Optimization of Electric Propulsion Orbit Raising

By

Michael Scott Kimbrel

B.S. Aerospace with Computer Science Minor
Worcester Polytechnic Institute, 2000

Submitted to the Department of Aeronautics and Astronautics
In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2002

Signature of Author: ....
Department of Aeronautics and Astronautics
May 24, 2002

Certified by: ....
Manuel Martinez-Sanchez
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: ....
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, committee on Graduate Students

Optimization of Electric Propulsion Orbit Raising

By

Michael Scott Kimbrel

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2002 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Aeronautics and Astronautics

ABSTRACT

The increasing power levels now available on geo-synchronous satellites have made it feasible to use electric propulsion engines to perform orbit raising from transfer orbits to GEO. Electric thrusters have very low thrust but are highly efficient, so transfers require the thruster to fire almost continuously for weeks or even months, but also provide significant savings in propellant mass compared to all chemical missions. The complicated nature of the transfer and almost continual firing of the thruster require the thrust angles to be calculated and optimized for the entire transfer time. It is also important to optimize the transition point between the chemical and electric transfers, however the available low-thrust optimization tools are not rapid and flexible enough to allow a broad survey of possible strategies. For this reason, highly analytic derivations have been completed and new optimization software (called MITEOR – MIT Electric Orbit Raising) has been developed in Matlab to optimize thrust angles for constant-low-thrust transfers with no plane changes (2D), as well as for transfers with plane changes (3D) that are restricted to not rotating the argument of perigee or longitude of the ascending node. The 2D version of MITEOR is robust, converges well, and can optimize for transfers with specific initial conditions or display multiple transfer optimizations at once and view trends between transfers. Derivations have also been completed for both 2D and 3D transfers that optimize both thrust angles *and* thrust magnitude. These variable thrust derivations have been found to be completely analytic and require no additional numerical routines. The results of the 2D and 3D variable thrust transfers are typically 5-10% more fuel-efficient than constant thrust, and can be used to easily calculate first cut approximation to the constant thrust cases, providing an optimum upper bound. This project has been completed with promising results and a strong understanding of the analysis. Continued work and improvements on the 3D analysis and code will provide more realistic optimizations and should allow Space Systems / Loral to directly apply MITEOR to the development of their next-generation GEO satellites.

Thesis Supervisor: Manuel Martinez-Sanchez
Title: Professor of Aeronautics and Astronautics

Acknowledgements

I would like to extend my greatest thanks to Prof. Martinez-Sanchez for his work in developing the highly analytic and complicated analysis, code, and for always being available for help. I would also like to thank all of the people involved at Space Systems / Loral, who's sponsorship and support have made this research possible. Special thanks to David Oh, who headed up all of the interactions, meetings, and visits with Space Systems / Loral, and who has put a strong effort into this project. Also thanks to Kwok Ong, who helped error check the analysis and provided a custom orbit propagator to confirm the transfer solutions reached GEO.

Table of Contents

1 Introduction

With the rapidly increasing availability of solar array power in geo-synchronous communication payloads, the possibility arises of performing a part of the orbit raising (see Figure 1) using the on-board electric thrusters, which are provided for orbital corrections. These electric thrusters are very low-thrust, requiring the engine to burn almost continuously throughout the transfers, which themselves can take months. However, the electric thrusters are highly efficient compared to chemical thrusters, so even with a conservative approach in which low-trust operation is restricted to a few weeks and to altitudes above the Van Allen belts, this could result in significant mass savings compared to an all-chemical insertion. The complicated nature of the transfer and almost continual firing of the thruster require the thrust angles to be calculated and optimized for the entire transfer time. The low-thrust portion of the transfer can be chosen to start from a range of orbits accessible to the chemical launcher, and it is important to optimize the combined chemical/electrical operation as well. For this purpose, the available low-thrust optimization tools are not rapid and flexible enough to allow a broad survey of possible strategies. We are developing alternative methods, which can quickly and easily optimize for single transfers, but also have the ability to display multiple transfers at once and show trends between transfers. This greatly facilitates mission planning and the difficult task of optimizing both the chemical and electric transfers. These flexible tools have been developed at MIT with the direct input of systems engineers and mission analysts at Space Systems / Loral. The results of this project will be directly applied to the development of Loral's next generation GEO satellites.

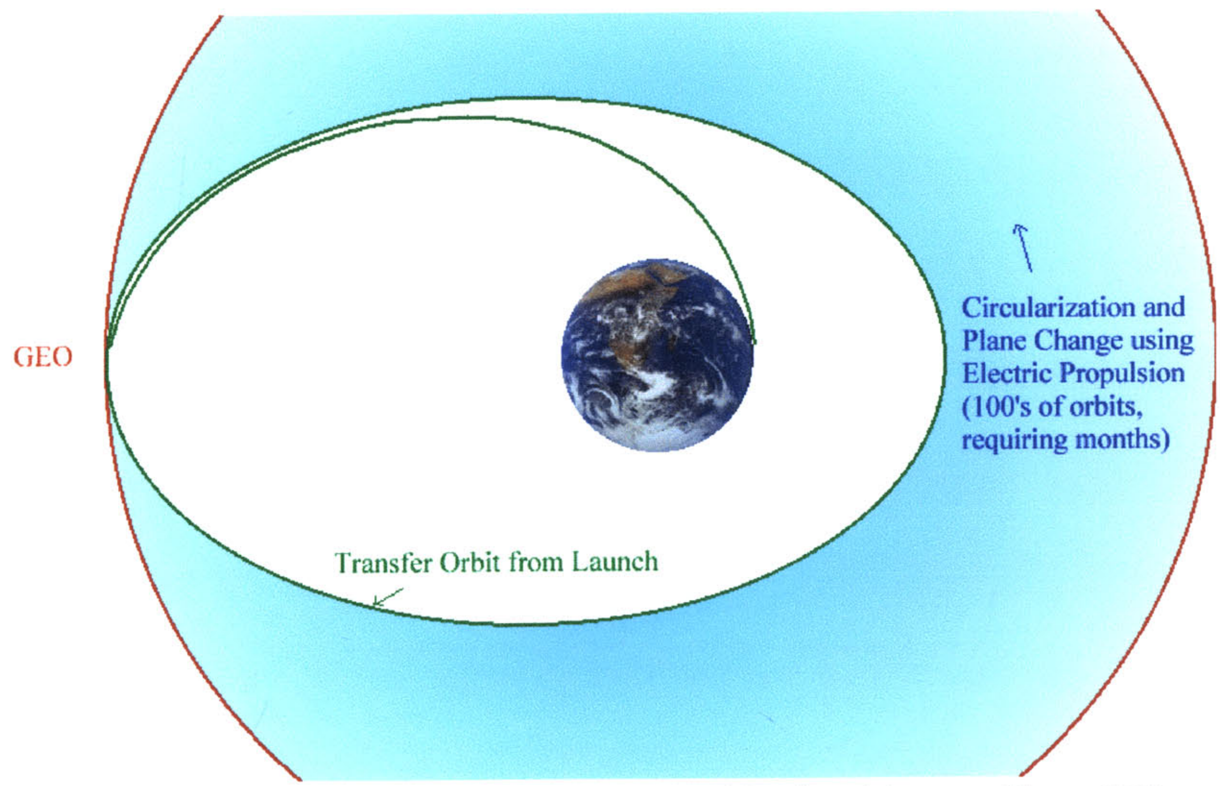


Figure 1: One example of the use of electric propulsion for raising a satellite to GEO

Our highly analytical technique relies on the slowly spiraling nature of the ascent to perform a formal orbit-averaging of the rates of change of the classical orbit elements (Kryloff-Bogoliuboff's method). A first layer of the optimization then derives the form of the intra-orbit perturbation laws of the direction and magnitude of the thrust vector, subject to local constraints on the long-term rates of change of the parameters. In this manner, the analytically derived thrust control laws are found to depend on a set of slowly varying parameters (such as eccentricity and Lagrange multipliers), as yet undetermined, which are found numerically using Runge-Kutta techniques. The outer layer of the optimization is precisely concerned with finding the long-term evolution of these parameters, in a manner that is consistent with the implied intra-orbit controls, and with the desired initial and final orbital conditions.

Analysis has been completed and software developed (called MITEOR – MIT Electric Orbit Raising) for optimizing constant thrust transfers with no plane change (2D). For constant thrust transfers that include plane change (3D), the core of the analysis and software have been completed, but improvements are still being developed. Derivations have also been completed for both 2D and 3D transfers that optimize both thrust angles *and* thrust magnitude. These variable thrust derivations have been found to be completely analytic and require no additional numerical routines. The results of variable thrust transfers are typically 5-10% more fuel-efficient than constant thrust, and can be used as an easily calculated first cut approximation to the constant thrust cases, providing an optimum upper bound. All derivations in this thesis use calculus of variations techniques to optimize by minimizing Δv (or a mass fraction), and currently assume two-body orbital mechanics. Secondary effects like J2, eclipsing, and solar cell degradation are currently not included in the optimizations, but should be added later. The optimizations derived here should work for most all combinations of starting and ending orbits (although the current ending orbit is always assumed to be GEO).

Although it is only now becoming feasible to use electric propulsion for orbit raising, the problem is by no means new, and many people have completed analysis on the subject and come up with optimization routines. While researching previous literature on electric propulsion orbit raising, a database was created to summarize the research papers and facilitate comparisons between the different techniques. The summaries from this database are located in Appendix A (also located in *references.mdb*), and contain most of the relevant orbit raising literature that was available at the MIT Aero/Astro library. An approach was created by Ilgen called HYTOP, and a similar code by Kluever and Oleson, which both promise robust convergence for 2D and 3D cases that also include secondary effects like J2, eclipsing, and solar cell degradation. Comparatively, this is much better than the "standard" numerical routine called SEPSPOT (or SECKSPOT), which is currently used in industry and is more than 20 years old. It often fails to converge, and is very difficult to use for comparing and selecting optimum transfers. Although HYPTOP converges well for most cases, its derivation is only a good approximation to the optimum. Our highly analytic derivation promises exact optimum results for the given conditions and assumptions. Currently, it converges for most all cases, and more importantly, it is custom tailored to be extremely useful for mission analysis and systems level studies.

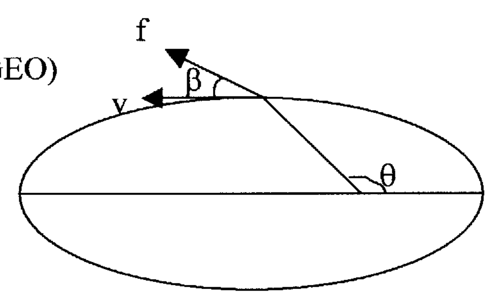
It should be noted that the derivations in this thesis and initial coding were created by Prof. Manuel Martinez-Sanchez of MIT. Michael Scott Kimbrel assisted Prof. Martinez-Sanchez with researching literature; code development, debugging, and testing; modeling and visualizations; document and presentation preparation; and general troubleshooting and brainstorming.

2 Derivations of EP Orbit Raising Optimizations

The highly analytic approach taken to create the derivations of electric propulsion (EP) orbit raising was to start simple and then expand in complexity, which can seen in the four major sections in this chapter. The first section (2.1) derives the optimization for constant thrust EP orbit raising with no plane change (2D). Section 2.2 extends this constant thrust derivation to include plane changes (3D), but also includes the assumption that the argument of perigee (ω) and longitude of the ascending node (Ω) remain constant during the transfer (to simplify the problem). Sections 2.3 (2D) and 2.4 (3D) both derive the optimizations for transfers involving variable thrust, finding both optimal thrust angle profiles and throttling profiles throughout the transfers. These variable thrust optimizations (2.3 and 2.4) can be solved completely analytically, compared to the constant thrust cases (2.1 and 2.2) that require numerical routines.

The optimizations were all derived using calculus of variations. The optimizations make use of orbit averaging (the assumption that the orbital elements are approximately constant during each orbit), and assume two-body orbital dynamics. Higher order terms (J2, etc) and other constraints can be added to the derivations in the future. The derivations also used the standard set of orbital elements (not the equinoctial elements), since they are easier to understand physically and no singularities occur in these derivations. The lack of singularities is because the longitude of the node and the argument of perigee are not involved. All derivations have the benefit of producing exact optimum solutions (given the above assumptions), since they have been extended analytically as far as possible (2.3 and 2.4 have completely analytic solutions). The optimizations can also be run for most all starting and ending conditions (within reason).

The following terminology applies for all derivations:



a	= semi-major axis (a_G is semi-major axis at GEO)
e	= eccentricity
i	= inclination
θ	= true anomaly
ω	= argument of perigee
Ω	= longitude of the ascending node
v	= velocity
Δv	= velocity increment ("delta v")
α	= out-of-plane thrust angle measured positive upwards from velocity
β	= in-plane thrust angle measured positive outwards from velocity
f	= thrust per unit mass
fo	= reference thrust per unit mass of the transfer
φ	= thrust modulation, on (1) or off (0), function of θ
λ	= Lagrange multiplier
Λ	= non-dimensional Lagrange multiplier (2D)
Λ_a	= non-dimensional Lagrange multiplier to constrain semi-major axis (3D)
Λ_e	= non-dimensional Lagrange multiplier to constrain eccentricity (3D)
Λ_i	= non-dimensional Lagrange multiplier to constrain inclination (3D)

μ = graviational constant of Earth (3.986×10^{14} m^3/s^2)
m = instantaneous mass of spacecraft
m_o = initial mass of spacecraft at beginning of transfer
m_f = final mass of spacecraft at end of transfer
m_{pay} = mass of payload
m_{ps} = mass of power supply and propulsion system
c = specific impulse (or exhaust velocity)
$\bar{c}$ = orbit averaged specific impulse
η = engine efficiency
t = time (as a variable)
T = final time of transfer

The following parameters have definitions specific to a particular derivation

C = integral defined in 2D Eq.(15), 3D (33), 2Dvar (94), 3Dvar (164)
M = integral defined in 2D (16), 3D (34), 2Dvar (95), 3Dvar (165)
I = integral defined in 3D (35), 3Dvar (166)
V = integral defined in 2D (17), 3D (36)
V_2 = integral defined in 2Dvar (96), 3Dvar (167)
Φ = cost function defined in 2D (20), 3D (41), 2Dvar (79), 3Dvar (156)
H = Hamiltonian function defined in 3D (43)
x,y,z = parameters defined in 3D (47)
J_{ci},J_{vi},J_{cv} = Jacobians defined in 3D (59)
F,G = ratios of Jacobians defined in 3D (58)
$\Lambda_{ao},\Lambda_{eo},\Lambda_{io}$ = initial values of non-dimensional Lagrange multipliers (2Dvar,3Dvar)
$\lambda_{io},\lambda_{ao}$ = ratios of $\Lambda_{ao},\Lambda_{eo},\Lambda_{io}$ defined in 3Dvar (186)
P = power (2Dvar,3Dvar)
F = thrust (2Dvar,3Dvar)
ε = function of e defined (114) (2Dvar,3Dvar)
Δv_{RMS} = RMS (root mean squared) velocity increment defined (124) (2Dvar,3Dvar)
v_{ch} = characteristic velocity defined (129) (2Dvar,3Dvar)

2.1 Derivation of 2D EP Orbit Raising

2.1.1 Analysis

2.1.1.1 Introduction

The derivation of 2D electric propulsion orbit raising shown here is a highly analytic approach that results in a truly optimum solution for transfers that assume constant thrust, two-body orbital dynamics, and no plane change. The derivation begins with the basic perturbation equations of the orbital elements, such as those in Battin's orbital dynamics book (Battin, pg 489). The equations are then expressed as differential equations with respect to the true anomaly instead of time. This form of the equations is the most convenient when assuming orbit averaging. This assumption stems from the fact that over the entire transfer the orbital elements (a, e and Δv) vary little within each orbit, so we can assume the orbital elements are constant within each orbit.

This assumption allows the optimization to be broken down into two levels. The first level is the intra-orbit optimization of the thrusting angle within the orbit subject to the local constraint of the long-term rate of change of the semi-major axis. This allows for the calculation of the thrusting angle direction for each orbit of the entire transfer. The second level of the analysis optimizes over the entire transfer to find the optimal change in e and a to give a minimum Δv. This assures that the thrusting profiles generated in the first level of the optimization will transfer the spacecraft to GEO (or other given end condition) in a way that minimizes the fuel required and maximizes payload.

2.1.1.2 Basic Governing Equations and Orbit Averaging

We first start with the basic orbital perturbation equations that can be found in Battin's orbital dynamics book (Battin, pg 489). The perturbation equation for the argument of perigee (ω) can be ignored, since we are assuming two-body orbital mechanics (no outside forces to modify ω) and the optimal solution would not require adjusting ω. Also, since it is only a 2D problem, there is no need for the perturbation equations of the longitude of the ascending node (Ω), and inclination (i). We then have these two equations to define the orbit.

$$\frac{da}{dt} = \frac{2a^2 v}{\mu} a_{dt} \tag{1}$$

$$\frac{de}{dt} = \frac{1}{v}\left[2(e+\cos\theta)a_{dt} - \frac{r}{a}(\sin\theta)a_{dn}\right] \tag{2}$$

To define the position within the orbit we can specify the perturbation of the true anomaly (θ) from the angular momentum (h) definition: $r^2\dot{\theta} = h = \sqrt{\mu a(1-e^2)}$ Solving for $\dot{\theta}$ gives the perturbation equation for the true anomaly.

$$\frac{d\theta}{dt} = \sqrt{\frac{\mu}{a^3(1-e^2)^3}}(1+e\cos\theta)^2 \tag{3}$$

The rate of change of Δv comes from the simple equation $F=ma$, giving

$$\frac{d\Delta v}{dt} = f \tag{4}$$

These equations can then be rearranged to suit our derivation by using the following definitions.

$$a_{dt} = f\cos\beta$$
$$a_{dn} = f\sin\beta$$
$$v = \sqrt{\frac{\mu}{a(1-e^2)}(1+e^2+2e\cos\theta)}$$

where f is the tangential thrust per unit mass, β is the in-plane thrust angle (measured positive outward), and v is the orbital velocity, which can be computed easily from the energy conservation equation $\frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}$ and $r = \frac{a(1-e^2)}{1+e\cos\theta}$. (Note also that Battin uses f for the true anomaly, where as we call this θ.)

Substituting these definitions into Equations (1) and (2) and combining that with a thrust modulation function $\varphi(\theta)$, the perturbation equations with respect to time are now found to be:

$$\frac{da}{dt}=2f_o\varphi(\theta)\sqrt{\frac{a^3}{\mu(1-e^2)}}(1+e^2+2e\cos\theta)\cos\beta(\theta) \tag{5}$$

$$\frac{de}{dt}=f_o\varphi(\theta)\sqrt{\frac{a(1-e^2)}{\mu}}\frac{2(e+\cos(\theta))\cos\beta(\theta)+\dfrac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{\sqrt{1+e^2+2e\cos\theta}} \tag{6}$$

$$\frac{d\theta}{dt}=\varphi(\theta)\sqrt{\frac{\mu}{a^3(1-e^2)^3}}(1+e\cos\theta)^2 \tag{7}$$

$$\frac{d\Delta v}{dt}=f_o\varphi(\theta) \tag{8}$$

The term $\varphi(\theta)$ is an arbitrary modulation function for the thrust acceleration (f) such that $f{=}f_o\varphi(\vartheta)$, where f_o is a reference thrust. Within each orbit $\varphi(\theta)$ can be determined by the analysis or by prescribing a specific modulation imposed by eclipsing or any other constraint. In its most simple form, it allows for the possibility of turning the engine on (φ=1) or off (φ=0) during the transfer. It is assumed in this derivation that φ=1 throughout the transfer, and it is only kept in the equations for ease of further work on transfers that utilize switching conditions (i.e. eclipsing) or other modulation functions.

Equations (5), (6) and (8) are now divided by dθ/dt (ignoring ω, the secular perigee rotation). This puts the equations in terms of dθ:

$$\frac{da}{d\theta}=2f_o\varphi(\theta)\frac{a^3(1-e^2)}{\mu}\frac{\sqrt{1+e^2+2e\cos\theta}}{(1+e\cos\theta)^2}\cos\beta(\theta) \tag{9}$$

$$\frac{de}{d\theta}=f_o\varphi(\theta)\frac{a^2(1-e^2)^2}{\mu}\frac{2(e+\cos(\theta))\cos\beta(\theta)+\dfrac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{(1+e\cos\theta)^2\sqrt{1+e^2+2e\cos\theta}} \tag{10}$$

$$\frac{d\Delta v}{d\theta}=f_o\sqrt{\frac{a^3(1-e^2)^3}{\mu}}\frac{\varphi(\theta)}{(1+e\cos\theta)^2} \tag{11}$$

Assuming that the slowly varying orbital elements remain constant over one orbit, we can perform 'orbit averaging' and write the long-term evolution equations as follows.

$$\left\langle \frac{da}{d\theta} \right\rangle_\theta = \frac{2 f_o a^3}{\mu} C \tag{12}$$

$$\left\langle \frac{de}{d\theta} \right\rangle_\theta = \frac{f_o a^2}{\mu} M \tag{13}$$

$$\left\langle \frac{d\Delta v}{d\theta} \right\rangle_\theta = f_o \sqrt{\frac{a^3}{\mu}} V \tag{14}$$

Where C, M, and V are the integrals over one orbit and are defined as:

$$C = \frac{1-e^2}{2\pi} \int_0^{2\pi} \varphi(\theta) \frac{\sqrt{1+e^2+2e\cos\theta}}{(1+e\cos\theta)^2} \cos\beta(\theta) d\theta \tag{15}$$

$$M = \frac{(1-e^2)^2}{2\pi} \int_0^{2\pi} \varphi(\theta) \frac{2(e+\cos\theta)\cos\beta(\theta) + \frac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{(1+e\cos\theta)^2\sqrt{1+e^2+2e\cos\theta}} d\theta \tag{16}$$

$$V = \frac{(1-e^2)^{3/2}}{2\pi} \int_0^{2\pi} \frac{\varphi(\theta)}{(1+e\cos\theta)^2} d\theta \qquad \text{Note: for } \varphi(\theta) \equiv 1, \text{ V=1} \tag{17}$$

From here on, we will omit the brackets in the understanding that the derivatives no longer contain intra-orbit variations, only those of a long-term nature.

We also want to put Eqs. (12) - (14) in terms of *de*, so dividing Eqs. (12) and (14) by Eq. (13) gives:

$$\frac{da}{de} = 2a \frac{C}{M} \tag{18}$$

$$\frac{d\Delta v}{de} = \sqrt{\frac{\mu}{a}} \frac{V}{M} \tag{19}$$

2.1.1.3 Intra-Orbit Optimization

For the first part of the optimization (intra-orbit), we want to minimize $\frac{d\Delta v}{de}$ but subject to the local value of $\frac{da}{de}$ (to be determined). This is done by combining the two Equations (18) and (19) with a Lagrange multiplier (λ):

$$\Phi = \frac{d\Delta v}{de} - \lambda \frac{da}{de} \tag{20}$$

Substituting Equations (18) and (19) into (20) and making the variations of Φ equal to zero, the following optimality condition is found:

$$\boxed{M\delta C + (V\Lambda - C)\delta M - \Lambda M \delta V = 0} \qquad \text{where } \Lambda = \frac{\lambda}{2}\sqrt{\frac{\mu}{a^3}} \tag{21}$$

The variations δC, δM, and δV can be computed directly from the definitions (15), (16), and (17), in terms of the variation profiles δβ and δφ. Since these variations are independent, setting the coefficient of δβ to zero, the equation for the β (thrusting angle) profile can be found:

$$\boxed{\cot\beta(\theta) = \frac{1+e\cos\theta}{(1-e^2)^2\sin\theta}\left[\left(\frac{M}{V\Lambda - C}\right)(1+e^2+2e\cos\theta) + 2(1-e^2)(e+\cos\theta)\right]} \tag{22}$$

Similarly, setting the coefficient of δφ to zero would give the engine off-on switching condition. This has not yet been implemented, as for now the engine is always on (V=1).

Equation (22) for β(θ) can be substituted into (15),(16), and (17) to calculate C, M, and V as functions of (e,Λ). (Iteration is required, because β (Eq. (22)) contains V, C, and M as well.)

If Λ were known, then the profile of β could be found. Finding Λ happens in the second part of the optimization, where the analysis optimizes the change in *e* and *a* to give a minimum *Δv* for the whole transfer.

2.1.1.4 Outer Optimization

The second layer of optimization involves minimizing the full *Δv* by choice of the optimum profile Λ(e) along the trajectory. We have

$$\Delta v = \int_{e1}^{e2} \frac{1}{2}\sqrt{\frac{\mu}{a}}\frac{V}{M}\,de \tag{23}$$

and can calculate δ(Δv) as an integral involving δa and δΛ. But these variations are interrelated through Eq. (18), from which, after taking variations, we obtain

$$\delta\Lambda = \frac{\dfrac{d(\delta a)}{de} - 2\dfrac{C}{M}\delta a}{2a\dfrac{\partial(C/M)}{\partial\Lambda}} \tag{24}$$

When this is substituted back into $\delta(\Delta v)$, an integration by parts is needed to deal with the $d(\delta a)/de$ term. After this, setting the coefficient of δa in the integral to zero gives the overall optimality condition:

$$\boxed{\sqrt{\frac{\mu}{a^3}}\left[\frac{1}{2}\frac{V}{M}-\frac{\frac{\partial(V/M)}{\partial\Lambda}}{\frac{\partial(C/M)}{\partial\Lambda}}\right]+\frac{d}{de}\left[\frac{1}{2}\sqrt{\frac{\mu}{a^3}}\frac{\frac{\partial(V/M)}{\partial\Lambda}}{\frac{\partial(C/M)}{\partial\Lambda}}\right]=0} \tag{25}$$

Rearranging Equation (25) and defining F as

$$F=\frac{\frac{\partial(V/M)}{\partial\Lambda}}{\frac{\partial(C/M)}{\partial\Lambda}} \tag{26}$$

gives the equation for the change in Λ with respect to e. This equation can be used to find Λ given an intial value of e (see Figure 2 in 2.1.2).

$$\boxed{\frac{d\Lambda}{de}=\frac{F\frac{C}{M}-\frac{\partial F}{\partial e}-\frac{V}{M}}{\frac{\partial F}{\partial\Lambda}}} \tag{27}$$

2.1.2 2D Constant Thrust Results

Using Equation 17, a graph can be drawn of Λ and e for a family of solutions (see Figure 2). Each line on this graph represents a possible optimal trajectory, depending on an initial Λ and e. For each one of these curves the corresponding values of Δv and a can be found by going back through the analysis, which leads to the graphs of Figure 3 and Figure 4.

