Comparison of Low-Thrust Control Laws for Application in Planetocentric Space

Robert D. Falck[*] and Waldy K. Sjauw [†]

NASA Glenn Research Center, Cleveland, OH, 44135, U.S.A.

David A. Smith [‡]

Vantage Parters LLC, Cleveland, OH, 44135, U.S.A.

Recent interest at NASA for the application of solar electric propulsion for the transfer of significant payloads in cislunar space has led to the development of high-fidelity simulations of such missions. With such transfers involving transfer times on the order of months, simulation time can be significant. In the past, the examination of such missions typically began with the use of lower-fidelity trajectory optimization tools such as SEPSPOT[1] to develop and tune guidance laws which delivered optimal or near- optimal trajectories, where optimal is generally defined as minimizing propellant expenditure or time of flight. The transfer of these solutions to a high-fidelity simulation is typically an iterative process whereby the initial solution may nearly, but not precisely, meet mission objectives. Further tuning of the guidance algorithm is typically necessary when accounting for high-fidelity perturbations such as those due to more detailed gravity models, secondary-body effects, solar radiation pressure, etc. While trajectory optimization is a useful method for determining optimal performance metrics, algorithms which deliver nearly optimal performance with minimal tuning are an attractive alternative.

Nomenclature

t	Time, days
a	Semi-major axis, km
e	Eccentricity
i	Inclination, deg
Ω	Right ascension of ascending node, deg
ω	Argument of periapsis, deg
θ	True anomaly, deg
œ	One of the classical orbital elements
E	Eccentric Anomaly
μ	Gravitational parameter of the central body, $\frac{km^3}{s^2}$
m	Spacecraft mass, kg
P_0	Spacecraft propulsion system input power, kW
η	Spacecraft propulsion system efficiency
$\eta_{œ}$	Trajectory maneuver efficiency for each orbital element
I_{sp}	Spacecraft propulsion system specific impulse, s
UTC	Universal Time Coordinated
LEO	Low Earth Orbit
GSO	GeoStationary Orbit
GTO	Geostationary Transfer Orbit
$METL$	Maneuver Efficiency Threshold Limit

[*] Aerospace Engineer, Mission Design and Analysis Branch, rfalck@nasa.gov, AIAA Member.

[†] Aerospace Engineer, Mission Design and Analysis Branch, waldy.k.sjauw@nasa.gov.

[‡] Aerospace Engineer, Mission Design and Analysis Branch, david.a.smith-1@nasa.gov, AIAA Member.

I. Introduction

The development of closed-loop guidance algorithms for low-thrust vehicles in the past decade offers the possibility of delivering near-optimal performance without the need for frequent solution updates due to uncertainty in the perturbation models. In this paper the authors examine the performance of the Proximity Quotient guidance law (Q-Law) developed by Petropoulos[2] and a modified version of the orbital element correction scheme developed by Ruggiero, Pergola, Marcuccio, and Andrenucci.[3] These control laws are intended to provide near-optimal thrust vectors which will deliver a spacecraft from its current state to a later state defined using classical orbital elements. Both algorithms support mechanisms for coasting during periods where the application of thrust is relatively inefficient at achieving the desired effect; for example, attempting to raise the apoapsis altitude when near apoapsis.

In this paper the authors discuss the implementation of the two guidance algorithms and perform a comparison of the two methods above in terms of robustness and performance. For consistency, the guidance laws are implemented in a high-fidelity solar electric propulsion simulation developed at NASA Glenn Research Center in support of the agency's low-thrust mission efforts. This ensures that the guidance routines are operating in a common environment with the same dynamics. The simulation is intended to operate in both 3DOF and 6DOF modes without simplifying assumptions that are often made in optimization tools, and so offers a good examination into how the algorithms would perform on actual spacecraft. The guidance laws are tasked to perform a set of missions common to low-thrust applications: orbit raising from low-Earth orbit (LEO) to geostationary orbit (GEO), orbit raising from geostationary transfer orbit (GTO) to GEO, and orbit raising from a low elliptical orbit to a lunar flyby interface. Guidance laws are compared based on their ability to converge to the targeted states. Performance is compared on the basis of the propellant required to deliver a payload in a given time (mass-optimal) and the minimum time required to deliver a given payload (time-optimal). These solutions are compared with optimal solutions generated by other tools. In addition to quantitative comparisons of performance, the authors present modifications to the guidance algorithms which are shown to improve performance.

II. Overview of Control Laws

A. Diretional Adaptive Guidance

Ruggiero, Pergola, Marcuccio, and Andrenucci[3] describe low thrust maneuvers for efficiently correcting orbital elements; it is based on the principle of determining the thrust angles, at each epoch of interest, that produce the highest instantaneous rate of change in the orbital elements of interest. The accompanying thrust direction for each orbital element is determined. For a general scenario, multiple thrust directions are combined to enable the simultaneous targeting of multiple orbital elements. The authors present, among others, well known analytic expressions for the thrusting angles, maneuver efficiencies and an overall formula for a combined classical orbital element correction (based on an adaptive weighting). Actual implementation of the algorithm (i.e."nuts and bolts") is not shown and left up to the reader. Also outlined is a mechanism for enforcing a threshold on maneuver efficiency to affect propellant consumption. A specific targeting algorithm, based on a subset of the presented material, along with further enhancements has been developed for the Directional Adaptive Guidance (DAG).

For purposes of the DAG algorithm, the thrusting angle expressions, maneuver efficiency expressions, adaptive weighting factor and efficiency threshold concept are taken from the original reference. The DAG targeting algorithm is implemented as a self-contained routine requiring the epoch and vehicle state vector (orbital elements) of interest along with directional weighting factors (new addition to algorithm), desired final targets (orbital elements), stopping tolerance (new addition to algorithm), maneuver efficiency threshold (optional) and control rate limit (optional, new addition to algorithm). The returned parameters consist of the unit thrust direction (RSW frame), mass flow multiplier, indicator of targets achieved (logical flag and epoch time) and the pitch and yaw steering angles (referenced to RSW frame). Depending on specific implementation, the unit thrust direction and mass flow multipliers can be directly used in the equations of motion of interest, or alternatively, the pitch and yaw steering angles maybe used in place of the unit thrust vector.

