Optimal Low Thrust Orbit Transfers with Eclipsing

John T. Betts *

January 14, 2014

Abstract

By choosing the optimal steering history of a spacecraft it is possible to maximize the mass delivered from a park orbit to a mission orbit. A low-thrust orbit transfer that models coasts when passing through the Earth's shadow can be formulated as a large scale optimal control problem with many distinct phases. This paper presents a technique that constructs an initial guess using a receding horizon algorithm. A series of large scale multiphase optimal control problems are then solved to refine the phase structure of the problem. The final optimal solution incorporates high fidelity physics and mesh refinement techniques within a large sparse nonlinear programming approach.

1 Introduction

Typically there are two steps required to place a satellite in orbit. First, a launch vehicle is used to insert the satellite in a low altitude park orbit, which is followed by an orbit transfer to the final mission orbit. The majority of operational satellites have utilized high thrust chemical propulsion systems to perform the orbit transfer function. A high thrust orbit transfer is characterized by relatively short transfer times, e.g. approximately six hours for a geosynchronous mission. However, high thrust propulsion systems are relatively inefficient, often inserting a payload in the mission orbit that is less than ten percent the weight in the park orbit. The opposite occurs when using modern low thrust propulsion systems, namely greatly improved fuel efficiency at the expense of much longer transfer times. Solar electric propulsion systems deliver these performance benefits, provided the vehicle is located in a region of sunlight so that the engine can operate. When the spacecraft passes through the Earth's shadow (eclipsing), no thrust is generated, and the trajectory design must address this behavior. Furthermore, the location of the shadow changes as a function of time, and since the satellite trajectory may or may not pass through the shadow, the impact of eclipse on the trajectory design can be very significant. Because the thrust acceleration is so small, it is important to include other small perturbations in the dynamic model as well. Finally, the impact of any small force early in the trajectory is greatly amplified because the duration of the transfer may take many months. In short, an optimal low thrust transfer with eclipsing is a a very challenging computational problem.

Techniques for addressing the challenging issues in low thrust trajectory design have been considered for many years. Early work in the field by Edelbaum [9] uses variational calculus to derive steering laws that model thrust as small perturbations to Keplerian orbital elements. Many other authors have proposed similar ideas that address the computational challenges by introducing simplifications to the physics of the problem. Unfortunately for long duration trajectories it is important to include small perturbing forces and so techniques that employ approximate physical models become problematic. A second class of methods that has been proposed utilizes a closed form feedback control law derived from a Lyapunov function. Early work by Ilgen [13, 14], as well as more recent contributions by Petropoulos [19], and Chang, et.al. [8] illustrate this approach. While these techniques often provide useful results incorporating coast phases and higher order perturbations often rely on heuristics. Furthermore the final trajectory may not be optimal and can diverge from the desired mission orbit. Classical indirect shooting methods have also been applied which require analytic derivation of the relevant adjoint equations, and this has been achieved in a series of contributions by Kechichian [17]. Ferrier and Epenoy [11] incorporate a simplified eclipse model in order to avoid discontinuous behavior when entering and leaving shadow regions. However, as with any indirect method it is often difficult to construct an initial guess, include high fidelity physical models (e.g. planetary ephemerides), and incorporate the switching required to model coast phases. Lastly, many authors have proposed direct methods that combine either simplified physics and/or a simplified control law.

The new method proposed here eliminates the shortcomings of all previous techniques. High fidelity physical models can be utilized. A robust technique for constructing an initial guess is exploited to compute

*Applied Mathematical Analysis, LLC

an accurate optimal steering history. Lastly, the algorithm can be used to correct unexpected errors by optimal (real-time) mission re-targeting.

2 Dynamic Model

The motion of a vehicle can be described by a system of second-order ODEs

$$\ddot{\mathbf{r}} + \mu \frac{\mathbf{r}}{r^3} = \mathbf{a}_d, \tag{2.1}$$

where the radius $r = \|\mathbf{r}\|$ is the magnitude of the inertial position vector $\mathbf{r}$ and μ is the gravitational constant. In this formulation referred to as Gauss' form of the variational equations, the vector $\mathbf{a}_d$ is defined as the *disturbing acceleration*. The Gauss' form isolates the disturbing acceleration from the central force gravitational acceleration. When the disturbing acceleration is zero $\|\mathbf{a}_d\| = 0$ the fundamental system (2.1) is a two-body problem whose solution can be stated in terms of constant orbital elements. For low thrust trajectories this formulation is appealing because we expect $\|\mathbf{a}_d\|$ to be "small" and consequently we expect that the solution can be described in terms of "almost constant" orbital elements. In order to exploit the benefits of the variational form of the differential equations (2.1) it is necessary to transform the Cartesian state into an appropriate set of orbit elements. Classical elements exhibit singularities for zero eccentricity, and inclinations of 0° and 90°. Furthermore the semimajor axis changes discontinuously for orbits with eccentricity equal to one. To eliminate these deficiencies, a modified set of equinoctial orbit elements is described in [2] based on the work in [20], and many other applications of this approach can be found in references [3, 4, 6, 7, 15, 16, 21].

2.1 Modified Equinoctial Coordinates

It is possible to express the dynamics (2.1) in terms of modified equinoctial elements

$$\mathbf{y}^{\mathsf{T}} = [p, f, g, h, k, L]. \tag{2.2}$$

Using modified equinoctial elements with state $\mathbf{y}$, control $\mathbf{u}$, and parameters $\mathbf{p}$, the equations of motion are

$$\dot{\mathbf{y}} = \mathbf{A}(\mathbf{y})\boldsymbol{\Delta} + \mathbf{b} = \widetilde{\mathbf{f}}(\mathbf{y}, \mathbf{u}, \mathbf{p}, t). \tag{2.3}$$

The equinoctial dynamics are defined by the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{2p}{q}\sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}}\sin L & \sqrt{\frac{p}{\mu}}\frac{1}{q}\{(q+1)\cos L + f\} & -\sqrt{\frac{p}{\mu}}\frac{g}{q}\{h\sin L - k\cos L\} \\ -\sqrt{\frac{p}{\mu}}\cos L & \sqrt{\frac{p}{\mu}}\frac{1}{q}\{(q+1)\sin L + g\} & \sqrt{\frac{p}{\mu}}\frac{f}{q}\{h\sin L - k\cos L\} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\cos L}{2q} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2\sin L}{2q} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{1}{q}\{h\sin L - k\cos L\} \end{bmatrix} \tag{2.4}$$

and the vector

$$\mathbf{b}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p}\left(\frac{q}{p}\right)^2 \end{bmatrix}, \tag{2.5}$$

where

$$q = 1 + f\cos L + g\sin L, \tag{2.6}$$
$$r = \frac{p}{q}, \tag{2.7}$$
$$\alpha^2 = h^2 - k^2, \tag{2.8}$$
$$\chi = \sqrt{h^2 + k^2}, \tag{2.9}$$
$$s^2 = 1 + \chi^2. \tag{2.10}$$

It is frequently useful to impose bounds on the variables:

$$p \in [p_0, \infty) \qquad f \in [-1.1, 1.1] \tag{2.11}$$
$$g \in [-1.1, 1.1] \qquad h \in [-1.1, 1.1] \tag{2.12}$$
$$k \in [-1.1, 1.1] \tag{2.13}$$

The equinoctial coordinates $\mathbf{y}$ are related to the Cartesian state $(\mathbf{r}, \mathbf{v})$ according to the expressions

$$\mathbf{r} = \begin{bmatrix} \frac{r}{s^2}\left(\cos L + \alpha^2 \cos L + 2hk \sin L\right) \\ \frac{r}{s^2}\left(\sin L - \alpha^2 \sin L + 2hk \cos L\right) \\ \frac{2r}{s^2}\left(h \sin L - k \cos L\right) \end{bmatrix}, \tag{2.14}$$

$$\mathbf{v} = \begin{bmatrix} -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(\sin L + \alpha^2 \sin L - 2hk \cos L + g - 2fhk + \alpha^2 g\right) \\ -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(-\cos L + \alpha^2 \cos L + 2hk \sin L - f + 2ghk + \alpha^2 f\right) \\ \frac{2}{s^2}\sqrt{\frac{\mu}{p}}\left(h \cos L + k \sin L + fh + gk\right) \end{bmatrix}. \tag{2.15}$$

As a result of this transformation, the disturbing acceleration vector $\mathbf{a}_d$ in (2.1) is replaced by

$$\boldsymbol{\Delta} = \boldsymbol{\Delta}_g + \boldsymbol{\Delta}_q + \boldsymbol{\Delta}_\mathsf{T} \tag{2.16}$$

with contributions due to oblate earth effects $\boldsymbol{\Delta}_g$, secondary bodies $\boldsymbol{\Delta}_q$, and thrust $\boldsymbol{\Delta}_\mathsf{T}$. In general the perturbing accelerations are functions of both state $\mathbf{y}$ and time t, although the functional dependence is suppressed. A more detailed description of these perturbations is given in Section 3. The disturbing acceleration is expressed in a rotating radial frame whose principal axes are defined by

$$\mathbf{Q} = \begin{bmatrix} \mathbf{i}_r & \mathbf{i}_\theta & \mathbf{i}_h \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{r}}{\|\mathbf{r}\|} & \frac{(\mathbf{r}\times\mathbf{v})\times\mathbf{r}}{\|\mathbf{r}\times\mathbf{v}\|\|\mathbf{r}\|} & \frac{\mathbf{r}\times\mathbf{v}}{\|\mathbf{r}\times\mathbf{v}\|} \end{bmatrix}. \tag{2.17}$$

For comparison the typical magnitude range of the perturbing accelerations is given in Table 2.1

Table 2.1: Perturbing Acceleration Range Comparison

μ/r^2	$\|\boldsymbol{\Delta}_\mathsf{T}\|$	$\|\boldsymbol{\Delta}_g\|$	$\|\boldsymbol{\Delta}_q\|$
$[3\times10^{-5}, 9\times10^{-3}]$	$[7\times10^{-7}, 2\times10^{-4}]$	$[3\times10^{-10}, 1\times10^{-5}]$	$[4\times10^{-11}, 2\times10^{-8}]$

As stated, (2.3) are perfectly general and describe the motion of a point mass when subject to the disturbing acceleration vector $\boldsymbol{\Delta}$. Notice that when the disturbing acceleration is zero, $\boldsymbol{\Delta} = 0$, the first five equations are simply $\dot{p} = \dot{f} = \dot{g} = \dot{h} = \dot{k} = 0$, which implies that the elements are constant. It is important to note that the disturbing acceleration vector can be attributed to any perturbing force(s). A more complete derivation of the equinoctial dynamics can be found in [2].