To apply the results, first use Figure 3 to select an initial a and e and identify the corresponding trajectory. Using that trajectory and the initial e with Figure 4 gives the minimum Δv required for the mission (as well as how it changes over the entire mission). Using that trajectory and the initial e with the information in Figure 2 can give the β profiles for the entire orbit raising. Some examples of this are shown in Figure 6 for a trajectory which starts at $eo = 0.5$ and $ao/af = 0.5$.

Figure 2 below shows the evolution of the Lagrange multiplier vs. eccentricity for a large number of optimal orbits, each characterized by an arbitrarily chosen initial value, and denoted by a number from 1 to 44.

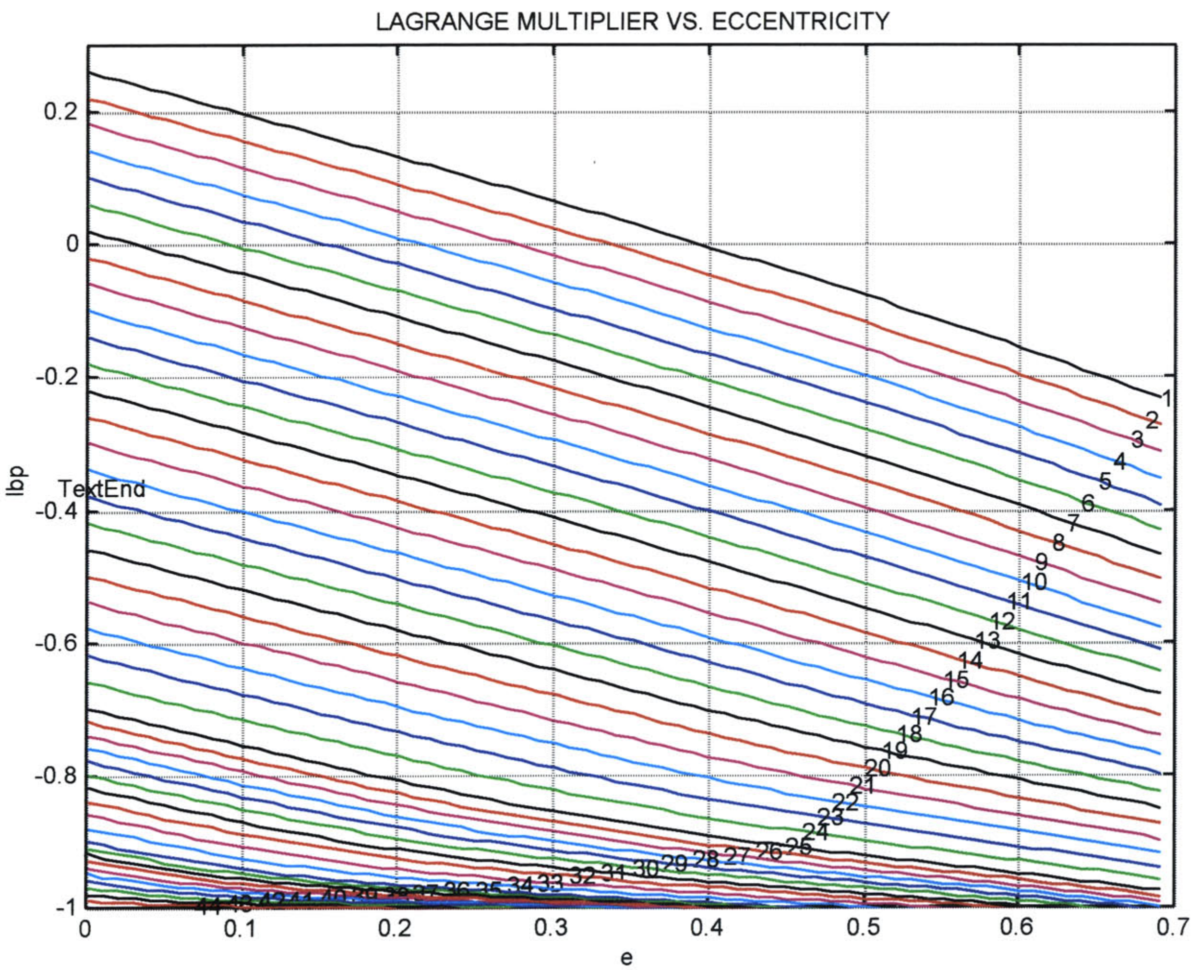


Figure 2: Variation of the multiplier along optimal orbits

Figure 3 shows the semi-major axis, normalized by the radius at GEO, for the same optimal orbits. The set of curves cover most of the likely initial values of the eccentricity and the semi-major axis, from pure circularization to almost pure climb.

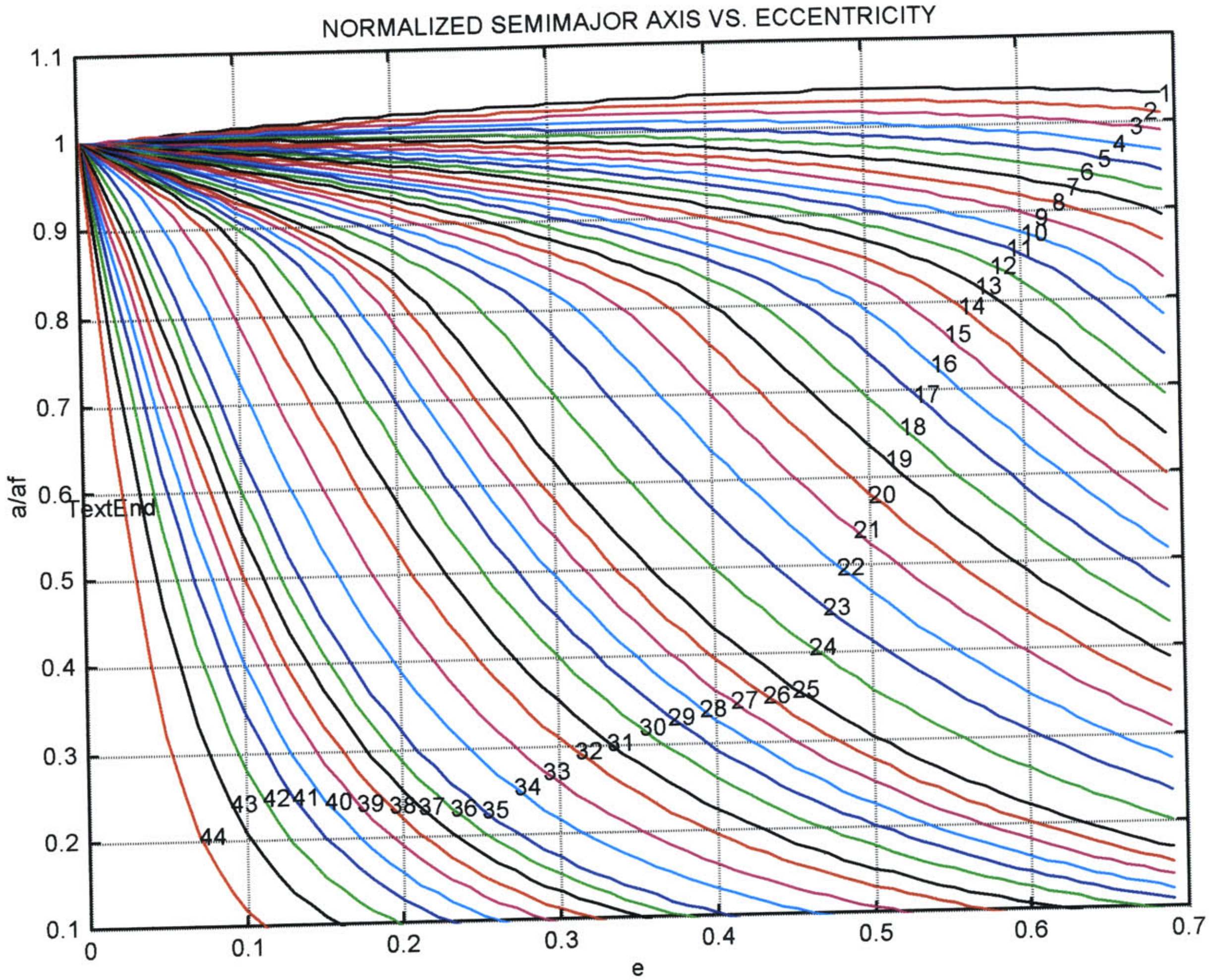


Figure 3: Variation of the semi-major axis along optimal trajectories

For the same trajectories, Figure 4 shows the velocity increments (normalized by the circular speed at GEO) required to get to GEO from each specified eccentricity, along each of the optimal trajectories:

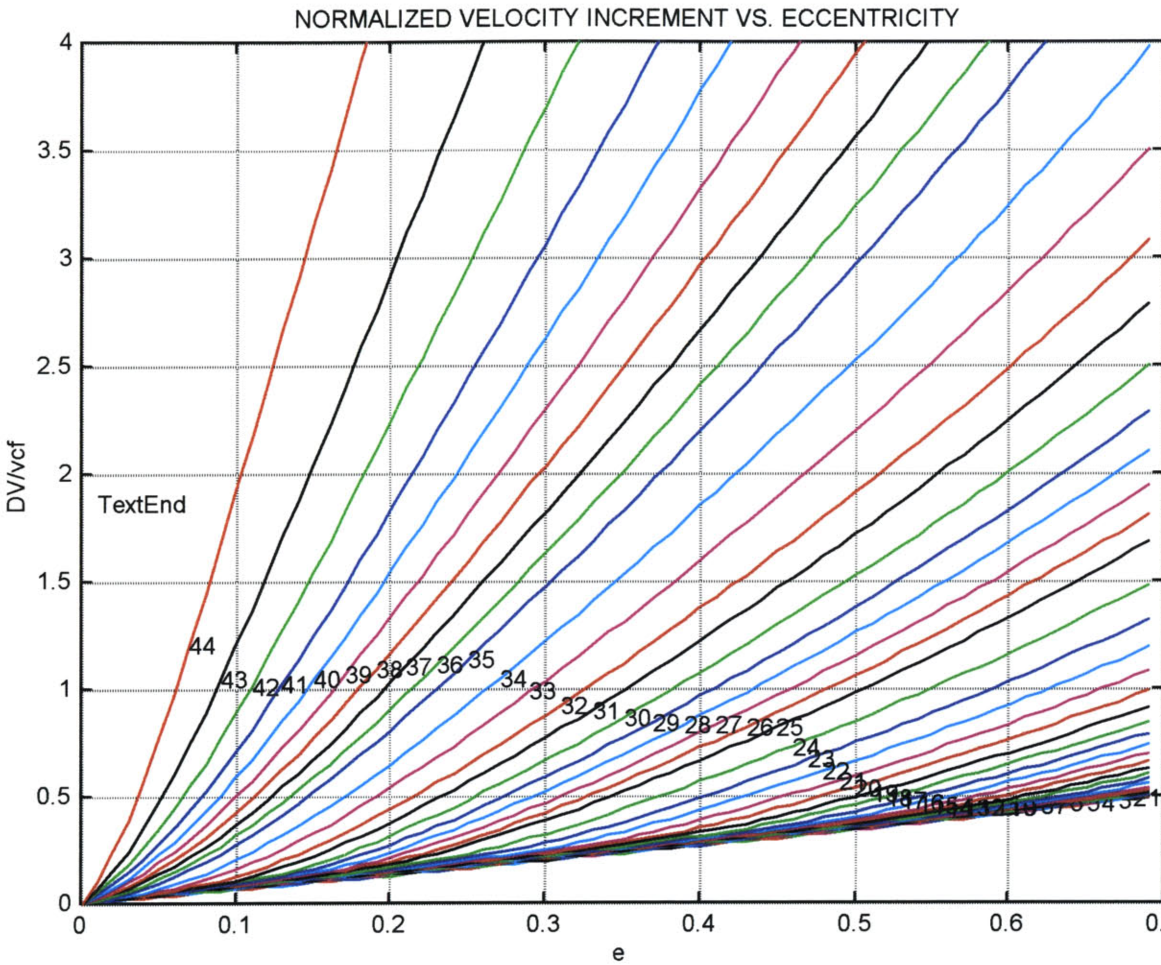


Figure 4: Minimum velocity increments to GEO (normalized by GEO circular velocity), from a given initial eccentricity, along various optimal trajectories.

The same information is presented in Figure 5 in the form of contour plots of velocity increment, in the plane of eccentricity and semi-major axis. This is probably all that is needed for mission optimization studies, but the information of the Lagrange multiplier, which is necessary to construct the thrust vector angles is then not directly available:

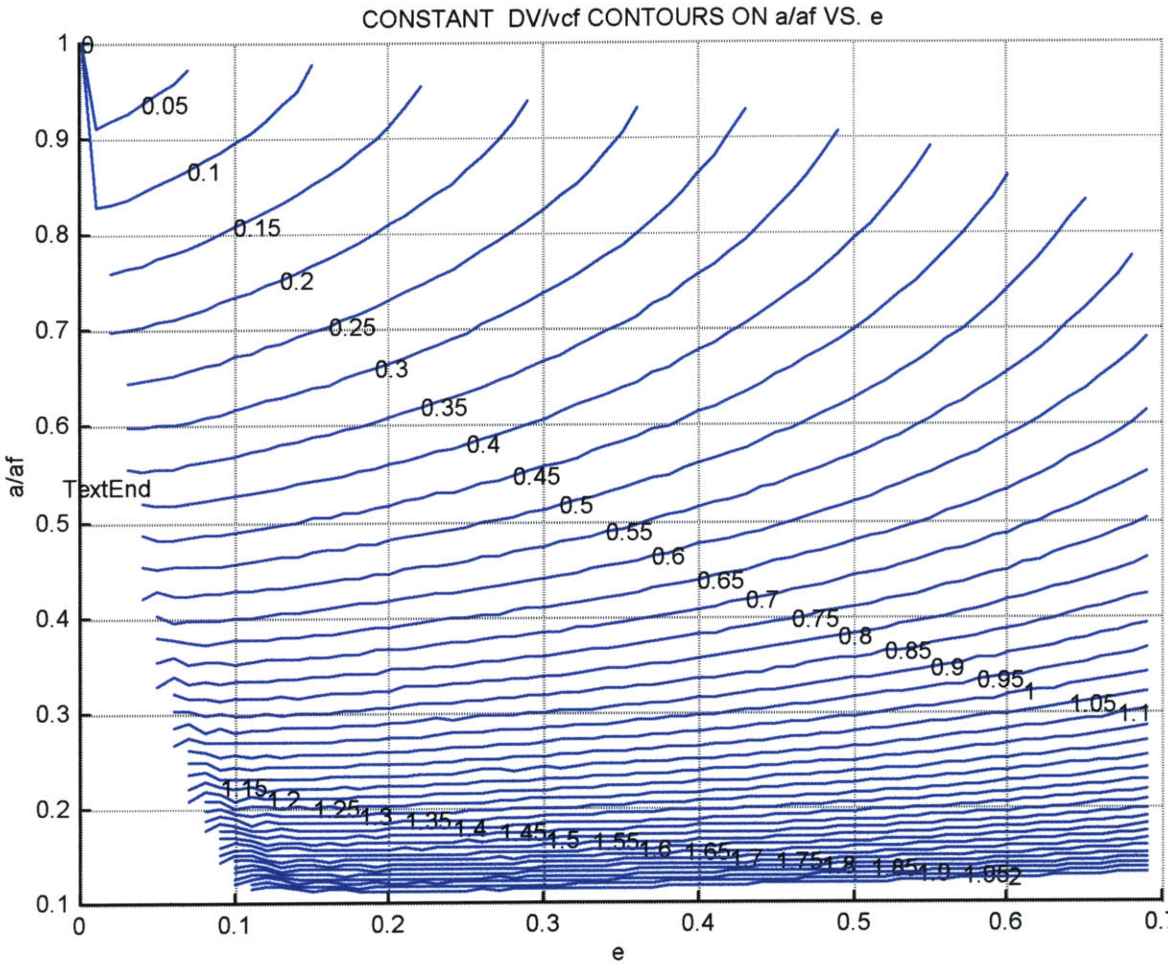


Figure 5: Lines of constant minimum velocity increments versus e and a/af

The nature of the optimized profiles of in-plane angle within each orbit is of interest as well. Figure 6 shows a number of such profiles at several points along a trajectory which starts at e=0.5 and a/af = 0.5. Near the beginning of the climb, the thrust is directed near the forward direction throughout the orbit, but starting at about e = 0.3, the β angle exceeds 90° (inwards) near the perigee passage. This amounts to "retro-firing" for that portion of the orbit. It seems to be the way to keep the apogee from growing too high following the forward firings at perigee in the initial parts of the trajectory.

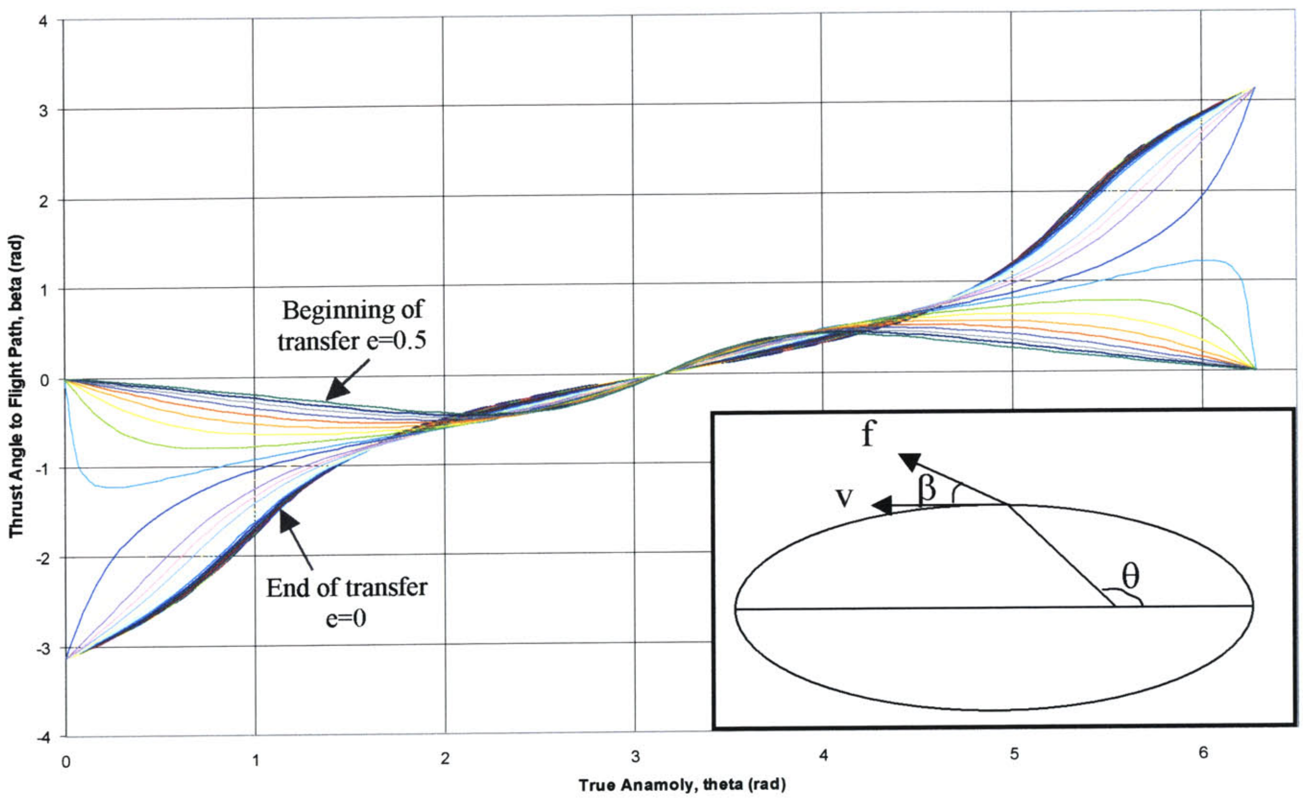


Figure 6: Thrust angle profiles at various points during an optimal trajectory (from $e=a/af$=0.5).

It is interesting to note that even with this partial retrofiring strategy, the apogee does climb temporarily above GEO attitude for this particular trajectory (Figure 7). However, the semi-major axis itself remains below GEO. The first part of the transfer focuses on raising the apogee and perigee. When the apogee pushes supersynchronous it becomes easier to circularize the orbit, so the transfer then focuses on circularization.

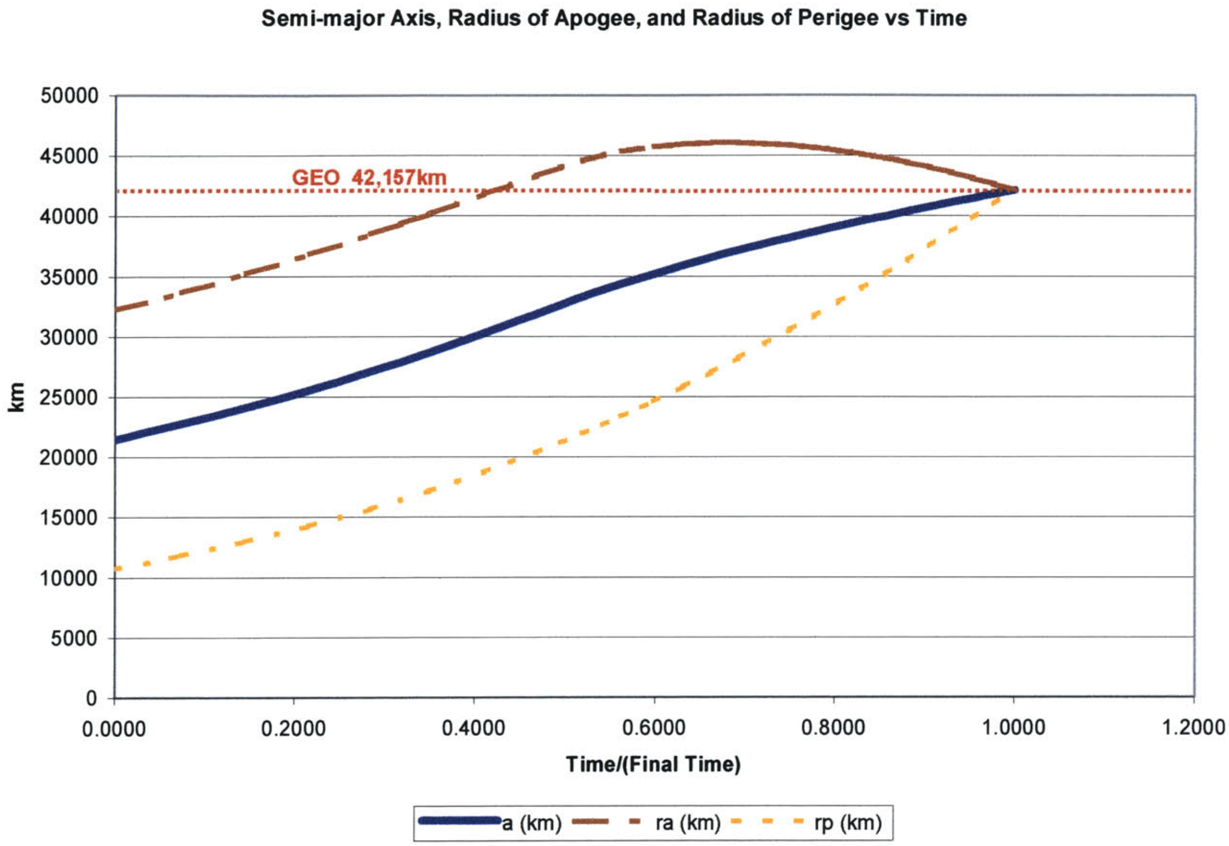


Figure 7: Semi-major Axis, Radius of Apogee and Perigee vs time/(final time) for the trajectory of Fig. 5.

2.2 Derivation of Restricted 3D EP Orbit Raising

2.2.1 Analysis

2.2.1.1 Introduction

The 2D analysis can be extended to include plane changes using the same techniques as in the previous section. However, we found that a cleaner derivation results if the differential equations for the inner optimization (over each orbit) use time instead of eccentricity as the independent variable. Using this method eliminates the need to iterate when solving for the integrals *C, M,* and *I*, which greatly speeds up the numerical process. The 2D analysis was kept as is since it is already very fast numerically.

In the full general 3D case, by including inclination (i) change we must also keep track of the argument of perigee (ω) and longitude of the ascending node (Ω). However, *if we assume two-body orbital mechanics and the argument of perigee is initially zero or π, then for the optimal trajectory it will remain equal to zero or π without any need to constrain it.* This leads to an easier problem to solve, and is the main assumption behind the "restricted" 3D case. This assumption occurs when the initial orbit has its line of apsides aligned with the line of nodes. This initial situation is true for most missions involving low-thrust orbit raising to GEO of an initially elliptic, inclined orbit, the normal mechanics of the chemical rocket delivery will place the initial apogee and perigee at the equatorial crossings. If launch is from the northern hemisphere, perigee will be at the descending node, and apogee at the ascending node. This means an initial condition $\omega(0) = \pi$ for the low-thrust segment, and for a southern hemisphere launch, $\omega(0) = 0$.

