

LOW-THRUST CONTROL OF A LUNAR MAPPING ORBIT

Nathan Harl¹ and Henry J. Pernicka²
University of Missouri-Rolla

Abstract

A method is presented for generating and maintaining a lunar mapping orbit using continuous low-thrust hardware. Optimal control theory is used to maintain a lunar orbit that is low-altitude, near-polar, and Sun-synchronous; three typical requirements for a successful lunar mapping mission. The analysis of the optimal control problem leads to the commonly seen two-point boundary value problem, which is solved using a simple indirect shooting algorithm. Simulations are presented for a 50-day mapping duration, in which it is shown that a very tight control is achieved with thrust levels below 1 N for a 1000 kg spacecraft. A straightforward approach for using the method presented to compute missions of any duration is also discussed.

Introduction

In 2004, President Bush announced his Vision for Space Exploration (VSE) plan, which defined many goals for NASA to fulfill in the next 20 to 30 years. One of the first and primary goals of the VSE is for NASA to commence unmanned and manned missions to the Moon by 2008 and 2020 respectively, with the hope of eventually establishing a manned station or colony on the Moon. The main purpose of the unmanned missions will be to facilitate the manned missions by mapping the Moon in order to identify locations of water in the form of ice as possible landing sites. The first unmanned mission under the VSE, the Lunar Reconnaissance Orbiter (LRO), is currently scheduled to launch in Fall 2008 [1]. LRO will consist of a one-year duration reconnaissance mission using a 50 km polar lunar orbit.

Since the mapping missions will involve using a satellite to continuously image the lunar surface, certain orbital properties for such a satellite would enhance its ability to collect the needed data. First, it is desired for the satellite to be in a low-altitude orbit in order to provide high resolution. A near-polar orbit is also desired to ensure global coverage of the Moon. Finally, it is desired for the satellite to be located in a Sun-synchronous orbit, in which the line of nodes rotates at such a rate that the Sun will always be in the same orientation with respect to the satellite orbit plane so that imaging will be taken with consistent lighting conditions. Sun-synchronous orbits are also useful in that the thermal environment is identical on a daily basis, which can be beneficial for thermal management of sensitive components needed to perform the mapping mission.

Of the three properties described, the most challenging to achieve is the Sun-synchronous condition. This is primarily because lunar Sun-synchronous orbits do not exist naturally, unlike Sun-synchronous missions about Earth in which the J_2 effect facilitates the generation of Sun-synchronous orbits that do not require control maneuvers to rotate the node line. The Earth's equatorial oblateness creates an out-of-plane force which, for orbits with certain altitudes and inclinations, rotates an orbit at the Sun-synchronous rate. However, since the Moon's oblateness is much less than the Earth's, this luxury is not present when attempting to establish a lunar Sun-synchronous orbit. Instead, generating and maintaining a lunar Sun-synchronous orbit requires continuously applying an out-of-plane thrust to rotate the orbit at the desired rate.

¹ Graduate Student, Department of Mechanical and Aerospace Engineering, 1870 Miner Circle, Rolla, MO 65409-0050. E-mail: nrh7b2@umr.edu.

² Associate Professor, Department of Mechanical and Aerospace Engineering, 1870 Miner Circle, Rolla, MO, 65409-0050. E-mail: pernicka@umr.edu.

This paper presents a method for establishing a low-altitude, polar, Sun-synchronous lunar orbit using continuous low-thrust. It is shown that by performing a continuous out-of-plane acceleration, the Sun-synchronous condition can be maintained for a long-term mission. Optimal control theory with a Lagrangian performance index is used to determine the required control accelerations. However, as discussed later in the paper, due to the physical inability to control the line of nodes during equatorial crossings a switch is made between two controllers in these regions. This leads to suboptimal (but still effective) trajectories.

The dynamic model used in this study includes the three primary perturbations that a low-altitude lunar orbiter would experience: the perturbations due to the Earth and Sun's gravity and due to the nonspherical lunar gravity field. The lunar gravity field is modeled using the LP165P gravity model [2] obtained from the NASA PDS Geosciences Node.

Previous Studies

Lunar Mapping Missions

Some of the first lunar mapping missions date back to the 1960s. As the United States and Russia were competing to be the first to perform a manned mission to land on the Moon, both countries began conducting unmanned scientific missions to the Moon in order to study the lunar surface and test soft landing approaches.

NASA began its first set of unmanned lunar missions in the early 1960s with the Ranger program [3]. The goal of each of the Ranger missions was to take high resolution images of the Moon and return them in real time. Each mission was designed such that the satellite was placed on a collision course with the lunar surface, with pictures of the surface being taken up to the impact time. However, these missions encountered a number of malfunctions, and only three of the nine missions were successful. Later in the decade, NASA initiated the Lunar Surveyor program [4]. While the primary goal of this program was to achieve soft landing on the Moon, the lunar surveyor satellites also took a large assortment of close range images of the lunar surface.

Russia also sent an assortment of scientific missions to the Moon in the form of the Luna program [5]. Many of these missions involved taking high resolution images of the lunar surface, and the Luna 3 mission is famous for obtaining the first images of the far side of the Moon. A total of 24 missions during the 1960s and 1970s were conducted in the Luna program with fairly substantial scientific gain.

More recently, Park and Junkins [6] propose possible lunar mapping orbits using a frozen orbit concept. In such a frozen orbit, the eccentricity, argument of periapsis, and inclination variations are approximately zero. The authors also showed that at an inclination of 101.4° , a near Sun-synchronous orbit exists in which the line of nodes deviates from the Sun-synchronous condition by approximately 10° over the course of one month. However, this approach would not necessarily be suitable for a long duration mission as the deviation grows in a secular manner.

Raman and Adimurthy [7] also studied lunar mapping orbits; however they chose to abandon the search for a Sun-synchronous orbit and instead focused on searching for a near-polar, circular, low-altitude lunar orbit with a long lifetime. For a circular 100 km altitude orbit, they found that the orbit lifetime depends directly on the inclination, minimizing at 20 days for an initial inclination of 11° and maximizing at over 730 days for an initial inclination of 95° .

With a renewed interest in the Moon, NASA has conducted recent lunar mapping missions that have greatly increased knowledge of the Moon. Two of these missions are the 1994 Clementine mission [8, 9] and the 1998 Lunar Prospector mission [10]. A large amount of the knowledge currently available about the Moon, including the lunar gravity model used in this work, comes directly from the results of these two missions.

Low-Thrust Stationkeeping

Some of the first works on low-thrust stationkeeping were conducted by Edelbaum and Hunziker. In 1964, Edelbaum [11] investigated optimal orbit-raising, rendezvous, and stationkeeping using low-thrust systems. By finding a closed-form analytic solution relating changes in orbital elements to Lagrangian multipliers which defined the control variables, Edelbaum's method allowed for the optimal correction of all six orbital elements of a general elliptical orbit. Later, in 1970, Hunziker [12] developed a method for

stationkeeping a nearly circular equatorial orbit for a 24-hour Earth satellite. Hunziker’s method involved combining variation of parameters with a Newton-Rhapson method to create a bang-bang controller.

