MANY-REVOLUTION LOW-THRUST ORBIT TRANSFER COMPUTATION USING EQUINOCTIAL Q-LAW INCLUDING J_2 AND ECLIPSE EFFECTS

Gábor I. Varga, José M. Sánchez Pérez

TERMA GmbH at ESOC, Mission Analysis Section, gabor.varga@esa.int
Mission Analysis Section, ESA-ESOC, jose.manuel.sanchez.perez@esa.int

ABSTRACT

Mission designers addressing the computation of low-thrust many-revolution transfers need versatile and reliable tools for solving the problem with efficient computational times. This paper proposes a Lyapunov feedback control method, Q-law by Petropoulos[1, 2] with algorithm modifications to accommodate for the singularities in the original equations and to include the most relevant perturbations, such as the J_2 perturbation and the effect of coasting during eclipse periods. The optimization of the control-law parameters via a multi-objective evolutionary algorithm (NSGA-II) improves the results significantly and permits to easily compute the minimum time transfer and a well-spread Pareto front, trading transfer time versus propellant.

Index Terms— Q-law, low-thrust, orbit transfer, optimization

1. INTRODUCTION

The recent development of low-thrust electric propulsion systems has brought new challenges in the field of trajectory design. Electric propulsion can offer important mass and launch cost savings compared to chemical propulsion and can become nowadays an enabling technology for many satellite missions despite the longer times that are required for the transfer.

The problem of computing many-revolution, low-thrust orbit transfers has been addressed since several decades and is still an interesting topic of research. The complexity of calculations is higher than that of conventional chemical propulsion, as manoeuvres cannot be considered as instantaneous changes in the velocity, but rather a continuous acceleration. The ultimate goal of a low-thrust orbit transfer computation algorithm is to find the optimal thrust direction at any instant of time, as well as the thrust switching times that ensure an optimal convergence to the target orbit. The optimization can be addressed in the sense of the minimum-time transfer and the minimum-propellant transfer, this latter allowing to study the trade-off between transfer time and propellant.

A direct, Lyapunov feedback control method is proposed for use during the initial stages of the mission design, at which versatile and efficient computations are favoured over high-precision. The Q-law method, as proposed and refined by Petropoulos, provides means of deriving the optimal thrusting angles and the coast phases for an arbitrary transfer between two orbits around a central body. We propose a few modifications to the Q-law algorithms such as formulating the problem in equinoctial elements to avoid the singularities occurring for circular or zero-inclination orbits, including an optimization layer based on Genetic Algorithms to ensure finding the best, near-optimal transfers, and incorporating the effect of the J_2 perturbation and the constraint of coasting during eclipses to have a more realistic model of the dynamics.

Overall the proposed method has proven efficient and robust to generate an optimal, well-spread Pareto-front of the transfer time versus the required propellant mass for an orbit transfer between two arbitrary orbits.

2. EQUINOCTIAL Q-LAW

We solve the orbit transfer problem using a Lyapunov feedback control method developed by Petropoulos[1][2] and called the Q-law, which solves the orbit transfer in an inverse-square gravity field where there is no constraint on the final true anomaly.

The Q-law is based on a proximity quotient, Q, which captures the interdependencies between the orbital elements by means of scaling functions that quantify the proximity of the osculating orbit to the target orbit. Essentially it is an optimistic estimation of the time-to-go, and has units of time squared. During the transfer at each instant the Q-law method chooses the thrust angles that reduce the Q value the most quickly. A coasting mechanism is also incorporated that is based on variable effectivity of the thrust in reducing Q at different true anomalies.

The classical Q-law formulation uses Keplerian orbital elements. In our formulation we use the following equinoctial orbital elements p, f, g, h, k, L as state variables:

$$p = a(1-e^2), \tag{1}$$
$$f = e\cos(\omega+\Omega), \tag{2}$$
$$g = e\sin(\omega+\Omega), \tag{3}$$

$$h = tan(i/2)\cos(\Omega), \quad (4)$$
$$k = tan(i/2)\sin(\Omega), \quad (5)$$
$$L = \omega + \Omega + \theta. \quad (6)$$

where a is the semi-major axis, e the eccentricity, i the inclination, Ω the right ascension of the ascending node, ω the argument of perigee, θ the true anomaly, p the semi-latus rectum and L the true longitude.

The advantage of using this formulation with respect to the classical set is the lack of singularities in the differential equations near circular and equatorial orbits, such as the geostationary ring. Since it is a popular operational orbit, this is a great advantage, as it stabilizes the numerical propagation of the trajectory towards the end of the transfer.

The modified Lyapunov function, or Q function, is defined as:

$$Q = (1+W_pP)\sum_{œ} S_{œ}W_{œ}\left(\frac{œ - œ_t}{\dot{œ}_{xx}}\right)^2, \quad œ = a, f, g, h, k. \quad (7)$$

Using the semi-major axis as the first variable instead of the semi-latus rectum proved to yield a better control when using the equinoctial orbital elements to formulate the Q-law. However, for the propagation of the orbit, the latter is used (p), as the right hand side of the differential equation is less expensive to evaluate.

In Eq. (7) œ are the current, while $œ_T$ are the desired orbital elements, whereas $\dot{œ}_{xx}$ are the maximum rate of change of the corresponding variable over the thrust direction and true anomaly on the osculating orbit. This law enables some elements to be changed in order to make it easier to induce greater changes in other elements later (i.e. increase semi-major axis in order to change inclination with less total delta-V expenditure).