This feature of the optimal steering laws can be seen analytically by examining the full (unrestricted) 3D optimal steering laws (not derived in this thesis). However, it can also be seen logically, that if one assumes the node and apsides are initially aligned and it is only a two-body problem (no external forces to fight), then it would only waste fuel to rotate the argument of perigee and the longitude of the ascending node. If these conditions hold, we can then conclude $\frac{d\Omega}{de} = 0$ and $\frac{d\omega}{de} = 0$, and these equations can be eliminated from restricted 3D analysis. This would then result in optimal steering laws that focus only on plane change, circularization, and perigee raising. Earth oblateness effects (J2 and above) will modify this conclusion, but this restricted 3D case will be valuable as a conceptual stepping-stone to the more general situation, and will probably yield numerically accurate estimates of optimal velocity increments and steering laws.

Utilizing these simplifications, the restricted 3D problem has been numerically implemented to the point of obtaining full optimal trajectories by direct integration of the optimality differential equations. The formulation used for this restricted 3D case is slightly different than the last formulation of the 2D case, but it still reduces smoothly in the planar limit.

2.2.1.2 Basic Governing Equations and Orbit-Averaging

We start with the basic variation of parameters formulation of the equations of motion (Battin, pg. 489).

This provides equations for $\frac{da}{dt}, \frac{de}{dt}, and \frac{di}{dt}$, which contain all the classical elements, as well as the true anomaly θ. As mentioned in the introduction, $\frac{d\Omega}{dt} and \frac{d\omega}{dt}$ remain zero in the restricted 3D case. Within one orbit, the elements $(a, e, i, \Omega, \omega)$ vary by a negligible amount, while θ goes through the whole range $(0, 2\pi)$. So, within one orbit, we accept the Keplerian relationship:

$$\frac{d\theta}{dt} = \sqrt{\frac{\mu}{a^3(1-e^2)^3}}(1+e\cos\theta)^2 \tag{28}$$

as if the elements were truly frozen in time. We use (28) to eliminate time from the problem, by dividing each element rate equation through by (28). The resulting set of equations is listed below:

$$\frac{da}{d\theta} = f_o\varphi(\theta)\frac{2a^3(1-e^2)}{\mu}\frac{\sqrt{1+e^2+2e\cos\theta}}{(1+e\cos\theta)^2}\cos\beta\cos\alpha \tag{29}$$

$$\frac{de}{d\theta} = f_o\varphi(\theta)\frac{a^2(1-e^2)^2}{\mu}\frac{2(e+\cos\theta)\cos\beta + \frac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta}{(1+e\cos\theta)^2\sqrt{1+e^2+2e\cos\theta}}\cos\alpha \tag{30}$$

$$\frac{di}{d\theta} = f_o\varphi(\theta)\frac{a^2(1-e^2)^2}{\mu}\frac{\cos(\theta+\omega)}{(1+e\cos\theta)^3}\sin\alpha \tag{31}$$

Here $\alpha(\vartheta)$ is the angle of the thrust vector to the orbital plane, $\beta(\vartheta)$ is the in-plane angle of the thrust to the velocity vector (positive outwards), and $\varphi(\vartheta)$ is a thrust modulation factor, such that $f = f_o\ \varphi(\vartheta)$ within each orbit. For much of the treatment here we will force $\varphi = 1$, but the factor is introduced with a view to a more general treatment. Also, $\omega = \pi$ in the restricted 3D case for a northern hemisphere launch.

In addition, the cumulative "propulsive Δv", defined through $\frac{d\Delta v}{dt} = f$, yields

$$\frac{d\Delta v}{d\theta} = f_o\sqrt{\frac{a^3(1-e^2)^3}{\mu}}\frac{\varphi(\theta)}{(1+e\cos\theta)^2} \tag{32}$$

We are now in a position to perform the orbit averaging of these equations. The following integrals are defined:

$$C = \frac{1-e^2}{2\pi}\int_0^{2\pi} \varphi(\theta)\frac{\sqrt{1+e^2+2e\cos\theta}}{(1+e\cos\theta)^2}\cos\beta(\theta)\cos\alpha(\theta)d\theta \tag{33}$$

$$M = \frac{(1-e^2)^2}{2\pi}\int_0^{2\pi} \varphi(\theta)\frac{2(e+\cos\theta)\cos\beta(\vartheta)+\dfrac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{(1+e\cos\theta)^2\sqrt{1+e^2+2e\cos\theta}}\cos\alpha(\theta)d\theta \tag{34}$$

$$I = \frac{(1-e^2)^2}{2\pi}\int_0^{2\pi}\varphi(\theta)\frac{\cos(\theta+\omega)}{(1+e\cos\theta)^3}\sin\alpha(\theta)d\theta \tag{35}$$

(again $\omega = \pi$ for northern hemisphere launch)

$$V = \frac{(1-e^2)^{3/2}}{2\pi}\int_0^{2\pi}\frac{\varphi(\theta)d\theta}{(1+e\cos\theta)^2} \tag{36}$$

The parameter φ models the on (φ=1) and off (φ=0) function of the engine. In particular, if the thrust is kept constant, so that $\varphi \equiv 1$, Eq. (36) can be integrated exactly to yield V=1. Of course, the other integrals would still depend on the thrust angle profiles, which are yet to be specified. For the restricted 3D case, it is assumed that the engine is left on during the entire transfer (φ=1).

In terms of these integrals, the orbit-averaged long-term evolution equations are:

$$\left\langle \frac{da}{dt} \right\rangle_\theta = 2f_o\sqrt{\frac{a^3}{\mu}}C \tag{37}$$

$$\left\langle \frac{de}{dt} \right\rangle_\theta = f_o\sqrt{\frac{a}{\mu}}M \tag{38}$$

$$\left\langle \frac{di}{dt} \right\rangle_\theta = f_o\sqrt{\frac{a}{\mu}}I \tag{39}$$

$$\left\langle \frac{d\Delta v}{dt} \right\rangle_\theta = f_o V \tag{40}$$

From here on, we will omit the brackets in the understanding that the derivatives no longer contain intra-orbit variations, only those of a long-term nature.

2.2.1.3 The Intra-Orbit Optimization

If at any point during the mission we had an idea of the (long-term) desired rates of adjustment of the semi-major axis, eccentricity, and the inclination, the immediate task would be to schedule the thrust angles α and β versus true anomaly within the next orbit. We are thus led to a first level of optimization in which we seek to minimize the quantity $\frac{d\Delta V}{dt}$ under the constraints of imposed values of the quantities $\frac{da}{dt}$, $\frac{de}{dt}$, and $\frac{di}{dt}$ using Lagrange multipliers (Λ) to balance the effect of each constraint. The "control variables" we can manipulate to effect this optimization are the two thrust angles α and β, plus the thrust modulation fraction φ; these quantities are now to be regarded as adjustable functions of θ.

A cost function can be created by combining Lagrange multipliers (λ), the constraints, and the quantity to be minimized:

$$\Phi = \frac{d\Delta v}{dt} - \lambda_a \frac{da}{dt} + \lambda_e \frac{de}{dt} + \lambda_i \frac{di}{dt}$$

This can be simplified by substituting Eqs. (37) - (40) into the above and absorbing common factors into the Lagrange multipliers (now denoted Λ) since their sign and value are arbitrary.

$$\Phi = V - \Lambda_a C + \Lambda_e M + \Lambda_i I \tag{41}$$

Notice that if $\varphi = 1$, then V=1, so there is really no "Δv optimization" inside an orbit, only a proper relative apportioning of Δe, Δa, and Δi. It should also be noted that the solution of the 3D case will simply to the 2D case if $\Lambda_i = 0$, even though the derivations are slightly different.

Using optimal control theory as a guide, we can redefine the cost function (41) in terms of a Hamiltonian.

$$\Phi = \int_0^{2\pi} \mathrm{H}\, d\theta \tag{42}$$

$$\mathrm{H} = \frac{\varphi}{2\pi(1-e^2)}\left[\frac{1}{\sqrt{1-e^2}} - \frac{\Lambda_a\sqrt{1+e^2+2e\cos\theta}}{1-e^2}\cos\beta(\theta)\cos\alpha(\theta) + \Lambda_e \frac{2(e+\cos\theta)\cos\beta(\vartheta) + \frac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{\sqrt{1+e^2+2e\cos\theta}}\cos\alpha(\theta) + \Lambda_i \frac{\cos(\theta+\pi)}{1+e\cos\theta}\sin\alpha\right] \tag{43}$$

The derivatives of the Hamiltonian with respect to α, β and φ will be zero.

(44)

$$\frac{\partial \mathrm{H}}{\partial \alpha}=\frac{\Lambda_a\sqrt{1+e^2+2e\cos\theta}}{1-e^2}\cos\beta(\theta)\sin\alpha(\theta)-$$
$$\Lambda_e\frac{2(e+\cos\theta)\cos\beta(\vartheta)+\frac{(1-e^2)\sin\theta}{1+e\cos\theta}\sin\beta(\theta)}{\sqrt{1+e^2+2e\cos\theta}}\sin\alpha(\theta)-\Lambda_i\frac{\cos(\theta+\pi)}{1+e\cos\theta}\cos\alpha=0$$
$$\frac{\partial \mathrm{H}}{\partial \beta}=\frac{\Lambda_a\sqrt{1+e^2+2e\cos\theta}}{1-e^2}\sin\beta(\theta)\cos\alpha(\theta)-$$
$$\Lambda_e\frac{2(e+\cos\theta)\sin\beta(\vartheta)+\frac{(1-e^2)\sin\theta}{1+e\cos\theta}\cos\beta(\theta)}{\sqrt{1+e^2+2e\cos\theta}}\cos\alpha(\theta)=0 \tag{45}$$

$$\frac{\partial \mathrm{H}}{\partial \beta}=\frac{\mathrm{H}}{\varphi}=0 \tag{46}$$

If thrust is not to be switched on or off (φ=1), then Eq (46) is not necessary. The first observation is that φ(θ) appears as a linear factor in H, so that $\frac{\partial H}{\partial \varphi}=0$ implies $H=0$. Once φ(θ), β(θ) are specified, this can only occur <u>at discrete *θ* values</u> which satisfy *H(θ)* = 0. Any small perturbation δφ(θ) must be zero in between these values, but is arbitrary at them. This means these points are the <u>switching points</u> where <u>thrust discontinuities</u> can take place. However, since we are here insisting on φ≡1 throughout the orbit (constant thrust) we disregard this possibility and concentrate on α(θ), β(θ) instead.

We can solve (44) and (45) for α and β respectively. For simplification, we define the following combinations of parameters.

$$\begin{cases} x\equiv-\Lambda_a\frac{1+e^2+2e\cos\theta}{1-e^2}+2\Lambda_e(e+\cos\theta) \\ y\equiv\Lambda_e\frac{(1-e^2)\sin\theta}{1+e\cos\theta} \\ z\equiv\Lambda_i\frac{\cos\theta}{1+e\cos\theta}\sqrt{1+e^2+2e\cos\theta} \end{cases} \tag{47}$$

The in-plane thrust angle β is then found to be:

$\tan\beta = \frac{y}{x}$ or it can also be written as

$$\boxed{\begin{aligned} \sin\beta &= \frac{-y}{\sqrt{x^2+y^2}} \\ \cos\beta &= \frac{-x}{\sqrt{x^2+y^2}} \end{aligned}} \tag{48}$$

The sign of β must be specifically checked depending on the quadrant of θ. The following rules apply for the in-plane thrust angle β:

$$\boxed{\begin{aligned} &\text{If } \cos\beta > 0 \text{ in } 0 \le \theta \le \pi, \\ &\quad \text{Then } \beta = \sin^{-1}(\sin\beta), \\ &\text{Else } \beta = -\pi - \sin^{-1}(\sin\beta) \end{aligned}} \tag{49}$$

For $\pi \le \theta \le 2\pi$, β is negative antisymmetric to the β values in the first half of the orbit.

The out-of-plane thrust angle α is:

$\tan\alpha = \frac{-z}{x\cos\beta + y\sin\beta}$ or it can also be written as

$$\boxed{\begin{aligned} \sin\alpha &= \frac{z}{\sqrt{x^2+y^2+z^2}} \\ \cos\alpha &= \frac{\sqrt{x^2+y^2}}{\sqrt{x^2+y^2+z^2}} \end{aligned}} \tag{50}$$

Since the sign of α depends on the sign of β, there is no need to check the sign of α if one of the equations in (50) is used. The values of α in the second half of the orbit ($\pi \le \theta \le 2\pi$) are symmetric with the α values in the first half of the orbit ($0 \le \theta \le \pi$).

These profiles depend parametrically upon the multipliers Λ_a, Λ_e, Λ_i, whose slow time evolution is yet to be determined. We note in passing that these optimized functional forms for α and β could be used for a direct search algorithm that would then search for the best λ time profiles. This is presumably more accurate than the linear superposition of sub-optimal θ profiles, as in the methods developed by Kluever-Oleson or Ilgen.

An important observation from (47) - (50) is that the angle sines and cosines are homogeneous functions of degree zero of the parameters (Λ_a, Λ_e, Λ_i). This means that only the ratios of these multipliers matter. For problems with some finite plane change, we will find that Λ_i never crosses zero, and so we will use the two parameters

$$m_a = \frac{\lambda_e}{\lambda_i} \quad ; \quad m_e = \frac{\lambda_e}{\lambda_i} \tag{51}$$

Once the forms of $\alpha(\theta)$, $\beta(\theta)$ are known (Eqs. (47)-(50)), the integrals *C, M, I* can be computed for each choice of the Λ's and of the eccentricity. Thus, *C, M, I* = functions of (e, m_a, m_e).

2.2.1.4 Calculation of the Integrals

Going back to Eqs. (33) to (36), we can see that the integrals *C, M,* and *I* are explicitly functions of e, ω, and of the profiles $\alpha(\theta)$ and $\beta(\theta)$. Since the integrals in this derivation do *not* depend explicitly on themselves, iteration is *not* needed for the computations, as was the case in the 2D analysis in 2.1.1. (Note that the same derivation could have been done in the 2D case). The integral calculation is straightforward:

(a) Specify e, i, m_a, and m_e
(b) Calculate β and check its sign, then calculate α
(c) Substitute all of above into *C, M* and *I.*
(d) Use any numerical quadrature formula to compute the integrals.

In the end, as the procedure implies, each integral ends up being a function of the variables e, i, m_a, and m_e.

2.2.1.5 The Outer Optimization for the Restricted 3D Case

For a given engine and spacecraft mass, minimizing the fuel consumption also minimizes the operating time, and so the total time T is not known a-priori. Since, in addition, time does not appear explicitly in any of the equations, it is advantageous to change the dependent variable to some monotonically varying dynamic quantity. For 3-D problems, there is no incentive to reverse the direction of the orbital plane rotation, and so the inclination i is a suitable independent variable.

We now tackle the problem of finding the optimal long-term variation with inclination of the semi-major axis, eccentricity and the two corresponding multipliers.

Using inclination i as the independent slow variable, we first divide Eqs. (37),(38), and (40), by Eq. (39):

$$\frac{da}{di} = 2a\frac{C}{I} \tag{52}$$

$$\frac{de}{di} = \frac{M}{I} \tag{53}$$

$$\frac{d\Delta V}{di} = -\sqrt{\frac{\mu}{a}}\frac{V}{I} \tag{54}$$

and we then minimize

$$\Delta V = \int_{i_i}^{i_f = o} \sqrt{\frac{\mu}{a}}\frac{V}{I}\,di \tag{55}$$

subject to satisfaction of (52) and (53). When perturbing (55), the variations $\delta a, \delta e, \delta m_a$ and δm_e will appear, the last three arising from the integrals V and I. However, δm_a and δm_e can be extracted from the perturbed forms of Eqs. (52)-(53), as linear functions of $\delta a, \delta e, \frac{d\delta a}{dt}$ and $\frac{d\delta e}{dt}$. The derivative terms which result in the integrand of (55) are handled through an integration by parts, with the integrated terms vanishing. After this, the integrand for $\delta\Delta V$ is of the form $[\cdots]\delta a + [\cdots]\delta e$, and we obtain the two optimality differential equations by equating the brackets to zero:

$$\frac{d}{di}\left(\frac{1}{2}\sqrt{\frac{\mu}{a^3}}F\right) + \sqrt{\frac{\mu}{a^3}}\left(\frac{1}{2}\frac{V}{I} + F\frac{C}{I}\right) = 0 \tag{56}$$

$$\frac{d}{di}\left(\sqrt{\frac{\mu}{a}}G\right) + \sqrt{\frac{\mu}{a}}\left(-\frac{\partial V/I}{\partial e} + F\frac{\partial C/I}{\partial e} + G\frac{\partial M/I}{\partial e}\right) = o \tag{57}$$

where

$$F = \frac{J_{VM}}{J_{CM}} \quad ; \quad G = \frac{J_{CV}}{J_{CM}} \tag{58}$$

$$J_{VM} = \frac{\partial(V/I, M/I)}{\partial(m_a, m_e)} \quad ; \; J_{CM} \frac{\partial(C/I, M/I)}{\partial(m_a, m_e)} \quad ; \; J_{CV} = \frac{\partial(C/I, V/I)}{\partial(m_a, m_e)} \tag{59}$$

As before, Eqs. (56), (57) must be expanded in order to isolate the derivatives of m_a and m_e. The results are

$$\frac{dm_a}{di} = \frac{Q_a \frac{\partial G}{\partial m_e} - Q_e \frac{\partial F}{\partial m_e}}{J_{FG}} \tag{60}$$

$$\frac{dm_e}{de} = \frac{Q_e \frac{\partial F}{\partial m_a} - Q_a \frac{\partial G}{\partial m_a}}{J_{FG}} \tag{61}$$

where $$Q_a = -\frac{V}{I} + F\frac{C}{I} - \frac{M}{I}\frac{\partial F}{\partial e} \tag{62}$$

$$Q_e = \frac{C}{I} - \frac{M}{I}\frac{\partial G}{\partial e} + \frac{\partial V/I}{\partial e} - F\frac{\partial C/I}{\partial e} - G\frac{\partial M/I}{\partial e} \tag{63}$$

$$J_{FG} = \frac{\partial(F,G)}{\partial(m_a, m_e)} \tag{64}$$

Equations (61) and (62) must be integrated from assumed initial values (at GEO) $m_a(i=0)=m_{ao}$, $m_e(i=o)=m_{eo}$, simultaneous by with Eqs. (52)-(54). The various partial derivatives $\frac{\partial}{\partial e}, \frac{\partial}{\partial m_a}, \frac{\partial}{\partial m_e}$ are evaluated numerically using central differencing, while the integrations for C, I, M, V are done using a trapezoidal scheme with 50 steps per half-orbit (symmetry allows the integrations to go from $\theta = 0$ to $\theta = \pi$ only). No anomalies or singularities are encountered. For reference, one full optimal 3D trajectory, with 100 time steps, is computed in about 1-2 sec. by a 1.4 GHz PC computer, using standard Matlab code.

2.2.1.6 Thrust Level, Mission Time and Spacecraft Mass

One of the consequences of the elimination of time from the formulation has been the simultaneous elimination of f_o, the scale for thrust acceleration, and indirectly for the specific power available. Thus, while the specific impulse c is assumed constant, the thrust F and the power P may vary on the slow time scale without affecting our results so far. From the engine power equation,

$$f = f_o = \frac{F}{m} = \frac{2\eta P}{cm} \tag{65}$$

and f_o might vary (on the slow time scale) due to any combination of mass change $m(t)$ and power variations $P(t)$. The simplest case occurs when $P = const.$ as well as c. This implies a constant thrust $F = \frac{2\eta P}{c}$ and flow rate $\dot{m} = \frac{2\eta P}{c^2}$. The mass at any time is related to the remaining Δv to be accomplished through the rocket equation

$$\frac{m}{m_o} = \exp\left[-\frac{\Delta v_{TOT} - \Delta v}{c}\right] \tag{66}$$

where Δv_{TOT} corresponds to the initial point of the trajectory. Since our optimization has yielded $\Delta v(i)$, the mass $m(i)$ corresponding to a particular inclination i can now be calculated. Following this, the elapsed time for that same condition is simply

$$t = \frac{m_o - m}{\dot{m}} = \frac{m_o c^2}{2\eta P}\left(1 - e^{-\frac{\Delta v_{TOT} - \Delta v}{c}}\right) \tag{67}$$

and, once again $t(i)$ results. In particular, the low-thrust total time T corresponds to $\Delta v = 0$:

$$T = \frac{m_o c^2}{2\eta P}\left(1 - e^{-\frac{\Delta v_{TOT}}{c}}\right) \tag{68}$$

which is, as expected, inversely proportional to the available specific power P/m_o.

2.2.2 3D Constant Thrust Results

The coding of the 3D restricted case has been accomplished and some results are shown below. Figure 8 to Figure 11 are examples of optimized mission ΔV values for initial inclination of 0.1, 0.2, 0.4 and 0.6 radians, respectively. Similar plots can be automatically generated for other initial inclinations, or a robust search routine can be used to zero in on selected initial conditions (i_i, a_i, e_i). It can be seen that at lower inclinations (i=0.1 initially), the 3D contour plot in Figure 8 is very similar to the 2D contour plot in Figure 5. Although the plots are generated in completely different ways, the 3D results still converge to the 2D cases as initial inclination goes to zero. It is of interest in Figure 11 that, for missions starting from low-energy highly inclined orbits, the required ΔV is nearly independent of the initial eccentricity. In fact, for synchronous or near-synchronous starting conditions ($a_i \sim 1$), ΔV decreases with initial eccentricity. This appears to be due to the fact that efficient plane change can be accomplished by out-of-plane thrusting near the apogee of a highly eccentric orbit. This trend is reversed, however, for very large eccentricities, where the circularization cost dominates over the plane change cost.

One feature of interest for missions with relatively large initial inclination is the possibility that optimality will call for an initial increase of the eccentricity, so as to raise the apogee and thereby reduce the fuel cost of the plane change. Figure 12 shows the evolution of the various parameters for an optimum trajectory from $i_i = -0.7 rad$, $e_i = 0.23$, $a_i/a_f = 0.18$. The first half of the mission (between is = - 0.7 rad and i = - 0.35 rad) is seen to feature an increase of e from 0.23 to 0.30; the orbit is then circularized while the remaining inclination is removed. The intra-orbit optimal profiles of the thrust angles are displayed in Figure 13 and Figure 14 for a few intermediate times. The out-of-plane angle $\alpha(\theta)$ (Figure 13) is large positive (near $\pi/2$ rad) around perigee, and large negative, although somewhat less, near apogee; it is to be recalled that these are also the nodal points, so this is where the out-of-plane thrust component is most effective in rotating the orbit. The in-plane angle $\beta(\theta)$ (Figure 14) has a similar behavior to that seen for 2D orbits in Figure 6 mostly forward thrusting initially ($0.7 < i \leq - 0.15$ rad), but thrust reversals near perigee the rest of the way.

Our work so far has produced complete solutions to the unconstrained EOR problems in either two dimensions, or in three dimensions, if the initial line of apses is aligned with the line of nodes, and no disturbances to these lines are included. The solutions include a rigorous derivation of the intra-orbit variations of the thrust angles, and they are obtained through a robust direct integration process, which is both fast and general enough to provide synoptic information on the nature of the optimal solutions. We have also verified that the 3D results do reduce to the 2D case when the multiplier Λ_i is set to zero. Further work is needed to (a) Provide a more complete outer shell which will solve for transfers with specific initial conditions or easily create contour graphs, (b) Code the general 3D algorithm, (c) Explore with a specialized orbit propagator the constraint violations incurred by the unconstrained codes, and (d) Devise ways to incorporate the important constraints into the optimization.

