ORBITAL MANEUVERING USING ELECTRODYNAMIC TETHERS

THESIS

Hakan San, 2nd Lieutenant, TuAF

AFIT/GA/ENY/02-3

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/ENY/02-3

ORBITAL MANEUVERING USING ELECTRODYNAMIC TETHERS

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Hakan San, BS

2nd Lieutenant, TuAF

March 2002

AFIT/GA/ENY/02-3

ORBITAL MANEUVERING USING ELECTRODYNAMIC TETHERS

Hakan San, BS
2 Lt, TuAF

Approved:

<u> *// Signed //* </u>		<u> 14 Mar 2002 </u>
Steven G. Tragesser		date
Thesis Advisor		

<u>   *// Signed //*   </u>   <u> 14 Mar 2002 </u>
William E. Wiesel   date
Committee Member

<u>   *// Signed //*   </u>   <u> 14 Mar 2002 </u>
Gregory S. Agnes   date
Committee Member

Acknowledgements

Table of Contents

List of Figures

List of Tables

Abstract

Considerably short electrodynamic tethers can be used as a low-thrust propulsion system with minimum or no mass expenditure. The propulsive force is generated by inducing a current flow through the tether which interacts with the magnetosphere of the planet (i.e. Earth, Jupiter and other planetary bodies with magnetosphere). The basics of electrodynamic tether systems have been studied and successful experiments such as boost and de-boost of spacecraft have been conducted in the past. This study presents a simple guidance scheme for the current in the tether in order to perform orbital maneuvers.

The general perturbation equations are used to develop the guidance scheme algorithm. The tether is assumed to be perfectly aligned with the local vertical and the tether flexibility is neglected. The guidance is capable of both in-plane and out-of-plane maneuvers, simultaneously changing the orbit parameters. Several numerical examples are simulated that demonstrate the ability of the guidance to accurately maneuver the vehicle. The simplicity of the guidance law allows it to be suitable for mission planning and on-board implementation.

ORBITAL MANEUVERING WITH ELECTRODYNAMIC TETHERS

I. Introduction

1.1 Overview

There are many potential uses of tethers and tethered systems. The electrodynamic tether, in which we are interested, is a conductive wire where a generated or provided current runs through. There are numerous possible applications of such a system. Main applications include power generation, propulsion and self powered ultra-low-frequency broadcast antenna [1].

Reasonably short electrodynamic tethers can be used as a low-thrust propulsion system with minimum or no mass expenditure. The propulsive force is generated by inducing a current flow through the conductive tether, which interacts with the magnetosphere of the planet (i.e. Earth, Jupiter and other planetary bodies with magnetosphere). The tether thrust depends on the current level, magnetic field strength and the orbital velocity. In addition, the available current level depends on the specifications of the spacecraft and the surrounding ion density which limits the conductivity of the current path [2].

Hollow cathode plasma contactors, field emitter array cathodes and bare tether systems are the existing interface concepts to facilitate electron transfer between the

tether and the surrounding ambient plasma. Hollow cathode plasma contactors have been widely used in electrodynamic tether experiments and spacecraft discharging applications. While they use some expendable gas for electron collection and emission, the mass of this gas is minimal compared to the propellant usage of traditional propulsion systems. On the other hand, field emitter array cathodes and bare conductive tethers do not require any expendable gas and are able to provide the necessary current level. Finally a bare conductive tether can be used for electron collection [2].

The electrodynamic tether propulsion system provides low-thrust continuous propulsion. Low-thrust propulsion can significantly reduce the amount of propellant used in chemical propulsion systems. However, low-thrust orbit maneuvering requires thrust steering and continuous guidance.

1.2 Problem Statement

A large percentage of the payload for many spacecraft launches is the propellant for either onboard usage or for upper stages. This increases the cost while reducing the actual payload capacity. Furthermore, a spacecraft's lifetime is limited by the amount of fuel carried by the spacecraft.

Maneuvering with electrodynamic tethers has potential uses for new ways of operating in the space. The nearly propellantless nature of the system could enable many possible applications and long-term missions in Low Earth Orbit (LEO). The basics of such systems have been studied widely and successful experiments on electrodynamic

tether systems have been conducted in the past. Some possible applications could be achieved in the near future.

A good example of electrodynamic tether propulsion is the boost and de-boost of spacecraft. This concept has been well-developed by several researchers [3, 4, 5] and will be demonstrated in the NASA ProSEDS mission. Other possible applications of electrodynamic propulsion systems could be a satellite servicing vehicle or a satellite tug. A satellite-servicing vehicle could rendezvous with multiple satellites in different orbits for refueling or replacing components and thus extend the lifetime of expensive space assets. An orbital tug vehicle could change the orbits of LEO satellites [2].

The main disadvantages of electrodynamic tethers are the operational altitude restriction and long period of time required to achieve orbital maneuvers. The electrodynamic tether uses the plasma in the ionosphere for the return path of the current. Unfortunately, the plasma density above LEO is not sufficient for conductivity. Secondly, the thrust that can be achieved with practical system design is low and maneuvers can take even months to perform. Therefore, an electrodynamic tether system should be designed as autonomous as possible limiting human support for these long duration operations and making the system more cost-effective.

As mentioned above, a continuous guidance scheme is required for the system to perform orbital maneuvers. The guidance law should be capable of accomplishing both in-plane and out-of-plane maneuvers. A simple guidance scheme that requires minimal calculations would enable onboard implementation. Such guidance scheme could also be used by mission planners to execute trade studies and system performance evaluation.

1.3 Objective

The objective of this study is to develop a simple guidance scheme for orbital maneuvering with electrodynamic tethers and to demonstrate its applicability. The guidance scheme should be capable of simultaneously changing the orbit's size, shape and plane to achieve the desired orbital maneuver. Furthermore, it should impose a minimal computational load.

II. Background and Literature Review

2.1 Introduction

The electrodynamic tether is a long conductive wire extended from a spacecraft that carries a current. The tether tends to stay aligned with the local vertical in an absence of any external force by the gravity gradient torque. The major applications suggested for an electrodynamic tether system are power generation, propulsion, and self-powered ultra-low-frequency broadcast antennas [1].

2.1.1 Power Generation. Electrodynamic tethers can be used in low earth orbits for power generation. Moving a conductor in a magnetic field generates an electromotive force (EMF), which drives electrons through the conductor. To achieve this current flow, the circuit must be closed either by emitting the electrons back to the surroundings or by collecting positive ions. For electrodynamic tether applications, the current flow takes place by collecting electrons from ionosphere at one end of the tether and emitting back into ionosphere at the other end of the tether. Using a conductive tether in low earth orbit at inclinations from low- to mid-latitude, 50 to 250 volts of EMF can be generated per a kilometer length of tether [2]. However, the cost of converting orbital energy to EMF is de-orbiting the satellite.

In 1994, Plasma Motor-Generator (PMG) rocket-borne tether experiment was flown. The PMG used hollow cathode devices as the plasma interface at each end of the system. The tether was only 500 meters and tether currents of 0.3 Amperes were achieved

[7]. A more significant current flow (~1A) through an electrodynamic tether was achieved near 300-km altitude during NASA's TSS-1R mission in 1996 [8].

2.1.2 Propulsion. Inducing a current flow through a conductive tether, the electrical energy can be converted into tether force. Fairly short electrodynamic tethers can be used as means of a propulsion system. The current in the tether interacts with the earth's magnetic field to produce Lorentz force on the tether. The force generated on the tether is a function of the current level, magnetic field strength and the orbital velocity. Electrodynamic propulsion is the subject of this study.

2.2 Electrodynamic Tether Propulsion

As mentioned above, the propulsive force generated by the electrodynamic tether depends on the current level and magnetic field strength. While the earth's magnetic field decreases by the inverse cube of the distance from the earth's center, the current level available to the tether is determined by a number of factors including the ambient ion density and the system's electron collection and emission capability. The electron transfer between the system and the ionosphere is conducted through plasma contactors. While hollow cathode emitters are commonly used in related space applications, field emitter array cathodes and bare tether systems has become attractive interface concepts, recently.

2.2.1 Earth's Magnetic Field. Around the mid-nineteenth century, Ampere and other scientist pointed out that electric currents are the source for all magnetic fields, including the geomagnetic field. Today, it is known that part of Earth's core is liquid with metallic properties, and it is believed that the motion of the fluid metal generates currents,

which induce the Earth's magnetic field. A simple approach to modeling Earth's magnetic field is the magnetic field of a sphere uniformly magnetized around a dipole axis. The points where this dipole axis intersects Earth's surface are called geomagnetic poles and the plane through the center of the Earth, perpendicular to the dipole axis is called geomagnetic equator. The angle between the geomagnetic equator and the geographical equator is 11.3 degrees [9].

This dipole model indicates that the magnetic intensity over the poles is twice the intensity over the equator, also the magnetic field strength drops proportional to the cube of the distance from the dipole center. Although the Earth's magnetic field is not a perfect dipole, its characteristics are similar to the simple dipole approximation. There are strong anomalies in the actual field mainly caused by the irregularities and eddies in the current system of Earth's core that drives the magnetic field. Less severe anomalies are caused by the ferromagnetic materials in the crust, solar and lunar gravitational effects and the magnetic disturbance from the Sun. Finally, with a little bit of sense of humor, we should expedite using electromagnetic tether systems because historical records indicate that the strength of Earth's magnetic field is decreasing at a rate that will eliminate the field in 3000 years [9].

2.2.2 Plasma Electron Density. An important parameter determining the thrust level that can be generated by the electrodynamic tether is the electron density in the surrounding plasma. There are several temporal and spatial variations in the electron density. The ionosphere has a vertical electron density profile with a distinct difference between day and night (Figure2.1). Moreover, the electron density varies between

latitudes and there are density differences in different parts of the ionosphere due to the structure of the magnetosphere.
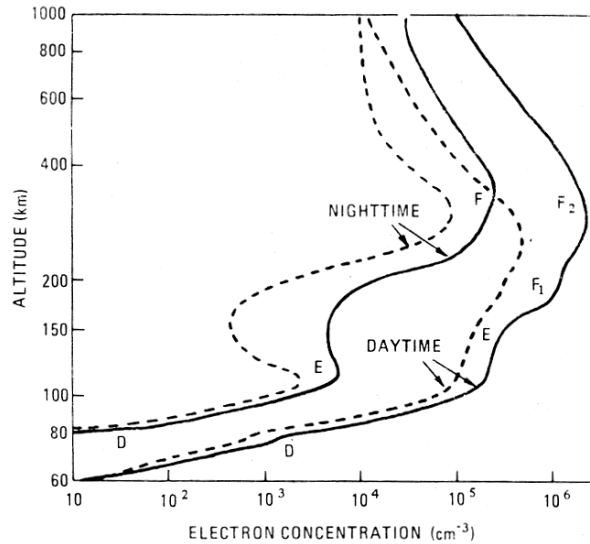
Figure 2.1 Typical Midlatitude Daytime and Nighttime Electron Density Profiles
for Sunspot Maximum (Solid Lines) and Minimum (Dashed Lines) [9]

The ionosphere has features dramatically varying with the geomagnetic latitude. Mainly, low latitude ionosphere is subject to instabilities due to variations in magnetosphere and high latitude ionosphere is strongly affected by auroral magnetic field. The mid-latitude ionosphere is easy to model and more stable. Besides the variations due to geomagnetic latitude, other variations occur between day and night, seasons of the year and due to solar activity [9].

The electron density goes under regular variations between day and night. Since the ionization occurs due to the solar radiation, the electron density falls down during the night. In addition to the change in the ionization rate between day and night, during the day the atmosphere expands by the solar heating and the ionosphere rises. There is a strict

2-4

relationship between the solar activity and electron density. During low solar activity, the electrons densities are half to a quarter of that during high solar activity [9]. There are also variations resulting from the atmospheric tides created by solar and lunar gravity. The electron density profile experienced by the International Space Station during a particular day is shown in Figure 2.2.

Electrons per Cubic Meter
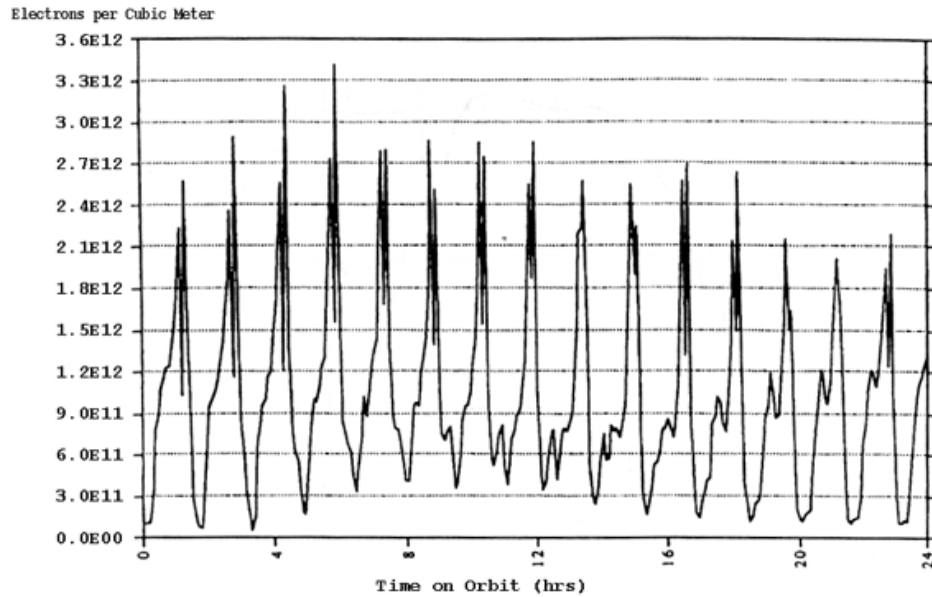
Figure 2.2 The Electron Density Profile Encountered by the ISS [2]

2.2.3 Plasma Interfaces. Normally, a circuit loop of conducting wire carrying a current will yield a net force of zero in a uniform magnetic field since the force on one side of the wire would be canceled by the opposite force on the other side. On the other hand, the current flow in the electrodynamic tether system is one-way and the circuit is closed trough the plasma and the force on the plasma does not affect the system. For this purpose, plasma contactors are used for the electron change between the plasma and the system. Up to date, hollow cathode emitters are used for electron emission and

2-5

conductive spheres are used for electron collection. However, more effective ways of electron transfer has been suggested. Field Emitter Array Cathodes for electron emission and Bare Tether electron collection are some of the new effective technologies suggested.

2.2.3.1 Hollow Cathode Emitters. The hollow cathode emitter uses some expendable gas for electron emission. The gas is heated and the electrons in the tether interact with heated gas to form an ion plasma. An electrode positively charged with respect to the system, draws the electrons and expels them to the space [10]. Since hollow cathodes establish a known vehicle ground reference potential with respect to the local plasma, they are considered to be safer for spacecraft systems. They also allow simple reversibility of the tether current for switching between power and thrust generation [1]. On the other hand, the major drawback to the hallow cathodes is the gas expenditure.

2.2.3.1 Field Emitter Array Cathodes (FEACs). A substitute for hollow cathode emitters are the Field-Emitter Array Cathodes (FEACs) that will enable propellantless propulsion with electrodynamic tethers. This new technology has potential applications in spacecraft electron emission and charge control. Instead of a gas consuming plasma contactor or a high-powered hot emitter, FEACs consist of many micron level cathode-gate pairs to perform cold field emission at relatively low voltages. While each individual cathode emits only micro-amp level currents, an array of these cathodes printed on a semiconductor wafer is capable of emitting amp/cm^2 current densities. FEACs offer relative low power, simple to integrate and cheap technique of electron emission for electrodynamic tethers [11].

2.2.3.1 Bare Tether Electron Collection. An alternative way of collecting electrons from the ambient plasma is using the tether itself which is suitable for

electrodynamic altitude boosting applications. The lower portion of the conductive tether is left uninsulated and functions as a very efficient anode. The tether is biased positively with respect to the plasma and collects the electrons from the plasma. The small cross-sectional area of the tether makes it a much more efficient collector in terms of the electron collection per unit area than a large conductive sphere. Eliminating the need for large massive and high-drag sphere or a resource-using plasma contactor at the upper and of the tether, the bare electron collection system reduces cost, complexity and gravity center shift in the system. Another benefit of the bare tether is its self-adjusting nature to the electron density changes. This is achieved by the natural expansion of the biased tether portion when the density drops [1].

Choiniere et al [12], suggests that injection of radio frequency (RF) power along tether can enhance electron collection by creating a time periodic field distribution by the RF excitation which results in the scattering off the electrons from their usual orbital motion limited trajectories. Their studies showed that large current enhancements could occur at some resonance frequencies but this enhancements costs high RF power. This could improve the limitation of electron density on electrodynamic power and thrust generation.

2.2.4 Applications of EDT Propulsion. An application of electrodynamic tether propulsion that is feasible in the near term, boost and de-boost of spacecraft, has been studied by several researchers [3, 4, 5] and will be demonstrated in the NASA ProSEDS mission. A satellite servicing vehicle and satellite tug are other possible applications of EDT propulsion systems. To extend the lifetime of expensive space assets, a satellite servicing vehicle could rendezvous with multiple satellites in different orbits for refueling

or replacing components. An orbital tug vehicle could change the orbits of LEO satellites [13].

2.3 Research Efforts

There have been a large number of studies on electrodynamic tethers. Most of these studies are focused on specific missions. The most likely implementation of electrodynamics in the near future is probably the de-orbiting spacecraft or upper stages of launch vehicles. The application that seems to have the second highest probability is orbit maintenance and altitude boosting. Therefore, most studies are focused on the experimental and future missions, their performance analysis and system configurations. A focus of many researchers' interest is the orbital maintenance of the International Space Station (ISS). The use of electrodynamic tethers to boost up the ISS could trim $2 billion [14] a year off the operating cost.

The study presented by Caroll [15] is significant in terms of this thesis study's scope and approach. His study includes the basic current laws for orbital maneuvering with electrodynamic tethers by controlling the tether current. Another study, presented by West et al [16], examines orbital maneuvering strategies for small satellites in low earth orbit. Electrodynamic tether propulsion and orbital maneuvering is also suggested by Tethers Unlimited Inc. in their "The μPET Propulsion System[TM]" project, but there is not any technical paper published about this project yet.

Brief information about some of the related studies and the studies mentioned above will be presented in the following pages.

2.3.1 EDT Propulsion Missions. The Tethered Satellite System Reflight (TSS-1R). The TSS-1R mission, on STS-75 in 1996 [1], is the second flight of the TSS hardware, which first flew on STS-46 in 1992. The goals of this mission were to explore space plasma-electrodynamic processes and the orbital mechanics of a gravity gradient stabilized system of two satellites linked by a conductive tether. The STS-75 mission flew in a circular orbit at 300 km altitude in a 28.5° inclination. The deployed tether length reached nearly 20 km before the tether broke because of an insulation flaw [17].

The results of the TSS-1R mission showed that current extraction from the ionosphere was extremely efficient and the current collected by the satellite at different voltages exceeded the levels predicted by the numerical models up to three times. The data collected throughout the experiment is encouraging for applications of electrodynamic tethers such as electrical power and thrust generation and the use of tethers as VLF/ULF antennas [17].

Propulsive Small Expendable Deployer System (ProSEDS) The ProSEDS mission is a flight demonstration of an electrodynamic tether propulsion system with the concepts and technology derived from the TSS missions results [14]. The system will fly aboard a Delta II rocket and the satellite will be the second stage of the rocket placed into a 400 km circular orbit. During this mission scheduled for June 2002, ProSEDS will deploy a 5 km bare conducting tether with a 10 km non-conducting tether at the far end to put enough distance on the tether so it stays taut. The ProSEDS experiment will demonstrate the ability of an electrodynamic tethered satellite to generate significant thrust without the use of any propellant by de-orbiting the Delta II second stage [14, 18].

The Terminator Tether[TM] Tethers Unlimited Inc., is currently developing the Terminator Tether system in a Small Business Innovation Research agreement with the Marshall Space Flight Center for de-orbiting satellites from low earth orbit.

The suggested system is a lightweight, low-cost device that will use electrodynamic drag generated by a conducting bare tether. Using their simulations, they found out that the system with a tether length of 5-10 km could utilize some of the generated power on the tether to drive its own circuitry without severely affecting the de-orbit rate therefore could be autonomous, independent of the host spacecraft. They also suggest that a tether device massing 2% of the host spacecraft can de-orbit an upper stage from 400 km in under two weeks and a mid-LEO satellite from 850 km in under three months. They developed a feedback-control scheme for dynamically stabilizing the tether [4].

Reboosting of International Space Station There are a number of studies on reboosting the International Space Station (ISS) to make up for the orbital energy loss due to aerodynamic drag. Furnishing the ISS with an electrodynamic tether propulsion system could dramatically reduce its dependency on rocket fuel and refueling missions, drastically lowering the operating costs and relaxing the risk of resupply mission failure.

For this purpose, NASA has designed a tether capable of generating 0.5-0.8 N of thrust using less than 10 kW of ISS's power. The proposed tether would consist of 10 km long aluminum ribbon with the dimensions of 0.6 mm by 10 mm in cross section [10].

The Microsatellite Propellantless Electrodynamic Tether Propulsion System[TM] Another system currently being developed by Tethers Unlimited Inc. is called Microsatellite Propellantless Electrodynamic Tether Propulsion System that will provide

propulsion to microsatellites with the use of electrodynamic tethers. The company claims that the system would provide long-duration boost, deboost, inclination change and stationkeeping propulsion [19].

2.3.2 Previous Studies. As mentioned above, most of the studies on electrodynamic tethered systems are focused on specific missions which are limited to power generation and orbital boosting. Other studies embrace the issues related to the stability of the tether, plasma-tether interactions and feasibility of electrodynamic tether systems. However, relatively little research effort has been put into orbital maneuvering with electrodynamic tethers.

In the study, Damping in Rigid Electrodynamic Tethers on Inclined Orbits, Palaez et al. [20] examined the dynamic instability of the tether originated from the electrodynamic forces that should be accounted for in a long-term operation of an electrodynamic tether on a circular inclined orbit. They analyzed two simple devices, an additional mass sliding up and down on the tether (dashspot) and modulation of the tether length, that could be used to damp the instability which is characterized by a coupling between the in-plane and out-of-plane oscillations of the tether. Their results showed that there is a stability domain for the tether with the dashspot device and the tether length modulation technique is too complex to implement.

In a study presented by Yoshiki [21], the performance of an electrodynamic tether orbit transfer system for round trip missions between low earth orbits is analyzed. He evaluated both electrodynamic tether orbit transfer system and ion thruster orbit transfer system for demonstrative parameters, such as the mass, the electric power and the mission time. According to his results, there is a lower limit to the mission time for a

given altitude of arrival orbit and there is a higher limit to the altitude of arrival orbit for a given mission time. Major mechanical parameter affecting the performance of the system is the tether length. He also showed that the tether orbit transfer system has advantages in terms of mass, relative to the ion propulsion orbit transfer system.

Martinez-Sanchez et al. [22] presented a systems study of a 100 kW electrodynamic tether. Reviewing the scientific and engineering challenges of the design of electrodynamic power and propulsive tether systems up to 100 kW, they examined the performance and cost of the most important applications of such systems. The study includes electrodynamic tether system design issues and cost comparisons. The main propulsive applications considered in their study are drag compensation, orbital altitude changing and inclination change. They suggested the use of the current law that was also determined by Carroll [15] for the inclination change. The power applications investigated in their study are converting orbital energy or chemical propellant energy to electrical power and orbital energy storage for solar arrays. They suggest that power generation with rocket force make-up offers large fuel savings compared to fuel cells.

Gilchrist et. al. [8] reviewed a number of aspects of space electrodynamic tether propulsion technology. This study includes system-level issues associated with effective electrodynamic tether operation including ionospheric and motional variability of the power generation, and investigation of the tether end contacts under varying ionospheric conditions. Brief information of electron collecting and emitting devices are also presented in this work. They summarized the characteristics and advantages of two relatively new technologies for space tether systems, which are bare tethers for electron collection and Field Emitter Array Cathodes. They also inspected potential low earth

orbit applications considered for electrodynamic tethers and the ProSEDS mission. Mentioning the encouraging propellantless nature of the electrodynamic tethers, the applications presented in their study are de-orbiting, orbit maintenance and reusable upper stage propulsion.

The use of electrodynamic tether propulsion for orbit maintaining, boosting and orbit maneuvering small satellites in low earth orbit was also studied by West et al. [16] in a recent study. They investigated the tether atmospheric drag and available current limited by the electron collection. They also discuss that altitude boosting with a constant electrodynamic tether thrust introduces changes in inclination and eccentricity, but a way of compensating these effects is unanswered. This problem will be addressed later in this thesis.

The Guidebook For Analysis of Tether Applications by Carroll [15] is a comprehensive study on tethers. In addition to a significant research on tether dynamics and tether materials, the basic principles of electrodynamic tethers and electrodynamic libration control issues are presented. Carroll determined the current law for each orbital element that describes the tether current behavior required to change the particular orbital element. Carroll also pointed out the electrodynamic tether stability issues related to the electrodynamic forces and suggested tether design and current modulation for the control of the librations.

2.3.3 Low Thrust Propulsion. Electrodynamic tether propulsion has similarities with conventional low propulsion systems, such as ion propulsion. Both have the relatively small thrust levels, long mission times, and continuous operation of the system.

2-13

However, the guidance schemes and optimization algorithms developed for low-propulsion systems by researchers are based on controlling the thrust vector direction. In electrodynamic propulsion, the thrust vector is determined by the electromagnetic field lines and the tether direction. Controlling the tether stabilized in a desired direction is an issue not solved yet. The best option for stabilizing the tether is around the local vertical with the help of gravity gradient torque.

The guidance schemes and optimization algorithms studied by other researchers are not directly applicable to electrodynamic tether propulsion concept and further research effort may be spent for adapting these approaches to electrodynamic tether propulsion. This study is focused on providing a simple guidance scheme with minimal computational load. Therefore, further optimization of the guidance scheme is out of this study's scope and will not be discussed. Some of the works of researches on low thrust propulsion orbit maneuvering are next presented.

Gaylor [23] analyzed low thrust orbit transfers starting with the perturbation equations and developed a simple closed loop guidance law. The maneuvers considered by Gaylor are orbit circularization and inclination change. One of the two control laws used to circularize the orbit is a discontinuous thrust scheme and the thruster is fired on perigee and apogee centered arcs of the orbit. This control law is similar to the current control law determined by Caroll [15] for eccentricity change for electrodynamic tether propulsion. Gaylor determined the thrust angles required for the maneuvers and analyzed propellant usage.

Dewell et al. [24] examined dynamic optimization problems from the point of view of genetic search and applied genetic search techniques for optimizing orbital

trajectories involving low-thrust orbit transfer. They suggested that genetic search methods may be appropriate solution for nonsmooth problems and poor initial guess conditions. The problem was parameterized in terms of thrust vector direction in three axes and genetic search techniques are discussed to solve the problem.

Kluever [25] also discusses developing a simple guidance scheme for low thrust orbit transfers by using the perturbation equations. He optimized the thrust vector angles by taking the derivative of time rate of change orbital elements, semi-major axis, eccentricity and inclination, with respect to thrust vector angles and solving for the angles by letting the derivatives equal zero. These optimal control laws are then weighed by a quadric function to get the weighting functions for each orbital element. However, Kluever does not determine the weighting parameters.

Herbiniere et al. [26] describes a low thrust positioning strategy that can be used for low earth orbit constellation deployment in quasi-circular orbits. Pointing modes for thrust vector are considered to find thrust vector direction parameters to correct eccentricity and inclination errors under spacecraft's pointing constraints. The longitude of ascending node and phase rendezvous are left to be performed during the initial drift phase of the spacecraft.

III. Methodology

3.1 Modeling the Earth's Magnetic Field

As mentioned in Chapter 2, Earth's magnetic field can be modeled with the magnetic field of a sphere uniformly magnetized in the direction of a dipole axis. In this model, the dipole axis goes through the center of the Earth, and is offset from the rotational axis by 11.3°. This simple approximation can lead to errors as great as 30% in some locations. On the other hand, the error can be reduced to 10% by displacing the dipole axis about 400 km towards the western Pacific from the center of the Earth [9].

1 : Dipole Axis
2 : Geomagnetic Equator
3 : Earth's Rotation Axis
4 : Equator
θ : 11.3°
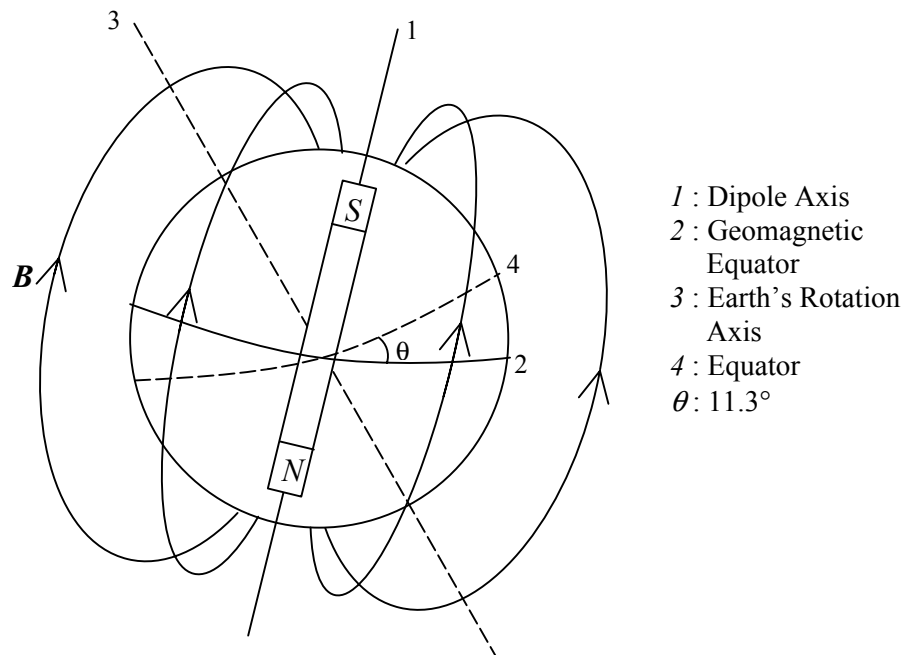
Figure 3.1 Earth's Magnetic Field Dipole Model

To keep the equations simple, the dipole axis is assumed to have zero offset from the Earth's center (Figure 3.1) and the dipole axis is assumed to be inertially fixed. For the dipole model, the geomagnetic induction vector $\vec{B}$ is [4];

$$\vec{B} = \frac{\mu_m}{r^3}\left[\hat{e}_M - 3\langle\hat{e}_M,\hat{e}_R\rangle\hat{e}_R\right] \qquad (3.1)$$

where μ_m is the magnetic moment of Earth's dipole, $\hat{e}_M$ the unit vector of the dipole axis, $\hat{e}_R = \vec{r}/r$ the unit radial vector, r the distance from Earth's center and $\langle\,,\rangle$ is the scalar product. The magnetic moment μ_m is 8.0711×10^6 Tesla/km^3 [4].