From the instantaneous vehicle state vector a set of thrusting angles are computed; α is the in-plane (i.e. pitch) angle while β is the out-of-plane (i.e. yaw) angle. These angles, shown in Table 1, are those that produce the largest rate of change (i.e. the instantaneously "optimal" search direction) in each of the five possible orbital elements that may be targeted: semi-major axis (sma), eccentricity (ecc), inclination (inc), right ascension of the ascending node (raan) and argument of periapse (argp). The results make up five sets of 'optimal' in-plane (pitch) and out-of-plane (yaw) angles.

Orbital Element	α	β
Semi-Major Axis (a)	$\arctan\left(\frac{e\sin(\theta)}{1+e\cos(\theta)}\right)$	0
Eccentricity (e)	$\arctan\left(\frac{\sin(\theta)}{\cos(\theta)+\cos(E)}\right)$	0
Inclination (i)	0	$\text{sgn}\left(\cos\left(\omega+\theta\right)\right)\cdot\frac{\pi}{2}$
Ascending Node (Ω)	0	$\text{sgn}\left(\sin\left(\omega+\theta\right)\right)\cdot\frac{\pi}{2}$
Argument of Periapsis (ω)	$\arctan\left(\frac{1+e\cos(\theta)}{2+e\cos(\theta)}\cot\theta\right)$	$\arctan\left(\frac{e\cot(i)\sin(\omega+\theta)}{\sin(\alpha-\theta)(1+e\cos(\theta))-\cos(\alpha)\sin(\theta)}\right)$

Table 1: *Optimal in-plane (α) and out-of-plane (β) thrust angles (RSW pitch/yaw) for maximum instantaneous change of each orbital element, from 3*

From the five sets of thrusting angles, taken from Table 1, the corresponding unit thrust directions (in RSW frame) are computed:

$$\vec{f}_{oe} = \begin{bmatrix} f_R \\ f_S \\ f_W \end{bmatrix} = \begin{bmatrix} \cos\left(\beta\right)\sin\left(\alpha\right) \\ \cos\left(\beta\right)\cos\left(\alpha\right) \\ \sin\left(\beta\right) \end{bmatrix} \tag{1}$$

An adaptive ratio is computed for each desired target, which quantifies the percentage of the change in each orbital element:

$$R_{\oe} = \frac{\oe_T - \oe}{\oe_T - \oe_0} \tag{2}$$

where $\oe$ indicates an instantaneous classical orbital element value, $\oe_T$ indicates a target value and $\oe_0$ indicates the initial value. For the general case of targeting multiple elements at the same time, the reference[3] multiplies each unit thrust direction component by the adaptive ratio and simply sums them to arrive at the overall unit thrust direction; this setup usually results in achievement of the targets.

$$\vec{f}_T = \sum_{\oe}\left(1 - \delta_{\oe,\oe_T}\right) R_{oe}\vec{f}_{\oe} \tag{3}$$

where: $\vec{f}_T$ is the thrust vector in the RSW frame and $\delta_{\oe,\oe_T}$ is the Kronecker delta function:

$$\delta_{\oe,\oe_T} = \begin{cases} 1, \oe = \oe_T \\ 0, \oe \neq \oe_T \end{cases} \tag{4}$$

It should be noted that a targeting algorithm is capable of achieving the targets but the resulting path is not necessarily optimal. The combining of the unit thrust directions in reference 3 ($\vec{f}_T$), in essence, uses a fixed equal weighting of the thrust directions $\vec{f}_{\oe}$ prior to multiplying by the adaptive ratio; this is a limitation that does not allow for different, and perhaps more optimal, solutions to be explored. In order to generate additional (and hopefully better) solutions, a modification of this original scheme is made whereby a set of directional weighting factors, $W_{dir,\oe}$ is introduced. Equation 3 is replaced with equation 5 below:

$$\vec{f}_T = \sum_{\oe} R_{\oe} W_{dir,\oe}\vec{f}_{\oe} \tag{5}$$

where parameter $W_{dir,œ}$ for each orbital element may be a constant (i.e.static) or time-varying control, or a function of some set of variables in the problem. These directional weighting factors $W_{dir,œ}$ are used to multiply the previously computed product of the unit thrust directions and adaptive ratio. For purposes of this discussion they are treated as constants. The sign of each directional weighting factor determines the direction of the targeting (increasing or decreasing). The actual values of the weighting factors are not of importance; it is their relative sizes that produce the desired tuning effect of the DAG algorithm when multiple orbital elements are simultaneously targeted. In fact, for a single target, one can simply set the weighting factor to any value (e.g. unity) and the algorithm will properly reach the target. Experience, so far, has shown good results by using simple constants for the directional weighting factors, making the tuning process, for a more optimal solution, less burdensome.

The use of the directional weighting factors $W_{dir,œ}$ has allowed for different solutions in achieving the targets. It should be noted that the adaptive ratio will naturally drive the unit thrust vector to a near zero value at target achievement. However, to avoid potential numerical noise that may allow the targeting process to continue beyond reasonable achievement, an additional mechanism is needed. The original scheme uses a Kronecker delta function (Equation 4) to guarantee turning off the targeting for a specific classical orbital element. The present implementation has replaced this with a stopping tolerance as follows: the magnitude of the combined thrust direction vector $\vec{f_T}$ is monitored against a stopping tolerance; when all targets are met (within the tolerance), the simulation is stopped. Alternatively, the test for target achievement may be skipped if a numerical optimizer is used in conjunction with the DAG algorithm. For those instances, the stopping is achieved by the optimizer. Prior to target achievement, the present implementation of the algorithm always unitizes the combined thrust vector $\vec{f_T}$ thereby ensuring 100% of the thrust is used unless the thruster is deactivated by other means (e.g. shadow cut out or efficiency threshold) that forces a coasting period. The new stopping scheme is shown below where reference thrust, f_{thrust}, and mass flow rate, $\dot{m}_{flow}$, come from the propulsion model while $mass_{mult}$ is a constant set by the DAG algorithm (default is 1.0; can be 0.0 based on maneuver efficiency cutout when propulsion is turned off).

```
while $||\vec{f_T}|| < tolerance$ do
    compute unit thrust vector $\hat{\mathbf{f}}_T$
    continue targeting using: $\hat{\mathbf{f}}_T * f_{thrust}$ and $mass_{mult} * \dot{m}_{flow}$
end
```

Algorithm 1: Stopping condition for DAG algorithm.