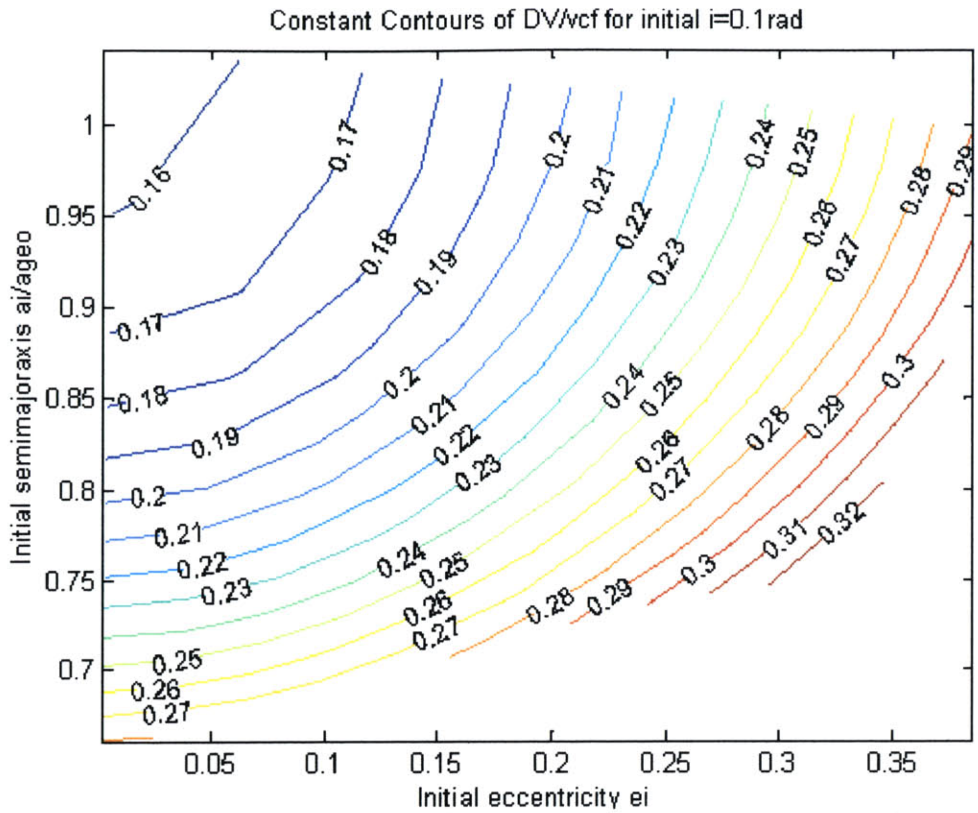


Figure 8: Constant DV/vcf contours for low inclination transfers

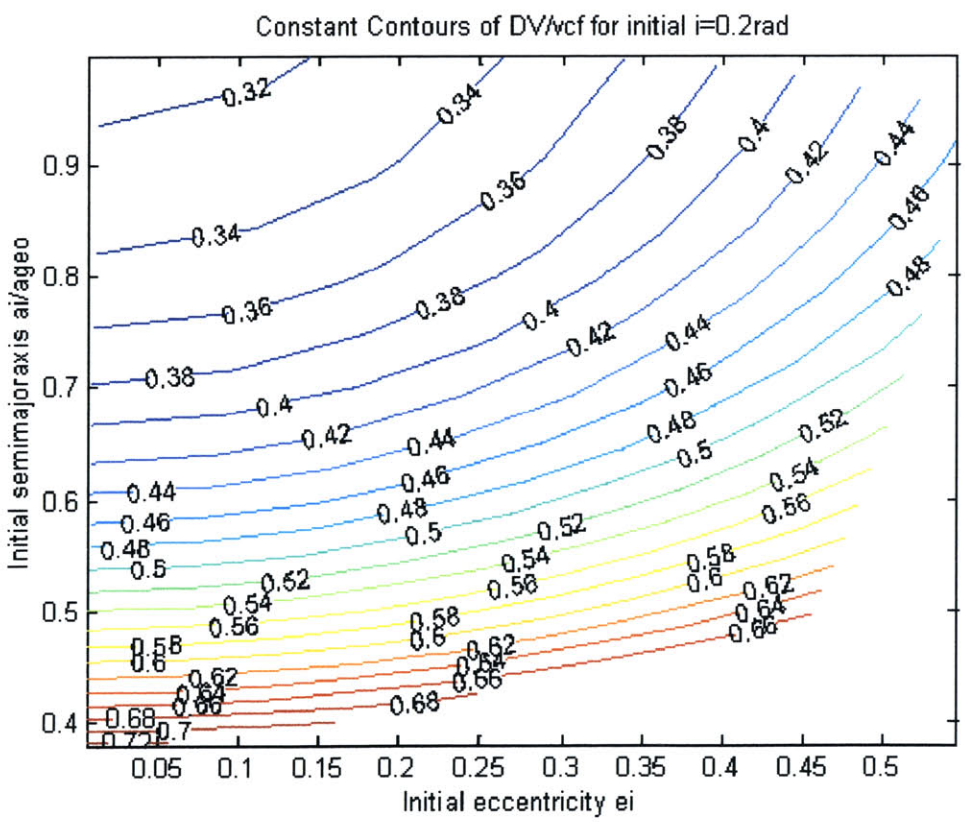


Figure 9: Constant DV/vcf contours for medium low inclination transfers

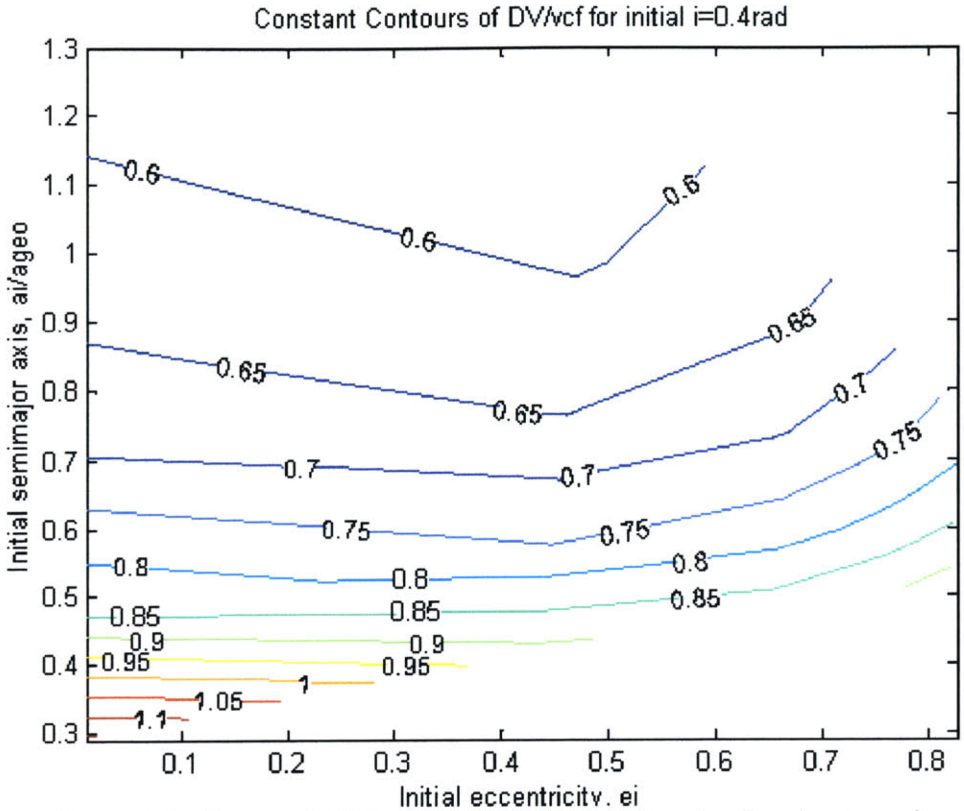


Figure 10: Constant DV/vcf contours for medium inclination transfers

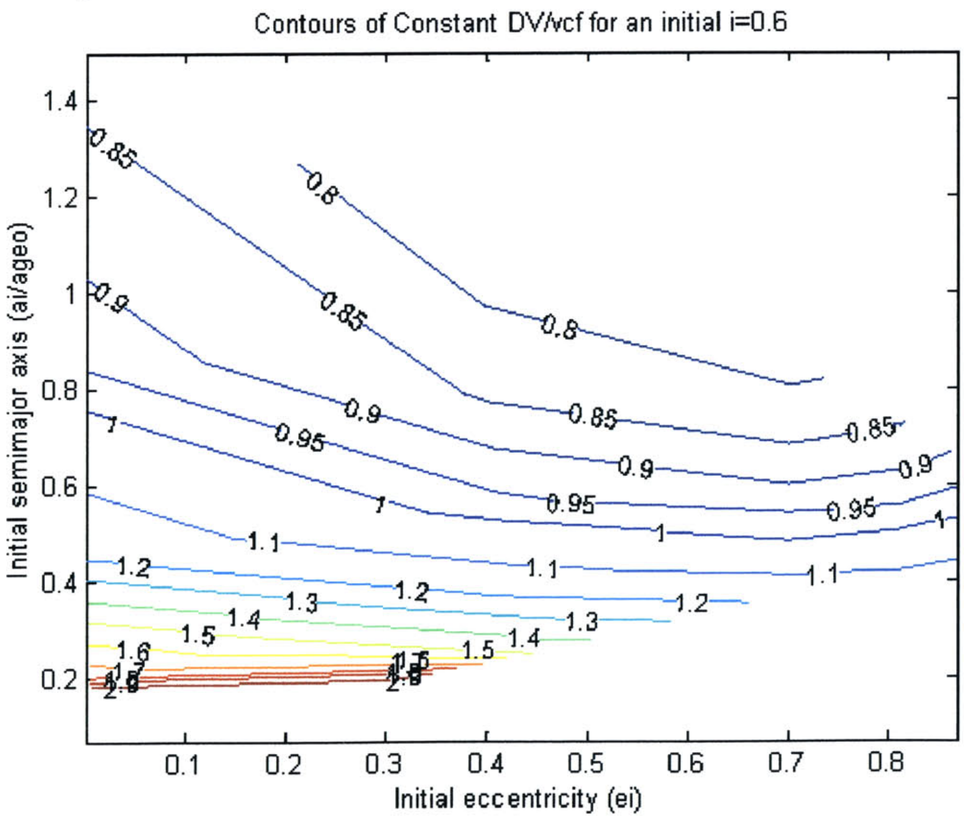


Figure 11: Constant DV/vcf contours for medium high inclination transfers

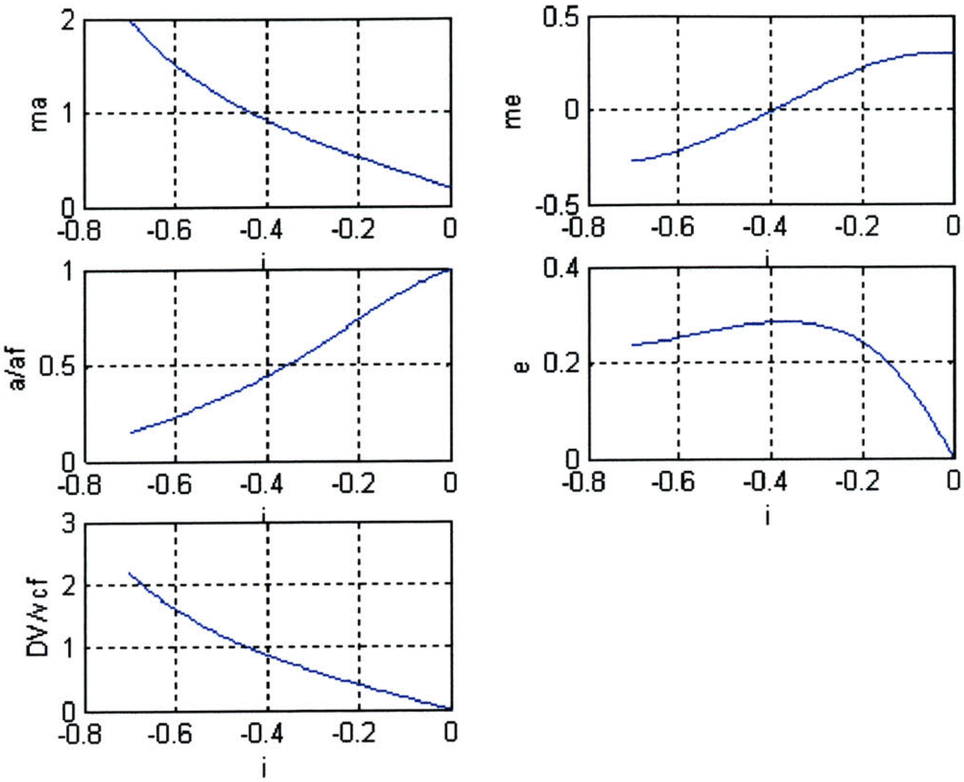


Figure 12: Example transfer starting at $i_i = -0.7rad$, $e_i = 0.23$, $a_i/a_f = 0.18$

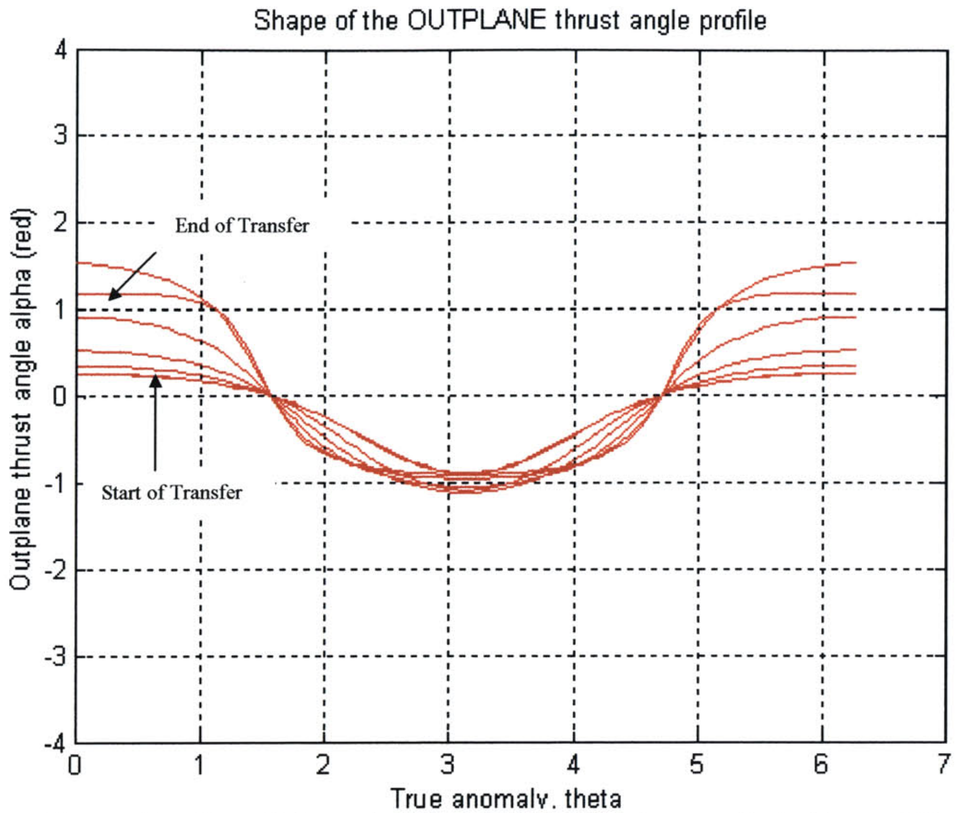


Figure 13: Out of plane thrust angles (α) for the transfer in Figure 12

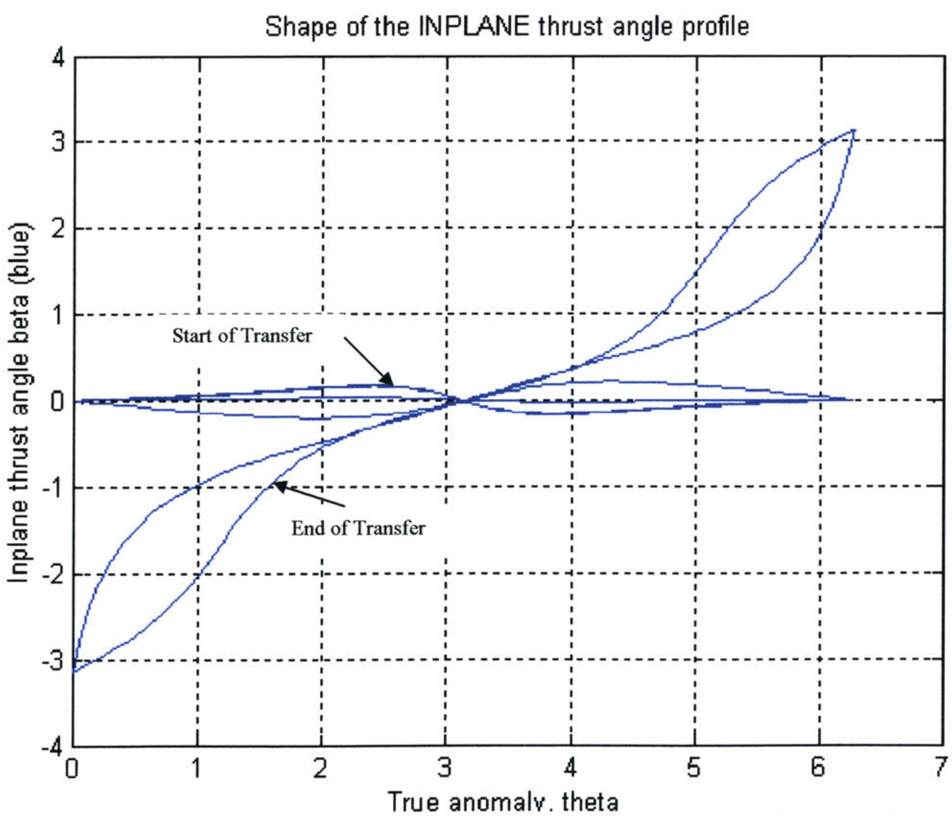


Figure 14: In plane thrust angles (β) for the transfer in Figure 12

2.3 Derivation of 2D Variable Thrust EP Orbit Raising

2.3.1 Analysis

2.3.1.1 Introduction

The effectiveness of thrust application varies with location within an elliptical orbit, and this makes it plausible that a control strategy in which thrust is allowed to vary in magnitude within each orbit will prove superior to one where thrust is kept constant. On the other hand, it seems obvious that continuous use of maximum available power will always be advantageous. This implies that specific impulse will be allowed to vary inversely with thrust, since

$$P = \frac{1}{2\eta} F c \tag{69}$$

at all times. In addition, if the object is to reduce eccentricity as well as increase energy, the angle β between the thrust vector and the velocity vector will also vary in some optimal manner within each orbit. And, of course, these variation laws will gradually change in time as the orbit evolves under the applied low thrust.

If the thrust degree of freedom is introduced, the mission time T must be explicitly constrained, because otherwise optimization would simply yield impulsive thrust applications at the best points within each orbit, and, with power limited, these would be infinitesimally small impulse bits, leading to an infinitely long mission. At the same time, the traditional measure of goodness in orbit optimization, which is the velocity increment $\Delta V = \int_0^T (F/m)dt$ is no longer significant, because it does not relate directly to fuel use when F varies with time. This follows from

$$m\frac{dv}{dt} = F = -c(t)\frac{dm}{dt}$$

or
$$\Delta v = \int_0^T c(t)\frac{d\ell n(m_o / m)}{dt}dt \tag{70}$$

which does not integrate directly as in the case when c is constant. This means that minimum Δv might not imply minimum fuel use, and a more direct approach is required.

This alternative approach is provided by the relationship

$$\frac{d\left(\frac{m_o}{m}\right)}{dt} = -\frac{m_o}{m^2}\frac{dm}{dt} = \frac{m_o \dot{m}}{m^2} = \frac{m_o\left(\dot{m}^2 c^2\right)}{\left(\dot{m}c^2\right)m^2}$$

and since $P = \frac{1}{2\eta}\dot{m}c^2$ and $F = \dot{m}\, c$

$$\frac{d\left(\frac{m_o}{m}\right)}{dt} = \frac{m_o f^2}{2\eta P} \tag{71}$$

where f=F/m. Integrating,

$$\frac{m_o}{m_f} - 1 = \frac{m_o}{2\eta P}\int_o^T f^2 dt \tag{72}$$

where the power and the efficiency have been assumed constant. For minimum fuel use ($m_{prop.} = m_o - m_f$), the integral in is to be minimized, consistent with the given initial and final conditions and with the orbital dynamics. Notice that this has the effect of replacing a metric $\int_o^T f dt$ which is linear in f by one which is quadratic in f (72). It is known from optimization theory that when an allowable control variable (f in this case) appears linearly in the cost functional, the optimum trajectories will in general have an on-off character, with switch points dictated by the optimization. Thus, the minimum Δv formulation with "free" thrust and constrained time will generate coasting periods, but, as noted, may not be mass optimal.

2.3.1.2 The Intra-Orbit Optimization

We start from the perturbation equations for the orbital elements *a(t)* and *e(t)* with the temporary assumption that the argument of perigee will remain constant for the optimal trajectories (this turns out to be true due to the symmetry of the thrust and thrust angle distributions within each orbit). No plane change is considered.

It is easily shown that, after orbit-averaging the standard perturbation equations, one obtains the "long-term", or secular rates

$$\left\langle \frac{da}{dt} \right\rangle = 2 f_o \sqrt{\frac{a^3}{\mu}} C \tag{73}$$

$$\left\langle \frac{de}{dt} \right\rangle = f_o \sqrt{\frac{a}{\mu}} M \tag{74}$$

$$\frac{d\left(\frac{mo}{m}\right)}{dt} = \frac{m_o f_o^2}{2\eta P} V_2 \tag{75}$$

where f_o is some reference thrust acceleration $(f \equiv f_o \varphi(t))$, to be identified later, and the quantities C, M, V_2 are given by

$$C=\frac{1-e^2}{2\pi}\int_0^{2\pi}\varphi(\vartheta)\frac{\sqrt{1+e^2+2e\cos\vartheta}}{(1+e\cos\vartheta)^2}\cos\beta(\vartheta)d\vartheta \tag{76}$$

$$M=\frac{\left(1-e^2\right)^2}{2\pi}\int_0^{2\pi}\varphi(\vartheta)\frac{2(e+\cos\vartheta)\cos\beta(\vartheta)+\dfrac{\left(1-e^2\right)\sin\vartheta}{1+e\cos\vartheta}\sin\beta(\vartheta)}{(1+e\cos\vartheta)^2\sqrt{1+e^2+2e\cos\vartheta}}d\vartheta \tag{77}$$

$$V_2=\frac{\left(1-e^2\right)^{3/2}}{2\pi}\int_0^{2\pi}\frac{\varphi^2(\vartheta)}{(1+e\cos\vartheta)^2}d\vartheta \tag{78}$$

The integrals C and M are in fact the same as in our previous constant-thrust analysis, while V_2 replaces a similar quantity V arising from the previously used $d(\Delta v)/dt$ equation.

Since the final time T is prescribed, it is in this case advantageous to retain time (t) as the independent variable. We now formulate the optimization problem as minimizing $\left\langle \frac{d(m_o/m)}{dt}\right\rangle$, subject to temporarily prescribed values of $\left\langle\frac{da}{dt}\right\rangle$ and $\left\langle\frac{de}{dt}\right\rangle$. For this purpose, using Eqs. (73)-(75), we introduce the augmented cost function

$$\Phi=\frac{m_o f_o^2}{2\eta P}\left(V_2-\Lambda_a C-\Lambda_e M\right) \tag{79}$$

where $\Lambda_a(t)$ and $\Lambda_e(t)$ are slowly varying Lagrange multipliers (non-dimensional). When small variations $\delta\beta(\vartheta)$, $\delta\varphi(\vartheta)$ are introduced for β and φ within each orbit, the integrals will vary by δC, δM, δV_2, such that, for optimality

$$\delta V_2-\Lambda_a\delta C-\Lambda_e\delta M=0 \tag{80}$$

The variations δC, δM, δV_2 can be explicitly calculated by varying the integrals in Eqs. (76)-(78), while keeping the slowly varying quantities a and e constant (as well as θ, the dummy variable of integration). The results can be grouped into a single integral equation of the form

$$\frac{1}{2\pi}\int_0^{2\pi}\left\{[\cdots]\delta\beta(\vartheta)+[\cdots\cdots]\delta\varphi(\vartheta)\right\}d\vartheta=0$$

and for optimality, the coefficients of $\delta\beta$ and $\delta\varphi$ must be zero for all ϑ. Equating to zero the coefficient of $\delta\beta$ leads, after rearrangement, to an expression for the thrust angle $\beta(\vartheta)$.

$$\boxed{\tan\beta = \frac{\left(1-e^2\right)\dfrac{\sin\vartheta}{1+e\cos\theta}}{\dfrac{\Lambda_a}{\left(1-e^2\right)\Lambda_e}\left(1+e^2+2e\cos\vartheta\right)+2\left(e+\cos\vartheta\right)}} \tag{81}$$

Similarly, equating to zero the coefficient of $\delta\varphi$ yields an expression for the normalized thrust acceleration $\varphi(\vartheta)$

$$\boxed{\varphi = -\frac{\Lambda_e}{2}\sqrt{\frac{1-e^2}{1+e^2+2e\cos\vartheta}}\sqrt{\left[\frac{\Lambda_a}{\left(1-e^2\right)\Lambda_e}\left(1+e^2+2e\cos\vartheta\right)\right]^2+\left[\left(1-e^2\right)\frac{\sin\vartheta}{1+e\cos\vartheta}\right]^2}} \tag{82}$$

In these formulae, the eccentricity e and the Lagrange multipliers Λ_a, Λ_e are slowly varying functions of time, to be regarded as constants over one orbit.

The expressions for $\beta(\vartheta)$ and $\varphi(\vartheta)$ can now be substituted back into the definitions (76)-(78) to calculate the integrals C, M and V_2. The form of Eqs. (81) and (82) is such that substantial simplifications occur when calculating $\varphi\cos\beta$, $\varphi\sin\beta$ and φ^2; in particular all square roots disappear. After some rearrangement, we obtain

$$C = \frac{1}{2}\sqrt{1-e^2}\,\Lambda_a I_1(e) + \left(1-e^2\right)^{3/2}\Lambda_e I_2(e) \tag{83}$$

$$M = \left(1-e^2\right)^{3/2}\Lambda_a I_2(e) + 2\left(1-e^2\right)^{5/2}\Lambda_e I_3(e) + \frac{1}{2}\left(1-e^2\right)^{9/2}\Lambda_e I_4(e) \tag{84}$$

$$V_2 = \frac{\left(1-e^2\right)^{1/2}}{4}\left[\Lambda_a^2 I_1(e) + 4\left(1-e^2\right)\Lambda_a\Lambda_e I_2(e) + 4\left(1-e^2\right)^2\Lambda_e^2 I_3(e) + \left(1-e^2\right)^4\Lambda_e^2 I_4(e)\right] \tag{85}$$

where the integrals I_1 to I_4 are

$$I_1 = \frac{1}{\pi}\int_0^{\pi}\frac{1+e^2+2e\cos\vartheta}{(1+e\cos\vartheta)^2}d\vartheta \tag{86}$$

$$I_2 = \frac{1}{\pi}\int_0^{\pi}\frac{e+\cos\vartheta}{(1+e\cos\vartheta)^2}d\vartheta \tag{87}$$

$$I_3 = \frac{1}{\pi}\int_0^{\pi}\frac{(e+\cos\vartheta)^2\,d\vartheta}{(1+e\cos\vartheta)^2\left(1+e^2+2e\cos\vartheta\right)} \tag{88}$$

$$I_4 = \frac{1}{\pi}\int_0^{\pi}\frac{\sin^2\vartheta\,d\vartheta}{(1+e\cos\vartheta)^4\left(1+e^2+2e\cos\vartheta\right)} \tag{89}$$

These integrals are all calculable by standard (although tedious), analytical methods. The results are

$$I_1 = \frac{1}{\sqrt{1-e^2}} \tag{90}$$

$$I_2 = 0 \tag{91}$$

$$I_3 = \frac{1}{e^2}\left(\frac{1}{1-e^2} - \frac{1}{\sqrt{1-e^2}}\right) \tag{92}$$

$$I_4 = \frac{1}{\left(1-e^2\right)^{7/2}}\left(4\frac{1-\sqrt{1-e^2}}{e^2} - \frac{3}{2}\right) \tag{93}$$

Substitution of (90)-(93) into (82)-(85) then yields,

$$\boxed{C = \frac{1}{2}\Lambda_a} \tag{94}$$

$$\boxed{M = \frac{5}{4}\left(1-e^2\right)\Lambda_e} \tag{95}$$

$$\boxed{V_2 = \frac{1}{4}\left(\Lambda_a^2 + \frac{5}{2}\left(1-e^2\right)\Lambda_e^2\right)} \tag{96}$$

The simplicity of these results is to be remarked. The C and M integrals, which control $\frac{da}{dt}$ and $\frac{de}{dt}$, are respectively proportional to the multipliers Λ_a, Λ_e. While the V_2 integral, which controls $\frac{d(m_o/m)}{dt}$ and hence plays the role of a cost function for the long-term optimization, is quadratic in the multipliers. In fact, it can be verified from (94)-(96) that

$$\frac{\partial V_2}{\partial \Lambda_a} = C \quad and \quad \frac{\partial V_2}{\partial \Lambda_e} = M \tag{97}$$

Eq. (97), together with Eqs. (73) and (74), show that the multipliers actually play the mathematical role of the generalized momentums associated with the generalized coordinates a and e, in a Hamiltonian dynamics formulation, where the Hamiltonian is proportional to V_2. This is at the root of further simplifications which appear in the "long-term", or "outer" optimization.