Much recent work related to low-thrust stationkeeping [13-15] considers the stationkeeping of satellites in geostationary orbit about Earth. In each of these works, it is desired to stationkeep the orbit by minimizing the variations in latitude and longitude, therefore maintaining the satellite over a certain position on Earth.

While low-thrust stationkeeping strategies have been studied for Earth orbits, no previous works that were identified address the application of low-thrust stationkeeping for orbits around other bodies, such as the Moon. The work presented in this paper considers mapping orbits about the Moon; however, the methods developed are general and can be applied for missions at other locations.

Properties of Sun-Synchronous Orbits

In this section, some of the general properties of Sun-synchronous orbits are reviewed to better facilitate the understanding of subsequent sections of the paper. In general, a Sun-synchronous orbit is defined as a near-polar orbit in which the line of nodes rotates at such a rate that it is always oriented at a constant angle with a vector directed from the Sun to the body the spacecraft is orbiting. The rate at which the node line must rotate in order for the orbit to be considered Sun-synchronous depends on which body the spacecraft is orbiting. The exact rate is a function of the period of the particular body’s orbit about the Sun, since in one period the node line must rotate by 360° . For example, since the Earth orbits the Sun in one sidereal year (365.2563 days), a Sun-synchronous orbit about Earth would require a nodal rotation rate of $360^\circ/365.2563$ days, or $0.9856^\circ/\text{day}$.

Since the Moon follows the Earth in its motion about the Sun, the Moon also orbits the Sun in one sidereal year. Therefore, the rate at which the line of nodes must rotate for a lunar Sun-synchronous orbit is $0.9856^\circ/\text{day}$. As mentioned previously, while an Earth orbit can acquire this nodal rotation rate naturally due to the J_2 effect, the same does not apply for the Moon since its J_2 value is much smaller than Earth’s. Because of this, a true lunar Sun-synchronous orbit can only be achieved through the application of control maneuvers.

Equations of Motion

In this section, the equations of motion used to model the motion of a low-altitude lunar orbiter are introduced. The equations incorporate perturbations from the Earth, Sun, and nonspherical gravity field of the Moon, which are the three primary perturbations that such a spacecraft would experience. Simulations are presented which show the long-term effects of these perturbations on the classical orbital elements.

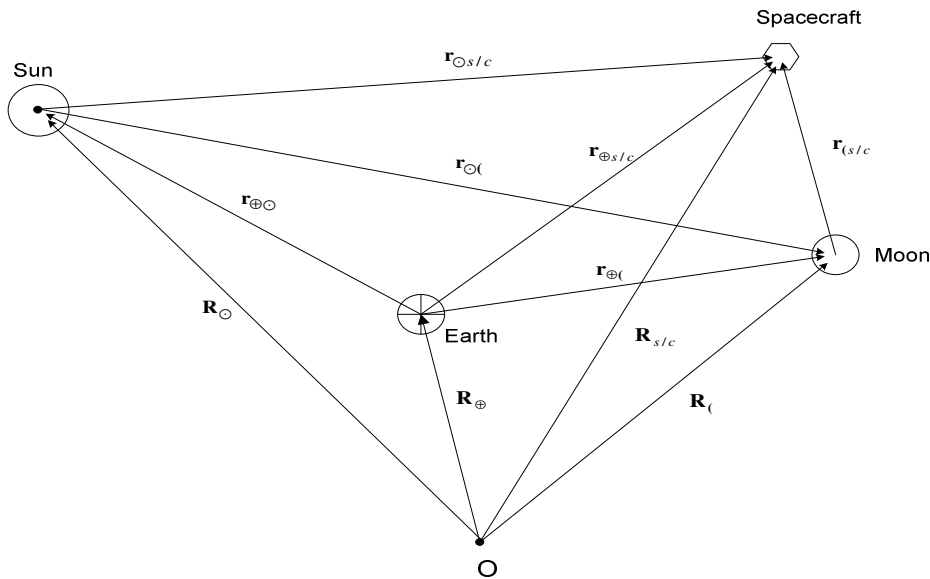


FIG. 1. Four-Body Sun-Earth-Moon-Spacecraft System.

Figure 1 shows the orientation of the four-body system that is analyzed in this work. Point “O” in Fig. 1 is an inertially fixed point in space from which the position vectors of the four bodies are defined. In order to compute the vectors $\mathbf{r}_{\odot\oplus}$ and $\mathbf{r}_{\oplus\odot}$, ephemeris-based equations are used. The $\mathbf{r}_{\odot\oplus}$ vector is found using ephemeris equations from the JPL software QUICK [16, 17], and the $\mathbf{r}_{\oplus\odot}$ vector is found from ephemeris equations given by Escobal [18]. Once these vectors are found, and noting that

$$\mathbf{r}_{(s/c)} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}} \quad (1)$$

the other position vectors can be found (using Fig. 1) as

$$\mathbf{r}_{\oplus sc} = \mathbf{r}_{\oplus\odot} + \mathbf{r}_{(s/c)} \quad (2)$$

$$\mathbf{r}_{\odot sc} = -\mathbf{r}_{\oplus\odot} + \mathbf{r}_{\oplus} + \mathbf{r}_{(s/c)} \quad (3)$$

$$\mathbf{r}_{\odot\oplus} = -\mathbf{r}_{\oplus\odot} + \mathbf{r}_{\oplus\odot} \quad (4)$$

Since it was desired to analyze the system dynamics in a reference frame fixed in the Moon, a mean selenocentric reference frame [18] was chosen. A selenocentric frame is defined as a rotating reference frame with the origin at the center of the Moon, where the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ axes correspond to the Moon’s principal axes of inertia. In particular, the $\hat{\mathbf{x}}$ axis lies in the Moon’s equatorial plane directed toward Earth, the $\hat{\mathbf{z}}$ axis is directed through the Moon’s north pole, and the $\hat{\mathbf{y}}$ axis completes the right-handed coordinate system. The mean selenocentric reference frame differs from the “true” selenocentric reference frame in that it does not incorporate the physical librations of the Moon. The effect of the librations will likely only cause small perturbations to the model and can be included in a future study.

In order to obtain equations of motion that can be readily propagated using a numerical integrator, a state-space formulation was used to convert the system into a set of six first-order differential equations in terms of a state vector $\mathbf{x}^T = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T \triangleq [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$ where x , y , z , \dot{x} , \dot{y} , and \dot{z} correspond to the Cartesian position and velocity components of the spacecraft in the selenocentric frame. The six first-order differential equations of motion for this system are then

$$\dot{x}_1 = x_4 \quad (5a)$$

$$\dot{x}_2 = x_5 \quad (5b)$$

$$\dot{x}_3 = x_6 \quad (5c)$$

$$\dot{x}_4 = 2\omega_m x_5 + \omega_m^2 x_1 - \frac{\mu_{\oplus} r_{\oplus sc x}}{|\mathbf{r}_{\oplus sc}|^3} - \frac{\mu_{\odot} r_{\odot sc x}}{|\mathbf{r}_{\odot sc}|^3} + \frac{\mu_{\oplus} r_{\oplus(x)}}{|\mathbf{r}_{\oplus}|^3} + \frac{\mu_{\odot} r_{\odot(x)}}{|\mathbf{r}_{\odot}|^3} + \nabla U_{(x)} + u_x \quad (5d)$$