For convenience, a modified orbital element set is used where inclination, i, and the argument of ascending node, Ω, are measured with respect to the magnetic equatorial plane instead of geographic equatorial plane. In that case, the unit vector of the dipole axis can be written in an inertial frame aligned with dipole axis and magnetic equatorial plane as;

$$\hat{e}_M = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}_{ijk} \qquad (3.2)$$

3.2 Thrust

The force, $\vec{F}$, on a charged particle, Q, moving in a magnetic field, $\vec{B}$, with the velocity, $\vec{v}$ is [27]

$$\vec{F} = Q(\vec{v}\times\vec{B}) \qquad (3.3)$$

This equation can be written in differential form as

$$d\vec{F} = dQ(\vec{v}\times\vec{B}) \qquad (3.4)$$

In the case of a current carrying conductor in an external magnetic field, the force exerted on electrons are transferred to the conductor they are confined to. The velocity of an electron moving through the electrodynamic tether can be written in terms of tether velocity, $\vec{v}_t$, and electron velocity relative to the tether, $\vec{v}_e$

$$\vec{v} = \vec{v}_t + \vec{v}_e \tag{3.5}$$

Substituting Equation 3.5 into Equation 3.4, Equation 3.4 becomes,

$$d\vec{F} = dQ(\vec{v}_t \times \vec{B}) + dQ(\vec{v}_e \times \vec{B}) \tag{3.6}$$

Since, current is $I = dQ/dt$,

$$dQ = I\,dt \tag{3.7}$$

As the net charge on the tether is zero, the number of ions is equal to the number of electrons and the net force created by tether velocity, $\vec{v}_t$, integrates to zero. Omitting the first term in Equation 3.6 and substituting Equation 3.7 into Equation 3.6,

$$d\vec{F} = (Idt)(\vec{v}_e \times \vec{B}) \tag{3.8}$$

$dl = \vec{v}_e dt$ is the elementary length in the direction of the current, I.

Assuming that the tether is straight and magnetic field along the tether is constant Equation 3.8 might be integrated to give

$$\vec{F} = I(\vec{L} \times \vec{B}) \tag{3.9}$$

where L is the length of the tether. The tether is assumed to be perfectly aligned with the local vertical, therefore the vector for the tether current is

$$\vec{L} = \frac{\vec{r}}{|\vec{r}|}L = \hat{e}_R L \tag{3.10}$$

Substituting Equation 3.2 and Equation 3.10 into Equation 3.9, the force generated by the current in the electromagnetic tether is

$$\vec{F} = \frac{IL\mu_m}{r^3}(\hat{e}_R \times \hat{e}_M) \tag{3.11}$$

The perturbation equations for Classical Orbital elements will be in the form expressed in the orbital frame. The orthogonal unit vectors for this frame are the unit radial vector, $\hat{r}$, the unit vector tangential to unit radial vector in the orbital plane, $\hat{\theta}$, and $\hat{\omega}$ unit vector perpendicular to these two and normal to the orbital plane. In order to substitute Equation 3.11 into perturbation equations for orbital elements, the unit vector for the dipole axis, $\hat{e}_M$, is expressed in the orbital frame. The rotation matrix from inertial frame, F^i, to orbital frame, F^o, is

$$R^{io} = \begin{bmatrix} c_{v+\omega}c_\Omega - s_{v+\omega}c_i s_\Omega & c_{v+\omega}s_\Omega + s_{v+\omega}c_i c_\Omega & s_{v+\omega}s_i \\ -s_{v+\omega}c_\Omega - c_{v+\omega}c_i s_\Omega & -s_{v+\omega}s_\Omega + c_{v+\omega}c_i c_\Omega & c_{v+\omega}s_i \\ s_i s_\Omega & -s_i c_\Omega & c_i \end{bmatrix} \tag{3.12}$$

where v is the true anomaly, ω is the argument of perigee and

$$s_x = \sin x$$
$$c_x = \cos x$$

Then unit vector for the dipole axis, $\hat{e}_M$, expressed in the orbital frame becomes

$$\hat{e}_M = R^{io}\hat{e}_{M\,ijk} = \begin{bmatrix} \sin(v+\omega)\sin i \\ \cos(v+\omega)\sin i \\ \cos i \end{bmatrix}_{r\theta v} \tag{3.13}$$

Combining Equation 3.11 and Equation 3.13, the propulsive tether force is written in the orbital frame as,

$$\vec{F} = \frac{IL\mu_m}{r^3} \begin{bmatrix} 0 \\ -\cos i \\ \cos(\nu + \omega)\sin i \end{bmatrix} \qquad (3.14)$$

Looking at the above equation (Equation 3.14), some observations can be made immediately. Since the force is perpendicular to the current and therefore to the tether, there is no radial component of the force. Once again, it should be remembered that the tether is assumed to be perfectly aligned with the local vertical. This assumption is reasonable, because without the applied current, there are no major forces other than gravity and gravity gradient forces stabilizes the tether around the local vertical. Secondly, the force is perpendicular to the magnetic field lines; thus out-of-plane forces cannot be attained when the orbit is coplanar with the geomagnetic equator ($i = 0°$), similarly in-plane forces cannot be attained when the orbit is "polar" with respect to the magnetic field ($i = 90°$).

3.3 Perturbation Equations

The orbital maneuvers are carried out by performing the desired changes in Classical Orbital Elements (COEs). In order to determine the form of the guidance law, the general perturbation equations [28] governing the evolution of the orbital elements are examined. The applied current is required to be in the form to have a secular change in the desired orbital element. The general perturbation equations that express the time rate of change in the COEs are as follows;

$$\frac{da}{dt} = \frac{2e\sin\nu}{n\sqrt{1-e^2}}a_r + \frac{2a\sqrt{1-e^2}}{nr}a_\theta$$

$$\frac{de}{dt} = \frac{\sqrt{1-e^2}\,\sin\nu}{na}a_r + \frac{\sqrt{1-e^2}}{na^2e}\left[\frac{a^2(1-e^2)}{r} - r\right]a_\theta$$

$$\frac{di}{dt} = \frac{r\cos(\nu+\omega)}{na^2\sqrt{1-e^2}}a_w$$

$$\frac{d\Omega}{dt} = \frac{r\sin(\nu+\omega)}{na^2\sqrt{1-e^2}\,\sin i}a_w$$

$$\frac{d\omega}{dt} = -\frac{\sqrt{1-e^2}\,\cos\nu}{nae}a_r + \frac{p}{eh}\sin\nu\left(1 + \frac{1}{1+e\cos\nu}\right)a_\theta - \frac{r\cot i\sin(\nu+\omega)}{na^2\sqrt{1-e^2}}a_w$$

$$\frac{dM}{dt} = n - \frac{1}{na}\left(\frac{2r}{a} - \frac{(1-e^2)}{e}\cos\nu\right)a_r - \frac{(1-e^2)}{nae}\sin\nu\left(1 + \frac{r}{a(1-e^2)}\right)a_\theta \qquad (3.15)$$

where

a : Semi-major axis

e : Eccentricity

i : Inclination

Ω : Argument of ascending node

ω : Argument of perigee

M : Mean anomaly

a_r : Acceleration in the $\hat{r}$ direction, along radial vector

a_w : Acceleration in the $\hat{\omega}$ direction, normal to orbit plane

a_θ : Acceleration in the $\hat{\theta}$ direction, perpendicular to $\hat{r}$ and $\hat{\omega}$

$n = \sqrt{\mu/a^3}$: Orbital mean motion

$p = a(1 - e^2)$: Semilatus rectum

$h = \sqrt{\mu p}$: Angular momentum per unit mass of satellite

The acceleration created by the tether force on the total satellite mass, m, can be interpreted as the perturbing acceleration. Dividing Equation 3.14 by the total mass yields

$$\vec{a} = \frac{IL\mu_m}{mr^3} \begin{bmatrix} 0 \\ -\cos i \\ \cos(\nu + \omega)\sin i \end{bmatrix} \tag{3.16}$$

Substituting the components of this perturbing acceleration into the general perturbation equations, the equations that govern the evolution of the orbital elements for a known current can be obtained:

$$\frac{da}{dt} = -\frac{L\mu_m}{m} \frac{2a\sqrt{1-e^2}}{nr^4} I \cos i$$

$$\frac{de}{dt} = -\frac{L\mu_m}{m} \frac{\sqrt{1-e^2}}{na^2 er^4} \left[a^2(1-e^2) - r^2 \right] I \cos i$$

$$\frac{di}{dt} = \frac{L\mu_m}{m} \frac{1}{na^2 \sqrt{1-e^2} r^2} I \cos^2(\nu + \omega)\sin i$$

$$\frac{d\Omega}{dt} = \frac{L\mu_m}{m} \frac{1}{na^2 \sqrt{1-e^2} r^2} I \sin(\nu + \omega)\cos(\nu + \omega)$$

$$\frac{d\omega}{dt} = -\frac{L\mu_m}{m} \frac{\sqrt{a(1-e^2)}}{ne\sqrt{\mu} r^3} I \sin \nu \cos i \left(1 + \frac{1}{1 + e\cos \nu} \right) -$$

$$-\frac{L\mu_m}{m} \frac{1}{na^2 \sqrt{1-e^2} r^2} I \sin(\nu + \omega)\cos(\nu + \omega)\cos i \tag{3.17}$$

For short periods of time, the radial distance to the spacecraft from the Earth's center, r, is assumed to be Keplarian and can be written as

$$r = \frac{a(1-e^2)}{1+e\cos\nu} \qquad (3.18)$$

Substituting this expression into the general perturbation equations and collecting some of the terms the following expressions can be obtained for the change in orbital elements for any given current

$$\frac{da}{dt} = -\frac{2L\mu_m \cos i}{nma^3(1-e^2)^{3.5}} I(1+e\cos\nu)^4$$

$$\frac{de}{dt} = -\frac{L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} I(2e\cos\nu + e^2\cos\nu + e^2)(1+e\cos\nu)^2$$

$$\frac{di}{dt} = \frac{L\mu_m \sin i}{nma^4(1-e^2)^{2.5}} I\cos^2(\nu+\omega)(1+e\cos\nu)^2$$

$$\frac{d\Omega}{dt} = \frac{L\mu_m}{nma^4(1-e^2)^{2.5}} I\cos(\nu+\omega)\sin(\nu+\omega)(1+e\cos\nu)^2$$

$$\frac{d\omega}{dt} = -\frac{L\mu_m \cos i}{nma^4(1-e^2)^{2.5}} I(1+e\cos\nu)^2[(2+e\cos\nu)\sin\nu + e\sin(\nu+\omega)\cos(\nu+\omega)] \quad (3.19)$$

3.4 Validity of the Equations

The validity of the above equations can be verified by numerically integrating the equations. Therefore, two simulations are setup. The first simulation directly integrates the Equations 3.19. The second simulation uses the two-body equation and numerically integrates in Cartesian coordinates. The vector equation for the second simulation is

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_p \qquad (3.20)$$

where μ is the Earth's gravitational constant and $\vec{a}_p = \vec{F}_{tether}/m$ is the acceleration produced by the tether force on the total system mass, m. The corresponding scalar equations of motion in Cartesian coordinates are

$$\ddot{x} = -\frac{\mu}{r^3}x + a_{p,x}$$

$$\ddot{y} = -\frac{\mu}{r^3}y + a_{p,y}$$

$$\ddot{z} = -\frac{\mu}{r^3}z + a_{p,z} \qquad (3.21)$$

The results of the two simulations are consistent hence proving the validity of the perturbation equations combined with the tether force. The source codes in Fortran can be found in Appendix A. Both simulations use a Runge Kutta 5/6 variable step integration with a tolerance of 1×10^{-10}.

These equations (Equations 3.19) can be integrated over time for a particular current to determine changes in orbital elements. The form of the current law is determined next.

3.5 Current Law

To achieve the desired change in the orbital elements, the current in the tether is modulated as the electrodynamic tether orbits the Earth, since the tether force is directly proportional to the current. The first step in developing the current law is to find out the form of current that will result secular changes in the orbital elements. This can be done by examining the perturbation equations (Equations 3.19) and determining the current

wave that will yield an integral with a secular term. As a first approximation, the orbital elements (a,e,i,Ω,ω) will be assumed constant since they vary slowly. Consequently, the changes in the orbital elements over a given a period of time are

$$\Delta a = -\frac{2L\mu_m \cos i}{nma^3(1-e^2)^{3.5}}\int_{t_0}^{t_f} I(1+e\cos v)^4 dt$$

$$\Delta e = -\frac{L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}}\int_{t_0}^{t_f} I(2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 dt$$

$$\Delta i = \frac{L\mu_m \sin i}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I\cos^2(v+\omega)(1+e\cos v)^2 dt$$

$$\Delta\Omega = \frac{L\mu_m}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I\cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 dt \qquad (3.22)$$

$$\Delta\omega = -\frac{L\mu_m \cos i}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I(1+e\cos v)^2\left[(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)\right]dt$$

Eliminating the higher order terms in e (which are small enough to be negligible for the LEO maneuvers applicable to this concept) these equations can be written in the following simpler form hence making it easier to spot the dominant terms,

$$\Delta a = -\frac{2L\mu_m \cos i}{nma^3(1-e^2)^{3.5}}\int_{t_0}^{t_f} I(1+4e\cos v)dt$$

$$\Delta e = -\frac{2L\mu_m \cos i}{nma^4(1-e^2)^{3.5}}\int_{t_0}^{t_f} I\cos v dt$$

$$\Delta i = \frac{L\mu_m \sin i}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I\cos^2(v+\omega)(1+2e\cos v)dt$$

$$\Delta\Omega = \frac{L\mu_m}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I\cos(v+\omega)\sin(v+\omega)(1+2e\cos v)dt$$

$$\Delta\omega = -\frac{L\mu_m \cos i}{nma^4(1-e^2)^{2.5}}\int_{t_0}^{t_f} I\left[2\sin v + 5e\cos v\sin v + e\sin(v+\omega)\cos(v+\omega)\right]dt \qquad (3.23)$$

It is desired to find a current form for each orbital element that will yield a secular change. The expression for the changes in each orbital element will be examined analytically and through a numerical simulation. The simulation in Cartesian coordinates mentioned in part 3.4 will be used for all of the orbital elements with the following parameters;

Table 3.1 Simulation Parameters

Total Satellite Mass		1000 kg
Tether Length		15 km
Applied Current		5 A
Time of Flight		20 Orbits
Initial Orbital Elements	Semi-major Axis	8235 km
	Eccentricity	0.0285
	Inclination	5 deg
	Argument of Ascending Node	60 deg
	Argument of Perigee	60 deg
	Mean Anomaly	0
	Orbital Period	2 hrs. 3 min. 57 sec

3.5.1 Semi-Major Axis. The expression for the change in semi-major axis (Equation 3.24) is the only expression with a constant term in the integral. The cosine term under the integral will average out through the orbit period and will not make a significant contribution to the change in the semi-major axis.

$$\Delta a = -\frac{2L\mu_m \cos i}{nma^3(1-e^2)^{3.5}} \int_{t_0}^{t_f} I(1+4e\cos v)dt \qquad (3.24)$$

Therefore, applying a constant tether current will yield a secular change in the semi-major axis (Figure 3.2). Applying this constant tether current, DC, following results can be obtained using the simulation (Figures 3.2-3.6)

DC current has the desired effect on semi-major axis while not affecting argument of perigee and argument of ascending node. However, the higher order terms in e in the expressions for eccentricity (Equations 3.22) contribute secular terms to the integral. Likewise, DC current also causes a secular change in inclination because of the $\cos^2(\nu + \omega)$ term inside the integral.
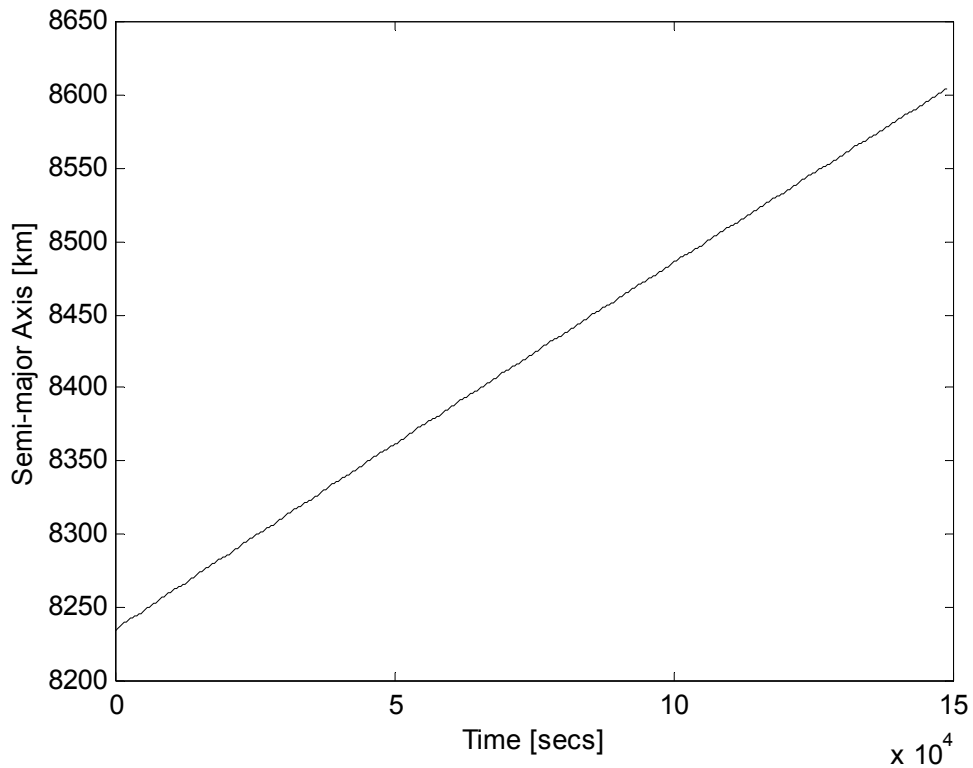
Figure 3.2 Change in Semi-major Axis due to DC Current
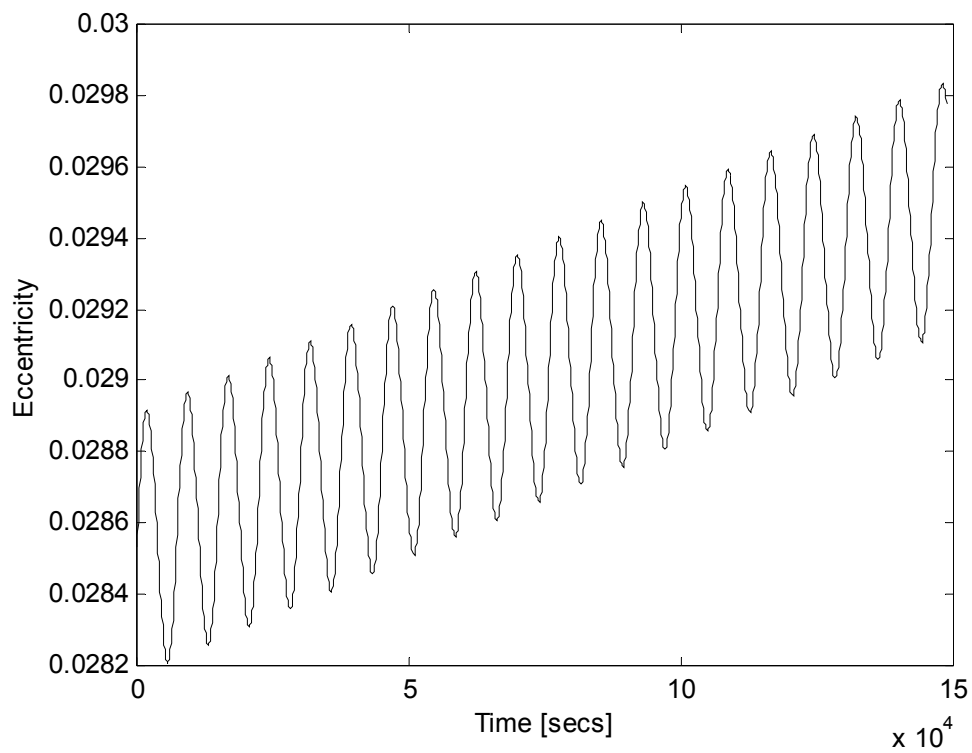
Figure 3.3 Change in Eccentricity due to DC Current
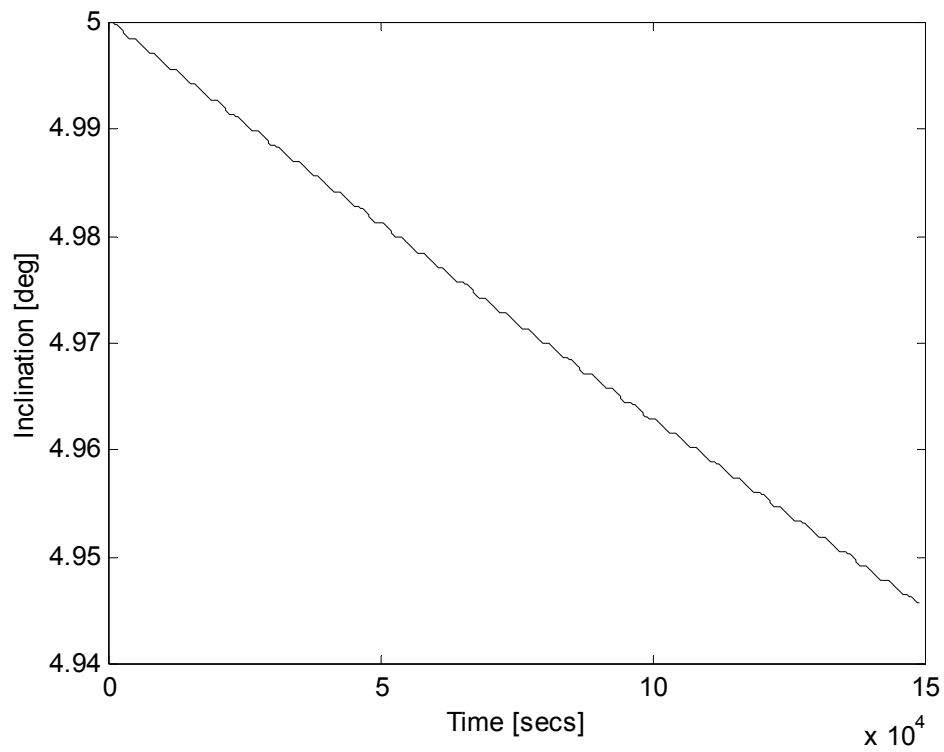
Figure 3.4 Change in Inclination due to DC Current
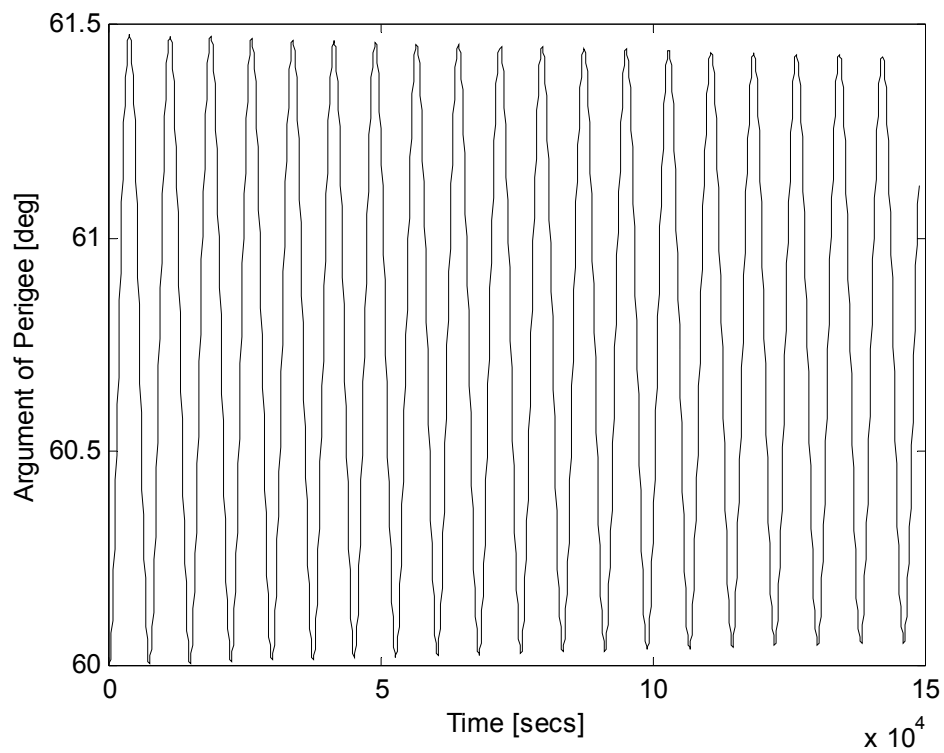
Figure 3.5 Change in Argument of Perigee due to DC Current
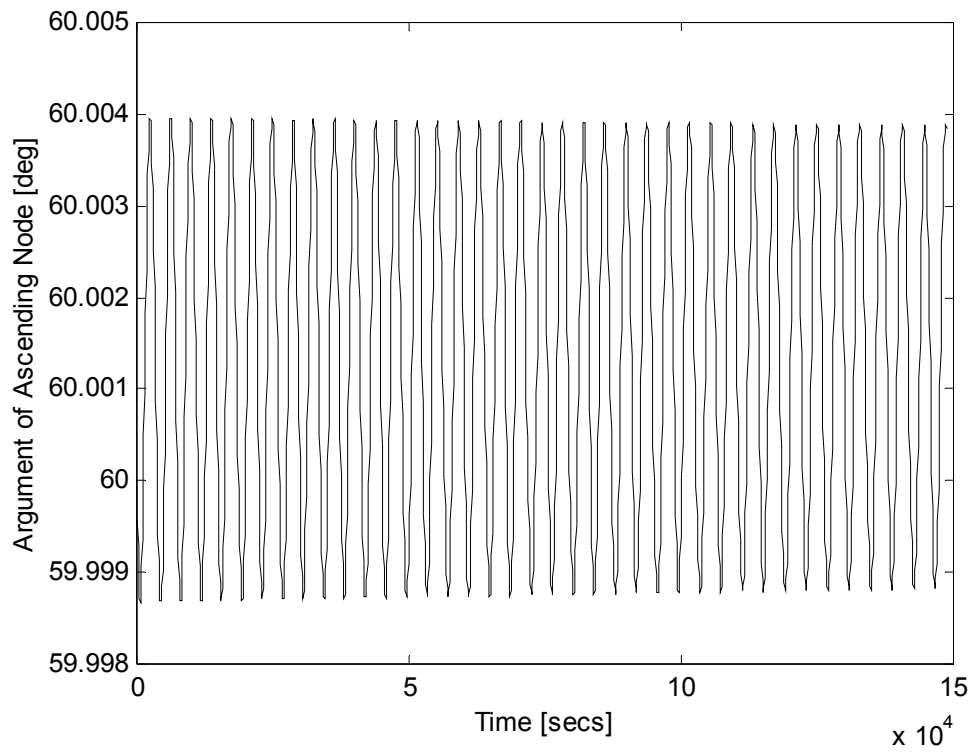
Figure 3.6 Change in Argument of Ascending Node due to DC Current

3.5.2 Eccentricity. Similarly, examining the simplified expression for the change in eccentricity

$$\Delta e = -\frac{2L\mu_m \cos i}{nma^4(1-e^2)^{3.5}} \int_{t_0}^{t_f} I \cos \nu \, dt \qquad (3.25)$$

it is observed that $\cos \nu$ term will average out in time. A current law of

$$I = I' \cos \nu \qquad (3.26)$$

can be applied to attain a $\cos^2 \nu$ term in the integral that will yield a secular change in eccentricity. The results of the simulation show that $\cos \nu$ current has the major affect on eccentricity (Figure 3.7) and there is a coupling with semi-major axis. When multiplied by this current law, the odd powers of $\cos \nu$ terms in the expression for semi-major axis (Equations 3.22) yield even powers of $\cos \nu$ to produce a secular change in semi-major axis (Figure 3.8). The effect of $\cos \nu$ current on inclination, argument of perigee and argument of ascending node averages out over a period (Figures 3.9-3.11).
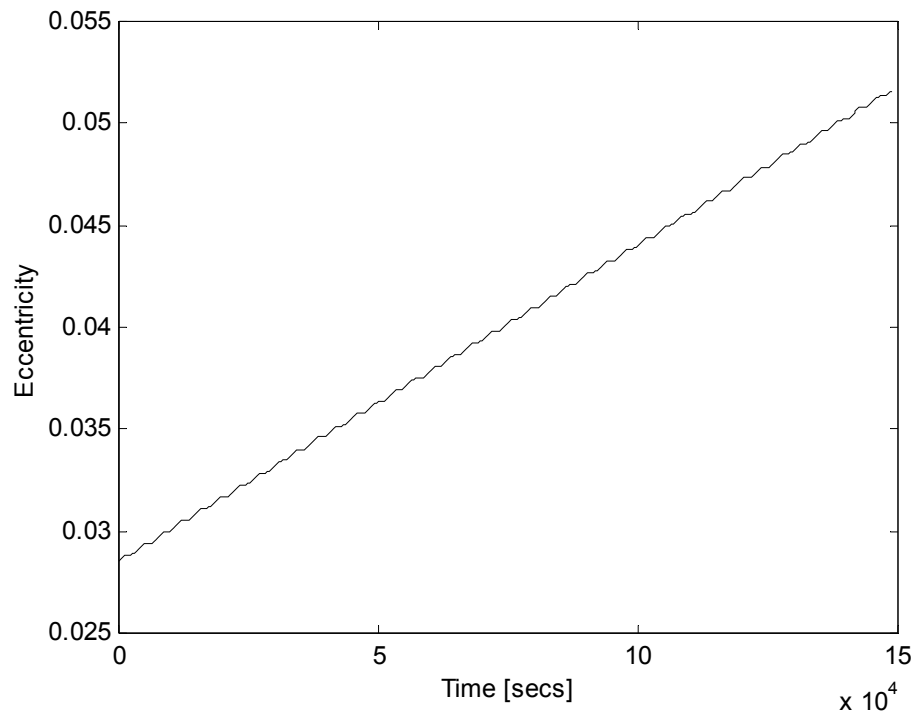
Figure 3.7 Change in Eccentricity due to $\cos\nu$ Current
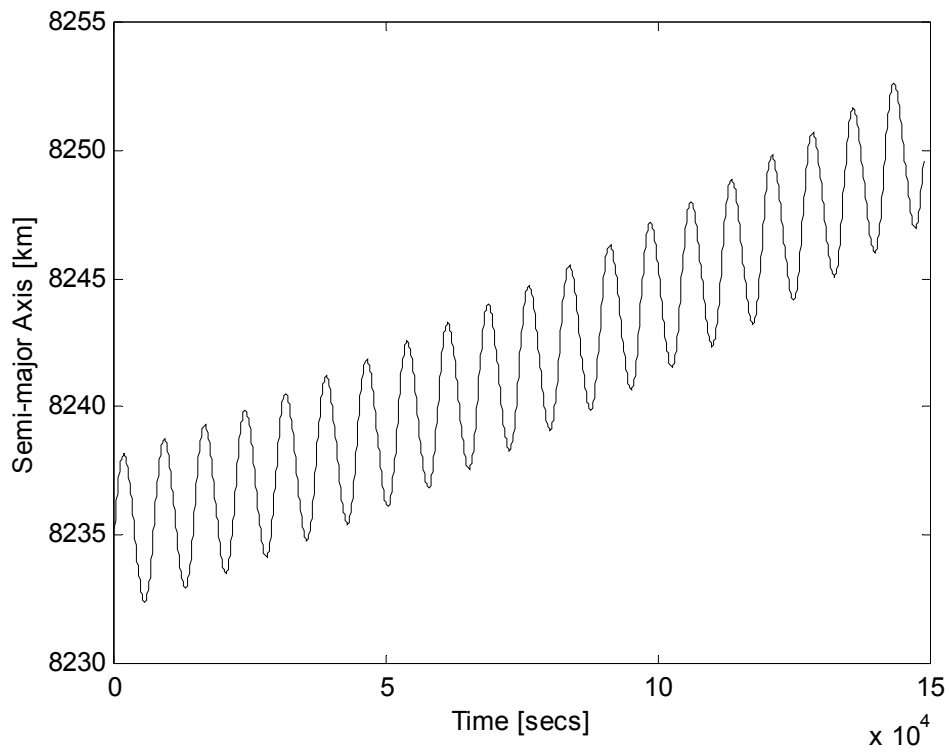
Figure 3.8 Change in Semi-major Axis due to $\cos\nu$ Current

