Lyapunov Feedback Control in Equinoctial Elements Applied to Low Thrust Control of Elliptical Orbit Constellations

by

Benjamin E. Joseph

B.S., Aerospace Engineering
University of Washington, 2003

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2006

© 2006 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: ____________________
Department of Aeronautics and Astronautics
January 20, 2006

Certified by: ____________________
Dr. Paul J. Cefola
Research Affiliate, Department of Aeronautics and Astronautics
Thesis Supervisor

Certified by: ____________________
Dr. Manuel Martinez-Sanchez
Professor, Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by: ____________________
Dr. Jaime Peraire
Professor, Department of Aeronautics and Astronautics
Chair, Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 10 2006
LIBRARIES

AERO

Lyapunov Feedback Control in Equinoctial Elements Applied to Low Thrust Control of Elliptical Orbit Constellations

by

Benjamin Ethan Joseph

Submitted to the Department of Aeronautics and Astronautics on January 20th, 2006 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aeronautics and Astronautics.

ABSTRACT

For a typical satellite constellation design problem, sometimes there are two conflicting goals. On one hand one wishes to provide continuous coverage of a particular area. On the other hand one would like to use the minimum number of satellites as possible. For coverage locations nearer the equator, these requirements are easily fulfilled by using a satellite in a geostationary orbit. However, coverage of higher latitude locales is not as effective with geostationary satellites because the viewing angle from the ground to the satellite will be closer to the horizon, which increases the likelihood of loss of signal due to some ground obstacle. Instead, a constellation consisting of satellites in eccentric orbits at high inclination angles, sometimes referred to as Molniya type orbits, can circumvent this problem and provide continuous coverage with very few satellites.

Elliptic-inclined constellations have been flown since the 1960s, chiefly by the governments of the former Soviet Union and the current Russian Federation. Not until the year 2000 has such a constellation been flown commercially. With the involvement of a commercial entity in flying constellations of this type, there has been some interest in reducing system costs by applying electric propulsion to spacecraft operating in these types of orbits. However, using electric propulsion for station keeping on elliptic-inclined constellations is not yet well characterized. For example, to maintain continuous coverage of a particular area can require these types of constellations to have a repeat ground track, thus greatly enhancing the effects of tesseral harmonics. Additionally, these orbits typically have a high apogee altitude, which increases the effect of perturbations due to third body point mass effects. Overall it is possible to see station keeping ΔV budgets much greater than one would typically see for a geostationary satellite. While there is some data on station keeping of an elliptical-inclined orbit using high thrust systems, very little of this appears to have entered the public domain. Furthermore, there seems to be no data for using electric propulsion to station keep elliptic-inclined constellations. Thus, the primary goal of this thesis is to remedy this gap in our knowledge. A generic and easy-to-use software tool based on Lyapunov feedback control and implemented in the Goddard Trajectory Determination System. This tool is used to determine station keeping ΔV budgets of two different elliptic constellations: John Draim's Teardrop orbit constellation and the Sirius Satellite Radio constellation.

Thesis Supervisor: Dr. Paul J. Cefola, Research Affiliate, MIT

Acknowledgement

A wise baseball player once said, "I'd rather be lucky than smart". My luck has been the people that are in my life—my family, friends, advisers, and all those who have dropped a line or two of helpful advice or some words of encouragement.

First, I am very grateful for the help of my academic adviser, Dr. Manuel Martinez-Sanchez in the space propulsion lab at MIT. It was Manuel who first put my thesis adviser and me in contact with each other.

I am extremely fortunate to have fallen in with a great group of friends from the space propulsion lab. Tanya, Felix, Michael, and Justin keeping me sane, entertained, and even helped me with my studies from time to time.

I would like to thank Zachary Folcik for his help in dealing with GTDS and various compiler issues. I am also grateful to Dr. Ron Proulx and Dr. Dave Carter for their helpful suggestions with regards to control law selection and the feasibility of including Lyapunov feedback control into the Draper Semi-analytic Satellite Theory.

Fred Donovan, the network administrator for the Aero/Astro Engineering Department, was essential in obtaining the astrodynamics workstation on which all of the simulation runs for this thesis were accomplished.

Without the efforts Captain (USN, ret.) John Draim there would be no "teardrop" orbit constellation. John was also very helpful with discussion of acceptable control tolerances of the ground track angle.

I would also like to acknowledge the effort of Anastassios Petropoulos, as much of this thesis is based on his work with the Q-law Lyapunov feedback control. Anastassios was also invaluable in validating the feasibility of using equinoctial elements with the Q-law.

I am eternally grateful to all of those who have had the "power of the purse strings" in one way or another that chose to support me. This includes my mother, Pam, my grandmother, Claire, and my thesis adviser, Paul Cefola. If it wasn't for the efforts of my mother in securing tuition during my undergraduate years it is doubtful if I would have completed the degree. The gift from my grandmother that allowed me to travel the Orient allowed me to change the way I approached life and gave me some time to think about things—this was instrumental in dealing with the application process to graduate school. Finally, the efforts of my thesis adviser kept me from starvation and copious amounts of debt.

Table of Contents

List of Tables

List of Figures

1.0 Introduction

1.1 Elliptic Constellations

For a typical satellite constellation design, there are two sometimes conflicting goals: One wishes to provide continuous coverage of a specific location, and one would like to use the minimum number of satellites as possible. The number of satellites in a constellation is strongly correlated to constellation cost. For coverage locations nearer the equator, these requirements are easily fulfilled by using a satellite in a geostationary orbit. However, coverage of higher latitude locales is not as effective with geostationary satellites because the elevation angle from the ground to the satellite will be closer to the horizon, which increases the likelihood of loss of signal due to some ground obstacle. In order to increase the elevation angle, a satellite must be flown at an inclined orbit instead of the equatorial geostationary orbit. If one were to just incline the satellite's orbit, it would spend equal time in the northern and southern hemisphere. In order to increase dwell time over any particular location on the rotating Earth, the orbit must be made elliptical as a satellite spends more time at apogee than at perigee. Accordingly, it is no coincidence that one of the words that we use to refer to elliptical-inclined orbits is Russian: Molniya. The scientists and engineers of the former Soviet Union were the first to think about using elliptical-inclined satellite orbits, as much of the country is located in medium and high latitude regions that are difficult to cover with geostationary satellites. Since 1964, the former Soviet Union and the Russian Federation have flown over 150 of these satellites.

Launched on April 23rd, 1965, the first of the Molniya satellites was lofted into an orbit with a semi-major axis of 26610 km, an eccentricity of about 0.74, an inclination of 65°, with the apogee centered over Soviet territory. The initial American reaction to the launch was that the Soviets had experienced a launch failure[1]. This launch came the same year that the Early Bird (Intelsat 1) was launched into geosynchronous orbit. One could speculate that the attention of the American analysts was fixated solely on the recently launched geosynchronous satellite, and that they were initially blind to the possibility of other useful orbits. Additional satellites were soon to follow, starting with a second launch of a Molniya satellite on the 14th of October of the same year. Since then, over 100 payloads[2] have been lofted into eccentric-inclined orbits by Russia, the latest of which was the launch of Molniya 3-53 in 2003[3]. To a much lesser extent, eccentric orbit payloads were also launched by the United States and other countries. These payloads were launched for the purpose of communications, ground observation, as well as a few science platforms[4].

A more recent development is that these types of orbits are starting to be used by commercial entities in the civilian marketplace. In the last decade two elliptical-inclined constellations had been planned, the Ellipso constellation and Sirius Satellite Radio's (SSR) constellation. Of these, only Sirius' has actually launched. For various non-technical reasons the Ellipso concept failed to pan out[5].

Another relatively new development has been the use of low thrust, high-Isp electric propulsion on satellites as station keeping thrusters. Only as recently as 1997 has the first geosynchronous satellite employed a Boeing XIPS ion thruster to handle north-south station keeping[6]. The main benefit from using electric propulsion is to reduce the fuel mass fraction, and hence the overall system mass of the satellite. Thus, the main impetus for adoption of electric propulsion technology for station keeping has been from the commercial sector. Lower system mass means decreased launch costs since a smaller booster can be chosen. Conversely, one may wish to use the same sized launch vehicle but just increase the number of transponders on-board.

Sooner rather than later these two trends will converge and someone will want to fly a satellite on an elliptical-inclined orbit and use electric propulsion for station keeping. Currently, there appears to be a dearth of such information on how to do this and what to expect in terms of a typical ΔV budget. The goal of this thesis is help remedy this situation. This can be accomplished by providing a method to easily determine how to station keep a satellite in *any* elliptical orbit with a low-thrust. This includes determining thruster pointing, station keep event duration, maintenance of ground track, and total ΔV budget.

In order to accomplish this, two things are needed: An accurate model of Earth's perturbative environment and a control method to figure out which direction to point the thruster when the satellite drifts off station.

There are a few accurate models of the force environment for Earth-orbiting satellites. Of these, the Goddard Trajectory Determination System (GTDS) was chosen for this because its ephemeris propagator is well-proven, it could be obtained for free, and the source code was readily modifiable. Determining a control method, however, is a much more difficult task.

1.2 The Low Thrust Control Problem

Problem Types

Simply put, the low thrust control problem is to determine where to point the satellite's thrust vector to get the satellite from state A to state B. The optimal problem is just accomplishing this while also minimizing some performance index.

When one only considers the two-body problem, there are eight types of low-thrust control problems based on three yes/no questions:

1) Is the problem going to take many orbits around the central body to converge? In other words, what is the ratio of the acceleration of the spacecraft due to thrusting and that of the central body? If the ratio is small, then the problem is of the *multiple revolution* type.
2) Is the change in the spacecraft state going to be large? For example, when a spacecraft undergoes a station keeping event, it does not change its orbital

elements by very much. But, when it uses electric propulsion for interplanetary trajectories its orbital elements will change significantly.

3) Are you trying to rendezvous with anything? In the equinoctial and Keplerian element formulations there are "slow" elements and "fast" elements. The true anomaly would be an example of a fast-element, but the semi-major axis or eccentricity would be a slow-element. If the desired final state of the spacecraft does not include the fast element then this is strictly a slow-element problem.

As one would expect, there are different control methods that can solve the different types of problems.

Solution Methods

After reading the literature that is available and discussing the low-thrust control problem with experienced individuals in industry[7], one can get a sense that there are three categories of control techniques: variational methods, direct methods, and heuristic techniques. The words *indirect* and *variational* can be used interchangeably. An indirect solution technique implies some optimality that is obtained with applying variational calculus. Direct methods, on the other hand, are not usually optimal but merely solve the boundary value problem.

Within these categories there are a large number of individual control methods to choose from. There are also some optimal analytical methods available simple cases (Edelbaum, Sackett, Malchow[8]; Konstantinov), but in general there are no analytical methods to solve the general problem. The following table is an incomplete list of numerical control methods. Most of this information comes from a survey of control techniques done by John T. Betts in 1997[9].

Table 1: List of Solution Methods to Low Thrust BVP

Control Method Name	**Category**
Direct Shooting	Direct

Indirect Shooting	Variational
Multiple Shooting	Direct/Variational
Indirect Transcription	Variational
Direct Transcription	Direct
Genetic Algorithms	Heuristic
Lyapunov Feedback Control*	Direct
Simulated Annealing*	Heuristic

*Not discussed in Betts paper

Betts goes into quite some detail as to the formulation of each control method, their advantages and disadvantages, as well as previous well-known implementations of each method. The scope of this thesis is not to brief the reader regarding each and every control method that exists, but more to explain the final choice of control method.

With the exception perhaps of direct shooting, the shooting and transcription methods are not simple to implement, but will usually solve the more difficult boundary value problems—these are the problems that require only one or two revolutions, a large change in elements, and the final state is a function of time (rendezvous). For extremely difficult problems that may have discontinuous constraints, genetic algorithms[10] may be the most appropriate choice.

Applications and Selection of Control Method

Current application of electric propulsion (EP) appears to be divided between two categories: It is used for station keeping on GEO satellites, and it is occasionally used for primary propulsion on space probes, most notably NASA's NEAR spacecraft and the ESA's SMART-1. SMART-1 used a small (64mN) Hall thruster to implement a many-revolution spiral from low Earth orbit to lunar orbit, whereas NEAR flew to its rendezvous in less than two orbits around the sun. Both spacecraft conducted a large element change transfer where the thruster was on constantly. In NEAR's case its NSTAR ion thruster was active for over 15,000 hours.

Station keeping GEO satellites, on the other hand, is a different problem. Station keeping operations are typically conducted once every two weeks and can last for a few hours at a time. They are small element change transfers completed in only one or two revolutions. The control problem that we wish to solve, namely that of station keeping a satellite in an elliptical orbit, is most similar to this type of problem.

Thankfully, the station keeping boundary value problem is much simpler to solve because the underlying assumptions of the control problem is that the satellite starts near an already converged state with respect to the fast variable. Thus, one can use a much simpler technique than either one of the transcription or shooting methods to solve this problem.

After weighing the pros and cons of each type of solution method, a technique using Lyapunov feedback seemed very direct and easy to apply to the station keeping problem. Lyapunov feedback control is a full state feedback control law where thrust direction is determined from current state information. This method seems particularly well suited for problems that are either multi-revolution or small-element change, but not for problems where the target state is time dependent.

This station keeping analysis technique will be applied to a few different elliptical constellations, including the Sirius Satellite Radio constellation and John Draim's "Teardrop" orbit constellation.

1.4 Thesis Overview

The organization of this document is fairly simple. First, one is introduced to the mathematical underpinnings and assumptions that are used in the analysis of station keeping elliptical-inclined constellations. This includes such topics as coordinate system definitions, Lyapunov stability, the derivation of the variation of parameter method for determine the orbital element rates.

Afterwards, the actual algorithm to be implemented is developed, as well as supporting algorithms. This includes a numerical function minimization technique as well as an expansion of the Lyapunov control theory into a workable controller. This is in turn is followed by some discussion of the implementation of the Lyapunov control code into the Goddard Trajectory Determination System (GTDS), as well as the efforts that were required to utilize the GTDS code (compiler issues, etc.).

The low thrust control technique that was discussed in the previous sections is then executed on a GEO satellite test case, and the results are compared to analytically predicted values of station keeping such a satellite. After the algorithm is validated, it is then applied to the Sirius Satellite Radio constellation and John Draim's Teardrop orbit constellation. The resulting station keeping ΔV requirements are then discussed along with some barriers to practical implementation of low-thrust propulsion on elliptical-inclined constellations. The major points of the thesis are then recapped in the conclusion section, followed by some discussion of the future work to better understand and optimize the Lyapunov feedback control technique in this application.

2.0 Math/Theory

2.1 Coordinate Systems

J2000 & B1950 Inertial Frames

Every coordinate system is defined by a fundamental plane and a fundamental direction. The J2000 coordinate system is an inertial frame defined by the plane of Earth's rotation around the sun (the ecliptic), and the direction is defined by the intersection of Earth's equatorial plane and the ecliptic plane. Since the ecliptic plane will vary, as well as Earth's equatorial rotation, the coordinate system is defined at a certain time (epoch)[11]. The epoch for J2000 is defined on January 1st, 2000 at 11:58:55.816 Greenwich Meridian Time (GMT, or sometimes UTC), or January 1st, 2000, 12:00:0.0 Terrestrial Time (TT). The difference between TT and GMT being an offset of 64.184 seconds[12].

It should be noted that the program inputs to the Goddard Trajectory Determination System (GTDS) can be done in both J2000 and B1950. However, all program inputs in this thesis are stated in the B1950 coordinate system. Thus, a specification of a satellite's initial Keplerian elements are actually in B1950. This does not affect the accuracy of our results, since all calculations are calculated in the same B1950 reference frame.

RSW Satellite Fixed Non-Inertial Coordinates

The RSW coordinate system is defined in Vallado[13]. It is a satellite fixed coordinate system, where the R direction is the vector from Earth's center of mass to the satellite, the W direction is the direction of the angular momentum vector (R x V), and the S axis points in the direction of (but not necessarily parallel to) the velocity vector and is perpendicular to the R direction.

2.2 Orbital Elements

Keplerian Elements

The standard set of osculating Keplerian elements are used to describe a satellite's orbital state. The following table lists each element, its symbol, and its definition. All elements except for true anomaly are referred to as "slow elements". In an ideal 2-body problem these elements will not vary at all. But, real world orbits typically undergo disturbing perturbations that will cause these elements to vary slowly with respect to the true anomaly, hence they are called slow elements. Likewise, the true anomaly is sometimes called the "fast element" or "fast variable".

Table 2: The Keplerian Elements

Element	**Symbol**	**Definition**
Semi-Major Axis	a	Distance along of the long axis from center of ellipse to the end (apoapsis/peripapsis)
Eccentricity	e	Ratio of distance from center of ellipse to focus and semi-major axis.
Inclination	i	Angle between z-axis of coordinate system and angular velocity momentum vector of orbit
Right Ascension of the Ascending Node	Ω	Angle from primary direction (x direction) of coordinate system to the ascending side of the line of nodes in the coordinate system primary plane.
Argument of Perigee	ω	Angle from line of nodes to perigee in orbit plane in direction of orbital motion.
True Anomaly	υ	Angle from perigee to satellite in orbit plane in direction of orbital motion.

The following diagram shows the definition of the Keplerian orbital elements.

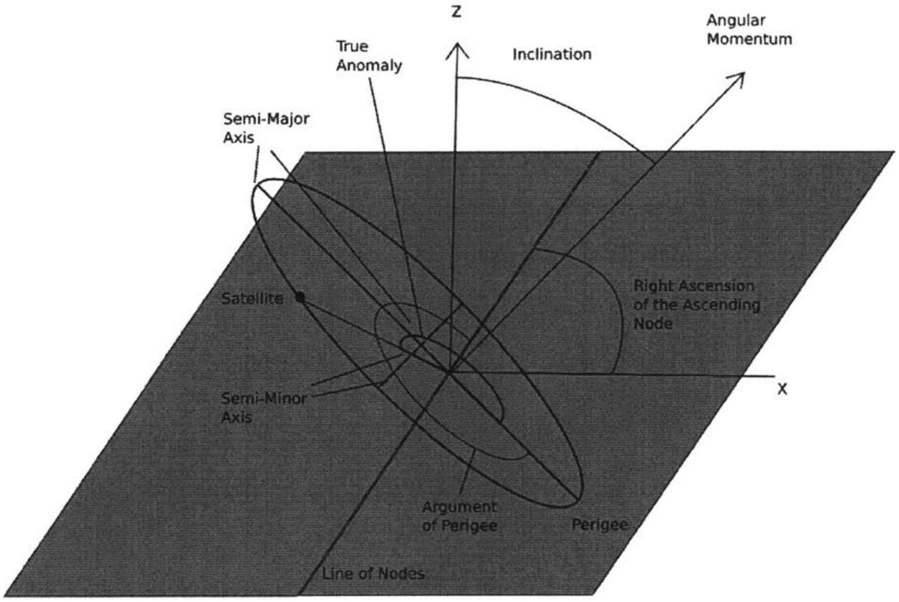

Figure 1: Definition of Keplerian Elements

Keplerian elements have some drawbacks due to the existence of singularities for orbits that are in the primary plane of the coordinate system (sometimes referred to as equatorial orbits) or that are circular. In the equatorial case, the right ascension of the ascending node becomes undefined. In the circular case, the argument of perigee and true anomaly are undefined. It is interesting to note that a geostationary satellite is affected by both of these singularities as it is both equatorial and circular.

Equinoctial Elements

Equinoctial elements are formulated to deal with the deficiencies of the Keplerian element set. Namely, equinoctial elements do not suffer from most of the singularities present in Keplerian elements. The only case where these elements are undefined is when an orbit is completely retrograde ($i = 180°$). There are ways of compensating for this case, so it should also be noted that this is by no means the only way to represent an orbit state in a nonsingular fashion. For example, the orbit state can be formulated in terms of quaternions[14].