The remaining terms are $W_{œ}$ the scalar weighting factors for each of the equinoctial orbital elements, $S_{œ}$ scaling factor where $S_{œ} = 1$ for $œ = f, g, h, k$ and

$$S_a = \left[1 + \left(\frac{|a - a_t|}{ma_t}\right)^n\right]^{1/r}, \quad (8)$$

which is introduced to prevent convergence to $a = \infty$ (since for $a = \infty$ all the $\dot{œ}_{xx}$ tend as well to an infinity value). P is a penalty function to restrict trajectories with too low perigee passage and has the form

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

where $r_p = p/(1+e)$ is the current periapsis radius and $r_{p\min}$ is the lowest permitted periapsis radius, with k determining the slope of the exponential barrier arising around the critical region.

Experiments show that this formulation is applicable to most transfers present in real life applications, and yields acceptable results. By varying the scalar weights and parameters, different transfers can be realized with distinct characteristics (e.g. ΔV, time of flight, eclipse durations, etc.). The optimal trajectory for a certain goal can be computed by optimizing these parameters.

2.1. Orbit Propagation

The equinoctial form of the Gauss equations is used to numerically integrate the orbit. These formulas are also used to determine the maximum rate of change of each equinoctial element as required by the Q-law. The equations are not disclosed here due to their complexity and length, but can be found in previous publications of the algorithm[3]. The Gauss's equations can be expressed in simplified matrix form:

$$\begin{bmatrix} \dot{a} \\ \dot{f} \\ \dot{g} \\ \dot{h} \\ \dot{k} \\ \dot{L} \end{bmatrix} = \begin{bmatrix} C_{at} & C_{ar} & 0 \\ C_{ft} & C_{fr} & C_{fn} \\ C_{gt} & C_{gr} & C_{gn} \\ 0 & 0 & C_{hn} \\ 0 & 0 & C_{kn} \\ 0 & 0 & C_{Ln} \end{bmatrix} \begin{bmatrix} F_t \\ F_r \\ F_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ C_L \end{bmatrix}. \quad (10)$$

where F_r, F_t and F_n are the perturbing accelerations in the directions $\mathbf{e}_r$, $\mathbf{e}_t$ and $\mathbf{e}_n$ ($\mathbf{e}_r || \mathbf{r}$, $\mathbf{e}_n || (\mathbf{r} \times \dot{\mathbf{r}})$, $\mathbf{e}_t || (\mathbf{e}_n \times \mathbf{e}_r)$), thus in the radial, circumferential and angular momentum directions, respectively.

From the Gauss equations the maximum rate of change of the orbital elements can be calculated by substituting

$$F_t = F\cos\beta\cos\alpha, \quad (11)$$
$$F_r = F\cos\beta\sin\alpha, \quad (12)$$
$$F_n = F\sin\beta, \quad (13)$$

and expressing the extrema of the resulting function with respect to α, β and L, the true longitude. α and β are the thrusting azimuth and declination angles respectively (in the osculating orbital frame). α is measured in the orbit plane from the circumferential direction and positive away from the gravitational centre. β is measured out of the orbit plane and positive along the angular momentum.

The classical Q-law supplies these values analytically, however they are expensive to evaluate due to their complexity and trigonometric expressions. In our formulation analytical expressions can be found for a, h, and k, but unfortunately not for f and g in closed form. However a moderately good approximation can be used for these values, as a numerical solution would be expensive computationally and would also introduce unknown derivatives in the equinoctial Q-law implementation to be accounted for. The maximum rate of change of our Equinoctial Q-law variables are computed as follows:

$$\dot{a}_{xx} = 2Fa\sqrt{\frac{a}{\mu}}\sqrt{\frac{1+\sqrt{f^2+g^2}}{1-\sqrt{f^2+g^2}}}, \quad (14)$$

$$\dot{f}_{xx} \approx 2F\sqrt{\frac{p}{\mu}}, \quad (15)$$

$$\dot{g}_{xx} \approx 2F\sqrt{\frac{p}{\mu}}, \quad (16)$$

$$\dot{h}_{xx} = \frac{1}{2}F\sqrt{\frac{p}{\mu}}\frac{s^2}{\sqrt{1-g^2}+f}, \quad (17)$$

$$\dot{k}_{xx} = \frac{1}{2}F\sqrt{\frac{p}{\mu}}\frac{s^2}{\sqrt{1-f^2}+g}. \quad (18)$$

We note that these are simple algebraic equations that do not involve trigonometric functions.

The time derivative of Q can be determined as

$$\frac{dQ}{dt} = \sum_{œ} \frac{\partial Q}{\partial œ}\dot{œ}, \quad œ = a, f, g, h, k. \quad (19)$$

The Q-law method is based on choosing the thrusting angles α and β at each instant during the transfer such that $\dot{Q}$ is the most negative, therefore ensuring the most rapid decrease of the "distance" Q to the target orbit. These angles can be computed analytically, as $\dot{Q}$ can be rewritten as

$$\frac{dQ}{dt} = D_1\cos\beta\cos\alpha + D_2\cos\beta\sin\alpha + D_3\sin\beta, \quad (20)$$

where D_1, D_2 and D_3 are parameters obtained from the derivatives present in Eq. (19) and can be calculated as

$$D1 = \sum_{œ} \frac{\partial Q}{\partial œ}\frac{\partial \dot{œ}}{\partial F_t}, \quad (21)$$

$$D2 = \sum_{œ} \frac{\partial Q}{\partial œ}\frac{\partial \dot{œ}}{\partial F_r}, \quad (22)$$

$$D3 = \sum_{œ} \frac{\partial Q}{\partial œ}\frac{\partial \dot{œ}}{\partial F_n}. \quad (23)$$