2.3.1.3 The Long-Term Optimization

We now return to Eq. (75) and integrate from $t = 0$ to $t = T$, the prescribed final time.

$$\frac{m_o}{m_f} = 1 + \int_0^T \frac{m_o f_o^2}{2\eta P} V_2 dt \tag{98}$$

This quantity is to be minimized by selection of the optimal $a(t)$ and $e(t)$, provided these variables also satisfy Eqs. (73) and (74). In taking variations of (98), we note that V_2 depends explicitly on e, Λ_a and Λ_e (Eq. (91)); and so we must have

$$\int_0^T \left(\frac{\partial V_2}{\partial e}\delta e + \frac{\partial V_2}{\partial \Lambda_a}\delta\Lambda_a + \frac{\partial V_2}{\partial \Lambda_e}\delta\Lambda_e \right) dt = 0 \tag{99}$$

Taking also variations of Eqs. (73) and (74)

$$\frac{d(\delta a)}{dt} = 2 f_o \sqrt{\frac{a^3}{\mu}} C \left[\frac{3}{2}\frac{\delta a}{a} + \frac{1}{C}\left(\frac{\partial C}{\partial e}\delta e + \frac{\partial C}{\partial \Lambda_a}\delta\Lambda_a \right) \right]$$

$$\frac{d(\delta e)}{dt} = f_o \sqrt{\frac{a}{\mu}} M \left[\frac{1}{2}\frac{\delta a}{a} + \frac{1}{M}\left(\frac{\partial M}{\partial e}\delta e + \frac{\partial M}{\partial \Lambda_e}\delta\Lambda_e \right) \right]$$

Noting that $\frac{\partial C}{\partial e} = 0$, these two equations can be solved for $\delta\Lambda_a$, $\delta\Lambda_e$ and the result substituted into (99):

$$\int_0^T \left\{ \frac{\partial V_2}{\partial e}\delta e + \frac{1}{2}\frac{\partial V_2/\partial\Lambda_a}{\partial C/\partial\Lambda_a}\left[\frac{1}{f_o}\sqrt{\frac{\mu}{a^3}}\frac{d\delta a}{dt} - 3C\frac{\delta a}{a} \right] + \frac{\partial V_2/\partial\Lambda_e}{\partial M/\partial\Lambda_e}\left[\frac{1}{f_o}\sqrt{\frac{\mu}{a}}\frac{d\delta e}{dt} - \frac{M}{2}\frac{\delta a}{a} - \frac{\partial M}{\partial e}\delta e \right] \right\} dt = 0 \tag{100}$$

The terms containing $\frac{d\delta a}{dt}$ and $\frac{d\delta e}{dt}$ can be integrated by parts, and since a and e are prescribed at both ends, the integrated parts will vanish. The rest can be reorganized into the form

$$\begin{aligned} \int_0^T \Bigg\{ & \left[\frac{d}{dt}\left(\frac{1}{2}\frac{\partial V_a/\partial\Lambda_a}{\partial C/\partial\Lambda a}\sqrt{\frac{\mu}{a^3}} \right) + \frac{3}{2}C\frac{f_o}{a}\frac{\partial V_2/\partial\Lambda_a}{\partial C/\partial\Lambda_a} + \frac{1}{2}M\frac{f_o}{a}\frac{\partial V_2/\partial\Lambda_e}{\partial M\ /\partial\Lambda_e} \right]\delta a + \\ & + \left[\frac{d}{dt}\left(\frac{\partial V_2/\partial\Lambda_e}{\partial M/\partial\Lambda_e}\sqrt{\frac{\mu}{a}} \right) - f_o\frac{\partial V_2}{\partial e} + f_o\frac{\partial V_2/\partial\Lambda_e}{\partial M/\partial\Lambda_e}\frac{\partial M}{\partial e} \right]\delta e \Bigg\} dt = 0 \end{aligned} \tag{101}$$

Using the explicit forms for C, M and V_2 (Eqs.(94) -(96)), and imposing that both bracketed terms in (101) be zero, yields finally the differential equations for Λ_a, Λ_e:

$$\boxed{\frac{d\Lambda_a}{dt} = -\frac{5}{4} f_o \sqrt{\frac{a}{\mu}} \left(1-e^2\right)\Lambda_e^2} \tag{102}$$

$$\boxed{\frac{d\Lambda_e}{dt} = f_o \sqrt{\frac{a}{\mu}} \frac{\Lambda_e}{2} \left(\Lambda_a + \frac{5}{2} e \Lambda_e\right)} \tag{103}$$

These are to be solved together with the equations for a and e. The latter are Eqs. (73) and (74), which using the explicit C and M formulae become

$$\boxed{\frac{da}{dt} = f_o \sqrt{\frac{a^3}{\mu}} \Lambda_a} \tag{104}$$

$$\boxed{\frac{de}{dt} = \frac{5}{4} f_o \sqrt{\frac{a}{\mu}} \left(1-e^2\right)\Lambda_e} \tag{105}$$

The starting values of a and e (a_i, e_i) as well as their final values (a_G, e_G) are prescribed, and constitute the four boundary conditions for the system. ((102)-(105))

2.3.1.4 *Integration of the Differential Equations*

The form of Eqs. (102) - (105) suggest elimination of the variables t and a by dividing each of the equations by Eq. (102):

$$\frac{d\ell na}{de} = \frac{4}{5} \frac{\Lambda_a}{\left(1-e^2\right)\Lambda_e} \tag{106}$$

$$\frac{d\Lambda_a}{de} = -\Lambda_e \tag{107}$$

$$\frac{d\Lambda_e}{de} = \frac{1}{1-e^2}\left(\frac{2}{5}\Lambda_a + e\Lambda_e\right) \tag{108}$$

and, in fact, the pair (107) and (108) is decoupled from (106). An equation for Λ_a alone can be obtained by differentiating (107) and substituting from (108):

$$\frac{d^2\Lambda_a}{de^2} + \frac{1}{1-e^2}\left(\frac{2}{5}\Lambda_a - e\frac{d\Lambda_a}{de}\right) = 0$$

Multiply times $2\left(1-e^2\right)\frac{d\Lambda_a}{de}$ and rearrange:

$$\underbrace{2\left(1-e^2\right)\frac{d^2\Lambda_a}{de^2}\frac{d\Lambda_a}{de}}-2e\left(\frac{d\Lambda_a}{de}\right)^2+\frac{4}{5}\Lambda_a\frac{d\Lambda_a}{de}=0$$

$$\left(1-e^2\right)\frac{d}{de}\left(\frac{d\Lambda_a}{de}\right)^2$$

$$\frac{d}{de}\left[\left(1-e^2\right)\left(\frac{d\Lambda_a}{de}\right)^2\right]+\frac{2}{5}\frac{d}{de}\left(\Lambda_a^2\right)=0$$

Therefore a first integral of the system of equations is

$$\boxed{\Lambda_a^2+\frac{5}{2}\left(1-e^2\right)\Lambda_e^2=4V_2=const.} \tag{109}$$

It was pointed out that V_2 played the role of the Hamiltonian (total energy) in this problem, and Eq. (109) can therefore be viewed as an "energy conservation" statement in a generalized sense.

One immediate consequence of this result is that the mass evolution equation (Eq. (75)) integrates directly:

$$\boxed{\frac{m(t)}{m_o}=\frac{1}{1+\frac{m_o f_o^2}{2\eta P}V_2 t}} \tag{110}$$

Of course, determining the value of V_2 must await incorporation of the particular initial and final conditions of the problem. To this end, we can solve for Λ_e from (109), using the (-) sign in the square root, because we are interested in a decreasing eccentricity (see Eq. (105)):

$$\Lambda_e=-\sqrt{\frac{4V_2-\Lambda_a^2}{\frac{5}{2}\left(1-e^2\right)}} \tag{111}$$

Substitution of (111) into (107) yields a separable first order differential equation for Λ_a:

$$\frac{d\Lambda_a}{\sqrt{rV_2-\Lambda_a^2}}=\frac{de}{\sqrt{\frac{5}{2}\left(1-e^2\right)}} \tag{112}$$

The solution is easily found. Imposing $\Lambda_a(e=o)=\Lambda_{ao}$,

$$\Lambda_a = \Lambda_{ao}\cos\left(\sqrt{\frac{2}{5}}\sin^{-1} e\right) + \sqrt{4V_2 - \Lambda_{ao}^2}\,\sin\left(\sqrt{\frac{2}{5}}\sin^{-1} e\right) \tag{113}$$

Noting that $\sqrt{4V_2 - \Lambda_{ao}^2} = -\frac{5}{2}\Lambda_{eo}^2$ (from (111)), and introducing the notation

$$\boxed{\varepsilon \equiv \sqrt{\frac{2}{5}}\sin^{-1} e} \tag{114}$$

we can write

$$\boxed{\Lambda_a = \Lambda_{ao}\cos(\varepsilon) - \sqrt{\frac{2}{5}}\Lambda_{eo}\sin(\varepsilon)} \tag{115}$$

One differentiation then yields Λ_e:

$$\boxed{\Lambda_e = \frac{\sqrt{\frac{2}{5}}\Lambda_{ao}\sin(\varepsilon) + \Lambda_{eo}\cos(\varepsilon)}{\sqrt{1-e^2}}} \tag{116}$$

The variation of the semi major axis with eccentricity follows from (106) and (108):

$$\frac{d\ell na}{de} = \frac{4}{5}\frac{\Lambda_a}{(1-e^2)\Lambda_e} = \frac{2}{\Lambda_e}\left(\frac{d\Lambda_e}{de} - \frac{e}{1-e^2}\Lambda_e\right) = 2\frac{d\ell n\Lambda_e}{de} + \frac{d\ell n(1-e^2)}{de}$$

$$\therefore \frac{a}{a_g} = (1-e^2)\left(\frac{\Lambda_e}{\Lambda_{eo}}\right)^2 \tag{117}$$

or, from (116),

$$\boxed{\frac{a}{a_G} = \left[\cos(\varepsilon) + \sqrt{\frac{2}{5}}\frac{\Lambda_{a_o}}{\Lambda_{e_o}}\sin(\varepsilon)\right]^2} \tag{118}$$

In the above, $a_G = a(e = o)$, i.e., at the Geo-synchronous and condition.

It remains only to refer all of these variables to time, by solving Eq. (105) for *t(e)*. This can be re-written as

$$dt = \frac{de}{\frac{5}{4}f_o\sqrt{\frac{a}{\mu}}(1-e^2)\Lambda_e}$$

or from (116) and (118), and in terms of $\varepsilon \equiv \sqrt{\frac{2}{5}}\sin^{-1} e$

$$dt = \frac{4}{5}\frac{\upsilon_G}{f_o\Lambda_{eo}}\frac{\sqrt{\frac{5}{2}}d\varepsilon}{\left(\cos\varepsilon + \sqrt{\frac{2}{5}}\frac{\Lambda_{ao}}{\Lambda_{eo}}\sin\varepsilon\right)^2} \tag{119}$$

where $\upsilon_G = \sqrt{\frac{\mu}{a_G}}$ is the orbital velocity in GEO. Integration of (119) with the end condition t = T at $\varepsilon = 0$ yields

$$T - t = \frac{2\upsilon_G}{f_o\Lambda_{a_o}}\left(\frac{1}{1-\sqrt{\frac{2}{5}}\frac{\Lambda_{ao}}{(-\Lambda_{eo})}\tan\varepsilon} - 1\right) \tag{120}$$

The GEO values $(\Lambda_{ao}, \Lambda_{eo})$ of the multiplier follow from (118) and (120) by imposing $a = a_i$ and $e = e_i$ at t = 0. The results are

$$\boxed{\Lambda_{a_o} = \frac{2\upsilon_G}{f_oT}\frac{\cos\varepsilon_i - \sqrt{\frac{a_i}{a_G}}}{\sqrt{\frac{a_i}{a_G}}}} \tag{121}$$

$$\boxed{\Lambda_{e_o} = -\frac{2\upsilon_G}{f_oT}\sqrt{\frac{2}{5}}\frac{\sin\varepsilon_i}{\sqrt{\frac{a_i}{a_G}}}} \tag{122}$$

The V_2 integral can now be evaluated explicitly in terms of initial conditions. Since V_2 = *const.*, we do the calculation (using (96)) at $e = 0$, where the $\Lambda's$ are given by (121) and (122). After simplification, we find

$$\boxed{V_2 = \left(\frac{\upsilon_G}{f_oT}\right)^2\left[1 + \frac{a_G}{a_i} - 2\sqrt{\frac{a_G}{a_i}}\cos\varepsilon_i\right]} \tag{123}$$

From Eqs. (71) and (75), we can see that $f_o^2 V_2$ is in fact $\langle f^2 \rangle$, i.e., the orbit average of the squared thrust/mass ratio. Since we have found this quantity to remain constant over the mission, it is useful to define a "RMS velocity increment" for the mission as:

$$\Delta v_{RMS} \equiv T\sqrt{\langle f^2 \rangle} = f_o T \sqrt{V_2} \tag{124}$$

Using (123), then,

$$\boxed{\Delta v_{RMS} = \upsilon_G \sqrt{1 + \frac{a_G}{a_i} - 2\sqrt{\frac{a_G}{a_i}}\cos\varepsilon_i}} \tag{125}$$

In terms of this ΔV, the mass evolution (from (110)) is now

$$\frac{m(t)}{m_o} = \frac{1}{1 + \frac{m_o \Delta v_{RMS}^2}{2\eta P T}\left(\frac{t}{T}\right)} \tag{126}$$

and, in particular, the final mass is

$$\boxed{\frac{m_f}{m_o} = \frac{m_{pay} + m_{ps}}{m_o} = \frac{1}{1 + \frac{m_o \Delta v_{RMS}^2}{2\eta\, P\, T}}} \tag{127}$$

Here m_{ps} is the portion of the final mass which can be attributed to the propulsion system, the rest being regarded as "payload". It is understood that this "payload" may in fact include the bulk of the power system, and that only the extra devices needed for propulsion (such as the Power Processor Unit and the thrusters themselves) are to be included in m_{ps}.

2.3.1.5 Selection of the Power Level

It is to be noted that the power level is to this point an arbitrary externally supplied constant. This is often the case in practice, as the power system is not designed primarily for the propulsion task, but rather for the primary mission of the spacecraft. If P, T and Δv_{RMS} are all prescribed, so is the RMS accelerations $\Delta v_{RMS}/T$, and from $\frac{P}{m} = \frac{F}{m} c/2\eta$, so will be the orbit-averaged specific impulse. No further optimization is required in this case. If, on the other hand, the size of the installed power system can be varied to suit the propulsion task, the well-known trade off between fuel mass and power system mass will lead to an optimum power level, and hence an optimum specific impulse level. The difference in our problem is that the specific impulse will have additional intra-orbit variations as thrust is modulated according to Eq. (82), and these modulations are, in relative terms, independent of the general level determined by the choice of power. It will also turn out that, because of the long–term variation of the mass, even the optimized specific impulse level will evolve over time as well.

Without loss of generality, we will from here characterize the power level through the mass m_{ps} of the propulsion-related power system. The ratio

$$\alpha \equiv \frac{m_{ps}}{P} \tag{128}$$

is a function of technology level, and will be regarded as a constant. Values in the range 0.01-0.1 Kg/w are reasonable.

In the classical analysis of Stuhlinger (Stuhlinger, pg 76), and in the EP optimization literature, a prominent role is played by the so-called "characteristic velocity", defined as

$$\upsilon_{ch} = \sqrt{\frac{2\eta T}{\alpha}} \tag{129}$$

which can be interpreted as that velocity to which m_{ps} could be accelerated by the power P operating with efficiency η during a time T. For our purposes, υ_{ch} (or $\upsilon_{ch} / \upsilon_G$) can be viewed as a specification of T, the mission duration, although it is more general, in that it also accounts for the specific mass α of the power/propulsion system.

We can now return to Eq. (127) and write the payload mass fraction as

$$\boxed{\frac{m_{pay}}{m_o} = \frac{1}{1 + \frac{m_o}{m_{ps}}\left(\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)^2} - \frac{m_{ps}}{m_o}} \tag{130}$$

This is optimized with respect to the trajectory, but with the power prescribed. If the power level is free, we can further optimize m_{pay}/m_o with respect to m_{ps}/m_o. By differentiation, we find the optimum choice to be

$$\boxed{\left(\frac{m_{ps}}{m_o}\right)_{OPT.} = \frac{\Delta v_{RMS}}{\upsilon_{ch}}\left(1 - \frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)} \tag{131}$$

and if this is substituted back into (130),

$$\boxed{\left(\frac{m_{pay}}{m_o}\right)_{OPT} = \left(1 - \frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)^2} \tag{132}$$

The end-of-burn mass fraction is $\frac{m_p}{m_o} = \frac{m_{pay} + m_{ps}}{m_o}$, so

$$\boxed{\left(\frac{m_f}{m_o}\right)_{OPT} = 1 - \frac{\Delta v_{RMS}}{\upsilon_{ch}}} \qquad) \tag{133}$$

and the propellant fraction is therefore

$$\boxed{\left(\frac{m_{prop}}{m_o}\right)_{OPT} = \frac{\Delta v_{RMS}}{\upsilon_{ch}}} \tag{134}$$

These very simple expressions are identical to those obtained in Stuhlinger (Stuhlinger, pg 104) for pure spiraling; the only difference being that Δv_{RMS} reduces in that case to the usual $\Delta v = \int_0^T f dt$. It is noteworthy (and unexpected) that the same results are recovered in our much more complex situation.

2.3.1.6 Variations of Thrust and Specific Impulse

We now return to the issue of specific impulse variation "in the large", i.e., not including the intra-orbit modulations. Starting from $c^2 = 2\eta P / \dot{m}$ and $\dot{m} = -dm/dt$, we use Eq. (126) for the mass to obtain

$$\bar{c}^2 = \frac{2\eta P}{m_o}\left[1 + \frac{m_o \Delta v_{RMS}^2}{2\eta P T}\left(\frac{t}{T}\right)\right]^2 \frac{2\eta P T^2}{m_o \Delta v_{RMS}^2}$$

where the overbar on $\bar{c}$ is a reminder that this is some form of orbit-averaged specific impulse. The specific form follows from c ~ $\frac{1}{f}$. Since the mean value of the thrust squared is involved, the correct definition for $\bar{c}$ is

$$\bar{c} = \sqrt{1/\langle 1/c^2 \rangle} \tag{135}$$

In non-dimensional form,

$$\boxed{\frac{\bar{c}}{\upsilon_{ch}} = \left(\frac{m_{ps}}{m_o}\right)\left(\frac{\upsilon_{ch}}{\Delta v_{RMS}}\right)\left[1 + \left(\frac{m_o}{m_{ps}}\right)\left(\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)^2 \frac{t}{T}\right]} \tag{136}$$

This shows the mean specific impulse <u>increasing linearly in time.</u> The result is particularly simple when the power is also optimized, as in (131):

$$\boxed{\left(\frac{\bar{c}}{\upsilon_{ch}}\right)_{OPT\ POWER} = 1 - \left(\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)\left(1 - \frac{t}{T}\right)} \tag{137}$$

In this case the optimum $\bar{c}$ starts at $\upsilon_{ch} - \Delta v_{RMS}$ and then increases linearly to $\bar{c} = \upsilon_{ch}$ at the end of the mission. Once again, these results are formally identical to those for pure spiraling.

The variations of specific impulse inside the orbit are due to those of the thrust instead of those of the mass. Therefore, a more appropriate starting point is P = Fc/2η, or $c = \frac{2\eta P}{m f}\frac{1}{\varphi}$, where φ is as given by (72). The long-term variation can be normalized out by dividing through by $\bar{c}$ (Eq. (137)), with the result

$$\boxed{\frac{c}{\bar{c}} = \frac{\sqrt{5/2}\left(\Delta v_{RMS}/\upsilon_G\right)}{\sqrt{\frac{a_G}{a_i}}\sin(\varepsilon_i - \varepsilon) + \sin\varepsilon} \frac{1 + e^2 + 2e\cos\vartheta}{\sqrt{\left[R\left(1 + e^2 + 2e\cos\vartheta\right) + 2\left(e + \cos\vartheta\right)\right]^2 + \left[\left(1 - e^2\right)\frac{\sin\vartheta}{1 + e\cos\vartheta}\right]^2}}} \tag{138}$$

where

$$\boxed{R = \frac{\Lambda_a}{\left(1 - e^2\right)\Lambda_e} = -\sqrt{\frac{5/2}{1 - e^2}}\,\frac{\sqrt{\frac{a_G}{a_i}}\cos(\varepsilon_i - \varepsilon) - \cos\varepsilon}{\sqrt{\frac{a_G}{a_i}}\sin(\varepsilon_i - \varepsilon) + \sin\varepsilon}} \tag{139}$$

It can be verified by direct calculation that $\left\langle \left(\frac{\bar{c}}{c}\right)^2 \right\rangle = 1$, where the time average is performed according to

$$\langle x \rangle = \frac{\left(1 - e^2\right)^{3/2}}{2\pi}\int_o^{2\pi} \frac{x(\vartheta)d\vartheta}{\left(1 + e\cos\vartheta\right)^2} \tag{140}$$

To complete this discussion, we recall that the RMS average (thrust/mass) ratio $\bar{f}$ is a constant, and so

$$\bar{c}(t) = \frac{2\eta P}{m(t)}\frac{1}{\bar{f}} \tag{141}$$

which is equivalent to Eq. (136). In addition, the intra-orbit variations are related simply as

$$\frac{f(\vartheta)}{\bar{f}} = \frac{\bar{c}}{c(\vartheta)} \tag{142}$$

2.3.1.7 *Explicit Long-Term Variations with Time*

Most quantities have so far been related to eccentricity, including time, as given by Eq. (120). It is useful to eliminate the intermediate dependency and express the variables directly as functions of time. First, the quantities Λ_{ao} and Λ_{eo} in (120) can be expressed in terms of initial conditions, using (121) and (122). The result for time is then

$$\frac{t}{T} = \frac{\sin(\varepsilon_i - \varepsilon)}{\sin(\varepsilon_i - \varepsilon) + \sqrt{\frac{a_i}{a_G}} \sin \varepsilon} \tag{143}$$

or, solving for $\varepsilon \equiv \sqrt{\frac{2}{5}} \sin^{-1} e$,

$$\tan \varepsilon = \frac{\tan \varepsilon_i}{1 + \frac{\sqrt{a_i / a_G}}{\cos \varepsilon_i} \frac{t}{T - t}} \tag{144}$$

Substituting this into Eq. (50), and simplifying, we obtain for the semi-major axis

$$\boxed{a = \frac{a_G}{\left(\frac{t}{T}\right)^2 + \frac{a_G}{a_i}\left(1 - \frac{t}{T}\right)^2 + 2\sqrt{\frac{a_G}{a_i}} (\cos \varepsilon_i) \frac{t}{T}\left(1 - \frac{t}{T}\right)}} \tag{145}$$

2.3.2 Results and Comparison to Constant-Thrust Optimization

We use for a limited comparison an example with $a_i = 0.5\ a_G$, $e_i = 0.5$ and $\upsilon_{ch} = 5\upsilon_G$. The latter can be regarded as a specification of mission time; assuming $\alpha = 0.012\,Kg/w$ and η=0.5, and using $\upsilon_G = 3071\,m/s$, Eq. (129) yields $T = 2.83\times10^6 s = 32.7$ days. For best calibration, we assume that the power level is optimized as well, so that Eqs. (131) – (134) and Eq. (137) apply. We then calculate

$$\frac{\Delta v_{RMS}}{\upsilon_{ch}} = 0.1141 \quad \left(\Delta v_{RMS} = 1752\,m/s\right)$$

and
$$\frac{m_{pay}}{m_o} = 0.7849 \quad ; \quad \frac{m_{ps}}{m_o} = 0.1011 \quad \left(\text{or } \frac{P}{m_o} = 8.43w/kg\right.$$

$$\frac{m_{prop.}}{m_o} = 0.1147$$

$$\bar{c}(t=o) = 13{,}600m/s \quad ; \quad \bar{c}(t=T) = 15{,}360m/s$$

For comparison, the computer codes which implement the optimization with constant thrust were applied to the same a_i and e_i; after some interpolation, this yields

$$\left(\frac{\Delta v}{\upsilon_G}\right)_{const.F} = 0.6234, \text{or } \Delta v = 1914m/s$$

The power optimization (or specific impulse optimization) requires in this case an additional calculation. Using the standard rocket equation, it can be shown that when c is constant,

$$\frac{m_{pay}}{m_o} = \left(1 + \frac{c^2}{\upsilon_{ch}^2}\right)e^{-\frac{\Delta V}{c}} - \frac{c^2}{\upsilon_{ch}^2} \tag{146}$$

The optimum value of c can be obtained by differentiation of (146). An approximate solution is

$$\frac{c_{OPT.}}{\upsilon_{ch}} \cong 1 - \frac{1}{2}\left(\frac{\Delta v}{\upsilon_{ch}}\right) - \frac{1}{24}\left(\frac{\Delta v}{\upsilon_{ch}}\right)^2 \tag{147}$$

For this example, with $\frac{\Delta v}{\upsilon_{ch}} = \frac{0.6234}{5} = 0.12468,$ we calculate

$$\frac{c_{OPT.}}{\upsilon_{ch}} = 0.93700, \text{ or } c_{OPT.} = 14{,}390m/s$$

and back-substituting into (78) yields

$$\frac{m_{pay}}{m_o} = 0.7660$$

Other results follow easily:

$$\frac{m_{ps}}{m_o} = 0.1094 \quad ; \quad \frac{m_{prop.}}{m_o} = 0.1246$$

Table 1 below summarizes this comparison.