3-16
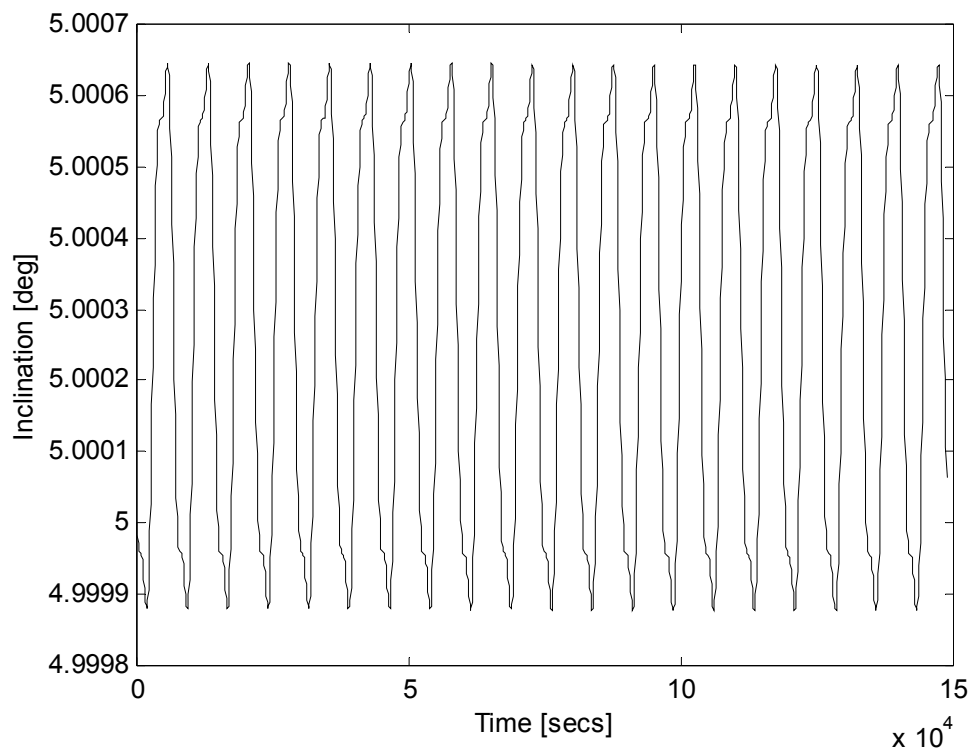
Figure 3.9 Change in Inclination due to $\cos \nu$ Current
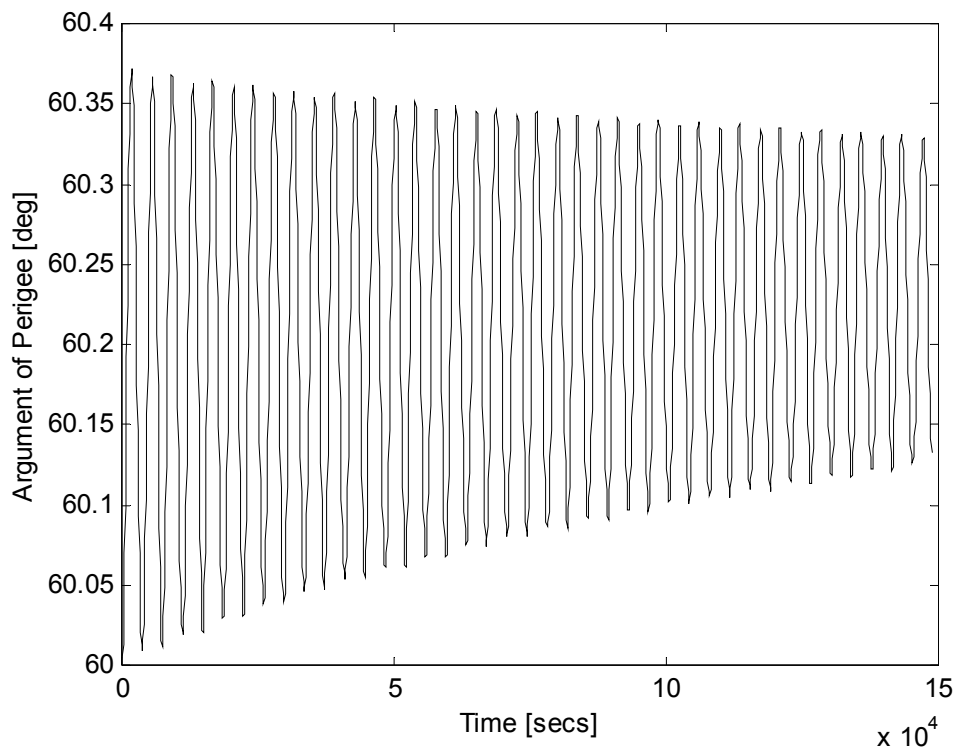
Figure 3.10 Change in Argument of Perigee due to $\cos \nu$ Current
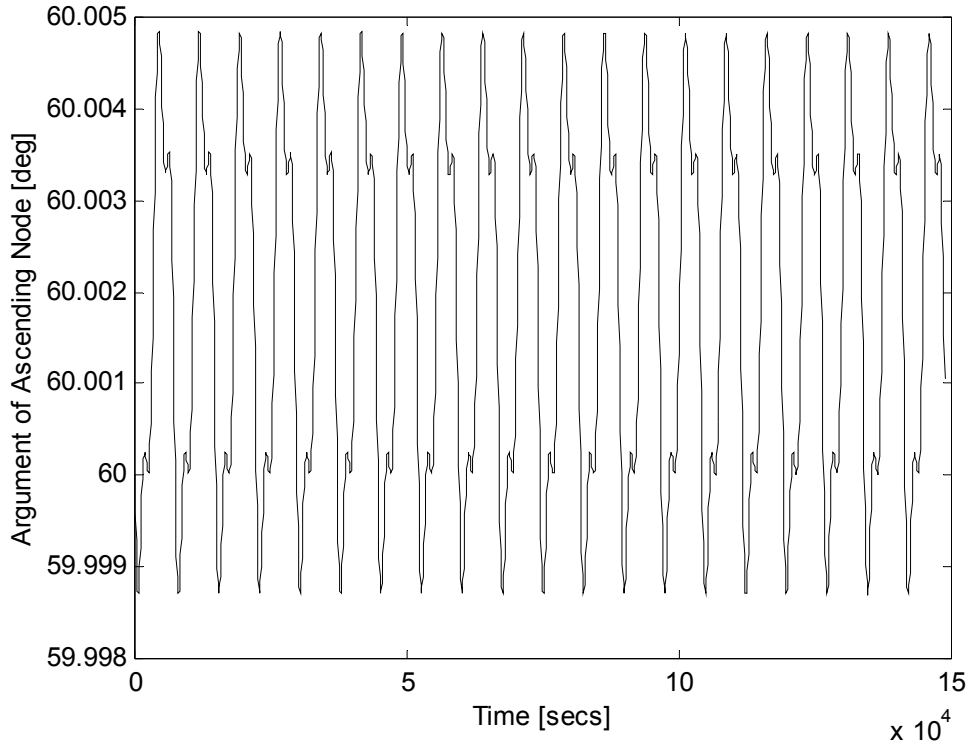
Figure 3.11 Change in Argument of Ascending Node due to $\cos \nu$ Current

3.5.3 Inclination. Looking at the simplified expression for the change in inclination

$$\Delta i = \frac{L\mu_m \sin i}{nma^4(1-e^2)^{2.5}} \int_{t_0}^{t_f} I \cos^2(\nu+\omega)(1+2e\cos\nu)dt \qquad (3.27)$$

and from the results of the DC current, it is seen that DC current has a major effect on the inclination as well as semi-major axis. However, another current form that has relatively smaller effects on the other orbital elements is required. Furthermore, the impact on the inclination by the DC current is desired to be compensated.

Proceeding with a similar approach to the above,

$$I = I'\cos[2(\nu+\omega)] \qquad (3.28)$$

form of tether current will provide the secular change in inclination (Figure 3.12) while

not introducing any secular terms to the semi-major axis (Figure 3.13). With the help of

the trigonometric expansion, ($\cos 2\alpha = 2\cos^2\alpha - 1$), $\cos^2(v+\omega)$ term in Equation 3.28

can be written in terms of $\cos[2(v+\omega)]$. The multiplication of the current law and this

term gives $\cos^2[2(v+\omega)]$ term that provides the secular change in the integral.

Again, higher order e terms in the expression for eccentricity yield a minor

secular effect on eccentricity (Figure 3.14). While the double angle terms inside the

brackets in the expression for the change in argument of perigee result in a secular

change (Figure 3.15), the argument of ascending node is not affected (Figure 3.16).
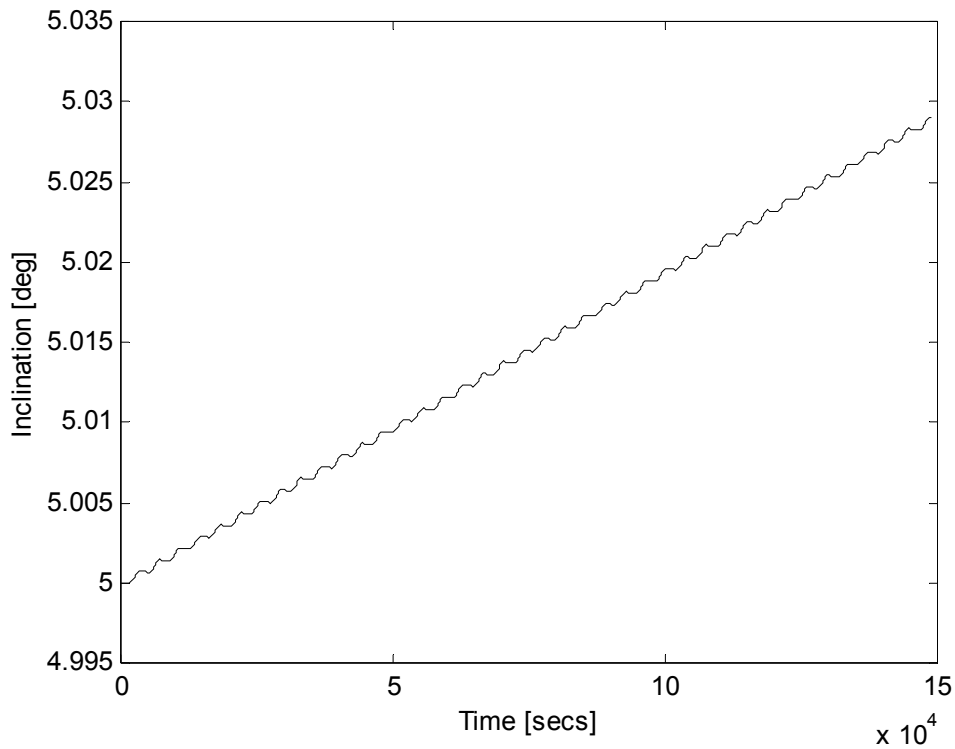
Figure 3.12 Change in Inclination due to $\cos[2(v+\omega)]$ Current

3-19

Figure 3.13 Change in Semi-major Axis due to $\cos[2(\nu+\omega)]$ Current
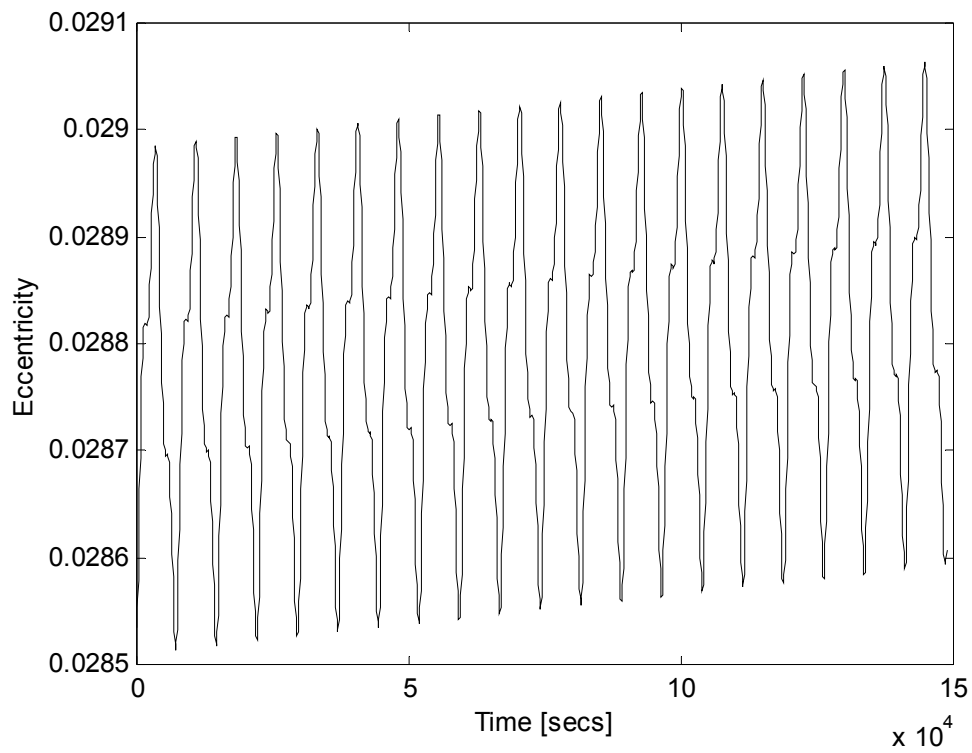
Figure 3.14 Change in Eccentricity due to $\cos[2(\nu+\omega)]$ Current
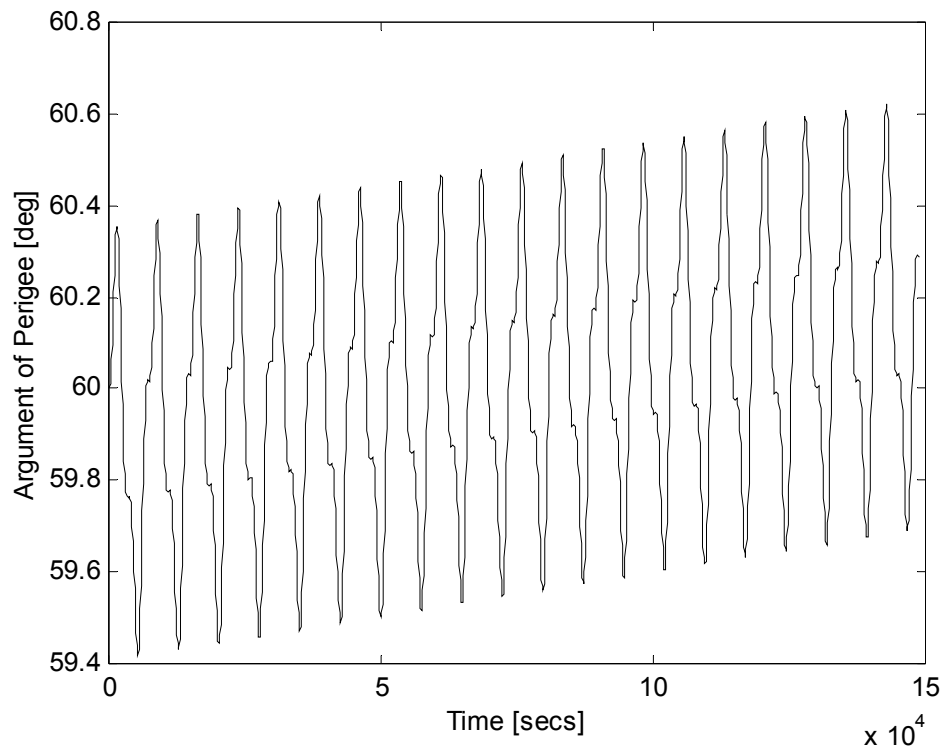
Figure 3.15 Change in Argument of Perigee due to $\cos[2(\nu + \omega)]$ Current
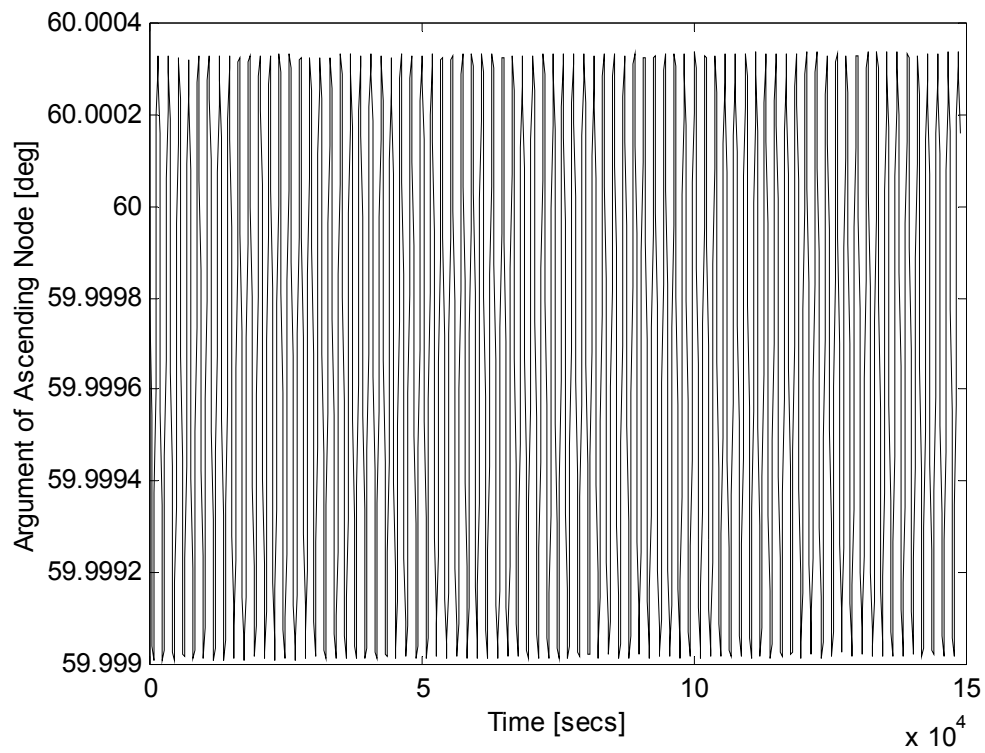
Figure 3.16 Change in Argument of Ascending Node due to $\cos[2(\nu + \omega)]$ Current

3-21

3.5.4 Argument of Ascending Node. Likewise, the current law of

$$I = I' \sin\left[2(v + \omega)\right] \qquad\qquad (3.29)$$

has the same effect on argument of the ascending node. Similar to the expression for

inclination, $\cos(v + \omega)\sin(v + \omega)$ term in the expression for argument of ascending node

in Equations 3.23 can be written in terms of $\sin\left[2(v + \omega)\right]$ since

$$\sin 2\alpha = 2 \sin\alpha \cos\alpha$$

Multiplying this term with the current law, $\sin^2\left[2(v + \omega)\right]$ term can be acquired. This

term has the secular effect on the integral (Figure 3.17).
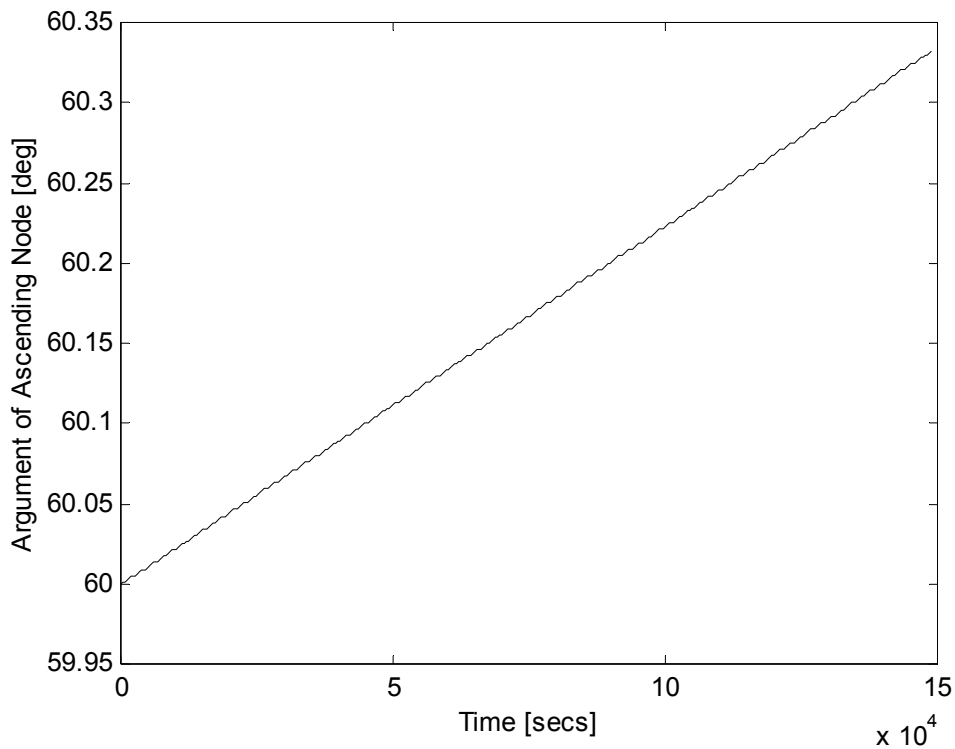
Figure 3.17 Change in Argument of Ascending Node due to $\sin\left[2(v + \omega)\right]$ Current

Figure 3.18 Change in Semi-major Axis due to $\sin[2(v + \omega)]$ Current

Figure 3.19 Change in Eccentricity due to $\sin[2(v + \omega)]$ Current

3-23

Figure 3.20 Change in Inclination due to $\sin[2(\nu + \omega)]$ Current
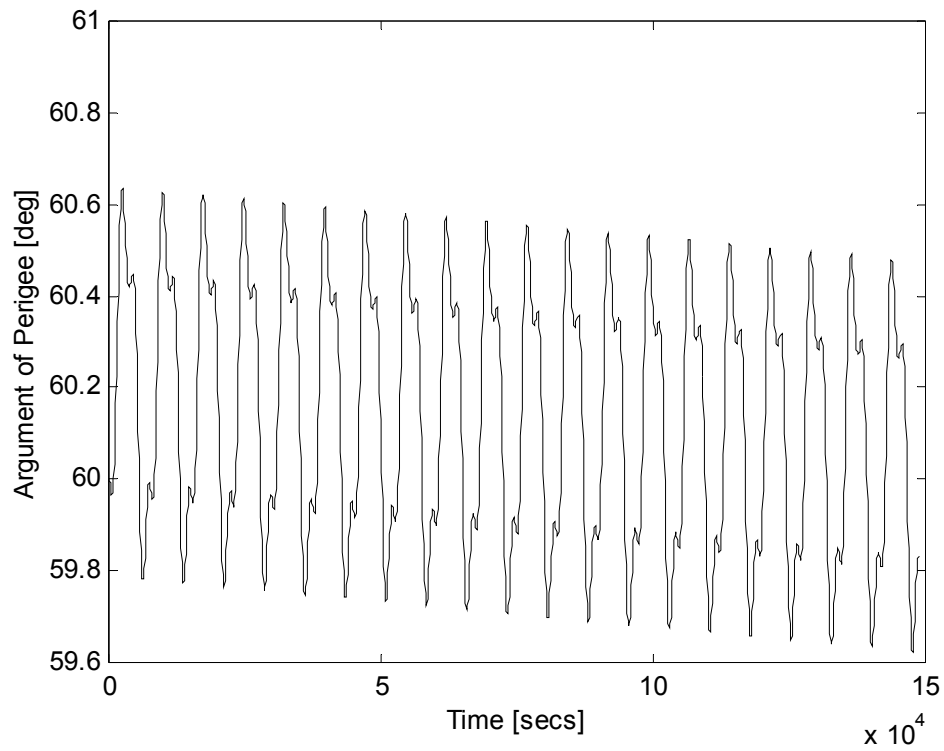
Figure 3.21 Change in Argument of Perigee due to $\sin[2(\nu + \omega)]$ Current

3-24

A negative drift occurs in the eccentricity and argument of perigee. The effect on the eccentricity is caused by the $e^2 \cos^2 \nu$ terms that appear after the trigonometric expansion of the expression (Figure 3.19). On the other hand, the drift in the argument of perigee (Figure 3.21) is caused by $e \sin(\nu + \omega) \cos(\nu + \omega)$ term inside the brackets under the integral in the expression for the argument of perigee (Equations 3.22). The effects of this current law on semi-major axis and inclination are averaged out through the orbit.