To obtain the optimal thrusting angles Eq. (20) has to be differentiated with respect to α and β and solved as a system of equations for zero

$$\frac{\partial \dot{Q}}{\partial \alpha} = -D_1\cos\beta\sin\alpha + D_2\cos\beta\cos\alpha \quad (24)$$

$$\frac{\partial \dot{Q}}{\partial \beta} = -D_1\sin\beta\cos\alpha - D_2\sin\beta\sin\alpha + D_3\cos\beta. \quad (25)$$

The solution of this problem is

$$\alpha^* = \arctan(-D_2, -D_1), \quad (26)$$

$$\beta^* = \arctan\left(\frac{-D_3}{\sqrt{D_1^2+D_2^2}}\right), \quad (27)$$

where the inverse tangent function in Eq. (26) is the 4-quadrant version to yield a thrusting azimuth angle on $(-\pi, \pi)$, whereas in Eq. (27) the usual $\arctan$ can be used to get a declination angle on $(-\pi/2, \pi/2)$.

2.2. Thrust Effectivity Thresholds

The minimum time transfer problem can be solved with the theory discussed so far, however, usually it is desirable to calculate the trade-off between transfer time and propellant consumption. This is best represented by a partial or full Pareto front of transfer time versus propellant or Delta-V.

For this purpose, two quantities are introduced to measure the effectivity of thrust at a given point on the transfer. This allows differencing between thrusting and coasting arcs. Critical values of these coefficients can be predetermined to cut thrust at certain areas of the orbit, increasing travel time and reducing the used propellant mass.

To calculate the effectivity coefficients, first the maximum and minimum $\dot{Q}$ has to be calculated with respect to the thrusting angles and the orbital position.

$$\dot{Q}_{\min} = \min_{L,\alpha,\beta} \dot{Q}, \quad (28)$$

$$\dot{Q}_{\max} = \max_{L} \min_{\alpha,\beta} \dot{Q}. \quad (29)$$

It is important, that in both Eqs. (28) and (29) the minimization and maximization are to be computed at the same time, such that the extrema of $\dot{Q}$ is to be determined over the 3 dimensional L, α, β space. The extrema of $\dot{Q}$ over α and β can be easily determined, as shown in reference[3].

$$\min_{\alpha,\beta} \dot{Q} = -\sqrt{D_1^2+D_2^2+D_3^2}, \quad (30)$$

$$\max_{\alpha,\beta} \dot{Q} = +\sqrt{D_1^2+D_2^2+D_3^2}. \quad (31)$$

However, the minimum/maximum on L is hard to find. Numerical methods are used (Brent's method and Golden section search) to find the extrema on $L = (0, 2\pi)$. This function usually has two minima and maxima, thus some method has to be added to find the global extrema.

The absolute and relative effectivity coefficients are defined as

$$\eta_a = \frac{\dot{Q}}{\dot{Q}_{\min}}, \quad (32)$$

$$\eta_r = \frac{\dot{Q} - \dot{Q}_{max}}{\dot{Q}_{\min} - \dot{Q}_{max}}. \quad (33)$$

These values are computed at each integration step throughout the transfer, and compared to predetermined cut-off values. Thrusting takes place in the optimal direction defined by α and β if the calculated efficiencies are above the threshold.

Obtaining the D_i values is taking the majority of the computation time in any scenario, but especially in the minimum propellant problem, where subsequent evaluation of Eqs. (28) and (29) is needed for the iteration process to find $\dot{Q}_{\min}$ and $\dot{Q}_{\max}$. Hence, it is essential to calculate these coefficients with the least number of CPU operations.

This requires the optimization of local operations (addition, multiplication, division) in such a way that no unnecessary calculations are carried out. Individual and compound computation sequences are generated for D_1, D_2 and D_3. Since these expressions share a number of operations, and they are not used individually in the algorithm, compound optimization is advised.

Table 1 contains the operations cost of the original and optimized calculations. Function evaluations correspond to mostly trigonometric and exponential functions. Although other methods are available to convert algebraic expressions to C compatible code, it can be clearly seen that optimization of the computation is highly favourable, as it greatly reduces the number of operations leading to large savings of computation time.

Table 1. Operation cost for different optimization levels

Optimization	Add(+)	Mul(*)	Div(/)	Func.
0 (None)	976	1930	564	522
1 (Individual)	164	630	65	58
2 (Compound)	69	270	25	21

2.3. Numerical Integration

The orbit is propagated numerically by integrating the Gauss equations[3] with the Runge-Kutta-Fehlberg[4] (RKF78) method.

Although RKF78 has built-in truncation error calculation and adaptive step size control, it is not used due to the nature of the problem. We use a regularization method to control the time step that based on experience performs well in integrating elliptical orbits. It uses the following formula:

$$\Delta t = \Delta L \sqrt{\frac{r^3}{\mu}} \frac{\sqrt{w}}{1+e} \tag{34}$$

where ΔL is a predefined angular step for the transfer, typically between $5° - 20°$.