Table 1: Comparison between variable and constant thrust optimizations

Optimized Variable	Constant Thrust	Variable Thrust	Percentage change (Variable-Constant)
c	14,390 m/s	13,600 – 15,360 m/s	-
m_{pay}/m_o	0.7660	0.7849	+2.5%
m_{ps}/m_o	0.1094	0.1011	-7.6%
m_{prop}/m_o	0.1246	0.1147	-8.4%

The long-term profiles of various quantities for this mission are plotted in Figure 15 and Figure 16. Figure 17 shows intra-orbit profiles of specific impulse, thrust angle (β) and of thrust (normalized as ($fT/\ \upsilon_{ch}$)). Regarding $\beta(\vartheta)$, the important observation is the absence of any thrust reversal in the perigee region. These reversals are a prominent feature of the constant-thrust optimal solution, and are required in order to keep the final apogee from exceeding the target orbit radius (a_G); although some intermediate apogee overshoot is typically present. In our case, β remains quite small throughout, which implies an efficient use of the propulsive force. The apogee control function is now taken over by the throttle, as the lower panel in Figure 17 shows: Thrust is strongly reduced (and specific impulse correspondingly increased) near perigee, especially towards the end of the mission, which is when thrust reversals are strongest if thrust is not modulated. At *t/T*=1, the normalized specific impulse (c/υ_{ch}) varies from 3.6 at perigee (c=55,300 m/s) to about 0.95 at apogee ($c \cong 14{,}700\, m/s$) a throttling ratio of 1/3.8. In terms of thrust, this ratio may be realistically approached by Hall thrusters, but the constant-efficiency approximation would then not be accurate enough (lower efficiency at deep throttle conditions).

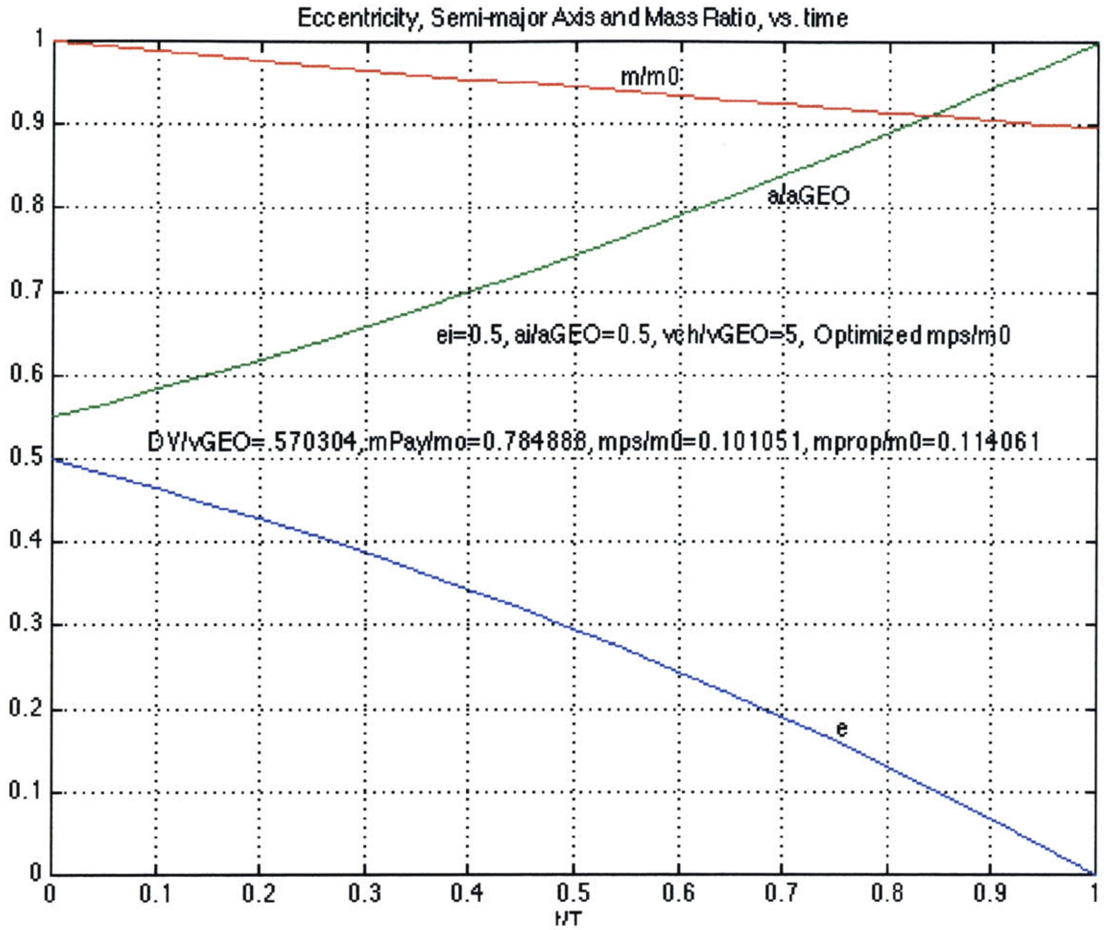


Figure 15: Long-term variation of the mass, semi-major axis and eccentricity

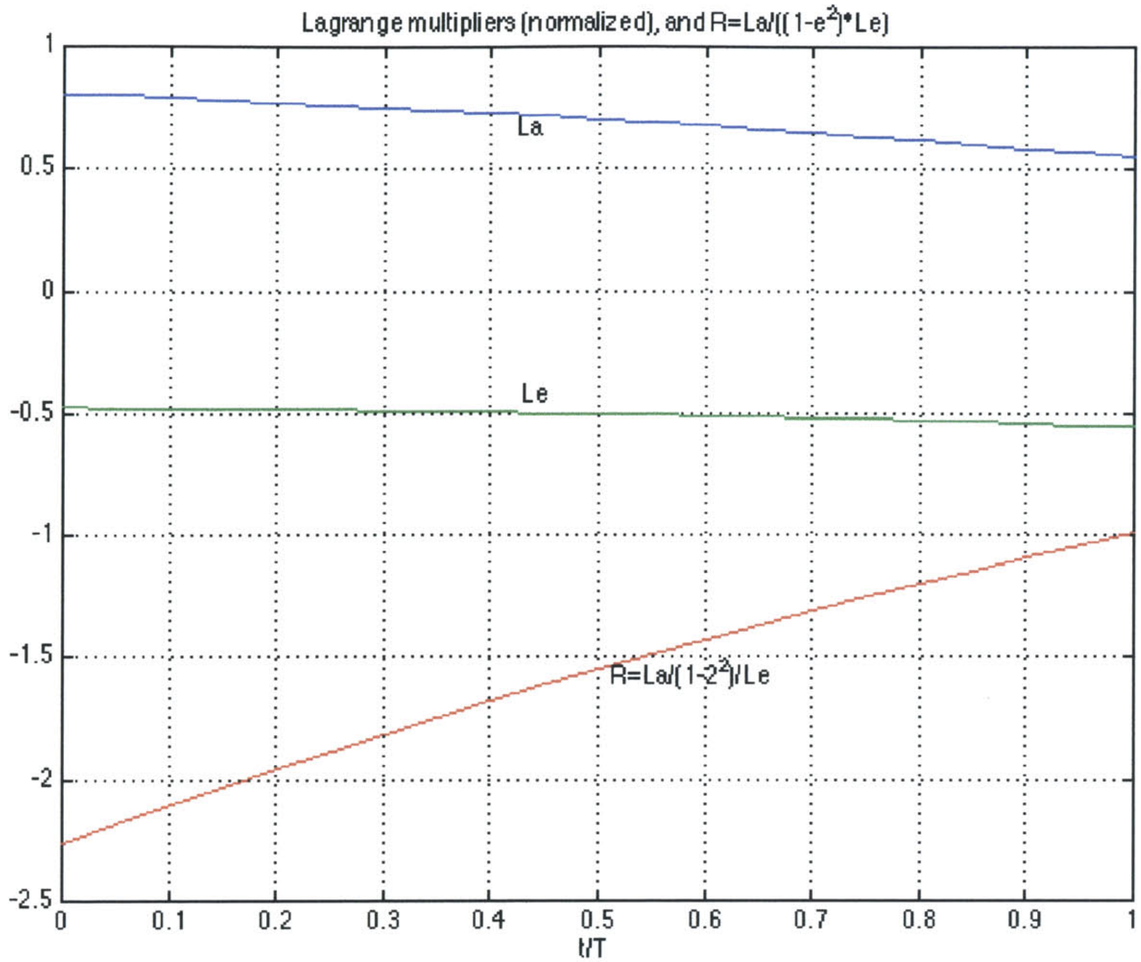


Figure 16: Long-term variation of the two Lagrange multipliers and the ratio R

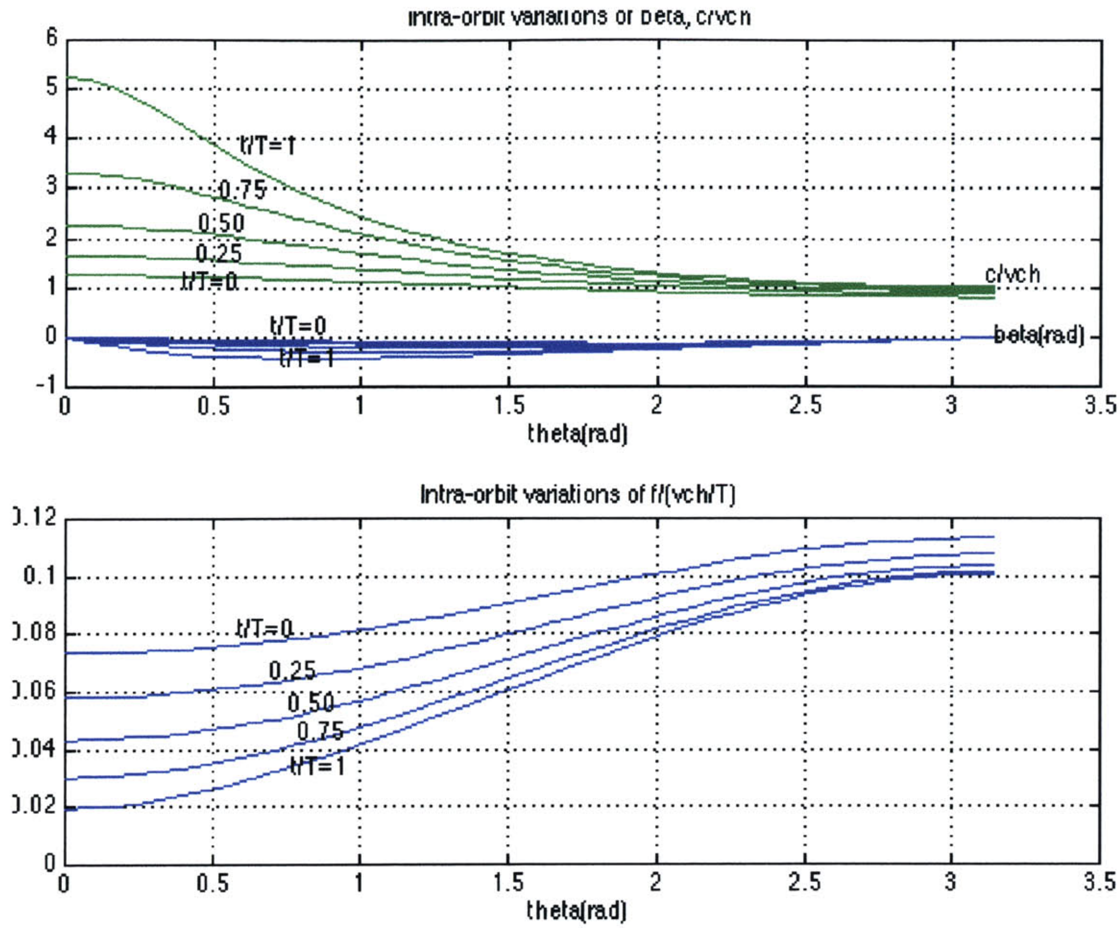


Figure 17: Intra-orbit variations of specific impulse, thrust angle and thrust, at different points along the trajectory

2.3.3 2D Variable Thrust Conclusions

The results of the analysis validate the concept of improving performance by intra-orbit thrust modulations. The magnitude of the payload gain is only moderate, and it is to be expected that it will be insignificant for less ambitious missions than the one in 2.3.2, particularly when less circularization is involved. On the other hand, several advantages of the throttling strategy can be quoted:

a) The results are all expressible as closed-form exact formulae, which allows very rapid and general visualization of trends and effects.
b) Many of these formulae can be used with some confidence even for constant-thrust cases. In particular, the calculated optimum mass ratios are a good approximation for that case, and the optimal RMS specific impulse of the variable-thrust case averages over the mission to nearly the same as the single optimal specific impulse of the constant-thrust case.
c) The thrust orientation profiles are much smoother than with constant thrust, with no reversals near perigee. This may in itself be advantageous in avoiding some attitudes where constraints may play a role.
d) From a theoretical viewpoint, the long-term variations appear to take a universal form which may be generalizable to more complex cases, like those involving plane changes. There are hints to this in the Hamiltonian nature of the outer problem, and more work on this is definitely desirable.

2.4 Derivation of 3D Variable Thrust EP Orbit Raising

2.4.1 Analysis

2.4.1.1 Introduction

The 3D variable thrust derivation can be directly extended from the 2D variable thrust analysis. The same techniques and methods are applied to the 3D variable thrust case, so this analysis will briefly cover the differences between the two. As with the 3D constant thrust case, this analysis also assumes two-body orbital mechanics, orbit averaging, and that the argument of perigee (ω) and longitude of the ascending node (Ω) do not change over the transfer or within the orbit. We are again able to achieve a completely analytic solution that is slightly more efficient than the constant thrust version. This derivation is very useful for quickly finding the upper bounds of performance for 3D transfers, and can be used as a first cut approximation to the constant thrust case.

2.4.1.2 Intra-Orbit Optimization

The traditional measure of goodness in orbit optimization, which is the velocity increment $\Delta V = \int_0^T (F/m)dt$, is no longer significant because it does not relate directly to fuel use when specific impulse (c) varies with time. Instead we minimize the fuel fraction m_o/m. We start with the perturbation equations for $a, e,$ and i, as well as the rate of change of the mass fraction m_o/m.

$$\left\langle \frac{d\left(\frac{m_o}{m}\right)}{dt} \right\rangle = \frac{m_o f_o^2}{2\eta P} V_2 \tag{148}$$

$$\left\langle \frac{da}{dt} \right\rangle = 2 f o \sqrt{\frac{a^3}{\mu}} C \tag{149}$$

$$\left\langle \frac{de}{dt} \right\rangle = f_o \sqrt{\frac{a}{\mu}} M \tag{150}$$

$$\left\langle \frac{di}{dt} \right\rangle = f_o \sqrt{\frac{a}{\mu}} I \tag{151}$$

where f_o is some reference thrust acceleration $(f \equiv f_o \varphi(t))$, to be identified later, and the quantities C, M, I and V_2 are given by

$$V_2 = \frac{\left(1-e^2\right)^{3/2}}{2\pi} \int_o^{2\pi} \frac{\varphi^2 d\vartheta}{(1+e\cos\vartheta)^2} \tag{152}$$

$$C = \frac{1-e^2}{2\pi} \int_o^{2\pi} \varphi \frac{\sqrt{1+e^2+2e\cos\vartheta}}{(1+e\cos\vartheta)^2} \cos\alpha \cos\beta d\theta \tag{153}$$

$$M = \frac{1-e^2}{2\pi} \int_o^{2\pi} \varphi \frac{2(e+\cos\vartheta)\cos\beta + \frac{\left(1-e^2\right)\sin\vartheta}{1+e\cos\vartheta}\sin\beta}{(1+e\cos\vartheta)^2\sqrt{1+e^2+2e\cos\vartheta}} \cos\alpha \cos d\vartheta \tag{154}$$

$$I = \frac{\left(1-e^2\right)^2}{2\pi} \int_o^{2\pi} \varphi \frac{(-\cos\vartheta)}{(1+e\cos\vartheta)^3} \sin\alpha d\vartheta \tag{155}$$

These equations are very similar to the 2D variable thrust case, except for the addition of the equation for *di/dt* (151), the corresponding *I* integral (155), and the $\cos\alpha$ term in the *C* and *M* integrals ((153) and (154)).

We now formulate the optimization problem as minimizing $\left\langle \frac{d(m_o / m)}{dt} \right\rangle$, subject to temporarily prescribed values of $\left\langle \frac{da}{dt} \right\rangle$, $\left\langle \frac{de}{dt} \right\rangle$ and $\left\langle \frac{di}{dt} \right\rangle$. For this purpose, using Eqs. (148) - (151), we minimize the augmented cost function

$$\Phi = \frac{m_o f_o^2}{2\eta P}\left(V_2 - \Lambda_a C - \Lambda_e M - \Lambda_i I\right) \tag{156}$$

with respect to the variations $\delta\,\alpha(\vartheta), \delta\,\beta(\vartheta), \delta\,\varphi(\vartheta)$. Using a similar method as the 2D variable thrust case, we can find the out-of-plane thrust angle (α), the in-plane thrust angle (β), and thrust modulation (φ), all with respect to the true anomaly (θ).

$$\boxed{\tan\alpha = \frac{S}{\sqrt{R^2+Q^2}}} \tag{157}$$

$$\boxed{\tan\beta = \frac{R}{Q}} \tag{158}$$

$$\boxed{\varphi = -\frac{\sqrt{R^2+S^2+Q^2}}{2P}} \tag{159}$$

where R,S,P and Q are defined as

$$R = -\Lambda_e\left(1-e^2\right)^3 \frac{\sin\vartheta}{(1+e\cos\vartheta)^3\sqrt{1+e^2+2e\cos\vartheta}} \tag{160}$$

$$S = \Lambda_i\left(1-e^2\right)^2 \frac{\cos\vartheta}{(1+e\cos\vartheta)^3} \tag{161}$$

$$P = \frac{\left(1+e^2\right)^{3/2}}{(1+e\cos\vartheta)^2} \tag{162}$$

$$Q = \Lambda_a \frac{\left(1-e^2\right)\left(1+e^2+2e\cos\vartheta\right) - \Lambda_e\left(1-e^2\right)^2 2(e+\cos\vartheta)}{(1+e\cos\vartheta)^2\sqrt{1+e^2+2e\cos\vartheta}} \tag{163}$$

Using the profiles of α, β, and φ, all integrations for C, M, I, and V_2 can be performed analytically and give the resulting equations.

$$C = \frac{1}{2}\Lambda_a \tag{164}$$

$$M = \frac{5}{4}\left(1-e^2\right)\Lambda_e \tag{165}$$

$$I = \frac{1+4e^2}{4\left(1-e^2\right)}\Lambda_i \tag{166}$$

$$V_2 = \frac{1}{4}\Lambda_a^2 + \frac{5}{8}\left(1-e^2\right)\Lambda_e^2 + \frac{1}{8}\frac{1+4e^2}{1-e^2}\Lambda_i^2 \tag{167}$$

2.4.1.3 The Long Term Optimization

For the long term optimization, we want to minimize

$$\int_o^T V_2 dt = \frac{2\eta P}{m_o f_o^2}\left(\frac{m_o}{m_f} - 1\right) \tag{168}$$

where $V_2 = V_2(e, \Lambda_a, \Lambda_e, \Lambda_i)$, so taking variations of (168) gives

$$\int_0^T \left[\frac{\delta V_2}{\partial e}\partial e + \frac{\delta V_2}{\partial \Lambda_a}\partial \Lambda_a + \frac{\delta V_2}{\partial \Lambda_e}\partial \Lambda_e + \frac{\delta V_2}{\partial \Lambda_i}\partial \Lambda_i\right] dt = 0 \tag{169}$$

The δΛ's are related to the δa, δe, and δI variations through the (averaged) variation of parameters equations:

$$\frac{d(\delta a)}{dt} = f_o\sqrt{\frac{a^3}{\mu}}\Lambda_a\left(\frac{3}{2}\frac{\delta a}{a} + \frac{\delta\Lambda_a}{\Lambda_a}\right) \tag{170}$$

$$\frac{d(\delta e)}{dt} = f_o\sqrt{\frac{a}{\mu}}\frac{5}{4}(1-e^2)\Lambda_a\left(\frac{1}{2}\frac{\delta a}{a} - \frac{2e}{1-e^2}\delta e + \frac{\delta\Lambda_e}{\Lambda_e}\right) \tag{171}$$

$$\frac{d(\delta i)}{dt} = f_o\sqrt{\frac{a}{\mu}}\frac{1+4e^2}{4(1-e^2)}\Lambda_i\left(\frac{1}{2}\frac{\delta a}{a} + \left(\frac{8e}{1+4e^2} + \frac{2e}{1-e^2}\right)\delta e + \frac{\partial\Lambda_i}{\Lambda_i}\right) \tag{172}$$

We next solve for the δΛ's, substitute them into the integral (169), and integrate the $\frac{d(\delta a)}{dt}$, etc, terms by parts. Then we set the coefficients of δa, δe, and δi to zero separately. The results are the Euler-Lagrange equations for this outer optimization problem:

$$\frac{d\Lambda_a}{dt} = f_o\sqrt{\frac{a}{\mu}}\left(-\frac{5}{4}(1-e^2)\Lambda_e^2 - \frac{1+4e^2}{4(1-e^2)}\Lambda_i^2\right) \tag{173}$$

$$\frac{d\Lambda_e}{dt} = f_o\sqrt{\frac{a}{\mu}}\left[\frac{\Lambda_e}{2}\left(\Lambda_a + \frac{5}{2}e\Lambda_e\right) - \frac{1+4e^2}{8(1-e^2)}\left(\frac{8e}{1+4e^2} + \frac{2e}{1-e^2}\right)\Lambda_i^2\right] \tag{174}$$

$$\frac{\Lambda_i}{\sqrt{a/\mu}} = const. = \frac{\Lambda_{io}}{v_G} \tag{175}$$

where $v_G = \sqrt{\frac{\mu}{a_G}}$ (the orbital speed in GEO)

These equations above are to be solved together with the perturbation equations below for *a, e,* and *i* ((149)-(151)) that now include the solutions for *C, M* and *I* ((164)-(166)).

$$\frac{da}{dt} = fo\sqrt{\frac{a^3}{\mu}}\Lambda_a \tag{176}$$

$$\frac{de}{dt} = \frac{5}{4}f_O\sqrt{\frac{a}{\mu}}(1-e^2)\Lambda_e \tag{177}$$

$$\frac{di}{dt} = f_O\sqrt{\frac{a}{\mu}}\frac{1+4e^2}{4(1-e^2)}\Lambda_i \tag{178}$$

These equations are to be solved using the boundary conditions:

$$\begin{array}{lll} a(t=0)=a_i; & e(t=0)=e_i; & i(t=0)=i_i \\ a(t=T)=0; & e(t=T)=0; & i(t=T)=0 \end{array} \tag{179}$$

2.4.1.4 *Integration of the Differential Equations*

In the same manner as the 2D variable thrust derivation, we can solve the system of differential equations previously described. From (175) we find directly

$$\boxed{\Lambda_i = \frac{\Lambda_{io}}{v_G}\sqrt{\frac{a}{\mu}}} \tag{180}$$

Also by direct calculation, V_2 is found to be constant. Evaluating V_2 at t=0 gives

$$\boxed{V_2 = \frac{\Lambda_a^2}{4} + \frac{5}{8}\Lambda_{eo}^2 + \frac{1}{8}\Lambda_{io}^2} \tag{181}$$

Then the mass equation (148) integrates directly to

$$\boxed{\frac{m}{m_o} = \frac{1}{1 + \frac{m_o f_o^2}{2\eta P} V_2 t}} \tag{182}$$

The semimajor axis ratio can then be found

$$\frac{a}{a_G} = \frac{4V_2 - \Lambda_a^2}{4V_2 - \Lambda_{ao}^2} \tag{183}$$

Writing (183) in terms of eccentricity leads to

$$\boxed{\frac{a}{a_G} = \left(\cos\varepsilon + \lambda_{a_o}\sqrt{\frac{2}{5+\lambda_{i_o}^2}}\sin\varepsilon\right)^2} \tag{184}$$

where we define

$$\varepsilon = \sqrt{\frac{2}{5}\frac{1+\frac{1}{5}\lambda_{i_o}^2}{1+\lambda_{i_o}^2}}\sin^{-1}\left(e\sqrt{1+\lambda_{i_o}^2}\right) \tag{185}$$

and

$$\lambda_{a_o} = \frac{\Lambda_{a_o}}{\Lambda_{e_o}} < 0 \quad , \quad \lambda_{i_o} = \frac{\Lambda_{i_o}}{\Lambda_{e_o}} \tag{186}$$

and we can find the initial Λ values by imposing $\varepsilon = \varepsilon_i$ at t = 0, so that

$$\boxed{\Lambda_{a_o} = \frac{2\upsilon_G}{f_o T}\frac{\cos\varepsilon_i - \sqrt{a_i/a_G}}{\sqrt{a_i/a_G}}} \tag{187}$$

$$\boxed{\Lambda_{e_o} = -\frac{2\upsilon_G}{f_o T}\frac{\sin\varepsilon_i}{\sqrt{a_i/a_G}}\sqrt{\frac{2}{5+\lambda_{i_o}^2}}} \tag{188}$$

If, in particular, we set a=a_i and ε=ε_i in Eq. (184), then a relationship results between λio and λao:

$$\lambda_{ao}\sqrt{\frac{2}{5+\lambda_{io}^2}} = \frac{\sqrt{\frac{a_i}{a_G}} - \cos\varepsilon_i}{\sin\varepsilon_i} \tag{189}$$

From the equation for *de/dt* (150), we can find the equation that describes the instantaneous time (*t*) of the transfer (where *T* is the total prescribed transfer time).

$$\boxed{T - t = \frac{2\upsilon_G}{f_o \Lambda_{a_o}}\left(\frac{1}{1-\sqrt{\frac{2\lambda_{a_o}^2}{5+\lambda_{i_o}^2}}\tan\varepsilon} - 1\right)} \tag{190}$$

Integrating the ratio of Eqs (178) and (177) for inclination gives

$$\boxed{i = \cot^{-1}\left(\frac{\sqrt{1+\lambda_{i_o}^2}}{\lambda_{i_o}}\cot\sqrt{\frac{5}{2}\frac{1+\lambda_{i_o}^2}{1+\lambda_{i_o}^2/5}}\,\varepsilon\right) - \frac{4}{\sqrt{10}}\frac{\lambda_{i_o}}{\sqrt{1+\lambda_{i_o}^2/5}}\varepsilon} \tag{191}$$

or in terms of eccentricity (*e*), this becomes

$$\boxed{i = \cot^{-1}\left(\frac{\sqrt{1-\left(1+\lambda_{i_o}^2\right)e^2}}{\lambda_{i_o}^2 e}\right) - \frac{4\lambda_{io}}{5\sqrt{1+\lambda_{i_o}^2}}\sin^{-1}\left(e\sqrt{1+\lambda_{i_o}^2}\right)} \tag{192}$$

Imposing $i = i_i$ when $e = e_i$ gives the equation for λ_{io}

$$i_i = \cot^{-1}\left(\frac{\sqrt{1-\left(1+\lambda_{i_o}^2\right)e_i^2}}{\lambda_{i_o}^2}\right) - \frac{4\lambda_{i_o}}{5\sqrt{1+\lambda_{i_o}^2}}\sin^{-1}\left(e_i\sqrt{1+\lambda_{i_o}^2}\right) \Rightarrow \lambda_{i_o} \tag{193}$$

From here, Eq (189) gives λ_{ao}, and other quantities like ε_i (185), etc, follow easily.

