CUBESAT ATTITUDE CONTROL UTILIZING LOW-POWER MAGNETIC TORQUERS & A MAGNETOMETER

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

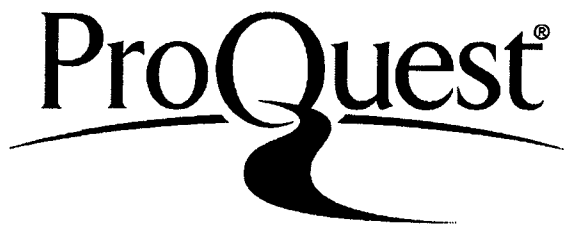
Doctor of Philosophy

By

Donald B. Mentch, B.S., M.S.

Fairbanks, Alaska

August 2011



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

CUBESAT ATTITUDE CONTROL UTILIZING LOW-POWER MAGNETIC TORQUERS & A MAGNETOMETER

By

Donald B. Mentch

RECOMMENDED:

Joseph S. Hawkins

Michael E. Polites

Denise Thorsen

Advisory Committee Chair

Charles E Mayer

Chair, Department of Electrical and Computer Engineering

APPROVED:

Dean, College of Engineering and Mines

Dean of the Graduate School

August 1, 2011

Date

Abstract

The CubeSat Project has lowered development time and costs associated with university satellite missions that conform to their 10 centimeter cube design specification. Providing attitude control to a spacecraft, of such small volume, with a very limited power budget has been a challenge around the world.

This work describes the development of an attitude control system based on a very low-power magnetic torquer used in conjunction with a magnetometer. This will be the first flight use of this torquer which is composed of a hard magnetic material wrapped inside of a solenoid. By discharging a capacitor through the solenoid, the magnetic dipole moment of this permanent magnet can be reversed. The completed attitude control system will make the first use of the low-power magnetic torquer to arrest satellite tip-off rates. It will then make the first known use of a dual axis magnetic dipole moment bias algorithm to achieve three-axis attitude alignment. The complete system is standalone for high inclination orbits, and will align the spacecraft to within 5 degrees of ram, nadir, and local vertical, without any requirement for attitude determination. The system arrests tip-off rates of up to 5° per second (in all 3 axes) for a satellite in a 600 kilometer polar orbit expending 0.56 milliwatts of power. Once in the proper alignment, it utilizes 0.028 milliwatts to maintain it. The system will function for low inclination orbits with the addition of a gravity boom.

The system utilizes the magnetometer to calculate spacecraft body rates. This is the only known use of a magnetometer to directly measure spacecraft body rates without prior knowledge of spacecraft attitude.

Table of Contents

Page

Signature Page i

Title Page ii

Abstract iii

Table of Contents iv

List of Figures xi

List of Tables xiii

List of Other Materials xiv

List of Appendices xv

Chapter 1 Introduction 1

1.1 Purpose of the Research 1

1.2 CubeSat Requirements 2

1.3 Reference Frames 3

1.3.1 Earth-Centered Inertial 4

1.3.2 Earth-Centered Earth-Fixed 5

1.3.3 Local North East Down 6

1.3.4 Orbital 6

1.3.5 Spacecraft Body-Fixed 7

1.3.6 Spacecraft Body Rates 8

1.4 Reference Frame Transformations 9

1.4.1 Definition of Variables 9

1.4.2 ECI to ECEF 10

1.4.3 ECEF to LNED 10

Page

1.4.4 ECI to Orbital 11
1.4.5 Quaternions 11
1.4.6 ECI to body 11
1.5 CubeSat Attitude Control Literature Review 12
1.5.1 No Attitude Control 12
1.5.2 Permanent Magnets 12
1.5.3 Open Air / Vacuum Coils 14
1.5.4 Torque Rods / Hysteresis Rods 15
1.5.5 Reaction Wheels 16
1.5.6 Control Moment Gyros 18
1.5.7 Gravity Boom 19
1.5.8 Reaction Control System 19
1.5.9 Attitude Determination 20
1.5.10 $\dot{B}$ Controller 20
1.6 Validation of Low-Power Magnetic Torquers 21
1.6.1 Description of LPMT 21
1.6.2 Hard Magnetic Material Core 21
1.6.3 Basic Stabilization System 23
1.6.4 Stabilization Control Logic 23
1.7 Geomagnetic Field of the Earth 26
1.7.1 Torquer Biased Alignment 26
1.7.2 International Geomagnetic Reference Field 27
1.7.3 Polar Regions 28

Page

1.7.4 Magnetic Flux Densities 29
1.7.5 South Atlantic Anomaly 31
1.8 Attitude Disturbance Torques 32
1.8.1 Aerodynamic Drag 32
1.8.2 Gravity Gradient 34
1.8.3 Solar Radiation Pressure 35
1.8.4 Magnetic 35
1.8.5 Torquer Design Criteria 35
Chapter 2 LPMT Design 36
2.1 LPMT Stabilization Concept 36
2.2 Stabilization System Modification 36
2.2.1 Detumble Mode vs. Alignment Mode 37
2.2.2 Large Torquers vs. Vernier Torquers 37
2.3 Satellite Assumptions 37
2.3.1 Mass Moment of Inertia 37
2.3.2 ARC Power System 38
2.4 Large Core Sizing 39
2.4.1 Core Miniaturization 39
2.4.2 Solenoid Sizing 40
2.4.3 Capacitor Sizing 42
2.4.4 Charging Circuit Properties 43
2.4.5 Circuit Testing Results 44
2.4.6 Saturated Core Analysis 46

Page

2.4.7 Saturated Core Results 47
2.4.8 ALNICO5 Cores 48
2.5 Vernier Core Sizing 48
2.5.1 Design Requirement 48
2.5.2 Fabrication 49
2.5.3 Thickness Adjustment 50
2.5.4 Torque Amount 50
Chapter 3 Rate & Position Determination Systems 51
3.1 Purpose of System 51
3.2 Magnetometer Placement 51
3.3 Cycle Time 52
3.4 Basic Body Rate Determination 52
3.5 Magnetometer Noise 53
3.6 MATLAB Line Smoothing 54
3.7 Position Determination Logic 55
3.7.1 Automatic Position Determination 55
3.7.2 Manual Position Determination 56
3.7.3 Adjustment Options 56
Chapter 4 ACDS Simulation Software 57
4.1 EXIST Origin 57
4.2 ACDS Simulation Code 57
4.2.1 ODE45 57
4.2.2 Gravity Gradient Torques 58

Page

4.2.3 IGRF-10 Field Model 58
4.2.4 Detumble & Alignment Gains 59
4.2.5 Alternate Stable Alignments 60
4.2.6 Alignment Modes 61
4.2.7 Alignment Offsets Defined 63
4.2.8 Alignment Offset Results 64
4.3 Detumble Results 64
4.3.1 ALNICO1 Cores 65
4.3.2 ALNICO5 Cores 67
4.3.3 Permalloy80 Cores 69
4.3.4 Other Inclination Angles 69
4.3.5 Variants in Core Sizing 72
4.3.6 Unbalanced Cores 72
4.3.7 Faster Charging Time 73
4.3.8 Higher Tip-off Body Rates 74
4.4 Alignment Results 74
4.4.1 Polar Orbit 74
4.4.2 Minimum Inclination Angle 75
4.4.3 Retrograde Correction 76
4.4.4 Prograde Reversed Condition 77
4.4.5 Variants in Vernier Core Sizing 78
4.4.6 Faster Charging Times 79
4.4.7 Unbalanced Cores 80

Page

4.5 ACDS – Ground Station Interaction 81
4.5.1 Force Mode 1 (Upgrade) 81
4.5.2 Force Mode 1 (No Upgrade) 81
4.5.3 Force Mode 1 Vernier (No Upgrade) 82
4.5.4 Force Mode 2 (Upgrade) 82
4.5.5 Force Mode 2 (No Upgrade) 82
4.5.6 Force Mode 3 82
4.5.7 Force Mode 3 Large Torquers 82
4.5.8 ACDS Off 83
4.5.9 Permanent Magnet Hysteresis Mode 83
4.5.10 Individual Commands 83
4.5.11 Charging Cycle Time 83
4.5.12 Position Determination Changes 84
4.5.13 Data Collection 84
Chapter 5 Future Work 85
5.1 Testing 85
5.2 LPMT Testing 85
5.2.1 Vernier Core Testing 85
5.2.2 ACDS Board Testing 85
5.2.3 Magnetic Balancing 86
5.2.4 ACDS Software Testing 86
5.3 Magnetometer Testing 86
5.3.1 Rate Determination System Testing 87

Page

5.3.2 Position Determination System 87
5.4 Next Prototype 87
Chapter 6 Conclusions 89
References 90
Literature Cited 90
Other References 92

List of Figures

Page

Figure 1: Alaska Research CubeSat 2
Figure 2: Earth-Centered Inertial Frame 4
Figure 3: Earth-Centered Earth-Fixed Frame 5
Figure 4: Local North East Down Frame 6
Figure 5: Orbital Frame 7
Figure 6: Spacecraft Body-Fixed Frame 8
Figure 7: Frame Transformation Elements 9
Figure 8: Permanent Magnet in Polar Orbit 13
Figure 9: Gravity Gradient Torque 19
Figure 10: LPMT Schematic 21
Figure 11: Idealized Core Material Hysteresis Loop 22
Figure 12: Very Basic Stability System 23
Figure 13: Commanded Spacecraft Body Rates 24
Figure 14: Magnetic Torque Dipole Command Quantizer 25
Figure 15: ACDS Control Loop 25
Figure 16: LPMT Bias Alignment 27
Figure 17: Magnetic Pole Locations [http://wdc.kugi.kyoto-u.ac.jp] 29
Figure 18: Global Magnetic Flux Densities [0° / 180° Longitude] 30
Figure 19: Northerly Component of Earth's Magnetic Field 31
Figure 20: South Atlantic Anomaly [NASA - ROSAT] 32
Figure 21: Wrapped Solenoid Bobbin 41
Figure 22: Capacitor Size Comparison (7.0 Input Voltage) 42
Figure 23: Solenoid H-Field 43
Figure 24: Charging Circuit Voltage / Current 44
Figure 25: Calculated Spacecraft Body Rates (Degrees / 10 Seconds) 54
Figure 26: Retrograde Orbital Alignment 60
Figure 27: Prograde Reversed Condition 61

Page

Figure 28: Mode 2 Bias Alignment 62
Figure 29: Biased Alignment Offset 63
Figure 30: Tip-off Rate Performance 65
Figure 31: ALNICO1 Body Rate Error (600 km / 90° inclination) 65
Figure 32: ALNICO1 Torquer Switches (600 km / 90° inclination) 66
Figure 33: ALNICO1 Dipole Moments (600 km / 90° inclination) 67
Figure 34: ALNICO5 Body Rate Error (600 km / 90° inclination) 68
Figure 35: ALNICO5 Torquer Switches (600 km / 90° inclination) 68
Figure 36: ALNICO1 Body Rate Error (600 km / 64° inclination) 69
Figure 37: ALNICO1 Torquer Switches (600 km / 64° inclination) 70
Figure 38: ALNICO1 Body Rate Error (600 km / 45° inclination) 70
Figure 39: Torquer Switches (600 km / 45° inclination) 71
Figure 40: ALNICO1 Body Rate Error (600 km / 0° inclination) 71
Figure 41: ALNICO1 Unbalanced Torquers Body Rate Error (600 km / 0° inclination) .73
Figure 42: ALNICO1 Body Rate Error (600 km / 90° inclination/1 sec charge time) 74
Figure 43: ALNICO1 Alignment (600 km / 90° inclination) 75
Figure 44: ALNICO1 Pointing Accuracy (600 km / 90° inclination) 75
Figure 45: ALNICO1 Pointing Accuracy (600 km / 64° inclination) 76
Figure 46: Retrograde Correction (600 km / 90° inclination) 77
Figure 47: Prograde Reversed Correction (600 km / 90° inclination) 78
Figure 48: Attitude Error for 0.00022 A-m^2 Vernier Torquer 78
Figure 49: Attitude Error for 0.00055 A-m^2 Vernier Torquer (10 Sec Cycle) 79
Figure 50: Attitude Error for 0.00055 A-m^2 Vernier Torquer (1 Sec Cycle) 79
Figure 51: Attitude Error for 0.04% Large Torquer Imbalance 80
Figure 52: Attitude Error for 0.05% Large Torquer Imbalance 81

List of Tables

Page

Table 1: Large LPMT Dipole Results 47
Table 2: IGRF-10 Main Field Coefficients for 2010 132
Table 3: IGRF-10 Secular-Variation Model for 2010 to 2015 133

List of Other Materials

CDROM – CubeSat Attitude Control Utilizing LPMTs & a Magnetometer

List of Appendices

Page

Appendix A [ACDS Simulation Software] 99

Appendix B [ODE45 Subroutine] 123

Appendix C [IGRF-10 Model] 126

Appendix D [IGRF-10 Coefficients] 132

Appendix E [LPMT Design Code] 134

Appendix F [Rate Calculation with Magnetometer Noise] 140

Chapter 1 Introduction

1.1 Purpose of the Research

The University of Alaska Fairbanks (UAF) is America's Arctic University and conducts extensive research directed at the northern regions of our planet. As a member of the National Space Grant College and Fellowship Program, UAF is in the process of designing and building their first satellite. The prototype is called the Alaska Research CubeSat (ARC) and is being constructed to the CubeSat Design Specification (CDS) [1].

The CDS was created in 1999 by California Polytechnic State (Cal Poly) and Standford Universities to provide a design standard for research and educational picosatellites. The overall goal was to reduce the development time and launch cost of university satellite projects. Completed CubeSats are carried to orbit within the Poly-Picosatellite Orbital Deployer (P-POD) which serves as the interface between them and the launch vehicle. This concept further reduces cost by eliminating the need for customized integration efforts. The CDS, current at the time of this paper, was Version 12 and is included on the attached CDROM.

The goal of ARC is to demonstrate UAF's ability to design and construct a working CubeSat. The prototype will carry a small camera which will be used to examine the polar region and verify CubeSat attitude. Future versions will carry a mass spectrometer to be used for atmospheric research. In either case, the satellite will be required to arrest any tip-off rates imparted by the P-POD and subsequently achieve a desired attitude. This work documents the design of the Attitude Control and rate Determination Systems (ACDS) that ARC will use to achieve the proper alignment.

ARC's ACDS is unique in the fact that it will be using Low-Power Magnetic Torquers (LPMTs) for attitude control. These torquers were conceived in 1971 by Dr. Michael Polites [2], but have yet to be incorporated into flight hardware. The power / size limitations of CubeSats and UAF's extreme latitude of 64° north (necessitating a high inclination orbit in order to make radio contact with the on-campus ground control station) make this system exceptionally well suited for ARC's mission requirements.

1.2 CubeSat Requirements

ARC will be a 1U CubeSat as detailed in the CDS Version 12. It is essentially a cube with 10 centimeter sides and a mass of no more than 1.33 kilograms. Four of ARC's edges will be 11.35 cm rails used to slide out of the P-POD. The camera was placed such that the gravity gradient torques, although extremely small, would tend to align the camera with the local vertical. The rest of the surface area is covered with solar panels to the maximum extent possible as shown in Figure 1.

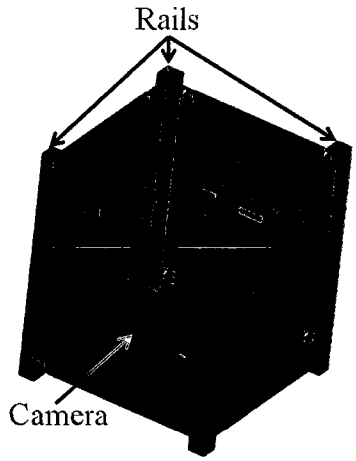


Figure 1: Alaska Research CubeSat

It is desired to place ARC in a circular orbit at an altitude of 600 km as this is the target trajectory of a future 3U (defined in the CDS) mass spectrometer version of the satellite. The 3U version is also planned to have the rails aligned with the local vertical. A validation of the ACDS on ARC, in this orbit, will result in the system being incorporated into the future mission. UAF's satellite will not contain a deorbit device since it is not required to comply with the CDS requirement of satellite reentry within 25 years of launch at the intended altitude.

ARC is planned for an inclination angle window of 64-116 degrees. UAF's Electrical and Computer Engineering Department is designing and constructing a ground control station for the satellite that will be located on the main campus in Fairbanks. Due to the campus latitude of 64° north, high inclination angles are required to provide as many contact opportunities as possible and eliminate the need for an off-campus control

station. An added benefit is that it will also increase the amount of the polar region the satellite is able to observe. The system described herein demonstrates its best performance in a polar orbit with acceptable results over the entire window.

A typical CubeSat experiences a tip-off rate of 1-3° per second [3]. This is a spin imparted on the satellite by its ejection from the P-POD. The ACDS is designed to arrest a tip-off rate of up to 5° per second (over twice the expected rate); this mode of operation is known as the detumble phase. In the event of higher than expected tip-off rates, the ACDS has a contingency mode discussed in Subsection 4.5.9 below that will be used to bring the rates passively under control. The control system will then align the satellite so that the camera points in the nadir direction and a predetermined face points in the ram (velocity vector) direction with a pointing accuracy of ±5 degrees for both; this mode of operation is known as the alignment phase. This will meet future mission requirements as well when the mass spectrometer is incorporated into a 3U CubeSat looking in the ram direction.

It is assumed that the only attitude or rate information available to ARC will come from an onboard three-axis magnetometer. The time rate of change of the flux field will be used to determine the spacecraft's angular velocity vector ($\boldsymbol{\omega}$) which is otherwise known as the spacecraft body rates. The magnitude of the measured magnetic field can be used to determine the spacecraft's relative position in relation to the equator and North / South Poles.

1.3 Reference Frames

The following section describes the various reference frames used by ARC's onboard flight control software as well as those used only in the ACDS simulation software contained in the appendix. This information is necessary to understand the attitude determination systems used on other CubeSat projects and the various methods by which ARC achieves the proper alignment while using only a rate determination system.

1.3.1 Earth-Centered Inertial

The Earth-Centered Inertial (ECI) frame's origin is collocated with the Earth's center of mass. The x-axis is parallel to a line drawn from the Sun to the Earth on the vernal equinox. This line once pointed to the constellation Aries and is sometimes known as the First Point of Aries, although due to precession, it now point towards Pisces. The z-axis is aligned with the Earth's orbital angular momentum vector and points in a northerly direction as shown in Figure 2. The y-axis of this frame is in an arbitrary direction defined by the right-hand rule for an orthogonal triad.

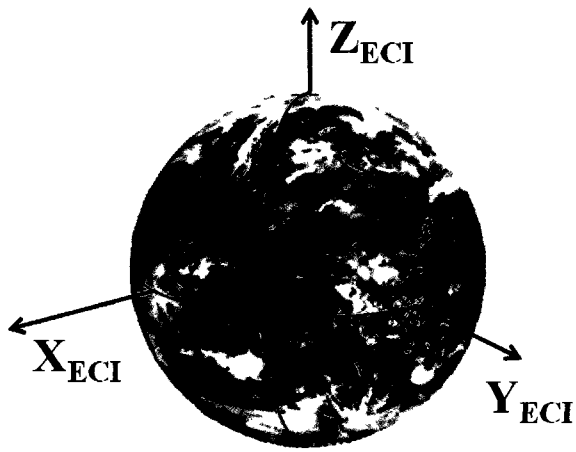


Figure 2: Earth-Centered Inertial Frame

This frame is considered fixed in inertial space (non-accelerating). In reality, this is untrue, but since the impacts of third body interactions from the Moon for example are negligible to an Earth orbiting CubeSat, the premise is valid for this purpose. As such, it is assumed that Newton's First Law is satisfied so that any free motion will have a constant magnitude and direction in this frame. The computer simulation of ARC's ACDS uses ECI to calculate disturbance torques and vehicle motion as it is the easiest frame in which to do so. However, ARC's onboard flight software will not be required to track or have any knowledge of this frame.

1.3.2 Earth-Centered Earth-Fixed

The Earth-Centered Earth-Fixed (ECEF) frame is essentially what most people would recognize as latitude and longitude. The x-axis points from the Earth's center through the equator at the prime meridian as shown in Figure 3. The z-axis is aligned with the true North Pole and intersects the Earth's surface at 90° north latitude. The y-axis is again in an arbitrary direction needed to complete an orthogonal right-hand rule triad; however in this case it passes through 90° east longitude and the equator.

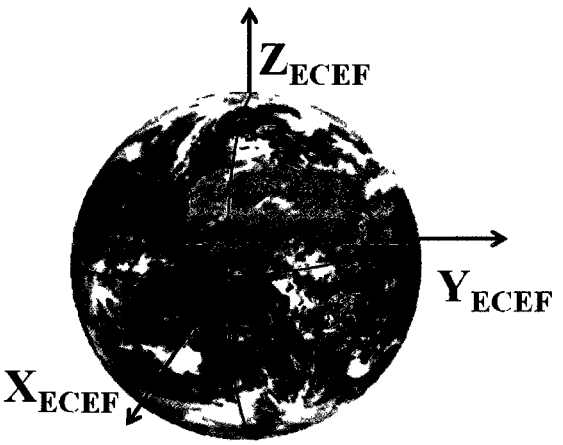


Figure 3: Earth-Centered Earth-Fixed Frame

Unlike ECI, this frame is fixed to the Earth and rotates with it through space. The ACDS simulation software tracks this frame in order to feed the latitude / longitude information into a model of the geomagnetic field. ARC's onboard flight software will not be required to track this frame of reference although it will track relative position to the equator and North / South Poles.

The ECEF frame assumes a perfectly spherical Earth and is in this respect different from the latitude and longitude one would read off of a Global Positioning System (GPS) receiver. GPS generally uses the World Geodesic System 1984 datum which assumes the Earth's shape to be that of an oblate spheroid. ARC's assumed circular orbit makes a datum correction unnecessary.

1.3.3 Local North East Down

The Local North East Down (LNED) frame is close to what a person standing on the ground would read off of a compass. The x-axis points from the position of interest in the direction of the true North Pole (a compass needle points to the magnetic North Pole and would need to be corrected for the local magnetic variation). The z-axis points to the Earth's center of mass which is also the origin of the ECI and ECEF frames. The y-axis is again in an arbitrary direction needed to complete a right-hand rule triad; however in this case it generally points east as shown in Figure 4.

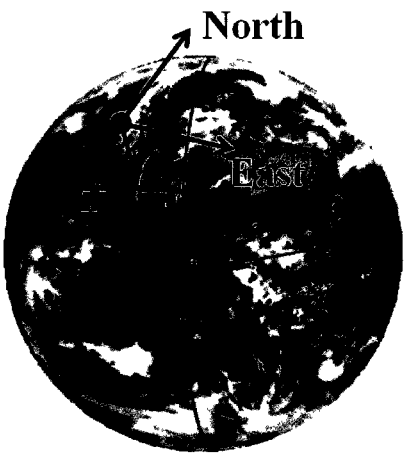


Figure 4: Local North East Down Frame

A computerized model of the Earth's geomagnetic field used in the ACDS simulation software provides the geomagnetic flux density vector at any given latitude / longitude in this frame. The ACDS simulation software tracks LNED so that the flux density vector can be translated into a spacecraft body centric frame defined in Subsection 1.3.5 and known as the body frame. ARC's onboard flight software is not required to track or have any knowledge of the LNED frame.

1.3.4 Orbital

When ARC is in the proper alignment, the body frame (described in the next subsection) and the orbital frame are aligned with each other. The x-axis, of the orbital frame, points from a location of interest along a line parallel to the orbit radius vector.

For the purpose of this paper the position of interest will always be the geometric center of the spacecraft. The y-axis is aligned with the ram direction, which is also known as the orbital velocity vector. The z-axis is aligned with the orbit normal assuming a right-hand rule orthogonal triad. The system is shown in Figure 5.

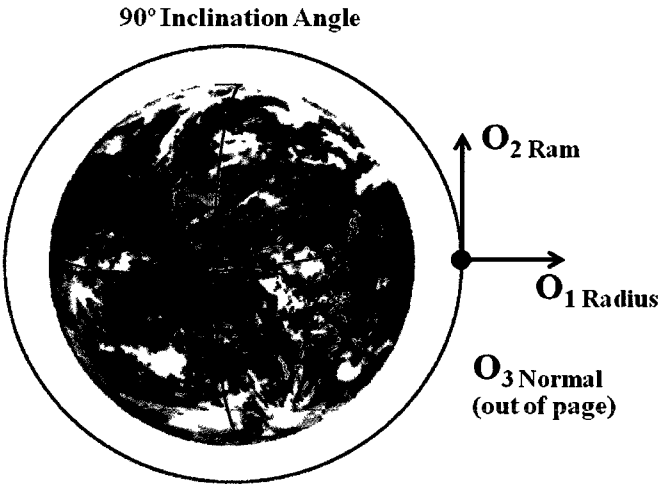


Figure 5: Orbital Frame

ARC's onboard flight control software does not track this frame. However, this frame is very useful in the ACDS simulation software as it makes it easy to determine what degree of pointing accuracy the spacecraft is achieving. This is done by measuring the difference between the orbital and body frames. In the ACDS simulation software, these results are presented in the second output plot produced when the code in Appendix A is run. There are numerous examples presented in Chapter 4.

1.3.5 Spacecraft Body-Fixed

The Spacecraft Body-Fixed (body) frame has its origin at the geometric center of the CubeSat. The x-axis points from the frame's origin in the opposite direction of the camera and is perpendicular to the top face of the CubeSat. The y-axis points from the origin through the face of the CubeSat that will be aligned with the ram direction. It is perpendicular to the ram face and as such points in the desired ram direction. The z-axis is an arbitrary direction to complete a right-hand rule triad; it is of course perpendicular

to the face of the CubeSat that it penetrates. This axis system was chosen because, when the spacecraft is in the proper alignment, the orbital and body axes are aligned. This body axes system is different from the one specified in the CDS and is shown in Figure 6.

In the alignment phase, the z-axis is the only axis upon which the satellite will rotate. One revolution around the z-axis per orbit will allow the camera to be continuously pointed towards the Earth; therefore allowing a photo to be taken anytime the CubeSat passes over a point of interest.

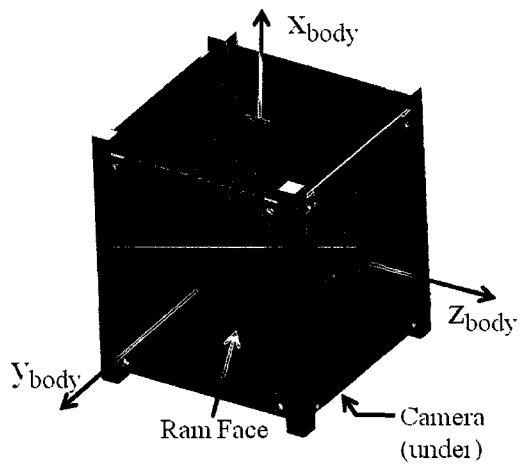


Figure 6: Spacecraft Body-Fixed Frame

ARC's onboard magnetometer will measure magnetic flux density in this frame. Those measurements will be used to calculate the spacecraft's body rates. Knowledge of those rates coupled with the bias plan discussed in Subsection 1.7.1 will ensure ARC is in the proper alignment. The ACDS simulation software also uses this frame to determine when it is appropriate to change the orientation of a torquer's magnetic dipole moment.

1.3.6 Spacecraft Body Rates

An understanding of spacecraft body rates is critical to comprehending the operation of the ACDS. The body rates are an angular velocity measurement in the body frame expressed in radians per second. The vector has three components which equate to rotation about the body x, y, and z-axes. The positive direction is defined by the right-hand rule.

1.4 Reference Frame Transformations

The following section shows how the ACDS simulation software performs transformations from one reference frame to another. In order for this to be accomplished, the simulation must keep track of several important spatial relations defined in the next subsection.

1.4.1 Definition of Variables

Orbital inclination (i) is the angle between the Earth's equatorial plane and the spacecraft's orbital plane. Longitude of the ascending node (Ω) is the angle between the ECI x-axis and the point (known at the ascending node) where the spacecraft's orbital plane crosses the ECI x / y-plane with the spacecraft travelling in a northerly direction. Argument of latitude (ν) is an angular measure from the ascending node along the orbital path to the current spacecraft position. This element is unique to a circular inclined orbit where the argument of periapsis and true anomaly are undefined. Magnetic reference angle (γ) is an angular measurement from the ECI x-axis to the prime meridian. The rate of change of this angle is equal to the Earth's rate of rotation. Position vector ($\boldsymbol{R}$) is a distance measurement from the Earth's center of mass to the current spacecraft position (measure of altitude). These parameters are easier to understand by referencing Figure 7.

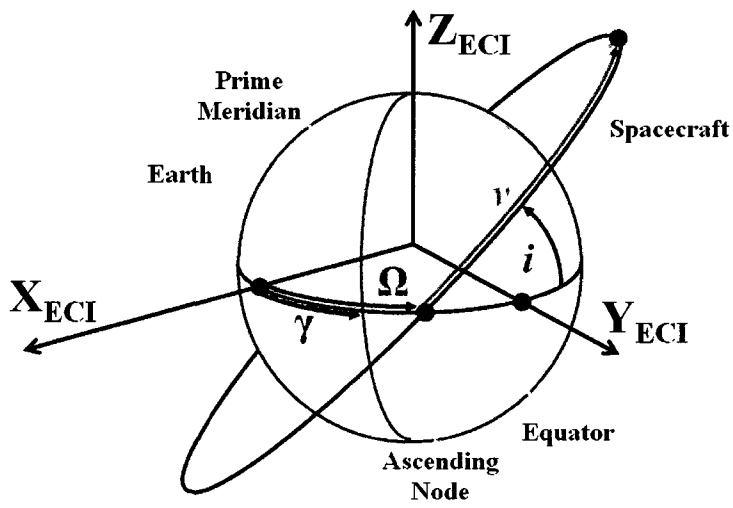


Figure 7: Frame Transformation Elements

1.4.2 ECI to ECEF

The ECI to ECEF transformation deals exclusively with Earth's daily rotation about its own axis and the annual orbit of the planet around the Sun. The impact of Earth's movement along its orbital path is negligible to ARC. The relative difference between the ECI and ECEF frames due to the Earth's orbit around the Sun is less than 1° per day, and thus makes tracking this parameter a waste of computational resources. The simplified transformation matrix can be written as:

$$\vec{x}_{ECEF} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{x}_{ECI}. \quad (1)$$

An onboard attitude determination system would most likely need to have knowledge of this transformation. This would be especially true if utilizing a star and / or sun tracker since the satellite would need to know where the objects were in inertial space. ARC will not require knowledge of this frame due to the unique nature of the LPMT bias plan discussed in Subsection 1.7.1.

1.4.3 ECEF to LNED

The ECEF to LNED rotation matrix is dependent on latitude (φ) and longitude (λ) which relates these two frames,

$$\vec{x}_{LNED} = \begin{bmatrix} -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ -\sin(\lambda) & \cos(\varphi) & 0 \\ -\cos(\varphi)\cos(\lambda) & -\cos(\varphi)\sin(\lambda) & -\sin(\varphi) \end{bmatrix} \vec{x}_{ECEF}. \quad (2)$$

An onboard attitude determination system that uses the Earth's geomagnetic flux density vector as a reference would need knowledge of this transformation. ARC's bias plan utilizes only the magnitude of the Earth's geomagnetic field which can be measured with the onboard magnetometer. The orientation of the field is not required for ARC to accomplish the alignment. However, a satellite with an onboard attitude determination system that utilized magnetic flux density as one of its reference vectors would need knowledge of this transformation.