While the conventional regularized step size determination algorithm is simply omitting the last product $\sqrt{w}/(1+e)$, this modified expression seems to perform better in estimating the global integration error in transfers where circular and elliptical intermediate orbits are both present.

2.4. Convergence

In some transfer cases the final convergence to the target orbit can present some issues. Especially in planar (and circular) problems, the spacecraft may arrive at the target orbit at a true longitude where reducing the final discrepancy of a and e can result in opposite optimal thrusting directions. As a consequence, the transfer takes longer than needed, which yields a suboptimal solution.

In addition, if the integration step and the thrust-to-mass ratio are both high, overshoot can happen in one or more orbital elements and an oscillatory behaviour begins, which greatly increases the time to converge properly.

To decrease the possibility of these effects, the integration step size is gradually reduced in the final phase of the transfer, when the current orbit approaches the target orbit (Q-value dependent). When the Q-value is approaching zero, the orbit propagation stops. A small error remains in the orbital elements which is adjusted by a stricter convergence criterion. The condition for stopping is

$$Q < R_c \sqrt{\sum_{œ} W_{œ}} \tag{35}$$

where R_c is a dimensionless quantity that controls the error of the remaining distance from the target orbit, nominally set to unity.

2.5. Perturbations and Constraints

The main purpose of the application of the Q-law for orbit transfers is to carry out preliminary mission design and to provide good initial guesses for the evaluation of low-thrust orbit transfers with high fidelity tools. However, certain effects still have great contribution to the dynamics of the system, and can not be neglected.

2.5.1. Zonal harmonics.

The major contributor of the non homogeneous gravity field of celestial bodies is the J_2 harmonic. For long duration transfer around the main body (usually Earth), the secular J_2 perturbation becomes significant and has to be included in the Gauss's equations to better model the dynamics. This is done by adding J_2 terms to the perturbative accelerations[3].

Although including the J_2 effect in some form in the Q-law might seem the proper choice at first, experiments showed that due to the nature of the formulation, the algorithm might spend propellant to fight the periodic terms of the perturbation rather than use the secular terms to its advantage. The possibility of including it in some useful form is not rejected, but our approach is to use the optimization of the Q-law parameters to obtain near-optimal results.

2.5.2. Eclipse.

In most of the applications solar electric propulsion is regarded, which requires a significant amount of power to operate the engine(s). Reduction of battery size generally dictates that thrusting is only possible when the solar panels are illuminated by the Sun. Switching-off the engine during the eclipse period has a significant impact in the orbit evolution, especially for cases in which the eclipse extends a large fraction of the orbit.

In our implementation we consider eclipses as a constraint ad-hoc, thus it is only used to determine whether thrust is possible and it is not included in any way in the computation of the Q-function. For the calculation of the eclipse a cylindrical model is used, whereas the Sun vector is approximated with analytical ephemeris with an accuracy of 36 arcsec between 1950 and 2050.

Three quantities control the thrust switching; the two effectivity parameters and the shadowing coefficient. During the orbit propagation the exact location of the transition is found with an iteration process. This is more important in the case where the switching is caused by eclipse, as the exact location is essential to know. Regarding the switching due to effectivity considerations, the propellant mass and time of flight are not influenced significantly with respect to the case where the iteration is omitted.

3. OPTIMIZATION LAYER

The performance of the transfer using Q-law algorithm is highly dependant on the scalar weights and other parameters present in the formulation. It is essential to introduce an optimization layer to adjust these quantities to obtain the best trajectories. Moreover, the equinoctial formulation of the Q-law does not permit to specify free variables in the initial and target classical orbital elements, as they are coupled in the modified definition. These can only be considered via an inclusion in the optimization.

The main drivers in selecting an appropriate optimization algorithm were

- Global search method,
- Able to handle relatively high dimension,
- Multi-objective.

Local search methods (such as gradient based methods) do not have the ability to converge on the global optimum, and are not able to consider multiple objectives. Evolutionary algorithms have been used previously in similar problems[5][6] with success. A popular method for multi-objective optimization is the NSGA-II genetic algorithm [7] (Elitist Non-dominating Sorting Genetic Algorithm). The properties of this method are summarized in Table 2.

NSGA-II was adjusted to handle multi-objective problems efficiently. The population is sorted based on non-domination into regions of the objective space. In one region no candidate is strictly better than the other, and each subsequent region contains less feasible specimen. Once the suboptimal regions are eliminated, those candidates are given priority that result in a well-spread Pareto-front. This is important so that the diversity of the population on multi-objective level remains high, and does not converge to a small area (i.e. reduce to a single objective solution).

Table 2. NSGA-II properties

Property	Value
Encoding	Real-valued
Initialization	Random on search space
Parent selection	Binary tournament
Recombination	Simulated Binary Crossover (SBX)
Mutation	Random
Survivor selection	Elitist, best on pareto

Formally, the optimization problem is defined as finding the set of parameters

$$(W_a, W_f, W_g, W_h, W_k, \eta_a, \eta_r, m, n, r, \Theta_{rot}) \tag{36}$$

with additional optimization variables corresponding to free boundary values (if the problem requires)

$$(e_0, i_0, \omega_0, \Omega_0, e_t, i_t, \omega_t, \Omega_t) \tag{37}$$

such that the goal is maximized (objective functions are minimized this case)

$$\min(\Delta m, \Delta t). \tag{38}$$

In case of the minimum time problem, only one objective is considered, while in the minimum propellant problem a Pareto-front is computed on the two dimensional objective space.