3.5.5 Argument of Perigee. Finally, the current law

$$I = I' \sin \nu \qquad\qquad (3.30)$$

multiplies with the dominant $2 \sin \nu$ term in the expression for argument of perigee to produce a secular change. Fortunately, no significant coupling occurs with the other orbital elements. This can be seen clearly in the results of the simulation. All the changes in the other orbital elements are periodic over the orbit period and average out (Figure 3.23-3.26) and there is a significant change in the argument of perigee providing the desired affect (Figure 3.22).
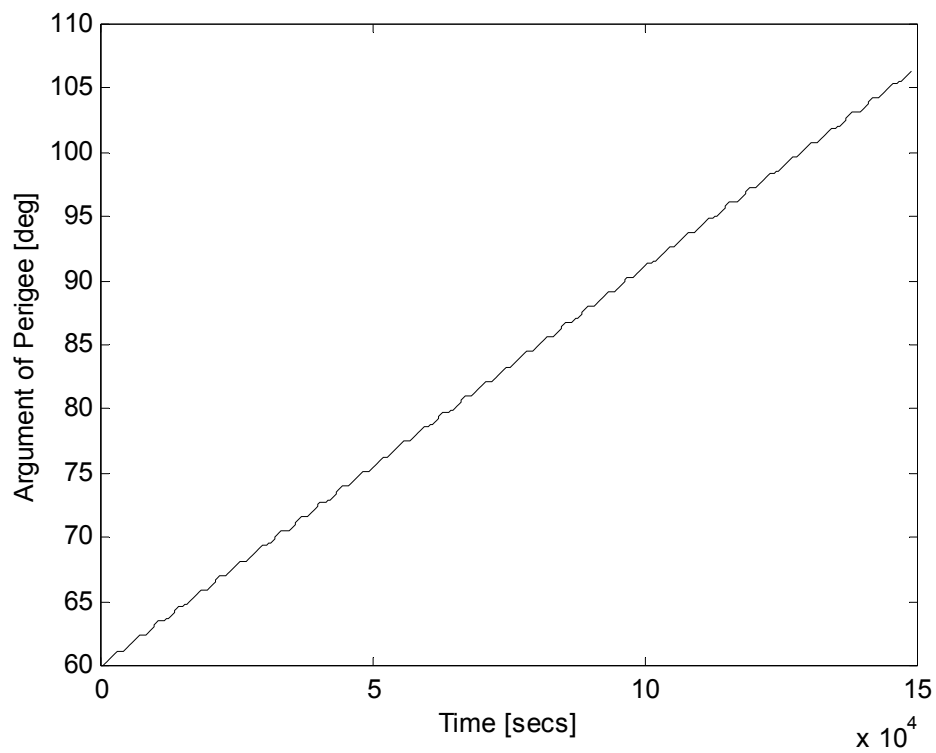
Figure 3.22 Change in Argument of Perigee due to $\sin\nu$ Current
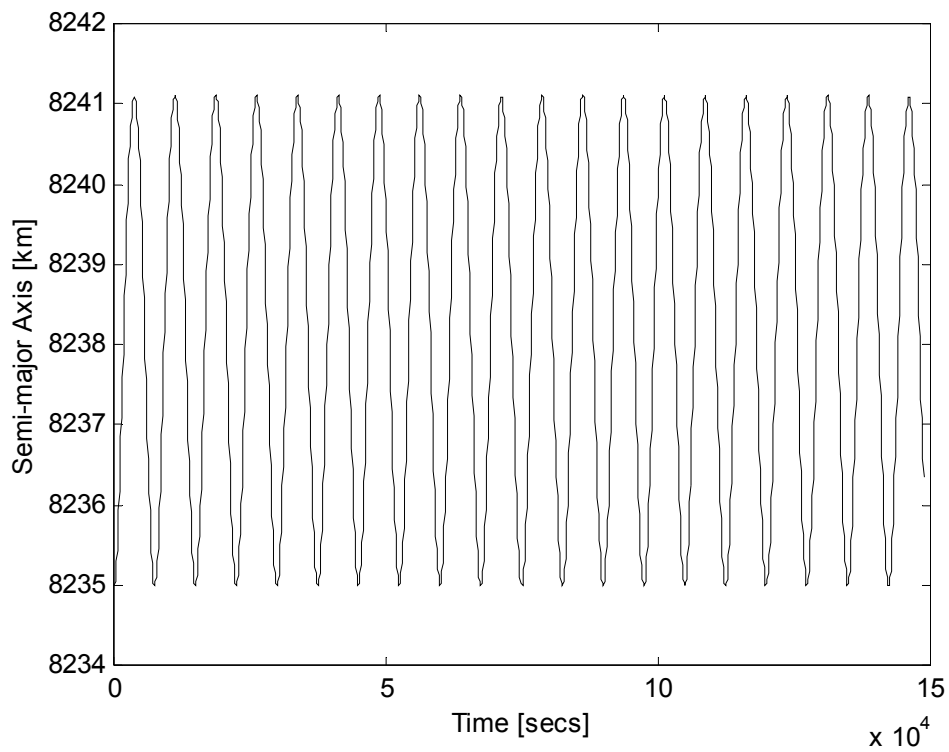
Figure 3.23 Change in Semi-major Axis due to $\sin\nu$ Current
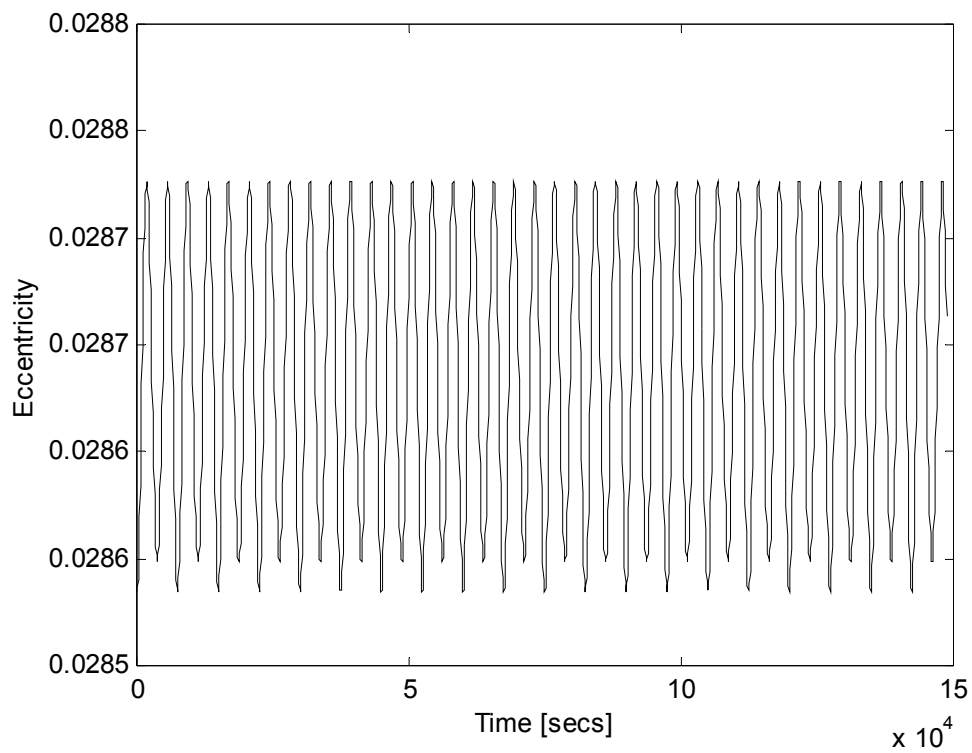
Figure 3.24 Change in Eccentricity due to $\sin \nu$ Current
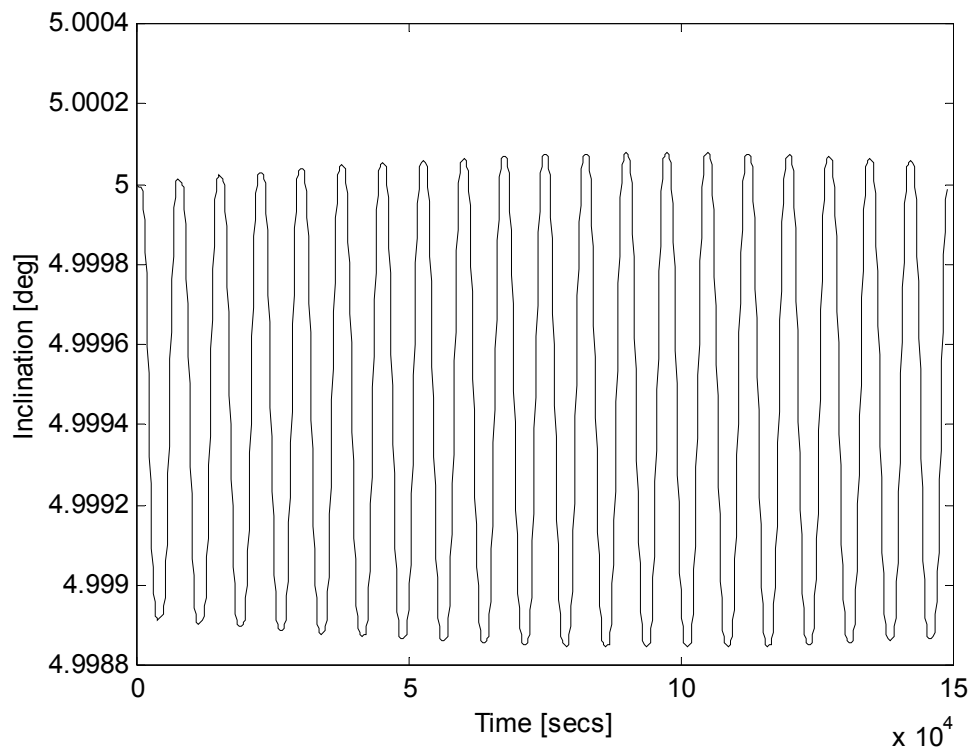
Figure 3.25 Change in Inclination due to $\sin \nu$ Current

3-27
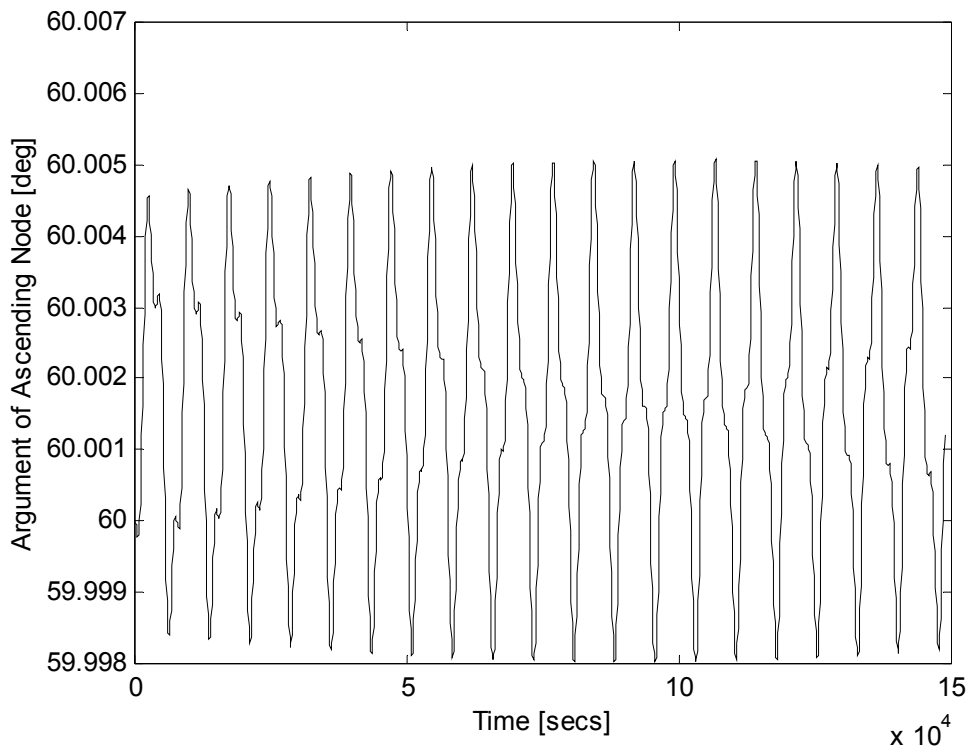
Figure 3.26 Change in Argument of Ascending Node due to $\sin \nu$ Current

To sum up, the current laws and their corresponding orbital elements are:

Current Law		Orbital Element
- DC		Semi-major Axis
- $\cos \nu$	*Provides a positive*	Eccentricity
- $\sin \nu$	*secular change in*	Argument of Perigee
$\cos[2(\nu + \omega)]$	$\Rightarrow$	Inclination
$\sin[2(\nu + \omega)]$		Argument of Ascending Node

The above relations agree with those presented by Caroll [15]. The current form suggested by Caroll for phase change, ΔM, is a saw tooth wave, however saw tooth wave does not give satisfactory results and affects the other elements dramatically.

3-28

As expressed above there are couplings between the various current laws and the orbital elements. An applied current in the above form not only changes the desired orbital element but also have an undesired effect on the other elements. The non-dominant terms in the integrals couple with other current laws to create a secular affect. Furthermore, these current laws are derived based on the assumption that the orbital elements are constant. A combination of these current laws will result in undesired effects because the constant approximation in this assumption begins to fail, as there are changes in multiple orbital elements. However, a combination of all five current laws can compensate the undesired effects in the results and minimize the error to a reasonable level.

The next step in developing the current control law is to combine the five current laws investigated above to perform simultaneous changes in the orbital elements and to reduce the effects of the couplings. Therefore, a superposition principle is assumed. Explicitly, a linear combination of the five independent currents laws is employed in order to execute an orbital maneuver between any two sets of five orbital elements. So the final form of the control law is

$$I = I_{avail}\left(X_1 + X_2\cos\nu + X_3\sin\nu - X_4\cos[2(\nu+\omega)] - X_5\sin[2(\nu+\omega)]\right) \qquad (3.31)$$

where I_{avail} is the available current and the X_i terms are the current coefficients that are to be calculated to maneuver the vehicle.

The final step is to construct a guidance scheme that includes time of flight for the maneuver and the above current control law.

3.6 Guidance Scheme

The guidance scheme must determine the current coefficients, X_i, and the time of flight for a given maximum available current and a set of changes in the orbital elements. The secular changes in the orbital elements are found by substituting the current control law (Equation 3.31) into the perturbation equations (Equation 3.21) and taking an averaged integral over one period of the orbit. The orbital elements are assumed to be constant except for the true anomaly, which is approximated by the mean anomaly,

$$v = nt \tag{3.32}$$

This yields five equations linear in the current coefficients, which can be written in the following matrix form:

$$[A] t \, \vec{X} \, I_{avail} = \vec{\Delta} \tag{3.33}$$

where t is the time of flight, $\vec{\Delta}$ is the set of changes in the orbital elements,

$$\vec{\Delta} = \begin{bmatrix} \Delta a \\ \Delta e \\ \Delta \omega \\ \Delta i \\ \Delta \Omega \end{bmatrix} \tag{3.34}$$

and $\vec{X}$ is the vector for the current coefficients,

$$\vec{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} \tag{3.35}$$

For the angular quantities, maneuvers greater than 180 degrees are performed by going the other way around (i.e. subtracting the absolute value from 360 and multiplying by the opposite sign.)

The A_{ij} element of the matrix in Equation 3.33 is computed by the averaged integral of perturbation equation for the orbital element that corresponds to ith element of the $\vec{\Delta}$ vector with the current law that corresponds to jth element in the $\vec{X}$ vector. The elements of the constant $[A]$ matrix are;

$$A_{11} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^3 (1-e^2)^{3.5}} \int_0^P (1+e\cos v)^4 \, dt$$

$$A_{12} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^3 (1-e^2)^{3.5}} \int_0^P \underline{\cos v} (1+e\cos v)^4 \, dt$$

$$A_{13} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^3 (1-e^2)^{3.5}} \int_0^P \underline{\sin v} (1+e\cos v)^4 \, dt$$

$$A_{14} = \frac{1}{P} \frac{2L\mu_m \cos i}{nma^3 (1-e^2)^{3.5}} \int_0^P \underline{\cos[2(v+\omega)]} (1+e\cos v)^4 \, dt$$

$$A_{15} = \frac{1}{P} \frac{2L\mu_m \cos i}{nma^3 (1-e^2)^{3.5}} \int_0^P \underline{\sin[2(v+\omega)]} (1+e\cos v)^4 \, dt$$

$$A_{21} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} \int_0^P (2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 \, dt$$

$$A_{22} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} \int_0^P \underline{\cos v} (2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 \, dt$$

$$A_{23} = -\frac{1}{P} \frac{2L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} \int_0^P \underline{\sin v} (2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 \, dt$$

$$A_{24} = \frac{1}{P} \frac{2L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} \int_0^P \cos[2(v+\omega)](2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 \, dt$$

$$A_{25} = \frac{1}{P} \frac{2L\mu_m \cos i}{nma^4 e(1-e^2)^{3.5}} \int_0^P \sin[2(v+\omega)](2e\cos v + e^2 \cos v + e^2)(1+e\cos v)^2 \, dt$$

$$A_{31} = -\frac{1}{Pe} \frac{L\mu_m \cos i}{nma^4 (1-e^2)^{2.5}} \int_0^{Pe} (1+e\cos v)^2 [(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)] \, dt$$

$$A_{32} = -\frac{1}{Pe} \frac{L\mu_m \cos i}{nma^4 (1-e^2)^{2.5}} \int_0^{Pe} \cos v (1+e\cos v)^2 [(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)] \, dt$$

$$A_{33} = -\frac{1}{Pe} \frac{L\mu_m \cos i}{nma^4 (1-e^2)^{2.5}} \int_0^{Pe} \sin v (1+e\cos v)^2 [(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)] \, dt$$

$$A_{34} = \frac{1}{Pe} \frac{L\mu_m \cos i}{nma^4 (1-e^2)^{2.5}} \int_0^{Pe} \cos[(v+\omega)](1+e\cos v)^2 [(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)] \, dt$$

$$A_{35} = \frac{1}{Pe} \frac{L\mu_m \cos i}{nma^4 (1-e^2)^{2.5}} \int_0^{Pe} \sin[(v+\omega)](1+e\cos v)^2 [(2+e\cos v)\sin v + e\sin(v+\omega)\cos(v+\omega)] \, dt$$

$$A_{41} = \frac{1}{P} \frac{L\mu_m \sin i}{nma^4 (1-e^2)^{2.5}} \int_0^P \cos^2(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{42} = \frac{1}{P} \frac{L\mu_m \sin i}{nma^4 (1-e^2)^{2.5}} \int_0^P \cos v \cos^2(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{43} = \frac{1}{P} \frac{L\mu_m \sin i}{nma^4 (1-e^2)^{2.5}} \int_0^P \sin v \cos^2(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{44} = -\frac{1}{P} \frac{L\mu_m \sin i}{nma^4 (1-e^2)^{2.5}} \int_0^P \cos[2(v+\omega)]\cos^2(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{45} = -\frac{1}{P} \frac{L\mu_m \sin i}{nma^4 (1-e^2)^{2.5}} \int_0^P \underline{\sin[2(v+\omega)]} \cos^2(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{51} = \frac{1}{P} \frac{L\mu_m}{nma^4 (1-e^2)^{2.5}} \int_0^P \cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{52} = \frac{1}{P} \frac{L\mu_m}{nma^4 (1-e^2)^{2.5}} \int_0^P \underline{\cos v} \cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{53} = \frac{1}{P} \frac{L\mu_m}{nma^4 (1-e^2)^{2.5}} \int_0^P \underline{\sin v} \cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{54} = -\frac{1}{P} \frac{L\mu_m}{nma^4 (1-e^2)^{2.5}} \int_0^P \underline{\cos[2(v+\omega)]} \cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 \, dt$$

$$A_{55} = -\frac{1}{P} \frac{L\mu_m}{nma^4 (1-e^2)^{2.5}} \int_0^P \underline{\sin[2(v+\omega)]} \cos(v+\omega)\sin(v+\omega)(1+e\cos v)^2 \, dt \qquad (3.36)$$

where $P = 2\pi/n$ is the period of the orbit. The current term in each element is underlined. Computing the integrals and rearranging the terms, the final form of the $[A]$ matrix is obtained (Table 3.2, Column 2). Furthermore, higher order e terms in the matrix can be neglected and the matrix becomes simpler (Table 3.2, Column 3). The error introduced by the simplification is reasonable and will be presented in the next chapter. The constant, C, in the $[A]$ matrix is

$$C = \frac{L\mu_m}{8mna^4 (1-e^2)^{2.5}} \qquad (3.37)$$

Table 3.2 The Elements of the $[A]$ Matrix

	Elements of A Matrix	Simplified Elements
A_{11}	$-2Ca\cos i(8+3e^4+24e^2)/(1-e^2)$	$-16Ca\cos i/(1-e^2)$
A_{12}	$-8Cae\cos i(3e^2+4)/(1-e^2)$	$-32Cae\cos i/(1-e^2)$
A_{13}	0	0
A_{14}	$2Cae^2(e^2+6)[\cos(i-2\omega)+\cos(i+2\omega)]/(1-e^2)$	0
A_{15}	$8Cae^2\cos i(e^2+6)\sin\omega\cos\omega/(1-e^2)$	0
A_{21}	$-7Ce\cos i(e^2+4)$	$-28Ce\cos i$
A_{22}	$-4C\cos i(5e^2+2)$	$-8C\cos i$
A_{23}	0	0
A_{24}	$Ce(2e^2+5)[\cos(i-2\omega)+\cos(i+2\omega)]$	$5Ce[\cos(i-2\omega)+\cos(i+2\omega)]$
A_{25}	$4Ce\cos i(2e^2+5)\sin\omega\cos\omega$	$20Ce\cos i\sin\omega\cos\omega$
A_{31}	$-2Ce^2\cos i\sin\omega\cos\omega$	0
A_{32}	$-4Ce\cos i\sin\omega\cos\omega$	$-4Ce\cos i\sin\omega\cos\omega$
A_{33}	$-2C\cos i(4+e^2+2e^2\cos^2\omega)/e$	$-8C\cos i/e$
A_{34}	$-2C\cos i\sin\omega\cos\omega(e^2+10)$	$-20C\cos i\sin\omega\cos\omega$
A_{35}	$-2C\cos i(4-e^2\cos^2\omega-10\cos^2\omega)$	$-4C\cos i(1-5\cos^2\omega)$
A_{41}	$C\sin i(4+e^2+2e^2\cos^2\omega)$	$4C\sin i$
A_{42}	$2Ce\sin i(1+2\cos^2\omega)$	$2Ce\sin i(1+2\cos^2\omega)$
A_{43}	$-4Ce\sin i\sin\omega\cos\omega$	$-4Ce\sin i\sin\omega\cos\omega$
A_{44}	$-2C\sin i(1+e^2\cos^2\omega)$	$-2C\sin i$
A_{45}	$-2Ce^2\sin i\sin\omega\cos\omega$	0
A_{51}	$Ce^2\sin(2\omega)$	0
A_{52}	$2Ce\sin(2\omega)$	$2Ce\sin(2\omega)$
A_{53}	$2Ce\cos(2\omega)$	$2Ce\cos(2\omega)$
A_{54}	0	0
A_{55}	$-C(2+e^2)$	$-2C$

The A_{ii} diagonal terms are the dominant terms in the equation and most of the off-diagonal terms are multiplied by eccentricity. The equation is linear with respect to time due to averaging over an orbital period to retain the secular effects. This matrix equation consists of five equations for the five unknown current coefficients. The sixth unknown variable is the time of flight, TOF, and it is constrained by the maximum available current, I_{avail}, that can be run through the tether in the environment.

Given a set of desired orbital elements and the duration of the maneuver, TOF, Equation 3.33 can be solved for the current control law coefficients:

$$\vec{X} = \frac{1}{I_{avail}TOF} \left[A^{-1} \right] \vec{\Delta} \qquad (3.38)$$

For determining the duration of the maneuver, a desired value for the duration can be given. If the duration of the maneuver is too short, the maximum current required for the maneuver, I_{max}, will exceed the maximum available current, I_{avail}. In this case, the current law coefficients and the duration of flight are recalculated.

Since the change in the orbital elements is linearly dependent on the TOF, the current coefficients can be scaled down by a constant factor and the TOF scaled up by the inverse of that same factor with the final orbit remaining unchanged to first order. So letting,

$$\vec{X}_{new} = \vec{X}_{old} \frac{I_{avail}}{I_{max}}$$

$$TOF_{new} = TOF_{old} \frac{I_{max}}{I_{avail}} \qquad (3.39)$$

the current constraint is enforced while satisfying Equation 3.15 and obtaining the same change in the orbital elements.

The maximum current required to carry out the maneuver, I_{max}, can be calculated in three different ways. A numerical simulation of the maneuver would provide the most precise answer. Unfortunately, this approach is computationally cumbersome and time consuming.

The maximum current can be computed by finding the maximums of the current expression Equation 3.31. Analytical derivatives of the current expression with respect to true anomaly and argument of perigee can be set to zero to find the maximum points, then the expression can be evaluated for the points that lie in the range of the orbital maneuver and the end points of this range. Taking the derivatives of the current expression:

$$\frac{\partial \vec{I}}{\partial v} = I_{avail}\{- X_2 \sin v + X_3 \cos v + 2X_4 \sin[2(v+\omega)] - 2X_5 \cos[2(v+\omega)]\}$$

$$\frac{\partial \vec{I}}{\partial \omega} = I_{avail}\{2X_4 \sin[2(v+\omega)] - 2X_5 \cos[2(v+\omega)]\} \tag{3.40}$$

and setting these expressions to zero, a set of equations can be obtained to give out four sets of results. Letting,

$$a = I_{avail} X_3$$

$$b = I_{avail} X_2$$

$$c = I_{avail}\left[(X_2^2 - X_3^2)X_5\right] - 2X_3 X_2 X_4$$

$$d = I_{avail}\left[(X_2^2 - X_3^2)X_4\right] + 2X_3 X_2 X_5 \tag{3.41}$$

solutions to the equations are;

	v	ω
1	$\tan_2^{-1}(a,b)$	$0.5\tan_2^{-1}(c,d)$
2	$\tan_2^{-1}(a,b)$	$0.5\tan_2^{-1}(-c,-d)$
3	$\tan_2^{-1}(-a,-b)$	$0.5\tan_2^{-1}(c,d)$
4	$\tan_2^{-1}(-a,-b)$	$0.5\tan_2^{-1}(-c,-d)$

Finally, a crude method of enumeration of the currents for a range of true anomaly and argument of perigee can be used to evaluate the maximum current. The true anomaly is varied from $0°$ to $360°$ while the argument of perigee can range from the initial value to the target value of the maneuver. For the simulation of the maneuvers, this approach is used.

For maneuvers with a large change in the orbital elements, the first order approximation that the orbital elements are constant for the integration of Equation 3.19 is not valid. Unfortunately, an accurate solution of these integrals would need to be numerical, which would then make it impossible to develop analytic expressions to calculate the coefficients of the current law. Since the objective of this study is to produce computationally efficient and robust design for concept exploration, the simple approached presented in this chapter is taken.

However, to improve the accuracy of the guidance scheme, the maneuver can be broken into a number of segments. By calculating the control law several times during the maneuver, the guidance scheme has an opportunity to correct the errors introduced by the inexact integrals. This guidance scheme employed is depicted in Figure 3.27.

For each segment, the control law coefficients and the time of flight to perform the rest of the maneuver in a single step is calculated. After flying for the period of the segment, this procedure is repeated for the next segment.

```
┌─────────────────┐
│    Guess TOF    │
└─────────────────┘
         │
         ▼◄──────────────────────┐
┌─────────────────┐              │
│    Calculate    │              │
│   Control Law   │              │
└─────────────────┘              │
         │                       │
         ▼              j=j+1     │
┌─────────────────┐   TOF=TOF-TOFⱼ│
│ Calculate Iₘₐₓ  │              │
└─────────────────┘              │
         │                       │
         ▼                       │
┌─────────────────┐              │
│     Modify      │              │
│ Control and TOF │              │
│  to meet Iₐᵥₐᵢₗ │              │
└─────────────────┘              │
         │                       │
         ▼                       │
┌─────────────────┐              │
│  Fly jth segment│──────────────┘
└─────────────────┘
         │
         ▼
┌─────────────────┐
│  Finish after   │
│   nth segment   │
└─────────────────┘
```
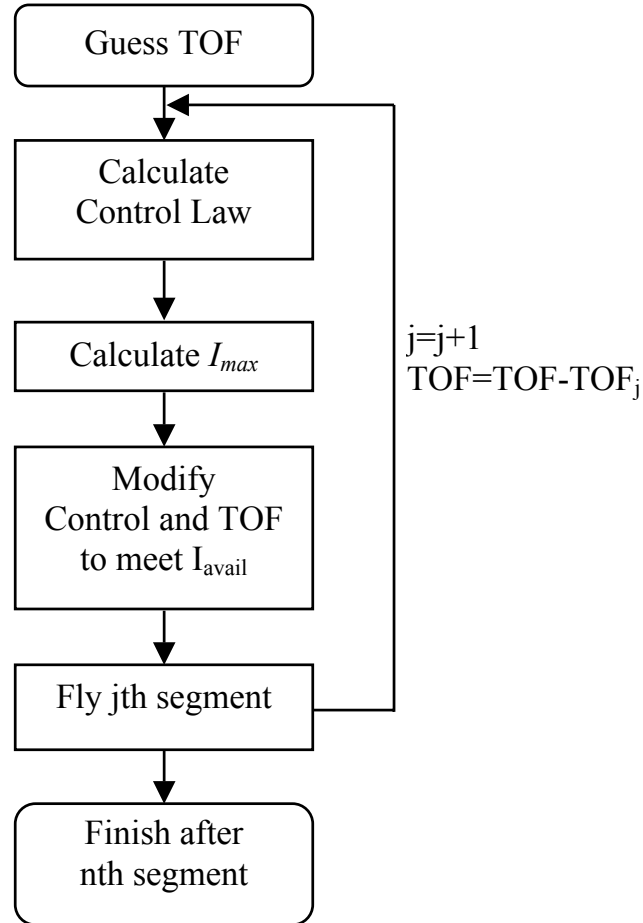
Figure 3.27 Guidance Scheme

3.7 Numerical Solution

An exact solution for the current control law can be determined by setting up the problem in a non-linear from and numerical non-linear equation solvers can be used. For

this purpose, the numerical non-linear equation solver in IMSL Math Libraries is used with Fortran. The equations for the problem are

$$F_1 = a_{t\arg et} - a_{final}$$

$$F_2 = e_{t\arg et} - e_{final}$$

$$F_3 = \omega_{t\arg et} - \omega_{final}$$

$$F_4 = i_{t\arg et} - i_{final}$$

$$F_5 = \Omega_{t\arg et} - \Omega_{final}$$

$$F_6 = I_{available} - I_{max} \qquad\qquad (3.42)$$

where $I_{available}$ is the available current to the system and I_{max} is the maximum current required for the maneuver. Basically, expressions F_1 through F_5 are the errors encountered at the end of the maneuver. All six equations are required to converge to zero for an exact answer.

A very good initial guess for the current law coefficients is essential to achieve the convergence for the solver because of the problem's complexity. Furthermore, the computational load of such a solver is relatively too large, since the maneuver must be simulated for each step of the solver.

The current law coefficients calculated by the control law determined in this study can be used as a good initial guess for such a solver.

IV. Analysis and Results

4.1 Simulation Tools

A set of computer programs has been developed and used to validate the guidance scheme for arbitrary maneuvers in LEO. These programs consist of a graphical user interface (GUI) for simulation inputs, a numerical simulation and a program for the visualization of the outputs. The communication between the programs is managed through intermediate files. Brief information about the tools is presented below.

Figure 4.1 Graphical User Interface

4.1.1 Graphical User Interface. A simple GUI for the simulation is developed using Visual Basic Programming Language 6.0. A screenshot of the user interface is given in Figure 4.1. This interface allows the user to input the required parameters, run the simulation and get the summarized outputs of the simulation. The required inputs for the simulation are the satellite parameters (total satellite mass, m, tether length, L, maximum available current, I_{avail}), Classical Orbital Elements (COEs) of the initial and the target orbit and the number of steps for the guidance scheme. The parameters can be entered into the associated fields in the interface or can be acquired from previously saved files.

4.1.2 Numerical Simulation. A numerical simulation is used to validate the current control law and the guidance scheme. The simulation is based on the two-body problem, which excludes the natural perturbations. The equation of motion for the electrodynamic tether system is

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \frac{1}{m}\vec{F} \qquad (4.1)$$

where $\vec{F}$ is the perturbing force due to the electrodynamics and m is the system mass (Equation 3.18). The corresponding scalar equations of motion (Equation 3.19) are integrated using a Runge Kutta 5/6 variable step integration with a tolerance of 1×10^{-10}.

The parameters for the satellite are read from a file (satellite.coe). The initial values and the target values of the COEs are read from separate files (initial.coe, target.coe) and the required change in the COEs are calculated. Following the guidance scheme described in Chapter 3, the satellite's motion is simulated by the numerical integration and the results are written in 5 different files with a hundred second sample

interval for each output. The results include position (position.2bp) and velocity (velocity.2bp) in kilometers in Cartesian coordinates relative to an earth centered inertial coordinate frame, the corresponding simulation time (time.2bp) in seconds, the history of the current used trough the maneuver (curhist.2bp) in amperes and the history of the thrust generated by the electrodynamic tether system (thist.2bp) in kilonewtons. Another set of results generated by the simulation consist of the current coefficients for each step of the guidance scheme (currents.txt) and the summary of the COEs, time elapsed and the error at the end of the simulation (output.txt).

This simulation is developed under DIGITAL Visual Fortran 5.0 and the IMSL Math Libraries are used for the numerical integration. The Fortran code of this simulation is given in the Appendix B.

4.1.3 Visualization. Another simple program running under MATLAB plots the results as the orbital trajectory and the change in the orbital elements. The outputs are read from the files created by the simulation and converted to orbital elements and presented in individual plots.

4.2 Orbital Maneuvering Examples

The validity and the accuracy of the guidance scheme are demonstrated with orbital maneuvering examples by the use of the above simulation tools. In addition to the examples, the affect of using the simplified A matrix (Table 3.2, Column 3) mentioned in Chapter 3 will be analyzed and presented.