1.4.4 ECI to Orbital

The ECI to orbital transformation depends upon where the spacecraft is within its orbit. In the next equation, cosine and sine are abbreviated c( ) and s( ), respectively, as

$$\vec{x}_{Orbit} = \begin{bmatrix} c(\nu)c(\Omega) - s(\nu)c(i)s(\Omega) & c(\nu)s(\Omega) + s(\nu)c(i)c(\Omega) & s(\nu)s(i) \\ -s(\nu)c(\Omega) - c(\nu)c(i)s(\Omega) & -s(\nu)s(\Omega) + c(\nu)c(i)c(\Omega) & c(\nu)s(i) \\ s(i)s(\Omega) & -s(i)c(\Omega) & c(i) \end{bmatrix} \vec{x}_{ECI}. \quad (3)$$

1.4.5 Quaternions

The ACDS simulation software uses quaternions for a majority of the attitude tracking calculations. Quaternions use algebraic relations to determine the elements of rotation matrices instead of trigonometric functions. This increases the speed of the calculations and prevents the possibility of singularities that sometimes occur when using angles and trigonometric functions.

The quaternion itself is simply a composite of a scalar and a vector, for a total of four elements. The vector defines an axis of rotation; while the scalar specifies an amount of rotation. The process is based on one of Euler's theorems which states that any rotation of a body with respect to another may be described by a single rotation through some angle about a fixed axis. A quaternion ($\boldsymbol{Q}$) is defined as:

$$\vec{Q} = i\,q_1 + j\,q_2 + k\,q_3 + q_4. \quad (4)$$

The first three parameters represent the vector while the fourth defines the rotation angle.

1.4.6 ECI to body

The ACDS simulation software initializes with a known difference between the ECI and body frames. This setting can be adjusted by the user to change the initial orientation of the spacecraft and thus simulate a variety of different initialization conditions. The difference between frames is converted into a quaternion that is tracked through each time step by a Runge-Kutta differential equation solver. When necessary, the quaternion can be converted into a transformation matrix using the following relationship:

$$\begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_1q_4 + q_2q_3) \\ 2(q_1q_3 + q_2q_4) & 2(-q_1q_4 + q_2q_3) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}. \quad (5)$$

Throughout the computation process, it is helpful to occasionally force the following relationship which is a requirement of quaternions:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \quad (6)$$

This helps prevent round off and truncation errors from growing during lengthy ACDS software simulations.

1.5 CubeSat Attitude Control Literature Review

As research tools and technology demonstrators, CubeSats have been constructed with a wide variety of attitude control systems. The following subsections discuss methods utilized on other projects as well as their effectiveness when that data is available. Unfortunately, many CubeSats have been destroyed during launch or have failed to initiate contact after separation from their P-POD. These missions are annotated with (*f*) and will obviously have no attitude control effectiveness information.

1.5.1 No Attitude Control

Few CubeSats have been planned with no form of attitude control onboard. Most of these satellites are from first time builders and serve as test beds for power and communication systems, as well as, various sensors such as sun trackers and GPS. Successful missions include CUTE-I from the Tokyo Institute of Technology (Japan), UWE-1 & 2 from University of Würzburg (Germany), and SEEDS-2 from Nihon University (Japan). Amateur radio satellites with omnidirectional antennas have also had success without using an attitude control system.

1.5.2 Permanent Magnets

Permanent magnets are the most popular passive attitude control means for vehicles that do not have specific pointing requirements. In this method, a permanent

magnet of sufficient size is placed in the satellite so that it will align itself with the Earth's magnetic field. When the magnet is out of alignment, a torque ($\boldsymbol{\tau}$) is passively produced, which is governed by the following equation:

$$\vec{\tau} = \vec{m} \times \vec{B}, \tag{7}$$

where $\boldsymbol{m}$ is the magnet's dipole moment and $\boldsymbol{B}$ is the Earth's magnetic field flux density vector. The attitude produced by this torque depends on the inclination of the satellite's orbit.

In an equatorial orbit, the permanent magnet will align with the Earth's magnetic field in a north / south direction. However, the vehicle will be free to rotate around the long axis of the magnet in an east / west direction. In a polar orbit, the magnet will again align with the Earth's field. However, the vehicle will be expected to rotate about an axis perpendicular to the magnet twice per orbit. This is due to the curvature of the flux lines at the north / south poles which is shown in Figure 8 below.

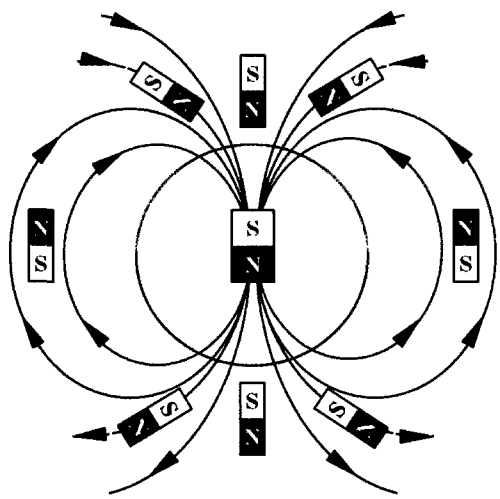


Figure 8: Permanent Magnet in Polar Orbit

In this case, the satellite will normally not rotate about the magnet's long axis due to the stabilizing effect of the rotation caused by the magnet crossing over the poles, provided the moments of inertia do not make this an unstable condition.

This system is not traditionally used by itself. Unlike a compass needle that centers on north and then stops, a permanent magnet in space tends to oscillate about the magnetic flux density vector. The compass has a liquid surrounding the needle that

provides damping, while the spacecraft does not. Usually spacecraft designers will add hysteresis rods, discussed in Subsection 1.5.4, to damp the oscillations. The missions using this combination approach are discussed in that subsection.

1.5.3 Open Air / Vacuum Coils

Open coils are a simple electromagnet attitude control solution. The torque provided is still governed by Equation (7). However, unlike permanent magnets where the magnetic dipole moment is fixed, here it can be controlled as shown below:

$$\vec{m} = iNA. \tag{8}$$

The average area (***A***) enclosed by each loop as well as the number of turns in the torquer coil (***N***) are set during satellite fabrication. The current (***i***) can be controlled in flight and allows the dipole moment to take on a range of values depending on the construction of the coil and the available power. There are three major limitations to using open coils for attitude control.

Power and weight are the first two and they are directly tied together. In order to maximize the amount of torque available, the area of the coil and the number of turns of wire should both be maximized. An increase in wire will result in an increase of weight and resistance but will lower the current required to achieve the same magnetic dipole moment. The power (***P***) required is dependent on the resistance (***R***) and current supplied to the coils as seen in the following equation:

$$P = i^2R. \tag{9}$$

The Norwegian NCUBE-1 (*f*) and NCUBE-2 (*f*) designs used 140 turn coils that utilized 25 mA. While CalPoly's CP-1 (*f*), CP-2 (*f*), CP-3(*f*), and CP-4 (*f*) used 54 turn coils that required 300 mA.

The third limitation of the system is generally the requirement for an attitude determination system. The Earth's magnetic field (***B***) in Equation (7) cannot be controlled. Although the detumble can be completed without attitude knowledge as discussed in Subsection 1.5.10, an alignment phase usually requires attitude knowledge. The CubeSat must apply current to the torque coils only at those times when the resulting

torque is in the desired axis / axes needed to achieve attitude alignment. This necessitates that the satellite and / or the ground control station knows the vehicle's attitude, location, and the orientation of the magnetic flux field; the appropriate guidance control commands must also make it to the torque coils. Subsection 1.5.9 discusses several attitude determination systems utilized on other projects.

The majority of missions utilizing this control mechanism have been destroyed in the launch phase or have failed to make contact with the ground after deployment from the P-POD. Compass-One by Fachhochschule Aachen (Germany) successfully arrested tip-off rates (detumble), but was unable to align itself due to a failure in the attitude determination system. Most open coil missions utilize a $\dot{\boldsymbol{B}}$ controller, which does not require attitude information (discussed in Subsection 1.5.10), to complete their detumble. Successful missions include CUTE 1.7+APD & CUTE 1.7+APD-2 by Tokyo Institute of Technology (Japan), Libertad-1 by the University of Sergio Arboleda (Columbia) and a commercial mission CSTB-1 by Boeing (USA). The university missions have very little on orbit performance data published in English other than to say they were successful. The Boeing mission claims to have achieved 100% of its mission objectives, but no other information is available due to proprietary concerns.

1.5.4 Torque Rods / Hysteresis Rods

Torque rods operate on the same basic principle as open coils; however their large area is reduced to a solenoid wrapped around a magnetically soft metallic core with high permeability. The magnitude of the magnetic dipole moment is given by the following equation:

$$\vec{m} = \frac{\pi r^2 N i}{\frac{1}{\mu_r} + N_d} \tag{10}$$

where $\boldsymbol{r}$ is the radius of the core material, $\boldsymbol{\mu_r}$ is the relative permeability of the core and $\boldsymbol{N_d}$ is the demagnetizing factor ($\boldsymbol{l}$ is the length of the core):

$$N_d = \frac{4\ln\left(\frac{l}{r}\right) - 1}{\left(\frac{l}{r}\right)^2 - 4\ln\left(\frac{l}{r}\right)}. \quad (11)$$

The introduction of the relative permeability of the core drastically reduces the amount of current required to achieve the same dipole moment as an open coil [4].

The major limitation of this method is the inherent hysteresis of the core material. A soft magnetic material is used so that when the current is removed, the core has as little residual magnetization as possible; however, it is quite often not completely zero. This residual continues to create a torque on the spacecraft in the same manner as a permanent magnet and can have a detrimental impact to the desired alignment.

A hysteresis rod on the other hand uses a ferromagnetic core for energy dissipation. As the solenoid rotates through the Earth's magnetic flux lines, a current is generated. If this current is dissipated through a resistor, it places a torque on the hysteresis rod; the resulting force tends to slow the spacecraft's rotation. This method is usually applied in concert with a permanent magnet. The permanent magnet aligns with the Earth's magnetic field while the hysteresis rods provide nutation or rate damping. This combination method has been used on XI-IV and XI-V by University of Tokyo (Japan), GeneSat-1 by Center for Robotic Exploration and Space Technologies (USA), CAPE-1 by University of Louisiana (USA), Delfi-C3 by Alborg University (Denmark), and several others.

1.5.5 Reaction Wheels

This attitude control system relies on conservation of the spacecraft's total angular momentum vector; which is constant in a closed system (no external torque), such as a satellite in orbit. The basis of this control is found in the definition of angular momentum (***H***); known to be the product of the spacecraft's mass moment of inertia (***I***) and angular velocity ($\boldsymbol{\omega}$) vector:

$$\vec{H} = I\vec{\omega}. \quad (12)$$

A reaction wheel is a small rotating mass placed inside of the vehicle. The wheel's angular momentum is a component of the spacecraft's total angular momentum vector:

$$\vec{H}_{Total} = \vec{H}_{Spacecraft} + \vec{H}_{Reaction\ Wheel}. \tag{13}$$

Since the total angular momentum is constant (with no external torque), a change in the reaction wheel's angular momentum must be compensated for with a change in the spacecraft's angular momentum. In Equation (12), the vehicle's mass moment of inertia is also constant. Therefore a change in the spacecraft angular momentum vector can only be precipitated by a change in the spacecraft's angular velocity vector. In other words, a change in the spacecraft's angular momentum results in a torque ($\boldsymbol{\tau}$) on the vehicle:

$$\vec{\tau}_{Spacecraft} = \frac{d\vec{H}_{Spacecraft}}{dt}. \tag{14}$$

By using an orthogonal triad of reaction wheels, it is possible to provide precise control to a spacecraft in all three axes.

Reaction wheels are the primary attitude control means for CubeSats that have very accurate pointing requirements. Unlike magnetic torquers, momentum control devices do not depend on the Earth's magnetic field. Thus they are able to provide precise control at any point in their orbit; however, there are two main disadvantages to using this system.

First, a reaction wheel is a mechanical device and is subject to any number of failures that can render the device inoperative. The most common failures are from the bearings wearing out or electrical control problems associated with the motor.

The second disadvantage of reaction wheels is the combination of the power required to keep the mass spinning and the size / weight of the system. Most Cubesat's use a zero-bias system meaning that the reaction wheels have little or no initial momentum. Power is expending spinning up the wheels to apply a torque. Wheels are powered down once the torque is no longer required. Over time, the system compensates for various disturbance torques. If the disturbances average from a predominate direction, the speed of the wheels is slowly increased to maintain alignment. This in turn

slowly increases the power requirements to maintain the desired attitude. Eventually one of the wheels becomes saturated meaning it has reached its maximum speed and physically cannot be made to go any faster; at which point the attitude can no longer be maintained. To compensate for this, most designers will add magnetic torquers to the vehicle design. These torquers can be used to provide an external torque which is compensated for by slowing down the reaction wheels. Of course this requires power for the magnetic torquers (although it eventually lowers reaction wheel power requirements), space inside the spacecraft for the magnetic torquers, and it adds to the satellite's overall weight.

Successful reaction wheel missions include CanX-2 by the University of Toronto (Canada) and BeeSat by Technical University of Berlin (Germany). AAUsat-2 by Alborg University (Denmark) was equipped with momentum wheels and is noteworthy for its rotation rate. This particular satellite was at one point spinning at 85 RPM, which was unintended and greatly in excess of the expected 1-2° per second. However, it appears the excessive rate was caused by something impacting the satellite or a serious flaw in the attitude control software / hardware.

1.5.6 Control Moment Gyros

A control moment gyroscope is essentially a reaction wheel that spins at a fixed velocity. This creates a large angular momentum vector. By mounting the gyro on motorized gimbals, it is possible for the spacecraft to push or pull itself around this large angular momentum vector.

This method requires a larger amount of power since the gyro must always be spun at a constant rate. It also requires additional space for the motorized gimbal assembly. For these reasons, it appears no CubeSat mission has attempted this control method to date.

1.5.7 Gravity Boom

The gravitational pull of the Earth decreases according to an inverse-square law. The parts of a spacecraft closer to the center of the planet will have a greater attraction than those parts further away. Unless the vehicle is perfectly symmetric, gravity gradient torque will tend to align the long axis with the local vertical.

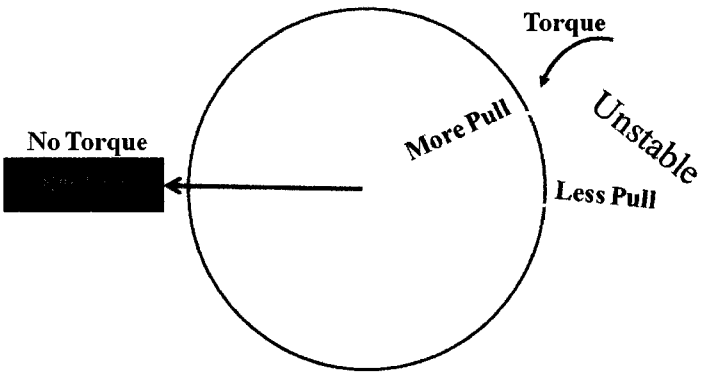


Figure 9: Gravity Gradient Torque

In some spacecraft, gravity gradient torque is exploited by extending a telescoping rod with a mass at the end. This exacerbates the advantage of a long slender spacecraft as seen in Figure 9 above. The design was intended for ICE Cube-1 (*f*) & ICE Cube-2 (*f*) by Cornell University (USA), as well as DTUSat-2 (under development) by Technical University of Denmark (Denmark). The major limitation of the concept is designing the telescoping mechanism, fitting it within the CubeSat, and having it reliably deploy in orbit.

1.5.8 Reaction Control System

A reaction control system utilizes thrusters or propellant expulsion to control rates and attitude. The complexity of such a system and the limited mass for fuel make this system completely impractical for CubeSats. However, ION (*f*) by the University of Illinois (USA) was equipped with vacuum arc micro-thrusters, a low-thrust electric propulsion system utilizing xenon. The intent of this system was to increase the spacecraft's velocity and thus push it into a slightly higher orbit.

1.5.9 Attitude Determination

Onboard attitude determination is in many ways a discipline in its own right. Although there are many ways to accomplish this complex mathematical problem, it is usually solved by referencing at least 2 known observation vectors. The TRIAD algorithm is an example of such a method [5].

In TRIAD, two observation vectors are measured in the body frame ($\hat{W}_1$ & $\hat{W}_2$) and compared to their corresponding representations in known reference frames ($\hat{S}_1$ & $\hat{S}_2$). In the absence of noise, these vectors satisfy the following relationships:

$$\hat{W}_1 = A\,\hat{S}_1 \tag{15}$$

$$\hat{W}_2 = A\,\hat{S}_2. \tag{16}$$

In the above equations, A is a 3 x 3 rotational matrix from which the spacecraft attitude is determined. In reality, noise causes no solution to exist and as such the bulk of attitude determination involves minimization of various cost functions to approximate spacecraft attitude.

The capabilities of a few of the algorithms, such as QUEST, are quite robust [6]. In many cases the spacecraft body rates are calculated from the time rate of attitude change. When the spacecraft is already in the correct attitude, this indirect method of rate determination is more than sufficient.

1.5.10 $\dot{B}$ Controller

In most spacecraft that do contain an onboard attitude determination system, the detumble must be finished prior to an attitude determination being attempted. One of the popular ways to accomplish this is the proportional $\dot{B}$ controller [7]. In this method, the amount of current supplied to the magnetic torquers is based on the time rate of change of the magnetic field. This rate is also used to determine the time the torque coils are energized. The cycle is repeated until the time rate of change of the flux field approaches zero. This allows the detumble phase to be completed without rate gyros or knowledge of attitude.

A less popular alternative method is the bang-bang $\dot{B}$ controller. In this case, the magnetic torquers are always activated at their maximum magnetic dipole moment, instead of using proportional control to calculate a smaller one. An adaptation of the bang-bang system is appropriate for LPMTs since their dipole moment is fixed to discrete values as described in the next subsection.

1.6 Validation of Low-Power Magnetic Torquers

The LPMT was conceived by Dr. Michael Polites in 1971 as a new type of magnetic torquer for satellite attitude control, which uses less power than conventional methods. It was later refined into an attitude stabilization system with the author's assistance during his master's degree work in 2003 [8].

1.6.1 Description of LPMT

The device consists of a solenoid wrapped around a hard magnetic material core that has a high residual intensity of magnetization. By discharging a capacitor of sufficient size through the solenoid, it is possible to reverse the polarity of the core's magnetic dipole. The following diagram is the simplest form of a single torquer unit.

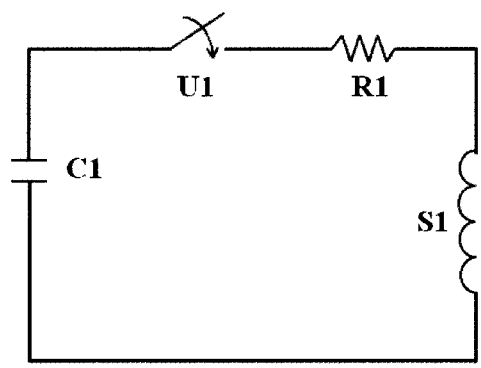


Figure 10: LPMT Schematic

1.6.2 Hard Magnetic Material Core

The core of the torquer has a square to rectangular hysteresis loop as seen in the following idealized figure. If the magnetic field intensity ($\boldsymbol{H}$) of the solenoid is $\boldsymbol{H_s}$ or

greater, it will drive the core material to magnetic saturation (M_s). When the solenoid's magnetic field is removed, the core will maintain a remanent magnetization (M_r). The dipole moment (m) from that remanence will passively provide torque to the vehicle in the same manner as a permanent magnet control system; however, the LPMT can reverse the direction of the dipole moment.

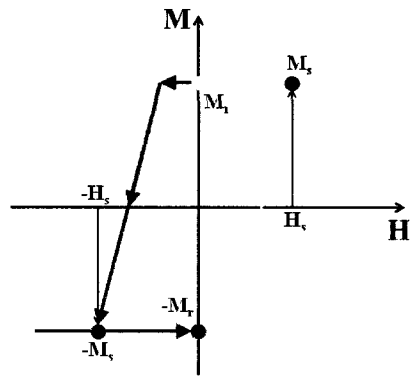


Figure 11: Idealized Core Material Hysteresis Loop

When the solenoid's magnetic field is $-H_s$ or lower, the core will be driven to saturation in the opposite direction. The capacitor size is the same for both conditions, so the polarity of the core depends only on which direction the current pulse travels through the solenoid.

Unlike torque or hysteresis rods, LPMTs use only saturated cores. This eliminates the need to use a coercive force to reduce the magnetization from saturation to zero or to deal with residual hysteresis. The capacitors are sized to drive the core to saturation on each firing, making the system extremely consistent and continuously repeatable.

It is important to remember, from Equation (7), that the torque supplied to the vehicle is completely dependent upon the flux density vector of the Earth's magnetic field. As such it will vary not only depending on where the vehicle is but also with what the vehicle's attitude is in relation to the field.

1.6.3 Basic Stabilization System

The simplest stabilization system using LPMTs would have 2 torquers in each axial plane for a total of 6 for the satellite. In the x / y plane for example, 2 torquers are parallel and on opposite sides of the center of mass. When these torquers have opposite polarity, no torque is supplied to the spacecraft. When one of the torquers is flipped so that they have the same polarity, a torque of twice the dipole moment (of a single core) is supplied to the spacecraft. The following diagram shows the three control options available to a pair of LPMTs.

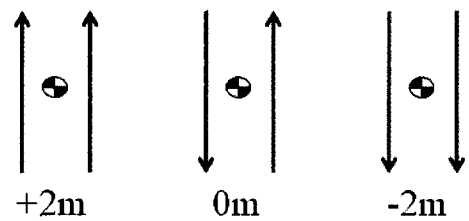


Figure 12: Very Basic Stability System

1.6.4 Stabilization Control Logic

A spacecraft stabilization system utilizing LPMTs is documented in Polites et al. 2004, with this paper's author as a co-contributor [8]. It is an adaptation of the system developed by Polites for emergency stabilization of the Hubble Space Telescope using torque rods in the event of control moment gyro failures [9]. Both systems attempt to set the spacecraft's angular velocity vector (ω) to a commanded ($\omega_{command}$) value. Since ARC will have one face aligned with the ram direction and another aligned with nadir, the commanded body rates are merely a function of the orbital period (time it takes the satellite to complete one orbit), as shown here:

$$\vec{\omega}_{\text{command}} = \begin{bmatrix} 0 \\ 0 \\ \dfrac{2\pi}{\text{orbital period}} \end{bmatrix} \left(\frac{\text{radians}}{\text{second}}\right). \tag{17}$$

This can be visualized in the following diagram where the satellite rotates about its z-axis once each orbit to maintain the desired alignment.

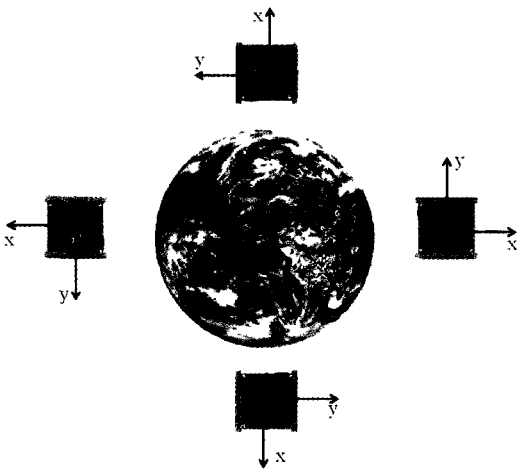


Figure 13: Commanded Spacecraft Body Rates

The actual spacecraft body rates ($\omega_{measured}$) are determined from the onboard 3-axis magnetometer (a method for this is discussed in Subsection 3.4). The measured rate is used to calculate an error:

$$\vec{\omega}_{error} = \vec{\omega}_{command} - \vec{\omega}_{measured}. \tag{18}$$

The magnetic dipole moment command ($m_{command}$) needed to zero the error can be determined from the classic magnetic torquing cross-product law from Polites 1971 [2]:

$$\vec{m}_{command} = k\frac{\vec{H} \times \vec{B}}{\vec{B} \cdot \vec{B}}. \tag{19}$$

By using the definition of angular momentum in Equation (12) and incorporating the mass moment of inertia (I) term into the gain (k), the dipole command becomes a function of the body rate error and the local magnetic flux field (B), which can be read from the onboard magnetometer as,

$$\vec{m}_{command} = k'\frac{\vec{\omega}_{error} \times \vec{B}}{\vec{B} \cdot \vec{B}} \tag{20}$$

where k' is a combination of the control law gain and the mass moment of inertia:

$$k' = kI. \tag{21}$$

The selection of control law gain is discussed in Subsection 4.2.4 below.

Unfortunately, LPMTs are not able to satisfy Equation (20) directly since they are only capable of taking on discrete values. This is the same reason that the traditional $\dot{B}$

controller is not applicable to this task. The desired dipole must therefore be quantized to magnetic dipole moment commands ($\boldsymbol{m}'_{command}$) that can be generated by two LPMTs in each axis (+2$\boldsymbol{m}$, 0$\boldsymbol{m}$, or -2$\boldsymbol{m}$ as seen in Figure 12). The switching logic for one axis is shown in Figure 14 below.

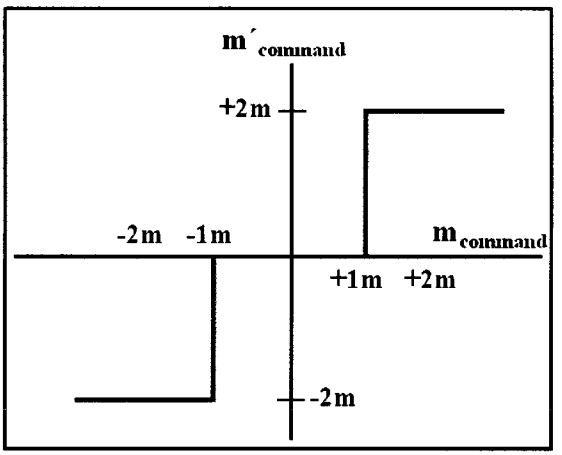


Figure 14: Magnetic Torque Dipole Command Quantizer

Torque commands less than a single core's dipole moment result in the cores being aligned opposite to each other; providing no dipole moment to the vehicle. Commands in excess of a single core's dipole moment will cause the ACDS to flip one of the cores into the same orientation as its twin, thus providing a dipole moment of ±2$\boldsymbol{m}$ to the satellite.

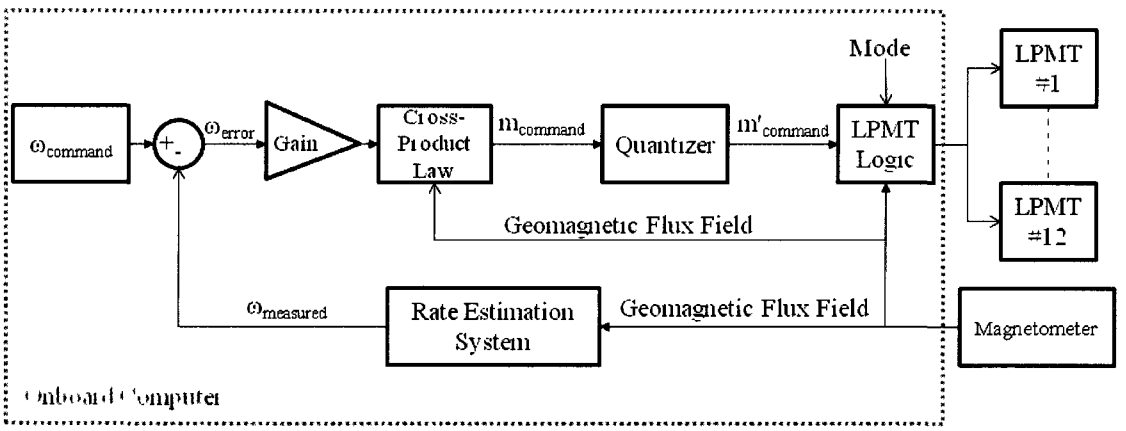


Figure 15: ACDS Control Loop

The completed control loop is seen in Figure 15 above. A magnetometer is used to estimate the spacecraft body rates (method described in Chapter 3) from which the

body rate error can be calculated by using Equation (17). The "gain" and "cross-product law" blocks represent Equation (20) above. The "quantizer" block is shown in Figure 14 and produces the magnetic dipole moment command. The "LPMT logic" block represents software control modes discussed in Subsection 4.2.6. This logic makes the final determination as to whether a LPMT is to have its pole flipped or not.

This design will be used to detumble ARC and aside from extensive miniaturization efforts is relatively unchanged from the original work in 2003. UAF's desire to place their satellite in a polar orbit opened the door to a modification of the system which will allow it to achieve the university's alignment phase goals (camera pointing in the nadir and body y-axis pointing in the ram direction). In order to fully understand the concept it is necessary to review a few important facts about the Earth's magnetic field.

1.7 Geomagnetic Field of the Earth

The geomagnetic field of the Earth is often approximated by a simple dipole located near the center of the planet. The points where the axis of this dipole intersects the Earth's surface are called the North and South Geomagnetic Poles. These poles are currently at about 80° north/south latitude. To make the best fit possible, the dipole should be offset approximately 500 km from the true center of the planet.

1.7.1 Torquer Biased Alignment

Once ARC has completed the detumble mode, it will begin to place itself into the mission attitude. This will be accomplished by biasing the magnetic torquers near the equator and the poles while still attempting to maintain $\omega_{command}$. The concept is simple. Near the equator the Earth's magnetic flux density vector roughly parallels the ram direction. The ram direction torquers will have their polarities matched so that they will attempt to align themselves with the flux density vector. As this occurs, it will generate a body rate error. Utilizing their quantizers, the other two axes (without the bias) will attempt to zero these unwanted angular rates. The bias will be terminated, on all axes, as

ARC leaves the equatorial region. It is the same algorithm, as discussed in Subsection 1.6.4 above, with the exception that the y-axis torquers are forced to be +***2m*** crossing the equator north bound and -***2m*** crossing the equator south bound.

Near the poles, the flux density vector roughly parallels the nadir direction. The nadir direction torquers will have their polarities matched to align themselves with the flux vector. In this case, the x-axis torquers are forced -***2m*** at the North Pole and +***2m*** at the South Pole. The other axes will again attempt to damp any unwanted rates. In the following figure, the camera always points towards Earth and the yellow arcs indicate which axis is being biased (polarity of LPMTs set to match local flux field).