2.2 True Longitude As The Independent Variable

Low thrust transfers are characterized by very long transfer times, and consequently it is convenient to restate the dynamics using another independent variable. As written (2.2) consists of the dynamic variables $\mathbf{y} = [p, f, g, h, k, L]$ viewed as functions of the independent variable t. However it is convenient to treat the true longitude variable L in (2.2) as the *independent* variable. Physically this represents an angle in the orbit plane and consequently one complete orbit revolution or "rev" corresponds to a change of 2π in the variable L. Now

$$\frac{dL}{dt} = \tilde{f}_6(\mathbf{y}, \mathbf{u}, \mathbf{p}, t) \tag{2.18}$$

and so it follows that

$$t' = \frac{dt}{dL} = \frac{1}{\tilde{f}_6(\mathbf{y}, \mathbf{u}, \mathbf{p})} \tag{2.19}$$

where differentiation with respect to L is denoted by a "prime." Thus if we divide each equation in the original system (2.3) by the quantity $\sigma = \tilde{f}_6(\mathbf{y}, \mathbf{u}, \mathbf{p})$ we obtain the following system of ordinary differential equations

$$p' = \tilde{f}_1(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \tag{2.20}$$

$$f' = \tilde{f}_2(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \tag{2.21}$$

$$g' = \tilde{f}_3(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \tag{2.22}$$

$$h' = \tilde{f}_4(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \tag{2.23}$$

$$k' = \tilde{f}_5(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \tag{2.24}$$

$$t' = 1/\sigma \tag{2.25}$$

Collecting results, we can write

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{p}, L) \tag{2.26a}$$

where $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$

$$f_k = \tilde{f}_k(\mathbf{y}, \mathbf{u}, \mathbf{p})/\sigma \qquad k \neq 6 \tag{2.26b}$$

$$f_k = 1/\sigma \qquad k = 6 \tag{2.26c}$$

for $k = 1, \ldots, 6$. Observe that in this formulation we can write $t(L)$ since time t is treated as a state variable, whereas the angle L is treated as the independent variable (of integration). Clearly this transformation is valid provided $\frac{dL}{dt} > 0$ and "time dependent" quantities are in fact angle dependent.

3 Perturbing Accelerations

3.1 Gravitational Disturbing Acceleration

Gravitational models that represent oblate Earth affects are typically defined in a local horizontal reference frame, that is,

$$\boldsymbol{\delta}\mathbf{g} = \delta g_n \mathbf{i}_n - \delta g_r \mathbf{i}_r, \tag{3.1}$$

where

$$\mathbf{i}_n = \frac{\mathbf{e}_n - (\mathbf{e}_n^\mathsf{T}\mathbf{i}_r)\mathbf{i}_r}{\|\mathbf{e}_n - (\mathbf{e}_n^\mathsf{T}\mathbf{i}_r)\mathbf{i}_r\|} \tag{3.2}$$

defines the local north direction with $\mathbf{e}_n = (0, 0, 1)$. A reasonably accurate model is obtained if the tesseral harmonics are ignored and only the first four zonal harmonics are included in the geopotential function. In this case, the oblate earth perturbations to the gravitational acceleration are given by

$$\delta g_n = -\frac{\mu \cos\phi}{r^2} \sum_{k=2}^{4} \left(\frac{R_e}{r}\right)^k P_k' J_k, \tag{3.3}$$

$$\delta g_r = -\frac{\mu}{r^2} \sum_{k=2}^{4} (k+1) \left(\frac{R_e}{r}\right)^k P_k J_k, \tag{3.4}$$

where ϕ is the geocentric latitude, R_e is the equatorial radius of the earth, $P_k(\sin\phi)$ is the kth-order Legendre polynomial with corresponding derivative P_k', and J_k are the zonal harmonic coefficients. Finally, to obtain the gravitational perturbations in the rotating radial frame, it follows that

$$\boldsymbol{\Delta}_g = \mathbf{Q}^\mathsf{T}\boldsymbol{\delta}\mathbf{g}. \tag{3.5}$$

3.2 Secondary Body Acceleration

It has been demonstrated in [2] that secondary bodies introduce perturbing accelerations. For this study it suffices to consider perturbations caused by the Sun and Moon

$$\boldsymbol{\delta}\mathbf{q} = -\mu_\mathsf{s}\left[\frac{\mathbf{d}_\mathsf{s}}{d_\mathsf{s}^3} + \frac{\mathbf{s}_\mathsf{s}}{s_\mathsf{s}^3}\right] - \mu_\mathsf{m}\left[\frac{\mathbf{d}_\mathsf{m}}{d_\mathsf{m}^3} + \frac{\mathbf{s}_\mathsf{m}}{s_\mathsf{m}^3}\right], \tag{3.6}$$

where $\mathbf{s}_\mathsf{s}$ is a vector from the Earth to the Sun, $\mathbf{s}_\mathsf{m}$ is a vector from the Earth to the Moon, and

$$\mathbf{d}_\mathsf{s} = \mathbf{r} - \mathbf{s}_\mathsf{s} \tag{3.7}$$

$$\mathbf{d}_\mathsf{m} = \mathbf{r} - \mathbf{s}_\mathsf{m} \tag{3.8}$$

are vectors from the secondary bodies to the vehicle. The gravitational constants of the Sun and Moon are denoted by μ_s and μ_m respectively, and the vector norm is denoted by a scalar (e.g. $s_\mathsf{s} = \|\mathbf{s}_\mathsf{s}\|$). Calculation of the gravitational disturbances due to the Sun and Moon can be achieved directly from (3.6). However Battin [1] has shown that this calculation is prone to cancellation because of the significantly different size of the terms involved. Instead he suggests defining the functions

$$F(q_\mathsf{s}) = q_\mathsf{s}\left[\frac{3 + 3q_\mathsf{s} + q_\mathsf{s}^2}{1 + (\sqrt{1+q_\mathsf{s}})^3}\right] \qquad q_\mathsf{s} = \frac{\mathbf{r}^\mathsf{T}(\mathbf{r} - 2\mathbf{s}_\mathsf{s})}{\mathbf{s}_\mathsf{s}^\mathsf{T}\mathbf{s}_\mathsf{s}}. \tag{3.9}$$

$$F(q_\mathsf{m}) = q_\mathsf{m}\left[\frac{3 + 3q_\mathsf{m} + q_\mathsf{m}^2}{1 + (\sqrt{1+q_\mathsf{m}})^3}\right] \qquad q_\mathsf{m} = \frac{\mathbf{r}^\mathsf{T}(\mathbf{r} - 2\mathbf{s}_\mathsf{m})}{\mathbf{s}_\mathsf{m}^\mathsf{T}\mathbf{s}_\mathsf{m}}. \tag{3.10}$$

Substituting into (3.6) yields

$$\boldsymbol{\delta}\mathbf{q} = -\frac{\mu_{\mathsf{s}}}{d_{\mathsf{s}}^3}\left[\mathbf{r} + F(q_{\mathsf{s}})\mathbf{s}_{\mathsf{s}}\right] - \frac{\mu_{\mathsf{m}}}{d_{\mathsf{m}}^3}\left[\mathbf{r} + F(q_{\mathsf{m}})\mathbf{s}_{\mathsf{m}}\right]. \tag{3.11}$$

Finally to obtain the Sun/Moon perturbations to the acceleration in the rotating radial frame it follows that:

$$\boldsymbol{\Delta}_q = \mathbf{Q}^{\mathsf{T}}\boldsymbol{\delta}\mathbf{q} \tag{3.12}$$

Calculation of the perturbing forces requires an ephemeris [12] which specifies the position and velocity of the moon and sun as a function of time.

3.3 Thrust Acceleration

To this point, the discussion has concentrated on incorporating perturbing forces due to oblate earth and secondary body effects. Of course, the major perturbation is the thrust acceleration defined by

$$\boldsymbol{\Delta}_{\mathsf{T}} = a_{\mathsf{T}}\mathbf{d}, \tag{3.13}$$

where the magnitude of the thrust acceleration is given by $a_{\mathsf{T}} = T/m$ (m/sec^2). In general, the mass flow rate is defined by the differential equation

$$\dot{m} = -T/c_{\mathsf{EX}}, \tag{3.14}$$

with the exhaust velocity defined as $c_{\mathsf{EX}} = g_0 I_s$ (m/sec) where I_s is the specific impulse and $g_0 = 9.80665$ (m/sec^2). When the thrust and exhaust velocity are constant, the mass flow rate $\dot{m}$ is also constant, and the mass flow differential equation (3.14) can be integrated analytically to give

$$m(t) = m_I + \dot{m}(t - t_I) \tag{3.15}$$

where $m_I = m(t_I)$ is the mass evaluated at time t_I. Thus the mass flow differential equation (3.14) can be replaced by the analytic expression (3.15). Observe that the analytic expression for mass is defined by the state variable t, the constant $\dot{m}$ and two parameters t_I and m_I. Thus in (3.13)

$$a_{\mathsf{T}} = T[m_I + \dot{m}(t - t_I)]^{-1}. \tag{3.16}$$

The vector $\mathbf{d}$ can be represented using either Cartesian or spherical coordinates where

$$d_r = d\sin\alpha \tag{3.17a}$$
$$d_\theta = d\cos\alpha\cos\beta \tag{3.17b}$$
$$d_h = d\cos\alpha\sin\beta \tag{3.17c}$$

and the inverse transformation is

$$d = \sqrt{d_r^2 + d_\theta^2 + d_h^2} \tag{3.17d}$$
$$\alpha = \sin^{-1}\left(\frac{d_r}{d}\right) \tag{3.17e}$$
$$\beta = \tan^{-1}\left(\frac{d_h}{d_\theta}\right). \tag{3.17f}$$

Here α is a thrust pitch angle measured in the orbit plane off of the circumferential direction, positive away from the center of gravity, and β is a thrust yaw angle measured off of the orbit plane and perpendicular to it, positive in the direction of the angular momentum vector.