The optimization variables include therefore 10 parameters of the Q-law, one additional parameter Θ_{rot} and extra variables for each free initial or target orbital element, although it is rare that a problem requires more than 2. Thus in general the total number of optimization parameters lie between 11 and 13.

The transformation parameter Θ_{rot} rotates the system around the north-south axis, and proved to be useful in high eccentricity, high inclination change scenarios. The dynamics are indifferent to this transformation, and other constraints are modified accordingly (e.g. Sun vector). In extreme cases this additional optimization parameter reduced the propellant mass by up to 30%.

Typical values of the NSGA-II parameters (population size, number of generations) used for an optimization is problem dependent. After numerous tests, a population size between 50 and 100 is confirmed to work well with both minimum time and propellant problems, where the higher value is advised to obtain a well-spread Pareto front. The number of generations to run is connected to the difficulty of the transfer, as in a simple GTO to GEO case as few as 10 iterations can yield well converged populations, yet the most challenging cases can require up to 100 generations.

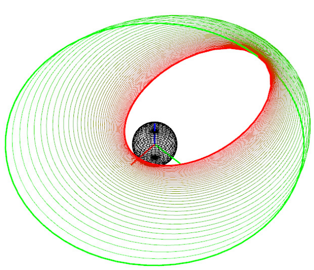
The runtime is in the order of couple minutes for a minimum time problem up to few hours in case of a difficult minimum propellant pareto-front optimization[1].

The optimization layer is programmed to be entirely multi-threaded. One of the advantages of evolutionary algorithms is the little effort it requires to parallelize them. Provided the fitness function computation is self contained, the population can be evaluated simultaneously, since the result of each calculation is independent from the others. This provides a vast speed-up compared to single-threaded implementation.

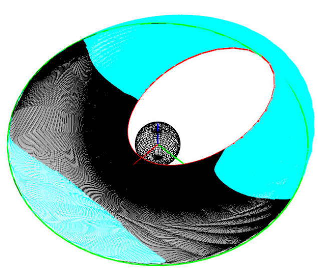
4. IMPLEMENTATION

The Q-law and optimization algorithms were implemented in a core program written in C++. This standalone application is capable of running orbit transfer computations, and communicates via simple input/output files. For easier usage, a second layer is introduced to provide a GUI interface, that contains various modules to run. These usually automate a process that would otherwise be tedious to do manually (e.g. batch computations, parametric analysis, etc.).

The GUI generates input files for the core program and parses the output to provide readily available figures and data, in a seamless way (see Figure 1). The additional advantage of this architecture is that in case changes are required in the GUI, a recompilation of the core program is not needed, and vice-versa. During long computation sequences, the GUI also provides real-time update on the process (e.g. generation of the pareto-front for a specific scenario – see Figure 2 for a GTO-GEO transfer optimization). This information is available after each iteration of the algorithm, which enables the user to interrupt, stop and restart the process as see fit.