$$\dot{x}_5 = -2\omega_m x_4 + \omega_m^2 x_2 - \frac{\mu_{\oplus} r_{\oplus sc y}}{|\mathbf{r}_{\oplus sc}|^3} - \frac{\mu_{\odot} r_{\odot sc y}}{|\mathbf{r}_{\odot sc}|^3} + \frac{\mu_{\oplus} r_{\oplus(y)}}{|\mathbf{r}_{\oplus}|^3} + \frac{\mu_{\odot} r_{\odot(y)}}{|\mathbf{r}_{\odot}|^3} + \nabla U_{(y)} + u_y \quad (5e)$$

$$\dot{x}_6 = -\frac{\mu_{\oplus} r_{\oplus sc z}}{|\mathbf{r}_{\oplus sc}|^3} - \frac{\mu_{\odot} r_{\odot sc z}}{|\mathbf{r}_{\odot sc}|^3} + \frac{\mu_{\oplus} r_{\oplus(z)}}{|\mathbf{r}_{\oplus}|^3} + \frac{\mu_{\odot} r_{\odot(z)}}{|\mathbf{r}_{\odot}|^3} + \nabla U_{(z)} + u_z \quad (5f)$$

In equation (5), μ is the gravitational parameter, ω_m is the rotational velocity of the selenocentric frame (2.66169×10^{-6} rad/sec), u is the control acceleration, with the subscripts x , y , and z corresponding to the Cartesian component of the particular vector.

The $\nabla U_{(i)}$ terms in equations (5d-f) are the components of the gradient of the lunar gravitational potential U , and admit the perturbations due to the nonspherical lunar gravity model. A method for computing $\nabla U_{(i)}$ is given in reference [19]. The computation of the gravitational potential function requires knowledge of the lunar gravity field in terms of $C_{l,m}$ and $S_{l,m}$ coefficients. These coefficients are typically computed from observing tracking data from missions that have been conducted about the desired central body and are compiled to form gravity models. Current lunar gravity models containing C and S coefficients have been determined from previous lunar missions such as the Clementine and Lunar Prospector missions. The lunar gravity model used in this work is the LP165P gravity model obtained from

the NASA PDS Geosciences Node, which contains lunar C and S coefficients up to a degree (l) and order (m) of 165×165 [2]. However, sufficient accuracy is obtained using only a subset of the coefficients, and testing showed that using coefficients up to degree and order 25×25 was adequate for the nature of this study.

The equations of motion (5) are integrated using the “ode113” integrator in MATLAB, which is a variable order and stepsize Adams-Bashforth-Moulton PECE solver. The inputs that the program requires are the initial orbital elements, the initial Julian date, and the maximum degree and order of the lunar gravity model.

In order to demonstrate the general effects of the perturbations on a lunar orbiter’s trajectory, a 100-day “baseline” simulation of the dynamic model was performed with orbital element histories shown in Fig. 2. The simulation was performed with a starting Julian Date of 2451545 (corresponding to January 1st, 2000) and with lunar gravity terms up to 25×25 . The initial orbital elements for the simulation are given in Table 1, and correspond to an initial low-altitude, near polar, and near circular orbit. No control accelerations were applied in this simulation.

TABLE 1. Initial Orbital Elements for 100-Day Simulation

<i>Orbital Parameter</i>	<i>Value</i>
Semimajor axis, a_i	1837.4 km
Eccentricity, e_i	0.0013
Inclination, i_i	90°
Ascending Node, Ω_i	45°
Argument of Periapsis, ω_i	45°
True Anomaly, ν_i	0°

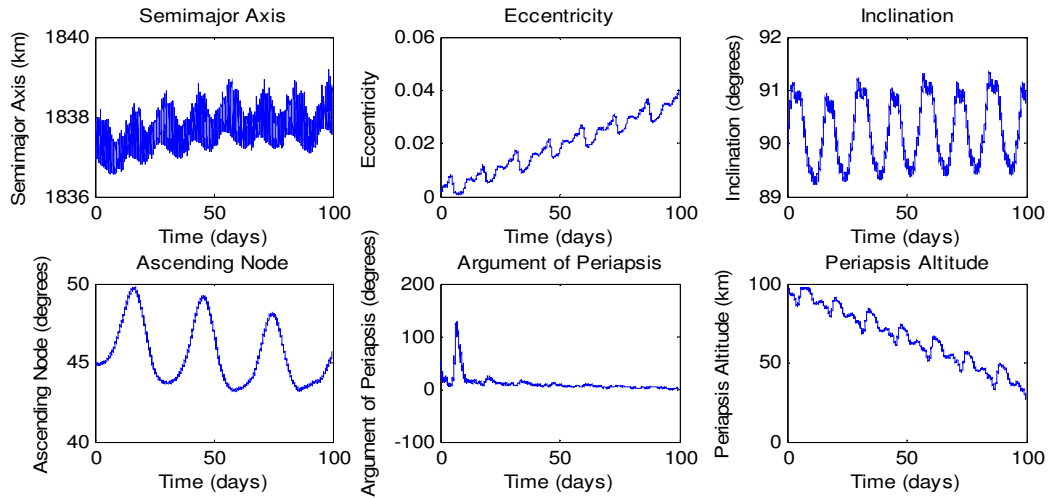


FIG. 2. 100-Day Dynamic Model Simulation.

Figure 2 shows that a low-altitude lunar orbit will experience drifts in semimajor axis and eccentricity that will eventually need to be corrected depending on how long it is desired for the satellite to remain in orbit. This drift is primarily due to the effects of the nonspherical lunar gravity field. Also, it is observed that near the beginning of the simulation the argument of periapsis experiences a spike. This occurs due to the eccentricity approaching zero in that region, making the argument of periapsis poorly defined. It can also be seen from Fig. 2 that the ascending node and inclination both stay in roughly the same region, performing oscillations of similar amplitudes.

The sixth plot in Fig. 2 shows the periapsis altitude, h_p , giving a good indication of the risk of the spacecraft impacting the Moon. The periapsis altitude plot shows that after 100 days, the periapsis altitude will have decreased by 80 km, leading to the conclusion that in a long-term mission the spacecraft will eventually impact unless maneuvers are made to correct the semimajor axis and eccentricity.

Overview of Optimal Control Theory

The general optimal control problem involves finding the optimal control input \mathbf{u} for a nonlinear system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t), \quad \mathbf{x}_{t_0} \text{ given} \quad (6)$$

where \mathbf{x} and \mathbf{u} are vectors of length n and m respectively that will minimize a performance index J of the form

$$J = \phi(\mathbf{x}_{t_f}, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t) dt \quad (7)$$

while satisfying the constraint

$$\psi(\mathbf{x}_{t_f}, t_f) = 0 \quad (8)$$

In equation (7), $\phi(\mathbf{x}_{t_f}, t_f)$ is a function of the terminal state \mathbf{x}_{t_f} that is desired to be minimized and $L(\mathbf{x}, \mathbf{u}, t)$ is a cost function that is to be minimized over the course of the integration. The $\phi(\mathbf{x}_{t_f}, t_f)$ function is considered a “soft” constraint because while it is desired to be as small as possible, it is not constrained to be necessarily zero. The $\psi(\mathbf{x}_{t_f}, t_f)$ function, on the other hand, is a “hard” constraint that must be exactly zero. The decision of whether to enforce either a soft or hard constraint (or both) is entirely up to the user.