The equinoctial elements used in this thesis are composed of the Keplerian elements in the following arrangement.

Eq. 1

$$
\begin{aligned}
&a = a \\
&h = e\sin(\omega + \Omega) \\
&k = e\cos(\omega + \Omega) \\
&p = \tan(\tfrac{i}{2})\sin(\Omega) \\
&q = \tan(\tfrac{i}{2})\cos(\Omega) \\
&L = \nu + \omega + \Omega
\end{aligned}
$$

The last element, L, is sometimes called "true longitude". It is also the fast variable. While the true longitude is the element that will be used as the fast variable in the rate equations throughout this document, the derivation of the Gaussian rate equations in the following section assumes the mean longitude is used as the fast variable. The following equation gives the definition of the mean longitude.

$$\lambda_M = M + \omega + \Omega$$

2.3 Gaussian Rate Equations

Variation of Parameters

The general variation of parameters method applied to the two body problem seeks to take the general equations of motion and allow the inclusion of perturbations in the previously static system. The end goal is to come up with a set of first order differential equations that describe the time rate of change of the orbital elements as a function of the perturbing (i.e. forces that are not due to the primary "two-body" force) acceleration:

$$\frac{d\vec{\chi}}{dt} = f(\vec{\chi},t)$$

It should be noted that each orbital element χ is independent, which allows us to make some assumptions when taking the derivative of $f(\vec{\chi},t)$ with respect to time. Also, the derivation utilizes the mean longitude

First, we can express the satellite state (position and velocity) as a function of the orbital elements and time.

Eq. 2

$$\vec{r} = \vec{r}(\chi_1,\chi_2,\chi_3,\chi_4,\chi_5,\chi_6,t) = \vec{r}(\vec{\chi},t)$$

$$\vec{v} = \dot{\vec{r}}(\chi_1,\chi_2,\chi_3,\chi_4,\chi_5,\chi_6,t) = \dot{\vec{r}}(\vec{\chi},t)$$

In the unperturbed equations of motion, all the orbital elements are constant, so the velocity can be expressed as follows:

Eq. 3

$$\frac{d\vec{r}}{dt} = \frac{\partial\vec{r}(\vec{\chi},t)}{\partial t}$$

However, in the perturbed system the orbital elements are function of time, which leads to this expression for the satellite velocity:

Eq. 4

$$\frac{d\vec{r}}{dt} = \frac{\partial \vec{r}(\vec{\chi},t)}{\partial t} + \sum_{i=1}^{6} \frac{\partial \vec{r}(\vec{\chi},t)}{\partial \chi_i} \frac{d\chi_i}{dt}$$

Additionally, we can impose the following constraint to solve the original equations of motion:

Eq. 5

$$\sum_{i=1}^{6} \frac{\partial \vec{r}(\vec{\chi},t)}{\partial \chi_i} \frac{d\chi_i}{dt} \equiv 0$$

The above constraint is not the only constraint that allows us to solve the original equations of motion, but it is the only constraint that allows both position and velocity to be related to these perturbed elements through the formulas of elliptic motion[15].

Differentiating again yields an expression for the acceleration.

Eq. 6

$$\ddot{\vec{r}} = \frac{\partial^2 \vec{r}(\vec{\chi},t)}{\partial t} + \sum_{i=1}^{6} \frac{\partial \dot{\vec{r}}(\vec{\chi},t)}{\partial \chi_i} \frac{d\chi_i}{dt}$$

This equation can then be substituted into the two-body equations of motion give us a useful result. The following is the equation of motion for an unperturbed two body problem.

Eq. 7

$$\ddot{\vec{r}} + \frac{\mu \vec{r}(\vec{\chi},t)}{\left|\vec{r}(\vec{\chi},t)\right|^3} = 0$$

But for a perturbed system, we use this equation of motion.

Eq. 8

$$\ddot{\vec{r}} + \frac{\mu\vec{r}(\vec{\chi},t)}{\left|\vec{r}(\vec{\chi},t)\right|^3} = \vec{a}_{perturbation}$$

Combining equation Eq. 6 with Eq. 8:

Eq. 9

$$\frac{\partial^2\vec{r}(\vec{\chi},t)}{\partial t} + \sum_{i=1}^{6}\frac{\partial\dot{\vec{r}}(\vec{\chi},t)}{\partial\chi_i}\frac{d\chi_i}{dt} + \frac{\mu\vec{r}(\vec{\chi},t)}{\left|\vec{r}(\vec{\chi},t)\right|^3} = \vec{a}_{perturbation}$$

This equation is a combination of the original unperturbed equations of motion and other terms, and the original unperturbed equations of motion evaluate to zero.

Eq. 10

$$\sum_{i=1}^{6}\frac{\partial\dot{\vec{r}}(\vec{\chi},t)}{\partial\chi_i}\frac{d\chi_i}{dt} = \vec{a}_{perturbation}$$

However, the element rates in this form are not too useful as they are and require some more development. One possible way of developing these equations is the Gaussian variation of parameters formulation.

Gaussian Variation of Parameters

Gaussian variation of parameters starts by taking the product of Eq. 5 and Eq. 10 with the derivative of the orbital elements with respect to position and velocity, respectively.

$$\sum_{i=1}^{6}\frac{\partial\chi_j}{\partial\vec{r}}\cdot\frac{\partial\vec{r}}{\partial\chi_i}\frac{d\chi_i}{dt}=0$$

$$\sum_{i=1}^{6}\frac{\partial\chi_j}{\partial\dot{\vec{r}}}\cdot\frac{\partial\dot{\vec{r}}}{\partial\chi_i}\frac{d\chi_i}{dt}=\frac{\partial\chi_j}{\partial\dot{\vec{r}}}\cdot\vec{a}_{perturbation}$$

These two equations are then added to each other.

$$\sum_{i=1}^{6}\left(\frac{\partial\chi_j}{\partial\vec{r}}\cdot\frac{\partial\vec{r}}{\partial\chi_i}+\frac{\partial\chi_j}{\partial\dot{\vec{r}}}\cdot\frac{\partial\dot{\vec{r}}}{\partial\chi_i}\right)\frac{d\chi_i}{dt}$$

The term within the parentheses just evaluates to the Kronecker delta function.

Eq. 11

$$\sum_{i=1}^{6}\delta_{ij}\frac{d\chi_i}{dt}=\frac{\partial\chi_j}{\partial\dot{\vec{r}}}\cdot\vec{a}_{perturbation}$$

The resulting partial derivatives of the orbital elements with respect to velocity yield the Gaussian variation of parameter equations.

The derivations of these equations (Eq. 12 through Eq. 16) is difficult and lengthy. One way of accomplishing this is to take the derivative of the Keplerian elements with respect to velocity (right hand side Eq. 11) explicitly in the desired coordinate frame. Typically the most useful frame is the non-inertial RSW coordinate frame. This yields the Gaussian rate equations in Keplerian elements. These rates can then be formulated in equinoctial elements by differentiating Eq. 1 with respect to time, and then substituting the Keplerian element rates. After much algebraic manipulation, one has the desired results: Gaussian rate equations in terms of accelerations in the RSW coordinate system[16].

Eq. 12

$$\frac{da}{dt}=2\sqrt{\frac{a^3}{\mu\left(1-h^2-k^2\right)}}\left(\left[k\sin L-h\cos L\right]F_R+\left[1+k\cos L+h\sin L\right]F_S\right)$$

Eq. 13

$$\frac{dh}{dt}=\sqrt{\frac{a}{\mu}\left(1-h^2-k^2\right)}\left(\left[-\cos L\right]F_R+\left[\frac{h+\sin L(2+k\cos L+h\sin L)}{1+k\cos L+h\sin L}\right]F_S+\left[\frac{k(q\sin L-p\cos L)}{1+k\cos L+h\sin L}\right]F_W\right)$$

Eq. 14

$$\frac{dk}{dt}=\sqrt{\frac{a}{\mu}\left(1-h^2-k^2\right)}\left(\left[\sin L\right]F_R+\left[\frac{k+\cos L(2+k\cos L+h\sin L)}{1+k\cos L+h\sin L}\right]F_S+\left[\frac{h(q\sin L-p\cos L)}{1+k\cos L+h\sin L}\right]F_W\right)$$

Eq. 15

$$\frac{dp}{dt}=\frac{1}{2}\sqrt{\frac{a}{\mu}\left(1-h^2-k^2\right)}\left[\frac{\left(1+p^2+q^2\right)\sin L}{1+k\cos L+h\sin L}\right]F_W$$

Eq. 16

$$\frac{dq}{dt}=\frac{1}{2}\sqrt{\frac{a}{\mu}\left(1-h^2-k^2\right)}\left[\frac{\left(1+p^2+q^2\right)\cos L}{1+k\cos L+h\sin L}\right]F_W$$

These rate equations are effective except for the case where the inclination is 180 degrees or the orbit is parabolic (e = 1). The disturbing acceleration (F) is in RSW coordinates which were defined earlier. The utility of these equations will be seen in the formulation of the Lyapunov feedback control algorithm.

2.4 Lyapunov Functions & Stability

Lyapunov feedback control presents a relatively easy and direct method for controlling nonlinear systems. It is easy because it requires no calculation of co-states, and if the proper selection of Lyapunov function has been accomplished, convergence is guaranteed, regardless of starting conditions or ending conditions[17]. The main difficulty of this method is determining an appropriate Lyapunov function for the system one wishes to control. There is no direct method to generate a Lyapunov function for an arbitrary nonlinear system. Sometimes, though, energy is a good place to start.

A Lyapunov function itself is a scalar function that generally measures the state of convergence of any particular nonlinear or linear system. A Lyapunov function also has some very important properties that will be useful for developing a control law.

Let us introduce the Lyapunov function.

Given the system:

$$\dot{\vec{x}} = f(\vec{x},t)$$

$$\vec{x} \in D$$

If there exists a function $V(\vec{x})$ that is continuous in the first derivatives (C_1) in D, and it has the following properties

Eq. 17

$V(0)=0$

Eq. 18

$V(\vec{x})>0 \ \ for \ \ \vec{x}\neq 0 \in D$

then that function may be called a *Lyapunov function*. What this means is that if one has a function of the state variables $\vec{x}$ that is always positive except for when the state variables are zero, then that function is a Lyapunov function.

Additionally, if

Eq. 19

$\nabla V(\vec{x}) \ \cdot \ f(\vec{x},t)\leq 0$

and this condition hold true for all of $\vec{x}$ in the domain D, then one has *Lyapunov stability*—$\vec{x}$ will not leave D if it started in D[18]. If the left hand side of Eq. 19 is always less than zero then the system state will asymptotically approach zero. That is why the zero state is sometimes called “the attractor”.

The utility of the Lyapunov function will be discussed in the next section.

2.5 Lyapunov Feedback Control

Nowhere during the description of the Lyapunov function is a control input mentioned. However, if one can find a Lyapunov function for a given system where the function is minimized at the target state, i.e.

$V(\vec{x}-\vec{x}_T)=0$

then it is possible to construct a feedback control system based on that function. The goal of this controller is to drive the Lyapunov function to zero, since zero is formulated to be the target state of the system.

The utility of all this is that even if the natural dynamics of the system may not have Lyapunov stability, it is possible to enforce Lyapunov stability by active control of the system. Thus, the goal is to make the left hand side of Eq. 19 as negative as possible by actively controlling the system. The more negative it is, the quicker it will converge to the target state.

3.0 Algorithms

3.1 Q-Law Lyapunov Feedback Control

The name “Q-law” comes from the formulation by Petropoulos[19] of a Lyapunov function to control a satellite’s orbital elements with low thrust propulsion. The Lyapunov function he chose to use was called “Q”, hence “Q-law”. Here “G” is used instead of “Q” since there is an equinoctial element also labeled “Q”. Please note that the Lyapunov controller is only for control of the slow orbital elements. Control of fast variables such as mean anomaly or mean longitude is possible with the addition of another control loop[20], as will be seen below.

The following Lyapunov function is introduced:

Eq. 20

$$G(t) = \sum_{i=1}^{5} W_i \left(\frac{\chi_i^C(t) - \chi_i^T}{Max\left|\dot{\chi}_i^C(t)\right|} \right)^2$$

W_i = Control weight

χ_i^C = Current slow element

χ_i^T = Desired slow element

$Max\left|\dot{\chi}_i^C\right|$ = Absolute maximum time rate of change of element over current orbit

$$\chi_1^C = a = a$$
$$\chi_2^C = h = e\sin(\omega + \Omega)$$
$$\chi_3^C = k = e\cos(\omega + \Omega)$$
$$\chi_4^C = p = \tan(\tfrac{i}{2})\sin(\Omega)$$
$$\chi_5^C = q = \tan(\tfrac{i}{2})\cos(\Omega)$$

Please also note that this function is not globally convergent, only locally convergent. Specifically, it is unstable for large changes in inclination. Since it is easier to make changes in inclination if the semi-major axis is greater, the control law will cause the semi-major axis to increase without bounds. This restriction is okay since we are interested in small element change transfers (orbit maintenance). A globally convergent function is given by Petropoulos[21]. This function results in more complex partial derivatives of the Lyapunov function but does not have problems converging for large element change transfers.

Differentiating G and using the chain rule results in the following expression.

Eq. 21

$$\frac{dG}{dt} = \sum_{i=1}^{5} \frac{\partial G}{\partial \chi_i^C} \dot{\chi}_i^C(t) = \sum_{i=1}^{5} \left[\frac{2W_i\left(\chi_i^C - \chi_i^T\right)}{Max\left|\dot{\chi}_i^C(t)\right|^2} \right] \dot{\chi}_i^C(t)$$

Assuming the spacecraft orbital state is controlled via low-thrust propulsion (the acceleration is a perturbation), then the element rates $\dot{\chi}_i^C(t)$ can be expressed using the Gaussian variation of parameters.

Eq. 22

$$\dot{\chi}^C(t,\phi,\beta)=\begin{bmatrix}\dot{a}_1 & \dot{a}_2 & 0\\ \dot{h}_1 & \dot{h}_2 & \dot{h}_3\\ \dot{k}_1 & \dot{k}_2 & \dot{k}_3\\ 0 & 0 & \dot{p}\\ 0 & 0 & \dot{q}\end{bmatrix}\begin{bmatrix}F_R\\ F_S\\ F_W\end{bmatrix} \qquad \begin{aligned}F_R &= \tfrac{|T|}{m_{s/c}}\sin\phi\cos\beta\\ F_S &= \tfrac{|T|}{m_{s/c}}\cos\phi\cos\beta\\ F_W &= \tfrac{|T|}{m_{s/c}}\sin\beta\end{aligned}$$

Eq. 22 is just Eq. 12 through Eq. 16 represented in matrix form. Since we will referring the matrix given by Eq. 22 a number of times, it will be represented in a more compact form:

Eq. 23

$$\underset{[5\times 1]}{\dot{\chi}^C(t,\phi,\beta)}=\underset{[5\times 3]}{[D\chi(t)]}\underset{[3\times 1]}{\vec{F}(\phi,\beta)}$$

F = acceleration vector in RSW coordinate system

R = radial unit vector

W = unit angular momentum vector

S = W x R

T = thrust

$m_{S/C}$ = spacecraft mass

$\phi\in[0,2\pi]$ = Yaw Angle, $\beta\in\left[-\pi/2,\pi/2\right]$ =Pitch Angle

The matrix [Dχ(t)] is a function of time simply because the fast variable L is a function of time, and the slow elements are assumed to vary only slightly. This assumption can be made because our perturbing acceleration is small.

Defining the partial derivative matrix [DG] with respect to the Lyapunov function:

Eq. 24

$$\frac{dG}{dt} = \sum_{i=1}^{5} \frac{\partial G}{\partial \chi_i^C} \dot{\chi}_i^C(t) = [DG] \dot{\chi}_i^C(t)$$

$$[DG] = \begin{bmatrix} \frac{\partial G}{\partial \chi_1} & \frac{\partial G}{\partial \chi_2} & \frac{\partial G}{\partial \chi_3} & \frac{\partial G}{\partial \chi_4} & \frac{\partial G}{\partial \chi_5} \end{bmatrix}$$

$$[1 \times 5]$$

Combining Eq. 24 with Eq. 23:

Eq. 25

$$\frac{dG}{dt} = [DG][D\chi]\vec{F}(\phi,\beta)$$

Again, the purpose of all this is to minimize (make the most negative) the above equation at any given instant of time by varying the pitch and yaw angles.

3.2 Minimizing Q-law Function

Calculating matrix [DG] and matrix [Dχ] and multiplying by the acceleration vector yields the following equation:

Eq. 26

$$\frac{dG}{dt} = \left[k_{G1}(t)\sin\phi\cos\beta + k_{G2}(t)\cos\phi\cos\beta + k_{G3}(t)\sin\beta \right]$$

Eq. 26 is just a reformulation of Eq. 25, but it allows us to see that at any given point of time there is a function that relates the Lyapunov function rate to the pitch and yaw angles with just 3 constants. It should now be possible to find the most negative Lyapunov rate by just taking the gradient of Eq. 26 with respect to ϕ and β and setting it equal to zero. This will yield two equations and two unknowns.

Eq. 27

$$\nabla\left(\frac{dG}{dt}\right) = \left\langle \frac{\partial}{\partial\phi}\hat{i}_x, \frac{\partial}{\partial\beta}\hat{i}_y \right\rangle \frac{dG}{dt} = (0,0)$$

Both are solvable and yield trigonometric solutions (multiple values of ϕ and β) of the inflection points of the Lyapunov rate function. Each inflection point will be a possible solution. The equations are differentiated again to find the second derivative of the original equations. Each possible solution is tested to find a solution that minimizes Eq. 26—i.e. both second derivatives are positive at the solution values. This solution is the direction to point the spacecraft's thrusters to drive the Lyapunov function to zero.

Keep in mind these angles are in the RSW coordinate system and need to be transformed into the inertial frame.

3.3 Maximizing Gaussian Rates

The maximum element rates (as seen in the original definition of the G function) have been neatly rolled up into [DG], but they still must be accounted for. Discussion of the merits of using the maximum element rates in the Lyapunov function can be found in the work of Petropoulos. In a nutshell, it normalizes the control weights and allows large element change transfers to be more optimal.

In general, each element rate due to thrusters has this form:

Eq. 28

$$\dot{\chi}_i^{C} = \tfrac{T}{m_{S/C}}\left[c_1(L)\sin\phi\cos\beta + c_2(L)\cos\phi\cos\beta + c_3(L)\sin\beta\right]$$

The "c" coefficients come from rate equations given previously in Eq. 22. For example, the semi-major axis would have the following coefficients:

$$c_1 = \dot{a}_1$$
$$c_2 = \dot{a}_2$$
$$c_3 = 0$$

By varying the thruster pointing angles ϕ and β, Eq. 28 makes a three-dimensional ellipsoid with the principle radii having lengths proportional to the value of the c coefficients. Therefore, the maximum element rate should simply be the coefficient with the greatest absolute value. The problem now becomes finding the extrema of the coefficients as a function of L, the fast variable.

The maximum rates for the elements *a*, *p*, and *q* can be solved for analytically by simply taking the derivative with respect to L of the functions, and setting that equal to zero. The functions for *h* and *k*, on the other hand, appear to not have any analytical solutions to the maximization problem. Instead, we use a numerical method called the *golden section method*, to find the extrema of the *h* and *k* rates. The golden section method will be discussed in the following section. Meanwhile, the resulting maximum element rates that are determined analytically are given below.