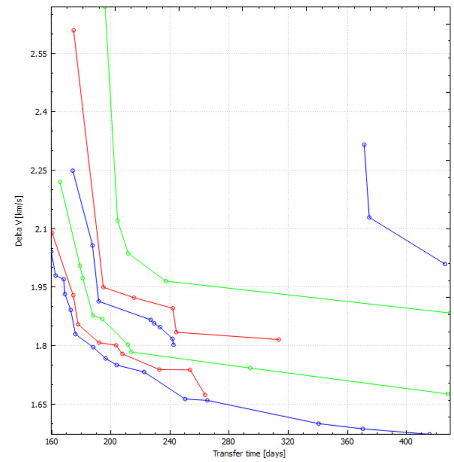
(a) Osculating ellipses



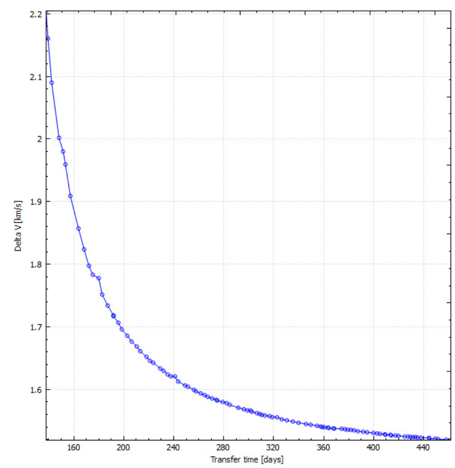
(b) Transfer coast/thrust arcs

Fig. 1. GUI example trajectory evolution plot for a GTO-GEO transfer

[1]Test machine: Intel Core i7-3630M @ 2.40 Ghz, 8 threads, 16 GB Ram



(a) Initial population



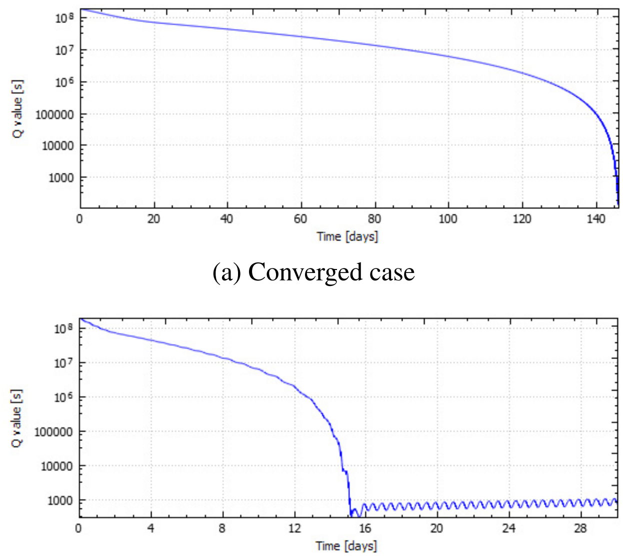
(b) Converged pareto front

Fig. 2. Evolution of the NSGA optimization for a GTO-GEO case

4.1. Challenges

There are some numerical problems that need to be solved in the Q-law algorithm. One is the issue of final convergence, already discussed in Section 2.4. Figure 3 depicts two cases, one that converges successfully, and another one where overshoot occurs in the last revolution. Due to the non-optimal step size an oscillation about the target can be observed. Generally, the optimization process takes care of these cases by eliminating them from the pool of candidates.

Another challenging problem is finding numerically the extrema of the Q-function over one revolution (as described in section 2.2). The function is periodic, but might have several minima/maxima that need to be located. Considerable resources are spent during a pareto-front optimization to find these extrema in an efficient manner. After an initial scan of the function over the period, a combination of derivative-free solvers are used to find the minima and maxima.



(a) Converged case

(b) Not converged case

Fig. 3. Example of difficulties with convergence

5. RESULTS

We selected a number of cases from recent papers to serve as verification for the developed algorithm. Table 3 contains a summary of all the orbit transfer scenarios considered.

Table 3. Transfer scenarios

Case	A	B	C	E
a (km)	7000 42000	24505.9 42165.0	9222.7 30000.0	24505.9 26500.0
e (1)	0.01 0.01	0.725 0.001	0.2 0.7	0.725 0.700
i (deg)	0.05 free	7.05 0.05	0.573 free	0.06 116.00
Ω (deg)	0.0 free	0.0 free	0.0 free	180.0 270.00
ω (deg)	0.0 free	0.0 free	0.0 free	180.0 180.00
T (N)	1	0.350	9.3	2
m_0 (kg)	300	2000	300	2000
I_{sp} (s)	3100	2000	3100	2000

The first batch of 4 transfer cases were originally developed by Petropoulos in his first Q-law paper[1] and have been consistently used since then as a short of test bench for further developments of the Q-law and for other low-thrust optimization algorithms. The same transfer cases have been used more recently to compare the performance of several low-thrust computation methods[8]. These cases are both able to demonstrate the effectiveness of an algorithm and showcase real-life transfers. The dynamics is straightforward in these cases, as it is reduced to the two-body problem with no eclipse constraint, hence the availability of numerous results from different methods. With permission from the authors, the plots from Reference [8] are copied directly in this paper and our equinoctial Q-law results are superimposed, as the raw data was not available to generate fresh graphics. A list of these methods follows, additional description can be found in references[3, 8].

- $MIPELEC$ (Geffroy and Epenoy, CNES)
- T_3D (Dargent, Thales Alenia Space France)
- $Ztool$ (Tarzi, JPL)
- $OPTIFOR$ (Lantoine, JPL)
- $Nominal-Q-law$ (Petropoulos, JPL)
- $GA-Q-law$ (Lee et al[9], JPL)

The remaining scenarios involve test cases with the J_2 perturbation and/or eclipse effects. Unfortunately results of many-revolution low-thrust transfers with these perturbations are scarce in the literature. Nevertheless, we have found a set of suitable test cases in order to perform a brief evaluation of the performance of the equinoctial Q-law.

5.1. A: LEO to GEO transfer

The in-plane circular to circular orbit raising problem has a known theoretical optimal solution[10], thus it is an appropriate test case for the algorithm. On Figure 4 the performance of the equinoctial Q-law as well as numerous other, well-tested algorithms are shown. The calculated pareto-front on the time of flight versus propellant mass shows good agreement across the methods. The minimum time solution provided by the equinoctial Q-law (41.3 kg propellant, 14.5 days time of flight) is also close to the theoretical optimum at around 41 kg.

5.2. B: GTO to GEO transfer

GTO to GEO is one of the most widely discussed orbit transfer scenario for telecommunication geostationary satellites equipped with electric propulsion. This test case assumes a geostationary transfer orbit reached from launch close to the equator, for instance with an Ariane 5 launcher vehicle.

The minimum time solution consists of two phases, where the first phase includes an increment in the semi-major axis, and the second phase applies a close-to-inertial thrust profile to reduce the eccentricity to zero. In the meantime, the small inclination discrepancy (in this particular case, 7°) is corrected by thrusting out of plane at each apogee passage. Figure 5 shows the pareto front for several algorithms. The equinoctial Q-law performs well in this transfer as well, producing results close to those of T_3D. The minimum time transfer for continuous thrust leads to 212.7 kg of propellant consumed and a flight time of 138 days. The computed Pareto-front also follows near perfectly the curve established by the other algorithms.

5.3. C: Elliptic in-plane transfer

This in-plane transfer is characterised by unusually high thrust to mass ratio, 31 mN/kg, which is two orders of magnitude higher than typical configurations nowadays. Being rather far from the very low thrust region, it was questionable whether the algorithm can handle this scenario. Nevertheless, by reducing the time step (or equivalently step in true longitude) drastically, we were able to produce a similarly well spread and good quality Pareto-front as for the previous case.