Now, the system dynamics specified by equation (6) can be adjoined to the cost function $L(\mathbf{x}, \mathbf{u}, t)$ to form the Hamiltonian H as

$$H = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}, t) \quad (9)$$

where $\boldsymbol{\lambda}$ is the costate vector $\boldsymbol{\lambda}^T = [\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n]^T$. As detailed in reference [20], the *necessary* conditions for a minimum of equation (7) can be found by taking the partial derivatives of H with respect to \mathbf{x} , \mathbf{u} , and $\boldsymbol{\lambda}$. The necessary conditions for a minimum of equation (7) are then

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (10)$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}^T \boldsymbol{\lambda} - \frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}^T \quad (11)$$

$$\frac{\partial H}{\partial \mathbf{u}} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}}^T \boldsymbol{\lambda} + \frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}} = 0 \quad (12)$$

The boundary condition for the costate dynamics in equation (11) is

$$\boldsymbol{\lambda}^T(t_f) = \frac{\partial \phi(\mathbf{x}_{t_f}, t_f)}{\partial \mathbf{x}_{t_f}} + \mathbf{v}^T \frac{\partial \psi(\mathbf{x}_{t_f}, t_f)}{\partial \mathbf{x}_{t_f}} \quad (13)$$

where \mathbf{v} is a vector of undetermined multipliers.

It should be noted that the necessary conditions for a minimum do not guarantee that the solution is a minimum. Rather, the necessary conditions reveal the *stationary* points, which include minima, maxima, and saddle points. In order to prove that the solution is a minimum, the *sufficient* conditions for optimality must be satisfied, which involve analyzing the second derivatives of H . However, in practice, some optimal control problems are solved using only the necessary conditions and then the resulting solution is visually inspected to determine if it is a minimum. For many problems, it is obvious to observe if the optimal solution is in fact a minimum.

By inspection of the necessary conditions (10-12), it can be seen that in order to find the optimal solution, the $2n$ differential equations (10) and (11) must be integrated from t_0 to t_f , where the optimal

control vector \mathbf{u} at each timestep is found using equation (12). Of the $2n$ boundary conditions, n are specified at t_0 ($\mathbf{x}(t_0)$ given) and n are specified at t_f (equation (13)). This implies that the solution of the optimal control problem involves solving a two-point boundary value problem, which is typically very difficult to solve for nonlinear systems and requires numerical methods. The following section presents a simple widely-used method for solving such a boundary value problem.

Solving the Two-Point Boundary Value Problem

Indirect Versus Direct Methods

Due in part to the fact that optimal control theory has been used for numerous applications over the past 40 years, much work has been done in generating methods and algorithms for solving optimal control problems. Reference [21] gives an overview of many of the methods currently being used. Of all the current methods available, most can be classified as one of two types: indirect or direct.

Indirect methods involve deriving the expressions for the necessary conditions (10-12) and then solving the resulting two-point boundary value problem. In general, these methods consist of first making a guess for the initial costate vector, $\lambda(t_0)$, and then applying an algorithm to correct that initial guess until equation (13) is satisfied. This leads to one of the disadvantages of indirect methods, in that they typically require a very good guess for the initial costates in order to achieve convergence. However, if a sufficient initial guess is available, these methods benefit from quick convergence and a guaranteed optimal solution.

Direct methods, on the other hand, avoid the calculation of the necessary conditions (10-12) and instead discretize the optimal control problem, transforming it into a nonlinear programming problem. Parametric representations of the control and state variables are introduced, and the control variables are adjusted at each iteration in an attempt to continually reduce the performance index. An immediate advantage of these types of methods is that the necessary conditions for optimality do not need to be derived, making these types of methods more appealing for complex systems. Direct methods also tend to have a much larger region of convergence than indirect methods, and therefore do not require as good of a guess for the initial costates. However, direct methods also have disadvantages. In particular, they tend to take longer to converge and do not necessarily guarantee an optimal solution since the necessary conditions are avoided. Direct methods also tend to be more elaborate and complex than indirect methods.

Both indirect and direct methods have their own advantages and disadvantages, as summarized above. However, if a good guess for the initial costate vector is available, indirect methods can be a good choice since they tend to converge more quickly, guarantee an optimal solution, and are relatively simple to implement. It was found, from simulations, that for the special case in which the initial states are on or near their optimal values and do not need to be corrected with an initial acceleration, an initial guess of $\lambda(t_0) = \mathbf{0}$ was sufficient for convergence. For this reason, an indirect method was chosen for this work, which is described in the following section.

Indirect Shooting Method

In order to solve the optimal control problem presented by equations (10-13), a simple indirect shooting method described in references [20] and [22] is used. A summary of the algorithm follows:

- *Step 1:* Guess the n unspecified costates for the $\lambda(t_0)$ vector.
- *Step 2:* Numerically integrate equations (10) and (11) from t_0 to t_f , using equation (12) to solve for the control vector \mathbf{u} .
- *Step 3:* Record $\lambda(t_f)$.

- *Step 4:* Determine the $(n \times n)$ transition matrix $\left[\frac{\partial \boldsymbol{\mu}(t_f)}{\partial \boldsymbol{\lambda}_{t_0}} \right]$, where

$$\boldsymbol{\mu}(t_f)^T \triangleq \left[\lambda_1(t_f) \quad \dots \quad \lambda_n(t_f) \right]^T \quad (14)$$

This matrix is found through a finite differencing method by performing n additional integrations of equations (10) and (11), where in each of the integrations one of the $\lambda_i(t_0)$ values is changed by a small amount (typically a change of 1×10^{-9} was used here). After each of the integrations,

the quantities $\delta\boldsymbol{\mu}(t_f)$ are calculated as the difference between the final costates achieved and the values recorded in Step 3

$$\delta\boldsymbol{\mu}(t_f)^T = [\partial\lambda_1(t_f) \quad \dots \quad \delta\lambda_n(t_f)]^T \quad (15)$$

- *Step 5:* Compute another $\delta\boldsymbol{\mu}(t_f)$ vector equal to

$$\delta\boldsymbol{\mu}(t_f) = \boldsymbol{\lambda}(t_f) - \boldsymbol{\lambda}(t_f)_{desired} \quad (16)$$

which will bring the next solution closer to the desired final costate values. This $\delta\boldsymbol{\mu}(t_f)$ is found by comparing the final costates recorded in Step 3 with the desired values specified by equation (13).

- *Step 6:* With the $\delta\boldsymbol{\mu}(t_f)$ vector computed in Step 5, invert the transition matrix from Step 4 to find $\delta\boldsymbol{\lambda}(t_0)$ as

$$\delta\boldsymbol{\lambda}(t_0) = \left[\frac{\partial\boldsymbol{\mu}(t_f)}{\partial\boldsymbol{\lambda}(t_0)} \right]^{-1} \delta\boldsymbol{\mu}(t_f) \quad (17)$$

- *Step 7:* Calculate the corrected initial costate vector as

$$\boldsymbol{\lambda}(t_0)_{new} = \boldsymbol{\lambda}(t_0)_{old} + \delta\boldsymbol{\lambda}(t_0) \quad (18)$$

and then repeat Steps 1-6 until the required $\delta\boldsymbol{\lambda}(t_0)$ from equation (17) is sufficiently small.