Imposing a specific throttling profile can be achieved via modifying the propulsion model's reference f_{thrust} and $\dot{m}_{flow}$ in the trajectory simulation code. Using the unit thrust direction $\hat{\mathbf{f}}_T$, the corresponding pitch and yaw angles are also computed (RSW frame). Propellant mass flow is controlled via a mass flow multiplier $mass_{mult}$ that is set to either one or zero by the targeting algorithm. This multiplier is applied to the reference mass flow parameter $\dot{m}_{flow}$ from the propulsion model.

Orbital Element	θ_{max}	$\eta_{œ}$
Semi-Major Axis (a)	$\theta_a = 0$	$\eta_a = \lvert\vec{V}\rvert \sqrt{\frac{a(1-e)}{\mu(1+e)}}$
Eccentricity (e)	$\theta_e = \pi$	$\eta_e = \frac{1+2e\cos(\theta)+\cos^2(\theta)}{2(1+e\cos(\theta))}$
Inclination (i)	$\sin(\theta_i + \omega) =$ -e $\sin(\omega)$	$\eta_i = \frac{\lvert\cos(\omega+\theta)\rvert}{1+e\cos(\theta)}(\sqrt{1-e^2\sin^2(\omega)} - e\lvert\cos(\omega)\rvert)$
Ascending Node (Ω)	$\cos(\theta_\Omega + \omega) =$ -e $\cos(\omega)$	$\eta_\Omega = \frac{\lvert\sin(\omega+\theta)\rvert}{1+e\cos(\theta)}(\sqrt{1-e^2\cos^2(\omega)} - e\lvert\sin(\omega)\rvert)$
Arg of Periapsis (ω)	$\cos(\theta_\omega) = \left[\frac{1-e^2}{2e^3} + \sqrt{\frac{1}{4}(\frac{1-e^2}{2e^3}) + \frac{1}{27}}\right]^{\frac{1}{3}} - \left[-\frac{1-e^2}{2e^3} + \sqrt{\frac{1}{4}(\frac{1-e^2}{2e^3}) + \frac{1}{27}}\right]^{\frac{1}{3}} - \frac{1}{e}$	$\eta_\omega = \frac{1+\sin^2(\theta)}{1+e\cos(\theta)} \frac{1+e\cos(\theta_\omega)}{1+\sin^2(\theta_\omega)}$

Table 2: *Point of the orbit providing the maximum rate of change and the expression of the maneuver efficiency, η, for each considered Classical Orbital Element, from 3. Note: corrections have been implemented to the reference definitions of η_e and $\cos(\theta_\omega)$*

In addition to the thrust direction, maneuver efficiencies for each orbital element are also computed per the equations from table 2.

For the targeted elements these efficiencies are combined via simple averaging. The combined efficiency is subject to a lower threshold (defaulted to zero). The thruster is turned off when the efficiency falls below the threshold; this introduces coasting arcs, where needed, into the trajectory. The efficiency test is as follows:

```
Compute $\eta_{avg}$
if $\eta_{avg} < \eta_{threshold}$ then
    $\vec{f}_T = \vec{0}$
    $mass_{mult} = 0$
end
```

Algorithm 2: DAG efficiency cutoff for coasting

By raising the efficiency threshold, the solution can be affected such that the propellant consumption is reduced; this, naturally, happens at the expense of the total mission elapsed time. There will also be a natural upper limit on how much the efficiency threshold may be increased since introducing too many coasting periods will eventually result in the increase in mission time outweighing the potential propellant savings or eventually not being able to achieve some or all of the targets since there is not enough propulsive impulse being applied. There is typically a range of feasible threshold values for a given mission. As a first targeting exercise for any mission it is best practice not to enforce any efficiency threshold (default) to derive a baseline case; the efficiency threshold can subsequently be used to perform parametric studies of interest or to achieve a specific mission objective. Figure 1 shows an example sweep on the combined maneuver efficiency for a GTO to GSO example case; the specific case is not discussed in this paper. Of interest, however, is the trade off between propellant saving and longer mission trip time. As the maneuver efficiency threshold is increased, coasting periods are inserted into the trajectory (when the combined maneuver efficiency is below the threshold); the impact is to lengthen the mission trip time. By forcing the propulsion to be applied more efficiently, the amount of propellant consumed decreases (at the expense of a longer trip time). However, beyond a certain point the increase in trip time outweighs the benefit of increased maneuver efficiency, requiring more propellant to be used to achieve the target orbit; this is a result of the impact of additional coasting periods on the underlying dynamics in the particular problem. An example of such a trend is shown in the "propellant consumed" plot in Figure 1 where going beyond an efficiency threshold of approx. 0.65 for this particular case, the benefits of the increased efficiency vanish. It should be noted that the particular shape of the curve and the crossover point location are very specific to the trajectory under examination. Continued increase of the efficiency threshold beyond the crossover point also shows a sharp increase in trip time. In fact, the efficiency threshold will reach a value beyond which the targets can no longer be achieved since there is simply not enough impulse (no matter how efficient) applied.

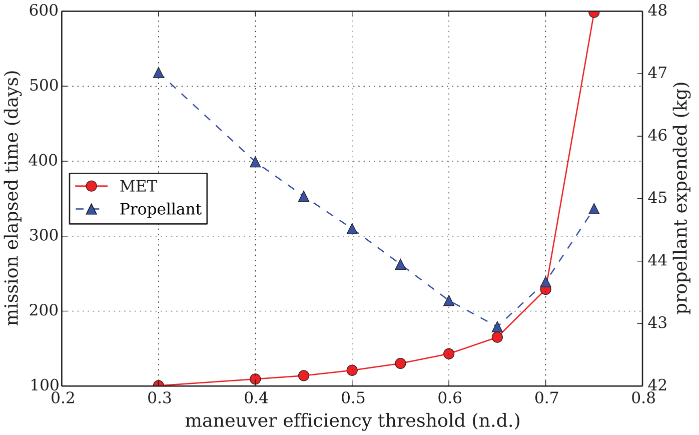

Figure 1: *Example of the effect of maneuver efficiency on flight time and propellant usage. Taken from an internal NASA GRC analysis*

So, we observe that there are two ways to affect a different solution with this targeting algorithm: (1) adjust the directional weighting factors $W_{dir,\oe}$ and (2) impose an efficiency threshold $\eta_{threshold}$.

