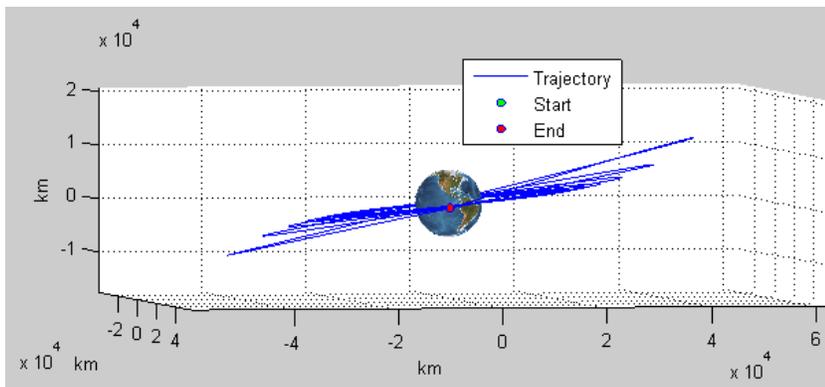


Figure 3. Trajectory of inclination change from 0.1 to 0.3 while holding all other elements constant.

Table 3. Error values between the functions ending values and target orbit values for each of the five orbital elements.

Element	[a, e, i, ω, Ω]
Error (%)	[152.06, 37.28, 0.95, 7.70 0.95]

D. Test Cases 4 & 5

When changes are attempted in the right ascension of the ascending node or argument of perigee the propagation is halted after 16 hours of flight time due to a “NaN” error in Matlab. After using the Debug tool the error has been narrowed down to the proximity value growing too large causing the thrust angles to become NaN’s indicating the function is driving the spacecraft away from the target orbit not towards it. Petropolis^[3] noted in his publication the argument of perigee encounters an anomaly when the thrust vector was purely in or out of plane.

Aerospace Engineering, California Polytechnic University, Undergraduate Student¹

termination condition is met. Furthermore the transfer time appears to be on the correct order of magnitude for electric propulsion, approximately 4.5 days. While the function is able to accomplish the inclination change as directed, there is an increase in semi-major axis that was not specified in the target elements. This is confirmed by looking at the percent differences in the target and ending element values in Table 3. The largest error occurs in the semi-major axis and eccentricity values leading credit to

the issue lying with improper thrust angle determination and constant thrust being applied. As with Case II the weights are varied in an attempt to decrease the variation in semi-major axis however this did not significantly change the orbit suggesting the issue lies with another portion of the function.

Plotting the points that were calculated for either case do not yield anything useful as there too little data to draw conclusions visually. Since Petropoulos never mentioned issues with the right ascension of the ascending node there is an anomaly in the implementation of the equations however cause of this anomaly is still unknown.

E. Future Work

The coasting mechanic that is featured in Petropoulos's derivation of the proximity quotient function would provide the next level of fidelity and realism to the propagated trajectories. This mechanic compares the rate of change calculated at the current true anomaly with the other locations on the osculating orbit. The implementation of this mechanic and how it actually works is not discussed by Petropoulos so it will require a "guess-n-check" methodology to determine the exact process. The current theory is the function uses a brute force method and calculates the set of angles that yield the minimum rate of the change for the full 360 degrees. While this method seems to agree with what Petropoulos stated, it is computationally expensive and would drastically increase the computing time since these trajectories have flight times of days. To practically implement this mechanic work needs to be done to find either a more efficient way to implement the mechanic or use another coding language such as C or Fortran.

As seen in the test cases attempting to change the right ascension of the ascending nod or argument of perigee halt the propagation because the function proximity value grows too large and the function cannot determine thrust angles that yield a converging solution. A less extreme version can be seen in Case II and Case III as the change in the desired element is achieved however the target orbit is not reached due to unintended variations in semi-major axis, eccentricity, or inclination. The current method of solving for the set of angles that drives the rate of change the most negative may place some constraints on the selection introducing the anomaly observed in the test cases. A possible solution is to implement the more complicate method suggested by Petropoulos^[1] where a second derivative test is used on the partials of the time rate of change equation with respect to the thrust angles to verify whether or not the angles are absolute minimums. To fully resolve this issue in future versions of this code, rigorous testing of the current angle selection method and Petropoulos's angle selection method must be performed.

V. Conclusion

The heuristic algorithm known as the proximity quotient, or Q-Law, is replicated using MATLAB and put through several test cases to assess its functional state. To simplify the function the coasting mechanic is not included and all perturbations are ignored. The results from the test cases indicate the function is able to calculate valid trajectories for changes in semi-major axis and nearly valid trajectories for inclination changes. When changes in argument of perigee and RAAN are attempted the function produces an incomplete transfer and the target element is never converged on. When changes in eccentricity are attempted the function propagates through an unintended inclination change and semi-major axis raise. The causes of these anomalies are unknown and will require further testing and research. In spite of the aforementioned complications, the function has demonstrated the ability to propagate a low thrust trajectory for a given set of input conditions. Additionally, because the propagation is carried out using variation of parameters, it can be easily adapted to include any number of perturbations meaning its fidelity is limited only by its versatility. The future development and implementation of the coasting mechanic will provide a better demonstration of real-world low thrust trajectory propagation and should help solve the thrust angle anomaly seen in the test cases. Additionally the method for thrust angle selection will be revisited and tested rigorously to determine the cause of the issues present most prominently in Cases IV and V. Once the coasting mechanic is implemented and the angle selection method is fixed the proximity quotient function will be able to operate to its full potential.

References

¹Petropoulos, Anastassios, "Low Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanic for Coasting," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109-8099.

²Whiffen, Gregory, "Mystic: Implementation of Static Dynamic Optimal Control Algorithm for High-Fidelity, Low Thrust Trajectory Design," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109-8099.

³ Petropoulos, Anastassios, "Refinements to the Q-Law for Low Thrust Orbit Transfers," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109-8099.