Calculation of Maximum Rate of Semi-Major Axis

Eq. 29

$$(\dot{a}_1(L))_{\max} \rightarrow \tan(L) = -\frac{k}{h}$$

$$(\dot{a}_2(L))_{\max} \rightarrow \tan(L) = -\frac{h}{k}$$

if $h = 0$ and $k = 0$ then L = 0 (or pick any L)

$$\dot{a}_{\max} = \max\{|\dot{a}_1|, |\dot{a}_2|\}$$

Calculation of Maximum Rates of p & q

Eq. 30

$$(\dot{p}(L))_{\max} \rightarrow \cos(L) = -k$$

$$(\dot{q}(L))_{\max} \rightarrow \sin(L) = -h$$

Note that the inverting the trigonometric functions will produce two values for L on the domain from 0 to 2π. Each value of L must be tested for.

Method for Finding Maximum Rates of h & k

The golden search method returns the minimum value of a function within a certain bounds. Thus, bounding the minima is very important to this search method, and it helps to know a little bit about the function before one attempts to minimize it. We know that the rate equations (Eq. 13 and Eq. 14) are periodic in nature so the search algorithm should be initially bounded between 0 and 2π. The algorithm will then converge to a minimum. However, there will sometimes be a second minimum that is located 180° around from the first, and that must also be found. Then, each minimum will be compared in magnitude, and the one with the highest value is then the maximum element rate. The procedure below outlines how to find the maximum of element "h". This procedure is identical for finding element "k".

The first possible maximum for both h and k are proportional to either the sine of cosine of L multiplied by the constant in front. Between 0 and 2π, the maximum of sine or cosine will just be unity.

Eq. 31

$$\left|\dot{h}_1\right|_{\max} = \sqrt{\frac{a}{\mu}\left(1 - h^2 - k^2\right)}$$

Dealing with the second and third coefficients from Eq. 13 and Eq. 14 requires the use of a numerical search method. Since the golden search method searches for minimums, the search operates on the negative absolute value of the rate equations. The search method will converge on one of the two minimums if there is one, otherwise it stops when the it exceeds the maximum allowable iterations.

$$\begin{aligned}\left[\left(\dot{h}_2\right)_{\max 1}, L_{\min}\right] &= golden\left(-\left|\dot{h}_2(L)\right|\right) \\ \left[\left(\dot{h}_3\right)_{\max 1}, L_{\min}\right] &= golden\left(-\left|\dot{h}_3(L)\right|\right)\end{aligned} \qquad L \in \left[0, 2\pi\right)$$

The search is then run centered 180° around from the first minimum.

$$\begin{aligned}\left[\left(\dot{h}_2\right)_{\max 2}, L_{\min}\right] &= golden\left(-\left|\dot{h}_2(L)\right|\right) \\ \left[\left(\dot{h}_3\right)_{\max 2}, L_{\min}\right] &= golden\left(-\left|\dot{h}_3(L)\right|\right)\end{aligned} \qquad L \in \left[L_{\min} + \frac{\pi}{2}, L_{\min} + \frac{3\pi}{2}\right]$$

$$\left|\dot{h}\right|_{\max} = \max\left\{\left|\dot{h}_1\right|_{\max}, \left|\dot{h}_2\right|_{\max 1}, \left|\dot{h}_2\right|_{\max 2}, \left|\dot{h}_3\right|_{\max 1}, \left|\dot{h}_3\right|_{\max 2}\right\}$$

3.4 Golden Section Minimization Algorithm

The golden section search method[22] is a commonly used numerical tool for finding local minima. Assuming the minimum is bounded between two points on the domain of the function, then the algorithm will converge.

The algorithm works by evaluating the values of the boundary points and two sample points located between the two boundary points. The two interior points are compared, and the higher value sample point becomes the new bound on that side.

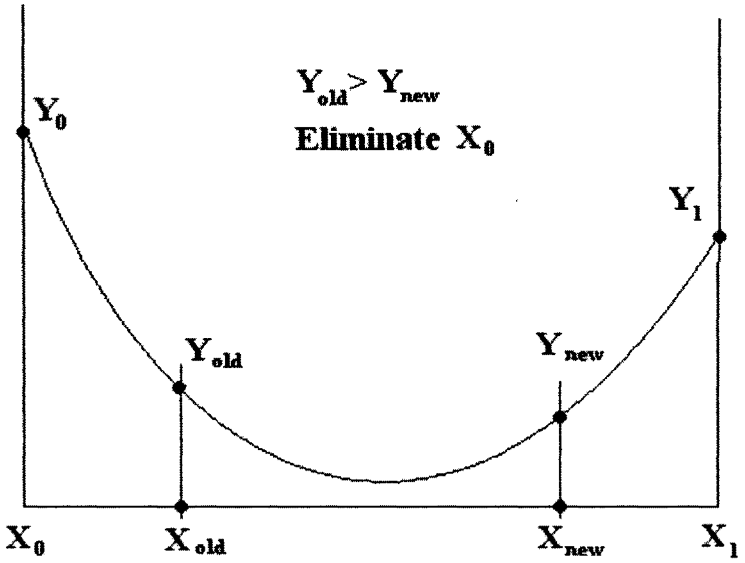

Figure 2: Golden Search Initial Step, Step 1

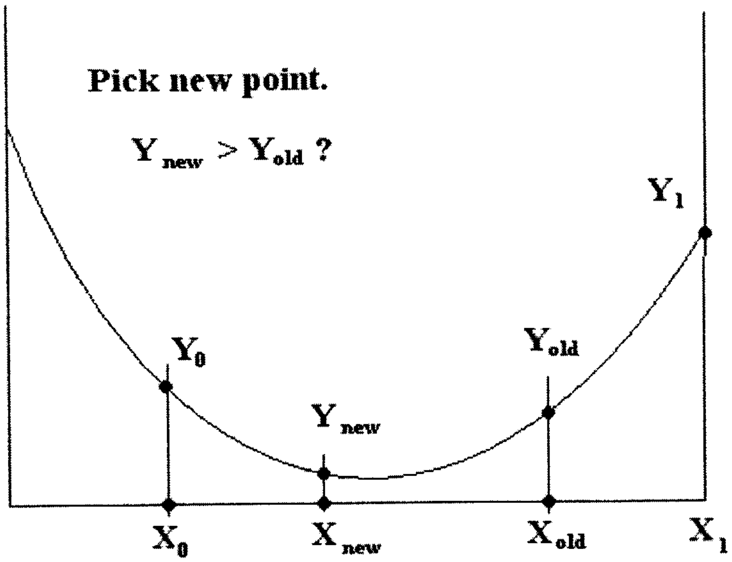

Figure 3: Golden Search Method, Step 2

If the minimum is located between the original boundary points, then the boundary points will be seen to move downhill towards to minimum. When the distance between the boundary points come within a certain tolerance, the algorithm has converged.

3.5 Fast Variable Control with Classical Controller

Introduction

The Q-law Lyapunov feedback control does a very good job controlling the slow orbital elements, but the current formulation does not allow for control of the fast variable. While an orbit's slow elements will determine the shape of the ground track that the satellite makes on Earth's surface, the fast variable will determine a satellite's position along the ground track. Ground tracks are the locus of points formed by the points on the

Earth directly below a satellite as it travels through its orbit[23]. Repeat ground tracks are orbits which retrace their ground track over a certain time interval[24]. For example, if a satellite has a period of about 8 hours, the ratio of its period to Earth's rotational period is 3:1. In other words, every three orbits the satellite will be over the same spot on Earth. For repeat ground track orbits, the satellite's position on the ground track at a given point in time is probably more important than maintaining the ground track shape. Therefore, we will need to implement another control loop to control the fast variable, L.

Control of a satellite's position along the ground track is simple. If the satellite's period is perturbed to be shorter than its original ideal period, the whole ground track will be observed to drift eastwards. The satellite takes longer to complete its orbit, thus the Earth will have rotated more (to the east) in the meantime. If the orbit's period is perturbed to greater than the initial ideal period, then the ground track will drift westwards. Control of the satellite's period is accomplished by controlling its semi-major axis. The end result of this is that there is now an outer loop formed around the original Lyapunov controller. The outer loop feeds in values of target semi-major axis for the Lyapunov controller to control towards. The following block diagram shows the arrangement of control loops.

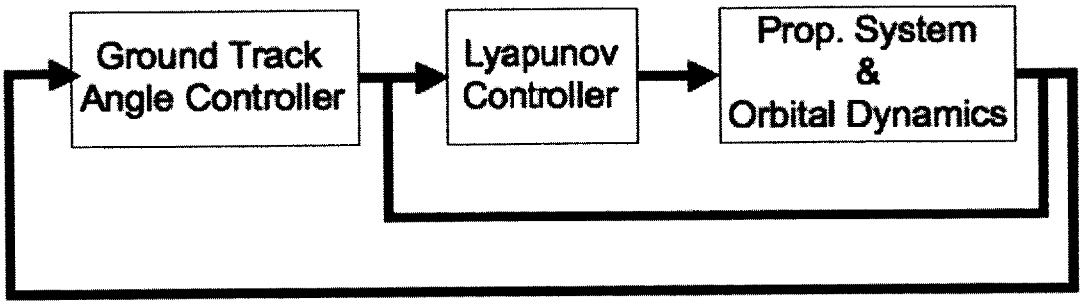
Ground Track Angle Controller
Lyapunov Controller
Prop. System & Orbital Dynamics

Figure 4: Control System Block Diagram

Ground track east/west location is also determined by the slow element Ω. This makes sense because changing the orbit's line of nodes will determine the east/west placement of the ground track. Due to Earth's oblateness (sometimes called J_2 from the coefficient in the Lagrange planetary equations), the line of nodes will be observed to drift at a constant rate. This can be greater than 10° per day[25] for satellites in low Earth orbit. Thus, the fast variable controller should take into account Ω.

Fast Variable Formulation

First, a new formulation of the fast variable is introduced: the ground track angle (λ). This angle is most easily described as the east/west placement of the ground track on Earth's surface. An orbit's slow elements will determine the shape of the ground track in the inertial frame, but the ground track angle will determine where the ground track is in relation to an object that is static in Earth's rotating frame.

The goal of this controller is to hold λ at constant while Ω drifts freely due to J_2 effects. Constant λ means the satellite's ground track always crosses Earth's equator at the same places. If the satellite starts on station without error, then keeping the ground track angle at its original value at t = 0 will maintain the repeat ground track.

The ground track angle is defined as the following:

Eq. 32

$$\lambda(t) = \Omega(t) + \int_0^{t_c} \dot{\psi}\, dt$$

Eq. 33

$$\dot{\psi} = \frac{2\pi}{P_{Earth}}\left(1 - \frac{2\pi\ k_{mult}}{P_{Earth}}\sqrt{\frac{a^3}{\mu}}\right)$$

k_{mult} = Desired ratio of satellite period to Earth's rotation period

P_{Earth} = Sidereal rotational period of Earth (86,164.09 seconds)

$\dot{\psi}$ = Angular drift rate due to satellite period error

a = Semi-major axis

μ = Gravitational parameter

Control of the ground track angle can be implemented by varying the semi-major axis to control $\dot{\psi}$.

Ω Drift Compensation

Unfortunately, the Lyapunov controller will still attempt to correct for the growing error in Ω. Therefore, we must allow for Ω "to slide" by varying the target equinoctial elements. In the equinoctial element set we are using, every element except the semi-major axis depends on Ω. Determining what these target elements will be is simply accomplished via the definition of the equinoctial element set.

Eq. 34

$$\begin{aligned}
\chi_2^T &= e\sin(\omega + \Omega(t)) \\
\chi_3^T &= e\cos(\omega + \Omega(t)) \\
\chi_4^T &= \tan(\tfrac{i}{2})\sin(\Omega(t)) \\
\chi_5^T &= \tan(\tfrac{i}{2})\cos(\Omega(t))
\end{aligned}$$

If each of the target Keplerian elements is held constant but Ω is allowed to drift, the last four target elements can be calculated. This leaves only the determination of the target semi-major axis to control $\dot{\psi}$.

Classical Control of Fast Variable

The dynamics of the ground track angle are such that in the absence of perturbations and with a constant value of Ω it will act much like an object in free space—an object on motion will tend to stay in motion. Thus, if one perturbs the semi-major axis slightly, the ground track angle will drift at a constant rate. In control terminology, this means the plant function has a single a pole at zero. Since this is a type one system, steady state error would be zero in the absence disturbances with just a proportional controller[26]. However, there is a constant disturbance since Ω drifts at a more or less constant rate, so steady state error would not be zero. To compensate for this, one may implement a proportional-integral controller[27] (PI). Additionally, a Ω drift compensation term is included since we can have *a priori* knowledge of the Ω drift rate via Lagrange's planetary equations. This avoids large initial overshoot due to the integral term "windup" in the face of the Ω drift disturbance.

The following equations shows the form of the PI controller:

Eq. 35

$$u(t_c) = K_P E(t_c) + K_I \int_0^{t_c} E(t)dt$$

The PI controller is applied to ground track angle control:

Eq. 36

$$\chi_1^T \leftarrow \dot{\psi} = K_P \lambda_{error} + K_I \int_0^{t'} \lambda_{error} dt - \dot{\Omega}_{estimate}$$

The following control gains were determined by trial and error. They appear to yield satisfactory results, although better gains and better control methods probably do exist.

$$K_P = 0.5 \cdot \left(T_{Wait}\right)^{-1}$$
$$K_I = 0.1 \cdot \left(T_{Wait}\right)^{-2}$$

T_{Wait} is the time in between station keeping events (7 days) in seconds.

Also, it was noticed that less ΔV was consumed if before each thrust event the target semi-major axis is held constant and held that way for the duration of the burn, as opposed to allowing the fast variable controller to modify the value of the target semi-major axis while thrusting.

3.6 Practical Considerations of Lyapunov and Fast Variable Controller

Gain Selection of Lyapunov Function

One of the benefits of the Q-law controller is that gain selection (control weights) is very easy. In fact, just setting all control weights to unity will result in satisfactory results for transfers in which one does not care about the fast variable. For transfers where one does care about controlling a fast variable, it is necessary to set the control gain for the semi-major axis considerably higher than the other gains. Setting the semi-major axis gain to ten, while letting the rest of the gains be unity seems to work. Increasing the semi-major axis gain beyond ten only increases ΔV consumption without significant accuracy improvements of the ground track angle. Setting it below ten results in unsatisfactory control of the ground track angle.

Petropoulos has done some work with regards to gain optimization. Optimization was carried out using a genetic algorithm for one particular large-element change transfer. Results indicated some propellant savings are to be had by optimization[28].

Noise Filter

It is also sometimes advantageous to employ a low-pass filter on the current semi-major axis (χ_1^C) state as tesseral effects can induce noise that retards controller performance. For certain elliptical orbits, a swing of more than twenty kilometers of semi-major axis is seen due to tesseral effects. For small element change transfers, this short-period variation of semi-major axis can cause the controller to prematurely trip the convergence threshold, as well as incur higher ΔV costs than station keeping with the low pass filter. This is probably due to the controller working to "fix" the suddenly higher semi-major axis error, thus neglecting the other orbital elements.

The time constant of the low-pass filter used was roughly an hour, or 3600 seconds.

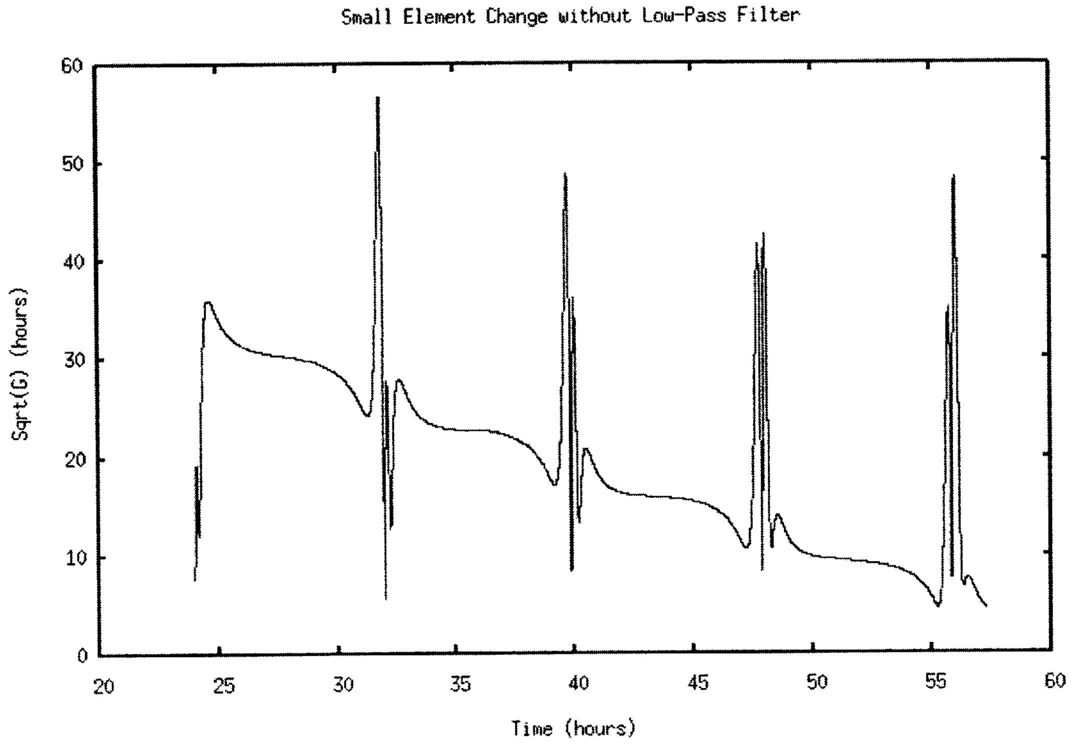

Figure 5: Variation of Lyapunov Function Without Low-Pass Filter

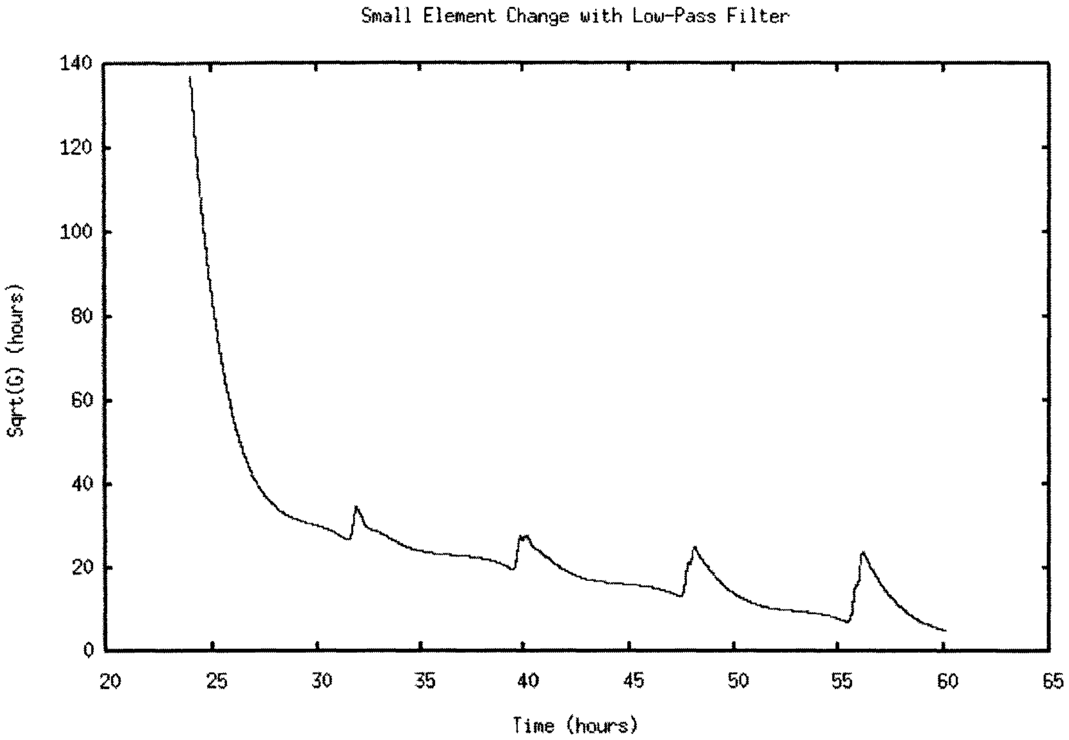

Figure 6: Variation of Lyapunov Function with Low-Pass Filter

These plots were generated using the first satellite in the Teardrop constellation that is defined later in this document.