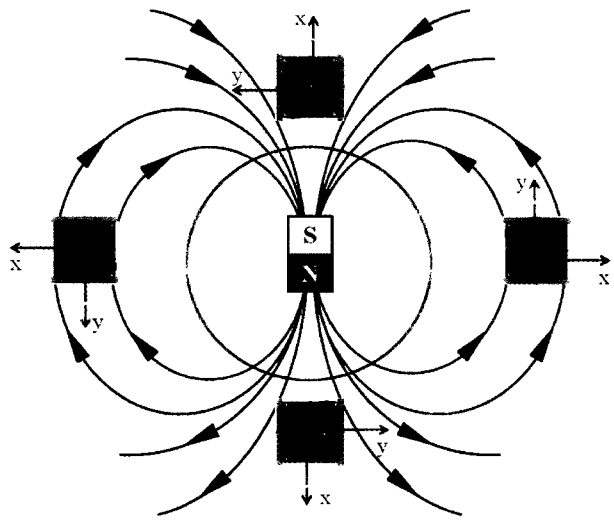


Figure 16: LPMT Bias Alignment

Although biasing a single magnetic axis has been accomplished many times in the past, a review of the literature did not find any cases where the bias was accomplished in two axes. It therefore appears that this is the first development and use of a dual axis magnetic bias system to achieve three axis attitude alignment. This system functions because the goal of ARC is to align the spacecraft body and orbital reference frames in a high inclination orbit.

1.7.2 International Geomagnetic Reference Field

To develop attitude control software, a computerized simulation of the Earth's magnetic field was required. This simulation is based on the International Association of

Geomagnetism and Aeronomy's (IAGA) International Geomagnetic Reference Field (IGRF) [10]. The IGRF was originally developed in 1968 in response to demands for a standardized model of the Earth's main field. Over the years it has been updated to improve accuracy and account for secular variations (changes in the field over time).

Version 10 (IGRF-10) was used in ARC's ACDS simulation. It is a spherical harmonic series that models the Earth's main field at source free regions above the surface, from sources internal to the planet. The model is based on data collected from satellite magnetic mapping missions and worldwide surveys / observations. IGRF-10 will be valid for ARC's proposed orbit and has been used for several CubeSat projects that attempted attitude determination using an onboard version compared with magnetometer readings.

The IGRF-10 flux field is represented as the negative gradient of the scalar potential in the following series expansion:

$$V(R,\theta,\lambda,t) = R_{\oplus}\sum_{n=1}^{13}\left(\frac{R_{\oplus}}{R}\right)^{n+1}\sum_{m=0}^{n}\left[g_n^m(t)\cos(m\lambda) + h_n^m(t)\sin(m\lambda)\,P_n^m\cos(\theta)\right]. \tag{22}$$

The inputs are orbital altitude (***R***), colatitude (**θ**) [90° – latitude], longitude (**λ**) and the date (***t***). The right side of Equation (22) is truncated at 13 expansions and has 195 coefficients; the components are Gauss coefficients (***g*** & ***h***) that correct for secular variation, Schmidt normalized Legendre functions (***P***), and the radius of the Earth ($\boldsymbol{R}_{\oplus}$). All coefficients found in Appendix D are valid until 2015.

The conversion of this mathematical model into MATLAB code was completed by Maurice Tivey at the Woods Hole Oceanographic Institution [11]. The IAGA has recently released IGRF-11; however this happened too late for the ACDS code to be modified for it. Additionally IGRF-10 remains valid until after ARC's proposed launch, so there are no plans to update the code at this time.

1.7.3 Polar Regions

As stated before, the Earth's geomagnetic field is often approximated as a simple dipole located at the Earth's center. The geomagnetic poles are the locations where the

long axis of this simple dipole intersects the Earth's surface and where the flux lines should be parallel to the local vertical. Unfortunately, the dipole simplification breaks down at the Polar Regions and the vertical flux lines do not actually coincide with the geomagnetic poles. This necessitates the definition of yet another pole type.

The magnetic pole is the location where the Earth's magnetic flux lines truly are vertical. This is important for ARC as this is the optimal location for the polar bias since the intent is to align the camera axis with the local vertical. In the Northern Hemisphere, the magnetic pole is closer to the Earth's rotational axis than the geomagnetic pole. At the South Pole however, the magnetic pole is significantly displaced from the Earth's axis of rotation. Both magnetic poles have drifted over time and their approximate current locations can be seen in the following diagrams [12].

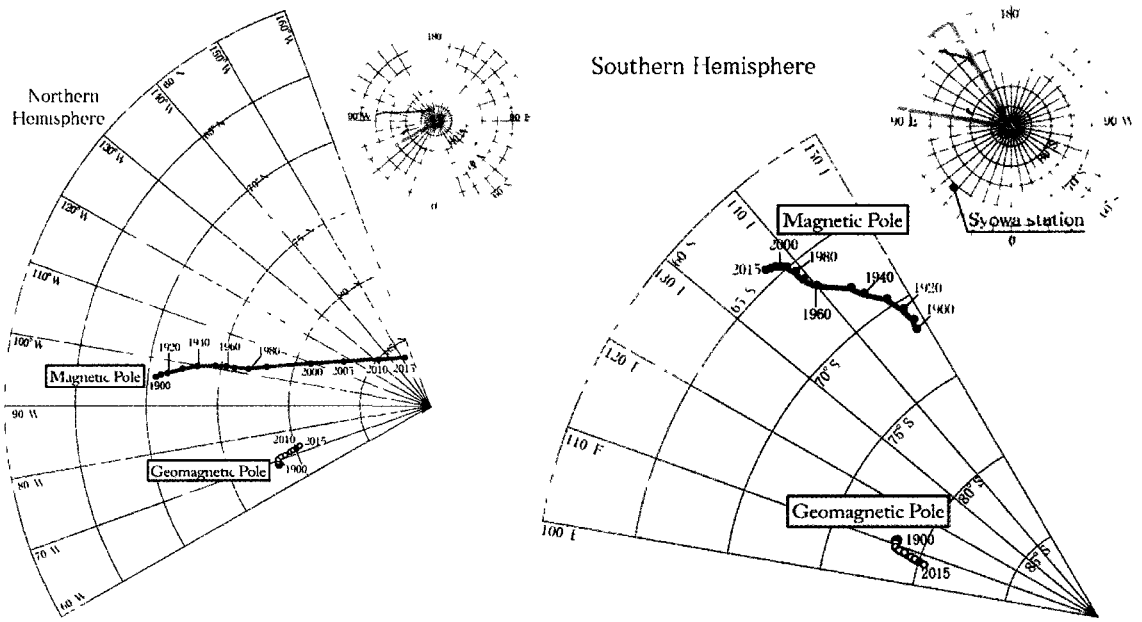


Figure 17: Magnetic Pole Locations [http://wdc.kugi.kyoto-u.ac.jp]

1.7.4 Magnetic Flux Densities

It is also important to consider the components / magnitude of the Earth's magnetic flux density vector. ARC will only achieve a high degree of alignment accuracy if the bias strategy is executed at the proper locations. Aside from ground

commands, the magnitude of the magnetic flux density vector is the only information ARC will have to make bias decisions.

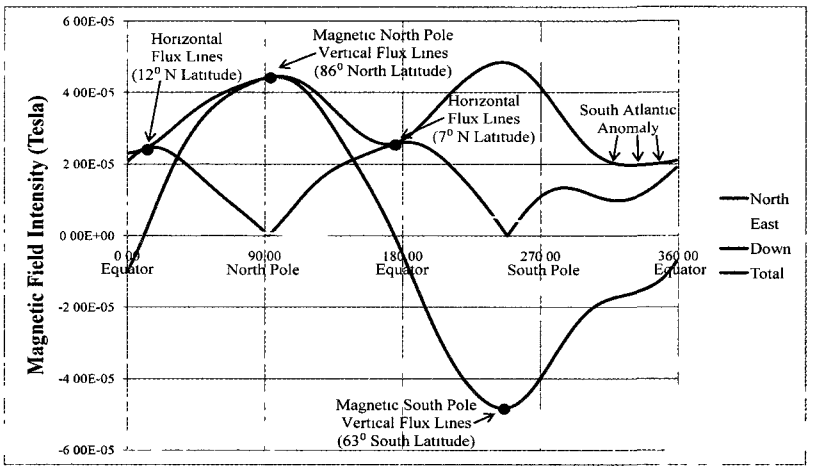


Figure 18: Global Magnetic Flux Densities [0° / 180° Longitude]

The specifics of the switching logic will be discussed in Chapter 3; however, it is appropriate to note a few generalized observations about Figure 18 at this time. The total flux density has two peaks; a smaller one at the North Pole and a larger one at the South Pole. The horizontal flux lines are generally symmetric about the North Pole and asymmetric about the South Pole. The North Pole is the most desirable location for precise alignment. Here the desired component (down) is at its highest and the undesired components (north & east) are at their lowest. An added benefit is that as the orbit undergoes precession due to the Earth's rotation, these qualities remain relatively unchanged as seen in the following figure depicting varying longitudes.

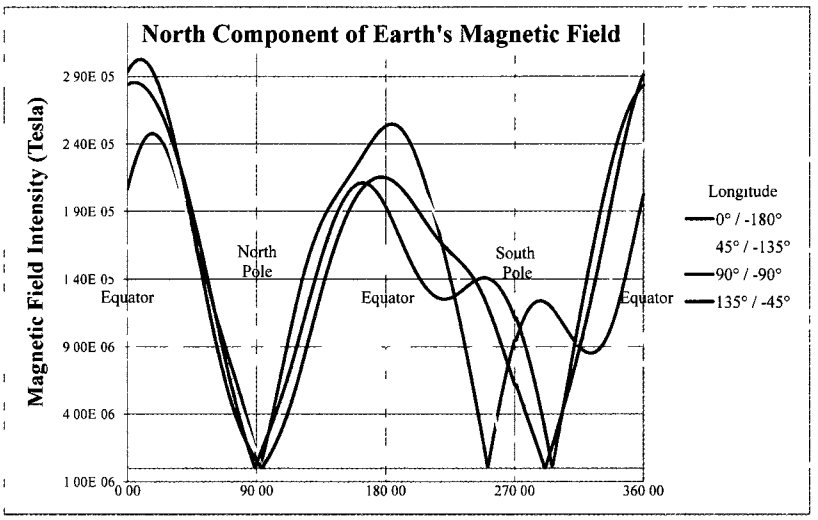


Figure 19: Northerly Component of Earth's Magnetic Field

The North Polar Region is very consistent regardless of the approach longitude. In the south, the condition is the opposite. The approach longitude is very important since the magnetic South Pole is so far north and it is easier for the satellite to miss the highest magnitude of the magnetic flux density vector.

1.7.5 South Atlantic Anomaly

As was previously discussed, when the Earth's magnetic field is approximated by a simple dipole, that dipole should be displaced from the Earth's center of mass by roughly 500 kilometers. This coupled with the Earth's multifarious composition and oblate spheroid shape result in a magnetic field that is non-concentric with the surface of the planet. The South Atlantic Anomaly (SAA) is the near-Earth region where the magnetic field is the weakest. This area is marked in Figure 18 where the **green** line levels off for quite some time. The following image from the ROSAT satellite shows the SAA at 560 km which is just under ARC's proposed orbital altitude [13].

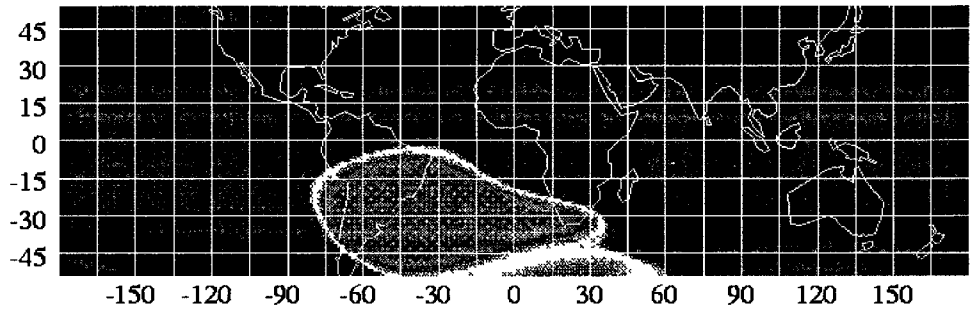


Figure 20: South Atlantic Anomaly [NASA - ROSAT]

It will be shown that the SAA does not impact ARC's detumble phase. It mainly creates issues during the alignment phase; however, by favoring alignment maneuvers in the Northern Hemisphere, it is easy to compensate for SAA irregularities.

1.8 Attitude Disturbance Torques

The following subsections discuss four attitude disturbance torques to which a satellite in ARC's proposed orbit would be exposed. The information is used in Chapter 2 to design LPMTs that will be capable of stabilizing / aligning ARC in the presence of these forces.

1.8.1 Aerodynamic Drag

This force is due to the small amount of the Earth's atmosphere present in low Earth orbit. The maximum torque is given by the following equation:

$$\tau_{aero} = C_D \frac{1}{2} \rho v^2 AL \tag{23}$$

where C_D is the coefficient of drag for a flat plate. Normally this value would not be expected to be more than 2.5 (non-dimensional) for a flat plate perpendicular to the free molecular flow. A is ARC's cross-sectional area, L is the distance between the center of gravity and the center of pressure, and v is the spacecraft velocity. The CDS requires the center of mass to be within 2 cm of the geometric center of the CubeSat, so that value is used in the torque calculation. The density (ρ) requires a few assumptions be made about how the Sun is affecting the upper atmosphere [14].

The major factors contributed from the sun examined here are x-ray output and particle precipitation. The solar x-rays tend to be absorbed at the base of the thermosphere which causes a direct heating effect that propagates upward eventually impacting low orbits. Solar 10 cm radio flux is often a good substitute for the x-ray output and has a normal range of 65 to 300 Solar Flux Units (1 SFU = 10^{-22} W/m^2/Hz). Coronal mass ejections emit particles which travel through space and eventually arrive at Earth. When these particles precipitate from the magnetosphere to the lower thermosphere they emit energy which is absorbed by the atmosphere. This causes heating and a density change. The normal range of precipitation is 0 during quiet times to over 400 (non-dimensional) during major geomagnetic storms. The atmospheric density can be calculated up to 500 km using the following equation:

$$\rho = 6 \times 10^{-10} e^{\frac{-(H-175)}{\frac{900+2.5(Flux-70)+1.5*precipitation}{27-0.012(H-200)}}} \quad (24)$$

where ***H*** is the altitude in kilometers, ***Flux*** is the 10 cm radio flux in SFUs, and precipitation is a non-dimensional quantity. For a 400 km orbit and high flux / precipitation values of 200 / 100 respectively, the density is expected to be 1.07 x 10^{-11} kg/m^3 and the expected torque would be 1.5 x 10^{-7} Newton-meters. At ARC's planned altitude of 600 km and with nominal Sun factors, the disturbance torque would be on the order of 10^{-9} Newton-meters or less.

It will be shown in Section 2 that the LPMTs will easily counteract the aerodynamic forces at the beginning of ARC's lifecycle at 600 km altitude. As the orbit decays, the aero forces will begin to match or exceed LPMT capability depending on Sun conditions. However, there are two methods for dealing with this disturbance torque.

The first is to decrease the distance between the center of pressure and the aerodynamic center. The smaller the distance, the smaller the disturbance torque as seen in Equation (23). A second option is preferred by the author. If the center of mass is moved forward in the ram direction (in front of the center of pressure), it creates positive static stability. The disturbance torques then act in a direction that tends to align the spacecraft with its desired mission orientation. The more the orbit decays, the more the

forces tending to return ARC to its desired alignment will increase. If there are any failures that render the ACDS inoperable, it may become possible to resume ARC's other functions when the aero forces become strong enough to align the spacecraft. For these reasons it is desired that ARC's center of gravity be displaced slightly forward to create a condition of static stability.

The ACDS simulation does not account for aerodynamics forces. The worst case scenario is a high orbit where this force is not helping to align the vehicle. As the orbit decays, the force will tend to properly align the vehicle. As such, the ACDS simulation is designed to prove that ARC is able to achieve the proper alignment without the assistance of aerodynamic forces.

1.8.2 Gravity Gradient

ARC is very close to being a symmetrical spacecraft, but the x-axis is very slightly longer than the other two as seen in Figure 1. As such, gravity gradient torques will tend to align the x-axis with the local vertical which is consistent with the camera pointing in the nadir direction. The camera pointing away from Earth is also a stable condition, so the LPMTs will need to be capable of correcting this condition. The future 3U mass spectrometer version of the satellite will be much more susceptible to gravity gradient torque; however, it will be a benefit since the long axis of the spacecraft is desired to be aligned with the local vertical.

The maximum gravity gradient torque is a function of the minimum / maximum principle moments of inertia and orbital rate. It is governed by the following equation:

$$\tau_{Gravity} = (I_{max} - I_{min})3(\text{orbital rate})^2. \quad (25)$$

The difference in the principle moments of inertia of 0.005 kg·m^2 is obtained from calculations shown in Subsection 2.3.1 below. This makes the extreme gravity gradient torque to be no more than 1.75×10^{-9} Newton-meters. ACDS simulation software shows the normal range of gravity gradient torque during the alignment phase is on the order of 10^{-10} Newton-meters.

1.8.3 Solar Radiation Pressure

The solar winds act much like the aerodynamic forces of the Earth's atmosphere and are mainly a function of the illuminated area and surface reflectance of the satellite. The maximum torque is given by the following equation:

$$\tau_{Solar} = \frac{F_s}{c} A_s(1 + r)L. \tag{26}$$

The solar constant (F_s) is 1358 W/m^2 at the Earth's orbit or one astronomical unit from the Sun. The remaining constants are the speed of light (c), the illuminated surface area (A_s), the surface reflectance (r), and the distance (L) between the center of gravity and center of pressure. This makes the extreme solar radiation torque to be no more than 1.63×10^{-9} Newton-meters. The normal expected torque is on the order of 10^{-10} Newton-meters.

1.8.4 Magnetic

There are several sources of magnetic disturbance torques in CubeSats. The most common one is residual magnetization in structural components caused by operating electronics. ARC will be thoroughly tested in a Helmholtz cage to determine the magnitude of these effects if any. It is important that they be eliminated to the maximum extent possible so that no torque is applied to the satellite when the LPMTs are in the balanced (no torque) condition.

1.8.5 Torquer Design Criteria

The maximum expected disturbance torque is on the order of 10^{-9} Newton-meters. The LPMTs are designed to be approximately 2 orders of magnitude greater than the largest disturbance torque. Rounding up for simplicity, this places the minimum torquing ability to be no more than 10^{-6} Newton-meters.

Chapter 2 LPMT Design

2.1 LPMT Stabilization Concept

The original LPMT stabilization system was envisioned as a spacecraft in a box with 12 torquers, 4 aligned with each axis. This configuration would allow for magnetic dipole moments of $0\boldsymbol{m}$, $\pm 2\boldsymbol{m}$, or $\pm 4\boldsymbol{m}$ in each axis. The box would be constructed out of hollow glass epoxy composite tubes. The torquer cores would be located inside the tubes with their solenoids wrapped around the outside. This novel feature would have added structural stiffness while saving on volume.

Utilizing ⅜ inch diameter, 8 inch long, ALNICO5 (Aluminum Cobalt Nickel) rods with a combined mass of 3.5 kg, a computer simulation showed that the system could stabilize a 1,400 kg spacecraft in a 600 km circular orbit to a local vertical orientation in five to ten orbits. This assumed initial rates of 1° per second or less in each axis. The average power required to settle was 125 mW or less.

ARC's cube shape, the desired orbital altitude of 600 km, and the low-power nature of LPMTs made the same system an appealing choice for attitude control. The next subsections discuss the miniaturization of the system to control a satellite conforming to the CDS definition of a 1U CubeSat.

2.2 Stabilization System Modification

The system described by Polites et al. [8] was extensively redesigned in order to be properly sized for a picosatellite. This was accomplished via an iterative process culminating in a system able to satisfy ARC's mission requirements. In order to understand the design of the individual components, it is necessary to describe a few of the conclusions up front.

2.2.1 Detumble Mode vs. Alignment Mode

The ACDS has two modes of operation. The first is known as the detumble phase. This is the initial mode of operation and arrests any tip-off rates the satellite has at ACDS initialization. It functions exactly as described in Subsection 1.6.4.

The second mode of operation is the alignment phase. During this stage of operation, the ACDS begins the torquer biased alignment described in Subsection 1.7.1. The ACDS enters this mode once the measured spacecraft body rates are within 1% of the commanded rates or on command from the ground station.

2.2.2 Large Torquers vs. Vernier Torquers

The spacecraft in a box design consisted of 4 LPMTs per axis. ARC will also have 4 per axis; however, 2 of them will be larger torquers used for the detumble phase and 2 smaller (vernier) torquers used for the alignment phase. This permits a short detumble time with increased pointing accuracy in the alignment phase.

2.3 Satellite Assumptions

In order to design the proper size LPMT cores and associated solenoids, it was necessary to make assumptions about the satellite. The following subsections detail the assumed mass moment of inertia matrix and capabilities of the power / charging system.

2.3.1 Mass Moment of Inertia

The satellite's size is fixed by the CDS specifications for a 1U CubeSat. The mass must be no more than 1.33 kilograms. The design therefore assumed the correct CubeSat shape at the maximum possible mass which was evenly distributed. The density is simply the mass over the volume, which as calculated as:

$$\rho = \frac{\text{mass}}{\text{volume}} = \frac{1.33\ \text{kg}}{10\text{cm} \cdot 10\text{cm} \cdot 11.35\text{cm}} = 1172 \frac{\text{kg}}{\text{m}^3}. \tag{27}$$

The mass moment of inertia matrix ($\boldsymbol{I}$) is calculated over the volume of the cube as shown below:

$$I_{xx} = \int_{-5cm}^{5cm}\int_{-5cm}^{5cm}\int_{-5.675cm}^{5.675cm} \rho(y^2 + z^2)dx\,dy\,dz = 2.217 \times 10^{-3} kg \cdot m^2 \quad (28)$$

$$I_{yy} = \int_{-5cm}^{5cm}\int_{-5cm}^{5cm}\int_{-5.675cm}^{5.675cm} \rho(x^2 + z^2)dx\,dy\,dz = 2.536 \times 10^{-3} kg \cdot m^2 \quad (29)$$

$$I_{zz} = \int_{-5cm}^{5cm}\int_{-5cm}^{5cm}\int_{-5.675cm}^{5.675cm} \rho(x^2 + y^2)dx\,dy\,dz = 2.536 \times 10^{-3} kg \cdot m^2. \quad (30)$$

The gain used in the ACDS simulation software was based on the above mass moment of inertia matrix calculations. It is important to remember that the gain ($\boldsymbol{k}$) in Equation (20) includes the matrix and will need to be adjusted if there is a significant difference in the physical properties of the flight hardware. The value of the gains used in the simulation is discussed in Subsection 4.2.4.

2.3.2 ARC Power System

The satellite's power system is an adaptation of an off-the-shelf version to be purchased from Clyde Space, which was specifically designed for CubeSat missions. It supplies power at 3.3, 5.0, and 7.5 volts. The ACDS charging circuit will be attached to the 7.5 volt supply. This line provides 8.2 volts when the batteries are fully charged and has its current disconnected if the voltage drops below 6.5 volts.

As discussed in Subsection 1.6.1, the LPMT charging circuit consists of a capacitor which is discharged through a solenoid wrapped around the core. Until adequate tests on the actual power system hardware can be accomplished, it will be assumed that the spacecraft's power system is capable of charging three capacitors every 10 seconds. This is a conservative educated guess from the power systems group based on the design specifications discussed in the subsections below.

In the final design, one capacitor will be dedicated to each axis, so ARC will be capable of flipping the dipole of one core per axis every 10 seconds. The same capacitors will be used for both the large and vernier torquers.

2.4 Large Core Sizing

The goal of large torquer sizing is to quickly arrest tip-off rates with the lowest weight and power requirements possible. The first step is the selection of an adequate core material. From Subsection 1.8.5, it was determined that the LPMT should produce a torque of approximately 10^{-6} Newton-meters. The magnitude of the Earth's magnetic flux density vector varies from approximately 0.2 Gauss at the equator to 0.5 Gauss at the poles in a 600 km orbit. Solving Equation (7) for magnetic dipole moment yields the following:

$$m = \frac{\tau}{B} \tag{31}$$

which mean for the polar region (600 km):

$$m = \frac{10^{-6}\mathrm{N}\cdot\mathrm{m}}{0.5\ \mathrm{Gauss}} = 0.02\ \mathrm{A}\cdot\mathrm{m}^2 \tag{32}$$

and for the equatorial region (600 km):

$$m = \frac{10^{-6}\mathrm{N}\cdot\mathrm{m}}{0.2\ \mathrm{Gauss}} = 0.05\ \mathrm{A}\cdot\mathrm{m}^2. \tag{33}$$

2.4.1 Core Miniaturization

The original stabilization system utilized ALNICO5 as the core material. This grade has a high residual induction ($\boldsymbol{B_r}$) of 12,800 Gauss with a low coercive force of 640 oersteds, equating to 1,280 milli-Tesla (***mT***) and 51 kilo-amps per meter (***kA/m***). This material was chosen because the high $\boldsymbol{B_r}$ means the LPMT will provide a lot of torque for its size while the low coercive force implies it will take little power to flip the dipole. The final reason is that ALNICO5 is manufactured by many companies and is easy to obtain in a wide variety of sizes. However, these properties must be reevaluated for every project in order to develop the best LPMT for each job.

ALNICO5 is a very brittle material. A bar of $^1/_{16}$ inch diameter is the smallest size most manufactures will produce. In the United States, it is common to find 1 inch length (***L***) bars at that diameter (***D***). According to Polites, it is desired to have a length to diameter ratio greater than 10. This prevents self-demagnetization of the core and fixes the shortening ratio ($\boldsymbol{R_s}$) to a known value of 0.75 (unit-less). Knowing the permeability

of free space (μ_o), it is possible to calculate the projected magnetic dipole moment of the core from the following equation:

$$m = \frac{\pi D^2}{4}\left(\frac{B_r}{\mu_o}\right) L \cdot R_s = 0.038\, A \cdot m^2. \quad (34)$$

Although this value falls within the limits of Equation (32) & (33), it was discovered in the simulation that a slightly lower magnetic dipole moment, coupled with the 10 second charging time, produced a better response, which is discussed in Subsection 4.3.2. To lower the magnetic dipole moment, a different core material was selected.

ARC's large torquers are composed of ALNICO1. This material has the same relative material strength and density of ALNICO5; however, it has a lower residual induction of 7,200 Gauss (720 mT) and a coercive force of 470 oersteds (37 kA/m). This means that it will produce less torque and has the benefit of requiring a smaller H-field in order to become saturated. Using Equation (34), the magnetic dipole moment of the ALNICO1 core is:

$$m = \frac{\pi D^2}{4}\left(\frac{B_r}{\mu_o}\right) L \cdot R_s = 0.022\, A \cdot m^2. \quad (35)$$

This is the value used in the ACDS simulation software for the large torquers utilized in the detumble phase of operation. Each of these cores has a mass of 0.347 grams. The total mass of all 6 cores required for detumble is 2.081grams.

2.4.2 Solenoid Sizing

The solenoids for ARC's large torquers are composed of 6 layers of 26 gauge wire. The wire's 0.4039 mm diameter allows for 300 turns around the core; which in simulation was shown adequate to drive ALNICO1 to saturation. The practical problem was wrapping a very brittle core with 300 turns of wire. The original concept of utilizing glass-epoxy composite tubes for the box structure of the CubeSat with the cores inside and solenoids wrapped around the outside was not possible for ARC. The CDS is very specific about the structural composition of the satellite so that it can be properly integrated into the P-POD. Solenoids partially on the outside of the skin would not be compatible with the rails the satellite slides down during P-POD ejection. It was decided

that moving the LPMTs inside the satellite's skin would be easier than attempting a non-standard P-POD integration. This necessitated a new approach to wrapping the solenoid around the very thin and easily broken cores. The UAF electrical engineering students innovated a novel approach to accomplish this task.

A plastic bobbin was constructed on a lathe by shaving down a hollow cylinder. The inner diameter of the cylinder is just large enough to accept the core. The outer diameter approximates a wrapped core's diameter. The bobbin can then be wrapped with gauge 26 wire on a separate machine without any danger of breaking the core material. The completed test version is shown in Figure 21. For the testing phase, it is very easy to swap core materials since they slide easily in and out of the bobbin.

Figure 21: Wrapped Solenoid Bobbin

The choice of 26 gauge wire was based on UAF having a quantity of it on hand and the failure of a 32 gauge version in testing. The resistance in the 32 gauge coil was much higher than predicted and resulted in the ALNICO1 cores failing to reach saturation. It was decided to convert to 26 gauge wire because of the lower resistance in the larger cross-sectional area. This gauge is capable of handling the current requirements and its small diameter results in low solenoid weight. However, the gauge of wire could easily be changed again if conditions warranted that to occur.

2.4.3 Capacitor Sizing

A capacitor of sufficient size must be selected so that the discharge pulse is capable of driving the core to saturation in either direction no matter what the initial condition of the core. This is an iterative process begun with a small capacitance which is slowly increased utilizing Polites's 1971 equations developed into software shown in Appendix E. The following figure shows core magnetization as the capacitor size is increased from below core saturation to a point above it.

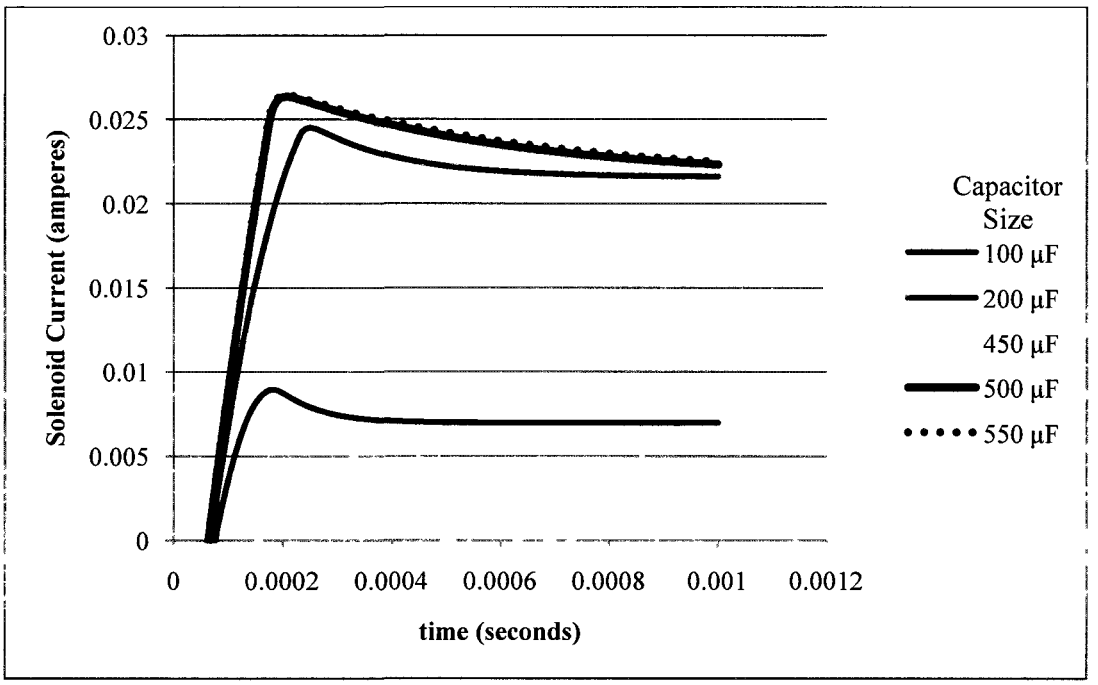


Figure 22: Capacitor Size Comparison (7.0 Input Voltage)

The capacitor size is increased slightly and the saturation of the core increases slightly. This continues to a point where the change is negligible. In this case, a 500 micro-Farad (***µF***) capacitor is sufficient to drive the ALNICO1 cores to saturation. This remains the case even at input voltages as low as 6.5 volts, where the current would be disconnected by the power system for dropping below the minimum voltage. As a side note, the simulation shows the core's residual magnetization stabilizes at approximately 0.022 A-m^2 as predicted in Equation (35). For a 500 µF capacitor charged with 7.5 volts (the design input voltage), the energy dissipated during each core pole flip is shown in the following equation:

$$E = \frac{1}{2}(500\ \mu F)(7.5\ V)^2 = 0.014\ \text{Joules}. \quad (36)$$

It is unknown if the solenoids will experience any eddy current losses once they are incorporated into the CubeSat structure. These losses will be avoided to the maximum extent possible but may be induced by the materials used in ARC's construction and their relative proximity to the solenoids and charging circuit components. If eddy current losses are experienced once the circuit is incorporated into the structure, the corrective course of action will be to increase the capacitor size. However this is unlikely since the current design supplies twice the field required to saturate the cores.