Finally, another addition to the algorithm is that of control rate limit. If needed, a single control rate limit may be specified which is then applied to both pitch and yaw. This feature can be expanded to separate control rates; however, a single rate limit has proven adequate thus far. Depending on specific simulation code implementation, control rate limit may be available outside the targeting algorithm. The following describes implementation of control rate inside the targeting algorithm. During the trajectory simulation the pitch and yaw angles from the targeting algorithm are tracked between two successive complete integration steps; this provides a control change history, which, coupled with the size of the integration step provides control rate information (i.e. pitch and yaw rates). These are first order approximations whose quality depend on time step and the true angular rate. If there is a rate limit specified (default is none), an allowed angular displacement may be computed from the rate limit and the integration step size. This allowed displacement is then compared against the actual change in the corresponding control angle. If the allowed displacement is violated, the allowed displacement is used, along with the prior control angle to derive the new control angle. Care must be taken when a control angle change crosses a boundary (e.g. 360 / 0 deg, or +/- 180 deg) depending on the trajectory simulation code convention; this is necessary to avoid "false" angular displacement violations. In case any of the control angles are recomputed, based on the control rate limit, the unit thrust direction is recomputed from the new control angles; this new unit thrust direction replaces the original one obtained from the targeting algorithm. The impact of any potential rate limit adjustment is then subsequently, naturally, accounted for in the simulation.

The DAG algorithm may be used with or without a numerical optimizer making it suitable to a wide range of trajectory simulation programs.

B. Proximity Quotient

The Proximity Quotient guidance law, also known as Q-Law, was initially developed by Petropoulos of the Jet Propulsion Laboratory in 2003[2] and saw significant refinements in 2005.[4] The Q-Law is presented in great detail by Petropoulos, so only a brief overview will be given here. The central premise of the Q-Law is that a candidate Lyapunov function, Q is defined:

$$Q = (1 + W_P P) \sum_{\oe} W_{\oe} S_{\oe} \left[\frac{\Delta\left(\oe, \oe_T\right)}{\dot{\oe}_{xx}} \right]^2 \tag{6}$$

The term œ represents each of the five osculating classical orbital elements (a,e,i,Ω,ω). The error function $\Delta(\oe, \oe_T)$ quantifies the difference between the current value of an orbital elements and the desired value. Terms $W_{\oe}$ are weighting factors that indicate the relative importance of changing each element quickly. Petropoulos used these parameters as static control parameters, though in this work we experiment with their use as dynamic control parameters. Term $\dot{\oe}_{xx}$ is the maximum rate of change of a given orbital element within the current orbit. Petropoulos introduced $S_{\oe}$ to scale the semi-major axis error and improve convergence.

$$S_a = \sqrt{\left[1 + \left(\frac{\Delta(a, a_T)}{3a_T}\right)^4\right]} \tag{7}$$

$$S_e = S_i = S_\Omega = S_\omega = 1 \tag{8}$$

Parameters a and a_T are the current and target values of semi-major axis, respectively. Similarly, the penalty P was introduced to prevent the algorithm from seeking solutions which involve unacceptably low periapsis altitudes.

$$P = \exp\left[k\left(1 - \frac{r_p}{r_{p\text{min}}}\right)\right] \tag{9}$$

Here k represents the strength of the penalty function, and r_{pmin} is the minimum allowable periapsis radius. The main idea behind proximity quotient is that Q represents some error from a desired state.

Through a bit of calculus, we can analytically determine an expression for $\dot{Q}$, which due to the Gauss form of the Lagrange Planetary Equations, will involve thrust angles α and β (pitch and yaw in the Hill frame).

$$\dot{Q} = \sum_{\text{œ}} \frac{\partial Q}{\partial \text{œ}} \dot{\text{œ}} \tag{10}$$

It turns out that an analytic solution exists which minimizes $\dot{Q}$ by varying α and β. By using those thrust angles the spacecraft guidance drives Q to zero as quickly as possible at any given instant, thus driving the orbit to the intended target orbit as quickly as possible at any given instant.

Petropoulos also introduces a relative efficiency factor for the rate of change of Q, where the current best-possible rate of Q ($\dot{Q}_n$) is compared to the minimum and maximum possible change rate within the current orbit ($\dot{Q}_{nn}$,$\dot{Q}_{nx}$). Similar techniques have been used by several authors to find Pareto-optimal minimum propellant solutions subject to an upper-bound on trip time or propellant expenditure.[5,3,6,7]

$$\eta_{rel} = \frac{\dot{Q}_n - \dot{Q}_{nx}}{\dot{Q}_{nn} - \dot{Q}_{nx}} \tag{11}$$

Thrust is applied only when this efficiency ratio is above some efficiency threshold parameter. By normalizing with respect to both the minimum and maximum rates of change of Q, the user-specified efficiency threshold has a continuous effect as it is varied from zero to one. The spacecraft thrusts continuously (when not in shadow) when the parameter is zero, and the thrust interval within the orbit becomes increasingly smaller as the efficiency parameter approaches unity. This parameter, as with the weight factors, can be used as either a static or dynamic control.

For this paper, the authors used both an averaging approach like that described in reference 8 and a higher fidelity simulation tool (see reference 9). Determining the extremal optimal rates of change of Q within the orbit ($\dot{Q}_{nn}$,$\dot{Q}_{nx}$) can be computationally expensive and reduce the speed of the simulation, especially when using orbital averaging. To find approximate values for $\dot{Q}_{nn}$ and $\dot{Q}_{nx}$ the following approach was used. First, an interpolating polynomial is established on $-\pi \leq \theta \leq \pi$ with nodes at the Legendre-Gauss-Lobatto points. At each node, $\dot{Q}_n$ is computed analytically. The extremal values of the polynomials are assessed by forming the companion matrix to the interpolating polynomial and computing its eigenvalues. Real-valued eigenvalues are candidates for extrema, along with the endpoints $-\pi$ and π. Once again $\dot{Q}_n$ is computed, this time at each candidate for the extrema, and from this $\dot{Q}_{nn}$ and $\dot{Q}_{nx}$ may be computed. Finally, a polynomial is fit to the *shifted* efficiency at the LGL nodes:

$$\eta_{shifted} = \frac{\dot{Q}_n - \dot{Q}_{nx}}{\dot{Q}_{nx} - \dot{Q}_{nn}} + \eta_{rel} \tag{12}$$

This has the effect of normalizing the values of $\dot{Q}_n$ (which is never positive) throughout the orbit on the range [-1,0] and shifting it upwards by the efficiency threshold η_{rel}. The roots of (12) can again by obtained from the eigenvalues of the polynomial companion matrix. These represent the approximate values of θ at which the spacecraft passes through the relative efficiency threshold (11). Thrust arcs are those regions where $\eta_{shifted} < 0$. Note that (11) and (12) are opposite in sign. In this work $\eta_{shifted}$ was left negative to reflect the fact that $\dot{Q}_n$ is negative.