Defining the Converged State

Another important consideration is when to turn off the thruster. The Lyapunov function given by Eq. 20 converges with the square root of the burn time. The consequence of this is that there are diminishing returns for leaving the thruster on longer. After a certain point, one must ask how many decimals places of accuracy are actually needed? Conveniently enough, there is an easy way to specify a cutoff threshold because the units of the square root of G(t) is seconds.

$$\left[\sqrt{G(t)}\right] = [s]$$

Cutoff threshold can then be specified in units of time. However, one should not confuse this Lyapunov time remaining with actual time it will take to converge since the Lyapunov time is based on maximum element rates and control gains. If one were to increase each of the control gains, then the Lyapunov time would increase accordingly, but convergence rate would remain the same. Also, if one were to raise the control weights but let the cutoff threshold remain the same, then the time from start to cutoff will increase. Accordingly, longer thruster operation will raise ΔV costs. What matters the most is the ratio of initial G(t) to the cutoff G(t). One cannot a priori predict what the initial value of G(t) will be, so some judgment must be exercised in selection of cutoff threshold values.

4.0 Implementation in GTDS

4.1 Introduction

The Goddard Trajectory Determination System (GTDS) is a continuously evolving astrodynamics code written in Fortran, the core of which has been used since the early 1970s. Several previous master's students at MIT have also added various modules to GTDS. While the built-in user interface is primitive (text file inputs) by modern standards, it is possible to access GTDS with a graphical user interface[29]. GTDS has been a widely used as a propagator baseline in the astrodynamics community since its inception. Using freely available the "sloccount" Linux utility, it was determined that the GTDS source code contains roughly 140,000 lines of code (not including comments). Incidentally, sloccount also estimated that GTDS would cost $5,000,000 to develop, requiring 2.2 years time by 18 programmers[30].

The version of the GTDS orbit determination program used in this research represents the integration of the VAX/VMS version and the Unix/PC version. Both of these been under development since the early 1990s. GTDS supports orbit propagation, orbit determination, differential correction, and more. For our purposes, GTDS is extremely useful because it models the perturbative environment in which an Earth-orbiting satellite moves. GTDS accurately models the disturbances due to geopotential harmonics, solid Earth tides, third-body effects, atmospheric drag, solar radiation pressure other radiative forces. Without GTDS we could not accurately determine station keeping delta-V requirements of the elliptical satellite constellations. For these reasons, the Lyapunov feedback control method outlined above was coded in Fortran and added to the ephemeris propagator GTDS.

4.2 Porting to GNU Fortran

Development for this project started from the GTDS capability that had been enhanced by Rick Lyon[31]. For developments previous to 2004, see Rick Lyon's thesis, appendix B.

Since then, a number of things have happened. First, the VAX version of GTDS with 50x50 geopotential model[32] and J2000 integration coordinates[33] was merged with the Unix version of GTDS through the labors of Zachary Folcik at MIT. This updated and merged version of GTDS could be compiled on a machine running Linux and an x86 chip architecture. However, portions of the GTDS source were originally programmed in non-standard Fortran extensions. This caused difficulties in compiling the GTDS source code using the freely available GNU Fortran compiler, as it does not support Fortran extensions. Alternative proprietary compilers, such as the Intel Fortran compiler were not considered due to budget constraints. An added benefit to compatibility with the GNU compiler is that it makes GTDS easy to parallelize in a cost effective manner for cluster processing with paradigms such as MPI.

In order to port the GTDS source code to the GNU compiler, the whole GTDS source code had to be swept clean of offending syntax. This task took approximately 3 months to complete successfully. The following changes were the most commonly encountered:

1) Use of DATA syntax to store values to INTEGER*8. Traditional Fortran only allows DATA to write to variables of type REAL*8. This issue was solved using EQUIVALENCE statement to link a REAL*8 and INTEGER*8 variable. The DATA command then stored to information to the REAL*8, which was linked to the original variable name that is an INTEGER*8.
2) The module included by a previous Master's student, Rick Lyon, had started GTDS on to a path of modernization by using object oriented structures with RECORD statements. However, the GNU compiler does not recognize these, thus the object oriented variables had to be changed to regular variables.
3) Several arrays with DATA statements had shorthand forms for storing successive rows of the same number.

There were numerous other cases, some more tricky than others to solve. Testing of the integrated GTDS program versus the benchmark cases defined by Metzinger is ongoing, but the new Linux version of GTDS has passed some basic benchmarks and appears to run well on 64 bit systems.

Happily, others have already found the Linux version of GTDS to be very useful. Most notably this version was modified and parallelized at the Maui Supercomputing cluster. The work was done by Dr. Matt Wilkins to reduce spacecraft ephemeris data to improve atmospheric density models[34].

4.3 Lyapunov Control Code

After being ported to GNU Fortran, the Lyapunov feedback control technique had to be developed. Prototyping was accomplished in Matlab but had to eventually be coded into Fortran. The Fortran code can be found in appendix A, and is a direct implementation of the equations outlined previously.

The following figure is a function tree of the new GTDS low thrust Lyapunov control code.

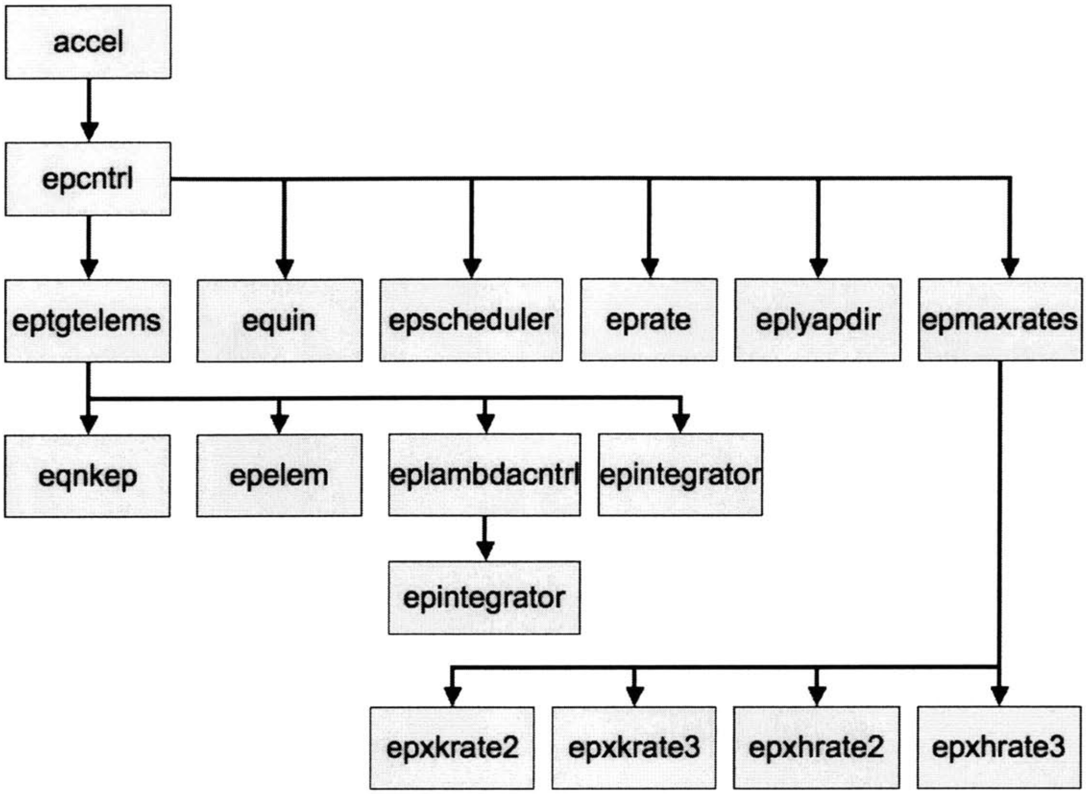

Figure 7: Function Call Tree of Low Thrust Control Code

Out of the entire function tree listed above, only the "accel" subroutine was original to the previous versions of GTDS. However, in order to code an interface to the new Lyapunov control functions, a few other original source files had to be modified as well as one additional new subroutine. The original subroutines that were touched include "setorb", "setog1", "setdaf". The new subroutine that was added is called "eplyapopt", and is called by setog1 to initialize global variables used by new Lyapunov routine.

The program execution is relatively simple. The Lyapunov control code initializes its variables when GTDS reads the EPCNTRL keywords. The code keeps track of the time since the last thrust event ended. If that time exceeds a certain input value, the thruster is turned on (pointing determined by Q-law control algorithm in equinoctial elements) until the square root of the Lyapunov function reaches the specified threshold, whereby the thruster shuts off. This process carries on until GTDS finishes execution by reaching the propagation end time. Currently no output reports are generated by the Lyapunov control code, but instead an ad-hoc output was generated by writing key variables (like orbit state) to external files, after which they were processed and plotted using the freely available and open source *Octave* mathematical software.

4.4 Program Inputs (Keywords)

GTDS is executed using a text input file of rigid formatting. Each input file is made up of keywords in the leftmost column, followed by three columns for integer inputs, and then three columns for floating point inputs. The behavior of the execution, initial satellite state, and so on are all controlled by this file. The following table shows a typical set of keywords that would be used to propagate a spacecraft's ephemeris.

Table 3: Typical GTDS Ephemeris Propagation Input Card

CONTROL	EPHEM					SIRIUS1
EPOCH				1020301.0	0.0	
ELEMENT1	1	2		42164	0.2684	63.4
ELEMENT2				165.0	270.0	345.0
OUTPUT	1	2	1	1020316.0	0.0	86400
ORBTYPE	2	1	1	5		
OGOPT						
POTFIELD	1	6				
OUTOPT	1			1020301200000.	1020301600000.	86400
END						
FIN						

This particular input file would propagate the ephemeris for one of Sirius Satellite Radio's satellites. Propagation would run from March 1st, 2002, to March 16th, 2002. The integrator is GTDS' fixed time step Cowell integrator with a time step of 5 seconds. No thrust options are turned on. The default setting for GTDS has drag effects and solar radiation effects turned off. That is okay for the Sirius case because drag is nonexistent and solar radiation pressure has a much smaller effect on the Sirius satellite than the lunar and solar point masses. Thus, this input file would yield fairly accurate propagation.

For detailed documentation with regards to the controlling keywords, syntax, and much more, it is recommended that one obtain a copy of the GTDS user guide. There is also a GTDS math specification that details all of the algorithms used in GTDS. At the writing of this thesis, the math specification was available in PDF format from the orbits website (orbits.mit.edu), but the GTDS user manuals are only available in paper format. Nevertheless, this does not rule out the existence of electronic versions.

4.5 New Keywords

In order to input control parameters to the Lyapunov control algorithm, it was required to create new control keywords. The following table shows all three new keywords. All three keywords must be present in order for the control algorithm to work.

Table 4: New GTDS Keywords for Low Thrust Control

Keyword	**INTEGER**	**INTEGER**	**INTEGER**	**REAL**	**REAL**	**REAL**
EPCNTRL1	*Weight 1*	*Weight 2*	*Weight 3*	*Target KepElm 1*	*Target KepElm 2*	*Target KepElm 3*
EPCNTRL2	*Weight 4*	*Weight 5*	*Repeat groundtrack*	*Target KepElm 4*	*Target KepElm 5*	*Kmult*
EPCNTRL3	NULL	NULL	NULL	*Accelmag (km/s^2)*	*Lyap Cutoff (seconds)*	*Wait interval (days)*

The description of the new keywords can be found in the following table.

Table 5: Explanation of New GTDS Keywords

Input	**Definition**
Weight 1-5	Control weights defined in Eq. 20

Target Keplerian Element 1-5	Target elements in Keplerian form. These are χ_i^T defined in Eq. 20. These are converted to equinoctial elements before program execution.
Repeat Groundtrack	Specifies if there is of control the groundtrack angle, defined in Eq. 32. The integer "0" is for no, "1" for yes, "2" for repeat groundtrack for equatorial case. This disables line of nodes tracking, but still tracks the time phase component of Eq. 32.
K_{mult}	Specifies the ratio of satellite's orbital period vs. Earth's sidereal rotational period. This number should be a ratio of two integers, and is defined in Eq. 33.
Acceleration magnitude	Sets the constant acceleration magnitude experienced by the spacecraft due to the low-thrust thruster. Units are in km/s^2.
Lyapunov cutoff time	Sets the threshold that determines when the thrust event has finished.
Station keeping wait interval	Spacing between start of station keeping events, in days. Thus, a 7 day wait interval means the spacecraft will have station keeping operations every week.

These keywords are inserted *after* the OGOPT keyword.

5.0 Results

5.1 Procedure

The ultimate goal is to develop realistic station keeping budgets using low thrust propulsion in the GTDS perturbation model for two elliptic-inclined constellations: Sirius Satellite Radio's constellation and John Draim's proposed "Teardrop" orbit constellation.

The general procedure is outlined by the following diagram:

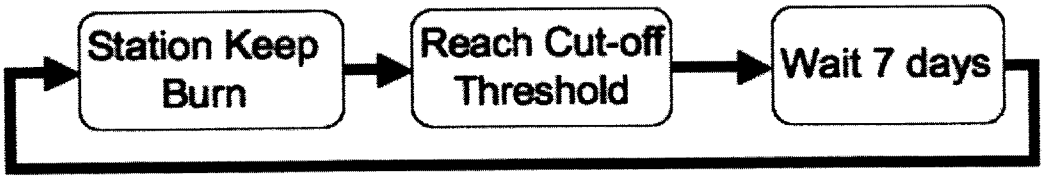

Figure 8: Station Keeping Simulation Program Flow

The spacecraft starts on-station at the beginning of the simulation. Every 7 days afterwards the Lyapunov control module is turned and the spacecraft undergoes a station keeping event. The orbital environment that the spacecraft is in is such that there will always be disturbances in the system that tend to move the spacecraft away from its desired state. Thus the purpose of the station keeping event is to compensate for this drift by moving the spacecraft to its desired state using the spacecraft's thrusters. This is exactly what this simulation models.

The following list explains the assumptions made in this simulation.

- When thrusting, the spacecraft undergoes a constant acceleration magnitude of 5 x 10^{-5} m/s^2, or 50 mN / 1000kg. This was thought to be a reasonable acceleration as a 2.5 kW ion thruster (NSTAR) can provide 92mN of thrust, and a GEO satellite can have up to 8 electric thrusters.

- The most important perturbation sources in the GTDS ephemeris propagator were turned on. This includes perturbations from the 50 x 50 geopotential model and third body point mass effects.
- During a maneuver, the thruster is on even when spacecraft is in shadow.
- Station keeping events occur once every 7 days.
- Station keeping events last until $\sqrt{G(t)} < 4$ hours .
- Control weights set to [10, 1, 1, 1, 1] unless otherwise noted. This allows for higher tolerances in control of the fast variable.
- All simulation runs in GTDS start execution March 1st, 2002.
- GTDS will sometimes exhibit bizarre behavior when the mean anomaly at epoch is set to 0° or 180°. When a satellite's orbit state demands being input at one of these values, it is instead incremented by one degree.
- A fixed step-sized Cowell integrator with a time step of 5 seconds was used for all simulation runs.
- All satellites were controlled to within ±1.25° of ground track angle

The duration of each simulation run will depend on that satellites particular orbit and its major perturbation sources, as will be discussed below.

Elliptical-Inclined Orbit Perturbative Environment

Both the Sirius Satellite Radio constellation and John Draim's "teardrop" constellations are repeat ground track orbits with high altitude apogees. These factors will play a dominant role in the perturbative environment of the satellites:

1) A repeat ground track orbit can (but will not always) experience much higher station keeping costs due to tesseral resonance, as opposed to an orbit that does not have a repeat ground track.
2) Tesseral resonance is entirely dependent on a satellite's ground track. Therefore, satellites within the same constellation that share a ground track will see very

similar perturbations from this source.

3) In general, a repeat ground track orbits' tesseral perturbations may increase in magnitude as a satellite's altitude drops.
4) An orbit with a high apogee altitude will experience larger drifts in inclination due to the lunar and solar point mass effects.
5) Perturbations from the lunar and solar point masses will vary only with the angle between the satellite's orbital plane and the ecliptic plane or the Earth-Moon plane. Since an orbit will be observed to precess at some approximately constant rate, the orientation of the orbital plane with respect to the Earth-Moon and Earth-Sun planes will vary with time. Therefore, the disturbances due to the lunar and solar point masses will vary with this precession period.

Points 4 and 5 are especially important to the discussion for choosing the duration of the simulation runs, as one would want an accurate picture of what to expect in terms of ΔV expenditure for a 15 year mission. For example, running a simulation for just one year (simulation time) may not yield accurate results for the average ΔV costs for the entire mission if the orbit was found to precess once every 3 years due to the variation of the lunar and solar point mass effects.

These facts will play a critical role in the following discussion of the two constellation's station keeping costs. The effects of these perturbations will be discussed in more detail in the following sections.

5.2 Verification and Testing

Before applying the Lyapunov code to the station keeping problem, it was first tested to see if it could complete a large element transfer. Setting the satellite thrust acceleration to a very high value would allow one to see the rapid evolution of the transfer orbit. The following plots show the convergence of the algorithm when transferring from a 300 kilometer altitude, equatorial, circular low Earth orbit to one of the Teardrop orbits (defined later). For our application, this is just to verify qualitatively that the algorithm is in fact working (Figure 9 and Figure 10).