As discussed in Subsection 1.6.2, if the applied field is $\boldsymbol{H_s}$ or higher, the core will be driven to saturation. For ALNICO1 this value is 4 x 10^4 amperes per meter. The gauge 26 cores are well in excess of this as seen in the top half of the following figure.

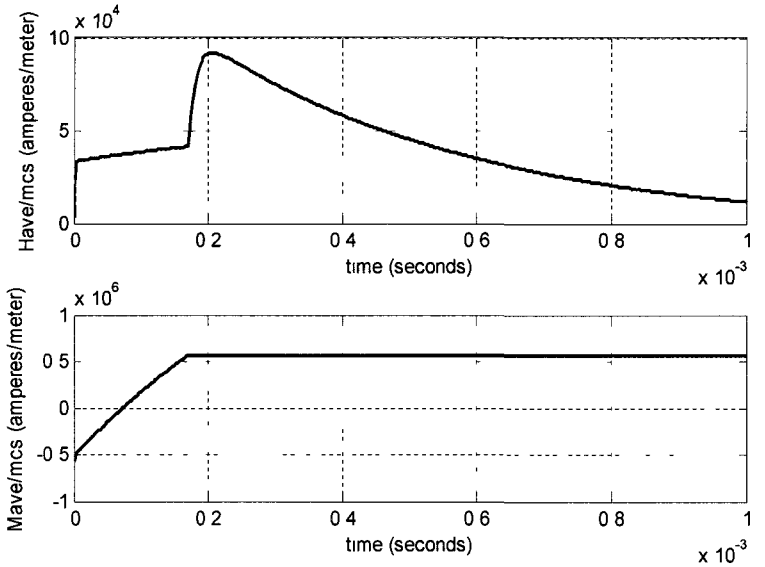


Figure 23: Solenoid H-Field

2.4.4 Charging Circuit Properties

The charging circuit was based on a 7.5 input voltage. During capacitor discharge the 26 gauge wire is subjected to a peak current of 7.4 amperes. The American Wire Gauge rating for this gauge is only 2.2 amperes; however due to the short duration of the

pulse and time between pulses, testing has shown no heat buildup or burn though. The following figure shows the short duration of the current pulse which is less than 0.1 milliseconds.

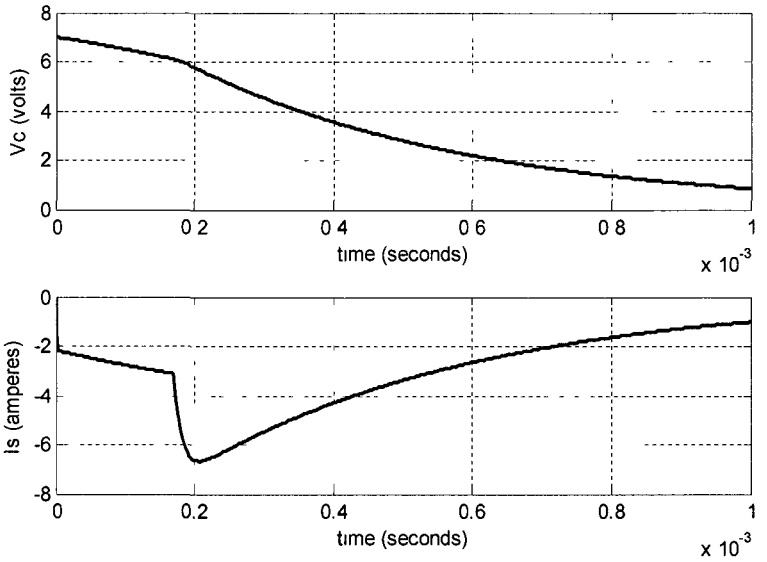


Figure 24: Charging Circuit Voltage / Current

2.4.5 Circuit Testing Results

The circuit shown in Figure 10 was constructed in the laboratory. The power supply was set to 7.5 volts. The original bobbin was wrapped with 500 turns of 32 gauge wire as discussed in Subsection 2.4.2. A 250 micro-Farad capacitor and 1000 ohm charging resistor were used, which is correct for that size of wire. ALNICO1 cores of $^{1}/_{16}$-inch diameter and 1-inch length were obtained from a commercial source.

The laboratory experiments showed that the design was unable to flip the magnetic dipole moment of the large torquer cores. After further investigation, it was determined that the original simulation assumed the solenoid wire was wrapped in a single layer. ARC's 32 gauge solenoid was 4 layers of wire wrapped around the 1 inch length bobbin. This equated to exactly 500 turns of wire. The total length of the solenoid wire is calculated as follows:

$$L_{wire} = \sum_{n=1}^{Layers} L_{torquer} \sqrt{\left\{\pi\left[(2n-1) + \frac{Radius_{bobbin}}{Radius_{wire}}\right]\right\}^2 + 1} = 3.771 \text{ m.} \quad (37)$$

This computation assumes the wire is wrapped directly around the core. The number will have to be recomputed once the diameter of the production model of the bobbin is determined. Only a few experimental versions exist at this time, which have a diameter of 2.08 millimeters. The resistance of the wire is then calculated as follows:

$$Resistance_{solenoid} = \frac{Length_{wire}\rho}{a} = 2.005 \text{ ohms.} \quad (38)$$

In this formulation, the cross-sectional area (***a***) is for the gauge 32 wire and the resistivity ($\boldsymbol{\rho}$) of copper is 1.72×10^{-8} ohms-meter.

The original simulation assumed the solenoid was a single layer of wire and calculated the resistance as follows:

$$Resistance_{solenoid} = \frac{\pi Diameter_{wire}\,(Turns)\,\rho}{a} = 1.326 \text{ ohms.} \quad (39)$$

The total difference is 0.6795 ohms. To compensate, this extra resistance was added into the simulation in Line 43 of the code. It was then necessary to recalculate the required voltage and capacitance. With the added resistance, it was determined that a capacitor of 400 μF was required at 10.5 volts in order to flip the dipole. The simulation showed the pole is just barely flipped with this configuration. Since the required 10.5 volts is higher than the power system can supply and the measured resistance in the coil was 3.77 ohms (almost twice the predicted value), it was decided to redesign the coil with 26 gauge wire.

The first prototype of the new coil has been fabricated. Initial testing has shown the coil resistance closely matches the calculated value and that core saturation is possible with as little as 6 volts. The solenoids were designed to function at 7.0 volts and had extra resistance added into the design simulation shown in Appendix E below to account for unforeseen losses. At this point, the coil would have to be subjected to impossibly high eddy current losses for the core to fall short of the saturation point. This should correct the deficiencies of the 32 gauge design. Based on the length of the wire, it is assumed that each coil will have a mass of 6.2 grams. The total weight of 12 solenoids will be 74 grams. This will be verified with the first production round of solenoids.

2.4.6 Saturated Core Analysis

It was necessary to verify the magnetic dipole moment of the saturated cores matched the results predicted in the simulation software in Appendix E. This was done by using the method developed by the University of Alabama in 2004 [15]. It is based on the rotational dynamics equation of motion for the torquer:

$$I\ddot{\theta} = -m \cdot B_{horiz}\sin(\theta) \tag{40}$$

where $\boldsymbol{B_{horiz}}$ is the horizontal component of the local magnetic flux density vector, $\boldsymbol{\theta}$ is the angle between $\boldsymbol{B_{horiz}}$ and the torquer's magnetic dipole moment, and $\boldsymbol{I}$ is the moment of inertia of the torquer core. Assuming small angles, the equation may be rewritten into the following form:

$$I\ddot{\theta} + \omega_n^2\theta = 0 \tag{41}$$

where:

$$\omega_n = \sqrt{\frac{m \cdot B_{horiz}}{I}}. \tag{42}$$

Since the system is second order with zero damping and exhibits sustained oscillations if an initial condition for $\boldsymbol{\theta}$ is provided, the period of oscillation ($\boldsymbol{T_p}$) is:

$$T_p = \frac{2\pi}{\omega_n} = 2\pi\sqrt{\frac{I}{m \cdot B_{horiz}}}. \tag{43}$$

Solving for magnetic dipole moment results in the following:

$$m = \left(\frac{2\pi}{T_p}\right)^2\left(\frac{I}{B_{horiz}}\right). \tag{44}$$

The final computation is to calculate the torquer's moment of inertia. The core is weighed in order to obtain its mass and measured down the long axis to determine the length. The moment of inertia is then calculated from the following formula:

$$I = \frac{1}{12}\text{mass} \cdot \text{length}^2 = \frac{1}{12}(0.347\text{ g})(25.4\text{ mm})^2 = 1.865\text{x}10^{-8}\text{kg} \cdot \text{m}^2. \tag{45}$$

A core is placed inside a solenoid bobbin. The charging circuit capacitor is discharged several times in the same direction in order to ensure that the core is at saturation. The core is removed from the bobbin and has a very thin string tied around its

center of mass. At the same time, UAF's Helmholtz cage power supplies are set to provide a horizontal magnetic field of 0.1 gauss at a point inside the cage. The field is verified with a magnetometer to an accuracy of 0.001 gauss.

The magnetometer is removed from the field and the core is suspended to the point where the 0.1 gauss horizontal field exists. The torquer core will align itself with the local magnetic field. The core is then rotated in the horizontal plane out of alignment with the field and released. It will begin to oscillate. The period of the oscillation is measured and entered into Equation (44) above. It is suggested that the total time of 5 oscillations is recorded to limit the influence of aerodynamic drag over longer periods. This value divided by 5 is entered into Equation (44) above as the period. The result is an estimate of the magnetic dipole moment of the core material.

2.4.7 Saturated Core Results

The large torquer cores were tested in the Helmholtz cage at 0.1, 0.2, and 0.5 Gauss. The total time for 5 complete oscillations was recorded and used to determine the period in Equation (44) above. The results are shown in Table 1 below.

Table 1: Large LPMT Dipole Results

Field	0.1 G	0.2 G	0.5 G
Time for 5 oscillations	9.1	6.2	4.2
	9.0	6.1	4.0
	9.1	6.2	4.1
	9.0	6.1	3.9
	8.8	6.3	4.1
	8.9	6.2	4.0
	8.9	6.3	4.3
	9.1	6.2	4.0
	8.9	6.3	4.0
	9.0	6.1	3.9
Ave (sec)	8.9	6.2	4.0
Dipole ($A\text{-}m^2$)	0.02282579	0.023942	0.02278

These results approximated the expected value of 0.022 A-m^2 predicted in Equation (35). This is the same method to be used to determine the magnitude of the vernier torquer magnetic dipole moment, once the first batch of them has been produced.

2.4.8 ALNICO5 Cores

ALNICO5 is a much easier material to obtain than the preferred ALNICO1. In the event that future versions of ARC must make use of ALNICO5, the only change required is to increase the capacitor size. Simulation showed the 500 μF capacitor was able to drive the core to saturation. This resulted in the same energy dissipation of 0.014 joules per torquer flip. This of course is predicated on the resistance in the charge / discharge circuit. ALNICO1 can still be driven to saturation with an extra 0.5 ohms in the circuit. This is not the case for ALNICO5, which may require the solenoid wire diameter be increased to compensate for a higher than expected resistance in the circuit.

2.5 Vernier Core Sizing

The large torquers were unable to achieve the alignment goals of ARC's mission requirements. A compass needle centers itself on north due to the friction of the surrounding liquid; a spacecraft experiences very low friction in a 600 km orbit. The biased torquers do not center themselves on the field lines but rather oscillate about them just like a pendulum. In the absence of friction, this oscillation can only be damped with the unbiased torquers in the other 2 axes. The large torquers, when used with a 10 second charging time, were far too powerful to provide the desired damping.

2.5.1 Design Requirement

A vernier core size of 0.00011 A-m^2 was determined to be optimal for ARC's alignment phase. This is $^1/_{200}{}^{th}$ the magnetic dipole moment of the large torquers and was determined by experimentation with the ACDS simulation software. In order to minimize complexity and save on construction costs, it was highly desired to utilize the

same core, solenoid, and capacitor size. The problem was to locate a core material with the correct magnetic properties. In this case it would be a very weak permanent magnet.

The market for such weak magnets is extremely low and a search of material properties revealed few suitable candidates. The alternative was to use strong permanent magnets of extremely small size. However, in order to maintain a length to diameter ratio of 10, the cores would need to be impossibly thin, which would make them very easy to break. The solution was to make the cores larger without increasing their magnetic dipole moment.

2.5.2 Fabrication

The University of Alaska Institute of Northern Engineering, Advanced Materials Group (***AMG***) is an innovator of novel materials. For the vernier torquers they suggested beginning with a plastic core of the correct size. The core is placed inside of a vacuum chamber while a laser is targeted at a hard magnetic material. A vapor deposition process deposits the hard magnetic material onto the plastic core. The coating is uniform with a thickness that can be controlled to the angstrom level if required.

Any hard magnetic material could be applied to the plastic core provided it is properly sized to fit within the application equipment. Instead of ordering a new material, it was decided to utilize a Permalloy80 supply currently on hand. This material has a residual induction of 3,700 gauss and will be driven to saturation by capacitors designed for ALNICO1 or ALNICO5.

A vernier torquer, 1 inch in length, composed completely of Permalloy80 would have a diameter of 0.159 millimeters, as can be seen by adapting Equation (34):

$$m = \frac{\pi(0.159\ \text{mm})^2}{4}\left(\frac{3700\ \text{G}}{\mu_o}\right)(1\ \text{in}) \cdot (0.75) = 0.00011\ \text{A} \cdot \text{m}^2. \tag{46}$$

If this amount of material were spread on to a plastic core, the required thickness would be 0.004 mm. This will be the starting thickness of the vernier cores which have yet to be produced.

2.5.3 Thickness Adjustment

It is assumed that the fabricated cores will not have the same performance as solid cores with a smaller diameter. The fabricated cores will need to be tested in order to determine their magnetic dipole moment. It will then be possible to order a second set of cores with an adjusted Permalloy80 thickness.

2.5.4 Torque Amount

The torque supplied by the vernier cores is very small, but large enough to overcome the various attitude disturbance torques present. The lowest magnitude of the Earth' geomagnetic field for a 600 km orbit is 0.2 Gauss and occurs at the equator. Utilizing Equation (7), a torque of 10^{-9} Newton-meters will be produced. This is of the same order magnitude as the extreme case attitude disturbance torques and an order of magnitude below the average expected torques. Additionally, with the exception of solar radiation pressure, all of the disturbance torques tend to push ARC back towards the desired alignment, provided the center of gravity is displaced slightly forward. As such, during times of abnormally high solar activity, ARC may be temporarily pushed from its desired alignment. This becomes less likely as ARC's orbit decays and aerodynamic drag (an attitude aligning force) increases along with the magnitude of the Earth's geomagnetic field (increasing LPMT available torque).

Chapter 3 Rate & Position Determination Systems

3.1 Purpose of System

The rate determination system utilizes data from a 3-axis magnetometer to calculate the spacecraft's body rates. The difference between measured body rates and commanded body rates, discussed in Subsection 1.6.4 is the mechanism by which the ACDS determines which LPMTs will have their poles flipped.

As discussed in Subsection 1.5.9, many CubeSats with onboard attitude determination systems calculate body rates from the time rate of attitude change. This indirect method of rate determination will not work on ARC because the satellite is never required to know its own attitude.

Since the magnetometer is a must for ACDS switching logic, utilizing the same device as a means of rate determination conserves the greatest amount of space and power. Although many CubeSats use magnetometers coupled with an onboard model of the Earth's geomagnetic field for attitude determination, a search of the literature has not shown any that utilize magnetometers for direct rate determination. It is generally not required since the $\dot{B}$ controller discussed in Subsection 1.5.10 does not require it and the time rate of attitude change can be used to determine body rates.

3.2 Magnetometer Placement

The magnetometers are placed in a position within the CubeSat where they will receive the lowest interference from the onboard electronics and the LPMTs. In the absence of these disturbances, the same readings of the Earth's geomagnetic field will be obtained from any point within the satellite. As such, they can be placed in the most convenient position as long as the magnetometer axes are aligned with the spacecraft body axes. If their final placement is misaligned, a rotation matrix will need to be derived that corrects the reading to the spacecraft body frame. To save on computation time and reduce errors this should be avoided if possible.

3.3 Cycle Time

Until samples of the flight hardware can be obtained, it is assumed that the magnetic environment inside the CubeSat will be very noisy during the capacitor charge / discharge cycle. The simulation has shown that the discharge completes within 1 second. Assuming the Clyde power system is able to recharge the capacitor within 6 seconds, the remaining 3 seconds of the cycle are available for relatively clean magnetometer readings. At 10 hertz, 30 magnetometer readings can be taken inside the clean window. Rate determination testing, in the Helmholtz cage, will be required to refine the collection times and number of data points required.

3.4 Basic Body Rate Determination

ARC's body rate determination strategy is based on the definition of velocity in a rotating reference frame. The familiar form of the equation is shown below:

$$\frac{d}{dt}\vec{r}_{inertial} = \frac{\partial}{\partial t}\vec{r}_{body} + \vec{\omega} \times \vec{r}. \tag{47}$$

In the case of the spacecraft, the position vector (***r***) will be substituted for the magnetic flux density vector (***B***):

$$\dot{B}_{inertial} = \dot{B}_{body} + \vec{\omega} \times \vec{B}. \tag{48}$$

ARC's sample period for the flux density vector is only 3 seconds. The IGRF-10 model shows no significant change to the vector over that period of time in the projected orbit. It is therefore valid to assume that there is no time rate of change for the magnetic flux density vector in the inertial frame of reference.

$$0 = \dot{B}_{body} + \vec{\omega} \times \vec{B}. \tag{49}$$

The equation is then rewritten into the following form:

$$\dot{B}_{body} = \vec{B} \times \vec{\omega}. \tag{50}$$

Since the magnetometer will be unable to detect rotations about the magnetic flux density vector, it is helpful to break the spacecraft body rates down into those parallel to the flux vector (undetectable) and those perpendicular to it (measureable).

$$\vec{\omega} = \overrightarrow{\omega_{\perp}} + \overrightarrow{\omega_{\parallel}}. \tag{51}$$

It is now possible to substitute Equation (51) into Equation (50) as seen below:

$$\dot{B}_{body} = \vec{B} \times (\overrightarrow{\omega_{\perp}} + \overrightarrow{\omega_{\parallel}}) = \vec{B} \times \overrightarrow{\omega_{\perp}} + \cancel{\vec{B} \times \overrightarrow{\omega_{\parallel}}}. \tag{52}$$

The parallel term is dropped since the magnetometer cannot detect this motion. This term has the strikethrough shown in Equation (52) above. Both sides of the equation are then crossed with the magnetic flux density vector.

$$\dot{B}_{body} \times \vec{B} = (\vec{B} \times \overrightarrow{\omega_{\perp}}) \times \vec{B} = (\vec{B} \cdot \vec{B})\overrightarrow{\omega_{\perp}} - \cancel{(\overrightarrow{\omega_{\perp}} \cdot \vec{B})\vec{B}}. \tag{53}$$

Since the furthest right term in Equation (53) equates to zero, it is also dropped. The spacecraft body rates are therefore approximated from the magnetic flux density vector time rate of change in the spacecraft body reference frame.

$$\overrightarrow{\omega} = \frac{\dot{B}_{body} \times \vec{B}}{(\vec{B} \cdot \vec{B})}. \tag{54}$$

The time rate of change can be determined by means of a central difference formulation.

$$\dot{B}_{body} = \frac{\vec{B}(k+1) - \vec{B}(k-1)}{2\Delta t}. \tag{55}$$

The end result is that spacecraft body rates are obtained from a succession of three magnetometer readings:

$$\overrightarrow{\omega} = \frac{[\vec{B}(k+1) - \vec{B}(k-1)] \times \vec{B}(k)}{2\Delta t(\vec{B} \cdot \vec{B})}. \tag{56}$$

Testing of the system in the controlled environment of the Helmholtz cage will be required once a working model is developed. This will determine if the system can operate from the simple finite difference seen above or if a more elaborate method discussed in the following subsection is required.

3.5 Magnetometer Noise

Magnetometers are often publicized as having very little associated noise; however, the magnetometer inside UAF's Helmholtz cage detected significant amounts of noise. This error was most likely induced by noise in the computer card used to collect the data. Versions of the magnetometer to be integrated into ARC's ACDS have not been obtained yet, so no testing information is available at this time.

The following subsection details a method for smoothing the data and thus correcting for noise in the magnetometer's data collection. The method can be applied to the flight hardware if required.

3.6 MATLAB Line Smoothing

MATLAB contains several built-in functions capable of batch curve fitting magnetometer data. The most effective approach was to line fit the data with the polyfit function. The flux data was collected from a stationary magnetometer in the UAF Helmholtz cage at 10 hertz as was suggested in Subsection 3.3. The data was broken down into 30 point segments that could be collected during ARC's magnetically quiet period.

Although polynomials of higher order magnitude were tested, a simple first order one was found to be the most effective. In this case, the first coefficient of the polynomial is also the first derivative of the entire function. This information was used in Equation (54) as the $\dot{\boldsymbol{B}}$ term. The last magnetometer reading was used for the current flux field ($\boldsymbol{B}$) term. The code used for the test can be found in Appendix F.

In this test the magnetometer was fixed so the calculated body rates should be zero for each run. The following results show degrees per 10 second period.

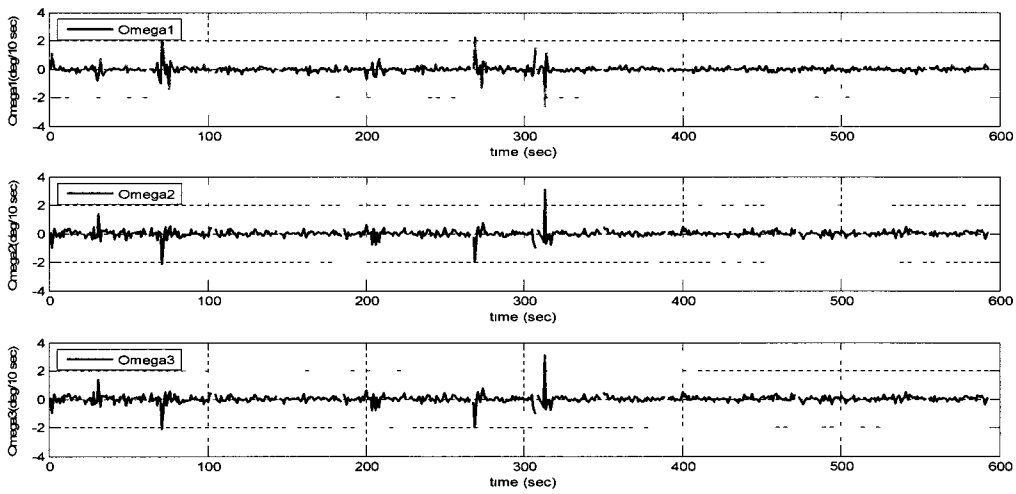


Figure 25: Calculated Spacecraft Body Rates (Degrees / 10 Seconds)

Aside from a few small deviations, the calculated body rates will result in less than a 1° movement during the 10 second charge / discharge period. In most cases, the calculated body rates are significantly lower than that. This has been shown to be sufficient for attitude control in the ACDS simulation software. These results will need to be verified for a rotating magnetic field, since the current results reflect only a stationary field.

3.7 Position Determination Logic

The position determination system relies on the magnetometer's ability to determine the magnitude of the Earth's geomagnetic field. The North Pole will be passed at a very rhythmic rate and has the 2nd highest magnitude of anywhere in the orbit. The timing at the South Pole is less defined due to the offset of the South Magnetic Pole, but this point is marked with the overall highest magnitude field. The equatorial regions are close to the lowest magnitude fields.

3.7.1 Automatic Position Determination

The automatic position determination mode maps the interval between North Pole crossings. The time between polar crossings is used to determine the orbital rate. The relative magnitude of the flux field can then be used to determine spacecraft position in relation to the polar and equatorial regions.

The complexity of this project is intensified by the satellite not flying directly over the magnetic poles on each orbit and the magnetic environment created by the torquers. This will necessitate that the flux environment around the magnetometer is clearly understood and that it can be calibrated out of the readings.

How close ARC gets to the poles on each orbit changes due to Earth's rotation. Although a very complex pattern, it is relatively well defined and somewhat repeatable. It is therefore suggested that the problem can be solved via the use of fuzzy logic. The satellite measures the maximum / minimum intensities of the geomagnetic field over several orbits. It can then reference these to a lookup table to predict its position in orbit.

3.7.2 Manual Position Determination

Timing is the backup option. ARC's orbital rate will be relatively constant for long periods. If this rate and a clock time of when ARC crosses the North Pole are uploaded, the satellite will be able to operate independently. Since the polar and equatorial crossings occur at known rates, their relative positions can be calculated from an onboard clock.

Position information is not required for the detumble phase. ARC can execute the LPMT biased alignment off of timing alone and achieve mission requirements once the data upload is completed.

3.7.3 Adjustment Options

The biased logic utilizes alignment windows discussed in Subsection 4.2.8. These values should be adjustable from the ground station. This will permit corrections to the logic to be incorporated into the manual and automatic modes. Since this is an experimental spacecraft, it is desired to have control over a few of the ACDS decision parameters. Adjusting these parameters in flight may make it possible to correct for unwanted attitudes or improve alignment based on actual performance. The recommend adjustable parameters are discussed in Subsection 4.5.12.

Chapter 4 ACDS Simulation Software

4.1 EXIST Origin

The ACDS simulation software is a continuation of code developed to simulate the Energetic X-ray Imaging Survey Telescope (EXIST) which was proposed in 2001 [16]. EXIST was to be a hard x-ray imaging all-sky deep survey mission, but was not ranked for funding in the Astro2010 Decadal Survey.

The original concept utilized reaction wheels with magnetic torquers for desaturation purposes. The code was modified in 2003 to simulate the spacecraft in a box by removing the reaction wheels, converting the magnetic torquers to LPMTs, and changing the mass moment of inertia properties.

The code was modified a final time for ARC. The mass moment of inertia and LPMT sizes were reduced to reflect a CubeSat. Additionally, logic was added to simulate the vernier torquers and LPMT biased alignment logic.

4.2 ACDS Simulation Code

The code contains comments and references to this document to help the reader understand the how the simulation is designed to operate. It can be found in Appendix A-C and electronically on the enclosed CDROM. The following subsections expand upon a few sections that might otherwise be difficult to follow.

4.2.1 ODE45

This built-in MATLAB function uses a variable step Runge-Kutta Method to solve differential equations numerically. In the ACDS simulation software, ODE45 is integrated over the 10 second charge cycle. The routine (titled ExistSub2.m in the Appendix B code) is passed the current spacecraft body rates, attitude, and magnetic dipole moment commands. Accounting for the influence of the LPMTs and gravity gradient torques, a new attitude and spacecraft body rates are determined.

The ACDS then makes decisions as to which torquers are to have their poles flipped. This is applied by using Equation (20) and then sending those results into a quantizer (shown in Figure 14) for each axis. The commands are sent to ODE45 and the 10 second cycle is repeated. To test different charge cycle times, ODE45 is integrated over the desired charging window.

4.2.2 Gravity Gradient Torques

For a satellite in a circular orbit, the gravity gradient torques ($\boldsymbol{T_{gg}}$), in the body axis system, are calculated from the following expression:

$$T_{GG} = 3\Omega_o^2 \tilde{r} I r. \tag{57}$$

The orbital rate (Ω_o) is obtained from the Earth's mass ($\boldsymbol{M_\oplus}$), the gravitational constant ($\boldsymbol{G}$), Earth's radius ($\boldsymbol{R_\oplus}$), and orbital altitude ($\boldsymbol{h}$):

$$\Omega_o = \sqrt{\frac{G M_\oplus}{(R_\oplus + h)^3}} = \sqrt{\frac{3.98 \times 10^{14} \frac{m^3}{sec^2}}{(6.37 \times 10^6 \text{ m} + 600 \text{ km})^3}} = 0.0011 \frac{rad}{sec}. \tag{58}$$

The position vector ($\boldsymbol{r}$) in Equation (57) is a unit vector in a direction from the center of the Earth to the spacecraft's center of mass. The components of this vector are used to compute the final matrix:

$$\tilde{r} = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}. \tag{59}$$

As the spacecraft's attitude changes, T_{gg} also changes. As such, it must be continuously recalculated during the time integration cycle.

4.2.3 IGRF-10 Field Model

The Geomagnetic field model was downloaded from Maurice Tivey's ftp site at the Woods Hole Oceanographic Institute [11]. It is maintained as its own subroutine (called magfd.m in Appendix C code) so that is may be called multiple times during the ODE45 integration period. The code was modified from providing the flux density vector only in the LNED frame to providing it in both LNED and the spacecraft body

frames. The main inputs are discussed in Subsection 1.7.2; however, there are three additional inputs to be defined.

The first is the date. This is required so that the proper secular variations can be applied to the model. It is required to be in decimal format, so the ACDS uses 2012.5 for this input. It was originally assumed that ARC would be launched sometime mid-2012, so that date was appropriate. No changes in performance were noted when the date was changed ±2 years, so it was fixed to reduce computation time. Although the proposed launch date has slipped to early 2013, it will have no impact to the results contained herein. However, this date can easily be modified for future simulations if required.

The next input is the ITYPE. The simulation uses type 2 which means all altitude references are measured from the center of the Earth in kilometers. Type 1 denotes altitude measured above sea level, which is inappropriate for this code.

The final input is a direction cosine matrix. This matrix is calculated from the quaternion that tracks the body frame in relation to the ECI frame, by utilizing Equation (5) above. Using this matrix and the transpose of Equation (2), the flux density vector is converted to the body frame. The total conversion process is LNED to ECEF to ECI to body. This is important because the onboard magnetometer will only measure the Earth's magnetic field in body coordinates.