III. Application of Control Laws

The first two cases to be examined are LEO to GSO and GTO to GSO applications. These examples have been used by Kluever and Oleson for the demonstration of their minimum-time control law development.[10,11] All cases assume that thrust is not available while in the shadow of the Earth.

A. LEO to GSO Transfer

Parameter	**Initial Orbit**	**Final Orbit**
Semi-Major Axis (km)	6927	42164
Inclination (deg)	28.5	0
Eccentricity	0	0
Argument of Perigee (deg)	0	unconstrained
Ascending Node (deg)	0	unconstrained
Mass (kg)	1200	unconstrained
Epoch	01 Jan 2000 00:00:00 UTC	unconstrained

Table 3: *Initial and final orbits for the LEO to GSO case*

The propulsion system has a specific impulse of 3300 seconds, an input power of 10 kW, and an operating efficiency of 65%.

B. GTO to GSO Transfer

Parameter	**Initial Orbit**	**Final Orbit**
Semi-Major Axis (km)	24364	42164
Inclination (deg)	28.5	0
Eccentricity	0.7306	0
Argument of Perigee (deg)	0.1	unconstrained
Ascending Node (deg)	179.6	unconstrained
Mass (kg)	1200	unconstrained
Epoch	22 Mar 2000 00:00:00 UTC	unconstrained

Table 4: *Initial and final orbits for the GTO to GSO case*

For case B the propulsion system has a specific impulse of 1800 seconds, an input power of 5 kW, and an operating efficiency of 55%.

C. GTO to GSO (or Equatorial sub-GSO)

The next case demonstrates the benefit of selecting a longer solution to satisfy a vehicle constraint. The case is that of a fictitious SEP spacecraft with an initial mass of 178 kg and is to, ideally, perform a GTO to GSO transfer; however, there is a 30 kg propellant limit. If the spacecraft cannot perform the full GTO to GSO transfer then a compromise is needed. The final GSO inclination of zero degrees and the final GSO apogee altitude must still be achieved. However, the final perigee altitude may be reduced; it is to be maximized such that the propellant limit is not exceeded. Additionally, care must be taken not to excessively increase the total mission trip time. The use of the maneuver efficiency threshold is demonstrated in this example.

This transfer is performed with the assumption that thrust is not available while in the shadow of the Earth. The propulsion system has a specific impulse of 1300 seconds, an input power of 90 W, and an operating efficiency of approximately 64%.

Parameter	**Initial Orbit**	**Final Orbit**
Semi-Major Axis (km)	24396	42164 or adjust
Inclination (deg)	28.5	0
Eccentricity	0.7283	0 or adjust
Argument of Perigee (deg)	0.1	unconstrained
Ascending Node (deg)	179.6	unconstrained
Mass (kg)	178	unconstrained
Epoch	01 Jan 2016 00:00:00 UTC	unconstrained

Table 5: *Initial and final orbits for the GTO to GSO (or Equatorial sub-GSO) case*

IV. Results

The simulation method used for the targeting guidance algorithms, discussed here, is a high fidelity trajectory simulation using Cartesian EME2000 equations of motion for examples A and B while example C uses Modified Equinoctial Elements (MEE). Algorithm tuning for DAG is based on manual tuning of constant directional weights while using time history plots of the targeted elements to inform the user of the "optimality" of the solution. Depending on the numerical expense of the high fidelity trajectory simulation program, a numerical optimizer may be of benefit in the tuning process. The Q-law algorithm tuning is performed using orbital averaging combined with a collocation-based direct optimization approach. This method is similar to that described in reference 8, but instead of the modified equinoctial costates being implemented as time-varying optimal controls, the input parameters of Q-Law were treated as static optimal controls.

The optimal reference for cases A and B were obtained using a simulation method based on an orbital averaging technique for the equations of motion and a guidance algorithm based on optimal control theory driven by a numerical optimizer.[8,10,11]

Despite the fact that the orbital averaging approach uses simplified perturbation models, the results obtained by it agree very well with those obtained with higher fidelity trajectory simulators. The guidance parameters tuned in the much faster orbital averaging program yield nearly the same results as when applied to the high fidelity simulation, significantly reducing the time spent tuning the algorithms.

A. LEO to GSO Transfer

The results of the LEO to GSO transfer cases are shown in Table 6 while Figures 2 to 3 show corresponding plots for selected parameters.

LEO to GSO Transfer

	MET	**Mass**	**sma**	**ecc**	**inc**	**alt_p**	**alt_a**	**SEP dV**	**Thrusting**	
	days	kg	km	n/d	deg	km	km	m/sec	days	% on
Initial Orbit										
LEO Orbit	0.00	1200.00	6927.00	0.000000	28.50	548.86	548.86	0.00		
Final Orbit										
GSO reference case	198.99	-	**42164**	**0**	**0**	-	-	-	-	-
GSO using DAG	209.11	997.04	42164.46	0.000110	0.028370	35781.87	35790.77	5989.55	189.24	90.50
GSO using Q-law	211.95	992.56	42154.91	0.001100	0.007720	35730.51	35823.03	6135.97	193.42	91.26
- = *information n/a*										