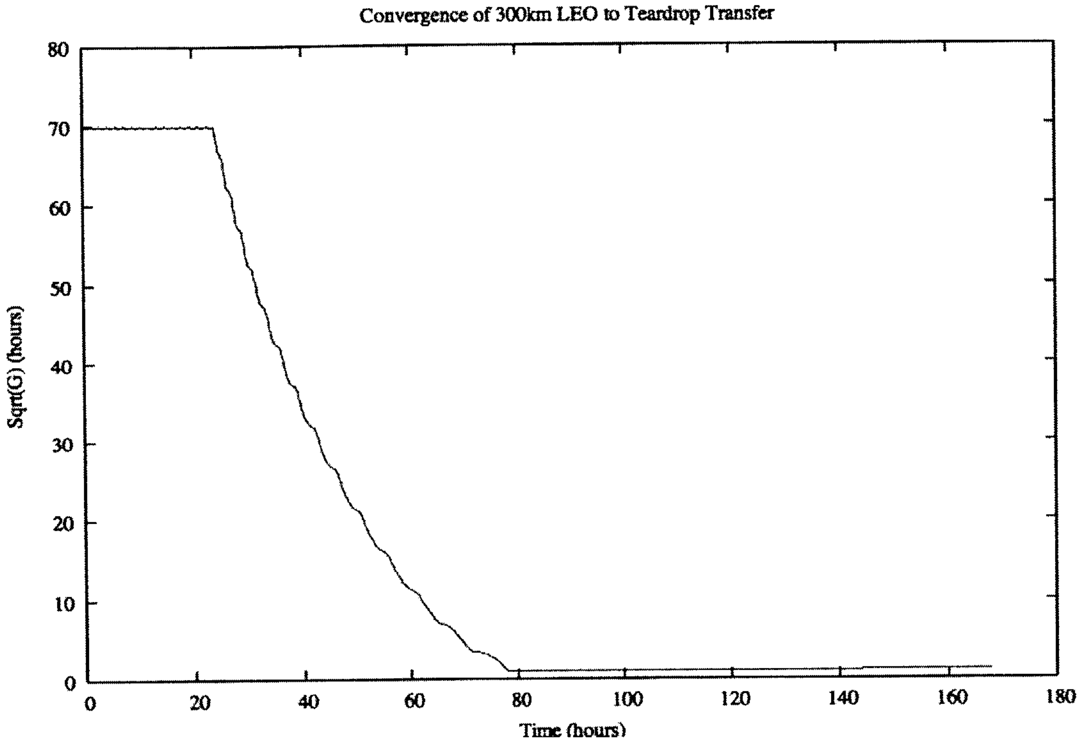

Figure 9: Convergence of Lyapunov Function for Large Element Change Transfer

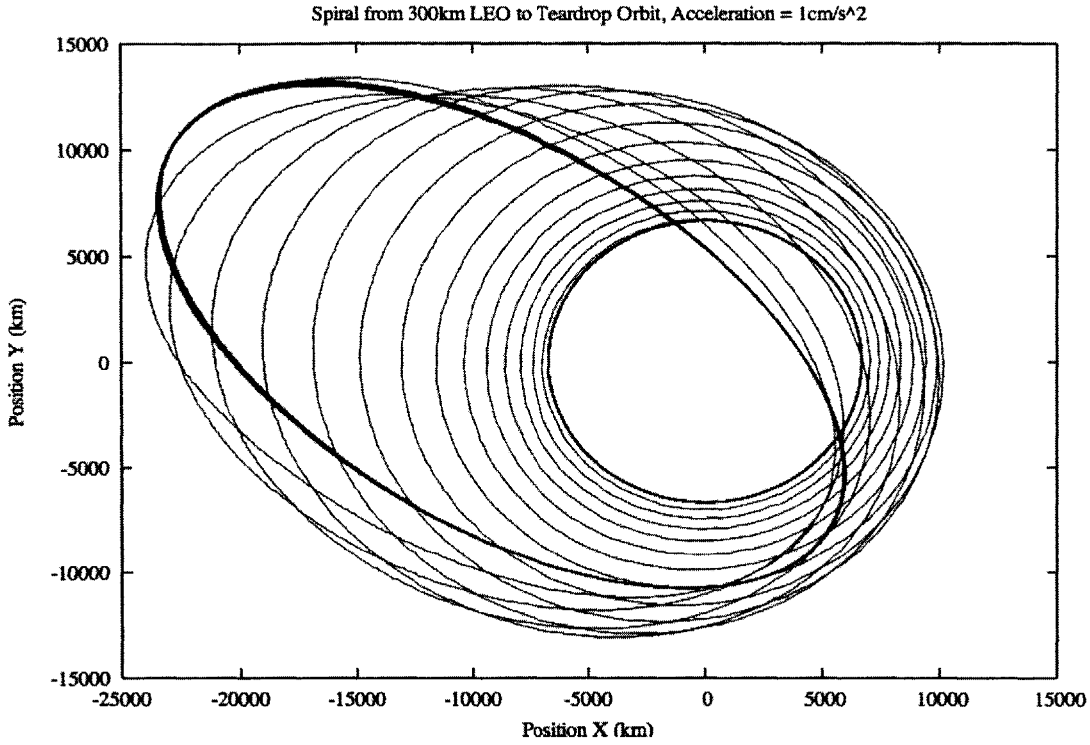

Figure 10: Evolution of Transfer Orbit from LEO to Teardrop

To verify quantitatively that the Lyapunov feedback control algorithm yields results consistent with well known real case , a geostationary Earth orbit (GEO) orbit is put through the simulation, with Ω being held at zero for purposes of ground track control. Two sets of control gains were used, one set where each gain is equal, and the other set for fine control over the semi-major axis. Satellite longitude was approximately 204° east of the Greenwich meridian.

The bulk of station keeping budget for GEO satellites is spent on north-south station keeping. These perturbations are due to the Sun and Moon torquing the orbit, causing a shift in the orbit's inclination. The ΔV expenditure due to this is more or less calculable analytically. For a high thrust orbit correction, 51.4 m/s must be spent annually[35] to counter this drift.

In terms of ΔV expenditure, low-thrust systems are always more inefficient. It can be shown analytically with a simple calculation of the spacecraft's moment arm on the orbit that correcting for the north-south drift with an electric thruster operating continuously has an efficiency that is 63.6% (2 / π) a high-thrust system[36] that would just normally fire at the nodes. Annual ΔV expenditure for a low-thrust system is then calculated:

$$51.4 \times \frac{\pi}{2} = 80.7 \text{ m/s/yr}$$

For comparison a geostationary satellite (a = 42164 km, e = 0, i = 0) is run through the Lyapunov feedback control algorithm in GTDS. The results are given below.

Table 6: Station Keeping a GEO Satellite

Gains	Low Thrust ΔV
[1,1,1,1,1]	78.2 m/s
[10,1,1,1,1]	92.3 m/s

Results seem to be comparable with the analytically predicted values. Increase of the semi-major axis control gain has a definite effect on the required ΔV. As discussed earlier, this is due to the ratio of the initial value of G(t) as compared to the cutoff threshold. Higher gains means that for the same thruster cutoff value of G(t) one will get a higher ratio between starting and ending values of G(t). This means the thruster will be on longer using the higher gain as compared to the lower gain case.

The semi-major axis was controlled to just a few kilometers of error, eccentricity was controlled to 2 decimal places, and all angular values such as inclination were controlled to at least 0.1 degrees.

5.3 Sirius Satellite Radio

Constellation Description

First launched in the year 2000, Sirius Satellite Radio's (SSR) constellation is an eccentric-inclined constellation with an orbital period of 24 hours. The satellites' orbits are such that at any given time one satellite is visible at elevation angle of at least 60° when over the continental United States. For convenience the constellation's orbital elements are given below.

Table 7: Sirius Constellation Orbital Elements

Designation	*a (km)*	*E*	*i (deg)*	*Ω (deg)*	*ω (deg)*	*M (deg) @Epoch*
Sirius 1	42164	0.2684	63.4	285	270	225
Sirius 2	42164	0.2684	63.4	165	270	345
Sirius 3	42164	0.2684	63.4	45	270	105

The Sirius constellation uses a high-thrust bipropellant propulsion system for orbit maintenance. The following table lists the estimated ΔV expenditures for each satellite in the constellation[37] using the high-thrust system.

Table 8: Sirius Constellation Annual High-Thrust ΔV Requirements

Designation	*Annual High-Thrust ΔV*
Sirius 1	55 m/s
Sirius 2	80 m/s
Sirius 3	35 m/s

Due to lunar and solar mass effects, and the high altitude at which this constellation operates, a wide variation in the station keeping ΔV is expected. Each satellite is on the same ground track so variation of ΔV is due almost entirely to the positioning of each satellite's orbital plane.

As mentioned earlier, an extremely important factor in perturbations from the sun and moon is the drift of the orbital plane (drift of Ω) as it causes the third body perturbations to vary. For this particular constellation, the drift rate is not large—only 4.2° a year. Thus, the precession cycle has a period of about 86 years. With this in mind, a year long simulation to determine average annual low-thrust ΔV requirements should yield results comparable with a 15 year long simulation. It should be noted that it took roughly 3 minutes of processor time to simulate a year in simulation time.

Table 9: Sirius Constellation Low-Thrust ΔV Requirements, Computed in GTDS

Designation	***Annual Low-Thrust ΔV***
Sirius 1	111 m/s
Sirius 2	137 m/s
Sirius 3	91 m/s

While the Sirius 2 annual ΔV expenditure at first glance appear to be extremely high, we will see later that low-thrust electric propulsion will still yield large mass savings.

The following plot shows the Lyapunov function G(t) for the first month of Sirius 1 station keeping. One can clearly see cutoff happening at $\sqrt{G(t)} = 4$ hours.

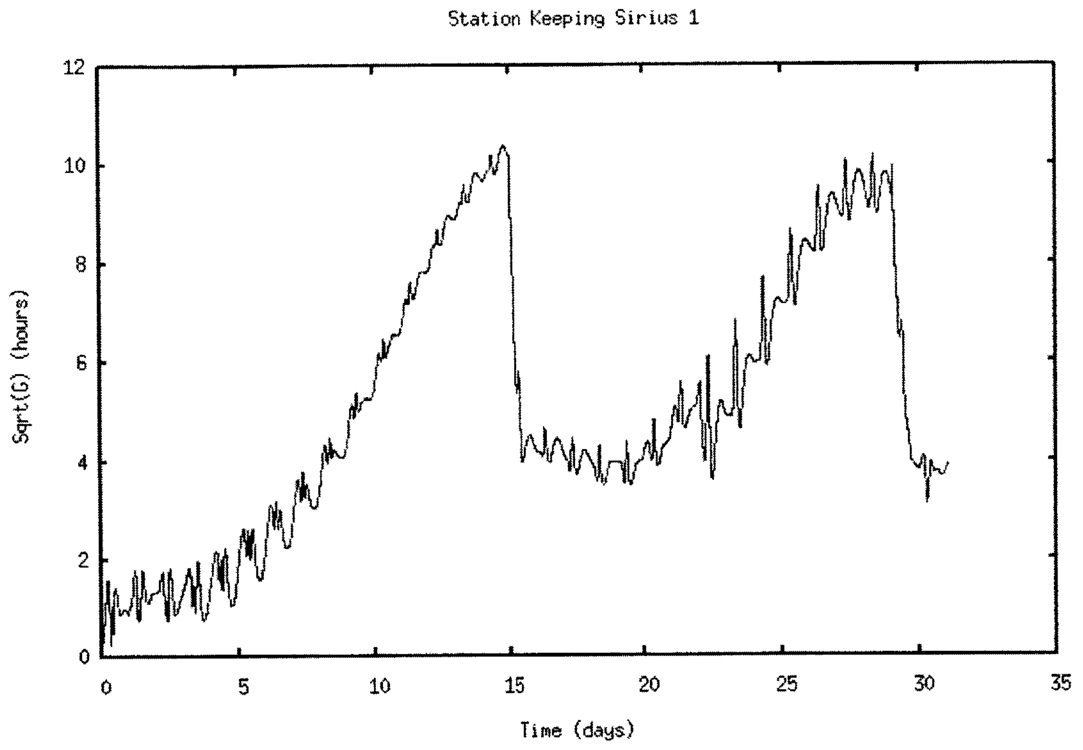

Figure 11: Convergence of Station Keeping in Perturbative Environment

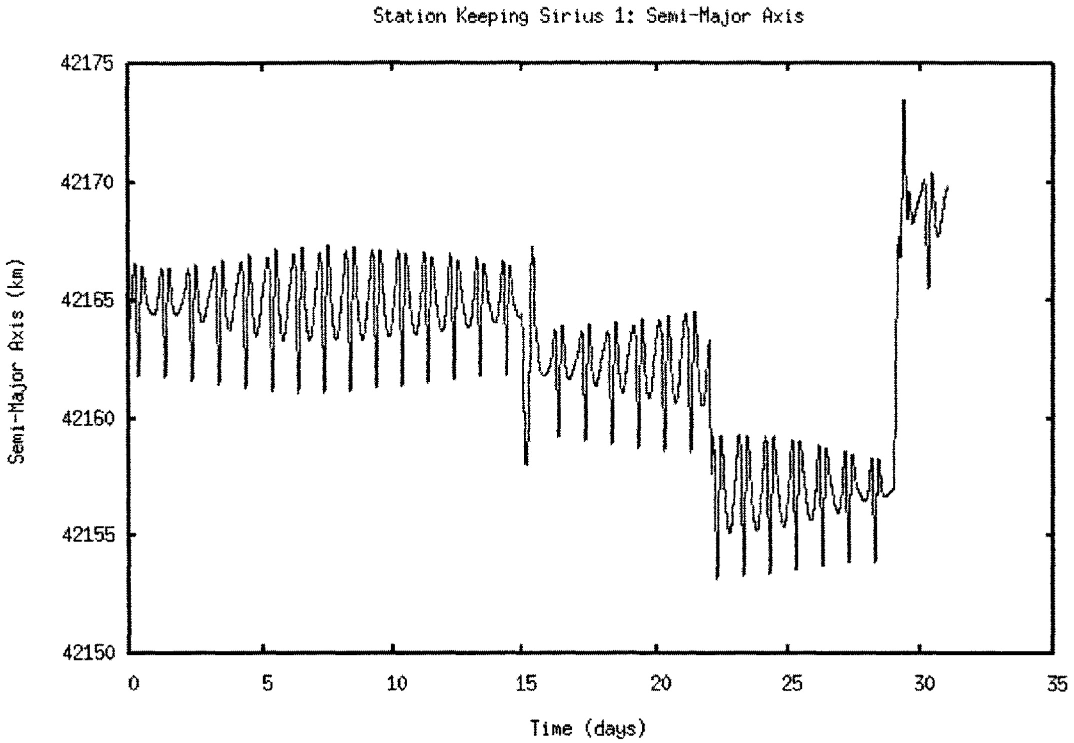

Figure 12: Station Keeping Sirius 1, Variation of *a* in First Month

Viewing the variation of the semi-major axis while Sirius 1 is on station, one can see outer loop controller varying the target semi-major axis to control the ground track angle. Also, one can clearly see the variation of the semi-major axis due to tesseral effects.

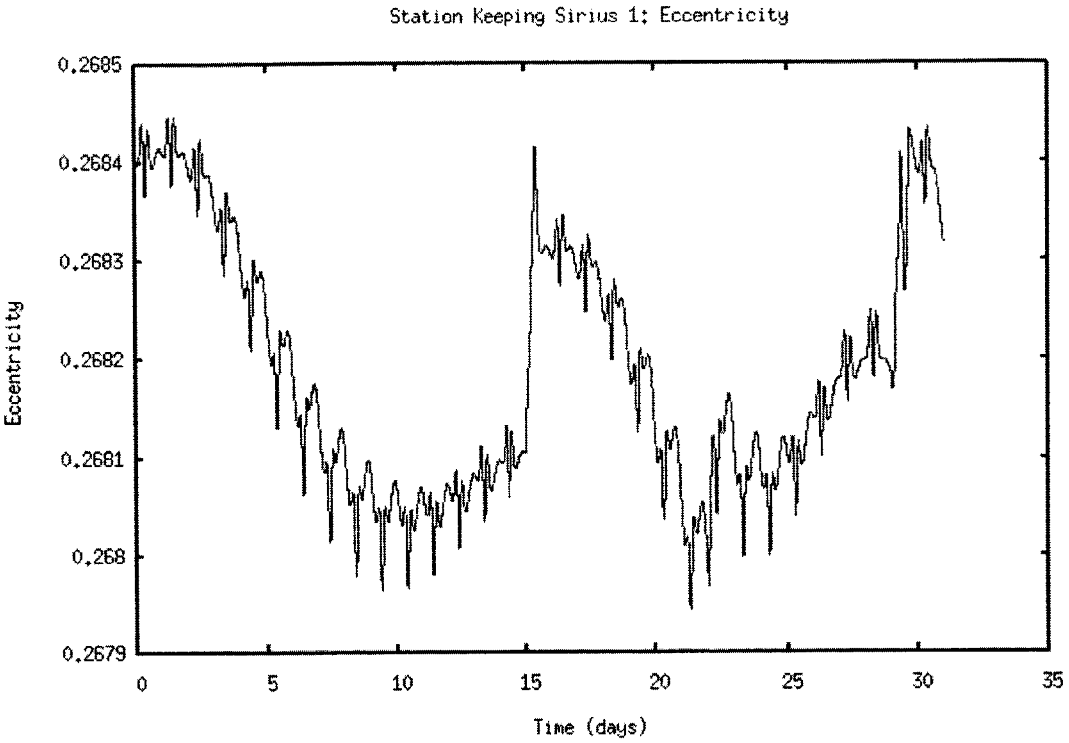

Figure 13: Station Keeping Sirius 1, Variation of *e* in First Month

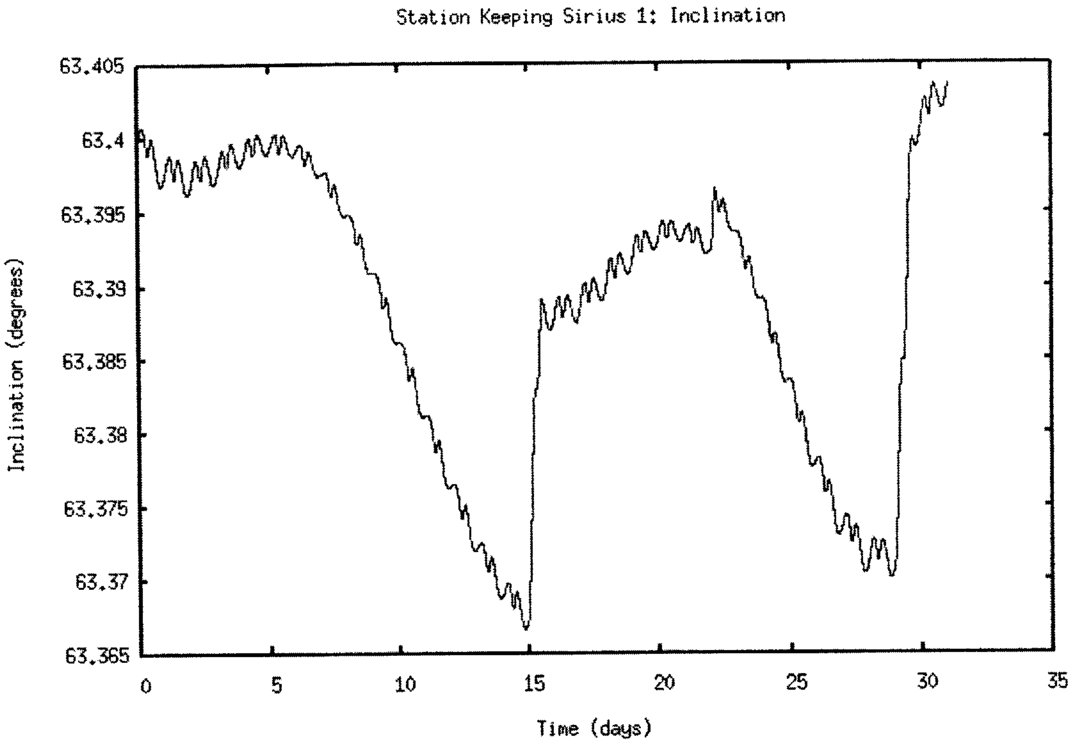

Figure 14: Station Keeping Sirius 1, Variation of *i* in First Month

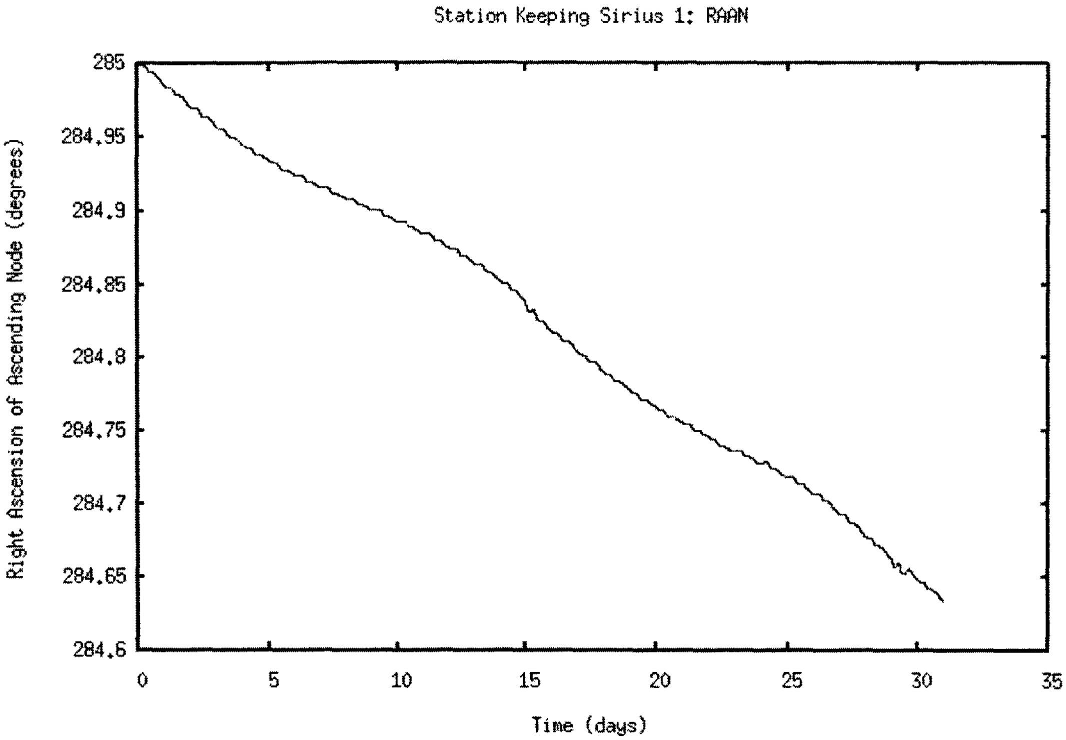

Figure 15: Station Keeping Sirius 1, Variation of Ω in First Month

The controller is operating successfully by allowing Ω to drift freely while still controlling the other orbital elements.