Controller Design

Two controllers were created for use in this work, named Controller A and Controller B. Controller A is used for a majority of the total mission time, and is designed to minimize a cost function that includes an ascending node “error” term and the sum of the squares of the control accelerations. It was found that Controller A cannot be used during equatorial crossings due to the physical inability to control the ascending node in these regions, so when entering these regions a switch is made to Controller B, a terminal state controller. Once the spacecraft leaves the equatorial region, a switch is then made back to Controller A. The goal of this combined controller effort is to maintain the Sun-synchronous condition for any specified mission duration. In the remainder of this section, the design of both controllers is discussed.

Controller A

Controller A is the primary controller used in this work, and is always active except for a small range in latitude around equatorial crossings. The goal of this controller is to maintain the Sun-synchronous condition while minimizing the total amount of control acceleration. In the future, this controller will be modified to control other orbital elements as well, such as the inclination, eccentricity, and semimajor axis.

The performance index used in the creation of Controller A is of the form

$$J = \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t) \quad (19)$$

No terminal constraints are imposed, so the performance of this controller depends entirely on the selection of the cost function $L(\mathbf{x}, \mathbf{u}, t)$. Since at this stage Controller A is only required to maintain the Sun-synchronous condition while minimizing the total control acceleration, a cost function was chosen as

$$L(\mathbf{x}, \mathbf{u}, t) = w_1 \left[\cos(\Omega_R(t)) - \cos(\Omega_{t_0} + \dot{\Omega}_{sunsynch}(t-t_0) - \omega_m(t-t_0)) \right]^2 + w_2 (u_x^2 + u_y^2 + u_z^2) \quad (20)$$

where $\Omega_R(t)$ is the actual (simulated) ascending node with respect to the rotating frame, Ω_{t_0} is the initial ascending node with respect to the inertial frame, and $\dot{\Omega}_{sunsynch}$ is the desired Sun-synchronous rate of 0.9856 °/day. The ascending node is expressed with a cosine function in order to simplify the partial derivatives of the cost function. Also, w_1 and w_2 are weighting parameters, used to define the priority of

minimizing one element of the cost function over another. Weight w_1 is unitless while w_2 has units of s^4/km^2 in order for the units in equation (20) to be consistent.

The first term in equation (20) involves minimizing the difference between the actual ascending node and the desired Sun-synchronous ascending node. The $\omega_m(t-t_0)$ portion of this term accounts for the fact that the ascending node Ω_R is measured with respect to a rotating frame rather than an inertial frame. The second term in equation (20) indicates that the total control acceleration is to be minimized.

In terms of the state space coordinates, the cost function (20) can be written as

$$L(\mathbf{x}, \mathbf{u}, t) = w_1 \left[\frac{x_1 x_6 - x_4 x_3}{\sqrt{(x_2 x_6 - x_3 x_5)^2 + (x_4 x_3 - x_6 x_1)^2}} - \cos(\Omega_{t_0} + \dot{\Omega}_{\text{sunsynch}}(t-t_0) - \omega_m(t-t_0)) \right] + w_2 (u_x^2 + u_y^2 + u_z^2) \quad (21)$$

where, as before, $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$.

Now the necessary conditions (10-12) can be applied. Evaluating equation (10) specifies the state dynamics, which are defined by equation (5) with the boundary condition $\mathbf{x}(t_0)$ given. The second condition, equation (11), specifies the costate dynamics as

$$\dot{\lambda} = -\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)^T}{\partial \mathbf{x}} \lambda - \frac{\partial L(\mathbf{x}, \mathbf{u}, t)^T}{\partial \mathbf{x}} \quad (22)$$

It can be seen that solving equation (22) requires the calculation of $\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}$ (a 6x6 matrix) and $\frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}$ (a 1x6 row vector). Due to space considerations these partials are not listed here, but it is noted that they are straightforward to obtain. Since there are no terminal constraints imposed, equation (13) gives the boundary condition for the costate dynamics as

$$\lambda^T(t_f) = \mathbf{0} \quad (23)$$

The final necessary condition is given by equation (12) as

$$\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)^T}{\partial \mathbf{u}} \lambda + \frac{\partial L(\mathbf{x}, \mathbf{u}, t)^T}{\partial \mathbf{u}} = \mathbf{0} \quad (24)$$

where for this analysis $\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}}$ and $\frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}}$ are obtained as

$$\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{u}} = [2w_2 u_x \quad 2w_2 u_y \quad 2w_2 u_z] \quad (25)$$

Substituting equation (25) into equation (24) and solving for the control accelerations leads to

$$u_x = -\frac{1}{2w_2} \lambda_4 \quad (26a)$$

$$u_y = -\frac{1}{2w_2} \lambda_5 \quad (26b)$$

$$u_z = -\frac{1}{2w_2} \lambda_6 \quad (26c)$$

Equations (26a-c) provide a useful relationship between the control accelerations and the costates and show that the control accelerations depend entirely on λ_4 , λ_5 , and λ_6 .

This analysis results in twelve differential equations supplied by equations (5) and (22) with boundary conditions $\mathbf{x}(t_0)$ given and equation (23), where the three control acceleration components are found via equations (26a-c). This two-point boundary value problem is solved using the indirect shooting method described previously using an initial guess of $\lambda(t_0) = \mathbf{0}$.

One limitation of Controller A comes from the fact that it cannot be used during equatorial crossings because the ascending node cannot physically be controlled in these regions. In order to account for this limitation, a switch is made to a second controller, Controller B, at 10° in latitude *before* an equatorial crossing. Controller B is then used until 10° in latitude *after* the equator crossing, at which point a switch is made back to Controller A. However, it is noted that since a switch is made between controllers, the resulting trajectories must be considered suboptimal. It is also noted that the choice to switch controllers at 10° is an arbitrary one, and other values of latitude are under consideration.

Comparing equations (23) and (26) indicates another difficulty of Controller A not being able to operate during equatorial crossings. Controller A is designed such that the final costates are constrained to zero from equation (23), and since equation (26) reveals a linear relationship between the control accelerations and the costates, this implies that the final control accelerations will also necessarily be zero. However, constraining the control accelerations to zero 10° before an equatorial crossing is not beneficial, because it is near equatorial crossings where the ascending node requires the largest control effort. On the other hand, during a polar crossing, the ascending node can be controlled with the least amount of control effort and it would be plausible to constrain the control at the polar crossing to zero. For this reason, the operation of Controller A is divided into two segments between equator crossings. The first segment is a forward integration from 10° after an equatorial crossing up to the next polar crossing, where the costates (and thus control accelerations) are constrained to be zero at the polar crossing. The second segment is a *backward* integration from 10° before the next equatorial crossing to the polar crossing, where again the costates are constrained to be zero at the polar crossing. These two segments are then pieced together to obtain the full trajectory between equatorial crossings.