In general, the time-varying vector $\mathbf{d}(t) = (d_r, d_\theta, d_h)$ in the rotating radial frame, can be chosen arbitrarily as long as it has unit length at all points in time. One approach is to explicitly treat the vector $\mathbf{d} = \mathbf{u}$ as controls subject to the additional path constraint $\|\mathbf{d}(t)\| = 1$. In this case we have

$$\boldsymbol{\Delta}_{\mathsf{T}} = a_{\mathsf{T}}\mathbf{u} \tag{3.18}$$

subject to the path constraint

$$\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1. \tag{3.19}$$

A second alternative is to define the thrust direction vector in the rotating radial frame using the two angles α and β and then it is natural to choose the control variables $\mathbf{u}^\mathsf{T} = [\alpha, \beta] = [u_1, u_2]$. With this formulation the thrust acceleration is

$$\mathbf{\Delta}_\mathsf{T} = a_\mathsf{T} \begin{bmatrix} \sin u_1 \\ \cos u_1 \cos u_2 \\ \cos u_1 \sin u_2 \end{bmatrix}. \tag{3.20}$$

While both the vector formulation (3.18)-(3.19) and the angle formulation (3.20) are equivalent representations of the physical process, there are computational distinctions between the two techniques. When using the angles (3.20) it is common for the iterative process to introduce "wrapping." This can occur because angular quantities are periodic, and the direction defined by an angle β is the same as the direction for an angle $\beta \pm 2\pi$. This "wrapping" behavior commonly occurs during optimization iterations from a poor initial guess. To reduce the prospect of wrapping we restrict the angles to:

$$-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2} \tag{3.21}$$

$$-\pi \leq \beta \leq \pi \tag{3.22}$$

Observe that the bounds (3.22) will be problematic if the optimal solution requires retrograde thrust. In contrast, when using the vector formulation (3.18) the direction is uniquely defined and wrapping cannot occur. However, there is a computational penalty incurred for improved robustness; it is necessary to introduce an additional control variable and path constraint (3.19) thereby increasing the problem size in comparison to the angle formulation. In short, the vector formulation is more robust than the angle formulation, but the angle formulation may be more efficient computationally.

As a practical matter limits are often imposed on the angular rate of change in the thrust direction vector. For a short time interval δ the approximate rotation angle θ between two unit direction vectors is defined by

$$\mathbf{d}^\mathsf{T}(t)\mathbf{d}(t+\delta) = \cos\theta. \tag{3.23}$$

It then follows that an approximate rotation rate is given by

$$\dot{\theta} \approx \frac{1}{\delta} \cos^{-1}\left[\mathbf{d}^\mathsf{T}(t)\mathbf{d}(t+\delta)\right]. \tag{3.24}$$

3.4 Coast Arcs

When the thrust acceleration is zero $\mathbf{\Delta}_\mathsf{T} = 0$ the dynamic model can be simplified. The dynamics are completely described by the state $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$ and since there is no thrust, there are no control variables $\mathbf{u}$ or parameters $\mathbf{p}$. Collecting results we can describe the motion by

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, L). \tag{3.25}$$

4 Eclipse Effects

To accurately model the solar electric propulsion system it is necessary to model the trajectory as it passes through the shadow of the Earth. In particular there is no solar energy when in the umbral shadow, and only partial energy when in the penumbral region. Thus when the vehicle is either in the umbra or penumbra, the thrust is zero $T = 0$, and $\mathbf{\Delta}_\mathsf{T} = 0$.

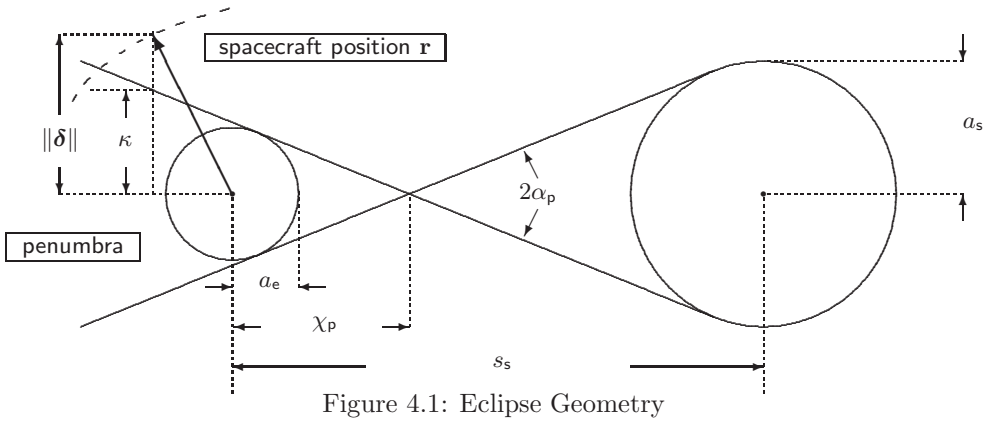

Figure 4.1: Eclipse Geometry

It is reasonable to assume that both the Sun and Earth are spherical bodies as suggested in [10, p. 160] and [18]. Referring to the penumbra geometry illustrated in Figure 4.1 define $\mathbf{s}_\mathsf{s}$ as the vector from the Earth to the Sun, with norm $s_\mathsf{s} = \|\mathbf{s}_\mathsf{s}\|$. The distance from the Earth center to the apex of the penumbral cone is given by

$$\chi_\mathsf{p} = \frac{a_\mathsf{e} s_\mathsf{s}}{(a_\mathsf{s} + a_\mathsf{e})} \tag{4.1}$$

and the penumbra cone angle is defined by

$$\alpha_\mathsf{p} = \sin^{-1}\left(\frac{a_\mathsf{e}}{\chi_\mathsf{p}}\right) = \sin^{-1}\left(\frac{a_\mathsf{s} + a_\mathsf{e}}{s_\mathsf{s}}\right) \tag{4.2}$$

where a_e is the radius of the Earth, and a_s is the radius of the Sun. In a similar fashion, the distance from the Earth center to the apex of the umbral cone is given by

$$\chi_\mathsf{u} = \frac{a_\mathsf{e} s_\mathsf{s}}{(a_\mathsf{s} - a_\mathsf{e})} \tag{4.3}$$

and the umbra cone angle is defined by

$$\alpha_\mathsf{u} = \sin^{-1}\left(\frac{a_\mathsf{e}}{\chi_\mathsf{u}}\right) = \sin^{-1}\left(\frac{a_\mathsf{s} - a_\mathsf{e}}{s_\mathsf{s}}\right). \tag{4.4}$$

It should be noted Figure 4.1 is not drawn to scale, since typical values for $\chi_\mathsf{p} \sim 1337357.1$ km and $\alpha_\mathsf{p} \sim 0.2732567°$. The shadow entry and exit locations are computed by locating the umbral and penumbral cone terminators at the projected spacecraft location. Specifically if we define the solar unit vector

$$\widehat{\mathbf{s}} = \frac{\mathbf{s}_\mathsf{s}}{s_\mathsf{s}} \tag{4.5}$$

the projected spacecraft position is given by

$$\overline{\mathbf{r}} = (\mathbf{r}^\mathsf{T}\widehat{\mathbf{s}})\widehat{\mathbf{s}}. \tag{4.6}$$

Now, observe that a shadow terminator can only be found when $\mathbf{r}^\mathsf{T}\widehat{\mathbf{s}} < 0$. Using this we can define the vector

$$\boldsymbol{\delta} = \mathbf{r} - \overline{\mathbf{r}} \tag{4.7}$$

which extends from the spacecraft to the center of the umbral cone. Finally, let us define the distances between the center of the umbral cone and the penumbra terminator at the projected spacecraft location

$$\kappa = (\chi_\mathsf{p} + \|\overline{\mathbf{r}}\|)\tan\alpha_\mathsf{p}. \tag{4.8}$$

and the distance between the center of the umbral cone and the umbral cone terminator at the projected spacecraft location

$$\xi = (\chi_u - \|\overline{\mathbf{r}}\|) \tan \alpha_u \tag{4.9}$$

The analysis of spacecraft behavior in the presence of solar energy shadows can be summarized by the following rules:

Rule 1 *A shadow terminator can only be encountered if* ..... $\mathbf{r}^T \widehat{\mathbf{s}} < 0$.

Rule 2 *The spacecraft will be in sunlight if* ..... $\|\boldsymbol{\delta}\| > \kappa$.

Rule 3 *The spacecraft will be in the penumbra if* ..... $\xi < \|\boldsymbol{\delta}\| < \kappa$.

Rule 4 *The spacecraft will be in the umbra if* ..... $\|\boldsymbol{\delta}\| < \xi$.

Rule 5 *The penumbra terminator is defined by the condition* ..... $\|\boldsymbol{\delta}\| = \kappa$.

Rule 6 *The umbra terminator is defined by the condition* ..... $\|\boldsymbol{\delta}\| = \xi$.

When locating the penumbra terminator using the rule 5 it is of interest to examine the behavior in the neighborhood of a root

$$E(t) = \|\boldsymbol{\delta}\| - \kappa = 0. \tag{4.10}$$

The slope of this condition can be used to distinguish between entering and leaving a shadow condition, i.e. when *entering* we must have $\dot{E}(t) < 0$ and when *leaving* we must have $\dot{E}(t) > 0$. Clearly, similar conditions must hold when entering and leaving the umbra which is defined by the event condition $\|\boldsymbol{\delta}\| = \xi$.