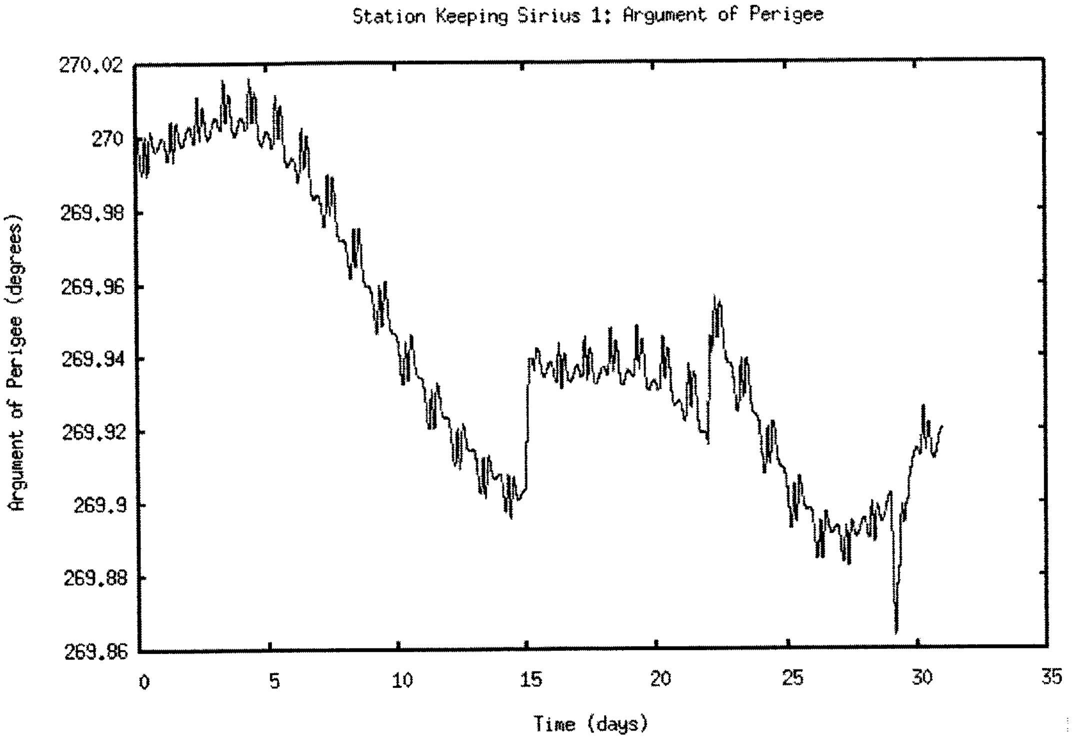

Figure 16: Station Keeping Sirius 1, Variation of ω in First Month

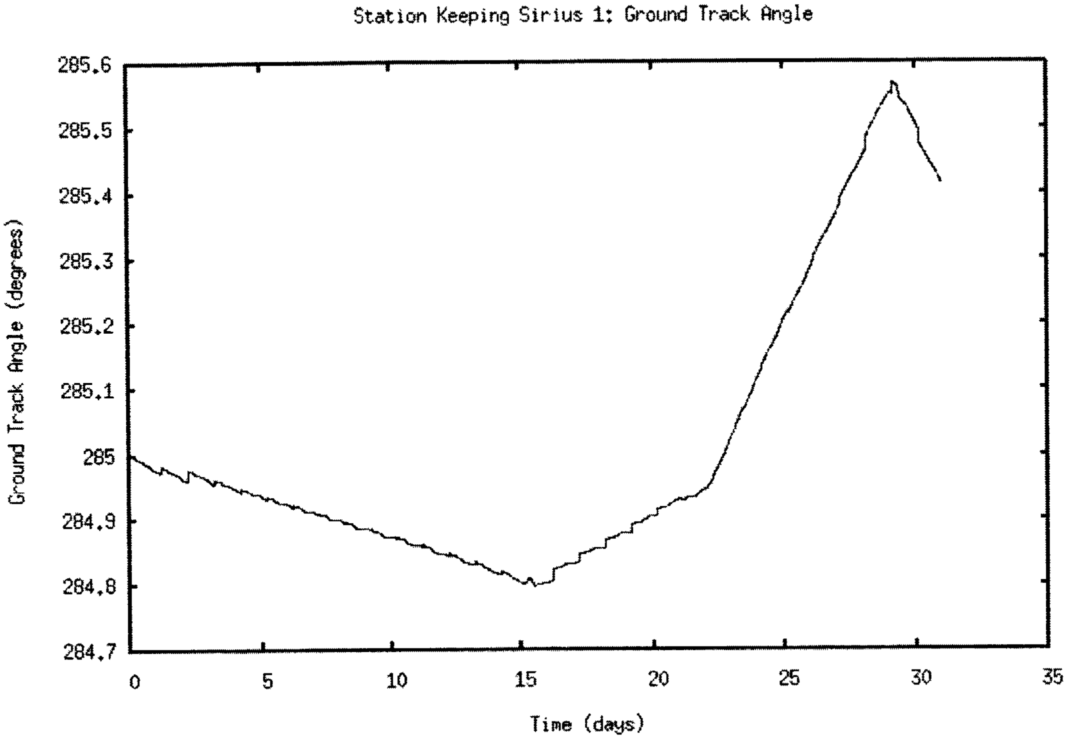

Figure 17: Station Keeping Sirius 1, Variation of Ground Track Angle (λ)

The ground track is controlled to within 0.8 degrees in the first month. Variation of ground track angle has never been observed to exceed more than 1.25 degrees in either direction. This translates into the satellite never being later or earlier than 5 minutes from expected passage overhead when the observer is on the ground. Tighter control can be achieved by decreasing time between station keeping events and lowering the Lyapunov cutoff threshold, but station keeping ΔV expenditure will increase.

5.4 Teardrop Constellation

The Teardrop Constellation[38] was invented by John Draim[39] around the year 2000 as an efficient method for providing continuous coverage at various locations at medium to high latitudes at 3 different locations spaced 120° apart in longitude. It is called a teardrop orbit because the overlap of the constellation ground tracks creates a teardrop-shaped region (see Figure 18) of continuous coverage. A teardrop constellation has the following characteristics:

- The constellation consists of 6 satellites in highly eccentric and inclined orbits.
- Each satellite has an orbital period of approximately 8 hours.
- There are two ground tracks, a left-leaning and a right-leaning ground track, and each ground track has 3 satellites. Teardrop 1,2, and 3 are left leaning, and teardrop 4, 5, and 6 are right leaning.
- At any given time at least one satellite is in the teardrop-shaped ground track region, thus providing continuous coverage in that region.

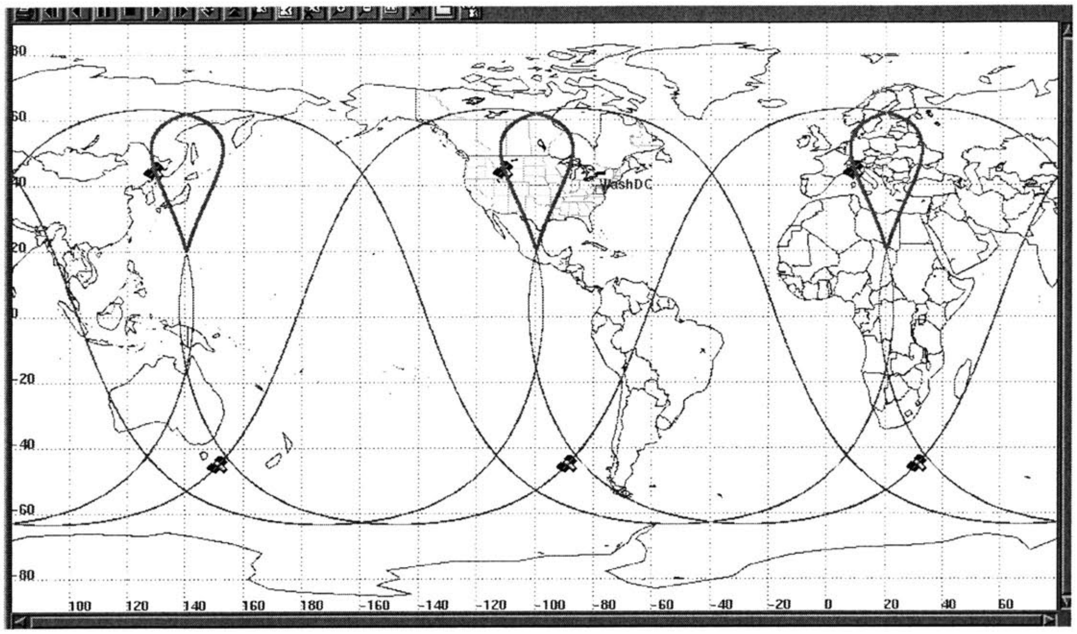

Figure 18: Basic 6-Satellite Teardrop Array. Left-Leaning Satellites at Apogees; Right-Leaning Satellites at Perigee

The orbital elements that provide this ground track are given below.

Table 10: Teardrop Constellation Orbital Elements

Designation	*a (km)*	*e*	*i (deg)*	*Ω (deg)*	*ω (deg)*	*M (deg) @Epoch*
Teardrop 1	20261	0.6458	63.41	138.5	232	181
Teardrop 2	20261	0.6458	63.41	18.5	232	181
Teardrop 3	20261	0.6458	63.41	258.5	232	181
Teardrop 4	20261	0.6458	63.41	100.2	308	1
Teardrop 5	20261	0.6458	63.41	340.2	308	1
Teardrop 6	20261	0.6458	63.41	220.2	308	1

The teardrop orbits were found to have Ω drift at a constant rate of approximately 85° a year. Every 4.23 years a teardrop satellite will precess completely, thus perturbations due to lunar and solar point masses will vary periodically in the time it takes for the each satellite to precess one revolution. For a 15 year service life, three complete precessions would be observed. Because of this, no satellite will have significantly more or significantly less ΔV costs due to the third-body perturbations—it will all more or less average, unlike the Sirius constellation. However, it is still possible that we will see definite differences between the right-leaning and left-leaning ground tracks.

To confirm this, two sets of simulations were run. One simulation was the one-year duration simulation, and the other was run for 4 years and 3 months (March 1st, 2002 – June 1st 2006).

Table 11: Teardrop Constellation Low-Thrust ΔV Requirements, Computed in GTDS (one year duration simulation)

Designation	***Annual ΔV Cost***
Teardrop 1	131 m/s
Teardrop 2	161 m/s
Teardrop 3	118 m/s
Teardrop 4	95 m/s
Teardrop 5	58 m/s
Teardrop 6	96 m/s

Table 12: Teardrop Constellation Low-Thrust ΔV Requirements, Computed in GTDS (4.25 years duration simulation)

Designation	***Annual ΔV Cost***
Teardrop 1	137 m/s
Teardrop 2	136 m/s
Teardrop 3	136 m/s
Teardrop 4	82 m/s
Teardrop 5	80 m/s

Teardrop 6	83 m/s

The data tends to support the notion that teardrop orbit ΔV expenditure experiences variations with a period of approximately 4.25 years. For a 15 year mission, average annual ΔV expenditure should be very close to the values reported by Table 8. The left leaning ground track results appear to be very close to Sirius' worst case.

5.5 Discussion and Application

Mass Savings

Now that the low thrust control algorithm has been successfully applied to the Sirius and Teardrop constellations, the usefulness of the results should be examined.
As can be seen with the Sirius case and the geostationary case, low thrust ΔV expenditures for station keeping will be higher than the high thrust case. While a worst case ΔV of 137 m/s/yr might seem excessively high, using a high-Isp system like a Boeing XIPS 25cm ion thruster[40] (Isp = 3500s, results obtained via the rocket equation) yields a propellant mass fraction of less than 6% for a 15 year mission. The Sirius high-thrust worst case (80 m/s/yr) yields approximately 31% propellant mass fraction if of one assumes a bipropellant system with an Isp of 325 seconds. Thus, there are significant performance advantages to using low thrust propulsion on these constellations.

Actual propellant mass fractions will almost certainly be larger than the approximate calculations given above. In addition to station keeping, there must be ΔV available for disposal, a certain number of orbital slot changes, and possibly transfers from the launch vehicle's final orbit to the operational orbit[41].

Practical Implementation

From a payload integration standpoint, though, implementation of an actual low-thrust system is trickier than this analysis might suggest. Electric propulsion thrusters on GEO satellites are typically located on the anti-nadir side of the satellite[42]. This is because the anti-nadir side is relatively unused "real estate" in which one can add new items to a standard spacecraft bus, like a thruster. Furthermore, these thrusters are typically gimbaled, but only slightly to compensate for center of mass uncertainty. The analysis put forth by this thesis makes no restriction on the thrust direction—one may always use the most optimal direction. Since a spacecraft's attitude profile is restricted during operation, it may be impossible to point the thruster in the most optimal direction without compromising pointing requirements.

For example, the Sirius Satellite Radio uses a concept called "Yaw Steering" to keep the satellite's solar arrays pointed at the sun for most of the year. This is best described as having one of the satellite's axes pointing nadir, but the satellite rotates about the nadir axis (yaw rotation) while pivoting its solar arrays to keep them pointed directly at the sun[43]. This attitude profile absolutely cannot be changed, thus any fixed thrusters would be constrained by this attitude profile. The result of this would to be to increase propellant requirements of the station keeping system. Proposing payload integration solutions, however, is beyond the scope of this thesis.

6.0 Conclusion

Lyapunov feedback control presents a simple and direct method available to determine thruster pointing directions. The extension of Petropoulos' work into equinoctial elements allows for robust control as the equinoctial element set does not suffer from the singularities that the Keplerian element set does. Implementation of this Lyapunov feedback control method in the Goddard Trajectory Determination System allows for determination of realistic station keeping ΔV requirements for satellites using low-thrust propulsion systems. Since the Lyapunov feedback controller only controls the slow orbital elements, the addition of a simple PI controller to control the fast variable proved adequate for keeping the satellite's ground track within desired tolerances.

The capability of Lyapunov feedback control system was verified by comparing the simulation results of a geostationary satellite to a well known and verifiable benchmark. Specifically, based on an analytic method it can be shown that using a continuous low-thrust propulsion system to maintain a geostationary orbit requires approximately 81 m/s of ΔV annually. The expenditure of ΔV in the GTDS simulation required 78 m/s of ΔV annually, which is close enough to the predicted value to confirm the simulation's accuracy.

This station keeping algorithm was then applied to two elliptical orbit constellation concepts: Sirius Satellite Radio's constellation and John Draim's "teardrop" constellation. It was found that for both constellations the worst-case average annual ΔV expenditure was approximately 137 m/s. While this at first appears to be a very high annual ΔV requirement when compared to a typical geostationary satellite station keeping budget, using a high Isp propulsion system will still yield a very small propellant mass fraction.

7.0 Future Work

There are a number of points that should be addressed by future work to alleviate some uncertainties in this analysis method.

Optimization of ΔV

First and foremost, in the work of Petropoulos there is some discussion of "effectivity". What this boils down to is that some sections of an orbit are more optimal to burn at than others—i.e. for a given impulse a greater reduction of the Lyapunov function G(t) is seen. If a satellite experiences the same perturbations, then the optimal points of firing will tend to be the same, too. Thus, it makes sense to only turn the thrusters on in the most effective places. This is what Petropoulos discusses for large element change transfers. Searching for these effective zones for large element change transfers is fairly straightforward: Determine the derivative of G(t) over one orbit (as if you were to fire the thruster) and find the sections of the orbit where this has the greatest magnitude. The overall change in G(t) over one orbit of continuous thrusting will not significantly change the location of the most effective thrusting zones. In contrast, small element change transfers are completed in a revolution or two of continuous thrusting. Thus, after some thruster firing at the initial most effective zone will have changed, and the zone that used to be the most effective can quickly become the worst. As one can see, some further study will be required to determine a way to optimize these firings.

Implementation of DSST

From an algorithmic efficiency standpoint, the current implementation in GTDS is not very efficient at all. This is mostly due to the fact the integration of the equations of motion is accomplished via fixed-step Cowell integration. Optimal efficiency could be attained by averaging the equations of motions and applying Draper Semi-analytic Satellite Theory (DSST). From a large element change perspective, DSST could be applied while thrusting since the thrust profile will not vary much from orbit to orbit. For small element change transfers application of DSST while thrusting seems to be of dubious value since thrusting only occurs for a short period of time, and each orbit will have a different thrust profile. Application of DSST while the spacecraft is not thrusting will of course speed up the simulation, although GTDS would have to be coded to switch integration methods when thrusting occurs. As an added benefit, the use of average elements will alleviate the need for a low pass filter to reduce the influence short periodic effects.

Extension of Low-Thrust Code Functionality

In the current effort, a very simplistic thrust model was developed. Namely, thrust was modeled as a constant acceleration. In the future one may wish to implement features such as variable thrust, variable Isp, propellant mass tracking, and so on. Additionally, there will undoubtedly be a desire to use these models beyond the ephemeris propagator module in GTDS, such as orbit determination and differential correction.