4.2.4 Detumble & Alignment Gains

The ACDS simulation gains are shown in Line 101 of the code in Appendix A. This number is the product of the gain and the mass moment of inertia matrix as shown in Equation (21) above. The value was determined experimentally via the ACDS simulation software. It was varied until the end-state body rate error of the detumble phase was at its lowest amount. If the mass moment of inertia matrix changes in the future, the gain should be divided by the old inertia matrix and then multiplied by the new matrix. This should be a sufficient correction for the flight hardware. Simulation runs should be accomplished to verify this conclusion.

4.2.5 Alternate Stable Alignments

ARC's desired alignment is discussed in Subsection 1.7.1 above; however, there are two alternate stable conditions where $\omega_{command}$ and torquer biases are both being satisfied but the attitude is incorrect. The conditions are defined here, while their corrective actions are discussed in the next subsection. The first of these is the retrograde orbit.

In the proper attitude, the spacecraft's z-axis and the orbital O_3 axis are aligned. In a retrograde orbit, they point in opposite directions. At the poles and the equator, where the biases occur, the correct body axes are aligned with the local magnetic field. However, outside of the bias regions there is nothing to force the spacecraft's z-axis to align with the O_3 axis. In Figure 26, the O_3 axis is out of the page but the spacecraft z-axis is into the page.

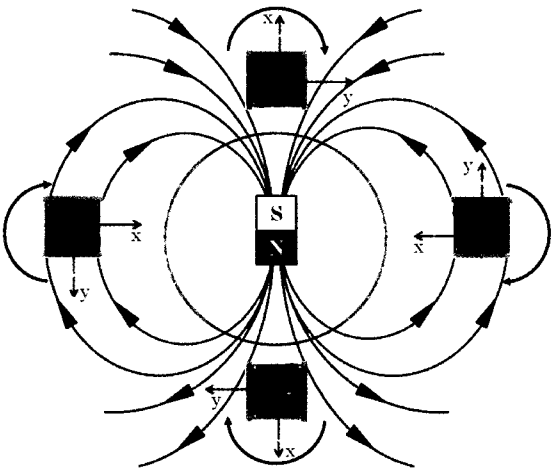


Figure 26: Retrograde Orbital Alignment

The end result is that the camera points at the Earth twice per orbit instead of continuously as desired, but at all four bias points the correct axis is aligned.

The other undesirable stable condition is a prograde orbit where the biased axes are 180° from their desired positions. This alignment is being called the prograde reversed condition and is shown in Figure 27. It is possible due to the nature of the cross-product law shown in Equation (7). When the biased axis is opposite the local field lines,

very little torque is imposed on the spacecraft. If the spacecraft is maintaining the correct body rates, this has the potential to be a stable condition.

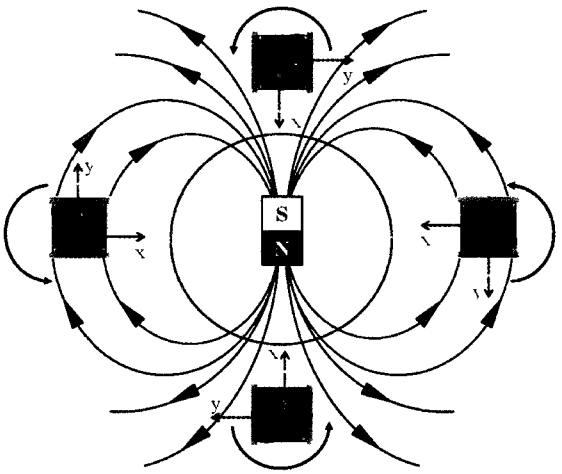


Figure 27: Prograde Reversed Condition

This alignment is highly undesirable since the camera is never pointed at the Earth.

4.2.6 Alignment Modes

The ACDS simulation uses three modes of operation. The first is Mode I which is also known as the detumble phase. This mode operates from ACDS initialization until measured body rates are within 0.2% of commanded body rates. This equates to the spacecraft turning no more than 1° in each axis during the 10 second charging time. The spacecraft can be forced to reenter this mode on command from the ground station.

Mode 2 is the first stage of alignment. It is very similar to the LPMT biased alignment shown in Figure 16, but in this case no bias is made at the South Pole.

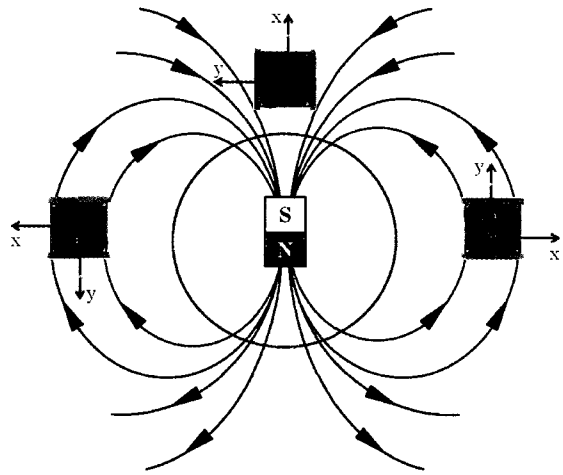


Figure 28: Mode 2 Bias Alignment

As can be seen in Figure 28, the y-axis vernier torquers are biased at the equator while the x-axis vernier torquers are biased at the North Pole only. It was shown in simulation that ARC tended to enter a retrograde orbit when the bias was executed at the South Pole. To prevent this, the South Pole bias was suspended. This is due to the dependence of longitude seen at the South Pole in Figure 19. The approach also negates any influence of the South Atlantic Anomaly.

After completing 10 orbits in Mode 2, ARC will transition into Mode 3. In this mode, ARC will verify the Earth's magnetic flux density vector direction is correct for the desired alignment. At the North Pole, the direction of the x-axis body component of the geomagnetic field is tested. A positive value indicates a prograde reversed condition. The x-axis is also sampled near 45° north latitude. A positive value indicates retrograde exists.

An incorrect direction triggers corrective logic below. If retrograde is detected, ARC will bias the large x-axis torquers at the North Pole one time and then reenter Mode 2. If prograde reverse is detected, ARC will bias the large x-axis torquers immediately for one charge cycle and then reenter Mode 2. This has been shown to be over 80% effective at breaking out of those undesirable conditions on the first correction attempt. In the other 20%, it takes multiple attempts to correct the condition. Results can be seen in Subsection 4.4.3 and 4.4.4.

Provided the proper alignment is confirmed, Mode 3 will further reduce the bias alignment logic to the North Pole only. Additionally, the trim torquers will be used in a manner consistent with Mode 1 when outside the polar bias window. This configuration has been shown to use the least power during long-term operations.

4.2.7 Alignment Offsets Defined

It was found that centering the bias alignment on the point of horizontal or vertical field lines tended to promote the spacecraft entering a retrograde orbit. Consider the following diagram.

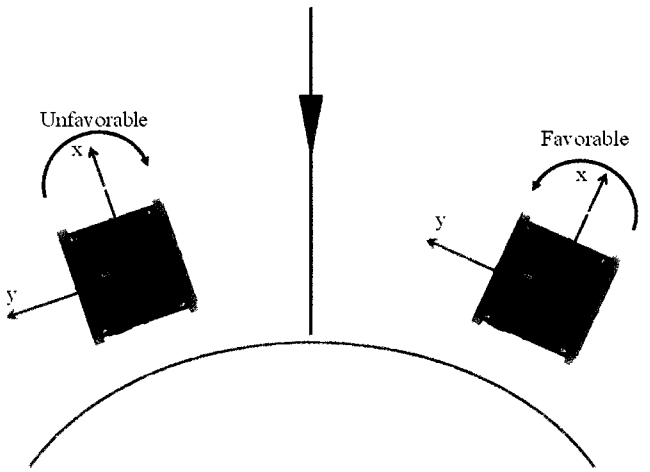


Figure 29: Biased Alignment Offset

As the spacecraft approaches the vertical field lines, a rotation about the z-axis consistent with a prograde orbit is created. However, as the spacecraft passes the vertical field lines, a backspin is created. Since the ACDS prevents a negative rotation about the z-axis, the satellite tends to stabilize in retrograde. The effect is compounded by ARC leaving the bias zone and thus receiving no further alignment corrections. The same principle applies to the horizontal field lines as well.

To prevent retrograde from occurring, the bias is begun at a point prior to the vertical or horizontal field lines. This significantly reduces the number of times the satellite settles into retrograde. Ideally the bias window ends at the point of the pure vertical or horizontal field lines. This tends to result in the most precise alignment.

4.2.8 Alignment Offset Results

A polar window for the LPMT bias of 10° of latitude was found to be the most effective. This window was originally centered on the latitude of the inclination angle. For a polar orbit, it would be centered on 90° north latitude. For a 64° orbit, it was centered on 64° north latitude. It was discovered that this tended to promote the undesired stable attitudes. To prevent this, the window was offset 5° early. Although the undesired retrograde and prograde reversed attitudes still occur, it is with a greatly reduced frequency.

The same rules apply to the equatorial regions which are only used during alignment Mode 2. The window is 15° with an offset of 10° north. This is because the true horizontal field lines occur slightly north of the Earth's equator.

4.3 Detumble Results

Utilizing the large torquers, the ACDS is capable of arresting tip-off rates up to 0.1 radians per second (1 revolution per minute) in all three axes. The limitation in slowing higher tip-off rates is the 10 second charging time. Guerrant 2005 [17] came to a similar conclusion while working with open coil torquers on Cal Poly's CP2 (*f*). In his case, the question was how long to energize the coils for a $\dot{\mathbf{B}}$ controller. Since ARC's charge cycle time is fixed (equivalent to the amount of time an open coil is energized), there is an upper limit to the tip-off rates the system is able to actively arrest. An estimate of these limits is shown in the following figure.

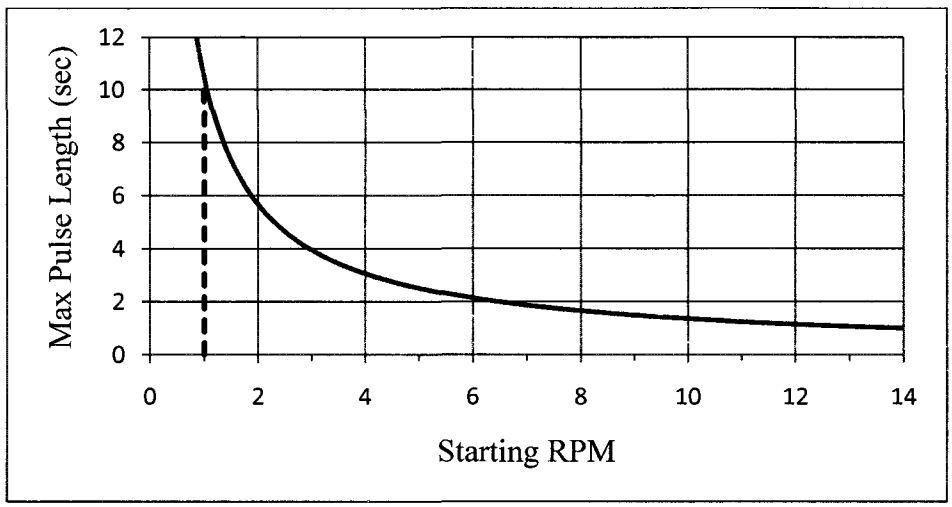


Figure 30: Tip-off Rate Performance

The **blue** line indicates ARC's maximum planned rate of 1 RPM, which is just within the capability of a 10 second charging time. In order to accept higher tip-off rates, the charging time needs to be reduced or an alternate method such as the one discussed in Subsection 4.3.8 must be used.

4.3.1 ALNICO1 Cores

The ALNICO1 cores have the best overall performance. The following plot shows maximum tip-off rates being arrested in under 2 polar orbits (1 orbit at 600 km altitude takes approx. 5793 seconds).

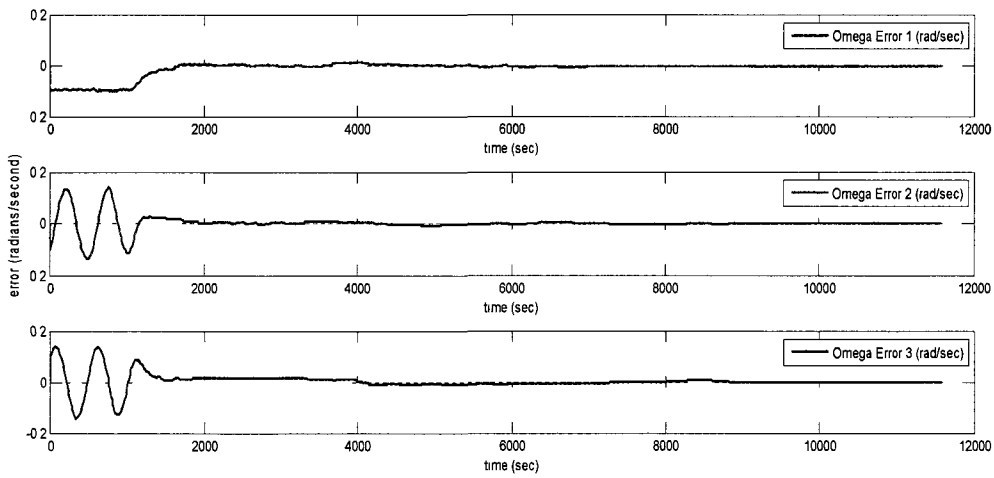


Figure 31: ALNICO1 Body Rate Error (600 km / 90° inclination)

The plot shows the error between the commanded body rates and the measured body rates. By the end of the second orbit, the error is less than 0.1% which is more than enough to activate Mode 2.

In the above 2 orbit run, which was sufficient to complete the detumble phase, 466 large torquer pole flips were required. As a reminder, each large torquer pole flip dissipated 0.014 joules of energy. In order to determine the average power, it is necessary to examine the timing of the torque firings.

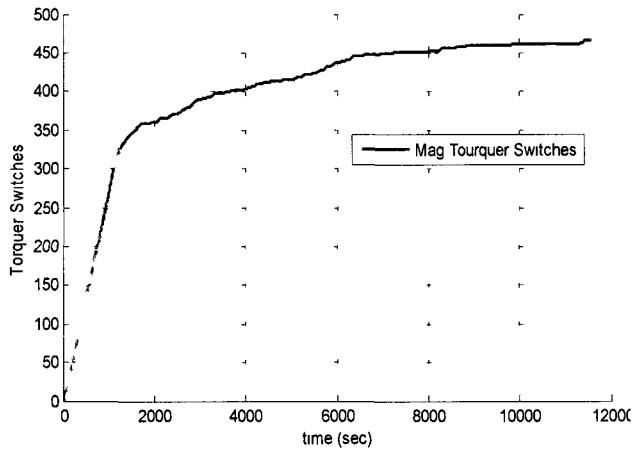


Figure 32: ALNICO1 Torquer Switches (600 km / 90° inclination)

The average power dissipated over any time interval is the energy dissipated over that interval divided by the elapsed time. In Figure 32, the maximum power used is in the first 2,000 seconds (highest slope of the curve), which was determined to be 3.2 milliwatts. The average power dissipated over the entire 2 orbits was 0.56 milliwatts. The following figure shows in which axes the flips were performed.

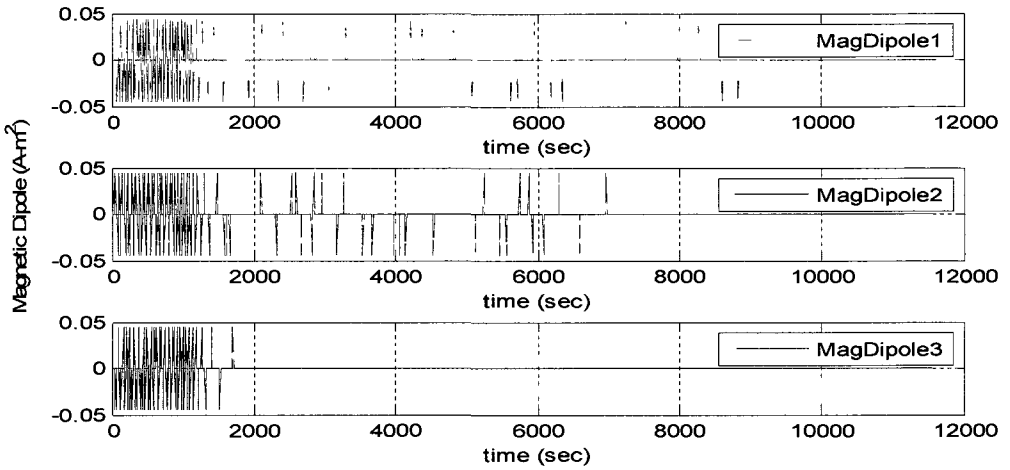


Figure 33: ALNICO1 Dipole Moments (600 km / 90° inclination)

As in Figure 32, the greatest number of switches in Figure 33 occurs in the first 2,000 seconds where the bulk of the undesired angular momentum is corrected.

4.3.2 ALNICO5 Cores

Although UAF has acquired a source for ALNICO1 cores, ALNICO5 cores are still a workable alternative. This is a benefit since ALNICO5 is much more easily obtained. However, as discussed above, the maximum tip-off rate is a function of the charging cycle time. Therefore, although ALNICO5 cores provide a high magnetic dipole moment, they will not be able to arrest a higher tip-off rate.

ALNICO5 cores are able to slow the maximum tip-off rate more quickly but are unable to reduce body rate error to the same degree as the ALNICO1 cores. This is due to the larger magnetic dipole moment term in Equation (20) coupled with the nature of the quantizer shown in Figure 14. The following plot shows maximum tip-off rates being arrested in under 1 polar orbit.

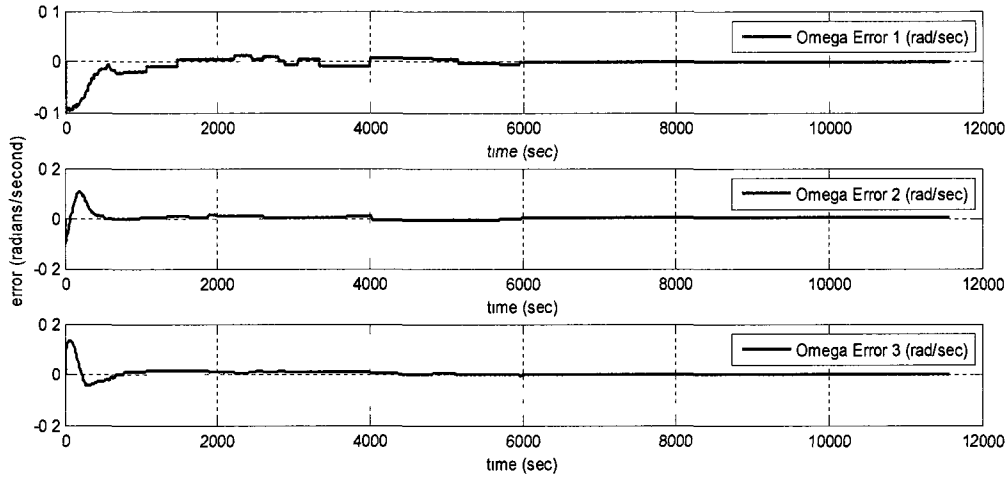


Figure 34: ALNICO5 Body Rate Error (600 km / 90° inclination)

After 2 orbits, the measured body rate error was 0.25%, which is just above the threshold used to change alignment modes. The vernier torquers are capable of achieving alignment at this higher rate, so the threshold would simply be raised.

In the above 2 orbit run, 146 large torquer switches were executed. Most of these occur in the first 700 seconds as can be seen in the following figure.

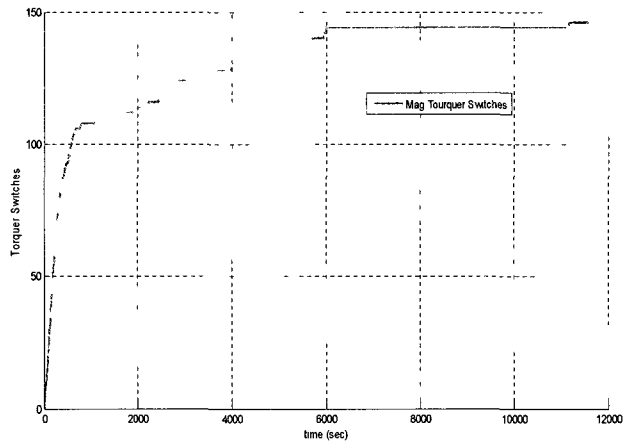


Figure 35: ALNICO5 Torquer Switches (600 km / 90° inclination)

This equates to a maximum power use of 2.13 milliwatts, which is approximately a fourth the amount required for ALNICO1. The average power for the entire run was 0.18 milliwatts which is about half that of ALNICO1; however, the end state body rates are higher and will require more vernier torquer flips to correct.

4.3.3 Permalloy80 Cores

The Permalloy80 (vernier) cores are by design $^{1}/_{200}{}^{th}$ the magnetic dipole moment of the large torquers. It is therefore expected that they will require a much longer time to complete a maximum tip-off rate detumble. Simulations have shown that a detumble at maximum rates would require approximately 350 orbits to complete. Since this process would take over 3 weeks, it is reserved as a contingency only.

4.3.4 Other Inclination Angles

The detumble phase will be completely successful regardless of orbital inclination angle. In a polar orbit, the spacecraft makes full use of the higher geomagnetic field strength at the poles to damp tip-off rate the most efficiently. At lower inclination angles, the maximum field strength is lower, so the detumble takes longer to complete. The following figure shows the same tip-off rates used in Figure 31, but the orbital inclination angle has been reduced to 64°, which is the lowest angle acceptable for ARC's launch.

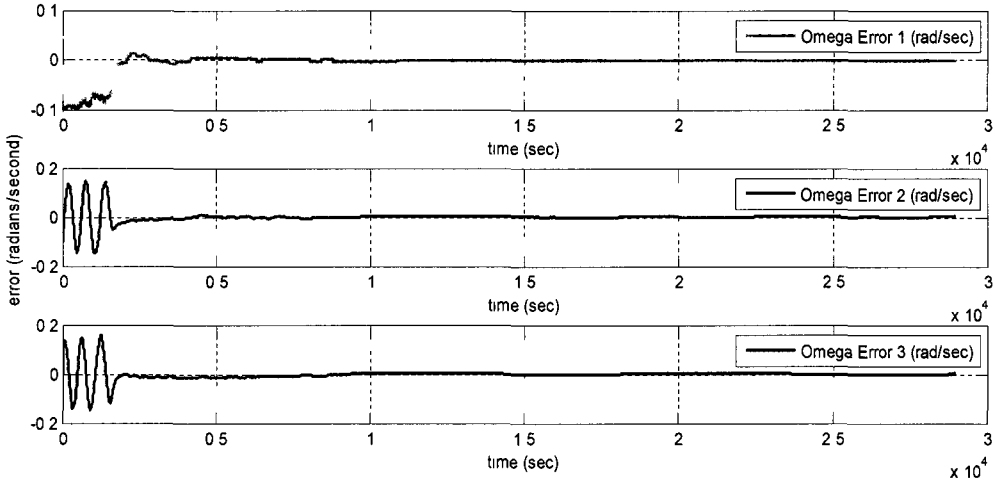


Figure 36: ALNICO1 Body Rate Error (600 km / 64° inclination)

In this case the detumble time has increased minimally and is still completed in under 2 orbits. The end state errors are within 0.02% of the commanded values. However, the lower magnetic field strengths had to be compensated for with slightly higher power expenditures. The following figure shows the torquer switches.

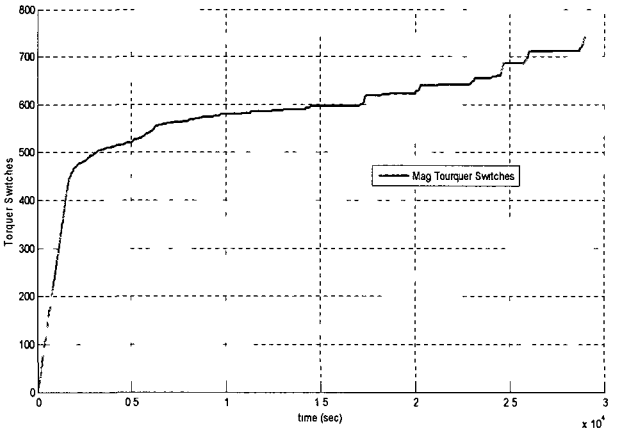


Figure 37: ALNICO1 Torquer Switches (600 km / 64° inclination)

The maximum average power expended during the run occurs in the first 2,000 second and is the same as the polar orbit run at 3.2 milliwatts. The average power over 2 orbits is 0.90 milliwatts compared to the polar orbit's 0.56 milliwatts.

When the inclination angle is further reduced to 45° the detumble time increases significantly. The following figure shows settling time increasing to just under 10 orbits.

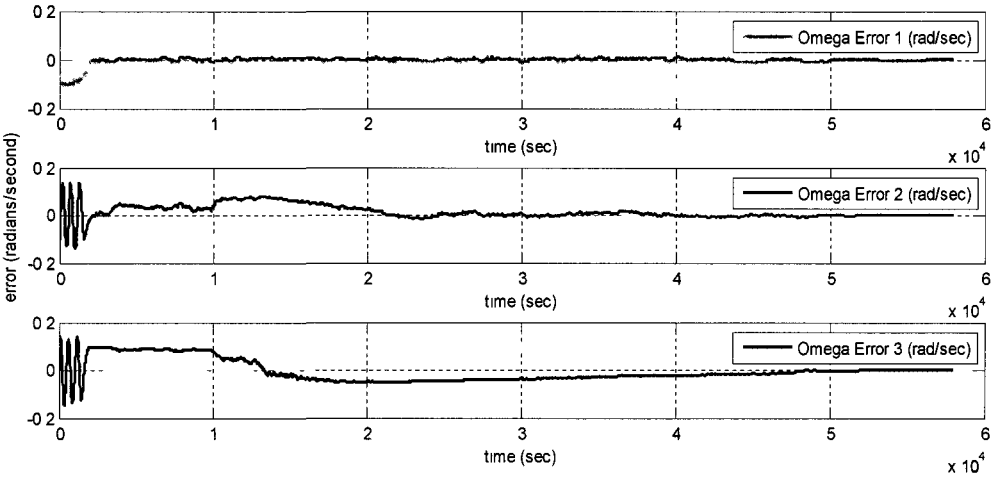


Figure 38: ALNICO1 Body Rate Error (600 km / 45° inclination)

The end state body rate error after the 10 orbits is within 0.02% of commanded values. The maximum average power is just slightly higher than a polar orbit at 3.4 milliwatts and again occurs within the first 2,000 seconds. The average power expended over the

entire detumble has increase to 0.70 milliwatts. The switches can be seen in the figure below.

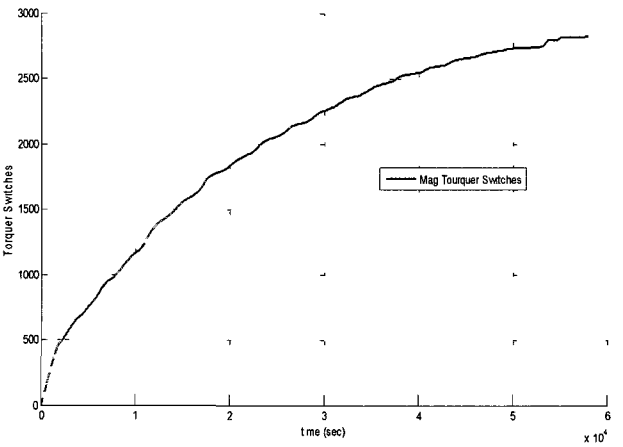


Figure 39: Torquer Switches (600 km / 45° inclination)

The longest settling time occurs for an equatorial orbit, as shown in Figure 40. In this case, the detumble phase completes in just under 53 orbits.

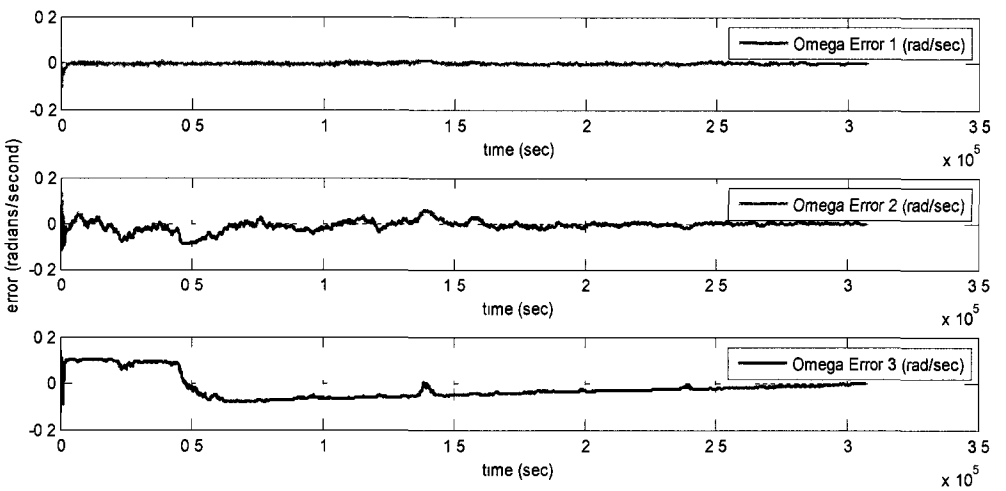


Figure 40: ALNICO1 Body Rate Error (600 km / 0° inclination)

The maximum average power was 3.68 milliwatts during the first 900 seconds, while the average power for the detumble was 0.36 milliwatts.

4.3.5 Variants in Core Sizing

There is flexibility in the selection of the large torquer's magnetic dipole moment. As was shown in Subsection 4.3.3 above, the moment can be reduced down to the value of a vernier torquer and the detumble can still be accomplished, although the time is greatly increased.

As the magnetic dipole moment is increased, a limit is reached in the amount the ACDS is able to minimize body rate error. This requires Mode 2 to be initiated earlier and maintained longer to achieve proper alignment. Large core sizes up to 0.03 A-m^2 require no software changes. Core sizes up to 0.05 A-m^2 (over 2 times the design size) have been effective at reducing the body rate error to 0.5% in just 1 orbit. Cores above this size would need to be investigated prior to use in order to properly set the Mode 2 threshold.

4.3.6 Unbalanced Cores

In order to supply no torque to the spacecraft, both cores in a single axis must balance each other when their poles are opposite. If the poles are unbalanced, a small torque is applied to the spacecraft during those times when no torque is commanded. It was assumed that the balancing of the large LPMT cores would be the limiting factor. The detumble was conducted repeatedly with the balanced condition being a percentage of the dipole of a single large LPMT, instead of the usual 0$\boldsymbol{m}$ magnetic dipole moment.

It was determined that unbalanced large cores had very little impact on the detumble phase. The following run has the 0$\boldsymbol{m}$ condition set to 50% of the large magnetic dipole moment or 0.011 A-m^2.