Controller B

While Controller A attempts to continuously maintain the Sun-synchronous condition, Controller B is designed as a terminal state controller which enforces the constraint that the orbit must only be Sun-synchronous at the *end point* of the controller's duration. In other words, while there are no constraints placed on the ascending node's motion during the integration, at the end of the integration the ascending node must converge to the desired Sun-synchronous ascending node location.

Thus, while Controller B is designed to minimize a performance index in the form of equation (19), the cost function $L(\mathbf{x}, \mathbf{u}, t)$ is simplified to

$$L(\mathbf{x}, \mathbf{u}, t) = \left(u_x^2 + u_y^2 + u_z^2 \right) \quad (27)$$

in order to incorporate the fact that no constraint is placed on the ascending node *during* the integration.

Once again, the necessary conditions can be applied. As with the analysis for Controller A, the evaluation of the first necessary condition leads to the state dynamics defined by equation (5) with $\mathbf{x}(t_0)$ given. Applying the second necessary condition (11) and noting that $\frac{\partial L(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}} = \mathbf{0}$, the costate dynamics are obtained as

$$\dot{\lambda} = - \frac{\partial f(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}}^T \lambda \quad (28)$$

Aside from a simplified set of dynamics for the costates, the only other difference between Controller B and Controller A is in the terminal boundary conditions. While the terminal boundary conditions for Controller A are found by evaluating equation (13), the terminal boundary condition for Controller B corresponds to $\mathbf{x}(t_f)$ given, where $\mathbf{x}(t_f)$ corresponds to a polar, Sun-synchronous orbit. The additional requirement that the terminal state correspond to a polar orbit was included in order to ensure that the inclination is never allowed to stray very far from 90° . Essentially, the six conditions given by equation (13) are traded with another set of conditions, $\mathbf{x}(t_f)$ given, so that the problem still has twelve boundary conditions [20].

In summary, Controller B consists of solving twelve differential equations supplied by equations (5) and (28) with boundary conditions of $\mathbf{x}(t_0)$ and $\mathbf{x}(t_f)$ given, where the three control acceleration

components are again found from equations (26a-c). As with Controller A, this two-point boundary value problem is solved using the indirect shooting method with an initial guess of $\lambda(t_0)=\mathbf{0}$. However, since the specified terminal boundary condition is in terms of the states rather than the costates, the $\delta\mathbf{\mu}(t_f)$ vectors required in the indirect shooting algorithm are found by comparing the final values of the states rather than the costates, taking the form

$$\delta\mathbf{\mu}(t_f)^T = [\delta x_1(t_f) \quad \delta x_2(t_f) \quad . \quad . \quad . \quad \delta x_6(t_f)]^T \quad (29)$$

It should be stressed that Controller B is *only* used in a range of 10° in latitude before and after each equatorial crossing, which, for an orbit with a semimajor axis of 1837.4 km, accounts for only approximately six minutes of the total orbital period (roughly 120 minutes). Controller B is only needed because of Controller A's physical limitation in these short regions.

An Analytic Solution

In this section, an approximate analytic solution for the Sun-synchronous stationkeeping problem is found using Gauss's variational equations of motion [19]. This solution is presented for the simplified case of a spacecraft orbiting a perfectly spherical Moon with no third-body effects. The analytic solution is then compared to the results from Controller A described in the previous section for validation.

Gauss's variational equations of motion provide analytic equations for the rates of change of the classical orbital elements as a function of a perturbing acceleration $\mathbf{F} = F_r\hat{\mathbf{r}} + F_\theta\hat{\boldsymbol{\theta}} + F_h\hat{\mathbf{h}}$, where the $\hat{\mathbf{r}} - \hat{\boldsymbol{\theta}} - \hat{\mathbf{h}}$ frame is identical to the RSW satellite coordinate system defined in reference [19]. In particular, the equation for the rate of change of the ascending node is given as

$$\dot{\Omega} = \frac{r \sin \theta}{h \sin i} F_h \quad (30)$$

where θ is the argument of latitude of the spacecraft and is calculated as the sum of the argument of periapsis and true anomaly. As would be expected, equation (30) shows that a change in the ascending node can only be caused by an acceleration (F_h) that is applied normal to the orbit plane.

Now, assume that no third-body forces are present and the perturbing acceleration \mathbf{F} is provided from control hardware. Equation (30) can be rearranged and solved for the control acceleration F_h as

$$F_h = \frac{h \sin i}{r \sin \theta} \dot{\Omega} \quad (31)$$

Equation (31) gives an approximate analytic expression for the control acceleration required to produce a desired $\dot{\Omega}$. It is noted that equation (31) experiences a singularity at θ values of 0° and 180° . This implies that the ascending node cannot be controlled during equator crossings, as was also discussed in the design of Controller A.

By setting $\dot{\Omega}$ to the Sun-synchronous rate ($0.9856^\circ/\text{day}$), equation (31) allows for the estimate of a control force history that will produce a Sun-synchronous orbit. This control force history can then be used to validate the control histories produced by Controller A. Since the derivation of equation (31) requires the assumption of a perfectly spherical Moon with no third-body effects, for this comparison the equations of motion (5) are simplified by setting μ_\oplus and μ_\odot to zero and by replacing ∇U_c with $-\frac{\mu_c}{r_{s/c}^3} \mathbf{r}_{s/c}$.

The initial orbital elements for this simulation can be found in Table 2. Figure 3 shows a comparison of the analytic control history generated by equation (31) and the numerical control history produced from implementing Controller A for a spacecraft with a mass of 1000 kg.

TABLE 2. Initial Orbital Elements for Analytic Control Simulation

<i>Orbital Parameter</i>	<i>Value</i>
Semimajor axis, a_i	1837.4 km
Eccentricity, e_i	0.0013
Inclination, i_i	90°
Ascending Node, Ω_i	45°
Argument of Periapsis, ω_i	10°
True Anomaly, V_i	0°

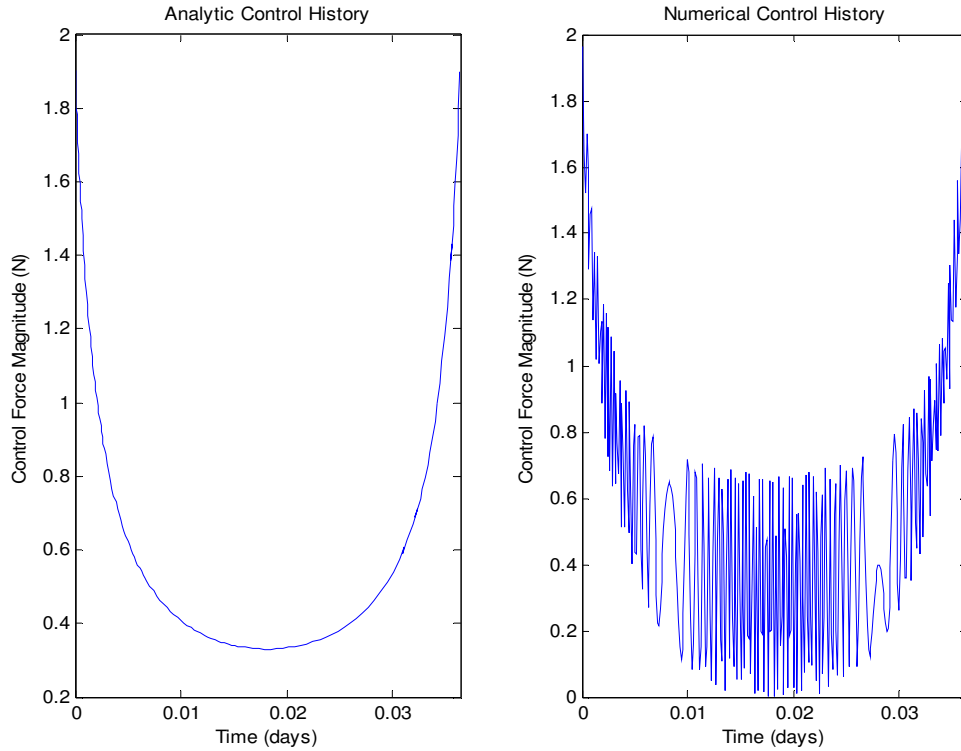


Figure 3: Comparison of Analytic and Numerical Control Histories.