2.4.1.5 Power and Mass

As with the 2D variable thrust case, the propulsion system power level can be specified two ways (see 2.3.1.5). It can be an arbitrary externally supplied constant, as when the power system is not designed primarily for the propulsion task, but rather for the primary mission of the spacecraft. Alternatively, the size of the installed power system can be varied to suit the propulsion task, the well-known trade off between fuel mass and power system mass will lead to an optimum power level, and hence an optimum specific impulse level. For this case the specific impulse will have additional intra-orbit variations as thrust is modulated according to Eq. (159), and these modulations are, in relative terms, independent of the general level determined by the choice of power. It will also turn out that, because of the long–term variation of the mass, even the optimized specific impulse level will evolve over time as well.

When the power level is an arbitrary externally supplied constant, the payload mass fraction becomes:

$$\boxed{\frac{m_{pay}}{m_o} = \frac{1}{1+\frac{m_o}{m_{ps}}\left(\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)^2} - \frac{m_{ps}}{m_o}} \tag{194}$$

where we define

$$m_{ps} = \alpha P$$

$$\upsilon_{ch} = \sqrt{\frac{2\eta T}{\alpha}}$$

$$\Delta v_{RMS} = T\sqrt{\langle f^2 \rangle} = f_o T\sqrt{V_2}$$ which can be calculated explicitly as

$$\boxed{\Delta v_{RMS} = \upsilon_G \sqrt{1+\frac{a_G}{a_i} - 2\sqrt{\frac{a_G}{a_i}}\cos\varepsilon_i}} \tag{195}$$

If the power system size and mass are free to be optimized as well, then differentiating with respect to $\frac{m_{ps}}{m_o}$, we obtain the following results

$$\boxed{\begin{aligned} &\left(\frac{m_{ps}}{m_o}\right)_{OPT} = \frac{\Delta v_{RMS}}{\upsilon_{ch}}\left(1-\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right) && (196)\\ &\left(\frac{m_{pay}}{m_o}\right)_{OPT} = \left(1-\frac{\Delta v_{RMS}}{\upsilon_{ch}}\right)^2 && (197)\\ &\left(\frac{m_{prop}}{m_o}\right)_{OPT} = \frac{\Delta v_{RMS}}{\upsilon_{ch}} && (198)\end{aligned}}$$

And the mean specific impulse $\bar{c} = \sqrt{1/\left\langle 1/c^2 \right\rangle_{orbit}}$ varies in general as

$$\boxed{\frac{\bar{c}}{v_{ch}} = \frac{m_{ps}}{m_o}\frac{v_{ch}}{\Delta v_{RMS}}\left[1+\frac{m_o}{m_{ps}}\left(\frac{\Delta v_{RMS}}{v_{ch}}\right)^2\frac{t}{T}\right]}$$

And if power is selected optimally, the mean specific impulse is

$$\boxed{\frac{\bar{c}}{v_{ch}} = 1-\frac{\Delta v_{RMS}}{v_{ch}}\left(1-\frac{t}{T}\right)}$$

In particular, $\bar{c}_{opt} = v_{ch} - \Delta v_{RMS}$ at $t = 0$

and $\bar{c}_{opt} = v_{ch}$ at $t = T$

2.4.2 Results and Conclusions

As an example to compare to the 3D constant thrust case, we assume

$e_i = 0.5$ $\qquad a_i/a_G = 0.4$ $\qquad i_i = -0.205\text{rad} = -11.75°$

Solving the equation for λ_{io} numerically gives λ_{io} =-1.1585, and then

$\varepsilon_i = 0.4056\text{rad}$ $\qquad \Delta v_{RMS}/v_G = 0.77087$

For comparison, optimizing with <u>constant thrust</u> for the same initial conditions gives

$\Delta v_{RMS}/v_G = 0.83$

But we should compare <u>by mass</u>. Assuming <u>optimal power</u> is chosen, and a mission such that Vch/vG = 5 $\quad (\Delta v_{RMS}/v_G = 0.15417)$, then we get the following results:

Table 2: Comparison of 3D variable and constant thrust missions

Variable Thrust	$\frac{m_{pay}}{m_o} = 0.71542$	$\left\langle \frac{\bar{c}}{v_{ch}} \right\rangle = 0.92292$
Constant Thrust	$\frac{m_{pay}}{m_o} = 0.69518$	$\frac{c_{opt}}{v_{ch}} = 0.91585$

We can see from the results that the *variable thrust payload mass fraction is **2.9% more** than with constant thrust.*

This variable thrust analysis provides a complete analytical optimization tool for 3D orbits (restricted case). The results imply variable thrust and Isp within each orbit, and may require unrealistic throttling. However, they provide upper bounds for performance. The theory also ignores efficiency variations with specific impulse. Including that greatly complicates the algebra. There is a small performance increase compared to constant-thrust cases (except when ΔV_{RMS} is large, comparable to v_{ch}). This means simple formulae can be used for the first cut optimization.

3 Optimization Software Development and Description

3.1 2D MITEOR Optimization Software

The 2D MITEOR (MIT Electric Orbit Raising) optimization software was developed in Matlab to solve the 2D constant thrust transfers described in Section 2.1. It can be used to either optimize a single transfer for given initial conditions or optimize multiple transfers to be viewed at one time. The basic algorithm used is a Runge-Kutta (low order, *ode23*) routine that solves the system of differential equations (18), (19), and (27) starting at the known boundary condition of GEO and working backwards to the initial conditions of the transfer. The partial derivatives in (27) are solved using a custom finite difference routine. The optimization outputs semimajor axis (as non-dimensional a/a_{final}), velocity increment Δv (as non-dimensional $\Delta v/v_{cf}$, where v_{cf} is the circular final velocity), and Lagrange multiplier Λ, for the given range of eccentricity values (the independent parameter). The thrust angle β for any orbit can then be found given the appropriate combination of eccentricity and the Lagrange multiplier values.

The software is made up of six different files that are named *miteor2D.m, main2D.m, main2Dall.m, paths2D.m, CandM2D.m,* and *angle2D.m.* The optimization is run by simply typing *miteor2D* at the command prompt (assuming the current path is set to the folder that contains all six files). If a single transfer for specific initial conditions is to be run, then the code follows the structure in the flowchart of Figure 18. Alternatively, if multiple transfers need to be compared, the structure of the code follows the flowchart in Figure 19. The *miteor2D.m* file controls the overall optimization process, and contains a text-based user interface, an algorithm for solving for initial conditions, outputs raw solution data, and creates a series of graphs. The *main2D.m* file is used only for optimizing transfers with specified initial conditions. It sets up and runs the Runge-Kutta (ode23) routine, and is called repeatedly by *miteor2D* until it outputs the transfer whose initial conditions match those requested by the user. The *main3Dall.m* file is used to create a whole family of transfers to be viewed all at once. It is called only once by *miteor2D*, but iterates internally to create a series of slightly different transfers over the given range of eccentricity values. It also creates a series of graphs that are useful for seeing general trends between different transfers. The *paths2D.m* file is used by the Runge-Kutta routine (ode23) to calculate the value of the derivatives in Equations (18), (19), and (27) and also contains custom finite difference code to calculate the partial derivatives found in Equation (27). The *CandM2D.m* file uses a simple iteration routine to calculate the integrals C (15) and M (16) as well as the thrust angle β (22), since they all rely on each other. The *angle2D.m* file is almost identical to the *CandM2D.m* file, but is run after the optimization to solve for the thrust angle values.

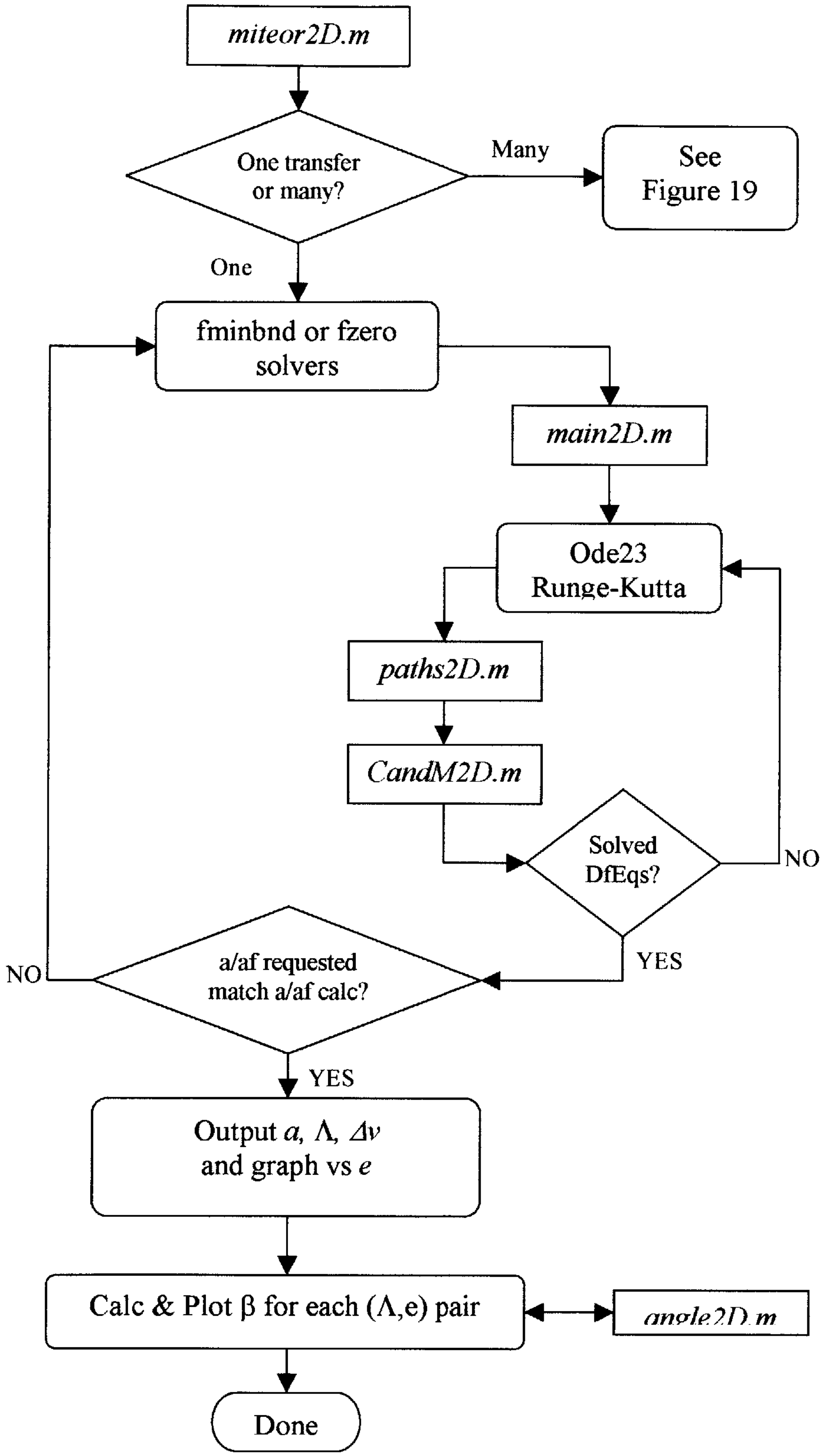


Figure 18: Flowchart for MITEOR 2D, solving for transfer of specific initial conditions

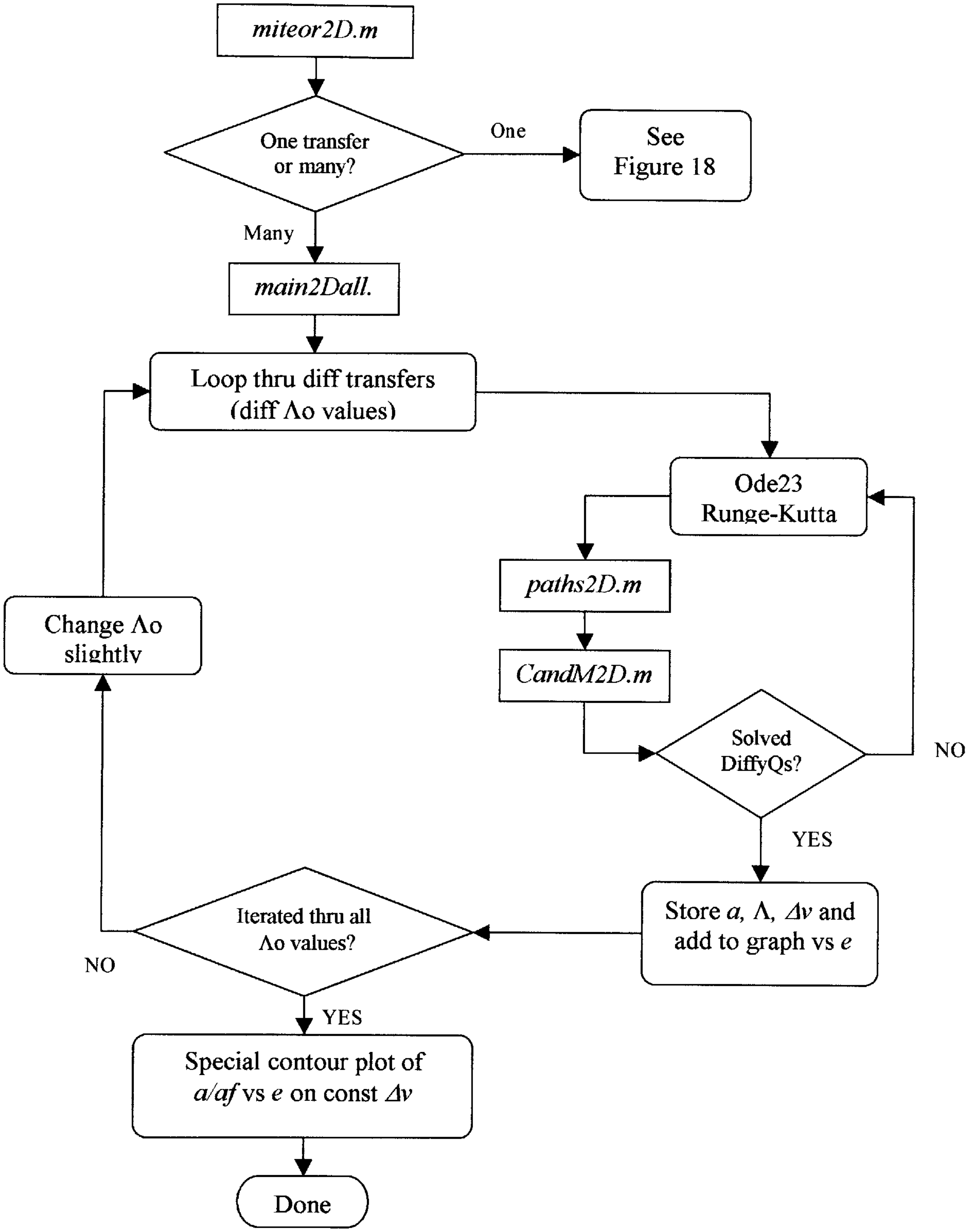


Figure 19: Flowchart for MITEOR 2D, solving for multiple transfers at once

3.1.1 2D MITEOR Code Description

3.1.1.1 Miteor2D.m

The *miteor2D.m* file controls the overall optimization process and is used to run the 2D MITEOR software (type *miteor2D* at the command prompt in Matlab). The first part of the code contains a text-based user interface, which first asks the user if they want to optimize one transfer for specific initial conditions or run multiple transfers to be viewed all at once.

If the user chooses to run multiple transfers, then the interface gathers information on the range of eccentricity values to optimize over. It then passes the eccentricity values to *main2Dall.m*, which then optimizes and graphs several transfers (see 3.1.1.3 for more details) and ends the program.

If the user chooses to optimize one transfer for specific initial conditions, then the interface collects data from the user on eccentricity ranges and initial semimajor axis ratio, and asks the user to choose between two routines for solving for the initial conditions of the transfer. The choices for the routines are *fminbnd* and *fzero*. The *fminbnd* routine minimizes the difference between the requested initial *a/af* value and the calculated *a/af* value over a fixed interval of Λ values. It is recommended over the *fzero* routine, since it is much faster, but is not often present on most Matlab 5 installations (comes default with Matlab 6). The *fzero* routine works by minimizes the same semimajor axis difference by finding where this difference changes sign.

After the user interface has finished, a few parameters are initialized, and then either *fminbnd* or *fzero* is run. Both routines call *main2D.m* (see 3.1.1.2), which optimizes one transfer for the value of Λ at GEO that is passed to it. It should be noted that *all* transfers are solved backwards, starting at the known boundary condition of GEO, and working backwards to some initial starting condition of the transfer. Therefore, within the code, the "initial" values that must be used to solve the differential equations are physically the final conditions at GEO.

The data from the resulting optimized transfer found from the search routines is then displayed and graphed. Graphs of Λ, a/af, and Δv/vcf versus eccentricity are displayed with the option of overlaying the radius of apogee and perigee. If desired, the thrust angle β is then calculated using *angle2D.m* (see 3.1.1.6) and plotted versus the argument of perigee θ for all values of eccentricity. The values of β are stored in the matrix *betam*, where the columns of *betam* correspond to the rows or the *espan* vector, and the rows of *betam* correspond to the rows of the *theta* vector (true anomaly).

3.1.1.2 Main2D.m

The *main2D.m* file is used only for optimizing transfers with specified initial conditions. It sets up and runs the Runge-Kutta (ode23) routine that solves the system of differential equations (18), (19), and (27) starting at the known boundary condition of GEO and working backwards to the initial conditions of the transfer. It is called repeatedly by *miteor2D* until it outputs the transfer whose initial conditions match those requested by the user.

To solve three differential equations at once, the starting values (at GEO physically) of the variables Λ, a/af, and $\Delta v/vcf$ must be placed into one vector, which is called Yv0. The first entry of Yv0 (or Yv0(1)) is where the initial guess for Λ is stored. The second entry of Yv0 (Yv0(2)), corresponds to value of a/af at GEO, which should be 1. The third entry of Yv0 (Yv0(3)), corresponds to the value of $\Delta v/vcf$ at GEO, which should be 0. Similarly, after running *ode23,* the solutions are in the form of the matrix Yv, the columns correspond to Λ, a/af, and $\Delta v/vcf$, and the rows correspond to the entries in the *espan* vector.

The *ode23* function calculates the values of the differential equations (18), (19), and (27) using *paths2D.m*, which also solves the partial derivatives in (27). After *ode23* completes, the difference between a/af requested by the user and a/af calculated is returned, which is used by *fminbnd* or *fzero* to find the requested transfer.

3.1.1.3 Main2Dall.m

The *main2Dall.m* function is used to create a whole family of transfers to be viewed all at once. The process at the heart of *main2Dall* is very similar to *main2D*, except it is called only once by *miteor2D,* and has the addition of an internal loop through different transfers (indexed by k) and the production of colored and labeled composite graphs of all the transfers.

The starting guess at Λ (physically at GEO), is determined by the k index in a way that evenly distributes the starting values of Λ from zero to < -1. The code then proceeds similarly to *main2D*, using *ode23* to solve each transfer. The transfer data for Λ, a/af, and $\Delta v/vcf$ are then compositely added to graphs versus eccentricity. A text label of the k index is added next to each curve and the colors of the curves are rotated in order to better identify transfer data between graphs.

Finally, after all transfers have been run, a special contour plot is created of a/af versus e for constant values of $\Delta v/vcf$. The tricky algorithm for this plot is basically an interpolation between the composite graph of $\Delta v/vcf$ versus e and the composite graph of a/af versus e. Basically, the algorithm loops through constant values of $\Delta v/vcf$ to create each contour, and for each contour, loops through the values in the eccentricity vector (*espan*). The algorithm interpolates the k index value of the transfer at the current e whose $\Delta v/vcf$ is equal to the current constant $\Delta v/vcf$ of the outer loop. Then it interpolates the value of a/af at the current e of the transfer whose index is the k value previously interpolated. The interpolation routine used is the built-in Matlab routine *interp1*. The result is a very useful graph of *a/af* versus *e* for constant values of *Δv/vcf*, which has been used for mission analysis studies by Space Systems/Loral.

3.1.1.4 Paths2D.m

The *paths2D.m* file is used by the Runge-Kutta routine (ode23) to calculate the value of the derivatives in Equations (18), (19), and (27) and also mostly contains custom finite difference code to calculate the partial derivatives found in Equation (27).

The code first extracts the current values of *Λ*, *a/af*, and *Δv/vcf* from the Y vector and the current *e*, which are all passed to it by the *ode23* routine. Then the code calculates the following partial derivatives that are found in Equation (27) and (26):

$$\frac{\partial F}{\partial e} \text{ and } \frac{\partial F}{\partial \Lambda} \quad \text{where } F = \frac{\dfrac{\partial (V/M)}{\partial \Lambda}}{\dfrac{\partial (C/M)}{\partial \Lambda}}$$

The finite difference routine makes small adjustments (±0.001) to the parameters *Λ* and *e*, and then calculates the integrals *C* and *M* using the function *CandM2D.m* (remember *V=1*). A central differences technique is then used with varied integral values to find the partial derivatives shown above. Once the partial derivative values are known, they are substituted into Equations (18), (19), and (27) to calculate the value of these derivatives, which are then passed back to *ode23*.

3.1.1.5 CandM2D.m

The *CandM2D.m* function uses a simple iteration routine along with the trapezoidal method to calculate the integrals *C* (15) and *M* (16) as well as the thrust angle β (22), since they all rely implicitly on each other.

The code takes in a current value of *e* and *Λ*, makes a guess at the initial values of the *C* and *M* integrals, and then sets up the calculation of the integrals. Since the *C* and *M* integrals depend on θ, a vector of θ values is created for half the orbit (since β turns out to be anti-symmetric, no need to calculate second half of orbit), and this is used to calculate the components of *C* and *M*. These components are then combined together inside the iteration, where β is then calculated from (22) and then used with the trapezoidal method to calculate the integrals. The error in the integrals just calculated and the initial guess is found, the guess is then updated to the current integral values, and the iteration continues until the error is approximately zero.

3.1.1.6 Angle2D.m

The *angle2D.m* file is almost identical to the *CandM2D.m* file, but is run after the optimization to solve for the thrust angle β values. The only difference in the code is that after the iteration routine completes and the β values are found for half the orbit, a small routine then adds the second half of the thrust angles to the β vector. This is possible because the β vector is anti-symmetric throughout the orbit.

3.2 3D MITEOR Optimization Software

The 3D MITEOR (MIT Electric Orbit Raising) optimization software is being developed in Matlab to solve the 3D constant thrust transfers described in Section 2.2. The code is setup very similar to the 2D MITEOR code, except a routine still needs to be implemented to solve for specific initial conditions (which is now a harder problem to solve), or to easily create contour plots of multiple transfers (such as in 2.2.2). The current code works by manually selecting initial m_a and m_e values, which then produce specific transfers.