5 Lyapunov Steering

In [8] Chang, Chichka and Marsden develop a steering law that can be used to construct an initial guess trajectory. Specifically, they describe the dynamics by (2.1) where $\mathbf{a}_d = \mathbf{F}$ is a control force and there are no other perturbing forces. Define the following orbital quantities:

$$\mathcal{L} = \mathbf{r} \times \dot{\mathbf{r}} \tag{5.1}$$

$$\mathcal{A} = \dot{\mathbf{r}} \times (\mathbf{r} \times \dot{\mathbf{r}}) - \mu \frac{\mathbf{r}}{\|\mathbf{r}\|} \tag{5.2}$$

$$\mathcal{E} = \frac{1}{2}\|\dot{\mathbf{r}}\|^2 - \frac{\mu}{\|\mathbf{r}\|} \tag{5.3}$$

where $\mathcal{L}$ is the angular momentum vector, $\mathcal{A}$ is the Laplace vector, and $\mathcal{E}$ is the orbital energy. Note that $\mathcal{A} = \mu \mathbf{e}$ where $\mathbf{e}$ is the eccentricity vector. Let us define

$$\boldsymbol{\Delta}\mathcal{L} = \mathcal{L} - \mathcal{L}_T \tag{5.4}$$

$$\boldsymbol{\Delta}\mathcal{A} = \mathcal{A} - \mathcal{A}_T \tag{5.5}$$

where $\mathcal{L}_T$ is the angular momentum of the target orbit and $\mathcal{A}_T$ is the Laplace vector of the target orbit. As a measure of the error between the target orbit and the current (osculating) orbit, the authors define a Lyapunov function

$$V(\mathbf{r}, \dot{\mathbf{r}}) = \frac{1}{2}k\|\boldsymbol{\Delta}\mathcal{L}\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}\mathcal{A}\|.^2 \tag{5.6}$$

This quantity can be minimized at every instant in time by choosing the control force as follows:

$$\mathbf{F} = -f(\mathbf{r}, \dot{\mathbf{r}}) \left[k\boldsymbol{\Delta}\mathcal{L} \times \mathbf{r} + \mathcal{L} \times \boldsymbol{\Delta}\mathcal{A} + (\boldsymbol{\Delta}\mathcal{A} \times \dot{\mathbf{r}}) \times \mathbf{r}\right]. \tag{5.7}$$

The function $f(\mathbf{r}, \dot{\mathbf{r}}) > 0$ and parameter k are both arbitrary in the original analysis. It can be demonstrated that this control force produces asymptotically stable convergence to the target orbit because $\dot{V} \leq 0$. A second form of the Lyapunov steering law which is applicable for transfers to circular orbits is also proposed. In this case the Lyapunov function being minimized is

$$V(\mathbf{r}, \dot{\mathbf{r}}) = \frac{1}{2}k\|\boldsymbol{\Delta}\mathcal{L}\|^2 + \frac{1}{2}(\Delta\mathcal{E})^2 \tag{5.8}$$

leading to the control force

$$\mathbf{F} = -f(\mathbf{r}, \dot{\mathbf{r}})\,[k\boldsymbol{\Delta}\mathcal{L} \times \mathbf{r} + \Delta\mathcal{E}\dot{\mathbf{r}}] \tag{5.9}$$

where the energy change is

$$\Delta\mathcal{E} = \mathcal{E} - \mathcal{E}_T. \tag{5.10}$$

It then follows that a unit vector in the direction of $\mathbf{F}$ transformed to the rotating radial frame can be used to define a control vector

$$\mathbf{u} = \mathbf{Q}^{\mathsf{T}} \frac{\mathbf{F}}{\|\mathbf{F}\|}, \tag{5.11}$$

which can be used as an initial guess in (3.18)-(3.19). Choosing the parameter $k = 2$ and normalizing the Lyapunov functions (5.6) and (5.8) yield the functions

$$\widehat{V}(\mathbf{r}, \dot{\mathbf{r}}) = \frac{\|\boldsymbol{\Delta}\mathcal{L}\|^2}{\|\mathcal{L}_T\|^2} + \frac{1}{2}\frac{\|\boldsymbol{\Delta}\mathcal{A}\|^2}{\|\mathcal{A}_T\|^2} \tag{5.12a}$$

$$\widehat{V}_c(\mathbf{r}, \dot{\mathbf{r}}) = \frac{\|\boldsymbol{\Delta}\mathcal{L}\|^2}{\|\mathcal{L}_T\|^2} + \frac{1}{2}\left(\frac{\Delta\mathcal{E}}{\mathcal{E}_T}\right)^2 \tag{5.12b}$$

that measure the error between the target orbit and current state.

6 Trajectory Optimization Algorithm

The calculation of a trajectory between a given initial and final orbit can be posed as a very large optimal control problem. When solving an optimal control problem, the *direct transcription method* has been very effective (cf Ref [5]). In general, the method has three basic steps:

1. **Direct Transcription** Transcribe the optimal control problem into a nonlinear programming (NLP) problem by discretization;
2. **Sparse Nonlinear Program** Solve the sparse NLP
3. **Mesh Refinement** Assess the accuracy of the approximation (i.e. the finite dimensional problem), and if necessary refine the discretization, and then repeat the optimization steps.

As such the approach can be considered a *sequential nonlinear programming* or (SNLP) method. In the $\mathbb{SOS}$ (Sparse Optimization Suite) software implementation the sparse NLP can be solved using either a sequential quadratic programming (SQP) or primal-dual barrier (interior point) method.

The complete trajectory can be modeled as a sequence of burn and coast phases as illustrated in Figure 6.1. The vehicle can be controlled during a burn phase when the propulsion system operates, however, when the vehicle passes through the Earth's shadow the solar electric propulsion system does not provide thrust and the vehicle coasts. However in order to define the final trajectory

- the number and location of burn and coast phases must be determined, and
- an initial guess for the optimal trajectory must be provided.

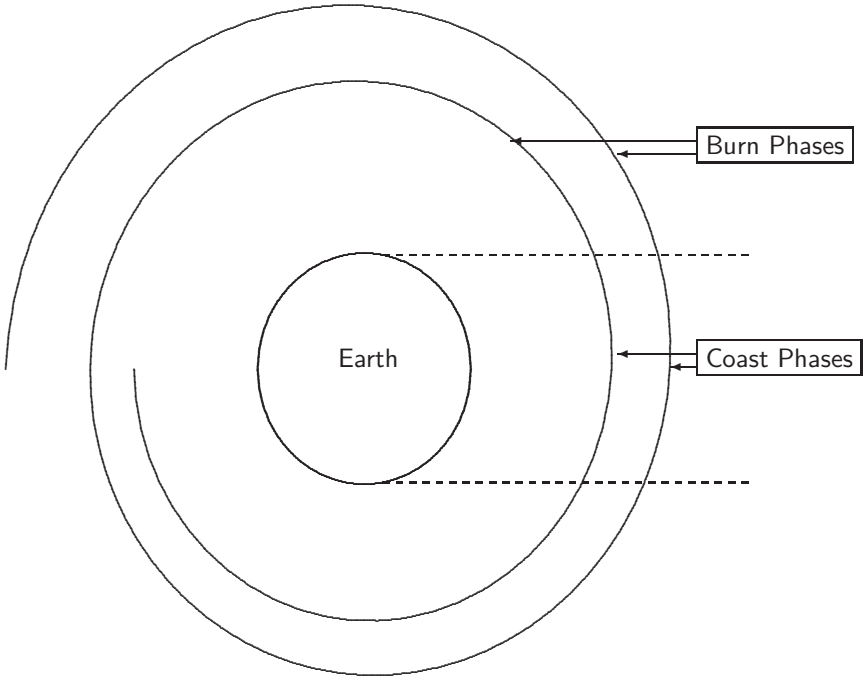

Figure 6.1: Trajectory Phases

These goals can be achieved by solving a sequence of problems, each designed to exploit results obtained in an earlier step. Broadly speaking there are two major operations. The first goal is to compute any trajectory between the initial and final orbit. This is accomplished using a receding horizon algorithm to get closer to the final orbit one orbit revolution at a time. The step by step process defines a dynamic history that serves as an initial guess for the second step, the overall trajectory optimization. When the overall trajectory is optimized, some of the candidate phases may be eliminated. Thus, the final phase structure is defined after a series of "passes" using candidate phase configurations. Once the final phase configuration is defined the entire trajectory optimization is completed. This final step incorporates high fidelity modeling as well as refinement of the discretization mesh.

6.1 Receding Horizon Algorithm

When solving an optimal control problem with an infinite time horizon, e.g. in chemical process control, a *receding horizon algorithm* is often used. When the process asymptotically approaches an equilibrium condition the time horizon is subdivided

$$t^{(0)} < t^{(1)} < t^{(2)} < \ldots < \infty$$

and then a sequence of optimal control problems are solved. For the orbit transfer application although the final time is not infinite, but very large, the idea is to subdivide the horizon into shorter segments

$$L_I^{(1)} < L_F^{(1)} = L_I^{(2)} < L_F^{(2)} = L_I^{(3)} < \ldots L_I^{(J)} < L_F^{(J)} \tag{6.1}$$

where the total number of subdivisions J is not known a-priori. In this setting the basic idea is to solve a sequence of optimal control problems for $j = 1, 2, \ldots, J$ over the domains

$$L_I^{(j)} \leq L \leq L_F^{(j)}. \tag{6.2}$$

For each subproblem the goal is to minimize an objective function at the final point

$$F^* = \min_{u,p} F\left[\mathbf{y}(L_F^{(j)}), \mathbf{u}(L_F^{(j)}), L_F^{(j)}\right]. \tag{6.3}$$

where each problem begins with specified initial conditions

$$\mathbf{y}\left[L_I^{(j)}\right] = \mathbf{y}^*\left[L_F^{(j-1)}\right] \tag{6.4}$$

and the initial conditions $\mathbf{y}^*\left[L_F^{(j-1)}\right]$ are the optimal states from the end of the preceding problem. Within the context of an orbit transfer it makes sense to consider an alternating sequence of burn and coast steps which can be pieced together as *phases* to form the overall trajectory as illustrated in Figure 6.2. With this in mind, let us consider the details of the individual receding horizon subproblems.