Table 6: *Summary of results for the LEO to GSO example cases*

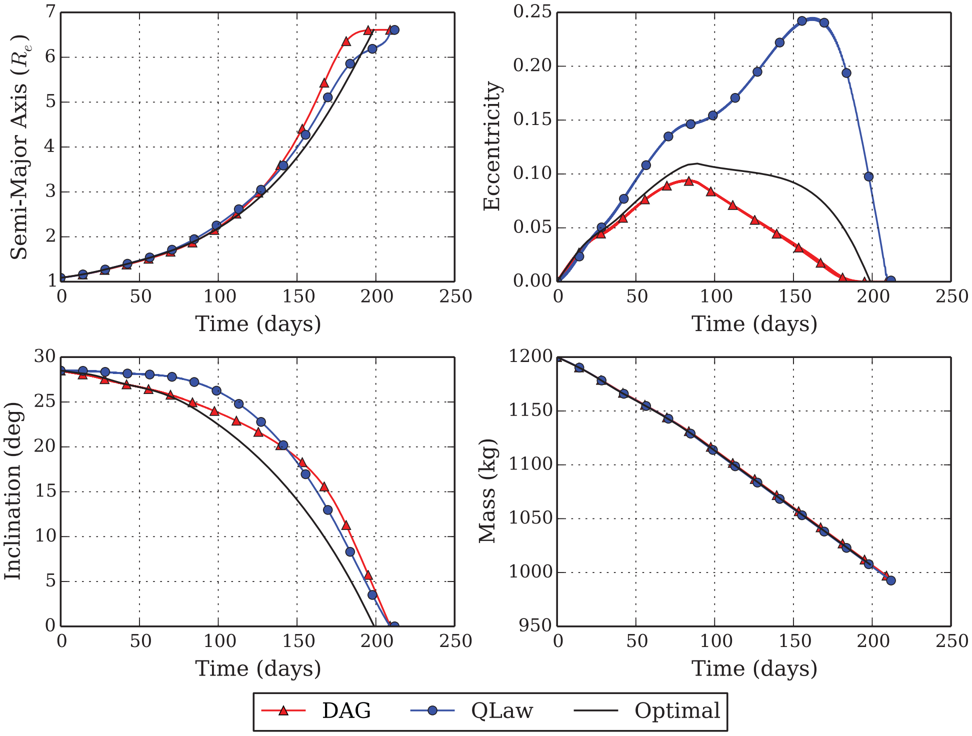

Figure 2: Orbital element and mass time-histories for the LEO to GSO case. Optimal solutions are from reference 8.

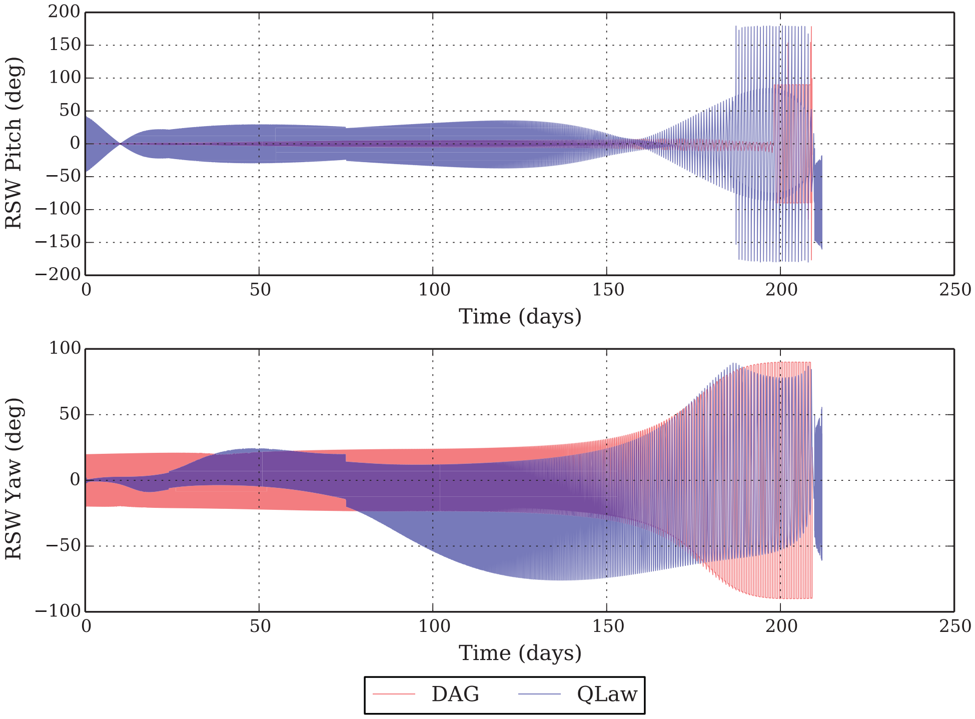

Figure 3: Steering time histories for the LEO to GSO case.

B. GTO to GSO Transfer

The results of the GTO to GSO transfer cases are shown in Table 7 while Figures 4 to 5 show corresponding plots for selected parameters.

GTO to GSO Transfer

	MET	**Mass**	**sma**	**ecc**	**inc**	**alt_p**	**alt_a**	**SEP dV**	**Thrusting**	
	days	kg	km	n/d	deg	km	km	m/sec	days	% on
Initial Orbit										
GTO Orbit	0.00	1200.00	24364.00	0.730600	28.5	185.52	35786.20	0.00		
Final Orbit										
GSO reference case	118.36	-	**42164**	**0**	**0**	-	-	-	-	-
GSO using DAG	125.60	1012.37	42163.58	0.000500	0.028690	35764.16	35806.71	3000.12	123.03	97.95
GSO using Q-law	120.02	1020.41	42158.73	0.001800	0.001000	35704.70	35856.47	2861.10	117.76	98.12
- = *information* n/a										