Appendix A: Source Code

EPCNTRL

```
SUBROUTINE EPCNTRL(ACTH,POS,VEL,TIME)
C ***************************************************************************
C
C  EPCNTRL is a lyapunov feedback control system that controls the slow
C  equinoctial elements.  This function relies upon the following
C  new subroutines:
C
C  epintegrator.for     Trapezoidal rule integration
C  eplambdacntrl.for    Groundtrack angle controller (a fast variable)
C  eplyapdir.for        Analytical determination of optimal thrust direction
C  eplyapopt.for        Sets initial values of common variables in lowthrust.cmn
C  epmaxrates.for       Determines maximum element rates based on current elements
C  eprate.for           Determines current element rates
C  epscheduler.for      Handles scheduling of station keeping events
C  eptgtelems.for       Changes target elements set to deal with repeat ground track
C  epxhrate2.for        Max element rate, minimized numerically
C  epxhrate3.for        Max element rate, minimized numerically
C  epxkrate2.for        Max element rate, minimized numerically
C  epxkrate3.for        Max element rate, minimized numerically
C  epelem.for           Determination of geodetic longitude for repeat groundtrack control (copy of elem.for)
C
C
C  Please see paper AAS 05-282 by Benjamin Joseph and Paul Cefola for
C  more information on how this algorithm works.

   IMPLICIT NONE

   INCLUDE 'lowthrst.cmn'

   REAL*8 ACTH(3)
   REAL*8 POS(3),POSMAG
   REAL*8 VEL(3),VELMAG
   REAL*8 ACC(3)
   REAL*8 EQNELM(6)
   REAL*8 TIME

   REAL*8 DPFRC
   REAL*8 XMU,SINL,COSL
   REAL*8 RATES(5,3)
   INTEGER*8 INFRC
   INTEGER*4 RETRO,I,J
   REAL*8 QPART(5),KS(3)
   REAL*8 MAXRATES(5),DIR(3)
   REAL*8 R(3),S(3),W(3),V(3)
   REAL*8 ACCELMAG,ACCELMAG2,INSHADOW
   REAL*8 REGULARA
   REAL*8 TACC(3)
   INTEGER INITDONE,WILLFIRE,FSUNLT
   REAL*8 DV,LAMBDA,LOWPASSA
   COMMON/FRC  / DPFRC(1300)     ,INFRC(50)
   COMMON /TRUELONG/ SINL,COSL
   COMMON /EPSWITCH/INITDONE
   EQUIVALENCE (DPFRC(2),XMU)
   EQUIVALENCE (FSUNLT,DPFRC(1269))

   ACCELMAG = EPACCELMAG

C   DETERMINE CONDITIONS THAT MAY NOT ALLOW US TO FIRE
```

```
C   CHECK FOR SUNLIGHT, currently doesn't do anything
      CALL CSHAD(FSUNLT)
      IF (FSUNLT .EQ. 0 .AND. EPSEP .EQ. 1) THEN
        INSHADOW = 1
      ELSE
        INSHADOW = 0
      ENDIF

C   CHECK IF IT IS THE SCHEDULED TIME TO FIRE

      CALL EPSCHEDULER(TIME,WILLFIRE)

      RETRO = 1
      IF (INITDONE .EQ. 1) THEN
C   RETRIEVE EQUINOCTIAL ELEMENTS
      CALL EQUIN   (EQNELM,RETRO,POS,VEL,XMU,.FALSE.)

C   CONDITION TARGET ELEMENTS BASED ON CONTROL TYPE
      CALL EPTGTELEMS(POS,VEL,TIME,LAMBDA)

C   IF WE'RE WORRIED ABOUT GROUND TRACK CONTROL, PUT SEMI-MAJ THROUGH LOWPASS
      REGULARA = EQNELM(1)
      IF (EPREPEATGRNDTRK .EQ. 1) THEN
        EQNELM(1) = EPLOWPASSA
      ENDIF

C   DETERMINE MAXIMUM ELEMENT RATES OVER THE CURRENT ORBIT
      CALL EPMAXRATES(MAXRATES,EQNELM,XMU)

C   DETERMINE THE CURRENT RATES
      CALL EPRATE(RATES,EQNELM,COSL,SINL,XMU)

C   CALCULATE THE LYAPUNOV FUNCTION PARTIALS WITH RESPECT TO TARGET ELEMENTS
      DO I=1,5
        QPART(I) =2*EPWEIGHT(I)/(MAXRATES(I)**2)*(EQNELM(I)-EPELMT(I))
      END DO

C   FORMULATE THE K's (COEFFICIENTS FOR THRUST DIRECTION DETERMINATION)
      DO J=1,3
        KS(J) = 0
        DO I=1,5
          KS(J) = KS(J) + QPART(I)*RATES(I,J)
        END DO
      END DO

C   DETERMINE THRUST DIRECTION THAT MINIMIZES LYAPUNOV FUNCTION RATES
      CALL EPLYAPDIR(KS,DIR)

C   CALCULATE ACCELERATION MAGNITUDE AND TRACK FUEL
C   (Sir Not Appearing in this Function)

C   EVALUATE LYAPUNOV FUNCTION
      EQNELM(1) = REGULARA
      QQ = 0
      DO I=1,5
        QQ = QQ + EPWEIGHT(I)*((EQNELM(I)-EPELMT(I))/MAXRATES(I))**2
      END DO
      QQ = QQ/ACCELMAG/ACCELMAG
C   RUN IT THROUGH A LOW PASS FILTER

C     YDOT = (QQ-QQQ)*(3e-4)

C     CALL EPINTEGRATOR(TIME,YDOT,4,QQQ)

C     QQ = QQQ

C   ROTATE THRUST INTO CORRECT COORDINATES

      POSMAG = SQRT(POS(1)**2+POS(2)**2+POS(3)**2)
      VELMAG = SQRT(VEL(1)**2+VEL(2)**2+VEL(3)**2)
      R(1) = POS(1)/POSMAG
      R(2) = POS(2)/POSMAG
      R(3) = POS(3)/POSMAG
      V(1) = VEL(1)/VELMAG
      V(2) = VEL(2)/VELMAG
      V(3) = VEL(3)/VELMAG
      CALL VCROSS(R,V,W)
      CALL VCROSS(W,R,S)
      S(1) = S(1)
      S(2) = S(2)
      S(3) = S(3)

C     Shadow call doesn't appear to be working -BEJ
C       IF ((INSHADOW .EQ. 0) .AND. (WILLFIRE .EQ. 1)) THEN
      IF (WILLFIRE .EQ. 1) THEN
        ACCELMAG2 = ACCELMAG
      ELSE
        ACCELMAG2 = 0
      ENDIF


      TACC(1) = ACCELMAG2*(R(1)*DIR(1)+S(1)*DIR(2)+W(1)*DIR(3))
      TACC(2) = ACCELMAG2*(R(2)*DIR(1)+S(2)*DIR(2)+W(2)*DIR(3))
      TACC(3) = ACCELMAG2*(R(3)*DIR(1)+S(3)*DIR(2)+W(3)*DIR(3))

      ACTH(1) = ACTH(1) + TACC(1)
      ACTH(2) = ACTH(2) + TACC(2)
      ACTH(3) = ACTH(3) + TACC(3)

C      Track Delta-V
      CALL EPINTEGRATOR(TIME,ACCELMAG2,2,DV)

C      record thrust directions

      IF (WILLFIRE .EQ. 1) THEN
        WRITE(116,*) TIME,TACC(1),TACC(2),TACC(3)
        WRITE(117,*) POS(1),POS(2),POS(3)
      ENDIF

       IF (TIME .GT. EPTIMEMULT*600) THEN
         EPTIMEMULT = EPTIMEMULT + 1
C           WRITE(115,*) TIME,EPACCELMAG,EPLYAPCUTOFF,EPTIMEWAIT
         WRITE(115,*) TIME,QQ,DV,LAMBDA,EPREPEATGRNDTRK
       ENDIF

      END IF

      RETURN

      END
```

EPINTEGRATOR

```
SUBROUTINE EPINTEGRATOR(XNEW,YNEW,J,CURRENTSUM)
      IMPLICIT NONE

      INCLUDE 'lowthrst.cmn'

C     This function uses a not-so-accurate trapezoidal integration method
C     of the incoming data to compute the integral at XNEW.  Can handle 10
C     different states.
      INTEGER J
      REAL*8 XNEW,YNEW,DX,CURRENTSUM
C     CHECK FOR FIRST INIT
      IF (EPINTCOUNT(J) .EQ. 0) THEN
        EPLASTX(J) = XNEW
        EPLASTY(J) = YNEW
      ENDIF
      EPINTCOUNT(J) = EPINTCOUNT(J) + 1
      DX = XNEW - EPLASTX(J)
      EPSUM(J) = EPSUM(J) + (YNEW+EPLASTY(J))/2*DX
      CURRENTSUM = EPSUM(J)
      EPLASTY(J) = YNEW
      EPLASTX(J) = XNEW


      RETURN
      END
```

EPLAMBDACNTRL

```
      SUBROUTINE EPLAMBDACNTRL(LAMBDA,TIME,RAAN,A)
      IMPLICIT NONE

C     A very simple controller that takes d(lambda)/dt and sets it equal to a drift
C     rate that will take it to zero by the time the next burn happens.  This should work
C     if the ratio between the time between burns ans the burn duration is large.
C     But yes, there will be steady state error.
      INCLUDE 'lowthrst.cmn'


      REAL*8 LAMBDA,A,RAAN
      REAL*8 RAANDRIFTRATE,LAMBDADRIFTRATE,L
      REAL*8 PSIDOTT
      REAL*8 XMU,TWOPI,SIDEREALDAY,DPFRC
      INTEGER*8 INFRC
      REAL*8 ACUBED,TIME
      REAL*8 ONETHIRD
      REAL*8 ERROR,INTERROR
      REAL*8 KI,KD,KP,KALL
      DATA TWOPI /6.2831 85307 17958 65/
      DATA SIDEREALDAY /86164.09/

      COMMON/FRC   / DPFRC(1300),INFRC(50)
      EQUIVALENCE (DPFRC(2),XMU)


      IF (EPBEENCALLED .EQ. 0) THEN

C     ESTIMATE DRIFT OVER LAST INTERVAL
        KALL = 1
C         KD = KALL/2
        KI = KALL*EPWAIT**(-2)/15
C         KI = 0
        KP = KALL*EPWAIT**(-1)/2
        KD = 0

        PSIDOTT = 0


C     STEADY STATE
      IF (EPREPEATGRNDTRK .EQ. 1) THEN
        RAANDRIFTRATE = (RAAN - EPLASTRAAN2)/(TIME-EPLASTTIME)
        PSIDOTT = -RAANDRIFTRATE
      ENDIF

        ERROR = EPLAMBDATGT - LAMBDA

C     PROPORTIONAL
        PSIDOTT = PSIDOTT + KP*ERROR

C     INTEGRAL
        CALL EPINTEGRATOR(TIME,ERROR,3,INTERROR)

        PSIDOTT = PSIDOTT + KI*INTERROR
```

```
C   DERIVATIVE
       LAMBDADRIFTRATE = (LAMBDA - EPLASTLAMBDA)/(TIME-EPLASTTIME)

       PSIDOTT = PSIDOTT + KD*LAMBDADRIFTRATE

       ACUBED=((1-PSIDOTT*SIDEREALDAY/TWOPI)*SIDEREALDAY/
     *       TWOPI/EPKMULT)**2*XMU
       ONETHIRD = 1
       ONETHIRD = ONETHIRD / 3
       EPLASTRAAN2 = RAAN
       EPLASTLAMBDA = LAMBDA
       EPLASTTIME = TIME

       EPBEENCALLED = 1

       A = ACUBED**ONETHIRD

     ENDIF

     RETURN

     END
```

EPLYAPDIR

```
      SUBROUTINE EPLYAPDIR(KS,DIR)
      IMPLICIT NONE
C     FUNCTION dQ/dT =  [KS(1),KS(2),KS(3)]* [SIN(PHI)COS(BETA);
C                         COS(PHI)COS(BETA);
C                         SIN(BETA)];
C     FINDS PHI AND BETA THAT MINIMIZES dQ/dT
C     IN A VERY NON-ELEGANT FASHION

      REAL*8 KS(3)
      REAL*8 DIR(3)
      REAL*8 PHIMAG,SINPHI,COSPHI
      REAL*8 BETAMAG,SINBETA,COSBETA,A1,A2
      REAL*8 PHIS(4,2),BETAS(4,2)

      INTEGER I,J
      PHIMAG = SQRT(KS(1)**2+KS(2)**2)
      IF (PHIMAG .LT. 1E-8) THEN
         SINPHI = 0
         COSPHI = 0
      ELSE
         SINPHI=KS(1)/PHIMAG
         COSPHI=KS(2)/PHIMAG
      ENDIF
      BETAMAG = SQRT(PHIMAG**2+KS(3)**2)
      SINBETA = KS(3)/BETAMAG
      COSBETA = PHIMAG/BETAMAG

C     CHECK SECOND DERIVATIVES TO MAKE SURE IT IS IN FACT A MINIMIZER

      PHIS(1,1) =  SINPHI
      PHIS(1,2) =  COSPHI
      PHIS(2,1) = -SINPHI
      PHIS(2,2) = -COSPHI
      PHIS(3,1) =  SINPHI
      PHIS(3,2) = -COSPHI
      PHIS(4,1) = -SINPHI
      PHIS(4,2) =  COSPHI

      BETAS(1,1)=  SINBETA
      BETAS(1,2)=  COSBETA
      BETAS(2,1)= -SINBETA
      BETAS(2,2)= -COSBETA
      BETAS(3,1)=  SINBETA
      BETAS(3,2)= -COSBETA
      BETAS(4,1)= -SINBETA
      BETAS(4,2)=  COSBETA

      DO J=1,2
        DO I=1,4
          A1=-KS(1)*PHIS(J,1)*BETAS(I,2)-KS(2)*PHIS(J,2)*BETAS(I,2)
          A2=-KS(1)*PHIS(J,1)*BETAS(I,2)-KS(2)*PHIS(J,2)*BETAS(I,2)
     -        -KS(3)*BETAS(I,1)
          IF ((A1 .GT. 0) .AND. (A2 .GT. 0)) THEN
            SINPHI = PHIS(J,1)
            COSPHI = PHIS(J,2)
            SINBETA = BETAS(I,1)
            COSBETA = BETAS(I,2)
          END IF
        END DO
      END DO
      DIR(1) = SINPHI*COSBETA
      DIR(2) = COSPHI*COSBETA
      DIR(3) = SINBETA
      RETURN
      END
```

EPLYAPOPT

```
      SUBROUTINE EPLYAPOPT()
      IMPLICIT NONE
C     THIS FUNCTION TAKES CARE OF SORTING OUT THE CONTROL INPUTS FROM
C     THE EPCNTRL1 AND EPCNTRL2 INPUT CARDS, AS WELL AS SOME COMMON
C     BLOCK VARIABLES INITIAL VALUES

      INCLUDE 'lowthrst.cmn'


      REAL*8 SIDEREALDAY
      INTEGER I

      DATA SIDEREALDAY /86164.09/
      EPWILLFIRE=0
C       EPSCHEDNUM = 1
      EPSCHEDNUM = 0
      QQ = 0
      EPLAMBDATGT = KEPELMT(4)
      EPTIMEMULT = 0
      EPLASTLAMBDA = EPLAMBDATGT
      EPLASTTIME = 60
      EPBEENCALLED = 0
      EPLASTRAAN = EPLASTLAMBDA
      EPLASTRAAN2= EPLASTLAMBDA
      EPRAANMULT = 0
      EPTRIPPED = 0
      EPLASTPOS3 = 0
      EPFIRSTGLON = 0

      IF (EPTIMEWAIT .LE. 0) THEN
        EPWAIT = 7 * SIDEREALDAY
      ELSE
        EPWAIT = EPTIMEWAIT
      ENDIF


C     NO GIMMICKS
      IF (EPCTRLTYPE .EQ. 0) THEN
        EPSEP = 0
        EPREPEATGRNDTRK = 0
      ENDIF

C     REPEAT GROUNDTRACK, SHUTOFF IN ECLIPSE, STATIONKEEP EVERY WEEK
      IF (EPCTRLTYPE. EQ. 1) THEN
        EPSEP = 1
        EPREPEATGRNDTRK = 1
      ENDIF

C     PLANAR REPEAT GROUNDTRACK
      IF (EPCTRLTYPE .EQ. 2) THEN
        EPSEP = 1
        EPREPEATGRNDTRK = 2
        EPLAMBDATGT = 0
      ENDIF



       DO I=1,10
         EPINTCOUNT(I) = 0
         EPSUM(I) = 0
       ENDDO
C     FOR LOWPASS FILTER FOR SEMI-MAJ. AXIS
      EPSUM(4) = KEPELMT(1)
      RETURN
      END
```

EPMAXRATES

```
      SUBROUTINE EPMAXRATES(RATES,EQNELM,XMU2)
      IMPLICIT NONE

C     Function calculates max velocity rates for equinoctial elements
C     from a true longitude of zero to 2*PI using a nonlinear iterative
C     method. WHICH ONE?
      REAL*8 RATES(5),EQNELM(6)
      REAL*8 A,XH,XK,P,Q,PP,PQ,NP,XMU
      REAL*8 COSL,SINL,TWOPI
      REAL*8 ARATE,ARATE1,ARATE2,PRATE,PRATE2,QRATE,QRATE2
      REAL*8 XHRATE,XHRATE2,XKRATE,XKRATE2,XHRATE3,XKRATE3
      REAL*8 PS,OE,PR,AX,BX,CX,TOL,XMIN,XMIN2
      REAL*8 EPXHRATE2,EPXKRATE2,EPXHRATE3,EPXKRATE3
      REAL*8 XMU2,ALMAG

      EXTERNAL EPXHRATE2,EPXKRATE2,EPXHRATE3,EPXKRATE3



      DATA TWOPI /6.2831 85307 17958 65/
      DATA TOL /0.0001/
      COMMON /MINX/A,XH,XK,P,Q,XMU
      XMU = XMU2
      A = EQNELM(1)
      XH = EQNELM(2)
      XK = EQNELM(3)
      P  = EQNELM(4)
      Q  = EQNELM(5)


C     A,P,Q RATES CAN BE SOLVED FOR ANALYTICALLY

C     FIND MAX OF A RATE

       PP = 2*SQRT(A**3/XMU/(1-XK**2-XH**2))

       ALMAG = (XH**2+XK**2)
       IF (ALMAG .GT. 0) THEN
         ALMAG = SQRT(ALMAG)
```

```
        SINL = -XK/ALMAG
        COSL = XH/ALMAG

        ARATE1 = PP*ABS(XK*SINL-XH*COSL)

        SINL = -XH/ALMAG
        COSL = XK/ALMAG
        ARATE2 = PP*(1+ABS(XK*COSL-XH*SINL))

        IF (ARATE1 .GT. ARATE2) THEN
          ARATE = ARATE1
        ELSE
          ARATE = ARATE2
        ENDIF
      ELSE
        ARATE = PP
      ENDIF
C    FIND MAX OF P RATE
      COSL = -XK
      SINL = SQRT(1-COSL**2)
      PR = XK*COSL+XH*SINL
      NP = SQRT((A/XMU)*(1-XK**2-XH**2))

      OE = (1+PR)
      PQ = (1+Q**2+P**2)

      PRATE = ABS(0.5*NP/OE*PQ*SINL)

      SINL = -SINL
      PR = XK*COSL+XH*SINL
      NP = SQRT((A/XMU)*(1-XK**2-XH**2))
      OE = (1+PR)
      PQ = (1+Q**2+P**2)

      PRATE2 = ABS(0.5*NP/OE*PQ*SINL)

      IF (PRATE2 .GT. PRATE) THEN
        PRATE = PRATE2
      ENDIF

C    FIND MAX OF Q RATE
      SINL = -XH
      COSL = SQRT(1-SINL**2)
      PR = XK*COSL+XH*SINL
      NP = SQRT((A/XMU)*(1-XK**2-XH**2))
      OE = (1+PR)
      PQ = (1+Q**2+P**2)

      QRATE = ABS(0.5*NP/OE*PQ*COSL)

      COSL = -COSL
      PR = XK*COSL+XH*SINL
      NP = SQRT((A/XMU)*(1-XK**2-XH**2))
      OE = (1+PR)
      PQ = (1+Q**2+P**2)

      QRATE2 =  ABS(0.5*NP/OE*PQ*COSL)

      IF (QRATE2 .GT. QRATE) THEN
        QRATE = QRATE2
      ENDIF

C    FIND MAX OF XH RATE

      AX = 0
      BX = 3
      CX = TWOPI;


      CALL GLDEN(AX,BX,CX,EPXHRATE2,TOL,XMIN)
      XHRATE2 = -EPXHRATE2(XMIN);
      AX = XMIN + 0.25*TWOPI
      BX = XMIN + 0.5*TWOPI
      CX = XMIN + 0.75*TWOPI
      CALL GLDEN(AX,BX,CX,EPXHRATE2,TOL,XMIN2)
      XHRATE = -EPXHRATE2(XMIN2)
      IF (XHRATE .GT. XHRATE2) THEN
        XHRATE2 = XHRATE
      END IF

      CALL GLDEN(AX,BX,CX,EPXHRATE3,TOL,XMIN)

      XHRATE3 = -EPXHRATE3(XMIN);
      AX = XMIN + 0.25*TWOPI
      BX = XMIN + 0.5*TWOPI
      CX = XMIN + 0.75*TWOPI
      CALL GLDEN(AX,BX,CX,EPXHRATE3,TOL,XMIN2)
      XHRATE = -EPXHRATE3(XMIN2)
      IF (XHRATE .GT. XHRATE3) THEN
        XHRATE3 = XHRATE
      END IF

      IF (XHRATE2 .GT. XHRATE3) THEN
        XHRATE = XHRATE2
      ELSE
        XHRATE = XHRATE3
      ENDIF
      IF (XHRATE .LT. NP) THEN
        XHRATE = NP
      ENDIF

C     FIND MAX OF XK RATE
      AX = 0
      BX = 3
      CX = TWOPI

      CALL GLDEN(AX,BX,CX,EPXKRATE2,TOL,XMIN)
      XKRATE2 = -EPXKRATE2(XMIN)
      AX = XMIN + 0.25*TWOPI
      BX = XMIN + 0.5*TWOPI
      CX = XMIN + 0.75*TWOPI

      CALL GLDEN(AX,BX,CX,EPXKRATE2,TOL,XMIN2)
      XKRATE = -EPXKRATE2(XMIN2)

      IF (XKRATE .GT. XKRATE2) THEN
        XKRATE2 = XKRATE
      END IF

      CALL GLDEN(AX,BX,CX,EPXKRATE3,TOL,XMIN)
      XKRATE3 = -EPXKRATE3(XMIN)
      AX = XMIN + 0.25*TWOPI
      BX = XMIN + 0.5*TWOPI
      CX = XMIN + 0.75*TWOPI

      CALL GLDEN(AX,BX,CX,EPXKRATE3,TOL,XMIN2)
      XKRATE = -EPXKRATE3(XMIN2)

      IF (XKRATE .GT. XKRATE3) THEN
        XKRATE3 = XKRATE
      END IF

      IF (XKRATE2 .GT. XKRATE3) THEN
        XKRATE = XKRATE2
      ELSE
        XKRATE = XKRATE3
      ENDIF
      IF (XKRATE .LT. NP) THEN
        XKRATE = NP
```

```
    ENDIF

    RATES(1) = ARATE
    RATES(2) = XHRATE
    RATES(3) = XKRATE
    RATES(4) = PRATE
    RATES(5) = QRATE
    RETURN
    END
```