4.2.1 Orbit Raising. The first example demonstrates an increase in the semi-major axis of a LEO satellite at an altitude of 800 km. The satellite is initially positioned in an orbit at this altitude with an inclination of 25 degrees with respect to the geomagnetic equator and an eccentricity of 0.02. The total mass of the satellite and the tether system is 1000 kg with a maximum available current of 5A and a tether length of 15 km [2]. The desired changes in the orbital elements are

$$\Delta a = 200 \text{ km}$$

$$\Delta e = \Delta \omega = \Delta i = \Delta \Omega = 0 \tag{4.2}$$

To achieve this maneuver, the control law and the time of flight are

$$I = -1.61 + 0.15 \cos \nu + 0.07 \sin \nu + 3.22 \cos[2(\nu + \omega)] - 0 \sin[2(\nu + \omega)]$$

$$TOF = 60 \text{ hr } 20 \text{ min} \tag{4.3}$$

There are two significant terms in this current control law. The DC current is the major component of the current law for semi-major axis change while the drift in inclination caused by the DC current is compensated by the $\cos[2(\nu + \omega)]$ current. To keep the inclination constant the $\cos[2(\nu + \omega)]$ current is considerably large.
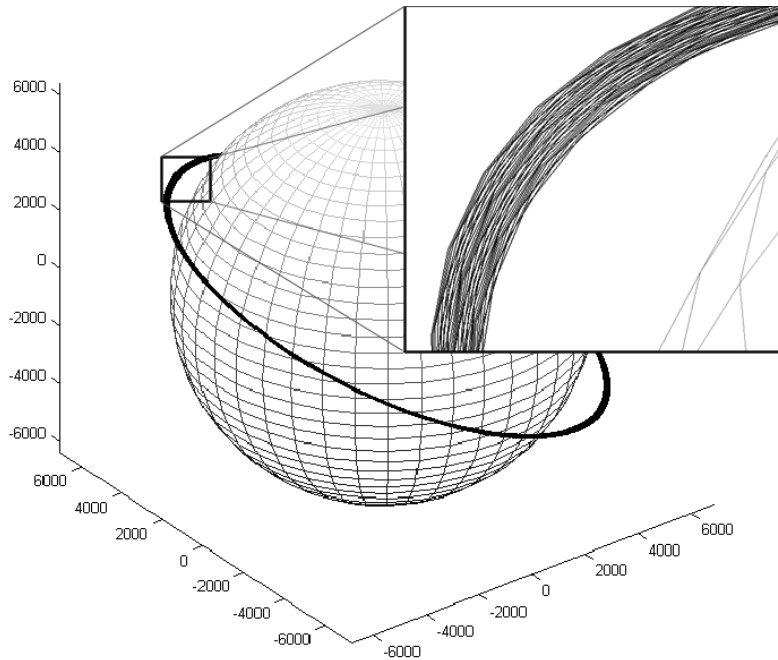
Figure 4.2 Orbital Trajectory for Orbit Raising

The simulation yields the results in Figures 4.2-4.6 for this control law. The semi-major axis exhibits the desired linear behavior (Figure 4.3). In Figure 4.4, the current shows the -1.61A DC offset with a 3.22A sinusoidal oscillation. The period of the oscillation is approximately half the orbital period, so this oscillation cycles through many oscillations. The current behavior for the first 2 orbits is shown in Figure 4.5a. Looking at Figure 4.4, the system has reached but not exceeded the –5A current level, which is the maximum current available to the system. The thrust generated by the tether has a parallel behavior to the current. The thrust history for the first 2 orbits is given in Figure 4.5b. These plots illustrate the potential difficulty in numerically optimizing the trajectory for electrodynamic maneuvers. To capture the rapidly changing control, many parameter optimization techniques would require thousands of degrees of freedom.
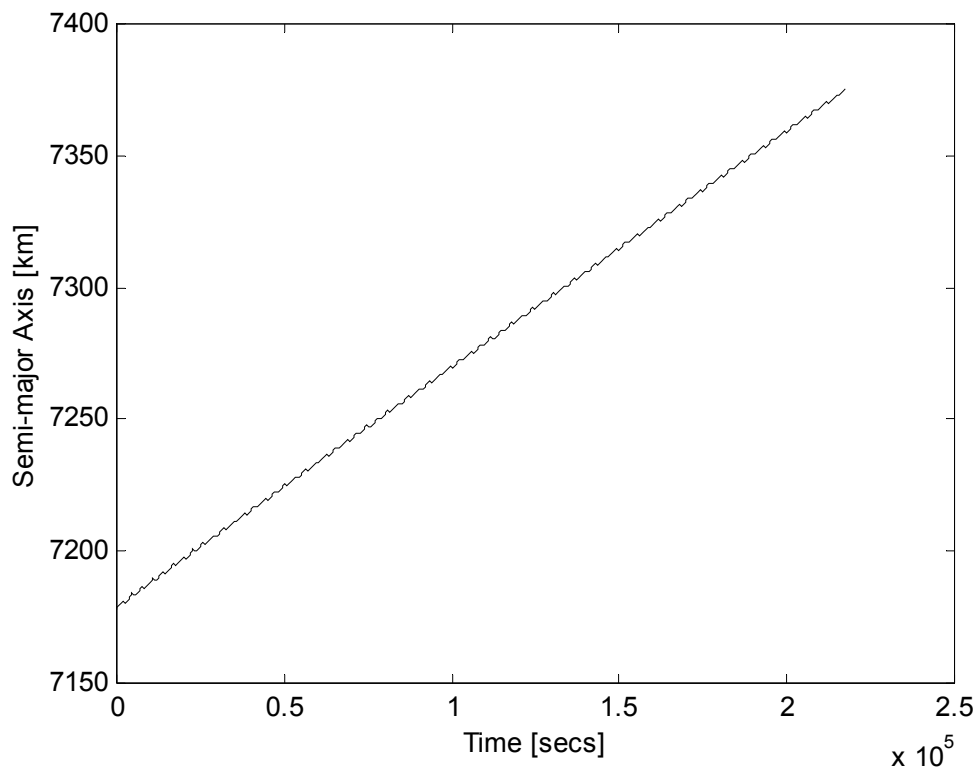
Figure 4.3 Change in Semi-major Axis for Orbit Raising
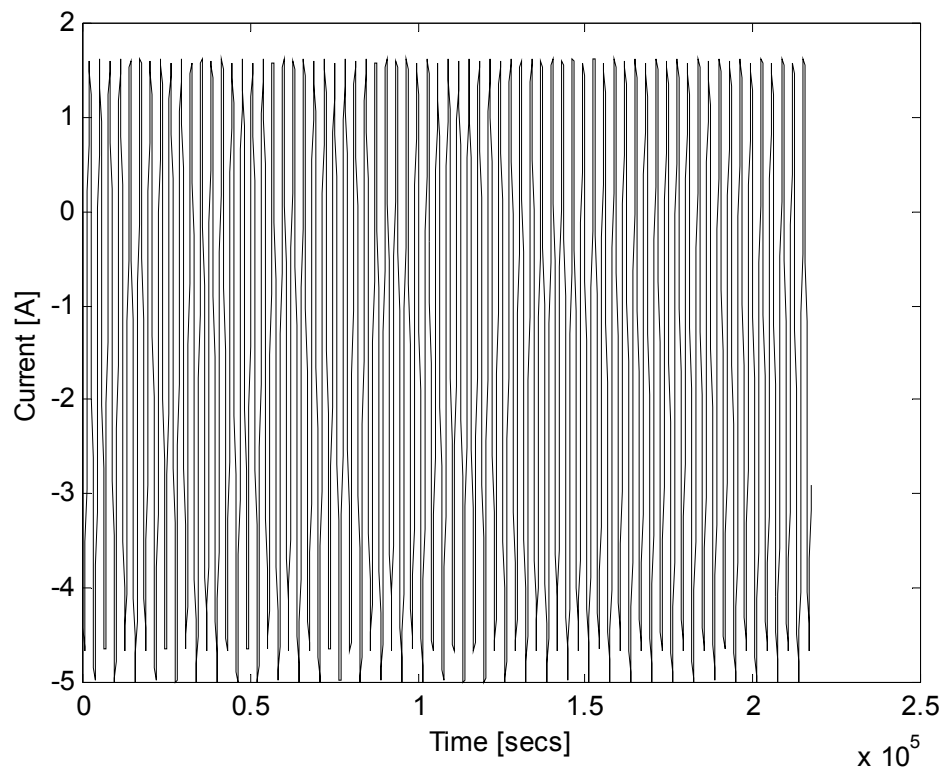
Figure 4.4 Tether Current History for Orbit Raising
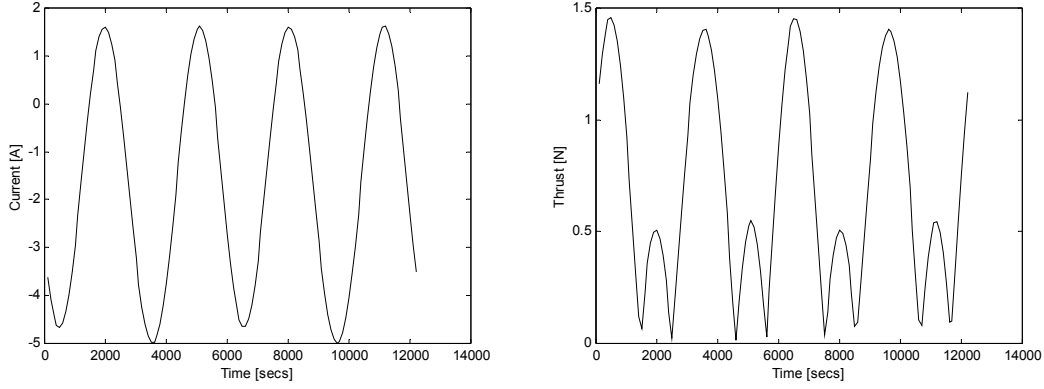
Figure 4.5 First Two Orbits of Orbit Raising (a) The Current Behavior for First Two Orbits (b) Tether Thrust for First Two Orbits

The absolute error in the final orbital elements for the orbit raising is

$$\Delta a_{err} = -2.58\,\text{km} \qquad \Delta e_{err} = -0.00053 \qquad \Delta \omega_{err} = -0.86°$$

$$\Delta i_{err} = -0.00046° \qquad \Delta \Omega_{err} = -0.00016° \tag{4.4}$$

The error is due to the inexact integration used in the current control law calculation. Better performances can be achieved if the guidance is employed in several phases. The same orbit raising maneuver was simulated with 6 separate calculations of the control law spaced out roughly 10 hours apart during the maneuver. The currents employed through the maneuver are close to the ones in performing the maneuver in a single step and the semi-major axis shows the same linear behavior (Figure 4.6a). But the final error is significantly improved:

$$\Delta a_{err} = -0.04\,\text{km} \qquad \Delta e_{err} = -0.00018 \qquad \Delta \omega_{err} = -0.22°$$

$$\Delta i_{err} = -0.00020° \qquad \Delta \Omega_{err} = -0.00066° \tag{4.5}$$

The behaviors of some of the orbital elements are shown in Figure 4.6a-c.

(a)	(b)
(c)	(d)
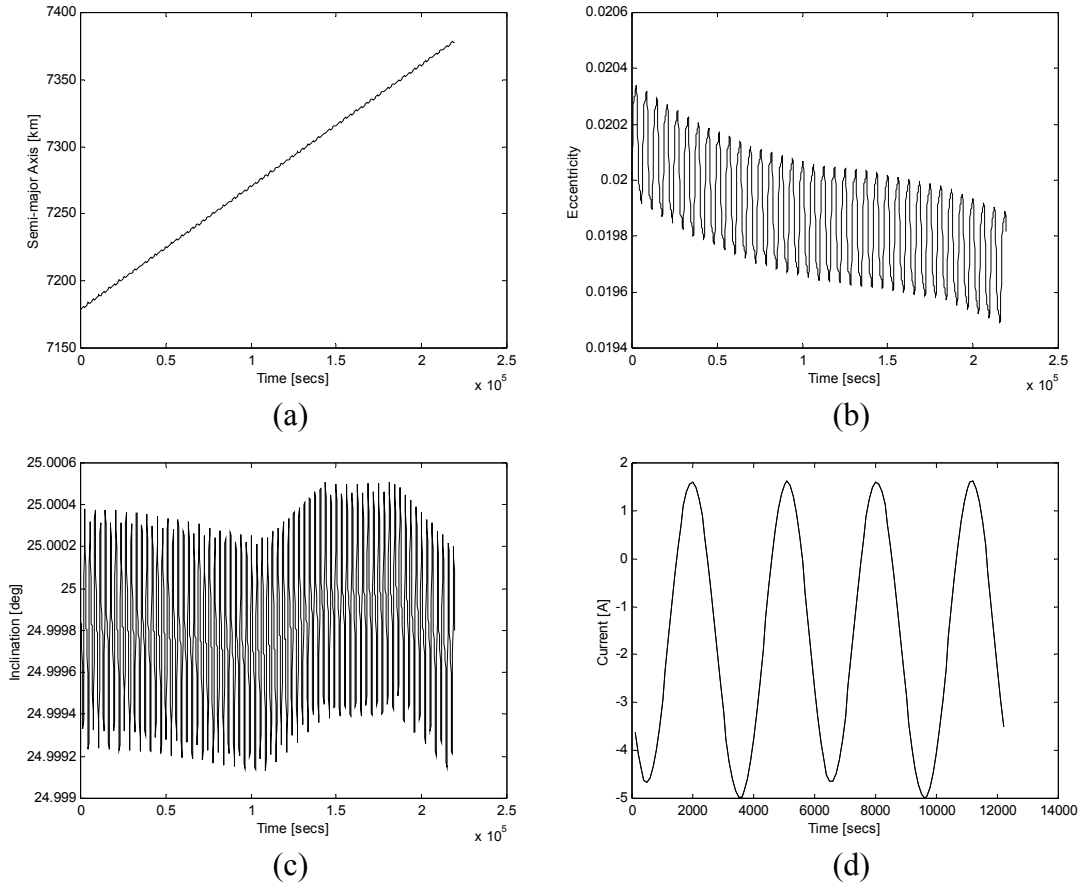
Figure 4.6 Orbital Raising with 6 Steps (a) Change in Semi-major Axis (b) Change in Eccentricity (c) Change in Inclination (d) Current Behavior for the First Two Orbits

*4.2.2 **Inclination Change.*** The same initial orbital elements and satellite parameters used in the orbit raising example are input to the simulation for an inclination change of 5 deg. The desired changes in the five orbital elements are

$$\Delta i = -5°$$

$$\Delta a = \Delta e = \Delta \omega = \Delta \Omega = 0 \qquad (4.6)$$

The control law and the required time of flight for the maneuver are

$$I = 0 - 0.06\cos\nu - 0.11\sin\nu - 4.89\cos[2(\nu + \omega)] - 0\sin[2(\nu + \omega)]$$

$TOF = 1067$ hr 44 min $\hspace{6cm}$ (4.7)

This maneuver takes almost 1.5 months because at this inclination the tether force component perpendicular to the orbital plane is relatively small since the tether force is always perpendicular to the electromagnetic field lines. As a result of the tether force's being perpendicular to the electromagnetic field lines, at lower inclinations in-plane forces are greater than the out-of-plane forces and the opposite is true for high inclinations. Therefore, in plane maneuvers such as changing eccentricity, argument of perigee and semi-major axis are most effective at lower inclinations and out of plane maneuvers such as inclination and argument of ascending nodes change are most effective at high inclinations. Performing the same maneuver for a similar satellite-orbit setup with 80° of initial inclination takes only 458 hours, 43% of the time of flight for the original maneuver.
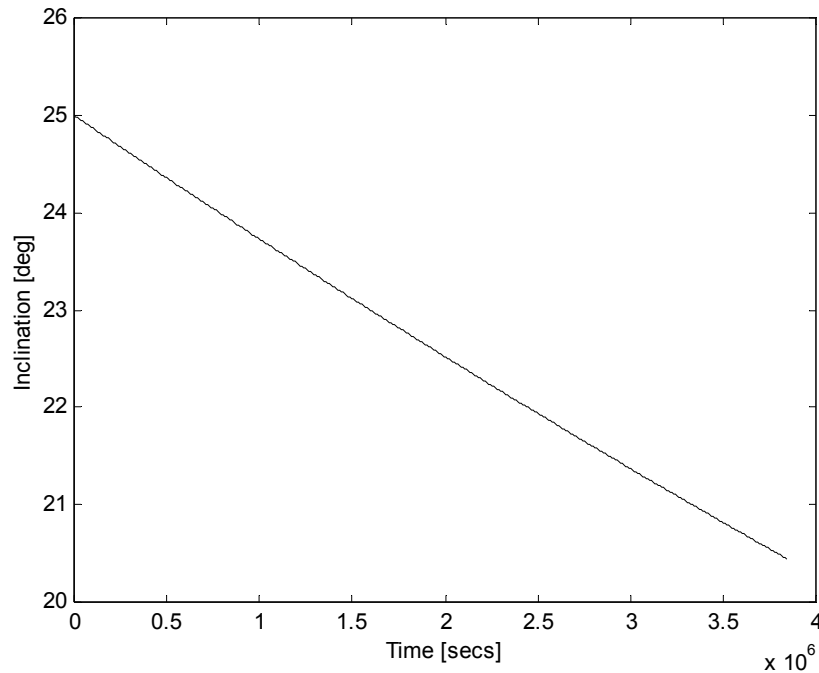
Figure 4.7 Inclination Change
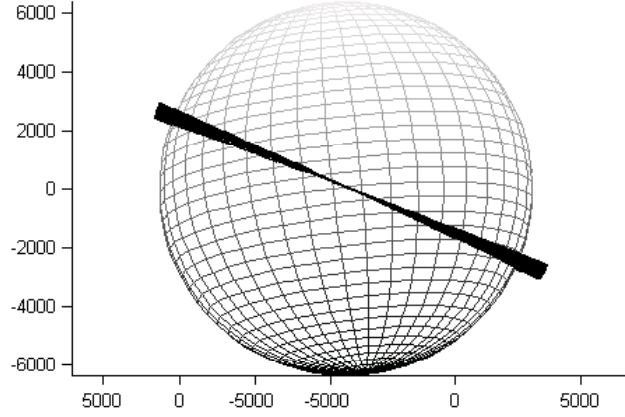
Figure 4.8 Orbital Trajectory for Inclination Change

Figure 4.8 shows the orbital trajectory through the maneuver and the resulting change in the inclination is depicted in Figure 4.7. The errors in the final orbital elements are

$$\Delta i_{err} = 0.44° \qquad \Delta a_{err} = -2.15\,km \qquad \Delta e_{err} = 0.005$$

$$\Delta \omega_{err} = 37.50° \qquad \Delta \Omega_{err} = -0.00044° \tag{4.8}$$

The errors introduced in this maneuver are significantly higher than the results of the orbit raising example. This is due to appearance of inclination in almost every element of the $[A]$ matrix (Table 3.2). As inclination changes, the approximation of assuming the orbital elements constant for the integration of the perturbation equations begins to fail. Examining Figure 4.7 carefully, it is seen that the long-term behavior of the inclination is no longer linear. The argument of perigee is one of the sensitive orbital elements and changing the argument of perigee takes less time. Hence, the error can be reduced by breaking the maneuver into a number of steps.

Performing the same maneuver in 20 steps results in considerable improvement in not only the error in argument of perigee but on other three orbital elements. Each phase of the maneuver takes approximately 61 hours. Considering the guidance computations have to be resulted in every 61 hours, the computational load is still very reasonable even for space rated hardware. The final orbit is much closer to the targeted orbit:

$$\Delta i_{err} = 0.002^{\circ} \qquad \Delta a_{err} = -0.38\,\text{km} \qquad \Delta e_{err} = 0.0007$$

$$\Delta \omega_{err} = 2.79^{\circ} \qquad \Delta \Omega_{err} = -0.0016^{\circ} \qquad\qquad (4.9)$$

The changes in eccentricity and the argument of perigee demonstrate the error correction by multi-phase maneuvering (Figures 4.9-4.10).
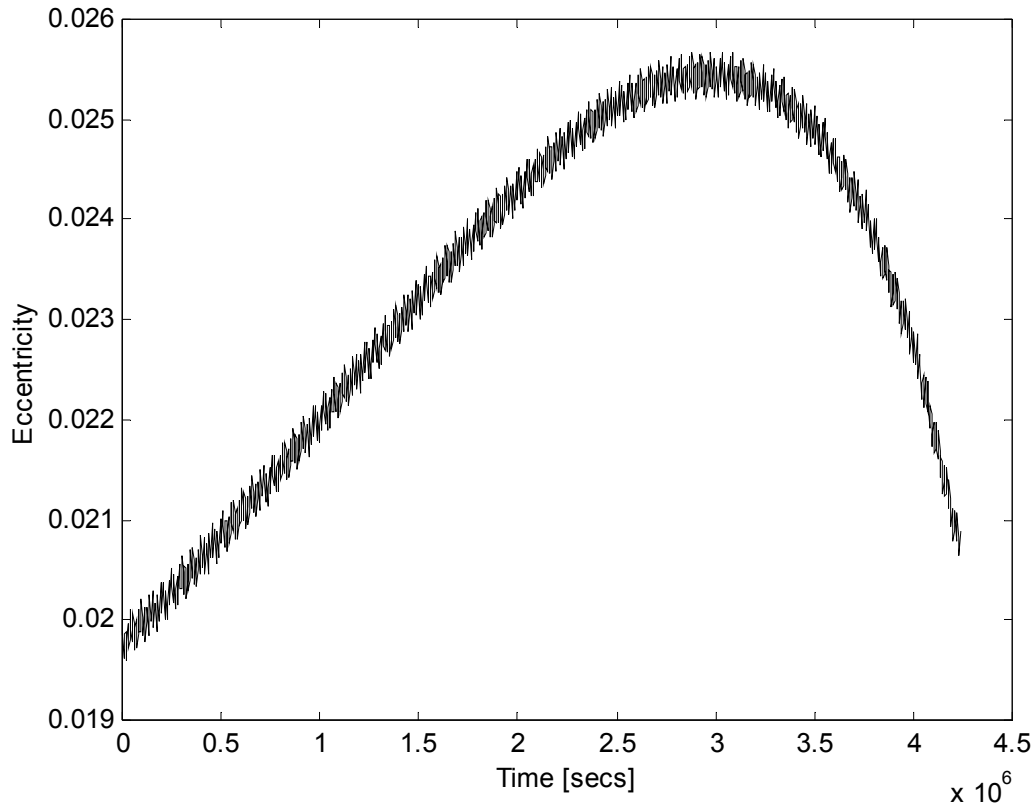
Figure 4.9 Change in Eccentricity for 20 Step Inclination Change Maneuver
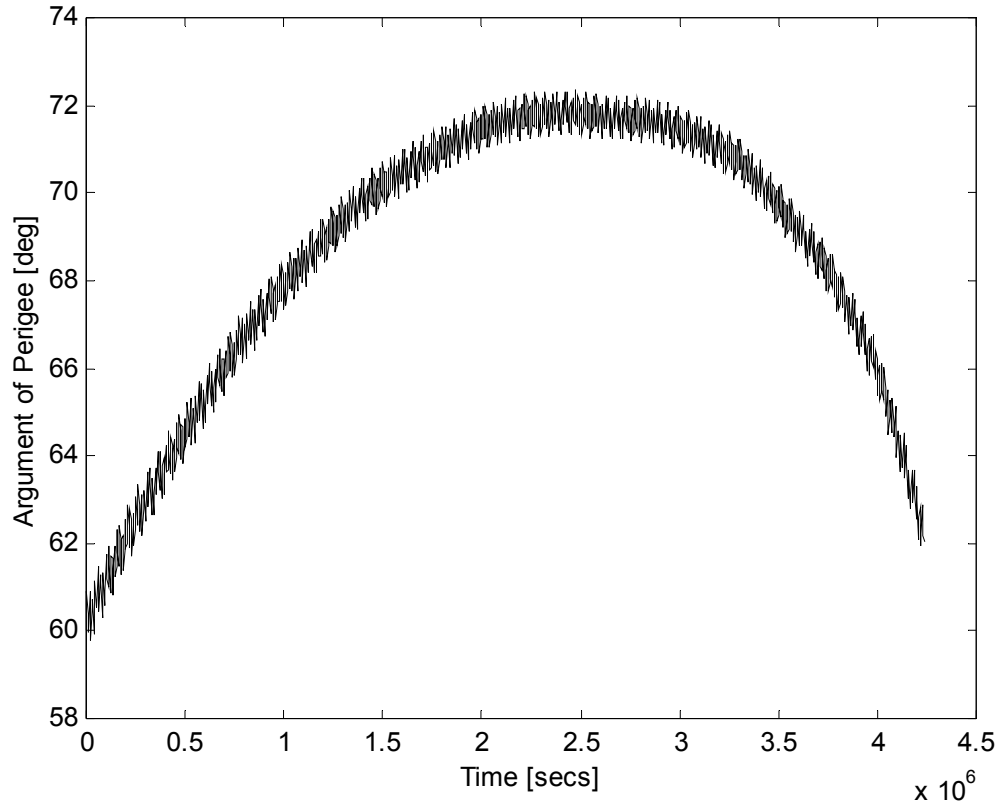
Figure 4.10 Change in Argument of Perigee for 20 Step Inclination Change Maneuver

The results of the maneuver performed with the initial inclination of 80° mentioned above show the dependency of the error on the initial orbital elements. The maneuver is executed in a single step. The errors are significantly less compared to the results of the single step maneuver at 25° of inclination:

$$\Delta i_{err} = 0.048^{\circ} \qquad \Delta a_{err} = -0.098\,\text{km} \qquad \Delta e_{err} = 0.00067$$

$$\Delta \omega_{err} = 3.865^{\circ} \qquad \Delta \Omega_{err} = 0.0010^{\circ} \tag{4.10}$$

4.2.3 General Orbit Change.   In this example the guidance scheme and the electrodynamic tether system is used to achieve a simultaneous change in all orbital elements. This will show the superposition rule assumed in Equation 3.31 is valid. The electrodynamic tethered satellite is supposed to be servicing Landsat-4, Landsat-5 and Landsat-7 satellites. The electrodynamic tethered satellite is initially co-orbital with the Landsat-5 satellite and will transfer to the Landsat-7 satellite. After servicing the Landsat-7 satellite, the electrodynamic tethered satellite will maneuver to meet the Landsat-4 satellite. The orbital elements of the satellites are [29]

Landsat-5: $a_5 = 7080.5\,\text{km}$ $e_5 = 0.001091$ $\omega_5 = 293.2112°$

$i_5 = 98.1763°$ $\Omega_5 = 115.5885°$

Landsat-7: $a_7 = 7080.6\,\text{km}$ $e_7 = 0.000491$ $\omega_7 = 38.6804°$

$i_7 = 98.2107°$ $\Omega_7 = 118.2413°$

Landsat-4: $a_4 = 6959.7\,\text{km}$ $e_4 = 0.008134$ $\omega_4 = 226.2668°$

$i_4 = 98.2336°$ $\Omega_4 = 126.0736°$ (4.11)

The first maneuver is from Landsat-5 satellite to Landsat-7 satellite. The electrodynamic tether system must perform the following orbital maneuvers

$\Delta a = 0.1\,\text{km}$ $\Delta e = \text{-}0.0006$ $\Delta\omega = 105.4692°$

$\Delta i = 0.0344°$ $\Delta\Omega = -2.6528°$ (4.12)

The control law and the time of flight to perform these simultaneous maneuvers are

$$I = 0.001 - 0.10\cos\nu + 0.356\sin\nu + 0.057\cos[2(\nu+\omega)] + 4.629\sin[2(\nu+\omega)]$$

$TOF = 244\,\text{hr}\,31\,\text{min}$ (4.13)
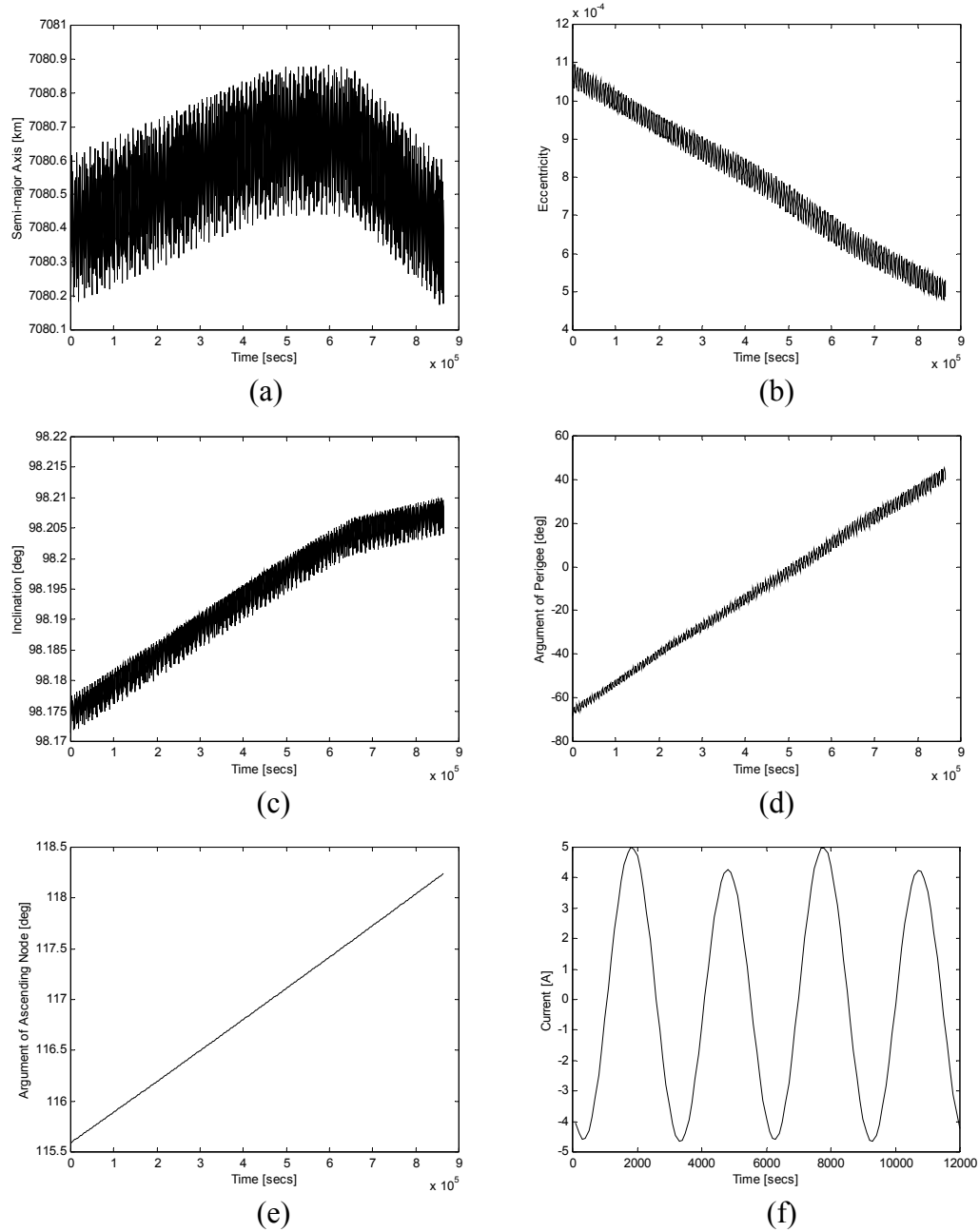
Figure 4.11   General Orbital Maneuver Between Landsat-5 and Landsat-7 (a) Change
in Semi-major Axis (b) Change in Eccentricity (c) Change in Inclination
(d) Change in Argument of Perigee (e) Change in Argument of Ascending
Node (f) Current Behavior in First Two Orbits
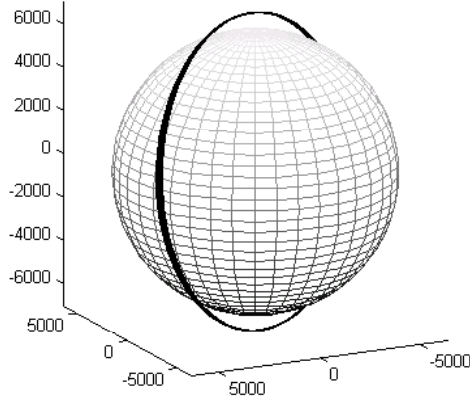
Figure 4.12 Orbital Trajectory for the Maneuver Between Landsat-5 and Landsat-7

A simulation with 4 segments in the guidance scheme yields the following errors for the first maneuver:

$$\Delta a_{err} = -0.007 \text{ km} \qquad \Delta e_{err} = 0 \qquad \Delta \omega_{err} = 0.75°$$

$$\Delta i_{err} = -0.0006° \qquad \Delta \Omega_{err} = 0.00012° \qquad (4.14)$$

The orbital trajectory is plotted in Figure 4.12; the changes in the orbital elements and the current history for the first two orbits are given in Figure 4.11.