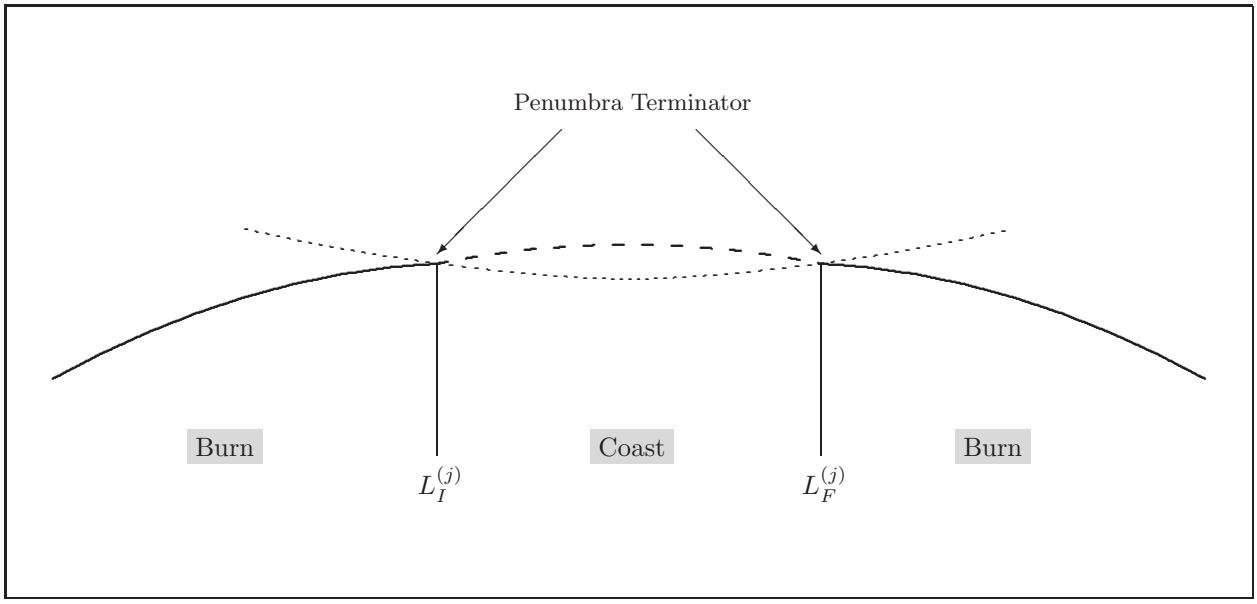

Figure 6.2: Receding Horizon Phases

6.1.1 Burn Phases

During a burn phase the dynamics are described by differential equations (2.26a)

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{p}, L) \tag{6.5}$$

with state $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$, control $\mathbf{u}^\mathsf{T} = [d_r, d_\theta, d_h]$, and parameters $\mathbf{p}^\mathsf{T} = [m_I, t_I] = [p_1, p_2]$. Furthermore we must impose the following path constraints:

$$1 = \mathbf{u}^\mathsf{T}\mathbf{u} \tag{6.6}$$

$$0 \le \|\boldsymbol{\delta}\| - \kappa \tag{6.7}$$

$$r_{min} \le r \tag{6.8}$$

Constraint (6.6) insures the control is a unit vector, and Rule (2) which insures the burn occurs in sunlight is enforced by (6.7). Finally (6.8) is imposed to insure that the radius r given by (2.7) exceeds a specified minimum value r_{min}.

In general, the initial conditions for receding horizon problem j are defined as the terminal conditions for the preceding problem. However when the preceding problem is a coast the mass m does not change. In this case, the initial condition for mass is defined by the preceding burn phase. Thus at the fixed initial point $L = L_I^{(j)}$ we impose the conditions:

$$\mathbf{y} = \mathbf{y}\mid_{L=L_F^{(j-1)}} \tag{6.9}$$

$$p_1 = \begin{cases} m_F^{(j-1)} & \text{if } j-1 \text{ is a burn phase} \\ m_F^{(j-2)} & \text{if } j-1 \text{ is a coast phase} \end{cases} \tag{6.10}$$

$$p_2 = t_F^{(j-1)} \tag{6.11}$$

There are two possible reasons to terminate a burn phase. One possibility is to enter a shadow, which is defined by the penumbra terminator condition (cf Rule (5)). The second possibility is to complete a full

orbital revolution without passing through the shadow. Furthermore to preclude a trajectory that is too close to the Earth, it is necessary to limit the perigee radius. Thus at the free final point $L = L_F^{(j)}$ we impose the following conditions:

$$0 = (\|\boldsymbol{\delta}\| - \kappa)(L_I^{(j)} + 2\pi - L) \tag{6.12}$$

$$0 \le r_p - r_{min} \tag{6.13}$$

where the perigee radius $r_p = p/(1 + \sqrt{f^2 + g^2})$. Observe that boundary condition (6.12) can be satisfied either by entering the penumbra $\|\boldsymbol{\delta}\| = \kappa$, *or* by completing an entire revolution $L = L_I^{(j)} + 2\pi$.

The goal of the optimization is to drive the trajectory closer to the mission orbit. Thus from (5.12a) or (5.12b) we choose $\mathbf{u}$ to minimize the objective

$$F = \begin{cases} \widehat{V}(\mathbf{r}, \dot{\mathbf{r}}) = \frac{\|\boldsymbol{\Delta}\mathcal{L}\|^2}{\|\mathcal{L}_T\|^2} + \frac{1}{2}\frac{\|\boldsymbol{\Delta}\mathcal{A}\|^2}{\|\mathcal{A}_T\|^2} & \text{elliptic} \\ \widehat{V}_c(\mathbf{r}, \dot{\mathbf{r}}) = \frac{\|\boldsymbol{\Delta}\mathcal{L}\|^2}{\|\mathcal{L}_T\|^2} + \frac{1}{2}\left(\frac{\Delta\mathcal{E}}{\mathcal{E}_T}\right)^2 & \text{circular} \end{cases} \tag{6.14}$$

at the end of the phase $L = L_F^{(j)}$.

Finally, since this optimal control problem must be solved repeatedly there are a number of important computational considerations. First an initial guess for the control can be computed by numerically integrating the Lyapunov steering law (5.11). Second, since the receding horizon solution will serve as an initial guess for the overall problem it suffices to utilize a simpler physical model. In particular we can ignore secondary body accelerations $\boldsymbol{\Delta}_q = 0$. Also it is reasonable to ignore the higher order terms in (3.3)-(3.4) and include only the J_2 terms when computing the oblate earth perturbation $\boldsymbol{\Delta}_g$.

6.1.2 Coast Phases

During a coast phase the dynamics are described by differential equations (3.25)

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, L) \tag{6.15}$$

with state $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$. To insure the coast is in the shadow region we can impose the path constraint

$$E(t) = \|\boldsymbol{\delta}\| - \kappa \le 0. \tag{6.16}$$

However, there are no controls during a coast arc and consequently the path constraint (6.16) serves to enhance the robustness of the iterations, and can only be satisfied by proper choice of the initial condition.

The initial conditions for receding horizon problem j are defined as the terminal conditions for the preceding problem, which must be a burn phase. Thus at the initial point $L = L_I^{(j)}$ we impose the conditions:

$$[p, f, g, h, k, t] = [p, f, g, h, k, t] \mid_{L = L_F^{(j-1)}} . \tag{6.17}$$

The coast phase terminates when exiting the shadow as defined by the penumbra terminator condition (cf Rule (5)). Thus at the boundary $L = L_F^{(j)}$ we impose the condition

$$\|\boldsymbol{\delta}\| = \kappa \tag{6.18}$$

During a coast phase no optimization is possible, and so the boundary condition uniquely determines the end of the phase $L = L_F^{(j)}$. The bounds (2.11)-(2.13) can be introduced to preclude excessive predictions during the iterative process. Finally, an initial guess for the dynamic variables is readily available using the analytic orbit prediction algorithm described in [5][p. 359] combined with a simple secant algorithm to locate the root of the termination condition (6.18)

6.1.3 Receding Horizon Algorithm Outline

1. **Initialization.**
   (a) Set horizon counter $j = 0$, last horizon counter $J = 0$, and define tolerances ϵ_V, ϵ_κ, and $\epsilon_{\dot{E}}$.
   (b) Set $L_F^{(0)} = L_0$ from Table A.1
   (c) Set $\mathbf{y}^*\left[L_F^{(0)}\right] = [p_0, f_0, g_0, h_0, k_0, t_0]$ and $\mathbf{p}^\mathsf{T} = [m_0, t_0]$ from Table A.1

(d) Define $\mathcal{L}_T$, $\mathcal{A}_T$, $\mathcal{E}_T$ from Table A.2

2. **Burn Phase.**

(a) Set $j = j + 1$

(b) Set $L_I^{(j)} = L_F^{(j-1)}$

(c) Set $\mathbf{y}\left[L_I^{(j)}\right]$ and $\mathbf{p}$ using (6.9)-(6.11).

(d) Compute initial guess using Lyapunov steering (5.11).

(e) Solve the burn phase optimal control problem defined in Section 6.1.1 and if $j = J$ omit the final boundary conditions (6.12)-(6.13).

i. Save the dynamic history for $\mathbf{y}_k$ and $\mathbf{u}_k$ and

ii. optimal objective function $F^{(j)}$ from Eq. (6.14).

3. **Termination Tests.**

(a) Last Horizon Tests

i. If $j = J$ the last horizon is complete; normal algorithm termination.

ii. If $F^{(j)} \leq \epsilon_V$ set $J = j$ to signify last horizon.

(b) Penumbra Encounter Tests

i. At $t_F^{(j)}$ compute the distance to the penumbra terminator $E = \|\boldsymbol{\delta}\| - \kappa$.

ii. If $E < \epsilon_\kappa$ the distance is negative, terminate algorithm with error condition.

iii. If $E > \epsilon_\kappa$ the distance is positive, bypass the coast phase, and return to step 2.

iv. If $|E| \leq \epsilon_\kappa$ and $\dot{E} > \epsilon_{\dot{E}}$ the distance is within tolerance but the distance rate is positive, bypass the coast phase, and return to step 2.

4. **Coast Phase.**

(a) Set $j = j + 1$

(b) Set $L_I^{(j)} = L_F^{(j-1)}$

(c) Set $\mathbf{y}\left[L_I^{(j)}\right] = \mathbf{y}^*\left[L_F^{(j-1)}\right]$

(d) Compute initial guess using analytic propagation. If coast is very short, go to step 4f.

(e) Solve the coast phase problem defined in Section 6.1.2.

(f) Save the dynamic history for $\mathbf{y}_k$.

6.2 Complete Trajectory Optimization Passes

Successful termination of the receding horizon algorithm in step 3, provides a reasonable initial guess for the final trajectory. In particular we have

- a guess for the total number of distinct phases, J,
- a tentative "burn-coast-burn-..." sequence, and
- a dynamic history for the state and control variables in all phases.