From an inspection of Fig. 3, it is seen that while both the analytic and numerical control histories follow the same general pattern with equivalent average control values, the numerical control history shows a much larger variation exhibiting a “chattering” behavior. A possible explanation for this behavior is that the numerical control history was created such that the control force at the polar crossing (the midpoint of the plot) is constrained to zero. The analytic control history indicates a finite (rather than zero) control force at the polar crossing.

Results

In this section, a 50-day simulation is presented in order to show the effectiveness of the controllers at maintaining a Sun-synchronous orbit. The initial orbital elements presented in Table 2 were used for this simulation, which correspond to a low-altitude, near-circular, polar orbit. Also, lunar gravity terms were incorporated up to degree and order 25x25. Finally, the weighting parameters w_1 and w_2 in equation (23) were set to 1 and 20 s⁴/km² respectively. Figure 4 shows the orbital element histories for the simulation, and Fig. 5 shows the control force component magnitudes for a 1000 kg spacecraft.

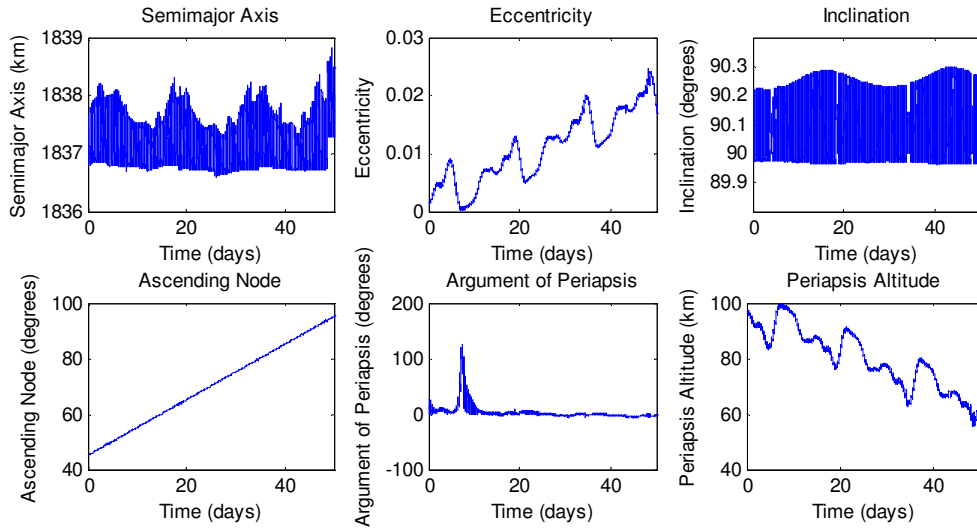


FIG. 4. Orbital Element Trajectories for 50-Day Control Simulation.

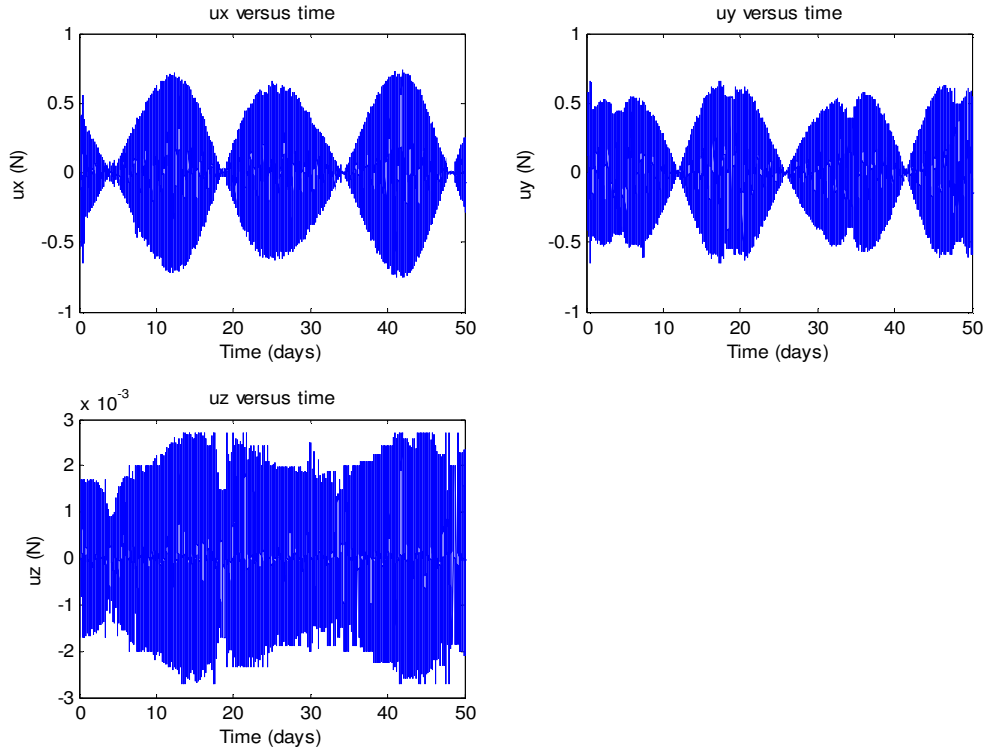


FIG. 5. Control Force Magnitudes for 50-Day Control Simulation.

As shown in Fig. 4, the ascending node rotates approximately 50° in 50 days as required. During the simulation, the ascending node trajectory never varies from the desired Sun-synchronous line by more than 0.1° . The inclination also remains near 90° , with a maximum deviation of approximately 0.3° . Therefore, a polar, Sun-synchronous orbit has successfully been achieved and maintained. Also, upon comparison of Fig. 4 and Fig. 2, it can be seen that for this controlled simulation the other orbital elements follow their uncontrolled trajectories and are thus not significantly affected by the continuous thrusting as expected.

Figure 5 shows that the polar, Sun-synchronous orbit is maintained using low thrust levels. In particular, the maximum control force applied along the \hat{x} and \hat{y} directions is approximately 0.76 N. It can also be seen that the control force in the \hat{z} direction is almost negligible. This is expected because the inclination is close to 90° and the \hat{z} axis is thus almost entirely contained inside the orbit plane.