EPRATE

```
SUBROUTINE EPRATE(RATES,EQNELM,COSL,SINL,XMU)
    REAL*8 A,XH,XK,P,Q,XMU,COSL,SINL
    REAL*8 PR,NP,PP,OE,PQ,QP
    REAL*8 RATES(5,3),EQNELM(6)


    A  = EQNELM(1)
    XH = EQNELM(2)
    XK = EQNELM(3)
    P  = EQNELM(4)
    Q  - EQNELM(5)

C   CALCULATE ELEMENT RATE MATRIX

    PR = XK*COSL+XH*SINL
    NP = SQRT((A/XMU)*(1-XK**2-XH**2))
    PP = 2*SQRT(A**3/XMU/(1-XK**2-XH**2))
    OE = (1+PR)
    PQ = (1+Q**2+P**2)
    QP = (Q*SINL-P*COSL)

    RATES(1,1) = PP*(XK*SINL-XH*COSL)
    RATES(1,2) =  PP*OE
    RATES(1,3) =  0

    RATES(3,1) =  NP*SINL
    RATES(3,2) =  NP*(XK+COSL*(2+PR))/OE
    RATES(3,3) =  NP*QP*XH/OE

    RATES(2,1) =  -NP*COSL
    RATES(2,2) =  NP*(XH+SINL*(2 + PR))/OE
    RATES(2,3) =  NP*QP*XK/OE

    RATES(4,1) =  0
    RATES(4,2) =  0
    RATES(4,3) =  0.5*NP/OE*PQ*SINL

    RATES(5,1) =  0
    RATES(5,2) =  0
    RATES(5,3) =  0.5*NP/OE*PQ*COSL

    RETURN
    END
```

EPSCHEDULER

```
    SUBROUTINE EPSCHEDULER(TIME,WILLFIRE)
    IMPLICIT NONE

    INCLUDE 'lowthrst.cmn'
```

```
      REAL*8 TIME
      INTEGER WILLFIRE
C     WAIT AND SEE IF IT IS TIME TO DO A STATION KEEPING EVENT
      IF (TIME .GT. EPSCHEDNUM*EPWAIT+86400) THEN
         EPWILLFIRE = 1
         EPSCHEDNUM = EPSCHEDNUM + 1
         EPQSTORE = QQ
C        RESET LAMBDA CONTROLLER
         EPBEENCALLED = 0
      ENDIF

      IF (SQRT(QQ) .LT. EPLYAPCUTOFF) THEN
         EPTRIPPED = EPTRIPPED + 1
         IF (EPTRIPPED .GT. 30) THEN
           EPWILLFIRE = 0
           EPTRIPPED = 0
         ENDIF
      ENDIF

      WILLFIRE = EPWILLFIRE


      RETURN
      END
```

EPTGTELEMS

```
      SUBROUTINE EPTGTELEMS(POS,VEL,TIME,LAMBDA)
      IMPLICIT NONE

      INCLUDE 'lowthrst.cmn'

      REAL*8 POS(3),VEL(3),TIME
      REAL*8 N(3),NMAG,H(3),HMAG
      REAL*8 RAAN,TWOPI,SIDEREALDAY,PI
      REAL*8 PSIDOT,PSI,LAMBDA,A,XMU
      REAL*8 DPFRC,CJ2NEG(2),AB(11)
      INTEGER*8 INFRC
      REAL*8 RMAG,VVMU,AINV,ATGT,YDOT
      REAL*8 GLAT,GLON
      REAL*8 SATPERIOD
      REAL*8 RSATPS
      INTEGER*8 ISATPS,ONE
      INTEGER*8 IBDY(9)

      REAL*8 CLASSE(6),ORBEL(20)
      REAL*8 ELEMS(6),SPHCOR(6)

      DATA TWOPI /6.2831853071795865/
      DATA PI    /3.1415926535897932/
      DATA SIDEREALDAY /86164.09/

      COMMON/FRC   / DPFRC(1300),INFRC(50)
      COMMON /SATPOS/  RSATPS (100)           ,ISATPS (6)
      EQUIVALENCE (DPFRC(2),XMU)
      EQUIVALENCE (DPFRC(859),CJ2NEG(1))
      EQUIVALENCE (DPFRC(24), AB(1))


      CALL VCROSS(POS,VEL,H)
      NMAG = SQRT(H(1)**2+H(2)**2)
      IF (EPREPEATGRNDTRK .EQ. 1) THEN
         IF (NMAG .GT. 1E-6) THEN
C         CALCULATE CURRENT RAAN
            N(1) = -H(2) / NMAG
            N(2) = H(1) / NMAG
      RAAN = ACOS(N(1))
      IF (N(2) .LT. 0) THEN
         RAAN = TWOPI-RAAN
      ENDIF
      RAAN = RAAN + TWOPI*EPRAANMULT
C   KEEP RAAN CONTINUOUS
      IF ((RAAN - EPLASTRAAN) .GT. PI) THEN
         EPRAANMULT = EPRAANMULT - 1
         RAAN = RAAN - TWOPI
      ELSE
         IF ((RAAN - EPLASTRAAN) .LT. -PI) THEN
            EPRAANMULT = EPRAANMULT + 1
            RAAN = RAAN + TWOPI
         ENDIF
      ENDIF

      KEPELMT(4) = RAAN

      CALL EQNKEP(EPELMT,1,KEPELMT,.FALSE.)

C     CALCULATE CURRENT SEMI-MAJ. AXIS
      RMAG=SQRT(POS(1)*POS(1)+ POS(2)*POS(2)+POS(3)*POS(3))
      VVMU=(VEL(1)*VEL(1)+VEL(2)*VEL(2)+VEL(3)*VEL(3))/XMU
      AINV = 2.D0 / RMAG - VVMU

      IF  (AINV .NE. 0.D0)  THEN
        A  =  1.D0 / AINV
      ELSE
        A  =  0.D0
      ENDIF
C     PUT SEMI-MAJOR AXIS THROUGH LOWPASS FILTER
      YDOT = (A - EPLOWPASSA)*(1e-4)

      CALL EPINTEGRATOR(TIME,YDOT,4,EPLOWPASSA)

C     TRACK GROUNDTRACK ANGLE


      SATPERIOD=TWOPI*SQRT(A**3/XMU)
      PSIDOT=TWOPI/SIDEREALDAY*(1-EPKMULT*SATPERIOD/SIDEREALDAY)

      CALL EPINTEGRATOR(TIME,PSIDOT,1,PSI)
      LAMBDA = RAAN + PSI


C     DETERMINE GEODETIC LONGITUDE OF ASCENDING NODE
       IF (EPLASTPOS3*POS(3) .LT. 0) THEN

      CALL EPELEM(TIME,POS,VEL,XMU,AB(1),CJ2NEG(1),1,CLASSE,ORBEL)
       GLON = ORBEL(13)
       GLAT = ORBEL(12)

         IF (EPFIRSTGLON .EQ. 0) THEN
            EPFIRSTGLON = 1
            EPGLONREF = GLON
         ELSE
C        APPLY CORRECTION TO GROUNDTRACK ANGLE TRACKER IF IT'S THE CORRECT GLON
C        (within 5 degrees)
            IF (ABS(GLON-EPGLONREF) < 5*PI/180) THEN
             EPSUM(1) = GLON - EPGLONREF + EPLAMBDATGT - RAAN
            ENDIF

         ENDIF

       ENDIF
       EPLASTPOS3 = POS(3)


C     CALL CONTROLLER
       CALL EPLAMBDACNTRL(LAMBDA,TIME,RAAN,ATGT)
```

```
          EPELMT(1) = ATGT

          EPLASTRAAN = RAAN

        ENDIF
      ENDIF


      IF (EPREPEATGRNDTRK .EQ. 2) THEN
        RAAN = 0

C        CALCULATE CURRENT SEMI-MAJ. AXIS
          RMAG=SQRT(POS(1)*POS(1)+ POS(2)*POS(2)+POS(3)*POS(3))
          VVMU=(VEL(1)*VEL(1)+VEL(2)*VEL(2)+VEL(3)*VEL(3))/XMU
          AINV = 2.D0 / RMAG - VVMU

          IF  (AINV .NE. 0.D0)   THEN
            A  =  1.D0 / AINV
          ELSE
            A  =  0.D0
          ENDIF


C        TRACK GROUNDTRACK ANGLE

         SATPERIOD=TWOPI*SQRT(A**3/XMU)
         PSIDOT=TWOPI/SIDEREALDAY*(1-EPKMULT*SATPERIOD/SIDEREALDAY)

         CALL EPINTEGRATOR(TIME,PSIDOT,1,PSI)
         LAMBDA = RAAN + PSI


         CALL EPLAMBDACNTRL(LAMBDA,TIME,RAAN,ATGT)
         EPELMT(1) = ATGT

      ENDIF
      RETURN
      END
```

EPXHRATE2

```
FUNCTION EPXHRATE2(L)
   IMPLICIT NONE
   REAL*8 SINL,COSL,XH,XK,P,Q,L,OE,PR,PS,NP,EPXHRATE2
   REAL*8 XMU,QP,A
   COMMON /MINX/A,XH,XK,P,Q,XMU

   COSL = COS(L)
   SINL = SIN(L)

   PR = XK*COSL+XH*SINL
   NP = SQRT((A/XMU)*(1-XK**2-XH**2))
   OE = (1+PR)
   QP = (Q*SINL-P*COSL)


   EPXHRATE2=-NP*ABS(((XH+SINL*(2+PR))/OE))
   END
```

EPXHRATE3

```
FUNCTION EPXHRATE3(L)
IMPLICIT NONE
REAL*8 SINL,COSL,XH,XK,P,Q,L,OE,PR,PS,NP,EPXHRATE3
REAL*8 XMU,QP,A
COMMON /MINX/A,XH,XK,P,Q,XMU

COSL = COS(L)
SINL = SIN(L)

PR = XK*COSL+XH*SINL
NP = SQRT((A/XMU)*(1-XK**2-XH**2))
OE = (1+PR)
QP = (Q*SINL-P*COSL)


EPXHRATE3=-NP*ABS((QP*XK/OE)**2)
END
```

EPXKRATE2

```
FUNCTION EPXKRATE2(L)
IMPLICIT NONE
REAL*8 SINL,COSL,XH,XK,P,Q,L,OE,PR,PS,NP,EPXKRATE2
REAL*8 XMU,QP,A
COMMON /MINX/A,XH,XK,P,Q,XMU

COSL = COS(L)
SINL = SIN(L)

PR = XK*COSL+XH*SINL
NP = SQRT((A/XMU)*(1-XK**2-XH**2))
OE = (1+PR)
QP = (Q*SINL-P*COSL)

EPXKRATE2=-NP*ABS(((XK+COSL*(2+PR))/OE))
END
```

EPXKRATE3

```
FUNCTION EPXKRATE3(L)
IMPLICIT NONE
REAL*8 SINL,COSL,XH,XK,P,Q,L,OE,PR,PS,NP,EPXKRATE3
REAL*8 XMU,QP,A
COMMON /MINX/A,XH,XK,P,Q,XMU

COSL = COS(L)
SINL = SIN(L)

PR = XK*COSL+XH*SINL
NP = SQRT((A/XMU)*(1-XK**2-XH**2))
OE = (1+PR)
QP = (Q*SINL-P*COSL)

EPXKRATE3=-NP*ABS((QP*XH/OE)**2)
END
```

References

[1] Soviet Space Programs 1962-1965, Library of Congress, Legislative Reference Service p. 327-329

[2] "The Evolution of Highly Eccentric Orbits", p. 37. Jack Fischer, Master's Thesis, MIT, 1998

[3] http://www.astronautix.com/craft/molniya3.htm, accessed on September 16, 2005.

[4] "An Evaluation of Semianlaytical Satellite Theory Against Long Arcs of Real Data for Highly Eccentric Orbits", Martin Feiger, Master's Thesis, MIT, 1987

[5] Personal communications with Dr. Paul Cefola

[6] *"Plasma Propulsion Benefits for the Orbit Transfer and Station Keeping Functions"* Chrisophe R. Koppel, 52nd International Aeronautical Congress

[7] Personal Correspondence with Anistassios Petropoulos

[8] "*Optimal Low Thrust Geocentric Transfer*", Edelbaum, Sackett, Malchow, AIAA 1973-1074

[9] *"Survey of Numerical Methods for Trajectory Optimization"*, John T. Betts, Journal of Guidance, Control, and Dynamics, Vol. 21, No. 2, March-April 1998

[10] *"Optimal Station Keeping Strategies via Parallel Genetic Algorithms"*, AAS 99-123, J. E. Smith, R. Proulx, and J. Draim

[11] Fundamentals of Astrodynamics and Applications, 2nd Ed., David Vallado, p. 212-213

[12] http://aa.usno.navy.mil/faq/docs/ICRS_doc.html, accessed on January 29th, 2006

[13] Fundamentals of Astrodynamics and Applications, 2nd Ed., David Vallado, p. 162

[14] *"Euler Parameters as Nonsingular Orbital Elements in Near-Equatorial Orbits"*, Pini Gurfil, Journal of Guidance, Control, and Dynamics, Vol 28. No. 5, Sept.-Oct. 2005

[15] *"A Recursively Formulated First-Order Semianalytic Artificial Satellite Theory Based on the Generalized Method of Averaging"*, W.D. McClain, Volume 1, Computer Sciences Corporation, CSC/TR-77/6010, 1977

[16] "*Application of the equinoctial elements for perturbation approach to orbit raising*", J.J. Martinez Benjamin, AAS 94-158

[17] Liapunov Functions and Stability in Control Theory, Lecture Notes in Control and Information Sciences, Vol. 267. Andrea Bacciotti and Lionel Rosier, 2001.

[18] Nonlinear Dynamics and Chaos, Steven H. Strogatz, 1994, p. 141

[19] *"Refinements to the Q-Law for Low-Thrust Orbit Transfers"*, Anastassios E. Petropoulos, AAS 05-162

[20] "Classical Element Feedback Control for Spacecraft Orbital Maneuvers", Bo J. Naasz, Master's Thesis, Virginia Polytechnic Institute

[21] See ref. 19

[22] Numerical Recipes in C, p. 397. Press, Teukolsky, Vetterling, Flannery, 2002.

[23] Fundamentals of Astrodynamics and Applications, 2nd Ed., David Vallado, p. 126

[24] Fundamentals of Astrodynamics and Applications, 2nd Ed., David Vallado, p. 788

[25] An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition, Richard H. Battin, p. 504

[26] Modern Control Systems, Ninth Ed., p. 241, Richard C. Dorf, Robert H Bishop, 2000

[27] Modern Control Systems, Ninth Ed., p. 695, Richard C. Dorf, Robert H Bishop, 2000

[28] Personal Correspondence with Anistassios Petropoulos

[29] Conversations with Dr. Paul Cefola

[30] http://www.dwheeler.com/sloccount/, accessed on 2/1/2006

[31] "Geosynchronous Orbit Determination Using Space Surveillance Network Observations and Improved Radiative Force Modeling", Richard H. Lyon, Master's Thesis, MIT, 2004

[32] "Implementing a 50x50 Gravity Field Model in an Orbit Determination System", Daniel J. Fonte, Master's Thesis, MIT, 1993

[33] "Precision Orbit Determination from GPS Receiver Navigation Solutions", Scott S. Carter, Master's Thesis, MIT, 1996

[34] "*Practical Challenges in Implementing Atmospheric Density Corrections to the NRLMSISE-00 Model*", Matthew P. Wilkins, Chris Sabol, Paul Cefola, Kyle Alfriend, AAS 06-170

[35] Space Mission Analysis and Design, Wertz, p. 157

[36] "*Optimum Low-Thrust Rendezvous and Station-Keeping*", Theodore N. Edelbaum, AIAA 1963-154

[37] "*S-DARS Broadcast From Inclined, Elliptical Orbits*", Briskman & Prevaux, IAF-01-M.5.04

[38] "*The Orbital Perturbation Environment for the Cobra and Cobra Teardrop Elliptical Constellations.*" Cefola, Draim, Inciardi, Proulx, 2002

[39] "*Satellite constellations (The Breakwell Memorial Lecture - 55th International Astronautical Congress*)", John E Draim, 55th International Astronautical Congress 2004, Oct. 2004. pp. 1-30.

[40] AIAA 2002-4348, "*Performance of XIPS Electric Propulsion in On-orbit Station Keeping of the Boeing 702 Spacecraft*", D.M.Goebel, M. Martinez-Lavin, T.A.Bond, A.M.King.

[41] Personal correspondences with Captain (ret.) USN John Draim

[42] See Ref. 40

[43] See Ref. 37