However, this information is deficient in a number of ways. The receding horizon approach is based on the *local* minimization of the orbital error as measured by the Lyapunov function (5.12a) or (5.12b). In general the value of the Lyapunov function is not zero at the final point, and as such the final point is "close" to the desired mission orbit. This can be corrected by explicitly introducing boundary conditions that define the mission orbit. Furthermore the step by step receding horizon approach does not address the potential benefits that accrue from a global view of the trajectory optimization problem. In particular the goal is to maximize the *payload*, i.e. the objective function

$$F = m_F. \tag{6.19}$$

Since the goal is to minimize the fuel consumed, i.e. maximize the mass at the final point $m_F = m\left[L_F^{(J)}\right]$, the tentative event sequence defined by the receding horizon algorithm may be suboptimal. In particular

we must expect that J is an upper bound on the optimal number of phases. As the overall trajectory optimization proceeds, we can expect that some of the coast phases will disappear, that is "shrink" to zero length. To accommodate this behavior, it is necessary to make a number of "passes" for the final trajectory optimization. The basic idea is to optimize the entire trajectory using a given number of phases. If the optimization drives the duration of one or more of the phases to a very small value, the pass is terminated. A new event sequence is then formulated, that eliminates the "near zero" phase. This procedure can be repeated until an optimal solution is found for a trajectory with strictly nonzero phases in the event sequence. The final pass is characterized by the correct event sequence, and can be completely optimized using all high order modeling options.

6.2.1 Total Trajectory Optimization

Optimization of the entire trajectory is accomplished using a sequence of phases. Specifically phase j is defined in the region

$$L_I^{(j)} \le L \le L_F^{(j)} \tag{6.20}$$

where $j = 1, \ldots, J$. For the very first phase we fix

$$L_0 = L_I^{(1)} \tag{6.21}$$

where L_0 is given in Table A.1. The quantities $L_I^{(j)}$ for $j = 2, \ldots, J$ and $L_F^{(j)}$ for $j = 1, \ldots, J$ are treated as optimization variables. In addition the parameter vectors $\mathbf{p}^{(j)}$ in the burn phases $j = 1, 3, 5, \ldots, J$ are optimization variables. The total trajectory is formulated as an alternating sequence of burn and coast phases. We arbitrarily, assume that phase 1 is a burn, and the final phase must necessarily be a burn.

Burn Phase The dynamics during burn phase j are described by the state $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$, control $\mathbf{u}^\mathsf{T} = [d_r, d_\theta, d_h]$, and parameters $\mathbf{p}^\mathsf{T} = [m_I, t_I]$ with differential-algebraic equations

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{p}, L) \qquad \text{for } L_I^{(j)} \le L \le L_F^{(j)} \tag{6.22a}$$

$$\mathbf{u}^\mathsf{T}\mathbf{u} = 1 \qquad \text{for } L_I^{(j)} \le L \le L_F^{(j)} \tag{6.22b}$$

and boundary conditions

$$L = L_0 \qquad \text{for } j = 1 \tag{6.22c}$$

$$\mathbf{y} = \mathbf{y}_0 \qquad \text{for } j = 1 \tag{6.22d}$$

$$[p_1, p_2] = [m_0, t_0] \qquad \text{for } j = 1 \tag{6.22e}$$

$$L = L_F^{(j-1)} \qquad \text{for } j = 2, \ldots, J \tag{6.22f}$$

$$\mathbf{y} = \mathbf{y}_F^{(j-1)} \qquad \text{for } j = 2, \ldots, J \tag{6.22g}$$

$$[p_1, p_2] = [m_F^{(j-2)}, t_F^{(j-1)}] \qquad \text{for } j = 3, 5, 7, \ldots, J \tag{6.22h}$$

$$\|\boldsymbol{\delta}\| = \kappa \qquad \text{for } j \ne J \tag{6.22i}$$

$$r_p \ge r_{min} \qquad \text{for } j \ne J \tag{6.22j}$$

$$[p, f, g, h, k] = [p_F, f_F, g_F, h_F, k_F] \qquad \text{for } j = J. \tag{6.22k}$$

The phase initial conditions (6.22c)-(6.22h) are evaluated at $L_I^{(j)}$. The phase terminal conditions (6.22i)-(6.22k) are evaluated at $L_F^{(j)}$. The simple bounds (2.11)-(2.13) are also included.

Coast Phase During coast phase j the dynamics are described by the state variables $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$ with differential equations

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, L) \qquad \text{for } L_I^{(j)} \le L \le L_F^{(j)} \tag{6.23a}$$

and boundary conditions

$$L = L_F^{(j-1)} \qquad \text{for } j = 2, \ldots, J-1 \tag{6.23b}$$

$$\mathbf{y} = \mathbf{y}_F^{(j-1)} \qquad \text{for } j = 2, \ldots, J-1 \tag{6.23c}$$

$$\|\boldsymbol{\delta}\| = \kappa \qquad \text{for } j \ne J \tag{6.23d}$$

$$\Delta L_{min} \le L_F^{(j)} - L_I^{(j)}. \tag{6.23e}$$

The initial conditions (6.23b)-(6.23c) are evaluated at $L_I^{(j)}$ and the terminal condition (6.23d) is enforced at $L_F^{(j)}$. Constraint (6.23e) limits the duration of the phase where the minimum coast angle $\Delta L_{min} = \pi/180$. The simple bounds (2.11)-(2.13) are also included.

Final Pass Optimization Using burn and coast phases as outlined in the preceding two paragraphs the objective function can be optimized. However, because the tentative burn/coast event sequence may not be correct at the end of the first (coarse) mesh problem, we examine the inequality constraint (6.23e). If this constraint is active, we assume the current phase configuration is wrong. In particular, to construct an optimal trajectory, the behavior suggests that any short coast can be eliminated from the phase sequence. Thus, we delete short coast phases, to form a new event sequence. When a coast phase is deleted, the burn phase that preceded it, and the burn phase that followed it are "merged" to form a single new burn. Thus when a single coast phase is deleted, the next optimization pass will have two fewer phases.

On the other hand if the inequality constraint (6.23e) is not active we presume the event sequence is correct and begin the final pass optimization. The final pass is characterized by a number of modeling changes to improve accuracy and computational efficiency. In particular, we include all perturbing accelerations. Oblate earth perturbations include all terms through J_4 in (3.3)-(3.4). Perturbations attributed to the Sun and Moon (3.12) are included. In addition, for the final pass burn phases we model the state $\mathbf{y}^\mathsf{T} = [p, f, g, h, k, t]$, however, as control we use $\mathbf{u}^\mathsf{T} = [\alpha, \beta] = [u_1, u_2]$ with the thrust acceleration given by (3.20). This is computationally beneficial because it eliminates one control variable as well as the algebraic path constraint (6.22b). Finally, the mesh refinement algorithm in $\mathbb{SOS}$ is utilized to achieve the required solution accuracy.

7 Computational Results

7.1 Geosynchronous Orbit

An optimal transfer to geosynchronous orbit illustrates the performance of the method. Table A.1 summarizes all of the parameters that define the mission, and Table A.2 gives the specific mission orbit parameters. The values in Table A.1 define initial conditions at the ascending node of a 500 km circular park orbit, with 28.5 degree inclination. The mission orbit defined in Table A.2 has zero inclination and a 24 hour period. The first step in the solution process is to execute a sequence of receding horizon steps as described in Section 6.1. Each of these optimal control problems can be solved efficiently even with a mesh refinement tolerance of $\epsilon \leq 1 \times 10^{-7}$ as shown in the first line of Table 7.1. For this mission, 438 steps were executed with the results summarized in Figure 7.1. As the orbit error $\widehat{V}_c$ (5.12b) is reduced, the length of the burn phases $L_F^{(j)} - L_I^{(j)}$ gradually increase, until a complete orbital revolution is executed without passing through the earth's shadow. Referring to (6.12), the early burn steps terminate because $(\|\boldsymbol{\delta}\| - \kappa) = 0$ whereas the final burn steps terminate because $(L_I^{(j)} + 2\pi - L) = 0$. The receding horizon procedure generates a trajectory from the park to mission orbit, step by step, where each step is approximately one orbital revolution. In so doing, the dynamic history constructed serves as an initial guess for the overall trajectory optimization. The step by step process also generates a tentative structure for the sequence of burn and coast phases. Of course the stepwise procedure may yield two or more successive burn phases, with no intervening coast phase. In this case, the successive burn phases can be coalesced into a single (longer) phase. For the geosynchronous example, the receding horizon solution had 438 phases which is displayed as pass 0 in Figure 7.2.

After compression, the trajectory can be modeled using 381 phases and the overall trajectory optimization process discussed in Section 6.2.1 can begin. The payload optimization for pass 1 shown in Figure 7.2 does two things. First, the payload delivered is increased. Second, at the optimal solution one of the coast phases–in this case phase 380–is very short, and constraint (6.23e) is active. Thus a new phase structure is constructed, that eliminates a coast, and coalesces two burns into a single phase. Consequently pass 2, has 379 phases, and the payload optimization continues with the new phase structure. The procedure is repeated, in this case ten more times, until reaching an optimal trajectory with the correct phase structure. During this process it is not necessary to use mesh refinement, and the more robust control vector formulation is used. The second line of Table 7.1 summarizes the computational behavior for the multi-pass process.