Table 7: *Summary of results for the GTO to GSO example cases*

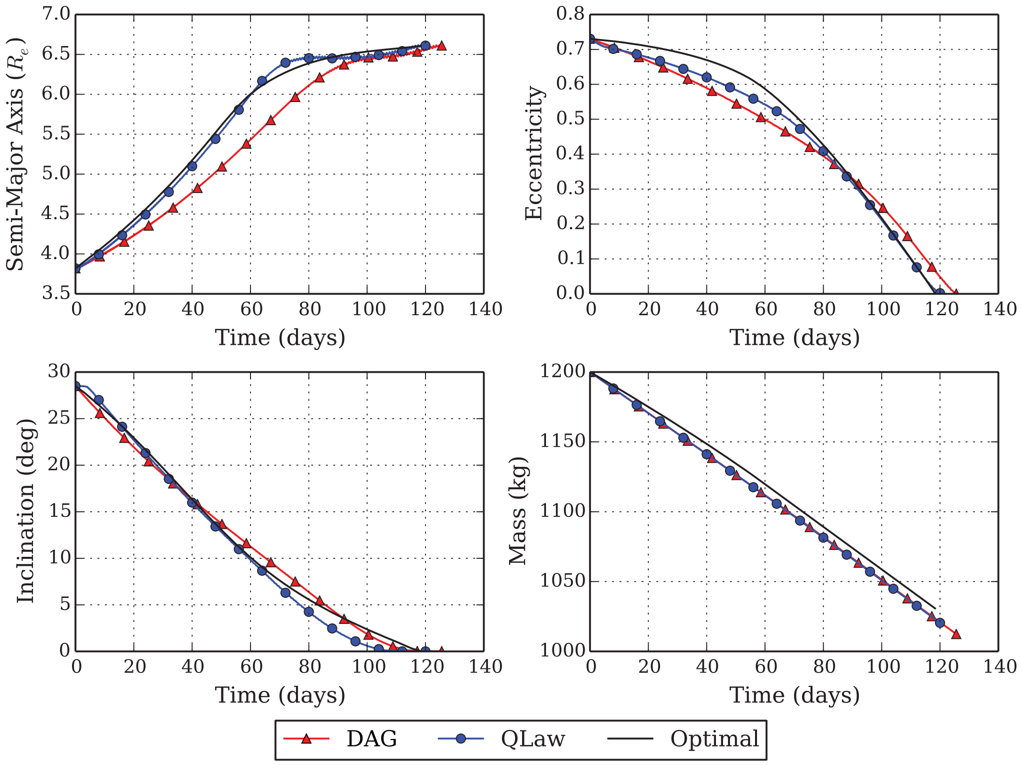

Figure 4: Orbital element and mass time-histories for the GTO to GSO case. Optimal solutions are from reference 8.

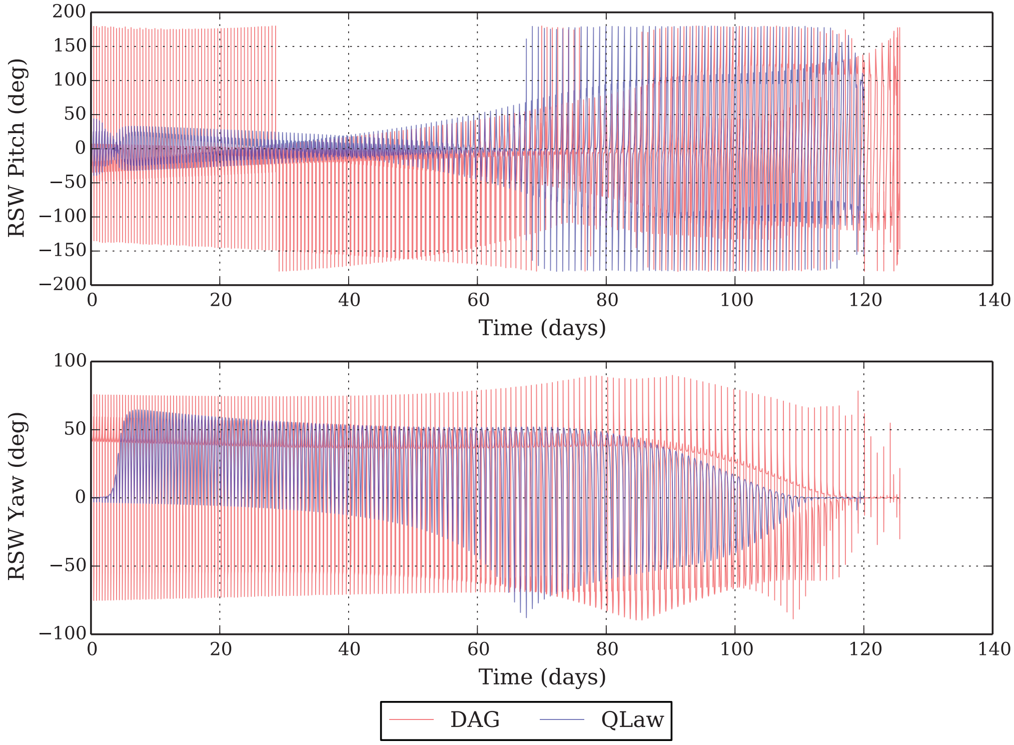

Figure 5: Steering time histories for the GTO to GSO case.

C. GTO to GSO (or Equatorial sub-GSO)

Table 8 shows the results of the GTO to Equatorial sub-GSO example. First, for DAG, the results of the minimum time full GTO to GSO transfer are shown. From this case we observe that the full transfer can be accomplished in approx. 646 days and consumes approx. 38.5 kg of propellant. The stated problem has a 30 kg propellant limit; it is also given that the final perigee altitude may be compromised while all other nominal targets are met and the baseline trip time is not increased excessively. In order to reduce the propellant consumption, a combined maneuver efficiency threshold limit (METL) is implemented (note that the baseline case does not have any limit). The result is to reduce the propellant usage at the expense of trip time. Increasing the maneuver efficiency threshold limit on the full GTO to GSO transfer does reduce the propellant usage; however, the trip time drastically increases compared to the baseline case; to reach the 30 kg propellant tank limit would lengthen the mission unacceptably. So, the compromise on perigee altitude is employed. For DAG, setting the METL to 0.50, 0.52 and 0.54, respectively, and adjusting perigee altitude, the sub-GSO-equatorial cases, shown in Table 8, are obtained. For these cases, the propellant limit of 30 kg is observed while the final perigee altitude is approx. 19k, 16k and 14k km, respectively, lower than that for GSO. The corresponding trip times are lengthened by approx. 7.5, 49.0 and 87.0 days, respectively. It should be noted that intermediate cases leading up to these final result are not shown; they are the result of a rather simple iterative process where the propellant savings of the METL are used to target higher perigee altitudes until the propellant limit is observed. Also note that these answers are not unique; i.e. further refinements are possible depending on the trade off between higher final perigee altitudes at the expense of longer trip times. These cases demonstrate how the maneuver efficiency threshold can be used to insert coasting into the problem to achieve specific mission constraints; in this example notice how the percentage of thruster ”on” time during the transfer decreases from the baseline case amount of approx. 98% to approx. 75, 72 and 68%, respectively, for these particular METL cases.