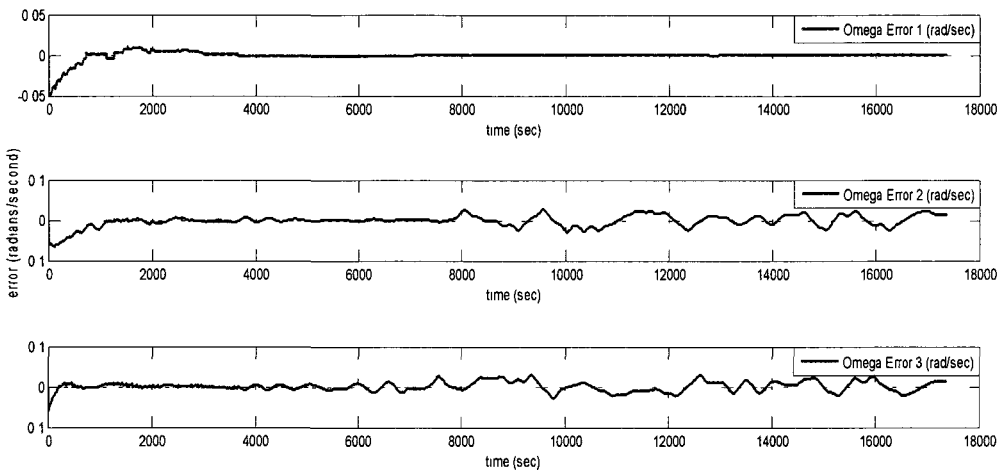


Figure 41: ALNICO1 Unbalanced Torquers Body Rate Error (600 km / 0° inclination)

As can be seen, the detumble still completed within 2 orbits (left side of graph), but due to the imbalance, the rates are not held and start to grow once the alignment phase begins (right side of graph).

4.3.7 Faster Charging Time

In the detumble phase, faster charging times allow the arrestment of higher tip-off rates and can achieve lower body rate error prior to initiating Mode 2. However, referencing Figure 30, it can be seen that cutting the charge time in half to 5 seconds will only increase the maximum tip-off rate to around 2 RPM. Since the expected tip-off rate is 1-2 degrees per second, and the true capability of ARC's power system is unknown, decreasing the charge time for the entire detumble phase is not warranted at this time. However, if the power system is shown to have the capability, reducing the charging time to 5 seconds until rotation rates are below 1 RPM would be beneficial. The following figure demonstrates the arrestment of a 2.1 RPM tip-off rate with a 5 second charge cycle.

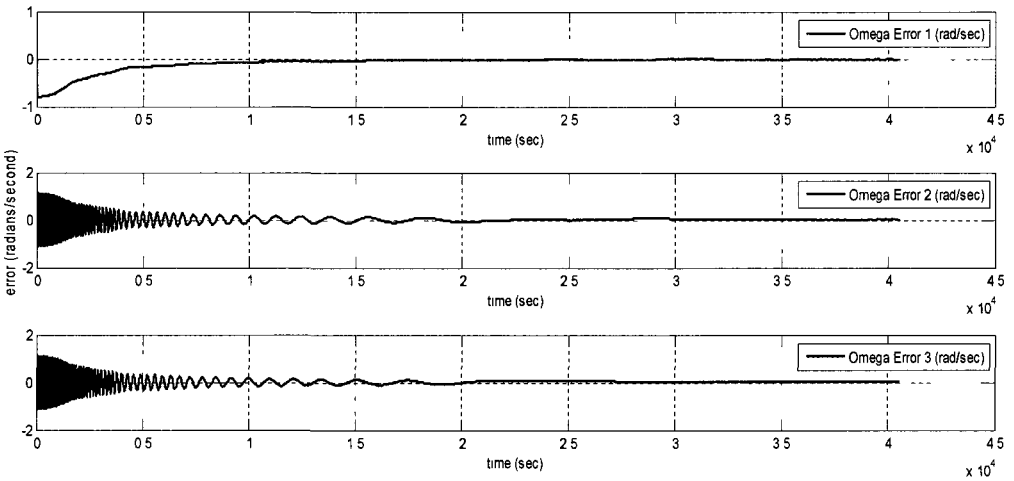


Figure 42: ALNICO1 Body Rate Error (600 km / 90° inclination/1 sec charge time)

4.3.8 Higher Tip-off Body Rates

In the event of extremely high tip-off rates, the larger torquers could be used as hysteresis rods. As the solenoids cut through the Earth's magnetic field lines, a current is generated. Dissipating this current through a resistor would have the long term effect of reducing angular momentum. Once the rotation is below 1 RPM, the ACDS could begin actively arresting the remaining momentum. This could be increased to 2.5 RPM provided the power system is capable of support a 5 second charge / discharge cycle as discussed in the last subsection.

4.4 Alignment Results

The ACDS is capable of aligning ARC to within ±5° of the planned attitude. The time required for alignment depends on the orbital inclination. Low angles tend to produce more retrograde or prograde reversed orbits, and it generally takes longer to correct these conditions.

4.4.1 Polar Orbit

Polar orbits have shown the best performance from the ACDS. The following figure depicts 20 orbits with the satellite starting at maximum tip-off rates.

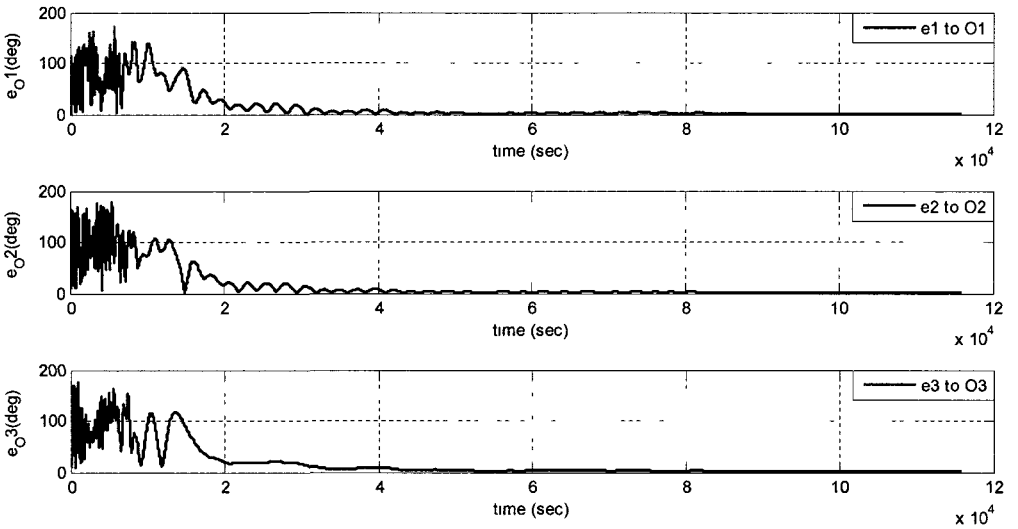


Figure 43: ALNICO1 Alignment (600 km / 90° inclination)

By the 11th orbit, ARC has entered Mode 3, which provides the tightest long-term alignment. The next plot is zoomed in to the last 4 orbits and shows pointing accuracy.

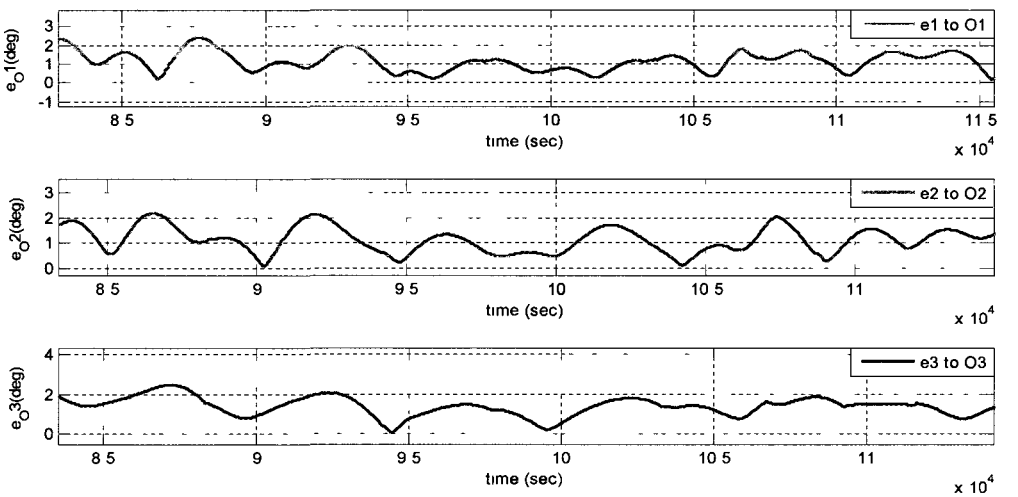


Figure 44: ALNICO1 Pointing Accuracy (600 km / 90° inclination)

ARC expends approximately 0.028 milliwatts to maintain its attitude within 2° of the desired alignment. This is better than the ±5° required for successful mission completion.

4.4.2 Minimum Inclination Angle

At the minimum inclination angle of 64°, the ACDS will still achieve mission requirements. The following figure depicts the last 4 orbits of a 20 orbit run.

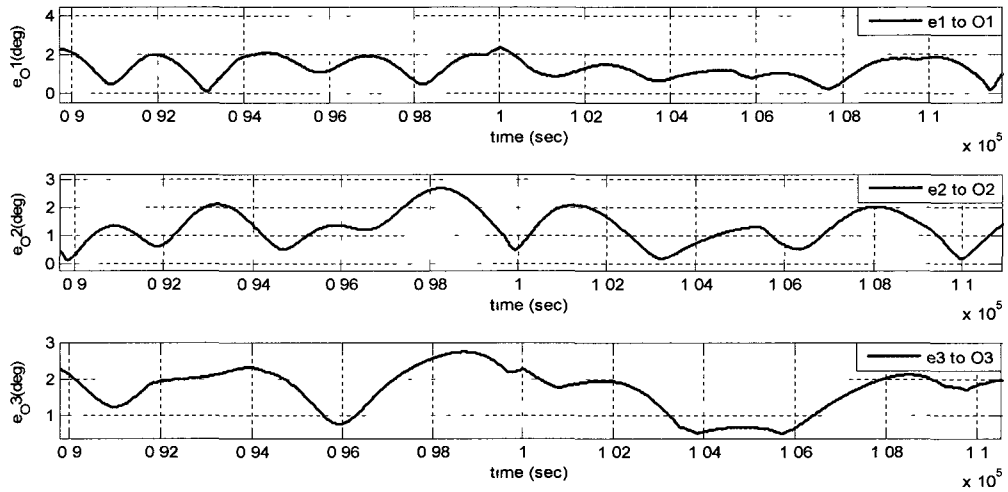


Figure 45: ALNICO1 Pointing Accuracy (600 km / 64° inclination)

The power required to maintain the alignment will increase, since the true vertical and horizontal magnetic field lines are no longer conveniently aligned with the orbital axes. In this case, ARC will expend approximately 0.078 milliwatts.

4.4.3 Retrograde Correction

In the event retrograde is detected, ARC will wait until the satellite is at the North Pole and then bias the large torquers in the positive x-axis. The satellite then reenters alignment Mode 2. This procedure has shown better than 80% effectiveness at correcting retrograde on the first attempt. The following figure depicts 25 orbits and shows a typical retrograde correction.

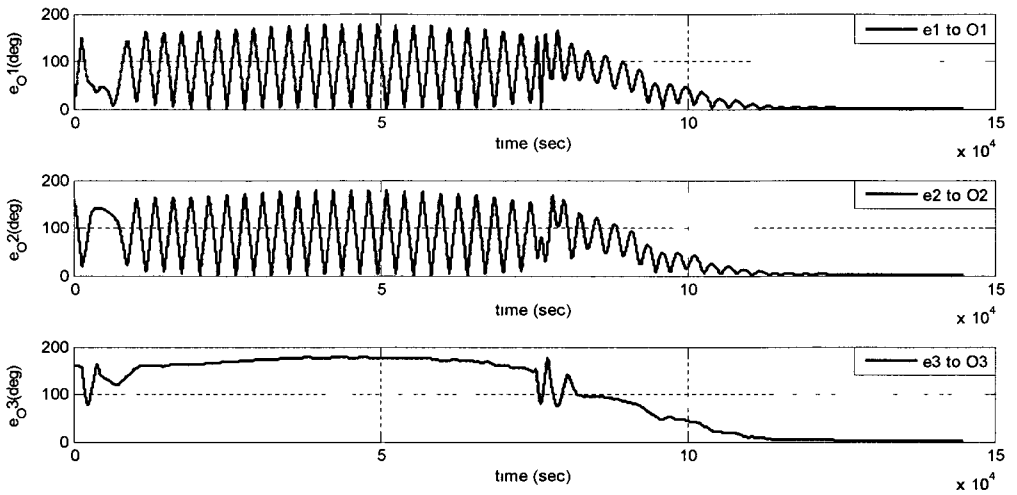


Figure 46: Retrograde Correction (600 km / 90° inclination)

ARC must remain in Mode 2 for at least 10 orbits. Otherwise it is difficult to determine if unexpected magnetic flux density vector directions are due to retrograde or due to the spacecraft not having settled yet. In this case, as the ACDS enters Mode 3, a retrograde condition is confirmed. The single larger torquer bias is sufficient to break the condition and the spacecraft settles in the correct attitude.

4.4.4 Prograde Reversed Condition

In the event prograde reverse is detected, ARC immediately biases the large x-axis torquers in the positive direction. The satellite then reenters alignment Mode 2. This procedure has also been better than 80% effective at correcting prograde reverse on the first attempt. In the other 20%, multiple attempts are required to correct the condition. The following figure depicts 25 orbits and shows a typical prograde reversed correction.

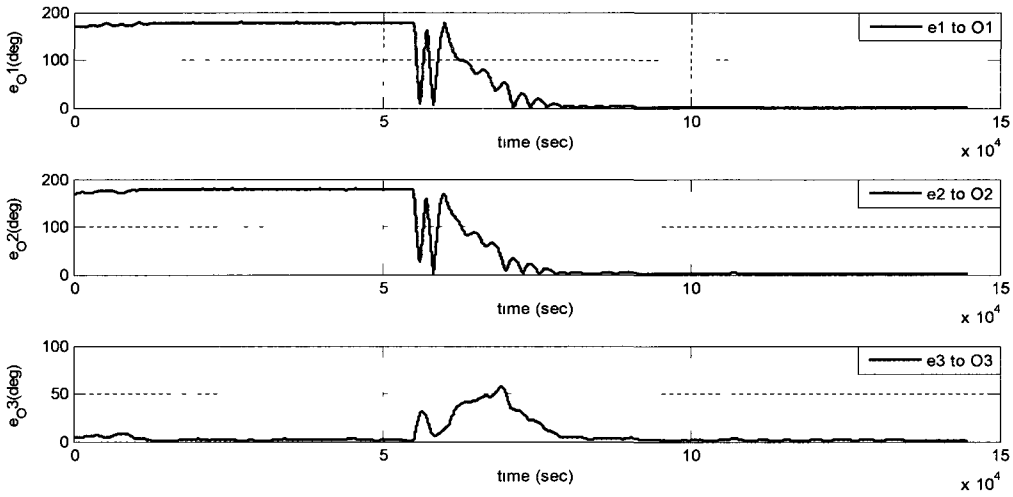
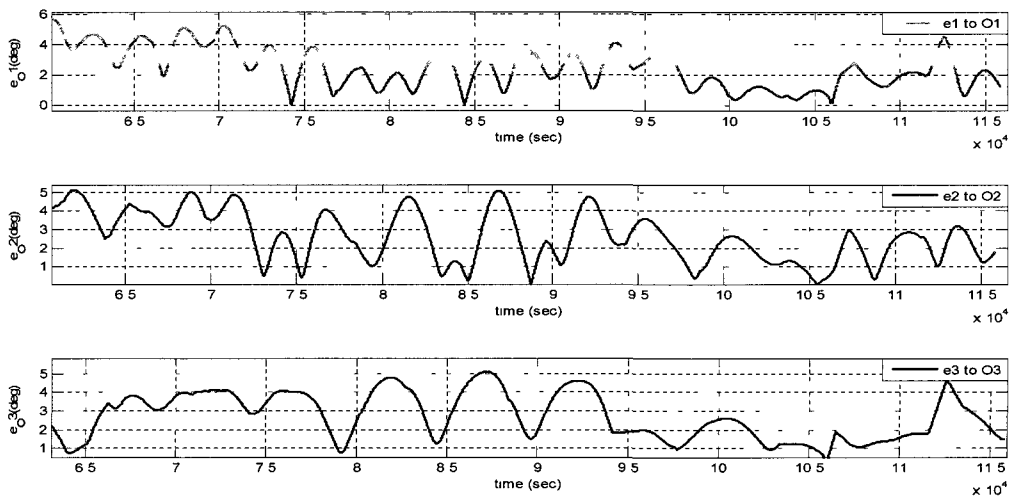


Figure 47: Prograde Reversed Correction (600 km / 90° inclination)

4.4.5 Variants in Vernier Core Sizing

There is flexibility in the sizing of the vernier cores. As a general rule, when the magnetic dipole moment increases, the size of the oscillations about the desired attitude increases. In the following case, the vernier dipole has been doubled, but the performance is degraded only slightly.



Double core size

Figure 48: Attitude Error for 0.00022 A-m^2 Vernier Torquer

The maximum allowable size is 5 times the baseline or 0.0055 A-m^2. At this magnetic dipole moment, the pointing requirement of ±5° is just barely being achieved more than 90% of the time. This can be seen in the following figure.

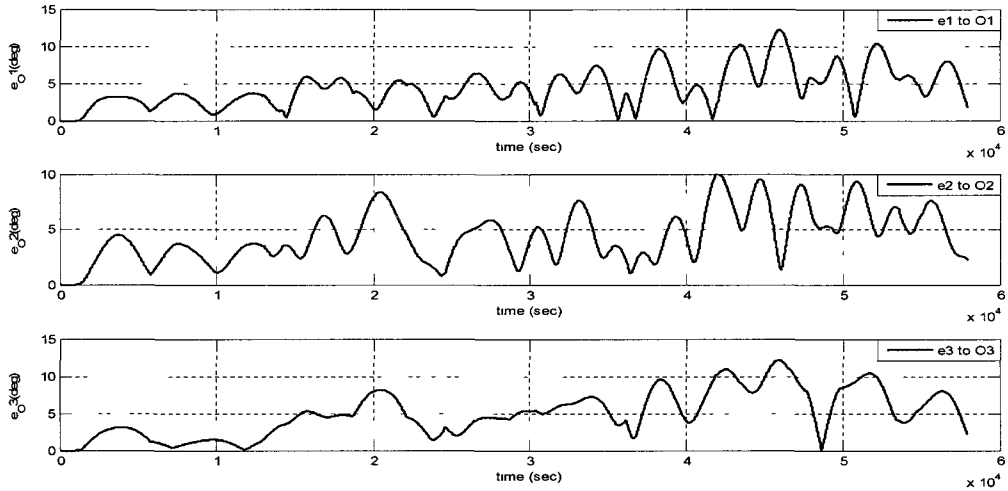


Figure 49: Attitude Error for 0.00055 A-m^2 Vernier Torquer (10 Sec Cycle)

4.4.6 Faster Charging Times

Faster charging times showed almost no improvement for baseline vernier torquers. The biggest change was noticed when using vernier torquers 5 times larger than baseline or 0.0055 A-m^2. Decreasing the cycle time to 1 second has the following result.

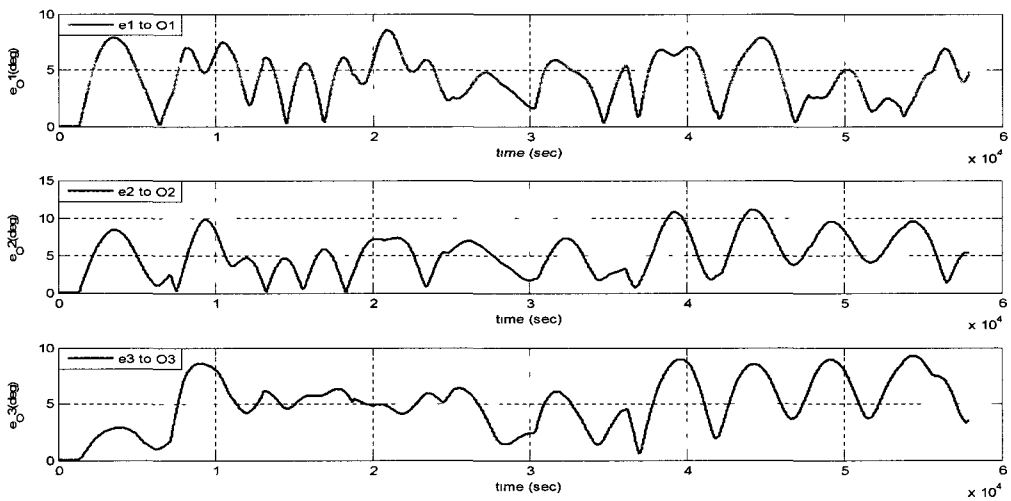


Figure 50: Attitude Error for 0.00055 A-m^2 Vernier Torquer (1 Sec Cycle)

Even this change is relatively insignificant, so for long term operations the 10 second cycle is recommended.

4.4.7 Unbalanced Cores

It is assumed that no torque is applied to the satellite when the torquer twins have opposite polarity. In reality this may be difficult to accomplish. The following subsection discusses to what degree the LPMT cores must be matched in order to maintain the proper attitude.

It is assumed that the limiting factor will be the balancing of the large LPMTs. The ACDS simulation software was run repeatedly with the imbalance being set as a percentage of the magnetic dipole moment of a large torquer core. The system is stable when the imbalance is 0.04 percent or less. The following figure shows an aligned spacecraft completing 15 orbits with the maximum imbalance.

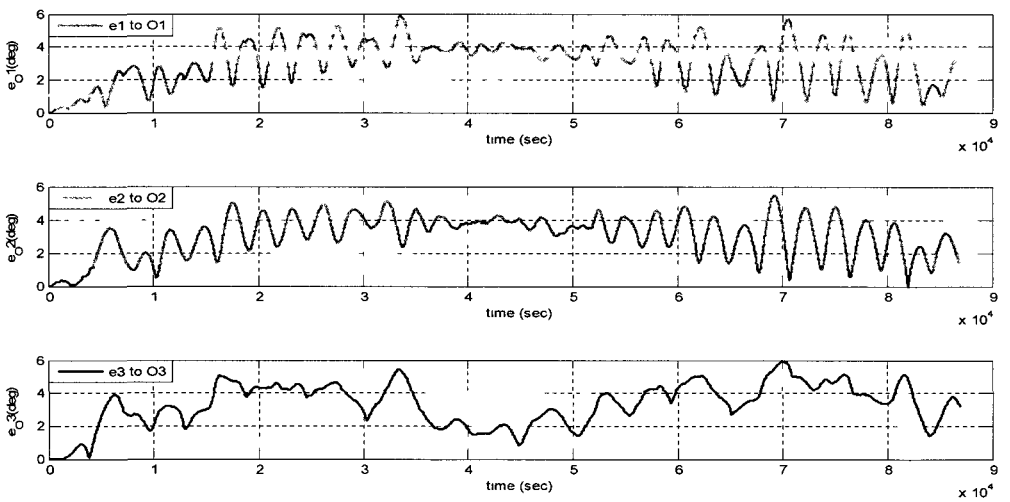


Figure 51: Attitude Error for 0.04% Large Torquer Imbalance

As the imbalance is increased to 0.05%, the stability of the system is exceeded and the spacecraft's alignment is eventually lost. This can be seen in the following figure where attitude is lost on the right side of the graph.

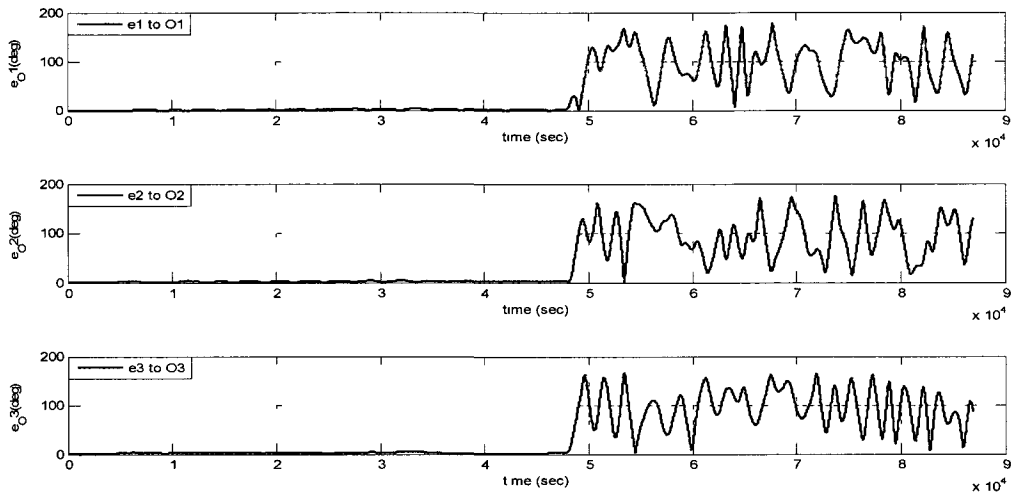


Figure 52: Attitude Error for 0.05% Large Torquer Imbalance

4.5 ACDS – Ground Station Interaction

The ACDS should be programmed to accept various commands from the ground station in the event of unforeseen problems while the satellite is in orbit. This may make it possible to salvage part of the mission by limiting or controlling parts of the ACDS that would otherwise run automatically.

4.5.1 Force Mode 1 (Upgrade)

This command would force the ACDS to return to Mode 1 from any other mode, and would allow the ACDS to automatically upgrade to Mode 2 when it determines that measured body rates are within 1% of the commanded rates. In the event that ARC is in alignment mode but is still tumbling, this command can force detumble logic to restart.

4.5.2 Force Mode 1 (No Upgrade)

This command forces ARC to return to Mode 1 from any other mode and stay in Mode 1 until directed to leave it by the ground station. In the event ARC is entering Mode 2 when it should not be, this command can force it to stay in Mode 1 for a longer time.

4.5.3 Force Mode 1 Vernier (No Upgrade)

This command forces ARC into Mode 1 from any other mode and requires it to stay there until directed to another mode by the ground station. The ACDS is also instructed to neutralize the large torquers and attempt the detumble with the vernier torquers. This may be a way to recover the mission if there is an electrical problem in the large torquers only.

4.5.4 Force Mode 2 (Upgrade)

This command forces ARC into Mode 2 from any other mode. It will stay in Mode 2 for 10 orbits and then go into Mode 3. This command would be useful if the ACDS was not detecting the end of the detumble phase.

4.5.5 Force Mode 2 (No Upgrade)

This command forces ARC into Mode 2 from any other mode and requires it to stay there until directed to another mode by the ground station. This command would be useful if the 10 orbit Mode 2 assumption was not working for some reason.

4.5.6 Force Mode 3

This command forces ARC into Mode 3 from any other mode and requires it to stay there until directed into another mode by the ground station. This command is the way to get out of the mode described above.

4.5.7 Force Mode 3 Large Torquers

This command forces the ACDS into Mode 3 from any other mode and requires it to stay there until directed to another mode by the ground station. The vernier torquers are neutralized. All Mode 3 operations are performed with the large torquers. This is a contingency mode that may work in the presence of very high solar radiation pressure or increased atmospheric density.

4.5.8 ACDS Off

This command stops the ACDS from operating. This may be required in later life if the power system is no longer able to support the entire spacecraft. The ACDS neutralizes the dipoles prior to shutdown.

4.5.9 Permanent Magnet Hysteresis Mode

This command sets the large torquers in the y-axis to $+2m$ and shorts all of the other torquers. The satellite would now have the same control mechanism described in Subsection 1.5.4 above. In the event an unknown disturbance imparts a body rate on ARC in excess of the LPMT detumble capability, this mode may in time slow the rates enough to reengage Mode 1. This is also another possible end of life mode that requires no power but will maintain a predefined alignment.

4.5.10 Individual Commands

The satellite should accept commands from the ground station to flip the dipoles of individual torquers. It should also accept commands to lockout sets of torquers. In the event that the electrical connection of 1 torquer fails, it should be possible to manually flip the pole of the torquer's twin. Once they are neutralized there should be a command to stop either of them from being flipped in any of the automatic modes.

4.5.11 Charging Cycle Time

If the power system is capable of supporting faster charging times than the assumed 10 second cycle, it should be possible to change the window in orbit. This could be used in the event of unusually high tip-off rates. It could also be used later in the spacecraft's life if the power system is no longer able to maintain the 10 second cycle but is able to maintain a longer one.

4.5.12 Position Determination Changes

New alignment windows and offsets need to be programmable while the spacecraft is in orbit. This is an experimental spacecraft and in flight adjustments will almost certainly be required. If the automatic system is not functioning as desired it may be possible to salvage the mission by changing the parameters.

The ground station needs the ability to reset the automatic logic or force the spacecraft into timing mode. The orbital period is required by the timing mode and needs to be adjustable. If the biases are not being performed at the correct locations, the timing mode could be used to correct the problem and regain mission attitude.

4.5.13 Data Collection

The spacecraft must record orbital data that is returned to the ground station on command. These parameters would be required to troubleshoot on orbit attitude problems. It is desired that the ACDS records 1 set of data every 10 seconds until 500 points are collected or the memory is full. The required parameters are: magnetometer readings, calculated body rates, commanded torquer settings (all 12 of them). It is desired to have the power output for each face of the CubeSat's solar panels. There should be a command from the ground station that erases the memory and starts the collection over.

Chapter 5 Future Work

5.1 Testing

As the components of the ACDS are brought together and integrated into the completed prototype, it will be important to ensure they are properly tested. The individual systems must be confirmed to operate as designed not only in a stand-alone setting but also after they are brought together with the other spacecraft components.

5.2 LPMT Testing

The following subsection discusses the progression of testing required for the LPMTs of both sizes. Portions of the large torquer design have already been verified and are therefore omitted from this discussion.

5.2.1 Vernier Core Testing

The magnetic dipole moment of the vernier torquers will need to be determined after they are constructed. Use the procedure in Subsection 2.4.6 above adjusted for the mass of the vernier torquer. The procedure should be repeated for 0.1, 0.2, and 0.5 Gauss. Verify that the large torquer capacitors are able to drive the vernier core to saturation.

5.2.2 ACDS Board Testing

It is possible that eddy current losses will prevent the 500 μF capacitors from driving the cores to saturation. Once integrated into the CubeSat structure, the cores of large and vernier torquers should be flipped by a single discharge pulse and then removed from the solenoid for testing. Verify the cores have been saturated. If not, the size of the capacitors will need to be increased.

Determine the length of the charge / discharge cycle the power system is able to provide. Verify the cycle time both in daylight and during simulated night. If possible determine the impact of the battery heaters while the spacecraft is in the dark and in the cold.

During vacuum testing, determine if the ACDS solenoids build up any heat in the spacecraft from repeated firings. If heat buildup occurs determine amount of time required to cool them at zero atmosphere.

5.2.3 Magnetic Balancing

Utilize the Helmholtz cage to verify that LPMT twins have the same magnetic dipole moment so they do not torque when their poles are opposite. Large torquers must be balanced to 0.4% of each other. Vernier torquers must be within 40% of each other.