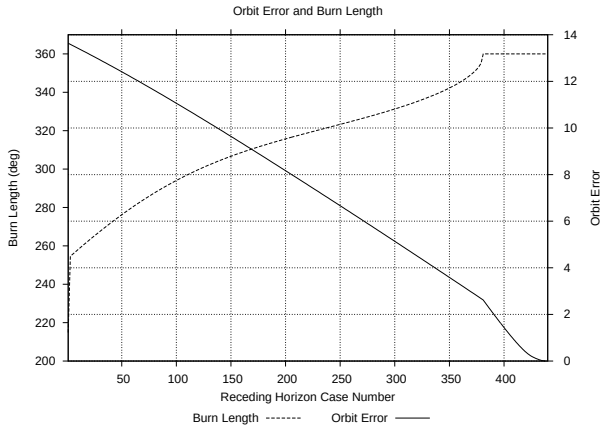

Figure 7.1: Receding Horizon Summary

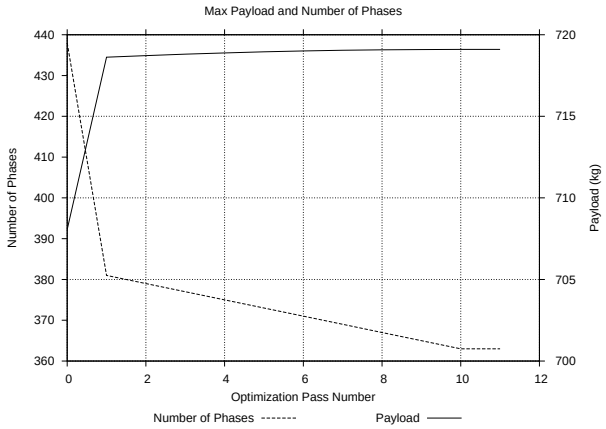

Figure 7.2: Multi-pass Summary

Table 7.1: Low-Thrust Algorithm Summary (Geosynchronous)

Procedure	No. of Steps	CPU Time (Min, Max, Avg)		
Receding Horizon	438	9.91×10^{-4}	2.11	.3524
Multi-Pass	10	107.46	600.83	179.58

The final pass yields the optimal solution for this problem which is illustrated in Figure 7.7. When the oscillatory behavior for many of the dynamic variables is displayed on the time scale of the entire mission the plots obscure the local behavior. To illustrate the behavior over a shorter period Figure 7.3 plots the yaw angle over the final phase and revolutions. Similar information for the pitch angle is shown in Figure7.4. It is interesting to note that the pitch angle exhibits very rapid changes near the end of the trajectory. On the other hand when we compute the *total rotation rate* (3.24) the maximum value is $\dot{\theta} \approx 0.05340318$ deg/sec which occurs at $L = 248.08939896$ revolutions. The optimal value for the objective function is $m_F^* = 718.788672$ kg which occurs at the final point defined by $L_F = 248.49972$ revolutions. The transfer duration was 43.129147 days and was modeled using 363 phases.

The computational performance of the optimization algorithm for the final pass is summarized in Table (7.2). Seven mesh refinement iterations (column 1) were needed to reduce the relative error $\epsilon \leq 1 \times 10^{-7}$. To achieve this accuracy the number of grid points M were increased from 4496 to 25409. At the final solution the number of nonlinear programming (NLP) constraints was $m = 175599$, the number of NLP variables was $n = 262240$, and the number of degrees of freedom was $n_d = 87173$. All calculations were performed on a desktop computer using an Intel I7 processor (3.06 Ghz), with Linux operating system, GNU Fortran compiler, and the total CPU time is summarized in the last column of the table.

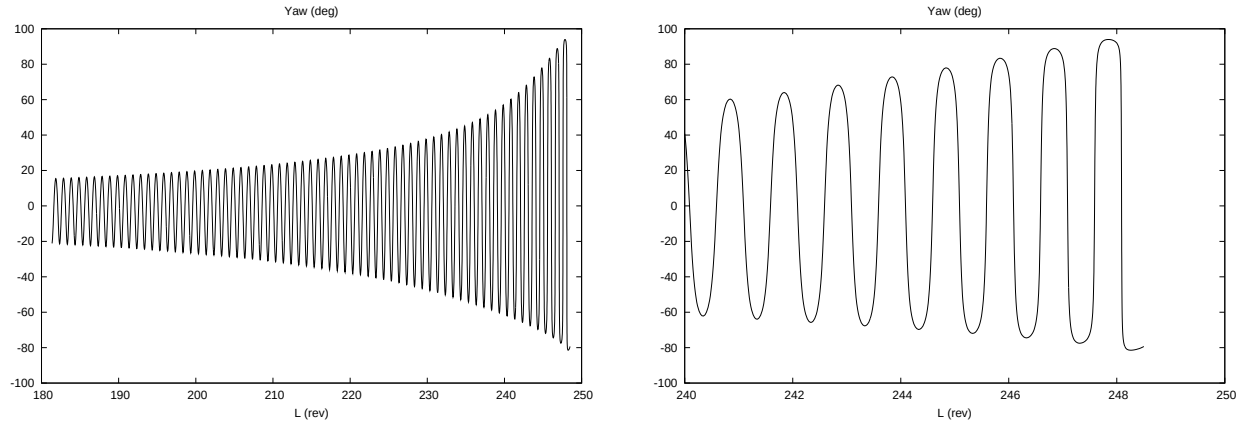

Figure 7.3: Optimal Geosynchronous Yaw History (Final Phase and Revolutions)

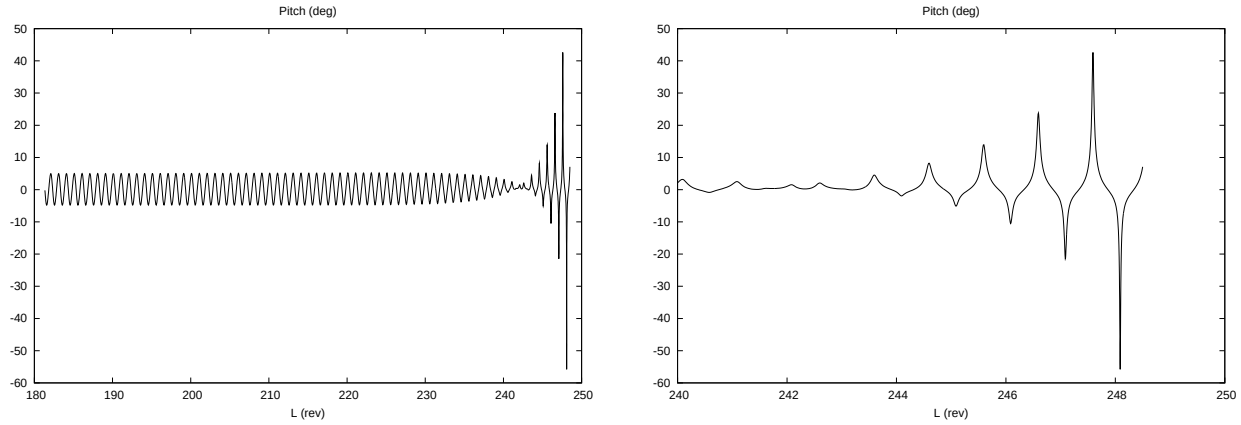

Figure 7.4: Optimal Geosynchronous Pitch History (Final Phase and Revolutions)

Table 7.2: Mesh Refinement Summary–Final Pass, Geosynchronous Orbit

k	M	m	n	n_d	ϵ	Time (sec)
1	4496	32153	46922	15301	2.5×10^{-2}	215.9
2	7708	54261	80374	26645	3.1×10^{-4}	67.2
3	14595	101830	152931	51633	2.0×10^{-5}	81.7
4	23318	161989	244374	82917	1.3×10^{-6}	177.1
5	25384	175424	261965	87073	1.0×10^{-7}	158.9
6	25399	175529	262130	87133	1.0×10^{-7}	188.7
7	25409	175599	262240	87173	9.9×10^{-8}	158.2
Total	25409					1048.

Local Solutions There are often many local optimal solutions for low thrust transfers. This can be attributed both to the periodic nature of the trajectories, as well as the complicated interaction of nonlinear constraints. Table 7.3 summarizes a number of local solutions that differ from the geosynchronous case in the first line. For the last three cases in Table 7.3 the launch date is changed from midnight on Dec. 21, 2015 to midnight on June 21, 2016, which alters the geometry of the earth shadow during the transfer. Second, when a solution is computed using an iterative method, the sequence of iterations defines which local solution is found and it can be altered by changing the initial guess and/or the problem scaling.

Table 7.3: Local Solutions Geosynchronous Orbit

Number of Phases	Payload (kg)
363†	718.78867 (kg)
373	719.23570 (kg)
377	722.25332 (kg)
387	722.12664 (kg)
†Reference Epoch: Dec. 21, 2015	

7.2 Molniya Orbit

An optimal transfer to a Molniya orbit serves as a second demonstration of the method. Table A.1 summarizes all of the parameters that define the mission, and Table A.2 gives the specific mission orbit parameters. The Molniya mission has an inclination of 63.4°, an argument of perigee of 270°, a right ascension of 0°, a perigee altitude of 350 nautical miles (nm), and an apogee altitude of 21450 nm, which corresponds to an orbital period of approximately 12 hours. The computational performance is summarized in Tables 7.4 and 7.5. For this example 743 receding horizon steps were needed to compute a transfer trajectory between the park orbit and Molniya mission orbit using the orbit error (5.12a). Comparing the two different missions, geosynchronous in Figure 7.1, and Molniya in Figure (7.5) an interesting trend can be observed. For both

missions the orbit error decreases monotonically with the number of receding horizon steps. The burn length increases monotonically for the geosynchronous mission, however, for the Molniya mission the burn length varies in length. In fact the receding horizon process leads to a sequence of 743 alternating burn and coast phases. However when the final payload for the entire trajectory is optimized as shown in Figure (7.6) many of the coast phases are eliminated and the final trajectory sequence has 693 phases–347 burn and 346 coast. The optimal trajectory is illustrated in Figure (7.8) and the optimal payload is $m_F^* = 652.409985$ kg which occurs after 372.19939 revolutions and 59.650620 days. It is interesting to note that some portions of the optimal transfer trajectory pass close to the Earth. In particular some of the boundary conditions (6.22j) are in the final active set, that is,

$$r_p^* = r_{min} \qquad \text{for } j = 105, 107, \ldots, 163 \tag{7.1}$$

where $r_{min} = 6655.94$ km, which corresponds to an altitude of 277.8 km, well below the park orbit altitude of 500 km. The control vector formulation was used for all optimization passes, because nearly retrograde steering is needed during the final phases. Five mesh refinement iterations were needed to reduce the relative error $\epsilon \leq 1 \times 10^{-6}$.