The results are repeated using the Q-law algorithm and shown in the lower section of Table 8. Note that only a single sub-GSO case was run with Q-Law, as it converged with significantly poorer performance than DAG for this case.

GTO to GSO (or Equatorial sub-GSO)												
	METL	**MET**	**Mass**	**sma**	**ecc**	**inc**	**alt_p**	**alt_a**	**SEP dV**	**Prop used**	**Thrusting**	
	%	days	kg	km	n/d	deg	km	km	m/sec	kg	days	% on
Initial Orbit (GTO)	0.00	0.00	178.00	24396.00	0.728300	28.500000	250.25	35785.47	0.00	0.00		
Final Orbit - DAG												
GSO (full transfer)												
Minimum time	0.00	645.95	139.51	42164.40	0.000035	0.000006	35784.78	35787.74	3105.92	38.49	631.04	97.69
Sub-GSO-Equatorial *(at ∼ prop limit)*												
Sub-GSO-Equatorial 1	50.0	653.54	148.05	32770.64	0.286613	0.000007	17000.02	35784.99	2347.95	29.95	491.11	75.15
Sub-GSO-Equatorial 2	52.0	694.77	147.40	34270.65	0.230301	0.000004	19999.96	35785.07	2404.75	30.60	501.68	72.21
Sub-GSO-Equatorial 3	54.0	733.15	147.70	35270.62	0.195419	0.000004	21999.94	35785.02	2377.55	30.30	496.71	67.75

Final Orbit - Q-Law												
GSO (full transfer)												
	METL	**MET**	**Mass**	**sma**	**ecc**	**inc**	**alt_p**	**alt_a**	**SEP dV**	**Prop used**	**Thrusting**	
	%	days	kg	km	n/d	deg	km	km	m/sec	kg	days	% on
Minimum time	0.00	646.62	139.45	42159.02	0.002590	0.001261	35671.70	35890.07	3111.65	38.55	631.69	97.69
Sub-GSO-Equatorial *(at ∼ prop limit)*												
Sub-GSO-Equatorial 1	48.6	850.00	148.60	35273.27	0.195665	0.001788	21993.39	35796.89	2301.69	29.40	481.96	56.70

Table 8: *Summary of results for the GTO to GSO example case that uses a maneuver efficiency threshold to satisfy a propellant limit*

V. Summary and Conclusions

Two guidance algorithms have been implemented to address the need for achieving near optimal solutions of low thrust trajectory transfers without the need for frequent solution updates. Additionally these algorithms may be used without a numerical optimization package in the loop; this is especially beneficial for use in high fidelity trajectory simulation codes where the spacecraft model and perturbation models may be very extensive and therefore numerically impractical to solve using numerical optimization. The ability to tune the guidance algorithms with relative ease and have the simulation achieve desired targets robustly outweighs any somewhat "sub-optimal" solution; i.e. unforeseen perturbations may require a slightly longer duration but the targets will still be achieved.

The example cases A and B demonstrate the effectiveness of these algorithms when compared to the result of two published optimal guidance cases. It should also be noted that the results of the DAG and Q-law, generated here, compare well for near-minimum time solutions. Additional features, beyond targeting, have been build into the algorithms. Example C demonstrates the use of a lower threshold on maneuver efficiency to trade mission trip time for additional performance; this is done to satisfy a specific mission constraint (here, propellant tank limit). Qualitatively, the implementation of coast arcs due to maneuver efficiency is significantly more complex in Q-Law, as no analytic solution to determine the extremal values of $\dot{Q}_n$ within an orbit are available, and therefore must be obtained numerically.

These algorithms are readily adaptable to a wide range of trajectory simulation programs and can be used with or without the use of a numerical optimization package. They enable the targeting of up to five orbital elements of a closed orbit.

References

[1] Edelbaum, T. N., Sackett, L. L., and Malchow, H. L., "Optimal Low Thrust Geocentric Transfer," AIAA Paper 73–1074, American Institute of Aeronautics and Astronautics, October 1973.

[2] Petropoulos, A. E., "Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting," AIAA Paper 2004–5089, American Institute of Aeronautics and Astronautics, 2004.

[3] Ruggiero, A., Pergola, P., Marcuccio, S., and M., A., "Low-Thrust Maneuvers for the Efficient Correction of Orbital Elements," AIAA Paper 2011-102, September 2011, Presented at the 32nd International Electric Propulsion Conference, Wiesbaden, Germany.

[4] Petropoulos, A. E., "Refinements to the Q-law for low-thrust orbit transfers," 2005, Presented at the 15th AAS/AIAA Space Flight Mechanics Conference.

[5] Gefert, L. P. and Hack, K. J., "Low-thrust control law development for transfer from low earth orbits to high energy elliptical parking orbits," *Astrodynamics 1999*, 2000, pp. 1695–1712.

[6] Falck, R. and Gefert, L., “A method of efficient inclination changes for low-thrust spacecraft,” Tech. Rep. TM–2002–211871, NASA, July 2002.

[7] Gao, Y., “Near-Optimal Very Low-Thrust Earth-Orbit Transfers and Guidance Schemes,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, March 2007, pp. 529–539.

[8] Falck, R. D. and Dankanich, J. W., “Optimization of Low-Thrust Spiral Trajectories by Collocation,” AIAA Paper 2012-4423, August 2012, AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, Minnesota.

[9] Witzberger, K., Smith, D., Martini, M., and Wright, T., *MASTIF’s User’s Manual*, NASA–GRC, 1st ed., 2011, working draft.

[10] Kluever, C. A. and Oleson, S. R., “Direct approach for computing near-optimal low-thrust earth-orbit transfers,” *Journal of Spacecraft and Rockets*, Vol. 35, No. 4, 1998, pp. 509–515.

[11] Conway, B. A. and Kluever, C. A., *Spacecraft Trajectory Optimization*, chap. 5, Cambridge University Press, New York, New York, 2010, pp. 113–137.