The second maneuver is the orbital transfer from Landsat-7 satellite to Landsat-4 satellite. Assuming that the electrodynamic tethered satellite system is co-orbital with Landsat-7 satellite, the maneuvers desired to be performed are

$$\Delta a = -120.9 \text{ km} \qquad \Delta e = 0.00764 \qquad \Delta \omega = 187.59°$$

$$\Delta i = 0.0223° \qquad \Delta \Omega = 7.83° \qquad (4.15)$$

The solution for the control law and time of flight for a single step maneuver are

$$I = -0.44 + 0.395 \cos v - 0.077 \sin v + 0.899 \cos[2(v + \omega)] - 4.055 \sin[2(v + \omega)]$$

$$TOF = 823 \text{ hr } 57 \text{ min} \qquad (4.16)$$

4-15
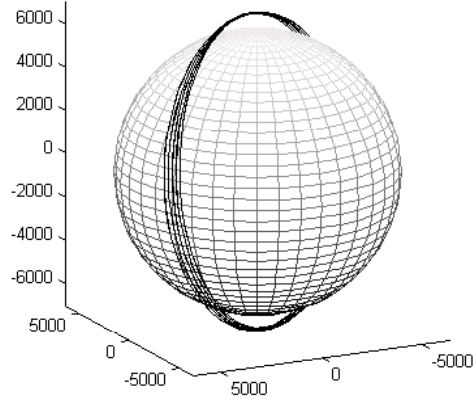
Figure 4.12 Orbital Trajectory for the Maneuver Between Landsat-7 and Landsat-4

The first maneuver was flown in 4 segments and the time of flight for each segment was approximately 61 hours. A simulation with 12 segments that corresponds to 68 hours for each segment is run for the second maneuver therefore roughly the same computational load is put in. The orbital trajectory of the maneuver is plotted in Figure 4.13. The errors at the end of the transfer are

$$\Delta a_{err} = 0.024 \, \text{km} \qquad \Delta e_{err} = 0 \qquad \Delta \omega_{err} = 1.22°$$

$$\Delta i_{err} = -0.0016° \qquad \Delta \Omega_{err} = -0.0026° \qquad (4.17)$$
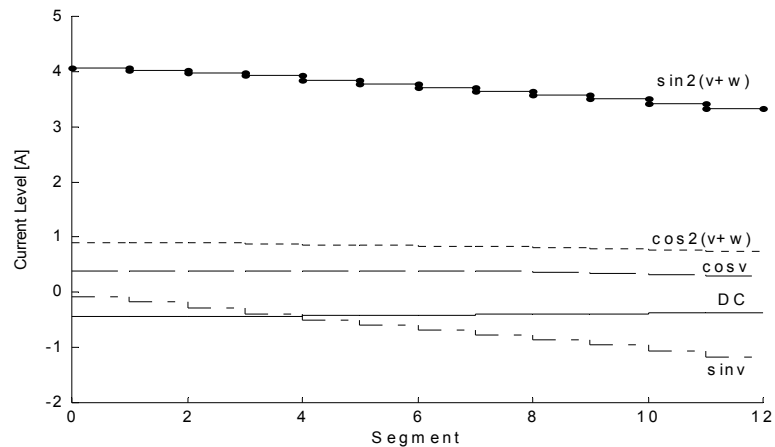
Figure 4.14 Current Levels for the Maneuver Between Landsat-7 and Landsat-4

4-16
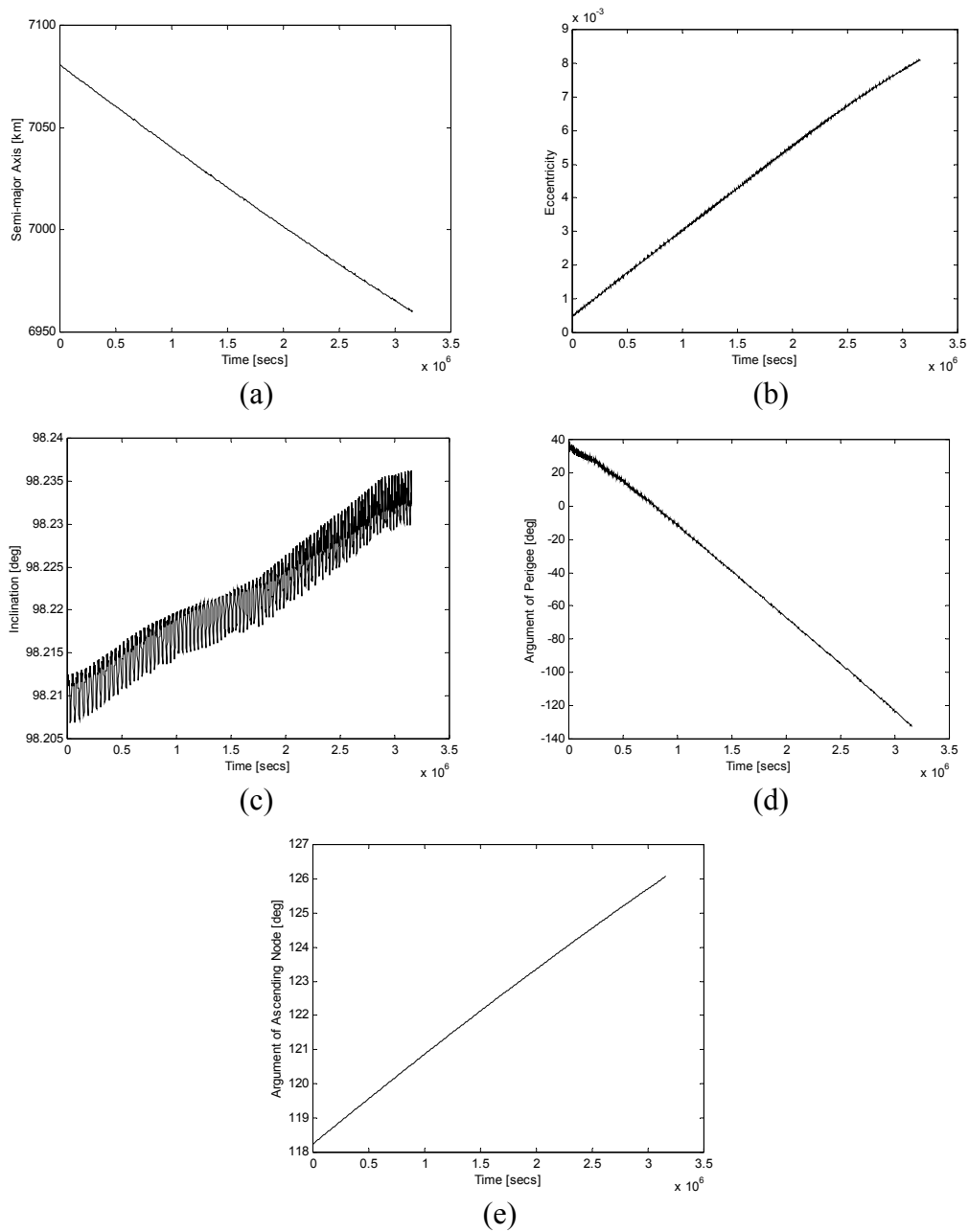
Figure 4.15   General Orbital Maneuver Between Landsat-7 and Landsat-4 (a) Change
in Semi-major Axis (b) Change in Eccentricity (c) Change in Inclination
(d) Change in Argument of Perigee (e) Change in Argument of Ascending
Node

The corrections made in the current control law throughout the maneuver at each segment are given in Figure 4.14. The changes obtained in the five orbital elements are given in Figure 4.15. Note that, even the current level of $\sin \nu$ current law, which corresponds to argument of perigee, is significantly altered through the maneuver (Figure 4.14), the change in argument of perigee has a linear-like behavior. This is due to the fact that the changes in orbital elements determine the level of current needed to have the same rate of change in a particular orbital element.

4.2.4 Simplified Matrix. As explained in Chapter 3, a more compact form of the matrix can be constructed by neglecting the higher terms of eccentricity, e. The second maneuver in general orbit change example will be simulated using the simplified matrix for the current control laws and the guidance scheme. The matrix is given in Table 3.2, Column 3.

Using the same initial and target orbital elements the simplified matrix gives almost the same current control law and the time of flight. The current coefficients are the same to the fourth digit after the point. A twelve-step guidance scheme introduces an additional $1 \times 10^{-4}{}^{\circ}$ error in inclination and the argument of ascending node in the final orbit.

These results show that the computational load can be even more reduced by the use of this simplified matrix.

4.3 Numerical Solution to the Problem

The non-linear solution mentioned in Chapter 3 is implemented in Fortran with the IMSL Math Libraries. The non-linear equation solver used in this implementation is a variation of Newton's method, which uses a finite-difference approximation to the Jacobian and takes precautions to avoid large step sizes or increasing residuals. Basically, the solver system starts with an initial guess and runs the simulation for the results. Comparing the results, a new set of currents is produced by the solver for the next step. The source code of the nonlinear solution implementation is given in Appendix C.

An arbitrary initial guess to the solver doesn't make any progress. Therefore, the results obtained from the current control law determined in this study are used as the initial guess for the non-linear equation solver. Unfortunately, even with these initial guesses and big tolerances given to the solver, the equation solver failed to converge to an answer for most of the problems including the general orbit change examples above.

The solver converged on an answer for a much simpler orbital maneuver with a time of flight of 197 hours. The computational cost of the system for a tolerance of 1×10^{-3} is more than 8 central processing unit minutes on a 900 MHz computer, which is a very large computation.

These results show that the non-linear solution is not robust and requires a large amount of computations. Furthermore, a very good initial guess is required.

V. Conclusions and Future Research

5.1 Conclusions

The current control law and guidance algorithm developed in this study is capable of performing different orbital maneuvers for systems using electrodynamic tether propulsion. The current control law relates the tether current modulation to the satellite's location in its orbit. The guidance scheme developed on this current control law can change the orbit size, shape and plane simultaneously to achieve a desired orbital maneuver. This guidance scheme works most effectively if the maneuver is broken up into several phases allowing for corrections on the current laws to compensate the errors introduced by the approximate solution to the general perturbation equations.

It is important to stress that the trajectories generated by this guidance scheme are not optimal, since the goal of this study was to keep the calculations as simple as possible. There is no iterative calculation or numerical simulation in the guidance equations. The simplicity of the guidance scheme and the minimal computational load required enable implementation of this scheme in an onboard system. This also allows rapidly generating many trajectories in a small period of time for concept exploration or mission planning.

Using non-linear equation solvers to determine an exact solution to the current control law showed that such a solver requires a very good initial guess, which can be provided by the guidance scheme presented here. However, the non-linear solution

involves large computational load and is not robust enough for an onboard implementation.

5.2 Future Research

The current control law and the guidance scheme presented in this study are developed under several assumptions. These assumptions need to be relaxed in order to provide more accurate evaluations of the feasibility of the electrodynamic tether maneuvering concept. Some issues need to be addressed in future research are pointed out in the following paragraphs.

As mentioned above, the trajectories generated by the guidance scheme are not optimal. A future study on adapting suitable optimization methods to this problem could provide more cost-effective solutions.

There is an aerodynamic drag force caused by the tether and plasma interaction that needs to be considered in determining the spacecraft's acceleration. Hence the electrodynamic tether propulsion concept is applicable in low earth orbits because of the plasma environment requirements, the drag force is considerable. Furthermore, the Lorentz force generated by the tether-magnetic field interaction should include the force generated by the orbital velocity of the tether.

The calculation of amount of current that can be flown through the tether needs to consider the physics of the space plasma. As the plasma density changes with the altitude and location, the amount of current that can be run through the plasma varies although the current loss due to the electron density drop can be compensated to a degree by voltage

adjustments. Additionally, there would be a generated EMF on the tether that needs to be overcome by the voltage driving the current that will limit the tether current.

The tether is assumed to be straight and perfectly aligned with the local vertical. However, the forces on the tether will cause significant bending on the tether unless a more sophisticated current control scheme is use.

Finally, the earth's oblateness and the rotation of the magnetic dipole in the inertial frame should be included in the models used to derive the guidance equations. The dipole's rotation with earth needs to be accounted for in the guidance equations. Including the oblateness, the effects of the precision of the line of nodes and apsides on the maneuver could potentially be offset by the electrodynamic tether.

Appendix A. Validity of the Equations – Fortran Simulation

A.1 Tether Force Perturbation Simulation in Cartesian Coordinates

```fortran
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C TetherForceSimulationXYZ.f90
C
C       SIMULATION ALGORITHM FOR TETHER FORCE PERTURBATION
C               IN CARTESIAN COORDINATES
C
C               Lt Hakan San - AFIT/ENY/GA-02M
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      implicit none
      parameter( mxparm=50,neq=6 )
      integer mxparm,neq,ido, i
      double precision  x(neq),param(mxparm),t,tend,tol
      double precision  n, a, tf, hr, min, sec

      double precision mu, mum, m, L, Ic, pi, currents(5), ccI
      common mu, mum, m, L, Ic, n, pi, currents, ccI

      external divprk, fcn, cross, norm

      pi = 3.1415926535897932384626433832795028841971693993751d0

      tol=1.d-10                 ! Numerical Integration Tolerance

C Parameters
      mu=3.986d5                 ! Earth's Gravitaional Constant
      mum=8.0711d6               ! Earth's Magnetic Dipole Moment
      Ic=-5.d0                   ! Applied Tether Current
      L=15.d0                    ! Tether Length
      m=1000.d0                  ! System Mass

      t=0.d0

C Read Initial Position and Velocity From File (Cartesian Coordinates)
      write (*,*) 'initial position and velocity'
      i=1
      open (1, FILE = 'initial.2bp')
      do while (.NOT. (eof(1) .OR. (i==7)))
          read (1,*) x(i)
          write (*,*) x(i)
          i=i+1
      end do
      close(1)

C Read Current Law Coefficients and Time of Flight from File
      write (*,*) 'currents'
      i=1
      open (2, FILE = 'currents.2bp')
      do while (.NOT. (eof(2) .OR. (i==6)))
          read (2,*) currents(i)
          write (*,*) currents(i)
          i=i+1
      end do
```

```fortran
      write (*,*) 'Time of Flight'
      read (2,*) tf
      hr=int(tf/3600)
      min=int(mod(tf,3600.d0)/60)
      sec=tf-hr*3600-min*60
      write (*,*) tf
      write (*,*) hr, 'hr', min, 'min', sec, 'sec'
      close(2)

C Open Output Files
C Position
      open(10,file='../position.2bp', STATUS='REPLACE')
C Velocity
      open(20,file='../velocity.2bp', STATUS='REPLACE')
C Time
      open(30,file='../time.2bp', STATUS='REPLACE')
C Current History
      open(40,file='../curhist.2bp', STATUS='REPLACE')

      ido=1
      param(4)=10000000.d0    ! Maximum Number of Iterations for
                              ! Numerical Integrator

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c START INTEGRATION LOOP
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      do 100 tend=0.d0,tf,100.d0
         call divprk (ido,neq,fcn,t,tend,tol,param,x)

C Write Outputs to Files
         write(10,*) x(1),x(2),x(3)
         write(20,*) x(4),x(5),x(6)
         write(30,*) t
         write(40,*) ccI

100   continue

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c END INTEGRATION LOOP
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      close(10)
      close(20)
      close(30)

      stop
      end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C EQUATIONS OF MOTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine fcn(neq,t,x,xd)

      implicit none

      integer neq
      double precision t, x(neq), xd(neq), r, em(3), Ict
```

```fortran
      double precision ap(3), temp

      double precision Hvec(3), rv(3), vv(3), tvec(3), evec(3), nvec(3)
      double precision v, a, e, i, w, Om, nu

      double precision mu, mum, m, L, Ic, n, as, pi, currents(5), ccI
      common mu, mum, m, L, Ic, n, pi, currents, ccI

      double precision norm, dot
      double precision I1, I2, I3, I4, I5

      double precision k1, k2, k3, k4


      em=[0.d0, 0.d0, 1.d0]         ! Earth's Magnetic Dipole Axis

C Convert Position and Velocity Classical Orbital Elements

      rv=[x(1), x(2), x(3)]         ! Position Vector
      vv=[x(4), x(5), x(6)]         ! Velocity Vector
      r=norm(rv)
      v=norm(vv)

      a=-mu/2.d0/(v**2.d0/2.d0-mu/r)        ! Semi-major Axis

      call cross(rv, vv, Hvec)
      call cross(vv, Hvec, tvec)

      evec=(tvec/mu-rv/r)                    ! Eccentricity Vector

      e=norm(evec)                           ! Eccentircity

      i=dacos(Hvec(3)/norm(Hvec))            ! Inclination

C If Inclined Orbit, Calculate Line of Nodes
      if (dabs(i)>1.d-13) then
          nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
      end if

C No Eccentricity and Inclination Case
      if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
          w=0.d0                       ! Argument of Perigee
          Om=0.d0                      ! Argument of Line of Nodes
          nu=datan2(rv(2),rv(1))       ! True Anomaly

C No Eccentricity Case
      else if (dabs(e)<1.d-13) then
          Om=datan2(nvec(2),nvec(1))
          w=0.d0
          nu=dacos(dot(nvec,rv)/r)
          if (rv(3)<0.d0) nu=-nu        ! Quadrant Check

C No Inclination Case
      else if (dabs(i)<1.d-13) then
          Om=0.d0
          w=datan2(evec(2),evec(1))
          nu=real(dacos(dot(evec,rv)/e/r))
          if (dot(rv,vv)<0.d0) nu=-nu

C Inclination and Eccentricity NON-Zero
      else
```

A-3

```fortran
        Om=datan2(nvec(2),nvec(1))
        w=dacos(dot(nvec,evec)/e)
        if (evec(3)<0.d0) w=-w
        as=dot(evec,rv)
        as=as/e/r

        ! The 'as' value must be smaller than zero (Fortran Error)
        if (as>1.d0) then
            as=1.d0
        end if

        nu=dacos(as)
        if (dot(rv,vv)<0) nu=-nu

    end if

    I1=currents(1)                      ! Current Law Coefficients
    I2=currents(2)
    I3=currents(3)
    I4=currents(4)
    I5=currents(5)

C Current Law Super Position
    Ict=Ic*(I1+I2*dcos(nu)+I3*dsin(nu)-I4*dcos(2.d0*(nu+w)))
    Ict=Ict+Ic*(-I5*dsin(2.d0*(nu+w)))
    ccI=Ict                             ! For Current History

C Perturbing Tether Acceleration
    temp=(Ict*L/m)*(mum/r**3.d0)*(1.d0/r)
    ap=[temp*rv(2), -temp*rv(1), 0.d0]

C Two Body Problem Equations of Motion
    xd(1)=x(4)
    xd(2)=x(5)
    xd(3)=x(6)
    xd(4)=-mu*x(1)/r**3.d0+ap(1)
    xd(5)=-mu*x(2)/r**3.d0+ap(2)
    xd(6)=-mu*x(3)/r**3.d0+ap(3)

    return

    end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Vector Cross Product
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

    subroutine cross(v1, v2, v3)

    implicit none

    double precision v1(3), v2(3), v3(3)

    v3(1) = v1(2) * v2(3) - v1(3) * v2(2)
    v3(2) = v1(3) * v2(1) - v1(1) * v2(3)
    v3(3) = v1(1) * v2(2) - v1(2) * v2(1)

    return
    end
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   Vector Dot Product
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      double precision function dot(v1, v2)
      implicit none

      double precision v1(3), v2(3)

      dot=v1(1)*v2(1)+v1(2)*v2(2)+v1(3)*v2(3)

      return

      end


cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   Vector Magnitude
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      double precision function norm(v3)
      implicit none

      double precision v3(3)

      norm=dsqrt(v3(1)**2.d0+v3(2)**2.d0+v3(3)**2.d0)

      return

      end
```

A.2 Direct Numerical Integration of Perturbation Equations

```fortran
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C TetherForceSimulationCOE.f90
C
C      SIMULATION ALGORITHM FOR TETHER FORCE PERTURBATION
C         DIRECT INTEGRATION OF PERTURBATION EQUATIONS
C
C               Lt Hakan San - AFIT/ENY/GA-02M
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      implicit none
      parameter( mxparm=50,neq=6 )
      integer mxparm,neq,ido, I
      double precision  x(neq),param(mxparm),t,tend,tol
      double precision  n, a, hr, min, sec, tf

      double precision mu, mum, m, L, Ic, pi, currents(5), Eold
      common mu, mum, m, L, Ic, n, pi, currents, Eold

      external divprk, fcn

      pi = 3.141592653589793238462643383279502884197169399375510d0

      tol=1.d-10                 ! Numerical Integration Tolerance

C Parameters
      mu=3.986d5                 ! Earth's Gravitaional Constant
      mum=8.0711d6               ! Earth's Magnetic Dipole Moment
      Ic=-5.d0                   ! Applied Tether Current
      L=15.d0                    ! Tether Length
      m=1000.d0                  ! System Mass

      t=0.d0

C Read Initial Classical Orbital Elements
      write (*,*) 'initial coes'
      i=1
      open (1, FILE = 'initial.coe')
      do while (.NOT. (eof(1) .OR. (i==7)))
          read (1,*) x(i)
          write (*,*) x(i)
          i=i+1
      end do
      close(1)

C Read Current Law Coefficients and Time of Flight from File
      write (*,*) 'currents '
      i=1
      open (2, FILE = 'currents.2bp')
      do while (.NOT. (eof(2) .OR. (i==6)))
          read (2,*) currents(i)
          write (*,*) currents(i)
          i=i+1
      end do
      read (2,*) tf
      close(2)
```

```fortran
      write (*,*) 'Time of Flight'
      hr=int(tf/3600)
      min=int(mod(tf,3600.d0)/60)
      sec=tf-hr*3600-min*60
      write (*,*) tf
      write (*,*) hr, 'hr', min, 'min', sec, 'sec'

      Eold=0

C Open Output Files
C Semi-major Axis, Eccentricity, Inclination (radians)
      open(10,file='../../aei.coe', STATUS='REPLACE')
C RAAN, Argument of Perigee, Phase (radians)
      open(20,file='../../owm.coe', STATUS='REPLACE')
C Time
      open(30,file='../../time.coe', STATUS='REPLACE')

      ido=1
      param(4)=10000000.d0     ! Maximum Number of Iterations for
                               ! Numerical Integrator

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c START INTEGRATION LOOP
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        do 100 tend=0.d0,tf,100.d0
           call divprk (ido,neq,fcn,t,tend,tol,param,x)

C Write Outputs to Files
         write(10,*) x(1),x(2),x(3)
         write(20,*) x(4),x(5),x(6)
         write(30,*) t

100     continue

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c END INTEGRATION LOOP
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      close(10)
      close(20)
      close(30)

      stop
      end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C EQUATIONS OF MOTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine fcn(neq,t,x,xd)

      implicit none

      integer neq, bool
      double precision t, x(neq), xd(neq), Ict

      double precision r, a, e, i, w, Om, nu, M1, C, E1, Enew, at, aw,
     &                 p, h, D
```

```fortran
      double precision mu, mum, m, L, Ic, n, pi, currents(5), Eold
      common mu, mum, m, L, Ic, n, pi, currents, Eold

      double precision I1, I2, I3, I4, I5

      a=x(1)
      e=x(2)
      i=x(3)
      Om=x(4)
      w=x(5)
      M1=x(6)

C Calculate True Anomaly
      E1=Eold
      bool=1
      do while (bool)
          Enew=E1-(E1-e*dsin(E1)-M1)/(1.d0-e*dcos(E1))
          if (dabs(Enew-E1)<1.d-12) then
              bool=0
          else
              E1=Enew
          end if
      end do
      Eold=E1

      nu=2.d0*datan(dsqrt((1.d0+e)/(1.d0-e))*dtan(E1/2.d0))

      I1=currents(1)                        ! Current Law Coefficients
      I2=currents(2)
      I3=currents(3)
      I4=currents(4)
      I5=currents(5)

      Ict=Ic*(I1+I2*dcos(nu)+I3*dsin(nu)-I4*dcos(2.d0*(nu+w)))
      Ict=Ict+Ic*(-I5*dsin(2.d0*(nu+w)))

      n=dsqrt(mu/(a**3.0d0))                          ! Mean Anomaly

      r=a*(1.d0-e**2.d0)/(1.d0+e*dcos(nu))            ! Position

      C=L*mum/(m*n*a**3.0d0*(1.0d0-e**2.0d0)**(5.0d0/2.0d0))
      D=Ict*L*mum/(m*r**3.d0)

C Perturbing Tether Forces
      at=-D*dcos(i)
      aw= D*dcos(nu+w)*dsin(i)


C Semi-major Axis
      xd(1)=(-2.0d0*C*dcos(i)/(1.0d0-e**2.0d0))*(1.0d0+e*dcos(nu))**4.0d0*Ict

C Eccentricity
      xd(2)=(-C*dcos(i)/(a*e))*(2.0d0*e*dcos(nu)+e**2.0d0*dcos(nu)**2.0d0+
      &      e**2.0d0)*(1.0d0+e*dcos(nu))**2.0d0*Ict

C Inclinatio
      xd(3)=(C*dsin(i)/a)*dcos(nu+w)**2.0d0*(1.0d0+e*dcos(nu))**2.0d0*Ict

C Argument of Ascending Node
      xd(4)=(C/a)*dsin(nu+w)*dcos(nu+w)*(1.0d0+e*dcos(nu))**2.0d0*Ict
```

A-8

```fortran
C Argument of Perigee
      xd(5)=(-C*dcos(i)/a/e)*(1.0d0+e*dcos(nu))**2.0d0*((2.0d0+
     &      e*dcos(nu))*dsin(nu)+e*dsin(nu+w)*dcos(nu+w))*Ict
C Phase
      p=a*(1.d0-e**2.d0)
      h=dsqrt(mu*p)
      xd(6)=n+(C*dsqrt(1.d0-e**2.d0)*dcos(i)/(a*e))*(2.d0+e*dcos(nu))*(1.d0+
     &      e*dcos(nu))**2.d0*dsin(nu)*Ict

      return

      end
```