Verify with a force table in the Helmholtz cage that the spacecraft does not turn in the presence of a magnetic field when the LPMTs are all neutralized. Repeat for all three axes. Run test for all modes of operation. Determine if any other subsystem places a magnetic torque on the spacecraft. The spacecraft should turn less than 5° for a 90 minute period, with all torquers in neutral configuration, in order for Mode 3 to be successful.

5.2.4 ACDS Software Testing

Using a force table in the Helmholtz cage, verify the ACDS flips the correct dipole to correct a spin of around 0.1 RPM (make sure it is not making the spin worse). Verify for all three axes. Simulate the magnetic field for the North Pole and equator. Verify the bias logic is activating the correct axis with the correct dipole polarity. Verify the unbiased axes correctly attempt to damp oscillations created by the biased axis (make sure it is not making the condition worse).

5.3 Magnetometer Testing

The magnetometer is being used in unique ways for this mission. Proper verification of its function is critical.

5.3.1 Rate Determination System Testing

The magnetometer should be spun in the Helmholtz cage at known rates up to 10 RPM. The rate determination system will calculate the body rate in one axis and have its accuracy verified. This should be repeated for all three axes.

The ACDS should flip three large LPMT cores in the charge window while the rates are calculated towards the end of the window. Verify accurate rates are calculated. Consider this information prior to adjusting the charge window timing. Determine if the presence of the LPMT pole flips needs to be calibrated out of the rate determination system. Since the rate determination systems bases its measurement on changes of the flux field, the presence of the LPMT should be zero impact. The pole flips occur outside of the measurement window and will not be seen by the rate determination system.

5.3.2 Position Determination System

The activity of the LPMTs will need to be calibrated out of the magnetometer in order to get an accurate reading of the geomagnetic field's magnitude. This magnitude must be accurate so that the position determination system is able to locate the North and South Poles. The electromagnetic field of the other electronics onboard ARC must also be calibrated out. If this cannot be accomplished, the position determination system will be unable to take a true reading of the Earth's geomagnetic field. This will make it impossible to determine spacecraft position from magnetometer readings alone and therefore LPMT bias must be done off of timing alone.

If calibration is successful, set Helmholtz cage to simulate the magnetic field over several orbits. Determine position determination system's ability to track orbital rate and relative latitude during various modes of operation. ARC must be able to determine the North Pole position to within the equivalent of 5^{o} of latitude.

5.4 Next Prototype

The next version of the ACDS should contain an onboard attitude determination system. This would give the next satellite a greater ability to make attitude control

adjustment decisions and should improve pointing accuracy. This system could make use of onboard magnetometer data compared with an IGRF-11 model and solar panel power / temperature readings used to determine the position of the Sun.

Chapter 6 Conclusions

The proposed ACDS is capable of providing detumble and attitude control to picosatellites at a fraction of the energy required by traditional systems. The most difficult developmental stage was the miniaturization of the LPMTs to the requirements of a 1U CubeSat. As such, it would be very easy to adapt the system to spacecraft with masses that are orders of magnitude greater than the satellite discussed herein. Although the sizes of the core material would be increased, the relative power requirements would still be a fraction of the traditional control methods for vehicles of that size.

The Alaska Research CubeSat will be the first flight use of low-power magnetic torquer technology. ARC's detumble phase will be the first use of the LPMT stabilization system the author helped to develop in 2003. As ARC begins the alignment phase, it will be the first known use of an algorithm that utilizes a dual axis magnetic bias to achieve three axis alignment. Additionally, it will be the first known use of the algorithm described in Chapter 3 to calculate spacecraft body rates directly from a magnetometer.

The final paragraph details the author's major contributions to the research contained within this paper. This included the selection of the physical sizes for the large and vernier torquers as well as their recommended magnetic dipole moments. It also involved the selection of the solenoid specifications as well as the size of their charging capacitors. The author determined the gain for the detumble phase and refined the detumble logic for CubeSat applications as well as creating the LPMT bias alignment logic, the first known use of a dual axis magnet bias alignment system. This was further developed into the stability conditions of the system as well as recommended alignment windows and offsets. Finally, the rate determination system is a unique, previously unseen, design suitable for LPMT control logic.

References

Literature Cited

[1] CubeSat Program, Official Website, http://www.cubesat.org/, (25 Apr 2011).

[2] M. E. Polites, "A Low-Power Magnetic Torquer for Satellite Attitude Control," Master's Degree Thesis, Department of Electrical Engineering, University of Alabama, Tuscaloosa, Alabama, 1971.

[3] J. Puig-Suari, Professor, Aerospace Engineering Department, California Polytechnic State University, San Luis Obispo, CA, e-mail conversations sent to the "Contact Us" section of the CubeSat Webpage, http://www.cubesat.org/, (04 Apr 2011).

[4] M. Mehrjardi and M. Mirshams, "Design and Manufacturing of a Research Magnetic Torquer Rod," *Contemporary Engineering Sciences*, Vol 3 No. 5, 2010, 227-236.

[5] C. Hall, "Attitude Determination: Chapter 4," 18 March 2003, http://www.dept.aoe.vt.edu/~cdhall/courses/aoe410/attde.pdf, (29 April 2011).

[6] M. D. Shuster, "The Quest for Better Attitudes," *The Journal of the Astronautical Sciences,* Vol. 54, Nos. 3 & 4, July-December 2006, pp. 657-683.

[7] T. Flatley, W. Morgenstern, A. Reth, and F. Bauer, "A B-Dot Controller for the RADARSAT Spacecraft," NASA Technical Paper No. 19970017192, Goddard Spaceflight Center, 1997.

[8] M. Polites, C. Quarles, D. Kadebek, J. Clements, and D. Mentch, "A Spacecraft Attitude Stabilization System with Low-Power Magnetic Torquers," *Proceedings of the*

Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 218(2): 99-109, 2004.

[9] M. Polites, "A Magnetic Torquing System for Emergency Stabilization and Backup Control for the LST," *Technical Memorandum NASA TM X-64912,* National Aeronautical and Space Administration , USA, 1975.

[10] International Association of Geomagnetism & Aeronomy, International Geomagnetic Reference Field 10th Gen., http://www.ngdc.noaa.gov/IAGA/vmod/igrf_old_models.html, (24 Apr 2011).

[11] Maurice Tivey FTP Site, Department Chair Geology & Geophysics, Senior Scientist, Woods Hole Oceanographic Institute, ftp://ftp.whoi.edu/pub/users/mtivey/mfiles, (24 Apr 2011).

[12] Data Analysis Center for Geomagnetism and Space Magnetism, World Data Center for Geomagnetism, Graduate School of Science, Kyoto University, Kyoto, Japan, http://wdc.kugi.kyoto-u.ac.jp/, (24 Apr 2011).

[13] ROSAT: The Roentgen Satellite, Goddard Spaceflight Center, National Aeronautics and Space Administration, http://heasarc.nasa.gov/docs/rosat/rosat.html, (24 Apr 2011).

[14] Australian Space Weather Agency, "Satellite Orbital Decay Calculations," Bureau of Meteorology, Australian Government, Sidney, Australia, http://www.ips.gov.au/Category/Educational/Space%20Weather/Space%20Weather%20Effects/SatelliteOrbitalDecayCalculations.pdf , (24 Apr 2011).

[15] M. Polites, J. Morris, T. Haskew, B. Dawson, C. Quarles, H. Dobbins, E. Amrock, and K. Witzberger, "Test Results of Low-Power Magnetic Torquers for Spacecraft

Attitude Control," *Institute of Electrical and Electronic Engineers Transactions on Mechanics*, 2004, (Submitted).

[16] National Aeronautics and Space Administration, EXIST: Energetic X-ray Imaging Survey Telescope Webpage, http://exist.gsfc.nasa.gov/, (24 Apr 2011).

[17] D. V. Guerrant, "Design and Analysis of Fully Magnetic Control for Picosatellite Stabilization," Master's Degree Thesis, Department of Aerospace Engineering, California Polytechnic State University, San Luis Obispo, CA, 2005

Other References

[18] Aalborg University, Denmark, AAU & DelFi-3 CubeSat WebPages, http://www.cubesat.auc.dk/, (24 Apr 2011).

[19] R. Azor, I. Y. Bar-Itzhack, and R. H. Harman, "Satellite Angular Rate Estimation from Vector Measurements," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 3, 1998, pp. 450-457.

[20] T. Bak, "Spacecraft Attitude Determination: A Magnetometer Approach," PhD Thesis, Department of Control Engineering, Aalborg University, Aalborg Ø, Denmark, 1999.

[21] Boeing Phantom Works, Seattle, WA, CSTB-1 Webpage, http://www.boeing.com/news/releases/2009/q1/090304b_nr.html, (25 Apr 2011).

[22] R. Braeunig, "Orbital Mechanics," http://www.braeunig.us/space/orbmech.htm, (24 Apr 2011).

[23] California Polytechnic State University, CA, CP-1, 2, 3, & 4 CubeSat Webpage, http://polysat.calpoly.edu/index.php, (25 Apr 2011).

[24] Center for Robotic Exploration and Space Technologies, CA, GeneSat-1 Webpage, http://genesat1.engr.scu.edu/dashboard/, (24 Apr 2011).

[25] M. Challa, G. Natanson, J. Deutschmann, and K. Galal, "A PC-Based Magnetometer-Only Attitude and Rate Determinations System for Gyroless Spacecraft," *Proceedings of the Flight Mechanics and Estimation Theory Symposium*, NASA Conference Publication No. 3299, Goddard Space Flight Center, Greenbelt, May 1995.

[26] M. Challa, G. Natanson, and C. Wheeler, "Simultaneous Determination of Spacecraft Attitude and Rates Using Only a Magnetometer," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference,* American Institute of Aeronautics and Astronautics, San Diego, CA, 1996, pp. 544-553.

[27] M. Challa and C. Wheeler, "Accuracy Studies of a Magnetometer-Only Attitude-and-Rate-Determination System," *Proceedings of the Flight Mechanics and Estimation Theory Symposium*, NASA Conference Publication No. 3333, Goddard Space Flight Center, Greenbelt, May 1996.

[28] V. A. Chobotov , *Spacecraft Attitude Dynamics and Control.* Malabar, FL: Krieger Publishing Company, 1991.

[29] Cornell University, NY, Ionospheric sCintillation Experiment CUBESat Project, ICE Cube 1 & 2 Webpage, http://www.mae.cornell.edu/cubesat/, (24 Apr 2011).

[30] J. Crassidis, "Angular Velocity Determination Directly from Star Tracker Measurements," *Journal of Guidance, Control, and Dynamics*, Vol 25, No. 6, 2002, pp. 1165-1168.

[31] J. D. DiMiceli, "Attitude Determination of a Student Satellite Using On-Orbit Magnetometer and Temperature Data," Master's Degree Thesis, Department of Aerospace Engineering, California Polytechnic State University, San Luis Obispo, California, 2009

[32] Fachhochschule Aachen, University of Applied Sciences, Germany, Compass-1 & 2 Webpage, http://www.raumfahrt.fh-aachen.de/, (25 Apr 2011).

[33] P. W. Fortescue and J. P. W. Stark, *Spacecraft Systems Engineering*, Chichester, England: John Wiley & Sons Ltd., 1991

[34] S. Ghuffar, "Design and Implementation of Attitude Determination Algorithm for the CubeSat UWE-3," Master's Degree Thesis, Department of Space Science, Luleå University of Technology, Luleå, Sweden, 2010.

[35] E. Kreyszig, *Advanced Engineering Mathematics* (6^{th} ed.). New York: John Wiley & Sons, 1988.

[36] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton, NJ: Princeton University Press, 1999.

[37] G. A. Natanson and M. S. Challa, "Magnetometer-Only Attitude and Rate Determination for a Gyro-Less Spacecraft," *Proceedings of the Third International Symposium on Space Mission Operations and Ground Data Systems,* NASA Conference Publication 3281, Greenbelt, MD, November 1994, pp. 791-798.

[38] Nihon University, Japan, SEEDS CubeSat Project Official Webpage, http://cubesat.aero.cst.nihon-u.ac.jp/english/seedsdetail_e.html, (24 Apr 2011).

[39] Y. Oshman and F. Dellus, "Spacecraft Angular Rate Estimation from a Single Directional Vector," *Proceedings of the 50th IAF International Astronautical Congress*, Amsterdam, The Netherlands, Oct. 1999, Paper No. IAF-99-A.302.

[40] Y. Oshman and F. Dellus, "Fast Estimation of Spacecraft Angular Velocity from Sequential Geomagnetic Field Observations," *Proceedings of the AIAA/AAS Astrodynamics Specialists Conference*, American Institute of Aeronautics and Astronautics, Reston, VA, 2002, pp. 322-330.

[41] Y. Oshman and F. Dellus, "Spacecraft Angular Velocity Estimation Using Sequential Observations of a Single Directional Vector," *Journal of Spacecraft and Rockets*, Vol. 40, No. 2, March-April 2003, pp. 237-247.

[42] M. Polites and C. Gibson, "A Low-Power Magnetic Torquer for Satellite Attitude Control," *Proceedings of the 8th Space Congress*, Vol. 1, 1971, pp. 7.21-7.30.

[43] M. L. Psiaki and Y. Oshman, "Spacecraft Attitude Rate Estimation from Geomagnetic Field Measurements," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 2, 2003, pp. 244-252.

[44] M. Psiaki, "Nanosatellite Attitude Stabilization Using Passive Aerodynamics and Active Magnetic Torquing," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, May-June 2004, pp. 347-355.

[45] J. Sellers, *Understanding Space: An Introduction to Astronautics* (2nd ed.). New York: McGraw-Hill Companies, Inc., 2000.

[46] M. D. Shuster and S. D. Oh, "Three-Axis Attitude Determination from Vector Observations," *AIAA Journal of Guidance and Control*, 4(1): 70-77, January 1981.

[47] M. D. Shuster, "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No 4, October-December 1993, pp. 439-517.

[48] Stanford University, CA, QuakeSat and QuakeSat LLC WebPages, http://www.quakefinder.com/services/spaceproducts.php, (24 Apr 2011).

[49] E. J. Strum, "Magnetic Attitude Estimation of a Tumbling Spacecraft," Master's Degree Thesis, Department of Aerospace Engineering, California Polytechnic State University, San Luis Obispo, California, 2005.

[50] K. Svartveit, "Attitude Determination of the NCUBE Satellite," Master's Degree Thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2003.

[51] Technical University of Berlin, Germany, BeeSAT-1, 2, & 3 Webpage, http://www.raumfahrttechnik.tu-berlin.de/beesat/v-menue2/project_overview/, (25 Apr 2011).

[52] Technical University of Denmark, Denmark, DTUsat Project Webpage, http://dtusat1.dtusat.dtu.dk/, (24 Apr 2011).

[53] M. Thomsen, "Michael's List of CubeSat Satellite Missions," http://mtech.dk/thomsen/space/cubesat.php, (24 Apr 2011).

[54] Tokyo Institute of Technology, Japan, Cubical TI Tech Engineering Satellite Website, CUTE-I, http://lss.mes.titech.ac.jp/ssp/cubesat/index_e.html, (24 Apr 2011).

[55] Tokyo Institute of Technology, Japan, Cubical TI Tech Engineering Satellite Website, Cute-1.7 APD, http://lss.mes.titech.ac.jp/ssp/cute1.7/cute1.7-1/index_e.html, (24 Apr 2011).

[56] P. Tortora and Y. Oshman, "Spacecraft Angular Rate Estimation from Magnetometer Data Only," *A Collection of Technical Papers of the AIA/AAS Astrodynamics Specialists Conference*, American Institute of Aeronautics and Astronautics, Reston, VA, 2000, pp. 304-310.

[57] P. Tortora and Y. Oshman, "Spacecraft Angular Rate Estimation from Magnetometer Data Only Using an Analytic Solution of Euler's Equations," Department of Aerospace Engineering, Israel Institute of Technology, TR TAE-090, Haifa, Israel, December 2002.

[58] P. Tortora, Y. Oshman, and F. Santoni, "Spacecraft Angular Rate Estimation from Magnetometer Data Only Using an Analytic Predictor," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, May-June 2004.

[59] University of Illinois, IL, Illinois Tiny Satellite Initiative, Illinois Observing Nanosatellite (ION) Webpage, http://cubesat.ece.illinois.edu/, (24 Apr 2011).

[60] University of Louisiana, Department of Electrical and Computer Engineering, LA, CAPE CubeSat Webpage, http://www.raumfahrttechnik.tu-berlin.de/beesat/v-menue2/project_overview/, (25 Apr 2011).

[61] University of Sergio Arboleda, Bogota Columbia, Libertad-1 CubeSat Webpage, http://www.usergioarboleda.edu.co/proyecto_espacial/, (25 Apr 2011).

[62] University of Tokyo, Japan, XI-IV & XI-V CubeSat WebPages, http://www.space.t.u-tokyo.ac.jp/cubesat/index-e.html, (24 Apr 2011).

[63] University of Toronto, Canada, Institute for Aerospace Studies, Space Flight Laboratory, CanX-1 Webpage, http://www.utias-sfl.net/nanosatellites/CanX1/, (24 Apr 2011).

[64] University of Würzburg, Germany, UWE-1, 2, 3 WebPages, http://www7.informatik.uni-wuerzburg.de/en/research/research_groups/space_exploration/projects/cubesat/ , (24 Apr 2011).