In the cases analysed so far the Pareto-front behaved as a smooth curve, however, here a critical point can be found in the curve at around 2.5-3 days time of flight, as displayed on Figure 6.

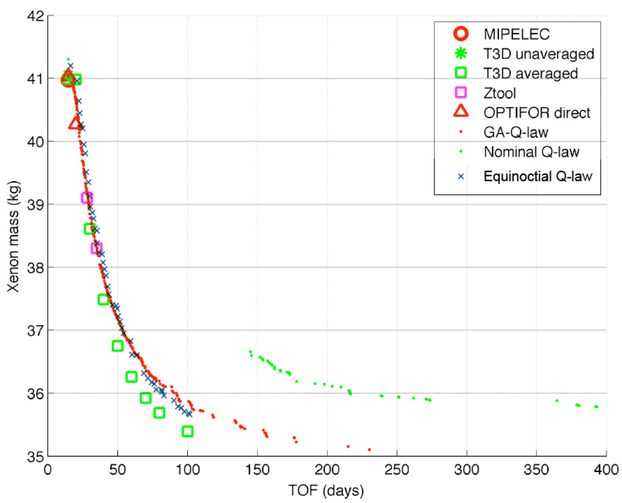


Fig. 4. Pareto front for scenario A, LEO to GEO transfer (additional data from Reference [8])

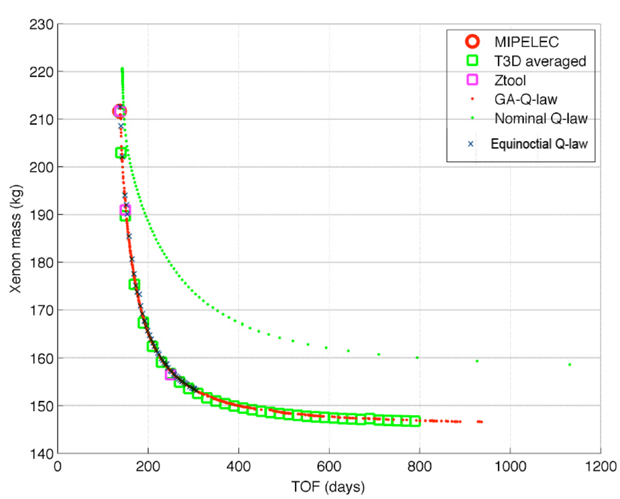


Fig. 5. Pareto front for scenario B, GTO to GEO transfer (additional data from Reference [8])

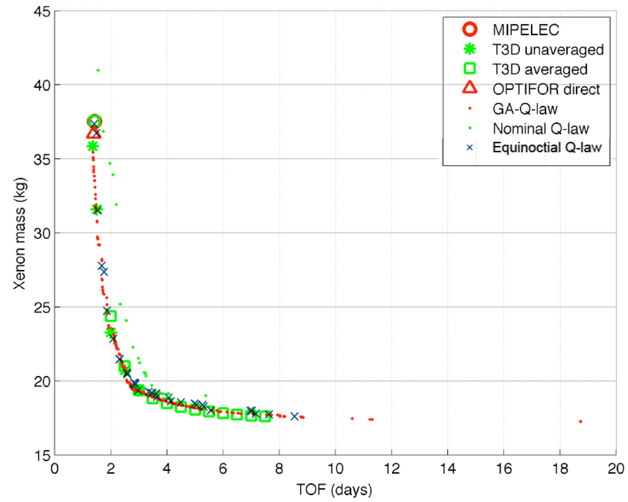


Fig. 6. Pareto front for scenario C, high thrust to weight ratio elliptical transfer (additional data from Reference [8])

5.4. E: GTO to retrograde Molniya

The last case analysed with no J_2 or eclipse effect is a very high inclination change scenario. This transfer truly tested the boundaries of the algorithm, as the optimized equinoctial Q-law performs the best and results in fastest convergence with limited inclination change. The extra optimization parameter (Θ_{rot}) was mainly inspired by this case, as the original algorithm tried to decrease the eccentricity in the first phase instead of increasing it, which is the optimal strategy.

With this extension of the optimization variables, the equinoctial Q-law generated results that are in relatively close agreement with the GA-Q-law (as seen on Figure 7). However, a considerable 10% penalty in propellant mass can be observed at flight times over 200 days between the optimised Q-law methods and the T_3D averaged method, which is recognised as the most mature and robust of the optimization methods analysed in Reference [8]. This is illustrating the intrinsic limitation of the Q-law algorithm as compared to full optimal control methods.

5.5. B-2: GTO to GEO with eclipse

This case is the same as B with the addition of considering the shadowing of Earth. Ferrier and Epenoy studied the effect of seasonal launch dates on the total eclipse duration and time of flight [11]. We reproduced analogous results with our algorithm, shown in Figure 8, where the total transfer time is plotted versus the launch date during one year. The two curves are in close agreement, and during the Apr-Oct period the equinoctial Q-law shows a slight improvement over the results from Ferrier and Epenoy in Reference [11].

We optimized the transfer for 1st January, and used the same optimization parameters to compute the transfers for different initial dates. The same process was repeated optimizing the Q-law parameters for all initial dates and we found no significant discrepancy with the previous results. This is not generally true for an arbitrary transfer, but experience shows that varying the initial date does not affect dramatically the performance of the algorithm when retaining the same optimization parameters. This can be exploited to speed up the computations when performing parametric analysis as is typically needed in early mission design.