Appendix B. Guidance and Simulation Algorithm for Orbital Maneuvers Using Electrodynamic Tethers

```fortran
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C TetherForceSimulationCOE.f90
C
C            GUIDANCE AND SIMULATION ALGORITHM FOR
C        ORBITAL MANEUVERS USING ELECTRODYNAMIC TETHERS
C
C                Lt Hakan San - AFIT/ENY/GA-02M
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      implicit none
      integer mxparm,neq,ido
      parameter( mxparm=50,neq=6 )
      double precision  x(neq),param(mxparm),t,tend,tol

      double precision  hr, min, sec
      integer i, j, num_steps, step

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

      external divprk, fcn

      double precision Icoes(6), Tcoes(6), Intcoes(6)
      double precision tf, ts, tleft, Int_st, Int_et
      double precision deltas(5,1), Intdeltas(5,1)
      double precision intmatr(5,5)
      double precision tempmatr(5,5)
      double precision k, r1, r2, curI, maxI, pe
      integer r10, r1f, stepper, stepperold
      integer     datetime(8)
      logical dispmode, targetmode

      intrinsic reshape

      pi = 3.141592653589793238462643383279502884197169399375 10d0

      tol=1.d-10               ! Numerical Integration Tolerance

C Parameters
      mu=3.986d5               ! Earth's Gravitaional Constant
      mum=8.0711d6             ! Earth's Magnetic Dipole Moment


      open (UNIT = 999 , FILE = 'USER')

C Read Satellite Parameters
      open (1, FILE = 'satellite.coe')
          read(1,*) Ic                     ! Maximum Available Current [A]
          read(1,*) L                      ! Tether Length            [km]
          read(1,*) m                      ! System Mass              [kg]

          write(*,'(A10,F5.1,A20,F8.2,A15,F8.2)') ' Max. Cur: ',
     &         Ic,'Tether Length: ', L,'Sat. Mass: ',m
      close(1)

      Ic=-dabs(Ic)
```

```
      Imax=0
      stepperold=0

C Output File for Screen Dump
      open(101,file='output.txt', STATUS='REPLACE')

C Date/Time Stamp
      call date_and_time(VALUES = datetime)
      write(101,'(A9,I2,A1,I2,A1,I4,A3,I2,A1,I2,A1,I2,A9)') '------- ',
     &          datetime(2),'/', datetime(3),'/', datetime(1),' * ',
     &          datetime(5),':', datetime(6),':', datetime(7),'------- '

C Output File Applied Current Coeffiecients During the Manuever
      open(102,file='currents.txt', STATUS='REPLACE')

C Date/Time Stamp
      write(102,'(A9,I2,A1,I2,A1,I4,A3,I2,A1,I2,A1,I2,A9)') '------- ',
     &          datetime(2),'/', datetime(3),'/', datetime(1),' * ',
     &          datetime(5),':', datetime(6),':', datetime(7),'------- '
      write(102,*)
      write(102,*) ' Coefficients For Currents Applied (I=x*Iav) [ x ; I ] '
      write(102,*) '---------------------------------------------------'
      write(102,*)


C Read Run Mode Info and Number of Steps to Perform the Manuever
      open (1, FILE = 'runinfo.coe')
          read(1,*) dispmode              ! Runtime Display Mode
          read(1,*) targetmode            ! Target File Mode
          read(1,*) num_steps             ! Number of Steps
      close(1)


C Read Initial Orbit Classical Orbital Elements (COEs)
      open (1, FILE = 'initial.coe')
          read(1,*) Icoes(1)
          read(1,*) Icoes(2)
          read(1,*) Icoes(3)
          read(1,*) Icoes(4)
          read(1,*) Icoes(5)
          read(1,*) Icoes(6)
      close(1)

C Output Initial COEs (deg) to Screen
      write(*,*) 'Initial COEs '
      write(*,*) '-------------------------------'
      write(*,'(A21,F11.5)') 'Semi M. Axis       : ',Icoes(1)
      write(*,'(A21,F11.5)') 'Eccentricity       : ',Icoes(2)
      write(*,'(A21,F11.5)') 'Inclination        : ',Icoes(3)*180.d0/pi
      write(*,'(A21,F11.5)') 'Arg. of Perigee    : ',Icoes(4)*180.d0/pi
      write(*,'(A21,F11.5)') 'R. Ascending Node  : ',Icoes(5)*180.d0/pi
      write(*,'(A21,F11.5)') 'Phase              : ',Icoes(6)*180.d0/pi
      write(*,*)

C Output Initial COEs (deg) to File
      write(101,*) 'Initial COEs '
      write(101,*) '-------------------------------'
      write(101,'(A21,F11.5)') 'Semi M. Axis       : ',Icoes(1)
      write(101,'(A21,F11.5)') 'Eccentricity       : ',Icoes(2)
      write(101,'(A21,F11.5)') 'Inclination        : ',Icoes(3)*180.d0/pi
      write(101,'(A21,F11.5)') 'Arg. of Perigee    : ',Icoes(4)*180.d0/pi
```

```fortran
      write(101,'(A21,F11.5)') 'R. Ascending Node : ',Icoes(5)*180.d0/pi
      write(101,'(A21,F11.5)') 'Phase             : ',Icoes(6)*180.d0/pi
      write(101,*)

C Read Parameters Target Orbit COEs

      if (targetmode) then                   ! Target Orbit COEs

          write(*,*) '.......mode 1: reading target.coe......'
          write(101,*) '.......mode 1: reading target.coe......'
          open (1, FILE = 'target.coe')
              read(1,*) Tcoes(1)
              read(1,*) Tcoes(2)
              read(1,*) Tcoes(3)
              read(1,*) Tcoes(4)
              read(1,*) Tcoes(5)
              read(1,*) Tcoes(6)
          close(1)

      else                                   ! Orbit Change (Delta COEs)

          write(*,*) '.......mode 0: reading delta.coe.......'
          write(101,*) '.......mode 0: reading delta.coe.......'
          open (1, FILE = 'delta.coe')
              read(1,*) Intcoes(1)
              read(1,*) Intcoes(2)
              read(1,*) Intcoes(3)
              read(1,*) Intcoes(4)
              read(1,*) Intcoes(5)
              read(1,*) Intcoes(6)
          close(1)

          Tcoes(1)=Icoes(1)+Intcoes(1)
          Tcoes(2)=Icoes(2)+Intcoes(2)
          Tcoes(3)=Icoes(3)+Intcoes(3)
          Tcoes(4)=Icoes(4)+Intcoes(4)
          Tcoes(5)=Icoes(5)+Intcoes(5)
          Tcoes(6)=Icoes(6)+Intcoes(6)
      end if


C Output Target COEs (deg) to Screen
      write (*,*) 'Target COEs '
      write (*,*) '-------------------------------'
      write(*,'(A21,F11.5)') 'Semi M. Axis      : ',Tcoes(1)
      write(*,'(A21,F11.5)') 'Eccentricity      : ',Tcoes(2)
      write(*,'(A21,F11.5)') 'Inclination       : ',Tcoes(3)*180.d0/pi
      write(*,'(A21,F11.5)') 'Arg. of Perigee   : ',Tcoes(4)*180.d0/pi
      write(*,'(A21,F11.5)') 'R. Ascending Node : ',Tcoes(5)*180.d0/pi
      write(*,'(A21,F11.5)') 'Phase             : ',Tcoes(6)*180.d0/pi
      write(*,*)

C Output Target COEs (deg) to File
      write (101,*) 'Target COEs '
      write (101,*) '-------------------------------'
      write(101,'(A21,F11.5)') 'Semi M. Axis      : ',Tcoes(1)
      write(101,'(A21,F11.5)') 'Eccentricity      : ',Tcoes(2)
      write(101,'(A21,F11.5)') 'Inclination       : ',Tcoes(3)*180.d0/pi
      write(101,'(A21,F11.5)') 'Arg. of Perigee   : ',Tcoes(4)*180.d0/pi
      write(101,'(A21,F11.5)') 'R. Ascending Node : ',Tcoes(5)*180.d0/pi
      write(101,'(A21,F11.5)') 'Phase             : ',Tcoes(6)*180.d0/pi
```

```
      write(101,*)




ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    GUIDANCE ALGORITHM
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc


C First Guess For Performing the Manuever in Single Step

C Calculate Delta COEs Vector

      deltas=reshape((/(Tcoes(1)-Icoes(1)),(Tcoes(2)-Icoes(2)),
     &           (Tcoes(4)-Icoes(4)),(Tcoes(3)-Icoes(3)),
     &           (Tcoes(5)-Icoes(5))/),(/5,1/))


C If the Manuever is Greater than 180 deg, Go Other Way Around
      do i=3,5
          if (abs(deltas(i,1))>pi) then
              deltas(i,1)=sign((2*pi-abs(deltas(i,1))),-deltas(i,1))
          end if
      end do


C Evaluate the Guidance Matrix
      call eval_matrix(Icoes,intmatr)

C Take The Inverse of the Matrix
      call invert(intmatr,tempmatr,5)

C Calculate Tether Current Law Coefficients
      currents=1.d0/Ic*matmul(tempmatr,deltas)

      k=1
      maxI=0
      r10=int(Icoes(4)*180.d0/pi)*k           ! Argument of Perigee Range
      r1f=int(Tcoes(4)*180.d0/pi)*k

      if (r10>r1f) then
          r1=r10
          r10=r1f-5
          r1f=r1+5
          else
          r10=r10-5
          r1f=r1f+5
      end if

C Brute-Force Search For Maximum Current
      do i=r10, r1f
          r1=i*pi/180/k
          do j=0, 360*k
              r2=j*pi/180/k
              curI=currents(1,1)+currents(2,1)*dcos(r2)+
     &        currents(3,1)*dsin(r2)-currents(4,1)*dcos(2.d0*(r2+r1))-
     &        currents(5,1)*dsin(2.d0*(r2+r1))
              if (dabs(curI)>maxI) maxI=dabs(curI)
          end do
```

B-4

```
      end do



C Recalculate Time Of Flight and Current Coefficients
C To Fit the Maximum Current Requirement

      pe=2.d0*pi/dsqrt(mu/Icoes(1)**3.d0)     ! Orbit Period
      tf=maxI*pe

      currents=pe/tf/Ic*matmul(tempmatr,deltas)

C Time of Flight

      hr=int(tf/3600)
      min=int(mod(tf,3600.d0)/60)
      sec=tf-hr*3600-min*60
      write(*,'(A21,F16.2,A5)') 'Est. Total Time   : ', tf, ' secs'
      write (*,*) '------------------------------'
      write(*,'(F6.0,A4,F4.0,A5,F5.2,A5)') hr, 'hr ', min, 'min ', sec, ' sec'

      write(101,'(A21,F16.2,A5)') 'Est. Total Time   : ', tf, ' secs'
      write (101,*) '------------------------------'
      write(101,'(F6.0,A4,F4.0,A5,F5.2,A5)') hr, 'hr ', min, 'min ', sec, ' sec'

      if (dispmode) then
          write (*,*) '.....press return.....'
          read(*,*)
      end if

      call coe2xyz(Icoes, x)


t=0.d0
Ict=0.d0
Thrust=0.d0

C Open Output Files
C Position
      open(10,file='../../position.2bp', STATUS='REPLACE')
C Velocity
      open(20,file='../../velocity.2bp', STATUS='REPLACE')
C Time
      open(30,file='../../time.2bp', STATUS='REPLACE')
C Current History
      open(40,file='../../curhist.2bp', STATUS='REPLACE')
C Thrust History
      open(50,file='../../Thist.2bp', STATUS='REPLACE')

C Write Outputs to Files
      write(10,*) x(1),x(2),x(3)
      write(20,*) x(4),x(5),x(6)
      write(30,*) t
      write(40,*) Ict
      write(50,*) Thrust


      ido=1
      param(4)=10000000.d0    ! Maximum Number of Iterations for
                              ! Numerical Integrator
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c START INTEGRATION LOOP AND DO THE STEPPING
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

do step=1, num_steps

    write(999,'(\,A1)') char(13)

    call xyz2coe(x, Intcoes)

C Ouput Current COEs to Screen
    write (*,*) 'Current COEs '
    write (*,*) '------------------------------'
    write(*,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)
    write(*,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)
    write(*,'(A21,F11.5)') 'Inclination        : ',Intcoes(3)*180.d0/pi
    write(*,'(A21,F11.5)') 'Arg. of Perigee    : ',Intcoes(4)*180.d0/pi
    write(*,'(A21,F11.5)') 'R. Ascending Node : ',Intcoes(5)*180.d0/pi
    write(*,'(A21,F11.5)') 'Phase              : ',Intcoes(6)*180.d0/pi
    write(*,*)

C Ouput Current COEs to File
    write (101,*) 'Current COEs '
    write (101,*) '------------------------------'
    write(101,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)
    write(101,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)
    write(101,'(A21,F11.5)') 'Inclination        : ',Intcoes(3)*180.d0/pi
    write(101,'(A21,F11.5)') 'Arg. of Perigee    : ',Intcoes(4)*180.d0/pi
    write(101,'(A21,F11.5)') 'R. Ascending Node : ',Intcoes(5)*180.d0/pi
    write(101,'(A21,F11.5)') 'Phase              : ',Intcoes(6)*180.d0/pi
    write(101,*)

    pe=2.d0*pi/dsqrt(mu/Intcoes(1)**3.d0)

C Calculate Remaining Delta COEs for the Rest of the Manuever
    Intdeltas=reshape((/(Tcoes(1)-Intcoes(1)),(Tcoes(2)-Intcoes(2)),
   &           (Tcoes(4)-Intcoes(4)),(Tcoes(3)-Intcoes(3)),
   &           (Tcoes(5)-Intcoes(5))/),(/5,1/))

C If the Manuever is Greater than 180 deg, Go Other Way Around
    do i=3,5
        if (abs(Intdeltas(i,1))>pi) then
            Intdeltas(i,1)=sign((2*pi-abs(Intdeltas(i,1))),-Intdeltas(i,1))
        end if
    end do

C Evaluate the Guidance Matrix
    call eval_matrix(Intcoes,intmatr)

C Take The Inverse of the Matrix
    call invert(intmatr,tempmatr,5)

C Calculate Tether Current Law Coefficients
    currents=pe/(tf-t)/Ic*matmul(tempmatr,Intdeltas)


    k=1
```

```fortran
      maxI=0
      r10=int(Icoes(4)*180.d0/pi)*k        ! Arg. of Perigee Range
      r1f=int(Tcoes(4)*180.d0/pi)*k

      if (r10>r1f) then
          r1=r10
          r10=r1f-5
          r1f=r1+5
          else
          r10=r10-5
          r1f=r1f+5
      end if

C Brute-Force Search For Maximum Current
      do i=r10, r1f
          r1=i*pi/180/k
          do j=0, 360*k
              r2=j*pi/180/k
              curI=currents(1,1)+currents(2,1)*dcos(r2)+
     &      currents(3,1)*dsin(r2)-currents(4,1)*dcos(2.d0*(r2+r1))-
     &      currents(5,1)*dsin(2.d0*(r2+r1))
              if (dabs(curI)>maxI) maxI=dabs(curI)
          end do
      end do

C Recalculate Time Of Flight and Current Coefficients
C To Fit the Maximum Current Requirement
      tleft=maxI*(tf-t)
      tf=t+tleft
      currents=pe/(tf-t)/Ic*matmul(tempmatr,Intdeltas)

C Write Step Current Law Coefficients to File
      write(102,'(A7,I3,A8,F16.2,A14,F16.2)') ' step : ',step,' time : ',
     &        t, ' step time : ', ts
      write(102,*)
'_____'
      write(102,*)
      write(102,'(A19,F8.5,A3,F8.5)') ' DC              : ',currents(1,1),
     &        ' ; ',currents(1,1)*Ic
      write(102,'(A19,F8.5,A3,F8.5)') ' Cos(nu)         : ',currents(2,1),
     &        ' ; ',currents(2,1)*Ic
      write(102,'(A19,F8.5,A3,F8.5)') ' Sin(nu)         : ',currents(3,1),
     &        ' ; ',currents(3,1)*Ic
      write(102,'(A19,F8.5,A3,F8.5)') '-Cos[2(nu+w)]    : ',currents(4,1),
     &        ' ; ',currents(4,1)*Ic
      write(102,'(A19,F8.5,A3,F8.5)') '-Sin[2(nu+w)]    : ',currents(5,1),
     &        ' ; ',currents(5,1)*Ic
      write(102,*)

      if (dispmode) then
          write (*,*) '.....press return.....'
          read(*,*)
      end if

C Calculate Step Time
      ts=(tf-t)/(num_steps-step+1)
C Output Step Time
      write(*,'(A7,I3,A8,F16.2,A14,F16.2)') ' step : ',step,' time : ', t, ' step
time : ', ts
      write(101,'(A7,I3,A8,F16.2,A14,F16.2)') ' step : ',step,' time : ', t, '
step time : ', ts
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c START INTEGRATION LOOP
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      write(999,'(I3,A1,\)') 0 ,'%'        ! Process Percentage Display

      Int_st=t                             ! Integration Start Time
      Int_et=t+ts                          ! Integration Stop Time

        do 100 tend=Int_st+100.d0,Int_et,100.d0
           call divprk (ido,neq,fcn,t,tend,tol,param,x)

C Write Outputs to Files
         write(10,*) x(1),x(2),x(3)
         write(20,*) x(4),x(5),x(6)
         write(30,*) t
         write(40,*) Ict
         write(50,*) Thrust

C Calculate Process Percentage and Display
         stepper=int(100*(t-Int_st)/ts)
         if (stepper<>stepperold) then
            write(999,'(\,A1)') char(13)
            write(999,'(I3,A1,\)') stepper ,'%'
            stepperold=stepper
         end if

100    continue

      if (t<Int_et) then

         call divprk (ido,neq,fcn,t,Int_et,tol,param,x)

         write(10,*) x(1),x(2),x(3)
         write(20,*) x(4),x(5),x(6)
         write(30,*) t
         write(40,*) Ict
         write(50,*) Thrust

   end if

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c END INTEGRATION LOOP
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

end do

      close(10)
      close(20)
      close(30)
      close(40)
      close(50)

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c Output Final Results

      call xyz2coe(x, Intcoes)
```

```fortran
      write(999,'(\,A1)') char(13)

C Final COEs
      write (*,*) 'Final COEs '
      write (*,*) '------------------------------'
      write(*,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)
      write(*,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)
      write(*,'(A21,F11.5)') 'Inclination        : ',Intcoes(3)*180.d0/pi
      write(*,'(A21,F11.5)') 'Arg. of Perigee    : ',Intcoes(4)*180.d0/pi
      write(*,'(A21,F11.5)') 'R. Ascending Node : ',Intcoes(5)*180.d0/pi
      write(*,'(A21,F11.5)') 'Phase              : ',Intcoes(6)*180.d0/pi
      write(*,'(A14,F11.2)') 'Final Time : ', t
      write (*,*)

      write (101,*) 'Final COEs '
      write (101,*) '------------------------------'
      write(101,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)
      write(101,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)
      write(101,'(A21,F11.5)') 'Inclination        : ',Intcoes(3)*180.d0/pi
      write(101,'(A21,F11.5)') 'Arg. of Perigee    : ',Intcoes(4)*180.d0/pi
      write(101,'(A21,F11.5)') 'R. Ascending Node : ',Intcoes(5)*180.d0/pi
      write(101,'(A21,F11.5)') 'Phase              : ',Intcoes(6)*180.d0/pi
      write(101,'(A14,F11.2)') 'Final Time : ', t
      write (101,*)

C Errors in the Final Orbit
      write (*,*) 'Error '
      write (*,*) '------------------------------'
      write(*,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)-Tcoes(1)
      write(*,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)-Tcoes(2)
      write(*,'(A21,F11.5)') 'Inclination        : ',(Intcoes(3)-
     &       Tcoes(3))*180.d0/pi
      write(*,'(A21,F11.5)') 'Arg. of Perigee    : ',(Intcoes(4)-
     &       Tcoes(4))*180.d0/pi
      write(*,'(A21,F11.5)') 'R. Ascending Node : ',(Intcoes(5)-
     &       Tcoes(5))*180.d0/pi
      write(*,'(A21,F11.5)') 'Phase              : ',(Intcoes(6)-
     &       Tcoes(6))*180.d0/pi

      write (101,*) 'Error '
      write (101,*) '------------------------------'
      write(101,'(A21,F11.5)') 'Semi M. Axis       : ',Intcoes(1)-Tcoes(1)
      write(101,'(A21,F11.5)') 'Eccentricity       : ',Intcoes(2)-Tcoes(2)
      write(101,'(A21,F11.5)') 'Inclination        : ',(Intcoes(3)-
     &         Tcoes(3))*180.d0/pi
      write(101,'(A21,F11.5)') 'Arg. of Perigee    : ',(Intcoes(4)-
     &         Tcoes(4))*180.d0/pi
      write(101,'(A21,F11.5)') 'R. Ascending Node : ',(Intcoes(5)-
     &         Tcoes(5))*180.d0/pi
      write(101,'(A21,F11.5)') 'Phase              : ',(Intcoes(6)-
     &         Tcoes(6))*180.d0/pi

      write (*,*) '.....press return to exit.....'
      read(*,*)


C Date/Time Stamp
      call date_and_time(VALUES = datetime)
      write(101,'(A9,I2,A1,I2,A1,I4,A3,I2,A1,I2,A1,I2,A9)') '------- ',
     &          datetime(2),'/', datetime(3),'/', datetime(1), ' * ',
     &          datetime(5),':', datetime(6),':', datetime(7),'------- '
```

```
C Maximum Current Reached During the Manuever
      write(102,'(A19,F8.5,A3,F8.5)') 'Maximum Current : ', Imax/dabs(Ic),
     &          ' ; ',Imax
      write(102,*)
      write(102,'(A9,I2,A1,I2,A1,I4,A3,I2,A1,I2,A1,I2,A9)') '------- ',
     &           datetime(2),'/', datetime(3),'/', datetime(1),
     &           ' * ',datetime(5),':', datetime(6),':', datetime(7),'------- '

      close(101)
      close(102)

      stop
      end


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C EQUATIONS OF MOTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine fcn(neq,t,x,xd)

      implicit none

      integer neq
      double precision t, x(neq), xd(neq), r, em(3)
      double precision ap(3), temp

      double precision Hvec(3), rv(3), vv(3), tvec(3), evec(3), nvec(3)
      double precision v, a, e, i, w, Om, nu

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

      double precision norm, dot, as
      double precision I1, I2, I3, I4, I5

      em=[0.d0, 0.d0, 1.d0]         ! Earth's Magnetic Dipole Axis

C Convert Position and Velocity Classical Orbital Elements

      rv=[x(1), x(2), x(3)]         ! Position Vector
      vv=[x(4), x(5), x(6)]         ! Velocity Vector
      r=norm(rv)
      v=norm(vv)

      a=-mu/2.d0/(v**2.d0/2.d0-mu/r)        ! Semi-major Axis

      call cross(rv, vv, Hvec)
      call cross(vv, Hvec, tvec)

      evec=(tvec/mu-rv/r)                   ! Eccentricity Vector

      e=norm(evec)                          ! Eccentircity

      i=dacos(Hvec(3)/norm(Hvec))           ! Inclination

C If Inclined Orbit, Calculate Line of Nodes
      if (dabs(i)>1.d-13) then
```

B-10

```
          nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
      end if


C No Eccentricity and Inclination Case
      if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
          w=0.d0                           ! Argument of Perigee
          Om=0.d0                          ! Argument of Line of Nodes
          nu=datan2(rv(2),rv(1))           ! True Anomaly

C No Eccentricity Case
      else if (dabs(e)<1.d-13) then
          Om=datan2(nvec(2),nvec(1))
          w=0.d0
          nu=dacos(dot(nvec,rv)/r)
          if (rv(3)<0.d0) nu=-nu           ! Quadrant Check

C No Inclination Case
      else if (dabs(i)<1.d-13) then
          Om=0.d0
          w=datan2(evec(2),evec(1))
          nu=real(dacos(dot(evec,rv)/e/r))
          if (dot(rv,vv)<0.d0) nu=-nu

C Inclination and Eccentricity NON-Zero
      else
          Om=datan2(nvec(2),nvec(1))
          w=dacos(dot(nvec,evec)/e)
          if (evec(3)<0.d0) w=-w
          as=dot(evec,rv)
          as=as/e/r

          ! The 'as' value must be smaller than zero (Fortran Error)
          if (as>1.d0) then
              as=1.d0
          end if

          nu=dacos(as)
          if (dot(rv,vv)<0) nu=-nu

      end if

      I1=currents(1,1)                     ! Current Law Coefficients
      I2=currents(2,1)
      I3=currents(3,1)
      I4=currents(4,1)
      I5=currents(5,1)

C Current Law Super Position
      Ict=Ic*(I1+I2*dcos(nu)+I3*dsin(nu)-I4*dcos(2.d0*(nu+w)))
      Ict=Ict+Ic*(-I5*dsin(2.d0*(nu+w)))

C Update Maximum Current Level Reached
      If (dabs(Ict)>Imax) Imax=dabs(Ict)

C Perturbing Tether Acceleration
      temp=(Ict*L/m)*(mum/r**3.d0)*(1.d0/r)
      ap=[temp*rv(2), -temp*rv(1), 0.d0]

C Thrust History
      Thrust=dsqrt(ap(1)**2.d0+ap(2)**2.d0+ap(3)**2.d0)*m
```

```
C Two Body Problem Equations of Motion
      xd(1)=x(4)
      xd(2)=x(5)
      xd(3)=x(6)
      xd(4)=-mu*x(1)/r**3.d0+ap(1)
      xd(5)=-mu*x(2)/r**3.d0+ap(2)
      xd(6)=-mu*x(3)/r**3.d0+ap(3)

      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Routine to Invert a Matrix by Gaussian Elimination
c
c    Ainv=inverse(A)    A(n,n)   Ainv(n,n)
c    Note: sizes maxed at 64 x 64 - re dimension for larger matrices
c          D matrix is extended.
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine invert( A, Ainv, n )

      implicit none

      double precision A(5,5),Ainv(5,5)
      double precision D(5,10)

      integer i, j, k, n, n2
      double precision beta, alpha

C Initialize the Reduction Matrix
      n2 = 2*n
      do 1 i = 1,n
         do 2 j = 1,n
            D(i,j) = A(i,j)
            d(i,n+j) = 0.
2        continue
         D(i,n+i) = 1.
1     continue

C Do the Reduction
      do 3 i = 1,n
         alpha = D(i,i)
         if(alpha .eq. 0.) go to 300
         do 4 j = 1,n2
            D(i,j) = D(i,j)/alpha
4        continue
         do 5 k = 1,n
            if((k-i).eq.0) go to 5
            beta = D(k,i)
            do 6 j = 1,n2
               D(k,j) = D(k,j) - beta*D(i,j)
6           continue
5        continue
3     continue

C Copy Result into Output Matrix
      do 7 i = 1,n
         do 8 j = 1,n
```

```
            Ainv(i,j) = D(i,j+n)
8       continue
7   continue

    return

300   print *,'*** ERROR: Singular matrix ***'
      return
      end


ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   Guidance Matrix
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

    subroutine eval_matrix(coes,i_matr)

    implicit none

    double precision coes(5), i_matr(5,5)
    double precision a, e, w, i0, o, n, pe, C
    double precision mat_line1(5)

    double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
    common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

        a =coes(1)
        e =coes(2)
        i0=coes(3)
        w =coes(4)
        o =coes(5)

        n=dsqrt(mu/a**3.d0)
        pe=2.d0*pi/n

        C=L*mum/(m*n*a**3.d0*(1.d0-e**2.d0)**(5.d0/2.d0))

        i_matr(1,1) = 1.d0/2.d0*C*dcos(i0)*pi*(8.d0+3.d0*e**4.d0
     &               +24.d0*e**2.d0)/n/(-1.d0+e)/(e+1.d0)
        i_matr(1,2) = 2.d0*C*dcos(i0)*e*pi*(3.d0*e**2.d0+4.d0)/
     &               n/(-1.d0+e)/(e+1.d0)
        i_matr(1,3) = 0.d0
        i_matr(1,4) = -1.d0/2.d0*C*e**2.d0*pi*(6.d0+e**2.d0)*
     &               (dcos(i0-2.d0*w)+dcos(i0+2.d0*w))/n/(-1.d0+e)/(e+1.d0)
        i_matr(1,5) = -2.d0*C*dcos(i0)*e**2.d0*pi*dsin(w)*dcos(w)*
     &               (6.d0+e**2.d0)/n/(-1.d0+e)/(e+1.d0)
        i_matr(2,1) = -7.d0/4.d0*C*dcos(i0)*e*pi*(e**2.d0+4.d0)/a/n
        i_matr(2,2) = -C*dcos(i0)*pi*(2.d0+5.d0*e**2.d0)/a/n
        i_matr(2,3) = 0.d0
        i_matr(2,4) = 1.d0/4.d0*C*e*pi*(5.d0+2.d0*e**2.d0)*
     &               (dcos(i0-2.d0*w)+dcos(i0+2.d0*w))/n/a
        i_matr(2,5) = C*dcos(i0)*e*pi*dsin(w)*dcos(w)*(5.d0+2.d0*e**2.d0)/n/a
        i_matr(3,1) = -1.d0/2.d0/n*C*dcos(i0)/a*e**2.d0*pi*dsin(w)*dcos(w)
        i_matr(3,2) = -1.d0/n*C*dcos(i0)/a*e*pi*dsin(w)*dcos(w)
        i_matr(3,3) = -1.d0/2.d0*C*dcos(i0)*pi*(4.d0+e**2.d0+
     &               2.d0*e**2.d0*dcos(w)**2.d0)/n/a/e
        i_matr(3,4) = -1.d0/2.d0*C*dcos(i0)*pi*dsin(w)*
     &               dcos(w)*(10.d0+e**2.d0)/n/a
        i_matr(3,5) = 1.d0/2.d0*C*dcos(i0)*pi*
```

```
     &                (-4.d0+10.d0*dcos(w)**2.d0+e**2.d0*dcos(w)**2.d0)/n/a
          i_matr(4,1) = 1.d0/4.d0*C*dsin(i0)*pi*(4.d0+e**2.d0+
     &                2.d0*e**2.d0*dcos(w)**2.d0)/n/a
          i_matr(4,2) = 1.d0/2.d0*C*dsin(i0)*e*pi*(1.d0+2.d0*dcos(w)**2.d0)/n/a
          i_matr(4,3) = -1.d0/n*C*dsin(i0)/a*e*pi*dsin(w)*dcos(w)
          i_matr(4,4) = -1.d0/2.d0*C*dsin(i0)*pi*(1.d0+e**2.d0*dcos(w)**2.d0)/n/a
          i_matr(4,5) = -1.d0/2.d0/n*C*dsin(i0)/a*e**2.d0*pi*dsin(w)*dcos(w)
          i_matr(5,1) = 1.d0/4.d0*C*dsin(2.d0*w)*pi*e**2.d0/n/a
          i_matr(5,2) = 1.d0/2.d0*C*dsin(2.d0*w)*pi*e/n/a
          i_matr(5,3) = 1.d0/2.d0*C*dcos(2.d0*w)*pi*e/n/a
          i_matr(5,4) = 0.d0
          i_matr(5,5) = -1.d0/4.d0*C*pi*(2.d0+e**2.d0)/n/a

      return

      end

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     Simplified Guidance Matrix
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine eval_matrix_simple(coes,i_matr)

      implicit none

      double precision coes(5), i_matr(5,5)
      double precision a, e, w, i0, o, n, pe, C
      double precision mat_line1(5)

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

          a =coes(1)
          e =coes(2)
          i0=coes(3)
          w =coes(4)
          o =coes(5)

          n=dsqrt(mu/a**3.d0)
          pe=2.d0*pi/n

          C=L*mum/(m*n*a**3.d0*(1.d0-e**2.d0)**(5.d0/2.d0))

          i_matr(1,1) = 4.d0*C*dcos(i0)*pi/n/(-1.d0+e)/(e+1.d0)
          i_matr(1,2) = 8.d0*C*dcos(i0)*e*pi/n/(-1.d0+e)/(e+1.d0)
          i_matr(1,3) = 0.d0
          i_matr(1,4) = 0.d0
          i_matr(1,5) = 0.d0
          i_matr(2,1) = -7.d0*C*dcos(i0)*e*pi/a/n
          i_matr(2,2) = -C*dcos(i0)*pi*(2.d0)/a/n
          i_matr(2,3) = 0.d0
          i_matr(2,4) = 5.d0/4.d0*C*e*pi*(dcos(i0-2.d0*w)+dcos(i0+2.d0*w))/n/a
          i_matr(2,5) = 5.d0*C*dcos(i0)*e*pi*dsin(w)*dcos(w)/n/a
          i_matr(3,1) = 0.d0
          i_matr(3,2) = -1.d0/n*C*dcos(i0)/a*e*pi*dsin(w)*dcos(w)
          i_matr(3,3) = -2.d0*C*dcos(i0)*pi/n/a/e
          i_matr(3,4) = -5.d0*C*dcos(i0)*pi*dsin(w)*dcos(w)/n/a
          i_matr(3,5) = C*dcos(i0)*pi*(-2.d0+5.d0*dcos(w)**2.d0)/n/a
          i_matr(4,1) = C*dsin(i0)*pi/n/a
```