While the trajectories in Fig. 4 appear promising, an examination of the periapsis altitude history raises a concern. Figure 4 shows that the periapsis altitude decreases rather rapidly during the simulation, by approximately 40 km in 50 days. This behavior would eventually lead to the spacecraft impacting the lunar surface. It should be noted that this decrease in the periapsis altitude occurs due to the dynamics of the system (particularly the higher-order lunar gravity terms) and *not* due to the continuous thrusting.

The periapsis altitude behavior is detrimental not only because it indicates an eventual impact with the Moon, but also because a requirement for a successful mapping mission is the maintenance of a relatively constant spacecraft altitude. It is desired for all ground images taken by the camera on the spacecraft to have relatively constant viewing conditions (one of the primary reasons a Sun-synchronous orbit was sought), and a varying altitude would impair this. It is for this reason that studies are currently being performed that include the control of other orbital elements into the algorithm in order to obtain an orbit that would be more suitable for a mapping mission.

It should be noted that while the simulation presented in this section is for a 50-day mission, it would be very straightforward to perform the same analysis for longer duration missions. However, since the current method does not control the decrease in periapsis altitude, simulations can only be performed up to a certain duration at which the spacecraft would impact. Once the method is modified to maintain a constant altitude, simulations will be performed for a wide range of possible mission durations.

Conclusion

A method for obtaining and maintaining a polar, Sun-synchronous orbit using continuous low-thrust systems has been presented. A simulation of the method for a 50-day mission is shown, and it is seen that the orbit can be maintained using thrust levels smaller than 1 N.

Optimal control theory was used in the creation of two controllers, Controller A and Controller B. The design of each controller resulted in a two-point boundary value problem, which is solved using an indirect shooting method. It is shown how Controller A and Controller B are used in tandem, with Controller A operating from 10° in latitude after an equatorial crossing to 10° before the next equatorial crossing while Controller B operates in the remaining 20° latitude swaths around the equator. Since switching is performed between the two controllers, the combined trajectories can only be considered suboptimal.

Future additions to this work will include incorporating the control of additional orbital elements into the controller design. The overall goal will be to have a controller that can not only generate a suitable mapping orbit, but that can be sufficiently versatile to be applied to a large variety of mission applications. In order to improve the versatility of the controller design, a search will also be made for possible direct optimal control methods that can be used for this work. The indirect shooting method, while relatively easy to implement, is somewhat limited in terms of robustness and its requirement for a good initial costate guess to achieve convergence. A sequential quadratic programming method is currently being examined as a possible replacement for the method used in this work.

Finally, future additions to this work will also consider the implementation of thruster constraints. An assumption made in this study is that control hardware is available that can produce a requested thrust in any direction and with any magnitude. Of course, modern control hardware is typically capable of producing only a specific magnitude of thrust that often cannot be throttled, and also may not necessarily have the capability of thrusting independently along three orthogonal directions. For this reason, future work will involve implementing constraints on the thrust magnitude and direction.

References

- [1] BECKMAN, MARK "Mission Design for the Lunar Reconnaissance Orbiter," presented as paper AAS 07-057 at the 29th Annual AAS Guidance and Control Conference, Breckenridge, Colorado, Feb. 4-8, 2006.

- [2] KONOPLIV, A.S. "Recent Gravity Models as a Result of the Lunar Prospector Mission," *Icarus*, Vol. 150, No. 1, pp. 1-18, 2001.
- [3] URL: <http://nssdc.gsfc.nasa.gov/planetary/lunar/ranger.html>.
- [4] URL: <http://nssdc.gsfc.nasa.gov/planetary/lunar/surveyor.html>.
- [5] URL: <http://nssdc.gsfc.nasa.gov/planetary/lunar/lunarussr.html>.
- [6] PARK, S.Y. and JUNKINS, JOHN L. "Orbital Mission Analysis for a Lunar Mapping Satellite," *The Journal of the Sciences*, Vol. 43, No. 2, April-June 1995, pp. 207-217.
- [7] RAMANAN, R.V. and ADIMURTHY, V. "An Analysis of Near-Circular Lunar Mapping Orbits," *Journal of Earth System Sciences*, Dec. 2005, pp. 619-626.
- [8] KAUFMAN, B., MIDDOUR, J., and RICHON, K. "Mission Design of the Clementine Space Experiment," AAS-95-124, February, 1995.
- [9] RICHON, KAREN V. and NEWMAN, LAURI K. "Flight Dynamics Support for the Clementine Deep Space Program Science Experiment (DSPSE) Mission," *Advances in the Astronautical Sciences*, Vol. 90, No. 2, 1995, pp. 2045-2064.
- [10] FOLTA, DAVID and BECKMAN, MARK "The Lunar Prospector Mission: Results of Trajectory Design, Quasi-Frozen Orbits, Extended Mission Targeting, and Lunar Topography and Potential Models," *Advances in the Astronautical Sciences*, Vol. 103, No. 2, 1999, pp. 1505-1523.
- [11] EDELBAUM, T.N. "Optimal Low-Thrust Rendezvous and Station Keeping," *Journal of Spacecraft and Rockets*, Vol. 40, No. 6, 1964.
- [12] HUNZIKER, RAUL R. "Low-Thrust Station Keeping Guidance for a 24-Hour Satellite," *AIAA Journal*, Vol. 8, No. 7, 1970, pp. 1186-1192.
- [13] OLESON, STEVEN R., MYERS, ROGER M., and KLUEVER, CRAIG A. "Advanced Propulsion for Geostationary Orbit Insertion and North-South Station Keeping," *Journal of Spacecraft and Rockets*, Vol. 34, No. 1, 1997.
- [14] CHAN, J., ARIASTI, A. and HUR-DIAZ, S. "Comparisons of Two Station-Keeping Strategies for Ionic Propulsion Systems," Flight Mechanics Symposium at NASA Goddard Space Flight Center, October 28-30, 2003.
- [15] LOSA, DAMIANA "Electric Station Keeping of Geostationary Satellites: a Differential Inclusion Approach," 44th IEEE Conference on Decision and Control, December, 2005.
- [16] SKINNER, DAVID L. "QUICK – An Interactive Software Environment for Engineering," presented as paper AIAA-1989-3051 at the AIAA Computers in Aerospace Conference, Monterey, CA, Oct 3-5, 1989, pp. 542-563.
- [17] SKINNER, DAVID L. "Spacecraft Design Applications of QUICK," presented as paper AIAA-1992-1111 at the Aerospace Design Conference, Irvine, CA, Feb 3-6, 1992.
- [18] ESCOBAL, PEDRO R. *Methods of Astrodynamics*, John Wiley & Sons, 1969.
- [19] VALLADO, DAVID A. *Fundamentals of Astrodynamics and Applications*, Microcosm Press, Second Edition, 2004.
- [20] BRYSON, A. and HO, YU-CHI, *Applied Optimal Control*, Taylor & Francis, Revised Edition, 1975.
- [21] BETTS, JOHN T. "Survey of Numerical Methods for Trajectory Optimization," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, pp. 193-207, March-April 1998.
- [22] LEWIS, F. and SYRMOS, V. *Optimal Control*, Wiley Interscience, Second Edition, 1995.