The code is still being developed, but currently it is structured similar to the 2D MITEOR (see Figure 20), and currently contains the six files *main3D.m, onetraj3D.m, paths3.m, jacob.m, CMI.m,* and *angle3D.m*. The *main3D* or *onetraj3D* functions set up and run the Runge-Kutta (low order, *ode23*) routine that solves the differential equations w.r.t. eccentricity for m_a (60), m_e (61), a/af (52), e (53), and $\Delta v/vcf$ (54). The *paths3.m* function is used by the *ode23* routine to calculate the values of the derivatives (just listed) at the current i, m_a and m_e. It also uses a custom finite difference routine to calculate the partial derivatives in Equations (60) and (61). Partial derivatives of the functions F and G are needed, which are actually ratios of Jacobians that are themselves made of partial derivatives. For this reason, the function *jacob.m* is used to create the functions F and G using a similar finite difference routine. The integrals C, M, and I also need to be calculated, so this is handled in the function *CMI.m*. Finally, the thrust angles α and β can be calculated using *angles3D.m* after the optimization finishes.

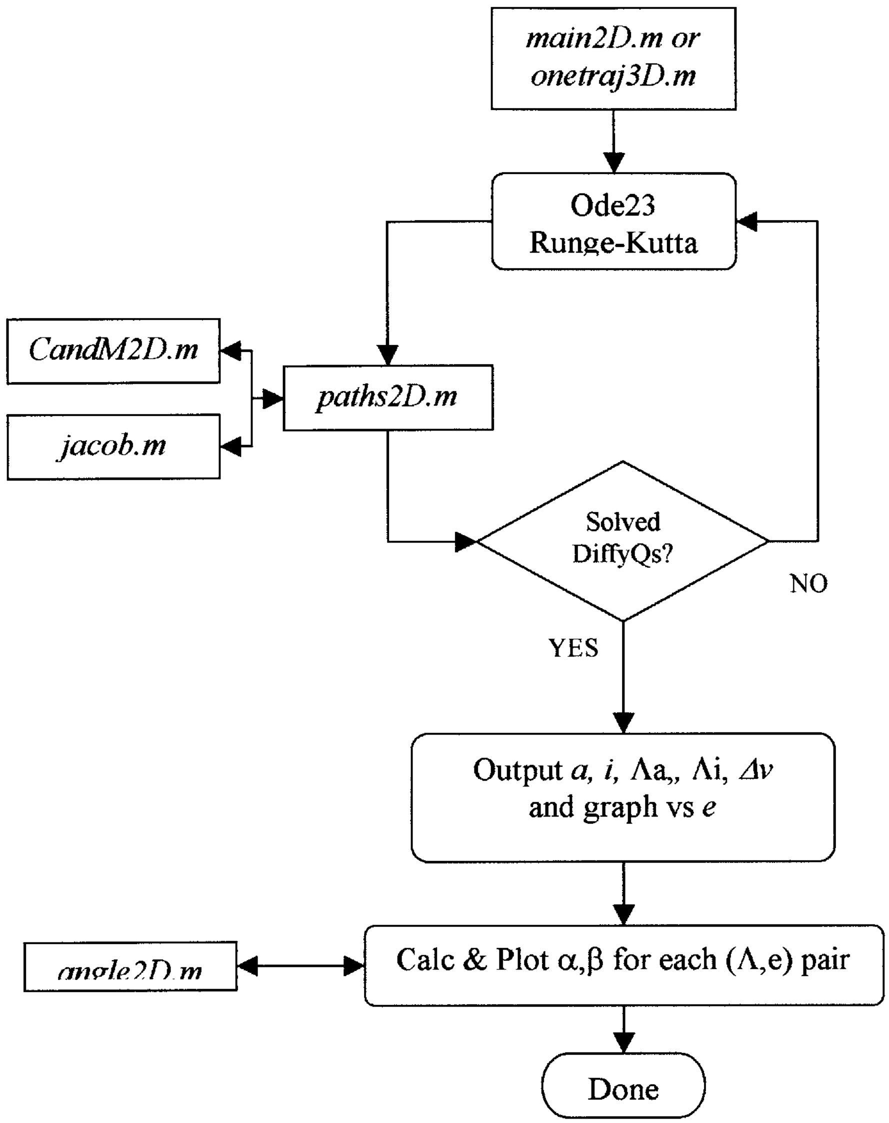


Figure 20: Flowchart for 3D MITEOR optimization software

3.2.1 3D MITEOR Code Description

3.2.1.1 Main3.m or Onetraj3D.m

The *main3D* or *onetraj3D* functions set up and run the Runge-Kutta (low order, *ode23*) routine that solves the differential equations w.r.t. eccentricity for m_a (60) , m_e (61), *a/af* (52), *e* (53), and $\Delta v/vcf$ (54). The *main3D* function is very similar to *main2D.m*, and was used for preliminary development of the 3D software. The *main3D* function eventually evolved into the *onetraj3D* function, which is being designed to be easily called to create a single trajectory with the given initial values of m_a, m_e, and initial inclination. This makes it easier to be called by search routines that can find the correct combination of multiplier values to match the users desired initial conditions of the transfer. It also currently has the ability to create and graph the thrust angles α and β using *angle3D.m,* as well as the basic plots of the orbital elements over the transfer.

3.2.1.2 Paths3.m

The *paths3.m* function is used by the *ode23* routine to calculate the values of the derivatives (just listed) at the current *i*, m_a and m_e. It also uses a custom finite difference routine to calculate the partial derivatives in Equations (60) and (61). The finite difference routine makes small adjustments (±0.0001) to the parameters Λ_a, Λ_i and *e*, and then calculates the functions *F* and *G* using jacob.*m* (remember $V=1$). A central differences technique is then used with varied *F* and *G* values to find the partial derivatives in Equations (60)-(64). Once the partial derivative values are known, they are substituted into Equations (60), (61), (52), (53), and (54) to calculate the value of these derivatives, which are then passed back to *ode23*.

3.2.1.3 Jacob.m

The *jacob.m* function uses the same finite difference techniques as *paths3* to calculate the functions *F* and *G* (58), which are actually ratios of the Jacobians in Eq. (59). The *jacob.m* function takes in the parameters *e*, m_a and m_e and makes small adjustments (±0.0001) to them and then calculates the integrals *C, M,* and *I* using the *CMI.m* function. A central differences technique is then used with varied integral values to find the partial derivatives found in the Jacobians (59), which are then used to create the Jacobians, and then the ratios *F* and *G*. The *F* and *G* values are then passed back to *paths3.m.*

3.2.1.4 CMI.m

The integrals *C, M,* and *I* are calculated using the *CMI.m* function. It is very similar to the *CandM.m* function (see 3.1.1.5), except for the addition of the *I* integral, and the lack of iteration. The thrust angles α (50) and β (48) no longer depend on the integrals because of the different formulation approach taken in the 3D case. This eliminates the need for iteration in *CMI.*

3.2.1.5 Angles3D.m

The *angles3D.m* function is the same as *CMI*, except that is called after the optimization completes, and it is used to produce the thrust angles α (50) and β (48). It also adds the second half of the orbit's thrust angles for α, which is symmetric, and β, which is negative anti-symmetric.

4 Conclusions

Analysis has been completed and software developed for optimizing constant thrust transfers for the 2D and restricted 3D cases. The software developed (called MITEOR) is robust, converges well for most all cases, and the 2D version of the code can optimize for transfers with specific initial conditions or be used to view multiple transfers at once. Derivations have also been completed for both 2D and 3D transfers that optimize both thrust angles *and* thrust magnitude. These variable thrust derivations have been found to be completely analytic and require no additional numerical routines.

The core of the restricted 3D analysis and software has been completed, but improvements are still being developed. Solutions have been produced but are restricted to cases where the initial line of apses is aligned with the line of nodes, and no disturbances to these lines are included. Further work is needed to extend the analysis and code to the more general 3D cases, provide an outer shell that will select the desired starting point for the mission, explore constraint violations incurred by the unconstrained codes, and devise ways to incorporate the important constraints into the optimization.

The results of the 2D and 3D variable thrust transfers are typically 5-10% more fuel-efficient than constant thrust, and can be used to easily calculate first cut approximation to the constant thrust cases, providing an optimum upper bound. The results are all expressible as closed-form exact formulae, which allows very rapid and general visualization of trends and effects. The thrust orientation profiles are much smoother than with constant thrust, with no reversals near perigee. This may in itself be advantageous in avoiding some attitudes where constraints may play a role.

Overall, the first segment of this project has been completed with promising results and a strong understanding of the analysis, which will be required to continue on to more complicated and realistic cases. During the next segment of this project, improvements in the 3D analysis and code should allow Loral to directly apply MITEOR to the development of their next-generation GEO satellites.

5 Bibliography

- Battin, Richard. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* AIAA Education Series, Reston, VA. 1999.

- Ilgen, Mark. "A Hybrid Method For Computing Optimal Low Thrust OTV Trajectories". Advances in Astronautical Sciences, Vol 87, Part II, 1994, pg 941-958

- Kluever, Craig and Oleson, Steven. "Direct Approach to Computing Near-Optimal Low-Thrust Earth-Orbit Transfers". Journal of Spacecraft and Rockets, Vol 35, No 4, July-August 1998, pg 509-515

- Kryloff and Bogoliuboff. *Introduction to Non-linear Mechanics.* Translated by Lefschetz, Princeton University Press, Princeton, N.J., 1947.

- Stuhlinger, Ernst. *Ion Propulsion for Space Flight.* McGraw-Hill Book Company, New York. 1964.

Other literature on electric propulsion orbit raising has been reviewed and summarized in Appendix A. This literature is not included here since it is not referenced explicitly in the thesis.

Appendix A: Literature Review Database

This appendix (also located in *references.mdb*) contains easy-to-compare summaries of most of the relevant orbit raising literature that was available at the MIT Aero/Astro library.

Reference Source Information

Title

A Hybrid Method For Computing Optimal Low Thrust OTV Trajectories

Author(s)
Mark R. Ilgen

Source
Advances in Astronautical Sciences, Vol 87, Part II, 1994, pg 941-958

Paper ID#
AAS 94-129

Year
1994

Reference Summary

Optimization Method
Direct Method (Hybrid: Direct shooting/NLP & optimal control theory)

Orbital Elements Used
Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☑	☑	☑	☑	☑	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Yes

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Yes

Advantages
Robust convergence properties and insensitive to initial guess of optimization variables
Much better convergence than SEPSPOT when including additional effects
Results matched SEPSPOT and LOWTOP within a few % (for cases that converged)
Determines optimal thrust steering for both min transfer time and min delta-v
Modular in design, readily upgradable, can handle additional effects, like J3, J4..., atmos drag, sun/moon
Very well written, extensive, easy to follow documentation, good description of code
Equinoctial elements used, so no singularities (except e=1, parabolic)

Disadvantages
HYTOP is written in FORTRAN 77, old language, may need port to Matlab for ease of use (graphs,STK...)

Sofware Developed or Used
Developed HYTOP (Hybrid Trajectory Optimization Program)
Uses NLP2 nolinerar programming package and set of FORTRAN 77 subroutines for integrations

Reference Source Information

Title

Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers

Author(s)

Craig A. Kluever, Steven R. Oleson

Source

Journal of Spacecraft and Rockets, Vol 35, No 4, July-August 1998, pg 509-515

Paper ID#

AAS 97-717

Year

1998

Reference Summary

Optimization Method

Direct Method (NonLinear Programming - NLP)

Orbital Elements Used

Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☑	☑	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Yes

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Yes

Circular to Circular (Spiral)
Not Sure

Advantages

Robust convergence properties and insensitive to initial guess of optimization variables
Much better convergence than SEPSPOT when including additional effects
Results matched SEPSPOT within a few %, better results for solar deg case
Allows for different types of thrusters to be used
Used sequential quadratic programming(SQP) to solve constrained parameter optimizationproblem (note)
NLP solved using gradient-based optimization method (note)
Used extremal feedback control laws to parameterize thrust direction (note)

Disadvantages

Convergence may be slower than indirect methods
Doesn't account for slew rates
What language is code in? FORTRAN? Work in Matlab, STK? Even Available?
Does not do both min transfer time and min delta-v cases
Documentation not extensive, no good description of code
Very similar, but not as impressive as "A Hybrid Method for Computing...", by Mark R. Ilgen (AAS 94-129)

Sofware Developed or Used

Developed Direct Method (DM) software for this techinique (where is it?)
Used NPSOL for SQP code, computes the gradients with both forward and central finite differences
Referenced HYTOP code, by Ilgen, not sure if evolved or used it, looks very familiar though

Reference Source Information

Title

Low-Thrust Orbit Transfer Optimization Using Genetic Search

Author(s)

Larry D. Dewell, P.K. Menon

Source

AIAA Microfiche

Paper ID#

AIAA 99-4151

Year

1999

Reference Summary

Optimization Method

Genetic Search

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☑	☑	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Yes

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Not Sure

Advantages

Genetic search is set of directed, discontinuous search methods inspired by biological genetics &evolution
Non-gradient based solution, ideal for optimization with non-smooth dynamics or performance measures
Software runs in Matlab (extensive tools for genetic search are available for Matlab, used here)
May yield results where gradient methods fail

Disadvantages

Not good for system with
  smooth partial derivative of the dynamics and cost function
  known switching structure of any discontinuous controls
  reasonable initial guess or the solution to a closely related problem is know (for all, gradient better)
Results are often sub-optimal
No extenisve research yet on this method
Not much extra included in the optimization, if even possible

Sofware Developed or Used

Developed some matlab code created for transfer using below
Used Genetic Search Toolbox for Matlab

Reference Source Information

Title

Optimal Low Thrust Geocentric Transfer

Author(s)

Theodore Edelbaum, Lester Sackett, Harvey Malchow

Source

AIAA

Paper ID#

N/A

Year

1973

Reference Summary

Optimization Method

Calculus of Variations (Newton-Raphson iteration)

Orbital Elements Used

Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☑	☑	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Yes

Elliptic to Elliptic

Yes

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Yes

Advantages

Includes oblateness and shadowing
Avoids singularities
Does averaging to cut computation time
Goes between any orbit
Paper is well detailed with all equations listed
Good example of a basic optimizer with most of the extras, similar to SEPSPOT

Disadvantages

Very old technology (1973)
No min-delta V calculation
Probably hard to find the software now, plus an old compiler

Sofware Developed or Used

Written in FORTRAN IV
Software developed for NASA Goddard

Reference Source Information

Title

Low Thrust OTV Guidance Using Lyapunov Optimal Feedback Control Techniques

Author(s)

Marc Ilgen

Source

AAS, Aerospace Corp.

Paper ID#

AAS 93-680

Year

1993

Reference Summary

Optimization Method

Lyapunov Optimal Feedback Control

Orbital Elements Used

Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☑	☑	☐	☑

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)	Elliptic to Elliptic
Yes	Yes
Circular to Elliptic to Circular	**Circular to Circular (Spiral)**
Not Sure	Yes

Advantages

Guidance law operates by determing at each point the thrust direction that minimzes a scalar function composed of the dot product of the gradient of the Lyapunov function and the vector of vehicle dynamics (note).
Mostly analytic, easy to compute, closed-loop, can be used onboard sat in realtime & adjusts for offsets
Guarantees the end values are eventually reached, even with thrust offsets, unmodeled dynamics
Works between any values of a,e,i, and includes second order effects
Can be extended to achieve needed values of RAAN, argument of perigee, and slew rates

Disadvantages

Approximation on the exact optimal control law can get from calc of var, but close and simple
No information is given on the software developed
Not sure how eclipse, J2, and degradation where included, just mentions they were
Only two test cases shown
Aerospace corp. is very protective of distributing software or info without contract

Sofware Developed or Used

No information given

Reference Source Information

Title

Low-Thrust Inclination Control in Presence of Earth Shadow

Author(s)

Jean Albert Kechichian, Aerospace Corp.

Source

Journal of Spacecraft and Rockets
Vol 35, No 4, July-August 1998

Paper ID#	Year
AAS/AIAA 91-157	1998

Reference Summary

Optimization Method

Analytic

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☐	☑	☑	☑	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)	Elliptic to Elliptic
Not Sure	Not Sure
Circular to Elliptic to Circular	**Circular to Circular (Spiral)**
Not Sure	Yes

Advantages

Analytic approach to inclinination change only with EP from circular orbits
Developes algorithms to calculate out of plane thrust profiles
Two methods of switching thrust angle between eclipses
Uses linearized variation of parameters approach

Disadvantages

Sub-optimal results
Only analytic, doesn't seem to have made any software
Circular to circular cases only
Simplistic, low-fidelity approach

Sofware Developed or Used

N/A

Reference Source Information

Title

Orbit Raising with Low-Thrust Tangential Acceleration in Presence of Earth Shadow

Author(s)

Jean Albert Kechichian, Aerospace Corp.

Source

Journal of Spacecraft and Rockets
Vol 35, No. 4, July-August 1998

Paper ID#

AAS/AIAA 91-513

Year

1998

Reference Summary

Optimization Method

Analytic

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☐	☐	☑	☑	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Not Sure

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Yes

Advantages

Analytic analysis of circular to circular orbit raising with shadow effects
Low computational time (if implemented)
Well documented equations

Disadvantages

No software developed
Sub-optimal results
Simplistic analysis, not flexible to all start and end scenarios

Sofware Developed or Used

N/A

Reference Source Information

Title

Optimal Low-Thrust LEO to GEO/HEO Trajectories

Author(s)

A.L. Herman and D.B. Spencer

Source

Advances in the Astronautical Sciences
Vol 103 Pt. II
Astrodynamics 1999

Paper ID#

AAS 99-408

Year

1999

Reference Summary

Optimization Method

Higher-Order Collocation (HOC)

Orbital Elements Used

Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Not Sure

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Yes

Circular to Circular (Spiral)
Yes

Advantages

The burn-coast-burn profile can be determined a priori and input into the optimization
Good reference for equinoctial element definition and equations

Disadvantages

Not stated that the optimization includes any second order effects (J2, eclipses, etc)
Results only show circular to circular or circular to elliptic to circular transfers

Sofware Developed or Used

Software was developed, but no details are given, other than the use of HOC for the NLP problem

Reference Source Information

Title

Simplified Approach for Assessment of Low-Thrust Elliptical Orbit Transfers

Author(s)

James E. Pollard, Aerospace Corp.

Source

International Electric Propulsion Conference

Paper ID#

IEPC-97-160

Year

1997

Reference Summary

Optimization Method

Calculus of Variations

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☑	☑	☐	☑	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Yes

Elliptic to Elliptic

Yes

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Not Sure

Advantages

Good reference for rates of change of standard (classical) orbital elements
Computationally quick, but sub optimal
Uses simple steering laws to complete optimization

Disadvantages

Not optimal results
Lacking some second order effects
Mostly analytic, no detailed description of software developed

Sofware Developed or Used

N/A

Reference Source Information

Title

Evaluation of Low-Thrust Orbital Maneuvers

Author(s)

J.E. Pollard

Source

JPC
July 13-15, 1998

Paper ID#

AIAA-98-3486

Year

1998

Reference Summary

Optimization Method

Calculus of Variations

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☑	☑	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Yes

Elliptic to Elliptic
Yes

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Not Sure

Advantages

Minimizes delta V x delta t to get good balance between the two
Good evaluation of different scenarios

Disadvantages

Lacks description of optimization method or program used
Not sure if any second order terms were used

Sofware Developed or Used

N/A

Reference Source Information

Title

Low-Thrust Maneuvers for LEO and MEO Missions

Author(s)

James E Pollard, Aerospace Corp.

Source

JPC
20-24 June 1999

Paper ID#

AIAA-99-2870

Year

1999

Reference Summary

Optimization Method

Analytic

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☐	☐	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Not Sure

Elliptic to Elliptic

Not Sure

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Yes

Advantages

Analyzes LEO and MEO missions, including drag, repostioning, timing, RAAN shifts

Disadvantages

Interesting generic results, but does not apply to orbit raising to GEO

Sofware Developed or Used

N/A

Reference Source Information

Title

Launch Vehicle and Power Level Impacts on Electric GEO Insertion

Author(s)

S.R. Oleson and R.M. Myers

Source

JPC
july 1-3 1996

Paper ID#

AIAA 96-2978

Year

1996

Reference Summary

Optimization Method

SEPSPOT

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☑	☑	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Yes

Elliptic to Elliptic
Yes

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Yes

Advantages

Puspose of paper is to compare the effects of different launch vehicles and power levels on orbit raising
Compares chemical and electric trade-offs using different lauch vehicles

Disadvantages

Uses SEPSPOT, does not develop own code

Sofware Developed or Used

Used SEPSPOT

Reference Source Information

Title

Optimization of Low Thrust Transfer between Noncoplanar Elliptic Orbits

Author(s)

M.S. Konstantinov

Source

48th International Astronautical Congress
October 6-10, 1997

Paper ID#

IAF-97-A.6.06

Year

1997

Reference Summary

Optimization Method

Calculus of Variations (Pontryagin's Max and Ave Principle)

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☐	☐	☐	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Yes

Elliptic to Elliptic

Yes

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Not Sure

Advantages

Algorithm is supposedly fast

Disadvantages

Hard to understand and follow, written by Russian
Equations and complex and difficult to understand

Sofware Developed or Used

Algorithm briefly described to solve optimization problem

Reference Source Information

Title

Simple Control Laws for Low-Thrust Orbit Transfers

Author(s)

Craig A. Kluever

Source

Advances in Astronautical Sciences
Vol 99, Pt. 2, 1998, pg 1455-1468

Paper ID#

AAS 98-203

Year

1998

Reference Summary

Optimization Method

Simple Optimal Feedback Control

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Not Sure

Elliptic to Elliptic

Not Sure

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Yes

Advantages

Can be used for onboard real-time control of S/C
Control laws are simple and easy to compute
Can do apse-line control (at least in a planar transfer)

Disadvantages

Suboptimal results, but meant to be fast guidance technique, not exact optimum technique
No mention of second order effects except eclipsing
Not sure if it can any other transfer orbit types (other than spiral)
Very few examples shown
Guidance parameters were selected via trial and error

Sofware Developed or Used

Does not go into detail on any software developed

Reference Source Information

Title

Maximum-Payload Transfers to Geosynchronous Orbit Using Arcjet Thrusters

Author(s)

Craig A. Kluever

Source

Journal of Spacecraft and Rockets
Vol 34, No 3, Engineering Notes
Pg. 405-407

Paper ID#

N/A

Year

1997

Reference Summary

Optimization Method

SEPSPOT

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☑	☑	☑	☑	☑	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)	Elliptic to Elliptic
Yes	Yes
Circular to Elliptic to Circular	**Circular to Circular (Spiral)**
Not Sure	Yes

Advantages

Compares combinations of electric and chemical propulsion to find an optimal mix
Tries to maximize payload to orbit
Good reference for example combined chem/electric transfer results

Disadvantages

Does not develop any new optimum method, just uses SEPSPOT

Sofware Developed or Used

SEPSPOT

Reference Source Information

Title

Low-Thrust Orbit Transfer Guidance Using an Inverse Dynamics Approach

Author(s)

Craig A. Kluiver

Source

Journal of Guidance, Control and Dynamics
Vol 18, No. 1
Jan-Feb 1995, pg 187-189

Paper ID#

Year

0

Reference Summary

Optimization Method

Inverse Dynamics Control

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☑	☑	☐	☐	☑	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)
Not Sure

Elliptic to Elliptic
Not Sure

Circular to Elliptic to Circular
Not Sure

Circular to Circular (Spiral)
Yes

Advantages

Can be used for onboard real-time control of S/C

Disadvantages

Suboptimal
Not sure if secondary effects are included (other than eclipsing)

Sofware Developed or Used

SEPSPOT used as a reference trajectory

Reference Source Information

Title

Near-Optimum Low-Thrust Transfer in Semi-Major Axis and Eccentricity

Author(s)

T.A. Bauer

Source

Paper ID#

AAS 92-134

Year

1992

Reference Summary

Optimization Method

Calculus of Variations

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☐	☐	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Yes

Elliptic to Elliptic

Not Sure

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Yes

Advantages

Good resource for the basic calculus of variation equations

Disadvantages

Near-optimum, not exact
Very basic calculus of variations approach
No inclination change
No second order effects

Sofware Developed or Used

N/A

Reference Source Information

Title

Time-Critical Low-Thrust Orbit Transfer Optimization

Author(s)

Rodney Burton, Carl Wassgren

Source

Journal of Spacecraft and Rockets
Vol 29, No 2, March-April 1992

Paper ID#

AIAA 90-2620

Year

1992

Reference Summary

Optimization Method

Analytic

Orbital Elements Used

Standard

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☑	☑	☐	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)

Not Sure

Elliptic to Elliptic

Not Sure

Circular to Elliptic to Circular

Not Sure

Circular to Circular (Spiral)

Not Sure

Advantages

Optimizes transfer rate and includes OTV effects
Good reference for systems level optimizations

Disadvantages

Basic analysitic approach
No second order terms visible
No software developed

Sofware Developed or Used

N/A

Reference Source Information

Title

A Set of Modified Equinoctial Orbit Elements

Author(s)

Walker, Ireland, Owens

Source

Celestial Mechanics 36
1985, 409-419

Paper ID#

N/A

Year

1985

Reference Summary

Optimization Method

Analytic

Orbital Elements Used

Equinoctial

Included in Optimization

Min Transfer Time	Min Delta-V	Inclination Change	Eclipse Effects	Oblateness (J2)	Solar Cell Degradation	Orbital Averaging	Slew Rates
☐	☐	☐	☐	☐	☐	☐	☐

Transfer Orbit Types Possible

Elliptic to Circular (GTO-GEO)	Elliptic to Elliptic
No	No
Circular to Elliptic to Circular	**Circular to Circular (Spiral)**
No	No

Advantages

Reference for definitions of the equinoctial elements

Disadvantages

N/A

Sofware Developed or Used

N/A

Appendix B: Optimization Software Code (SS/L Copies Only)

The MITEOR optimization software code is only available for Space Systems / Loral copies of this thesis, and is not included for MIT copies. The software is considered proprietary and should not be used or copied without direct consent of Space Systems / Loral.