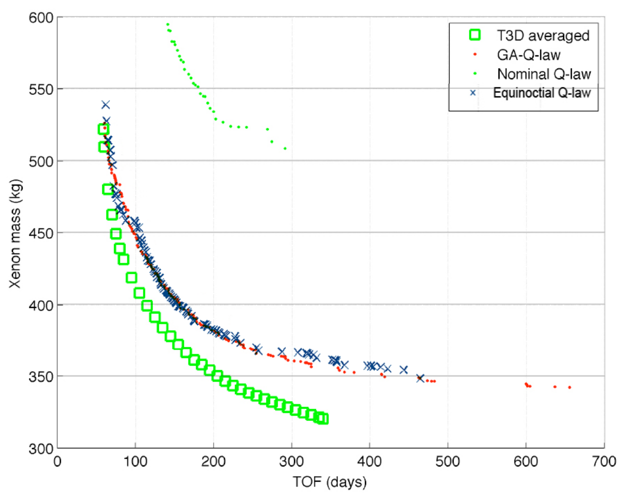


Fig. 7. Pareto front for scenario E, GTO to retrograde Molniya transfer (additional data from Reference [8])

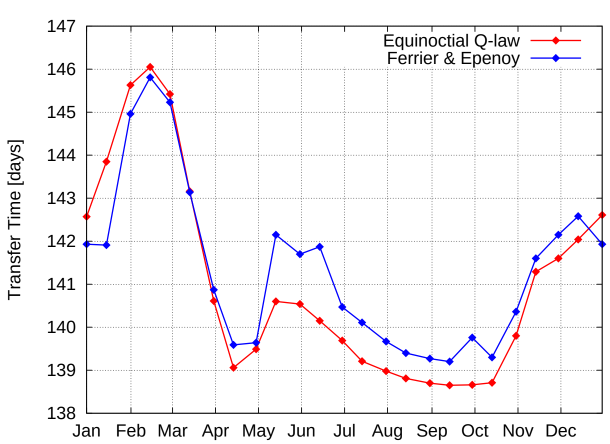


Fig. 8. GTO to GEO minimum-time transfer with eclipses as function of initial date

6. CONCLUSION

The equinoctial Q-law together with NSGA-II optimization proved to be versatile, robust and reliable method for the computation of low-thrust orbit transfers in the applications needed for early mission design. It produces results similar to other, popular methods used nowadays for low-thrust trajectory optimization. Our implementation compares particularly well against the classical Q-law, yielding close to identical results with the advantage that the formulation in equinoctial elements avoids the singularities for circular or zero-inclination orbits. It can efficiently handle eclipse constraint on the thrust showing little sensitivity of the optimal Q-law parameters to eclipse seasonal effects. By including as well the J_2 perturbation more realistic orbit dynamics can be modelled, even if the Q-law is not adjusted to take advantage of the secular J_2 effect, the best transfers can be identified by the NSGA-II optimizer.

Large savings in computation time were achieved with the introduction of parallelization, efficient step size control and the optimization of CPU operations. Compared to the reported classical GA-Q-law runtime it provides optimized solutions within comparable times[9], if not shorter.

7. REFERENCES

[1] A. E. Petropoulos, "Low-thrust orbit transfers using candidate lyapunov functions with a mechanism for coasting," *AIAA Paper 2004-5089, AIAA/AAS Astrodynamics Specialist Conference*, 2004.

[2] A. E. Petropoulos, "Refinements to the q-law for low-thrust orbit transfers," *AAS Paper 05-162, AAS Space Flight Mechanics Meeting*, 2005.

[3] J. M. Sanchez Perez G. I. Varga, "Many-revolution low-thrust orbit transfer computation using equinoctial q-law including j2 and eclipse effects," *AIAA/AAS Paper 15-590, AIAA/AAS Astrodynamics Specialist Conference, Vail, CO*, 2015.

[4] E. Fehlberg, National Aeronautics and Space Administration, 1968 October.

[5] A. E. Petropoulos, "Comparison of multi-objective genetic algorithms in optimizing q-law low-thrust orbit transfers," 2005.

[6] A. E. Petropoulos, "Multi-objective evolutionary algorithms for low-thrust orbit transfer optimization," 2005.

[7] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transaction of evoluationary computation*, vol. 6, 2002.

[8] A. Petropoulos, Z.B. Tarzi, G. Lantoine, T. Dargent and R. Epenoy, "Techniques for designing many-revolution electric-propulsion trajectories," *AAS Paper 14-373, AAS Space Flight Mechanics Meeting, Santa Fe, New Mexico*, 2014.

[9] S. Lee, P. von Allmen, W. Fink, A. E. Petropoulos, and R. J. Terrile, "Design and optimization of low-thrust orbit transfers," *IEEE Aerospace Conference*, vol. 2, 2005.

[10] T. N. Edelbaum, "Optimum power-limited orbit transfer in strong gravity fields," *AIAA Journal*, vol. 3, 1965.

[11] CH. Ferrier and R. Epenoy, "Optimal control for engines with electro-ionic propulsion under constraint of eclipse," *Acta Astronautica*, vol. 48, 2001.

[12] L. Mazzini, "Finite thrust orbital transfers," *Acta Astronautica*, vol. 100, pp. 107, 2014.