B-14

```
          i_matr(4,2) = 1.d0/2.d0*C*dsin(i0)*e*pi*(1.d0+2.d0*dcos(w)**2.d0)/n/a
          i_matr(4,3) = -1.d0/n*C*dsin(i0)/a*e*pi*dsin(w)*dcos(w)
          i_matr(4,4) = -1.d0/2.d0*C*dsin(i0)*pi/n/a
          i_matr(4,5) = 0.d0
          i_matr(5,1) = 0.d0
          i_matr(5,2) = 1.d0/2.d0*C*dsin(2.d0*w)*pi*e/n/a
          i_matr(5,3) = 1.d0/2.d0*C*dcos(2.d0*w)*pi*e/n/a
          i_matr(5,4) = 0.d0
          i_matr(5,5) = -1.d0/2.d0*C*pi/n/a

      return

      end


ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Clasical Orbital Elements to Position and Velocity
c              a,e,i,w,raan  to r,v (xyz)
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine coe2xyz(coes,rv)

      implicit none

      double precision coes(5), rv(6)

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

      double precision a, e, w, i, o, r, v
      double precision r0(3), v0(3)

      a =coes(1)
      e =coes(2)
      i =coes(3)
      w =coes(4)
      o =coes(5)

C Position
      r=a*(1.d0-e)
C Velocity
      v=dsqrt(2.d0*mu*(1.d0/r-1.d0/2.d0/a))

C Rotation Matrix Multiplication
      r0=r*(/(dcos(w)*dcos(o)-dsin(w)*dcos(i)*dsin(o)),
     &  (dcos(w)*dsin(o)+dsin(w)*dcos(i)*dcos(o)),(dsin(w)*dsin(i))/)
      v0=v*(/(-dsin(w)*dcos(o)-dcos(w)*dcos(i)*dsin(o)),
     &  (-dsin(w)*dsin(o)+dcos(w)*dcos(i)*dcos(o)),(dcos(w)*dsin(i))/)

      rv(1)=r0(1)
      rv(2)=r0(2)
      rv(3)=r0(3)
      rv(4)=v0(1)
      rv(5)=v0(2)
      rv(6)=v0(3)

      return

      end
```

```fortran
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Position and Velocity to Classical Orbital Elements
c            r,v (xyz) to a,e,i,w,raan
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine xyz2coe(rvx,coes)

      implicit none

      double precision rvx(6), coes(5)

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

      double precision a, e, w, i, Om, r, v, nu, as
      double precision rv(3), vv(3), Hvec(3), tvec(3), evec(3), nvec(3)
      double precision norm, dot


      rv=[rvx(1), rvx(2), rvx(3)]
      vv=[rvx(4), rvx(5), rvx(6)]
      r=norm(rv)
      v=norm(vv)

      a=-mu/2.d0/(v**2.d0/2.d0-mu/r)

      call cross(rv, vv, Hvec)
      call cross(vv, Hvec, tvec)
      evec=(tvec/mu-rv/r)
      e=norm(evec)
      i=dacos(Hvec(3)/norm(Hvec))
      if (dabs(i)>1.d-13) then
          nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
      end if
      if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
          w=0.d0
          Om=0.d0
          nu=datan2(rv(2),rv(1))
      else if (dabs(e)<1.d-13) then
          Om=datan2(nvec(2),nvec(1))
          w=0.d0
          nu=dacos(dot(nvec,rv)/r)
          if (rv(3)<0.d0) nu=-nu
      else if (dabs(i)<1.d-13) then
          Om=0.d0
          w=datan2(evec(2),evec(1))
          nu=real(dacos(dot(evec,rv)/e/r))
          if (dot(rv,vv)<0.d0) nu=-nu
      else
          Om=datan2(nvec(2),nvec(1))
          w=dacos(dot(nvec,evec)/e)
          if (evec(3)<0.d0) w=-w
          as=dot(evec,rv)
          as=as/e/r
          if (as>1.d0) then
              as=1.d0
```

B-16

```fortran
          end if
          nu=dacos(as)
          if (dot(rv,vv)<0) nu=-nu
      end if
      rv=[x(1), x(2), x(3)]        ! Position Vector
      vv=[x(4), x(5), x(6)]        ! Velocity Vector
      r=norm(rv)
      v=norm(vv)

      a=-mu/2.d0/(v**2.d0/2.d0-mu/r)      ! Semi-major Axis

      call cross(rv, vv, Hvec)
      call cross(vv, Hvec, tvec)

      evec=(tvec/mu-rv/r)                 ! Eccentricity Vector

      e=norm(evec)                        ! Eccentircity

      i=dacos(Hvec(3)/norm(Hvec))         ! Inclination

C If Inclined Orbit, Calculate Line of Nodes
      if (dabs(i)>1.d-13) then
          nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
      end if

C No Eccentricity and Inclination Case
      if (dabs(e)<1.d-13 .AND. (dabs(i)<1.d-13)) then
          w=0.d0                          ! Argument of Perigee
          Om=0.d0                         ! Argument of Line of Nodes
          nu=datan2(rv(2),rv(1))          ! True Anomaly

C No Eccentricity Case
      else if (dabs(e)<1.d-13) then
          Om=datan2(nvec(2),nvec(1))
          w=0.d0
          nu=dacos(dot(nvec,rv)/r)
          if (rv(3)<0.d0) nu=-nu          ! Quadrant Check

C No Inclination Case
      else if (dabs(i)<1.d-13) then
          Om=0.d0
          w=datan2(evec(2),evec(1))
          nu=real(dacos(dot(evec,rv)/e/r))
          if (dot(rv,vv)<0.d0) nu=-nu

C Inclination and Eccentricity NON-Zero
      else
          Om=datan2(nvec(2),nvec(1))
          w=dacos(dot(nvec,evec)/e)
          if (evec(3)<0.d0) w=-w
          as=dot(evec,rv)
          as=as/e/r

          ! The 'as' value must be smaller than zero (Fortran Error)
          if (as>1.d0) then
              as=1.d0
          end if

          nu=dacos(as)
          if (dot(rv,vv)<0) nu=-nu
```

B-17

```
      end if

      coes(1) = a
      coes(2) = e
      coes(3) = i
      coes(4) = w
      coes(5) = Om

C Avoid Negative Angles
      do i=3,5
          if (coes(i)<0) then
              coes(i)=2*pi+coes(i)
          end if
      end do

      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Vector Cross Product
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine cross(v1, v2, v3)

      implicit none

      double precision v1(3), v2(3), v3(3)

      v3(1) = v1(2) * v2(3) - v1(3) * v2(2)
      v3(2) = v1(3) * v2(1) - v1(1) * v2(3)
      v3(3) = v1(1) * v2(2) - v1(2) * v2(1)

      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Vector Dot Product
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      double precision function dot(v1, v2)
      implicit none

      double precision v1(3), v2(3)

      dot=v1(1)*v2(1)+v1(2)*v2(2)+v1(3)*v2(3)

      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Vector Magnitude
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
double precision function norm(v3)
implicit none

double precision v3(3)

norm=dsqrt(v3(1)**2.d0+v3(2)**2.d0+v3(3)**2.d0)

return

end
```

Appendix C. Nonlinear Solution For Tether Current Law Coefficients

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C NonLinearSolution.f90
C
C      NONLINEAR SOLUTION FOR TETHER CURRENT LAW COEFFICIENTS
C
C               Lt Hakan San - AFIT/ENY/GA-02M
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      implicit none

      integer         itMax, NLF, i
      parameter       (NLF=6)

      double precision  errRel
      double precision  fNorm, XNL(6), XNLGuess(6)

      double precision  tf, hr, min, sec
      double precision  tempc(5), tempx(6)

      double precision  mu, mum, m, L, Ic, simTol, targetCOEs(5), inrv(6),
     &                  currents(5), maxI
      common            mu, mum, m, L, Ic, simTol, targetCOEs, inrv, currents,
     &                  maxI,tf

      external          dneqnf, divprk, fcn, norm, dot, NLFCN


      pi = 3.141592653589793238462643383279502884197169399375510d0

      simTol=1.d-10            ! Numerical Integration Tolerance

C Parameters
      mu=3.986d5               ! Earth's Gravitaional Constant
      mum=8.0711d6             ! Earth's Magnetic Dipole Moment
      Ic=-5.d0                 ! Applied Tether Current
      L=15.d0                  ! Tether Length
      m=1000.d0                ! System Mass

C Read Initial Position and Velocity From File (Cartesian Coordinates)
      write (*,*) 'initial position and velocity'
      i=1
      open (1, file = 'initial.2bp')
      do while (.NOT. (eof(1) .OR. (i==7)))
          read (1,*) inrv(i)
          write (*,*) inrv(i)
          i=i+1
      end do
      close(1)

C Convert Initial r & v to COEs and Output
      write(*,*) 'initial coes'
      call xyz2coe(inrv,tempc)
      write (*,*) tempc(1)
      write (*,*) tempc(2)
      write (*,*) tempc(3)
      write (*,*) tempc(4)
```

```fortran
      write (*,*) tempc(5)



C Read Current Coefficients Generated by Guidance Algorithm as Initial Guess
      write (*,*) 'currents'
      i=1
      open (2, file = 'currents.2bp')
      do while (.NOT. (eof(2) .OR. (i==6)))
          read (2,*) currents(i)
          write (*,*) currents(i)
          i=i+1
      end do

C Read Time of Flight Guess
      write (*,*) 'total time'
      read (2,*) tf
      hr=int(tf/3600)
      min=int(mod(tf,3600.d0)/60)
      sec=tf-hr*3600-min*60
      write (*,*) tf
      write (*,*) hr, 'hr', min, 'min', sec, 'sec'
      close(2)

C Read Target COEs
      write (*,*) 'target coe'
      i=1
      open (3, file = 'target.coe')
      do while (.NOT. (eof(3) .OR. (i==6)))
          read (3,*) targetCOEs(i)
          write (*,*) targetCOEs(i)
          i=i+1
      end do
      close(3)

C Set Initial Guess Vector
C   XGUESS=( I1 I2 I3 I4 I5 tf )

      XNLGuess(1)=currents(1)
      XNLGuess(2)=currents(2)
      XNLGuess(3)=currents(3)
      XNLGuess(4)=currents(4)
      XNLGuess(5)=currents(5)
      XNLGuess(6)=Ic

      errRel = 1.0d-6          ! Relative Error for Nonlinear Equation Solver
      itMax  = 20000           ! Maximum Number of Iterations for NL Eq. Sol.

C Call the Solver
      call DNEQNF (NLFCN, errRel, NLF, itMax, XNLGuess, XNL, FNORM)

C Output the Results of Nonlinear Solution (Current Coefficients)
      write (*,*), 'Currents'
      write (*,*),XNL(1)
      write (*,*),XNL(2)
      write (*,*),XNL(3)
      write (*,*),XNL(4)
      write (*,*),XNL(5)
      write (*,*),'time: ', XNL(6)

C Norm of the Solution
```

C-2

```fortran
      write (*,*),'fnorm', FNORM



C Output to File
      open(10,file='nl-currents.2bp', STATUS='REPLACE')

      write (10,*) XNL(1)
      write (10,*) XNL(2)
      write (10,*) XNL(3)
      write (10,*) XNL(4)
      write (10,*) XNL(5)
      write (10,*) XNL(6)

      close(10)

      read(*,*)

      end



cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c NONLINEAR EQUATION SOLVER
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

SUBROUTINE NLFCN (XNL, F, NLF)
      integer NLF
      double precision    XNL(NLF), F(NLF)


      integer           mxparm, neq
      parameter         (mxparm=50, neq=6)

      double precision  x(neq),param(mxparm),t,tend
      double precision  tf, hr, min, sec
      double precision  rv(3), vv(3), Hvec(3), r, v, a, e, i, w, Om, evec(3),
      &                 nvec(3), Itot,as
      double precision  tvec(3), intCoes(5)

      double precision  mu, mum, m, L, Ic, simTol, targetCOEs(5), inrv(6),
      &                 currents(5), maxI
      common            mu, mum, m, L, Ic, simTol, targetCOEs, inrv, currents,
      &                 maxI, tf
      external          divprk, fcn

      t=0.d0

      param(4)=10000000.d0          ! Maximum Number of Iterations
                                    ! for Numerical Integrator
      x(1)=inrv(1)
      x(2)=inrv(2)
      x(3)=inrv(3)
      x(4)=inrv(4)
      x(5)=inrv(5)
      x(6)=inrv(6)

      currents(1)=XNL(1)            ! Variables for NL Eq.
      currents(2)=XNL(2)
```

C-3

```fortran
      currents(3)=XNL(3)
      currents(4)=XNL(4)
      currents(5)=XNL(5)
      Ic=XNL(6)
      maxI=0

C Run the Numerical Integrator
      ido=1
      call divprk (ido,neq,fcn,t,tf,simTol,param,x)
      ido=3
      call divprk (ido,neq,fcn,tf,tf,simTol,param,x)

      call xyz2coe(x,intCoes)

C Nonlinear Equations
      F(1) = targetCOEs(1) - IntCoes(1)
      F(2) = targetCOEs(2) - IntCoes(2)
      F(3) = targetCOEs(3) - IntCoes(3)
      F(4) = targetCOEs(4) - IntCoes(4)
      F(5) = targetCOEs(5) - IntCoes(5)
      F(6) = maxI-1

C Output Current Errors in the Final Orbit
      write(*,*)'--------------------'
      write(*,*)'a',F(1)
      write(*,*)'e',F(2)
      write(*,*)'w',F(3)
      write(*,*)'i',F(4)
      write(*,*)'o',F(5)
      write(*,*)'I',XNL(6)
      write(*,*)'t',tf

      return
      end

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C EQUATIONS OF MOTION
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine fcn(neq,t,x,xd)

      integer neq
      double precision t, x(neq), xd(neq), r, em(3), Ict
      double precision ap(3), temp

      double precision Hvec(3), rv(3), vv(3), tvec(3), evec(3), nvec(3)
      double precision v, a, e, i, w, Om, nu

      double precision mu, mum, m, L, Ic, simTol, targetCOEs(5), inrv(6),
     &                 currents(5), maxI
      common           mu, mum, m,L, Ic, simTol, targetCOEs, inrv, currents, maxI
      double precision n, as

      double precision norm, dot
      double precision I1, I2, I3, I4, I5

      em=[0.d0, 0.d0, 1.d0]        ! Earth's Magnetic Dipole Axis

C Convert Position and Velocity Classical Orbital Elements
```

```fortran
    rv=[x(1), x(2), x(3)]        ! Position Vector
    vv=[x(4), x(5), x(6)]        ! Velocity Vector
    r=norm(rv)
    v=norm(vv)

    a=-mu/2.d0/(v**2.d0/2.d0-mu/r)       ! Semi-major Axis

    call cross(rv, vv, Hvec)
    call cross(vv, Hvec, tvec)

    evec=(tvec/mu-rv/r)                  ! Eccentricity Vector

    e=norm(evec)                         ! Eccentircity

    i=dacos(Hvec(3)/norm(Hvec))          ! Inclination

C If Inclined Orbit, Calculate Line of Nodes
    if (dabs(i)>1.d-13) then
        nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
    end if

C No Eccentricity and Inclination Case
    if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
        w=0.d0                           ! Argument of Perigee
        Om=0.d0                          ! Argument of Line of Nodes
        nu=datan2(rv(2),rv(1))           ! True Anomaly

C No Eccentricity Case
    else if (dabs(e)<1.d-13) then
        Om=datan2(nvec(2),nvec(1))
        w=0.d0
        nu=dacos(dot(nvec,rv)/r)
        if (rv(3)<0.d0) nu=-nu           ! Quadrant Check

C No Inclination Case
    else if (dabs(i)<1.d-13) then
        Om=0.d0
        w=datan2(evec(2),evec(1))
        nu=real(dacos(dot(evec,rv)/e/r))
        if (dot(rv,vv)<0.d0) nu=-nu

C Inclination and Eccentricity NON-Zero
    else
        Om=datan2(nvec(2),nvec(1))
        w=dacos(dot(nvec,evec)/e)
        if (evec(3)<0.d0) w=-w
        as=dot(evec,rv)
        as=as/e/r

        ! The 'as' value must be smaller than zero (Fortran Error)
        if (as>1.d0) then
            as=1.d0
        end if

        nu=dacos(as)
        if (dot(rv,vv)<0) nu=-nu

    end if

    I1=currents(1)                       ! Current Law Coefficients
```

```fortran
      I2=currents(2)
      I3=currents(3)
      I4=currents(4)
      I5=currents(5)
C Current Law Super Position

      Ict=Ic*(I1+I2*dcos(nu)+I3*dsin(nu)-I4*dcos(2.d0*(nu+w))-
     &        I5*dsin(2.d0*(nu+w)))

C Maximum Current Level Reached
      if(dabs(Ict/Ic)>maxI) then
          maxI=dabs(Ict/Ic)
      end if

C Perturbing Tether Acceleration
      temp=(Ict*L/m)*(mum/r**3.d0)*(1.d0/r)
      ap=[temp*rv(2), -temp*rv(1), 0.d0]

C Two Body Problem Equations of Motion
      xd(1)=x(4)
      xd(2)=x(5)
      xd(3)=x(6)
      xd(4)=-mu*x(1)/r**3.d0+ap(1)
      xd(5)=-mu*x(2)/r**3.d0+ap(2)
      xd(6)=-mu*x(3)/r**3.d0+ap(3)

      return

      end


cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Position and Velocity to Classical Orbital Elements
c          r,v (xyz) to a,e,i,w,raan
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine xyz2coe(rvx,coes)

      implicit none

      double precision rvx(6), coes(5)

      double precision mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents(5,1)
      common mu, mum, m, L, Ic, pi, Ict, Thrust, Imax, currents

      double precision a, e, w, i, Om, r, v, nu, as
      double precision rv(3), vv(3), Hvec(3), tvec(3), evec(3), nvec(3)
      double precision norm, dot


      rv=[rvx(1), rvx(2), rvx(3)]
      vv=[rvx(4), rvx(5), rvx(6)]
      r=norm(rv)
      v=norm(vv)

      a=-mu/2.d0/(v**2.d0/2.d0-mu/r)

      call cross(rv, vv, Hvec)
      call cross(vv, Hvec, tvec)
      evec=(tvec/mu-rv/r)
      e=norm(evec)
```

```fortran
    i=dacos(Hvec(3)/norm(Hvec))
    if (dabs(i)>1.d-13) then
        nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
    end if
    if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
        w=0.d0
        Om=0.d0
        nu=datan2(rv(2),rv(1))
    else if (dabs(e)<1.d-13) then
        Om=datan2(nvec(2),nvec(1))
        w=0.d0
        nu=dacos(dot(nvec,rv)/r)
        if (rv(3)<0.d0) nu=-nu
    else if (dabs(i)<1.d-13) then
        Om=0.d0
        w=datan2(evec(2),evec(1))
        nu=real(dacos(dot(evec,rv)/e/r))
        if (dot(rv,vv)<0.d0) nu=-nu
    else
        Om=datan2(nvec(2),nvec(1))
        w=dacos(dot(nvec,evec)/e)
        if (evec(3)<0.d0) w=-w
        as=dot(evec,rv)
        as=as/e/r
        if (as>1.d0) then
            as=1.d0
        end if
        nu=dacos(as)
        if (dot(rv,vv)<0) nu=-nu
    end if
    rv=[x(1), x(2), x(3)]        ! Position Vector
    vv=[x(4), x(5), x(6)]        ! Velocity Vector
    r=norm(rv)
    v=norm(vv)

    a=-mu/2.d0/(v**2.d0/2.d0-mu/r)        ! Semi-major Axis

    call cross(rv, vv, Hvec)
    call cross(vv, Hvec, tvec)

    evec=(tvec/mu-rv/r)                    ! Eccentricity Vector

    e=norm(evec)                           ! Eccentircity

    i=dacos(Hvec(3)/norm(Hvec))            ! Inclination

C If Inclined Orbit, Calculate Line of Nodes
    if (dabs(i)>1.d-13) then
        nvec=[-Hvec(2), Hvec(1), 0.d0]/norm([-Hvec(2), Hvec(1), 0.d0])
    end if

C No Eccentricity and Inclination Case
    if (dabs(e)<1.d-13 .AND.  (dabs(i)<1.d-13)) then
        w=0.d0                              ! Argument of Perigee
        Om=0.d0                             ! Argument of Line of Nodes
        nu=datan2(rv(2),rv(1))              ! True Anomaly

C No Eccentricity Case
    else if (dabs(e)<1.d-13) then
        Om=datan2(nvec(2),nvec(1))
        w=0.d0
```

```
            nu=dacos(dot(nvec,rv)/r)
            if (rv(3)<0.d0) nu=-nu              ! Quadrant Check

C No Inclination Case
        else if (dabs(i)<1.d-13) then
            Om=0.d0
            w=datan2(evec(2),evec(1))
            nu=real(dacos(dot(evec,rv)/e/r))
            if (dot(rv,vv)<0.d0) nu=-nu

C Inclination and Eccentricity NON-Zero
        else
            Om=datan2(nvec(2),nvec(1))
            w=dacos(dot(nvec,evec)/e)
            if (evec(3)<0.d0) w=-w
            as=dot(evec,rv)
            as=as/e/r

            ! The 'as' value must be smaller than zero (Fortran Error)
            if (as>1.d0) then
                as=1.d0
            end if

            nu=dacos(as)
            if (dot(rv,vv)<0) nu=-nu

        end if

        coes(1) = a
        coes(2) = e
        coes(3) = i
        coes(4) = w
        coes(5) = Om

C Avoid Negative Angles
        do i=3,5
            if (coes(i)<0) then
                coes(i)=2*pi+coes(i)
            end if
        end do

        return

        end

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    Vector Cross Product
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        subroutine cross(v1, v2, v3)

        implicit none

        double precision v1(3), v2(3), v3(3)

        v3(1) = v1(2) * v2(3) - v1(3) * v2(2)
        v3(2) = v1(3) * v2(1) - v1(1) * v2(3)
        v3(3) = v1(1) * v2(2) - v1(2) * v2(1)
```

```fortran
      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   Vector Dot Product
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      double precision function dot(v1, v2)
      implicit none

      double precision v1(3), v2(3)

      dot=v1(1)*v2(1)+v1(2)*v2(2)+v1(3)*v2(3)

      return

      end

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c   Vector Magnitude
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      double precision function norm(v3)
      implicit none

      double precision v3(3)

      norm=dsqrt(v3(1)**2.d0+v3(2)**2.d0+v3(3)**2.d0)

      return

      end
```

Bibliography

1. Cosmo, M.L. and Lorenzini, E.C., *Tethers in Space Handbook,* 3rd. Ed., Report for NASA Marshall Space Flight Center Grant NAG8-1160, 1997.

2. Vas, I.E., Kelly, T.J., and Scarl, E.A., *Journal of Spacecraft and Rockets,* Vol. 37, No. 2, March 2000.

3. Gilchrist, B.E., Johnson, L., and Bilen, S.G., "Space Electrodynamic Tether Propulsion Technology: System Considerations and Future Plans," *35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Paper AIAA 99-2841, June 1999.

4. Hoyt R.P., Forward R.L., "The Terminator Tether: Autonomous Deorbit of LEO Spacecraft for Space Debris Mitigation", *38th AIAA Aerospace Sciences Conference*, Paper AIAA 00-0329, January 2000.

5. Lorenzini, E.C., "In-Space Transportation with Tethers," *NASA Report* 19990100963, 1999.

6. Santangelo, A., Johnson, L., Gilchrist, B., Hoffman, J., Hoffman, J., Lorenzini, E., Estes, R., "Advancing Electrodynamic Tethers to Commercially Viable Systems-STEP-AIRSEDS," *Proceedings of the NASA/JPL/MSFC/AIAA 10th Annual Advanced Space Propulsion Workshop*, Vol. 2, 1999, pp. 602-622.

7. Estes, R.D., Lorenzini, E. C., Martinez-Sanchez, M., Sanmartin, J.R., Savich, N.A., "New High-Current Tethers: A Viable Power Source for Space Station?,",White Paper, December 1995, http://cfa-www.harvard.edu/~spgroup/., February 2002

8. Gilchrist, B.E., Johnson L., Lorenzini E., "Electrodynamic Tethers as Propulsion Systems: System Considerations and Future Plans," *6th Spacecraft Charging Technology Conference*, AFRL-VS-TR-200001578, September 2000.

9. Tascione, T.F., *Intoduction to the Space Environment*, 2nd. Ed., Krieger Publishing Company, Malabar, 1994.

10. Johnson L., "The Tether Solution," *IEEE Spectrum*, Vol. 37, No. 7, July, 2000, NASA.

11. Moris, D., Gilchrist, B.E., Gillmore, A., Jensen, K., "Developing Field Emitter Array Cathode Systems for Electrodynamic Tether Propulsion," *36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Paper AIAA 2000-3867, July 2000.

12. Choinière, E., Gilchrist, B.E., Bilén, S.G., "Enhancement of Electrodynamic Tether Electron Current Collection Using Radio Frequency Power", *Space Technology & Applications International Forum 2001*, Albuquerque, New Mexico, February 2001

13. Yamagiwa, Y., "Performance Analysis of Electrodynamic Tether Orbit Transfer System for Round Trip Mission," *Journal of Space Technology and Science*, Vol. 13, No. 1, 177, pp. 17-25

14. Dooling D., "Up, Up, and Away (bit by bit)," http://science.nasa.gov/newhome/headlines/ast08sep97_1.htm, February 2002

15. Carroll, J.A., Guidebook for Analysis of Tether Applications, *Report for Martin Marietta Contract No.* RH4394049, 1985.

16. West, B., Gilchrist, B.E., "Life Extension and Orbit Maneuvering Strategies for Small Satellites in Low Earth Orbit Using Electrodynamic Tethers", *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, January 2001.

17. Stone N.H., Bonifazi C., "The TSS-1R Mission: Overview and scientific context," http://www.geophys.washington.edu/Space/GRL/articles/25/4/stone/text.pdf, February 2002

18. "Advanced Space Transportation Technology Summary, Electrodynamic Tether Propulsion," Marshall Space Flight Center, http://std.msfc.nasa.gov/news/Tether_propulsion.pdf, February 2002 (date seen on web page).

19. The Microsatellite Propellantless Electrodynamic Tether Propulsion System, Tethers Unlimited Inc., http://www.tethers.com, February 2002.

20. Palaex J., Lopez-Rebollal O., Ruiz M., Lorenzini E.C., "Damping in Rigid Electrodynamic Tethers on Inclined Orbits," *AAS/AIAA Space Flight Mechanics Meeting*, Paper AAS 01-190, February 2001.

21. Yamagiwa, Y., "An Analytical Study of Performance of Electrodynamic Tether Orbit Transfer System," *21st International Symposium on Space Technology and Science Proceedings*, Vol. 2, pp. 1183-1188.

22. Martinez-Sanchez M., Hastings D.E., "A Systems Study of a 100 KW Electrodynamic Tether," *Advances in the Astronautical Sciences*, Vol. 68, Paper AAS 86-224.

23. Gaylor, D.E., "Analysis of Low Thrust Orbit Transfers Using the Lagrange Planetary Equations," Lightsey Research Group at the University of Texas Center for Space Research, December 2000, http://gps.csr.utexas.edu/gaylor_pub/cm1project.pdf., February 2002

24. Dewell, L.D., Menon P.K., "Low-Thrust Orbit Transfer Optimization Using Genetic Search," *AIAA Guidance, Navigation and Control Conference*, Paper AIAA 99-4151, 1999.

25. Kluever, C.A., "Simple Guidance Scheme for Low-Thrust Orbit Transfers," *AAS/AIAA Space Flight Mechanics Meeting*, Paper AAS 98-203, July, 1998.

26. Herbiniere, S., Vincent, M., "Direct Approach for Low-Thrust Positioning Computation Under Satellite Constraints," *AAS/AIAA Space Flight Mechanics Meeting*, Paper AAS 01-211, February 2001.

27. Edminister, J.A., *Schaum's Outline of Electromagnetics,* 2nd Ed., McGraw-Hill Companies Inc., New York, 1993

28. Chobotov, V.A., Ed., *Orbital Mechanics*, 2nd Ed., American Institute of Aeronautics and Astronautics, Inc., 1996.

29. Kelso, T.S., "Celestrak", http://www.celestrak.com/.

Vita

Lieutenant Hakan San was born in 1977 in Bursa, Turkiye. He attended Bursa Anatolian High School in Bursa, Turkiye. During high school, he was awarded second place in his region in the National Informatics Olympics of 1994. Upon graduation in 1995, he entered undergraduate studies at the Turkish Air Force Academy in Istanbul, Turkiye where he graduated with a Bachelor Degree of Science in Aeronautical Engineering in August 1999.

His first assignment was at Air Corps Schools and Technical Training Center, Izmir, Turkiye as a student officer in Aircraft Maintenance School in August 1999. In September 2000, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology at Wright Patterson AFB, OH. During his study in AFIT, he was selected to be a member of Tau Beta Pi-Engineering Honor Society and Sigma Gama Tau-National Honor Society in Aerospace Engineering.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
26-03-2002	**Master's Thesis**	Sep 2000 – Mar 2002

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
ORBITAL MANEUVERING USING ELECTRODYNAMIC TETHERS	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
San, Hakan, 2Lt, TuAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	8. PERFORMING ORGANIZATION REPORT NUMBER
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 P Street, Building 640 WPAFB OH 45433-7765	AFIT/GA/ENY/02-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
Star Technology and Research, Inc. 3213 Carmel Bay Drive, Suite 200 Mount Pleasant, SC 29466-8513	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

.

14. ABSTRACT

Considerably short electrodynamic tethers can be used as a low-thrust propulsion system with minimum or no mass expenditure. The propulsive force is generated by inducing a current flow through the tether which interacts with the magnetosphere of the planet (i.e. Earth, Jupiter and other planetary bodies with magnetosphere). The basics of electrodynamic tether systems have been studied and successful experiments such as boost and de-boost of spacecraft have been conducted in the past. This study presents a simple guidance scheme for the current in the tether in order to perform orbital maneuvers. The general perturbation equations are used to develop the guidance scheme algorithm. The tether is assumed to be perfectly aligned with the local vertical and the tether flexibility is neglected. The guidance is capable of both in-plane and out-of-plane maneuvers, simultaneously changing the orbit parameters. Several numerical examples are simulated that demonstrate the ability of the guidance to accurately maneuver the vehicle. The simplicity of the guidance law allows it to be suitable for mission planning and on-board implementation.

15. SUBJECT TERMS

Electrodynamic Tether, Orbital Manuevering, Guidance

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
			UU	134	Steven G. Tragesser, AFIT (ENY)
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U			785-6565x 4286 Steven.Tragesser@Afit.Edu