Table 7.4: Low-Thrust Algorithm Summary (Molniya)

Procedure	No. of Steps	CPU Time (Min, Max, Avg)		
Receding Horizon	743	1.98×10^{-3}	22.60	.5739
Multi-Pass	4	39.421	5817.17	1601.65

Table 7.5: Mesh Refinement Summary–Final Pass, Molniya Orbit

k	M	m	n	n_d	ϵ	Time (sec)
1	6457	51767	70566	19796	4.8×10^{-3}	38.84
2	12062	94807	132426	38616	3.8×10^{-4}	80.91
3	17430	134649	187536	53884	2.5×10^{-5}	191.42
4	20706	158153	218736	61580	1.7×10^{-6}	58.43
5	20803	158735	219318	61580	9.9×10^{-7}	50.94
Total	20803					420.54

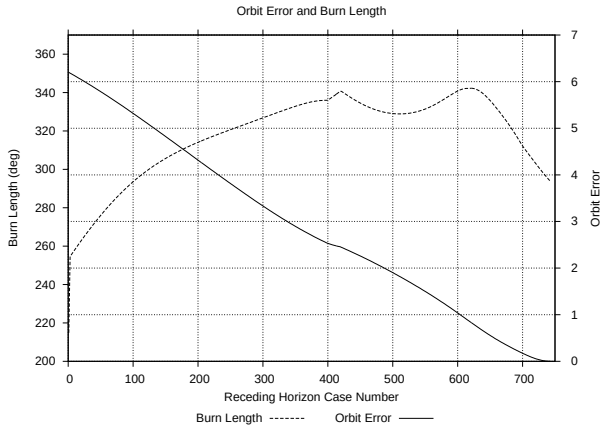

Figure 7.5: Receding Horizon Summary

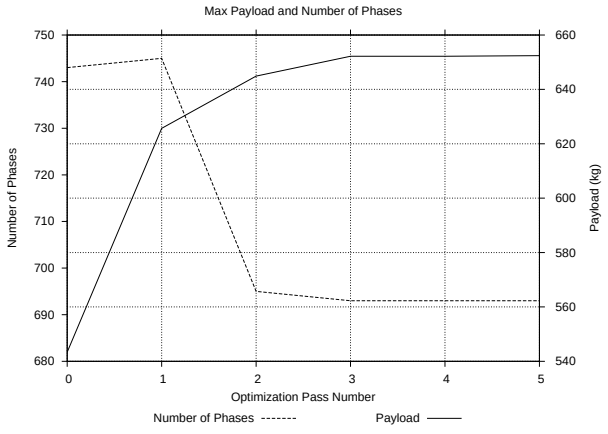

Figure 7.6: Multi-pass Summary

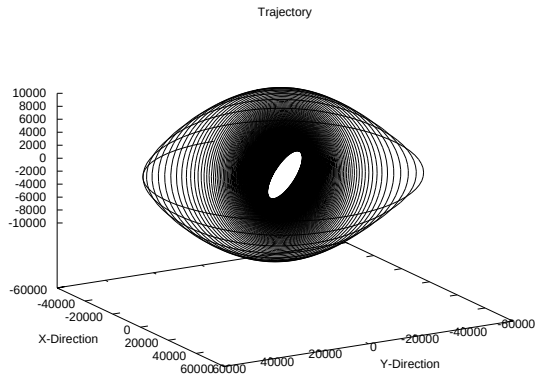

Figure 7.7: Optimal Geosynchronous Transfer

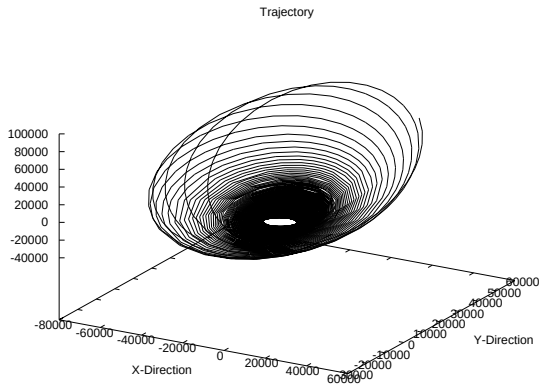

Figure 7.8: Optimal Molniya Transfer

8 Summary and Conclusions

An approach has been developed for constructing optimal low thrust orbit transfers that addresses the impact of eclipse regions. The complete trajectory is modeled using a sequence of burn and coast phases. An initial guess is constructed by stepping one phase at a time using a receding horizon technique to minimize the orbit error at the end of each phase. The initial guess and phase sequence is then modified during a number of passes that optimize the final mass. The approach is illustrated by computing optimal transfers to geosynchronous and Molniya mission orbits. The receding horizon philosophy is used here to construct an initial guess, however, it may also be applicable to on-board real time application. In particular, the technique could be used to correct for unexpected errors and/or mission requirements in a real-time re-targeting scenario.

Acknowledgment The author thanks David C. Eagle for graciously supplying software implementing the ephemerides described in Reference [12].

A Appendix

Table A.1: Mission Parameters

p_0	6878.14 (km)	μ	398600.43638081953 (km^3/sec^2)
f_0	0.	μ_s	132712440018. (km^3/sec^2)
g_0	0.	μ_m	4902.7988158612316 (km^3/sec^2)
h_0	-0.25396764647494369	a_e	6378.14 (km)
k_0	0.	a_s	695500. (km)
L_0	.5 (rev)	T	1.445 (N)
m_0	1000.0 (kg)	$\dot{m}$	$-7.96761955027139941\times10^{-5}$ (kg/sec)
J_2	1082.639×10^{-6}	a_T	1.445×10^{-6} (km/sec^2)
J_3	-2.565×10^{-6}	c_{EX}	18135.906099467051 (m/sec)
J_4	-1.608×10^{-6}	I_s	1849.3477486671852 (sec)

Reference Epoch: $t_0 = 0$	Julian Date: 2457377.5	Dec. 21, 2015; Midnight

Table A.2: Mission Orbit Parameters

	Geosynchronous	Molniya
$\mathcal{L}_{Tx}$	0.	0.
$\mathcal{L}_{Ty}$	0.	-62338.826307078591
$\mathcal{L}_{Tz}$	129758.69563408432	31216.958835717571
$\mathcal{A}_{Tx}$	0.	0.
$\mathcal{A}_{Ty}$	0.	-131270.39184634088
$\mathcal{A}_{Tz}$	0.	-262140.91512361390
$\mathcal{E}_T$	-4.7181593164749254	-7.5023778781510373
p_F	42241.095482827557	12194.235983352495
f_F	0.	0.
g_F	0.	-0.73550326106514829
h_F	0.	0.61761258786098949
k_F	0.	0.

$\mathcal{L}_T, \mathcal{E}_T$: km^2/sec^2 $\quad$ $\mathcal{A}_T$: km^3/sec^2 $\quad$ p: km

References

[1] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 1633 Broadway, New York, NY 10019, 1987.

[2] John T. Betts. Optimal Interplanetary Orbit Transfers by Direct Transcription. *The Journal of the Astronautical Sciences*, 42(3):247–268, July–September 1994.

[3] John T. Betts. Very Low Thrust Trajectory Optimization. In Hans-Joachim Bungartz, Franz Durst, and Christoph Zenger, editors, *High Performance Scientific and Engineering Computing, Proceedings of the International FORTWIHR Conference on HPSEC, Munich, March 16-18, 1998*, Berlin, Heidelberg, 1999. Springer-Verlag.

[4] John T. Betts. Very Low Thrust Trajectory Optimization Using a Direct SQP Method. *Journal of Computational and Applied Mathematics*, 120(1–2):27–40, August 2000.

[5] John T. Betts. *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. Second Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA., 2010.

[6] John T. Betts and Sven O. Erb. Optimal Low Thrust Trajectories to the Moon. *SIAM Journal on Applied Dynamical Systems*, 2(2):144–170, May 2003. `http://www.siam.org/journals/siads/2-2/40908.html`.

[7] Roger A. Broucke and Paul J. Cefola. On Equinoctial Orbit Elements. *Celestial Mechanics*, 5:303–310, 1972.

[8] Dong Eui Chang, David F. Chichka, and Jerrold E. Marsden. Lyapunov-Based Transfer Between Elliptic Keplerian Orbits. *Discrete and Continuous Dynamical Systems Series B*, 2(1):57–67, February 2002.

[9] T. N. Edelbaum. Optimum Power-Limited Orbit Transfer in Strong Gravity Fields. *AIAA Journal*, 3(5):921–925, May 1965.

[10] Pedro Ramon Escobal. *Methods of Orbit Determination*. Second Edition. Robert E. Krieger Publishing Company, Malabar, Florida, 1985.

[11] Ch. Ferrier and Richard Epenoy. Optimal Control for Engines with Electro-Ionic Propulsion Under Constraint of Eclipse. *Acta Astronautica*, 48(4):181–192, February 2001.

[12] T. C. Van Flandern and K. F. Pulkkinen. Low-Precision Formulae for Planetary Positions. *The Astrophysical Journal Supplement Series 41*, pages 391–411, November 1979.

[13] M. R. Ilgen. Low Thrust OTV Guidance Using Lyapunov Optimal Feedback Control Techniques. *Advances in the Astronautical Sciences*, 85:1527–1545, 1980.

[14] M. R. Ilgen. Low Thrust OTV Guidance Using Lyapunov Optimal Feedback Control Techniques. In *AAS/AIAA Astrodynamics Specialist Conference*, AAS 93-680, Victoria, B.C., Canada, August 1993.

[15] Jean A. Kechichian. Equinoctial Orbit elements: Application to Optimal Transfer Problems. In *AIAA/AAS Astrodynamics Specialist Conference*, AIAA 90-2976, Portland, OR, August 1990.

[16] Jean A. Kechichian. Trajectory Optimization with a Modified Set of Equinoctial Orbit Elements. In *AIAA/AAS Astrodynamics Specialist Conference*, AAS 91-524, Durango, CO, August 1991.

[17] Jean A. Kechichian. Inclusion of Higher Order Harmonics in the Modeling of Optimal Low-Thrust Orbit Transfer. *The Journal of the Astronautical Sciences*, 56(1):41–70, January–March 2008.

[18] Carlos R. Ortiz Longo and Steven L. Rickman. Method for the Calculation of Spacecraft Umbra and Penumbra Shadow Terminator Points. Technical Report NASA Technical Paper 3547, Lyndon B. Johnson Space Center, Houston, Texas, April 1995.

[19] Anastassios E. Petropoulos. Refinements to the Q-Law for Low-Thrust Orbit Transfers. In *AAS/AIAA Astrodynamics Space Flight Mechanics Conference*, AAS 05-162, Copper Mountain, Colorado, January 2005.

[20] M. J. H. Walker, B. Ireland, and J. Owens. A Set of Modified Equinoctial Orbit Elements. *Celestial Mechanics*, 36:409–419, 1985.

[21] T. Yee and Jean A. Kechichian. On the Dyanmic Modeling in Optimal Low-thrust Orbit Transfer. In *AAS/AIAA Spaceflight Mechanics Meeting*, AAS 92-177, Colorado Springs, CO, February 1992.