Appendix A [ACDS Simulation Software]

```
%*********************************************************************
%            Simulation of the UAF CubeSat Conceptual Design
%          As part of PhD Dissertation by Major Donald B. Mentch
%     The University of Alaska Fairbanks Department of Electrical Engineering
%  Derived from AEM 691-Special Topics in Spacecraft Attitude Dynamics & Control
%  with Dr. M. Polites, University of Alabama, Aerospace Engineering & Mechanics
%                      As of: 24 Apr 2011
%                         Phase 18
%*********************************************************************

clc; clear all; close all; CPU_time = cputime;

% **** MAKE THE NECESSARY VARIABLES GLOBAL **********************
global I Iinv IC incline eta_not CapOmega_not t1 t2 T_GG e1_O1 e3_O3
global gamma_dot gamma_not R FluxVector gamma SubVector T_Mag lat long
global M_CmdStar lambda_not lambda_dot Omega_CMD last status_old status_new
calc Torque_Mag
pi = 3.14159265358979; % Set a good value for pi
endplots = [1 1 1 1 1 1 1 1 1 1 1 1 1]; %Turns the 13 results plots on and off

% ******* SET AND CALCULATE THE ORBITAL PROPERTIES BELOW *******
% Program assumes a circular orbit w/ longitude of the ascending node NOT constant
incline = 90;                % Orbit Inclination (degrees)
pole_window = 10;            % Polar alignment windows in degrees
offset = 10;                 % Amount the window is offset in the wake direction (i.e.
earlier) in degrees
eq_window = 15;              % Equatorial alignment window in degrees
lat_trigger = incline-pole_window;  % Corrects polar window for inclination angle
switches = 0;                % Total number of torquer switches
status_old=[0 0 0]';         % Tracks magnetic torquer switches [current status]
status_new=[0 0 0]';         % Tracks magnetic torquer swutches [new status]
eta_not = 0;                 % Initial S/C Position wrt ascending node (degrees);
gamma_not = 0;               % Initial Magnetic axis position wrt Inertial
GM = 3.986004418*10^14;          % Gravitational constant (m^3/sec^2)
Re = 6.378137*10^6; h  = 600000;   % Radius of the Earth (m)[IUGG value of
                                   % equatorial radius]; Orbit Altitude (m)
R =  (Re + h)/1000;          % Position vector magnitude (km)
CapOmega_not=sqrt(GM/(Re + h)^3);   % Orbital Rate (rad/sec)
incline=incline*(pi/180);    % Calculates inclination angle in radians
lambda_not  = 0;             % Initial longitude of the ascending node (degrees)
lambda = lambda_not;         % This angle will vary with time due to equatorial bulge.
```

```
lambda_dot = -4.17089*10^7 *(R)^(-7/2)*cos(incline); % Rate that lambda changes
                                % (rad/sec)
lambda=lambda*(pi/180);  % Calculates inclination angle in radians
eta_not=eta_not*(pi/180);  % Calculates eta angle in radians
Tp = (2*pi)/CapOmega_not;  % Orbital period (sec)
gamma_dot = (2*pi)/(24*3600);  % Earth rotation rate (2 *pi in 24 hours*3600sec/hour)
M_CmdStar = [0 0 0]';       % Magnetic dipole moment commands in A-m2
Mode = 3;                    % Sets attitude control logic // 1 detumble // 2 alignment
                             % phase 1 // 3 alignment phase 2
M_C = [0 0 0]';
M_B = [0 0 0]';
lat_old = 0;                % Used in the simulation to determine if the spacecraft is
                            % approaching a pole or moving away from it
A = 1;                      % Used to determine if spacecraft is north bound or south
                            % bound // 1 north bound // 2 south bound
c=1;                        % Trigger used to initiate retrograde correction
trip = 1;                   % Used as a timer to determine when it is time to switch from
                            % Mode 2 to Mode 3
bias = 0;                   % Denotes which bias window the spacecraft is in  // 1 equator
                            % north bound // 2 north pole // 3 equator south bound
per = 0.0;                  % Used in the unbalanced torquer calculations  // value is
                            % amount of magnetic dipole moment present when no torque
                            % is commanded  // a percent of the large torquer magnetic
                            % dipole moment

% ***** SET AND CALCULATE THE INITIAL CONDITIONS *****************
ThetaInit = [90 0 0]';  % Initial Vehicle orientation wrt Inertial (E) axes (degrees)
Wind = [0 0 0]';        % Tracks roll, pitch, and yaw relative to orbital frame (degrees)
Omega_MeasInit=[0 0 CapOmega_not]'; % Initial Measured value of Body Rates
                                    % (rad/sec)
Omega_CMD = [0 0 CapOmega_not]';   % The commanded spacecraft body rates
ThetaInit = (ThetaInit*pi/180); % Converts initial attitude to radians for Quaternion
Calculation
Wind = Wind*pi/180;             % Converts wind to radians
Th=ThetaInit;                   % Sets up theta for calculation on next few lines

% STEP #1: Set up the direction cosine matrix [R123 Eq 1.10 // Page 5 of Chobotov]
DCM(1,1) = cos(Th(3,1))*cos(Th(2,1));  DCM(1,2) =
cos(Th(3,1))*sin(Th(2,1))*sin(Th(1,1))+sin(Th(3,1))*cos(Th(1,1));  DCM(1,3) = -
cos(Th(3,1))*sin(Th(2,1))*cos(Th(1,1))+sin(Th(3,1))*sin(Th(1,1));
DCM(2,1) = -sin(Th(3,1))*cos(Th(2,1)); DCM(2,2) = -
sin(Th(3,1))*sin(Th(2,1))*sin(Th(1,1))+cos(Th(3,1))*cos(Th(1,1)); DCM(2,3) =
sin(Th(3,1))*sin(Th(2,1))*cos(Th(1,1))+cos(Th(3,1))*sin(Th(1,1));
```

```
DCM(3,1) = sin(Th(2,1)); DCM(3,2) = -cos(Th(2,1))*sin(Th(1,1)); DCM(3,3) =
cos(Th(2,1))*cos(Th(1,1));

% STEP #2: Calculate quaternion from DCM
q(4) = 0.5*sqrt((1+DCM(1,1)+DCM(2,2)+DCM(3,3)));
q(1) = (1/(4*q(4)))*(DCM(2,3)-DCM(3,2)); q(2) = (1/(4*q(4)))*(DCM(3,1)-DCM(1,3));
q(3) = (1/(4*q(4)))*(DCM(1,2)-DCM(2,1));
q = q';        % Makes q into a column vector (column vectors are used throughout this
program)
Qe_EInit = q;  % Initial Vehicle Attitude with respect to Inertial (E) axes
qT1 = q(1,1); qT2 = q(2,1); qT3 = q(3,1); qT4 = q(4,1); % Rename to keep track of
variables

% STEP #3: Calculate Euler angles from the quaternion (uses the DCM reltationships)
ThetasM(1,1) = 0;
ThetasM(1,2) = Th(1,1); % Theta 1 from the given data
ThetasM(1,3) = Th(2,1); % Theta 2 from the given data
ThetasM(1,4) = Th(3,1); % Theta 3 from the given data
QE_eInit = -1*Qe_EInit; % Quaternion is inverse of the one above for initial calculations
QE_eInit(4,1)=-QE_eInit(4,1); % Fix the 4th element
Qc_EInit = q;

% ***** SET THE TIME ELEMENTS **************************************
tf = 2*5793;   % final time (seconds) // Presented as orbits * time for one orbit at 600 km
delta_t = 10.0;  % time increment (seconds) // The charge-discharge cycle time
count = 0;       % sets the count increment for data storage at zero // this number is the
                 % index used in data storage
steps = tf/delta_t; % # of time steps

% ******* MOMENTS OF INERTIA ******************************
% Set the principal moments of inertia
Ip(1,1) = 0.002217;  Ip(2,2) = 0.002536; Ip(3,3) = 0.002536; % (kg-m^2)

% Calculate the fully populated Inertia Matrix (kg-m^2)
I = Ip;
Iinv = inv(I); % Calculate the inverse for use in ODE45

% ****** CONTROL LAW PARAMETERS ***************************
Km = [0.00024 0.00024 0.00015]'; % Detumble gains
Ka = [0.00024 0.00024 0.00015]';  % Alignment gain
M_CmdLim_s = 0.00011;            % Magnetic Dipole Vernier Torquer (amperes-m^2)
M_CmdLim_b = 0.022;              % Magnetic Dipole Large Torquer (amperes-m^2)

% ***** INITIALIZE VARIABLES FOR FIRST TIME STEP ********************
```

```
QE_e = QE_eInit; Qe_E = Qe_EInit;
OmegaMeas  = Omega_MeasInit;     % Set OmegaMeas to OmegaInit for first time step
QuatCalc   = Qe_EInit;  % Set QuatCalc (used inside time loop) to QuatInit
Torque_Mag = [0 0 0]';           % Sets initial Torque due to magnetic field
ThetaOld = ThetaInit*pi/180;  % Sets initial value of theta to be integrated to the initial
                              % attitude value
T_Mag = [0 0 0]';  % Magnetic torque // Dipole multiplied by the local flux field

% * SET INITIAL VALS OF MATRICES CREATED FOR REC. KEEPING & PLOT *
OmegaM2(1,1) = 0; OmegaM2(1,2) = Omega_MeasInit(1,1); OmegaM2(1,3) =
Omega_MeasInit(2,1); OmegaM2(1,4) = Omega_MeasInit(3,1);
ThetaRGM(1,2) = ThetaInit(1,1); ThetaRGM(1,3) = ThetaInit(2,1); ThetaRGM(1,4) =
ThetaInit(3,1);
QuatM(1,2) = QE_eInit(1,1); QuatM(1,3) = QE_eInit(2,1); QuatM(1,4) = QE_eInit(3,1);
QuatM(1,5) = QE_eInit(4,1);

% Calculates the difference between spacecraft body axes and the orbital reference frame
%  axes // Equation (3) in the thesis
e1_O1(1,1) = 0; e1_O1(1,2) = acos(DCM(1,1)*(cos(lambda)*cos(eta_not)-
sin(lambda)*cos(incline)*sin(eta_not))+
DCM(1,2)*(sin(lambda)*cos(eta_not)+cos(lambda)*cos(incline)*sin(eta_not))+
DCM(1,3)*(sin(incline)*sin(eta_not)));
e2_O2(1,1) = 0; e2_O2(1,2) = acos(DCM(2,1)*(-cos(lambda)*sin(eta_not)-
sin(lambda)*cos(incline)*cos(eta_not))+ DCM(2,2)*(-
sin(lambda)*sin(eta_not)+cos(lambda)*cos(incline)*cos(eta_not))+DCM(2,3)*(sin(inclin
e)*cos(eta_not)));
e3_O3(1,1) = 0; e3_O3(1,2) = acos(DCM(3,1)*(sin(lambda)*sin(incline))-
DCM(3,2)*(cos(lambda)*sin(incline))+DCM(3,3)*(cos(incline)));

% **** START OF TIME LOOPING ************************************
for tSTEP=1:steps; %tf
   t1=(tSTEP-1)*delta_t; t2=tSTEP*delta_t; % Sets values for ODE time integration
   etaO = eta_not + CapOmega_not*t1;       % Tracks current spacecraft position wrt
                                           % ascending node
   lambda = lambda_not + lambda_dot*t1;    % Tracks current position of the longitude
                                           %  of the ascending node

% CALCULATE DCM FOR TRANSFORMATION FROM INERTIAL(E) TO
%  ORBITAL(O) // Equation (3) in thesis
DCMO_E(1,1) = cos(etaO)*cos(lambda)-sin(etaO)*cos(incline)*sin(lambda);
DCMO_E(1,2) = cos(etaO)*sin(lambda)+sin(etaO)*cos(incline)*cos(lambda);
DCMO_E(1,3) = sin(etaO)*sin(incline);
```

```
DCMO_E(2,1) = -sin(etaO)*cos(lambda)-cos(etaO)*cos(incline)*sin(lambda);
DCMO_E(2,2) = -sin(etaO)*sin(lambda)+cos(etaO)*cos(incline)*cos(lambda);
DCMO_E(2,3) = cos(etaO)*sin(incline);
DCMO_E(3,1) = sin(incline)*sin(lambda);                    DCMO_E(3,2) = -
sin(incline)*cos(lambda);                    DCMO_E(3,3) = cos(incline);

% CALCULATE THE QUATERNION TO GO FROM INERTIAL(E) TO
%  ORBITAL(O) FROM DCMO_E
   QO_E(4,1) = 0.5*sqrt((1+DCMO_E(1,1)+DCMO_E(2,2)+DCMO_E(3,3)));
   QO_E(1,1) = (1/(4*QO_E(4,1)))*(DCMO_E(2,3)-DCMO_E(3,2)); QO_E(2,1) =
(1/(4*QO_E(4,1)))*(DCMO_E(3,1)-DCMO_E(1,3)); QO_E(3,1) =
(1/(4*QO_E(4,1)))*(DCMO_E(1,2)-DCMO_E(2,1));

% CALCULATE QUAT TO GO FROM BODY(E) TO ORBITAL(O) FROM 4 by 4
   % a. Calculate the 4 by 4 matrix required
   QO_EM(1,1) = QO_E(4,1);  QO_EM(1,2) = QO_E(3,1);  QO_EM(1,3) = -QO_E(2,1);
QO_EM(1,4) = QO_E(1,1);
   QO_EM(2,1) = -QO_E(3,1); QO_EM(2,2) = QO_E(4,1);  QO_EM(2,3) = QO_E(1,1);
QO_EM(2,4) = QO_E(2,1);
   QO_EM(3,1) = QO_E(2,1);  QO_EM(3,2) = -QO_E(1,1); QO_EM(3,3) = QO_E(4,1);
QO_EM(3,4) = QO_E(3,1);
   QO_EM(4,1) = -QO_E(1,1); QO_EM(4,2) = -QO_E(2,1); QO_EM(4,3) = -
QO_E(3,1); QO_EM(4,4) = QO_E(4,1);
   % b. Calculate the Quaternion via Quaternion Algebra
   QO_e = QO_EM*QE_e;

   gamma = gamma_not + gamma_dot*t1;  % Calculates current magnetic reference
                                      % angle // used to relate ECI & ECEF frames
   eta = eta_not + CapOmega_not*t1;  % Calculates current spacecraft position wrt the
                                     %  ascending node
   lambda = lambda_not + lambda_dot*t1;   % Calculates current position of the
                                          %  longitude of the ascending node
   a(1,1) = (Qe_E(1))^2 - (Qe_E(2))^2 - (Qe_E(3))^2 + (Qe_E(4))^2;  % Calculation of
                                                    % DCM for body from ECI
   a(1,2) = 2*(Qe_E(1)*Qe_E(2)+Qe_E(3)*Qe_E(4));
   a(1,3) = 2*(Qe_E(1)*Qe_E(3)-Qe_E(2)*Qe_E(4));
   a(2,1) = 2*(Qe_E(1)*Qe_E(2)-Qe_E(3)*Qe_E(4));
   a(2,2) = -(Qe_E(1)^2) + Qe_E(2)^2 - (Qe_E(3)^2) + Qe_E(4)^2;
   a(2,3) = 2*(Qe_E(1)*Qe_E(4)+Qe_E(2)*Qe_E(3));
   a(3,1) = 2*(Qe_E(1)*Qe_E(3)+Qe_E(2)*Qe_E(4));
   a(3,2) = 2*(-(Qe_E(1)*Qe_E(4))+Qe_E(2)*Qe_E(3));
   a(3,3) = -(Qe_E(1))^2 - (Qe_E(2))^2 + (Qe_E(3))^2 + (Qe_E(4))^2;

   % Calcs required to convert eta, gamma, incline, lambda into latitude and longitude
```

```
ui=R*cos(eta)-R*sin(eta)*cos(incline);
vi=R*cos(eta)+R*sin(eta)*cos(incline);
we=R*sin(eta)*sin(incline);
ue=cos(gamma+lambda)*ui+sin(gamma+lambda)*vi;
ve=-sin(gamma+lambda)*ui+cos(gamma+lambda)*vi;
lat = asin(we/R)*180/pi;
long = (ve/abs(ve))*acos(ue/sqrt(ue^2+ve^2))*180/pi-45;

% Data is passed to magfd to obtain the local mag flux field in spacecraft body frame
[FluxVector] = magfd(2012.5,2,R,90-lat,long,a);

Omega_error = Omega_CMD-OmegaMeas;  % Equation (17) in thesis

M_Cmd = -(cross(Omega_error,FluxVector)/(dot(FluxVector,FluxVector))); % Cross
      % Product law for the Dipole moment // Equation (19) in thesis
M_Cmd(1)=M_Cmd(1)*Km(1);
M_Cmd(2)=M_Cmd(2)*Km(2);
M_Cmd(3)=M_Cmd(3)*Km(3);

M_CmdStar = [0 0 0]';  % This array contains the magnetic dipole moment commands
                       %  passed to ODE 45

if Mode == 1  % Degin detumble phase // This section contains the quantizer seen in
              %    Figure 14 in thesis

   if M_Cmd(1) > M_CmdLim_b && status_old(1) == 0  % x-axis quantizer // the
            % code ensures that the magnetic dipole commands
   M_CmdStar(1) = 2.0*M_CmdLim_b;  % comply with the 1 switch per axis per
                                   %  charge cycle assumption
   status_new(1)=1;  % preventing a switch from +2m to -2m in one cycle or vice versa
   end
   if M_Cmd(1) > -3.0*M_CmdLim_b && status_old(1) == -2
      M_CmdStar(1) = -2.0*M_CmdLim_b;
      status_new(1)=-1;
   end
   if M_Cmd(1) > -M_CmdLim_b && status_old(1) == -1
      M_CmdStar(1) = 0;
      status_new(1)= 0;
   end
   if M_Cmd(1) < -M_CmdLim_b && status_old(1) == 0
      M_CmdStar(1) = -2.0*M_CmdLim_b;
      status_new(1)= -1;
   end
   if M_Cmd(1) < 3.0*M_CmdLim_b && status_old(1) == 2
```

```
   M_CmdStar(1) = 2.0*M_CmdLim_b;
   status_new(1)= 1;
end
if M_Cmd(1) < M_CmdLim_b && status_old(1) == 1
   M_CmdStar(1) = 0;
   status_new(1)= 0;
end

if M_Cmd(2) > M_CmdLim_b && status_old(2) == 0  % y-axis quantizer
   M_CmdStar(2) = 2.0*M_CmdLim_b;
   status_new(2)=1;
end
if M_Cmd(2) > -3.0*M_CmdLim_b && status_old(2) == -2
   M_CmdStar(2) = -2.0*M_CmdLim_b;
   status_new(2)=-1;
end
if M_Cmd(2) > -M_CmdLim_b && status_old(2) == -1
   M_CmdStar(2) = 0;
   status_new(2)= 0;
end
if M_Cmd(2) < -M_CmdLim_b && status_old(2) == 0
   M_CmdStar(2) = -2.0*M_CmdLim_b;
   status_new(2)= -1;
end
if M_Cmd(2) < 3.0*M_CmdLim_b && status_old(2) == 2
   M_CmdStar(2) = 2.0*M_CmdLim_b;
   status_new(2)= 1;
end
if M_Cmd(2) < M_CmdLim_b && status_old(2) == 1
   M_CmdStar(2) = 0;
   status_new(2)= 0;
end

if M_Cmd(3) > M_CmdLim_b && status_old(3) == 0  % z-axis quantizer
   M_CmdStar(3) = 2.0*M_CmdLim_b;
   status_new(3)=1;
end
if M_Cmd(3) > -3.0*M_CmdLim_b && status_old(3) == -2
   M_CmdStar(3) = -2.0*M_CmdLim_b;
   status_new(3)=-1;
end
if M_Cmd(3) > -M_CmdLim_b && status_old(3) == -1
   M_CmdStar(3) = 0;
   status_new(3)= 0;
```

```
      end
      if M_Cmd(3) < -M_CmdLim_b && status_old(3) == 0
         M_CmdStar(3) = -2.0*M_CmdLim_b;
         status_new(3)= -1;
      end
      if M_Cmd(3) < 3.0*M_CmdLim_b && status_old(3) == 2
         M_CmdStar(3) = 2.0*M_CmdLim_b;
         status_new(3)= 1;
      end
      if M_Cmd(3) < M_CmdLim_b && status_old(3) == 1
         M_CmdStar(3) = 0;
         status_new(3)= 0;
      end

      % The following line detemines if the measured rates are close
      % enough to the commanded rates for the detumble to be considered
      % complete and Mode 2 to begin
      if (abs(Omega_error(1)) < 0.002) && (abs(Omega_error(2)) < 0.002) &&
(abs(Omega_error(3)) < 0.002)
        Mode = 2;
      end
   end

   if Mode == 2    % Begin alignment Mode 2
      if lat > lat_old  % Determines if spacecraft is traveling north bound
         A = 1;
      end
      if lat < lat_old  % Determines if spacecraft is traveling south bound
         A = 2;
      end

      if lat < (eq_window/2+offset) && lat > -(eq_window/2-offset) && A==2
%  Determines if the spacecraft is in the equatorial window south bound
         M_B = [0 -2.0*M_CmdLim_s 0]';      % Commands proper bias for this region
         omega_prime = Omega_CMD-OmegaMeas;  % Calculation of the body rate error
                                             %  Equation (17) in thesis
         M_C=-cross(omega_prime,FluxVector)/dot(FluxVector,FluxVector);  % Cross-
                        %  product law // Equation (19) in thesis
         M_C(1)=M_C(1)*Ka(1);
         M_C(2)=M_C(2)*Ka(2);
         M_C(3)=M_C(3)*Ka(3);
         bias = 3;

      elseif lat > (lat_trigger-offset) && lat > lat_old % Determines if the spacecraft is in
```

```
                                                    %  the north polar window
   M_B = [-2.0*M_CmdLim_s 0 0]';      % Commands proper bias for this region
   omega_prime = Omega_CMD-OmegaMeas;  % Equation (17)
   M_C=-cross(omega_prime,FluxVector)/dot(FluxVector,FluxVector);
                    % Equation (19)
   M_C(1)=M_C(1)*Ka(1);
   M_C(2)=M_C(2)*Ka(2);
   M_C(3)=M_C(3)*Ka(3);
   bias = 2;
   A = 2;

elseif lat > -(eq_window/2+offset) && lat < (eq_window/2-offset) && A==1
       % Determines if the spacecraft is in the equatorial region north bound
   M_B = [0 2.0*M_CmdLim_s 0]';       % Commands proper bias for this region
   omega_prime = Omega_CMD-OmegaMeas;  % Equation (17)
   M_C=-cross(omega_prime,FluxVector)/dot(FluxVector,FluxVector);
                    % Equation (19)
   M_C(1)=M_C(1)*Ka(1);
   M_C(2)=M_C(2)*Ka(2);
   M_C(3)=M_C(3)*Ka(3);
   bias = 1;
else
   M_B = [0 0 0]';  % Outside of the windows the bias is zero
   M_C = [0 0 0]';  % Outside of the windows the torque command is zero
   bias = 0;
end

M_T = M_C + M_B;    % Total magnetic torque is the bias plus the commanded

if M_T(1) < -M_CmdLim_s  && bias == 2  % Mode 2 quantizer for the x-axis
                              % (north pole)
   M_CmdStar(1) = -2.0*M_CmdLim_s;     % this section executes the bias
   status_new(1) = 2;
elseif M_T(1) > -M_CmdLim_s && bias == 2
   M_CmdStar(1) = 0;
   status_new(1) = 0;
end

if M_T(1) > M_CmdLim_s  && (bias == 3 || bias==1) % Mode 2 quantizer for the
                    % x-axis (other than north pole)
   M_CmdStar(1) = 2.0*M_CmdLim_s;
   status_new(1) = 1;
elseif M_T(1) < -M_CmdLim_s && (bias == 3 || bias ==1)
   M_CmdStar(1) = -2.0*M_CmdLim_s;
```

```
   status_new(1) = -1;
else
   if bias == 3 || bias ==1
      M_CmdStar(1) = 0;
   end
end

if M_T(2) > M_CmdLim_s  && (bias == 2 || bias ==4) % Mode 2 quantizer for y-
                                                   %  axis (other than equator)
   M_CmdStar(2) = 2.0*M_CmdLim_s;
   status_new(2) = 1;
elseif M_T(2) < -M_CmdLim_s && (bias == 2 || bias ==4)
   M_CmdStar(2) = -2.0*M_CmdLim_s;
   status_new(2) = -1;
else
   if bias == 2 || bias ==4
      M_CmdStar(2) = 0;
   end
end

if M_T(2) < -M_CmdLim_s  && bias == 3 % Mode 2 quantizer for y-axis (equator
                                      %   south bound)
   M_CmdStar(2) = -2.0*M_CmdLim_s;
   status_new(2) = 2;
elseif M_T(2) > -M_CmdLim_s && bias == 3
   M_CmdStar(2) = 0*M_CmdLim_s;
   status_new(2) = 0;
end

if M_T(2) > M_CmdLim_s  && bias == 1 % Mode 2 quantizer for y-axis (equator
                                     %   north bound)
   M_CmdStar(2) = 2.0*M_CmdLim_s;
   status_new(2) = 2;
elseif M_T(2) < M_CmdLim_s && bias == 1
   M_CmdStar(2) = 0*M_CmdLim_s;
   status_new(2) = 0;
end

if M_T(3) > M_CmdLim_s            % Mode 2 quantizer for z-axis (all regions)
   M_CmdStar(3) = 2.0*M_CmdLim_s;
   status_new(3) = 1;
elseif M_T(3) < -M_CmdLim_s
   M_CmdStar(3) = -2.0*M_CmdLim_s;
   status_new(3) = -1;
```

```
   else
      M_CmdStar(3) = 0;
   end

   trip = trip +1;    % After 10 orbits Mode 2 switches to Mode 3
   if trip > 10*550
    Mode = 3;
   end
end

if Mode == 3          % Begin Mode 3 // Polar bias only

   if lat > lat_old   % Determine if satellite is north bound
      A = 1;
   end
   if lat < lat_old   % Determine if satellite is south bound
      A = 2;
   end

   if lat > 45 && lat < 50 && A == 1 % Check for retrograde
      if FluxVector(1) > 0
         c = 3;
      end;
   end
   if lat > (lat_trigger) && lat > lat_old  % Determines if the spacecraft is in the north
                                           %    polar window
      M_B = [-2.0*M_CmdLim_s 0 0]';  % Commands proper bias for this region
      omega_prime = Omega_CMD-OmegaMeas;
      M_C=-cross(omega_prime,FluxVector)/dot(FluxVector,FluxVector);
      M_C(1)=M_C(1)*Ka(1);
      M_C(2)=M_C(2)*Ka(2);
      M_C(3)=M_C(3)*Ka(3);
      bias = 2;
      A = 2;
      if FluxVector(1) > 0  % Checks for Prograde Reversed Condition
         c = 2;
      end
   else % When outside of the polar reqion a detumble is commanded
       %  with the vernier torquers.  This prevents the spacecraft
       %  from drifting out of alignment.  Uses same procedure
       %  as Mode 1 but uses the vernier torquers.
      if M_Cmd(1) > M_CmdLim_s && status_old(1) == 0
      M_CmdStar(1) = 2.0*M_CmdLim_s;
      status_new(1)=1;
```

```
end
if M_Cmd(1) > -3.0*M_CmdLim_s && status_old(1) == -2
   M_CmdStar(1) = -2.0*M_CmdLim_s;
   status_new(1)=-1;
end
if M_Cmd(1) > -M_CmdLim_s && status_old(1) == -1
   M_CmdStar(1) = 0;
   status_new(1)= 0;
end
if M_Cmd(1) < -M_CmdLim_s && status_old(1) == 0
   M_CmdStar(1) = -2.0*M_CmdLim_s;
   status_new(1)= -1;
end
if M_Cmd(1) < 3.0*M_CmdLim_s && status_old(1) == 2
   M_CmdStar(1) = 2.0*M_CmdLim_s;
   status_new(1)= 1;
end
if M_Cmd(1) < M_CmdLim_s && status_old(1) == 1
   M_CmdStar(1) = 0;
   status_new(1)= 0;
end

if M_Cmd(2) > M_CmdLim_s && status_old(2) == 0
   M_CmdStar(2) = 2.0*M_CmdLim_s;
   status_new(2)=1;
end
if M_Cmd(2) > -3.0*M_CmdLim_s && status_old(2) == -2
   M_CmdStar(2) = -2.0*M_CmdLim_s;
   status_new(2)=-1;
end
if M_Cmd(2) > -M_CmdLim_s && status_old(2) == -1
   M_CmdStar(2) = 0;
   status_new(2)= 0;
end
if M_Cmd(2) < -M_CmdLim_s && status_old(2) == 0
   M_CmdStar(2) = -2.0*M_CmdLim_s;
   status_new(2)= -1;
end
if M_Cmd(2) < 3.0*M_CmdLim_s && status_old(2) == 2
   M_CmdStar(2) = 2.0*M_CmdLim_s;
   status_new(2)= 1;
end
if M_Cmd(2) < M_CmdLim_s && status_old(2) == 1
   M_CmdStar(2) = 0;
```

```
      status_new(2)= 0;
    end

    if M_Cmd(3) > M_CmdLim_s && status_old(3) == 0
      M_CmdStar(3) = 2.0*M_CmdLim_s;
      status_new(3)=1;
    end
    if M_Cmd(3) > -3.0*M_CmdLim_s && status_old(3) == -2
      M_CmdStar(3) = -2.0*M_CmdLim_s;
      status_new(3)=-1;
    end
    if M_Cmd(3) > -M_CmdLim_s && status_old(3) == -1
      M_CmdStar(3) = 0;
      status_new(3)= 0;
    end
    if M_Cmd(3) < -M_CmdLim_s && status_old(3) == 0
      M_CmdStar(3) = -2.0*M_CmdLim_s;
      status_new(3)= -1;
    end
    if M_Cmd(3) < 3.0*M_CmdLim_s && status_old(3) == 2
      M_CmdStar(3) = 2.0*M_CmdLim_s;
      status_new(3)= 1;
    end
    if M_Cmd(3) < M_CmdLim_s && status_old(3) == 1
      M_CmdStar(3) = 0;
      status_new(3)= 0;
    end
    bias = 0;
  end
  M_T = M_C + M_B;
  if M_T(1) < -M_CmdLim_s  && bias == 2
    M_CmdStar(1) = -2.0*M_CmdLim_s;
    status_new(1) = 2;
  elseif M_T(1) > -M_CmdLim_s && bias == 2
    M_CmdStar(1) = 0;
    status_new(1) = 0;
  end

  if M_T(2) > M_CmdLim_s  && (bias == 2 || bias ==4)
    M_CmdStar(2) = 2.0*M_CmdLim_s;
    status_new(2) = 1;
  elseif M_T(2) < -M_CmdLim_s && (bias == 2 || bias ==4)
    M_CmdStar(2) = -2.0*M_CmdLim_s;
    status_new(2) = -1;
```

```
      else
         if bias == 2 || bias ==4
            M_CmdStar(2) = 0;
         end
      end

      if M_T(3) > M_CmdLim_s
         M_CmdStar(3) = 2.0*M_CmdLim_s;
         status_new(3) = 1;
      elseif M_T(3) < -M_CmdLim_s
         M_CmdStar(3) = -2.0*M_CmdLim_s;
         status_new(3) = -1;
      else
         M_CmdStar(3) = 0;
      end

      if c==2  % Executes retrgrade correction
         M_CmdStar(1) = -2.0*M_CmdLim_b;
         status_new(1) = -2;
         c = 1;
         Mode = 2;
         trip = 0;
      end
      if c==3  && bias==2  % Executes prograde reverse correction
         M_CmdStar(1) = -2.0*M_CmdLim_b;
         status_new(1) = -2;
         c = 1;
         Mode = 2;
         trip = 0;
      end
   end

   M_CmdStar(1)=M_CmdStar(1)+per*M_CmdLim_b/100;

calc = status_new-status_old;
status_old = status_new;

% **** BUILD INITIAL CONDITION VECTOR TO PASS TO ODE45 *********
IC = [OmegaMeas; Qe_E];
tspan = [t1 t2];
[time,State] = ode45('ExistSub2',tspan,IC); % Returns State vector
npts=length(time);
%*****************************************************************
```

```
% * FILE RESULTS IN SEP. MATRICES FOR RECORD KEEPING/PLOTTING *
% This routine records data at a fraction of the rate of the calculation rate
count = count +1; % Increments count after ODE 45 for data storage minimizing scheme
fprintf('\n    %5.1f\n',  t1)
if count == 1 %determines how many points to record based on time increment
  Int=tSTEP+1;
  % ******** DATA STORAGE OCCURS HERE ************************
   % a. Omegas
   OmegaM2(Int,1) = time(npts);    % Puts time into the first column of Omegas
   OmegaM2(Int,2) = State(npts,1);
   OmegaM2(Int,3) = State(npts,2);
   OmegaM2(Int,4) = State(npts,3);
   % b. Quaternions
   QuatM(Int,1) = time(npts);   %Puts time into the first column of Quaternions
   QuatM(Int,2) = QO_e(1,1); QuatM(Int,3) = QO_e(2,1);
   QuatM(Int,4) = QO_e(3,1); QuatM(Int,5) = QO_e(4,1);
   % c. Wind
   WindM(Int,1) = time(npts);
   WindM(Int,2) =
(180/pi)*atan2(2*(QO_e(4,1)*QO_e(1,1)+QO_e(2,1)*QO_e(3,1)),QO_e(4,1)*QO_e(4,1
)-QO_e(1,1)*QO_e(1,1)-QO_e(2,1)*QO_e(2,1)+QO_e(3,1)*QO_e(3,1));
   WindM(Int,3) = (180/pi)*asin(-2*(QO_e(1,1)*QO_e(3,1)+QO_e(2,1)*QO_e(4,1)));
   WindM(Int,4) =
(180/pi)*atan2(2*(QO_e(1,1)*QO_e(2,1)+QO_e(3,1)*QO_e(4,1)),QO_e(4,1)*QO_e(4,1
)+QO_e(1,1)*QO_e(1,1)-QO_e(2,1)*QO_e(2,1)-QO_e(3,1)*QO_e(3,1));
   Wind(1,1) = WindM(Int,2);
   Wind(2,1) = WindM(Int,3);
   Wind(3,1) = WindM(Int,4);

   ModeM(Int,1) = time(npts);
   ModeM(Int,2) = Mode;

   % d. Euler Angles: between Inertial(E) and Body(e) Axes
   qT1 = State(npts,4); qT2 = State(npts,5); qT3 = State(npts,6); qT4 = State(npts,7);
   ThetasM(Int,1) = time(npts); %Puts time into the first column of Thetas
   ThetasM(Int,3) = asin(2*qT1*qT3 + 2*qT2*qT4); % Theta2 has to be calculated first
                                                 % due to direction cosine matrix
   ThetasM(Int,2) = acos((-qT1^2-qT2^2+qT3^2+qT4^2)/cos(ThetasM(Int,3))); %Theta1
                                                 % calculated using Theta2
   ThetasM(Int,4) = acos((qT1^2-qT2^2-qT3^2+qT4^2)/cos(ThetasM(Int,3)));  %Theta3
                                                 % calculated using Theta2

   a(1,1) = (qT1)^2 - (qT2)^2 - (qT3)^2 + (qT4)^2;
   a(1,2) = 2*(qT1*qT2+qT3*qT4);
```

```
   a(1,3) = 2*(qT1*qT3-qT2*qT4);
   a(2,1) = 2*(qT1*qT2-qT3*qT4);
   a(2,2) = -(qT1^2) + qT2^2 - (qT3^2) + qT4^2;
   a(2,3) = 2*(qT1*qT4+qT2*qT3);
   a(3,1) = 2*(qT1*qT3+qT2*qT4);
   a(3,2) = 2*(-(qT1*qT4)+qT2*qT3);
   a(3,3) = -(qT1)^2 - (qT2)^2 + (qT3)^2 + (qT4)^2;

   % e. Gravity Gradient Torques
   T_ggM(Int,1) = time(npts);
   T_ggM(Int,2) = T_GG(1);
   T_ggM(Int,3) = T_GG(2);
   T_ggM(Int,4) = T_GG(3);

    % calculate angles between Body(e) and Orbital(O) axes
   gamma = gamma_not + gamma_dot*t2;
   eta = eta_not + CapOmega_not*t2;
   lambda = lambda_not + lambda_dot*t2;
   e1_O1(Int,1) = time(npts);
   e1_O1(Int,2) = real(acos(a(1,1)*(cos(lambda)*cos(eta)-
sin(lambda)*cos(incline)*sin(eta))+
a(1,2)*(sin(lambda)*cos(eta)+cos(lambda)*cos(incline)*sin(eta))+
a(1,3)*(sin(incline)*sin(eta))));
   e2_O2(Int,1) = time(npts);
   e2_O2(Int,2) = real(acos(a(2,1)*(-cos(lambda)*sin(eta)-
sin(lambda)*cos(incline)*cos(eta))+ a(2,2)*(-
sin(lambda)*sin(eta)+cos(lambda)*cos(incline)*cos(eta))+a(2,3)*(sin(incline)*cos(eta))))
;
   e3_O3(Int,1) = time(npts);
   e3_O3(Int,2) = real(acos(a(3,1)*(sin(lambda)*sin(incline))-
a(3,2)*(cos(lambda)*sin(incline))+a(3,3)*(cos(incline))));
   last = e3_O3(Int,2)*180/pi;

   % h. Flux Density Vector in Body(e) frame
   FluxVectorM(Int,1) = time(npts); FluxVectorM(Int,2)=gamma;
   FluxVectorM(Int,3) = FluxVector(1,1); FluxVectorM(Int,4) = FluxVector(2,1);
FluxVectorM(Int,5) = FluxVector(3,1);
   FluxVectorM(Int,6) = sqrt(FluxVector(1,1)^2 + FluxVector(2,1)^2 +
FluxVector(3,1)^2);

   OEr(Int,1)=Omega_error(1);
   OEr(Int,2)=Omega_error(2);
   OEr(Int,3)=Omega_error(3);
```

```
   %number of switches
   SW(Int,1) = time(npts);
   switches  = switches+sqrt(calc(1)*calc(1))+sqrt(calc(2)*calc(2))+sqrt(calc(3)*calc(3));
   SW(Int,2) = switches;

   % i. Flux Density Vector in Geo(G) frame
   SubVectorM(Int,1) = time(npts); SubVectorM(Int,2)=gamma;
   SubVectorM(Int,3) = SubVector(1,1); SubVectorM(Int,4) = SubVector(2,1);
SubVectorM(Int,5) = SubVector(3,1);
   SubVectorM(Int,6) = sqrt(SubVector(1,1)^2 + SubVector(2,1)^2 + SubVector(3,1)^2);
   SubVectorM(Int,7) = lat;
   SubVectorM(Int,8) = lat_old;
   SubVectorM(Int,9) = long;
   SubVectorM(Int,10)= bias;
   SubVectorM(Int,11)= Mode;

   % j. Magnetic Torque
   Mag_TorqueM(Int,1) = time(npts);
   Mag_TorqueM(Int,2) = T_Mag(1,1); Mag_TorqueM(Int,3) = T_Mag(2,1);
Mag_TorqueM(Int,4) = T_Mag(3,1);

   % k. Magnetic dipole
   MagDipoleM(Int,1) =time(npts);
   MagDipoleM(Int,2) = M_CmdStar(1,1); MagDipoleM(Int,3) = M_CmdStar(2,1);
MagDipoleM(Int,4) = M_CmdStar(3,1);

   count = 0;
   lat_old=lat;
end

% * NEED TO SET NEW VALUES FOR THE ROUTINES *****************
OmegaNew(1,1) = State(npts,1); OmegaNew(2,1) = State(npts,2); OmegaNew(3,1) =
State(npts,3);
QuatNew(1,1)  = State(npts,4); QuatNew(2,1)  = State(npts,5); QuatNew(3,1)  =
State(npts,6);
QuatNew(4,1) = State(npts,7);
% **** SET UP FOR NEW TIME LOOP ****************************
% (1) RESET VALUES FOR THE LOOP
Qe_E = QuatNew; OmegaMeas = OmegaNew;
% (2) CALCULATE THE INVERSE OF THE NEW CALCULATED QUATERNION
QE_e = -1*Qe_E; QE_e(4,1) = -QE_e(4,1); % Multiplies all elements time -1 & then
                                        %  fixes the 4th element
end % END OF THE TIME LOOP
RUNTIME = cputime-CPU_time
```

```
Lambda_final = lambda*180/pi;
Lambda_final
save Omegas.txt OmegaM2 -ascii -double -tabs
save Quats.txt QuatM -ascii -double -tabs
save Thetas.txt ThetasM -ascii -double -tabs
save GravityGrads.txt T_ggM -ascii -double -tabs
save Flux.txt FluxVectorM -ascii -double -tabs
save TorqueMag.txt T_Mag -ascii -double -tabs
save SubVector.txt SubVectorM -ascii -double -tabs

if endplots(1,1) == 1 % Plots the Euler Angles
   figure(1)
   clf
   hold on
   subplot(3,1,1), plot(ThetasM(:,1), ThetasM(:,2)*180/pi,'r-')
   %axis([0 tf -0 200])
   legend('Theta1',2)
   grid on
   ylabel ('Theta1(deg)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(ThetasM(:,1), ThetasM(:,3)*180/pi,'g-')
   %axis([0 tf -30 30])
   legend('Theta2',2)
   grid on
   ylabel ('Theta2(deg)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(ThetasM(:,1), ThetasM(:,4)*180/pi,'b-')
   %axis([0 tf -10 360])
   legend('Theta3',2)
   grid on
   ylabel ('Theta3(deg)')
   xlabel('time (sec)')
end

if endplots(1,2) == 1 % Plots body axes to orbital axes
   figure(2)
   clf
   hold on
   subplot(3,1,1), plot(e1_O1(:,1), e1_O1(:,2)*180/pi,'r-')
   %axis([0 tf -30 30])
   legend('e1 to O1',2)
   grid on
   ylabel ('e_O1(deg)')
   xlabel('time (sec)')
```

```
   subplot(3,1,2), plot(e2_O2(:,1), e2_O2(:,2)*180/pi,'g-')
   %axis([0 tf -30 30])
   legend('e2 to O2',2)
   grid on
   ylabel ('e_O2(deg)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(e3_O3(:,1), e3_O3(:,2)*180/pi,'b-')
   %axis([0 tf -30 30])
   legend('e3 to O3',2)
   grid on
   ylabel ('e_O3(deg)')
   xlabel('time (sec)')
end

if endplots(1,3) == 1  % Plots Roll, Pitch and Yaw from proper alignment
   figure(3)
   clf
   hold on
   subplot(3,1,1), plot(WindM(:,1), WindM(:,2),'r-')
   %axis([0 tf -0.05 0.05])
   legend('Roll',2)
   grid on
   ylabel ('Roll (deg)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(WindM(:,1), WindM(:,3),'g-')
   %axis([0 tf -0.05 0.05])
   legend('Pitch',2)
   grid on
   ylabel ('Pitch (deg)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(WindM(:,1), WindM(:,4),'b-')
   %axis([0 tf -0.05 0.05])
   legend('Yaw',2)
   grid on
   ylabel ('Yaw (deg)')
   xlabel('time (sec)')
end

if endplots(1,4) == 1  % Plots spacecraft body rates
   figure(4)
   clf
   hold on
   subplot(3,1,1), plot(OmegaM2(:,1), OmegaM2(:,2),'r-')
   axis([0 tf -0.05 0.05])
```

```
   legend('Omega1',2)
   grid on
   ylabel ('Omega1(rad/sec)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(OmegaM2(:,1), OmegaM2(:,3),'g-')
   axis([0 tf -0.05 0.05])
   legend('Omega2',2)
   grid on
   ylabel ('Omega2(rad/sec)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(OmegaM2(:,1), OmegaM2(:,4),'b-')
   axis([0 tf -0.05 0.05])
   legend('Omega3',2)
   grid on
   ylabel ('Omega3(rad/sec)')
   xlabel('time (sec)')
end

if endplots(1,5) == 1 % Plots Gravity Gradient Torques
   figure(5)
   clf
   hold on
   subplot(3,1,1), plot(T_ggM(:,1), T_ggM(:,2),'r-')
   legend('GG Torque1',2)
   grid on
   ylabel ('Torque1(N-m)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(T_ggM(:,1), T_ggM(:,3),'g-')
   legend('GG Torque2',2)
   grid on
   ylabel ('Torque2(N-m)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(T_ggM(:,1), T_ggM(:,4),'b-')
   legend('GG Torque3',2)
   grid on
   ylabel ('Torque3(N-m)')
   xlabel('time (sec)')
end

if endplots(1,6) == 1 % Plots Number of Torquer Switches
   figure(6)
   clf
   hold on
   plot(SW(:,1),SW(:,2), 'r-')
```

```
   legend ('Mag Tourquer Switches')
   ylabel ('Torquer Switches')
   xlabel('time (sec)')
   grid on
end

if endplots(1,7) == 1 % Plots Magnetic Torque
   figure(7)
   clf
   hold on
   subplot(3,1,1), plot(Mag_TorqueM(:,1), Mag_TorqueM(:,2),'r-')
   %axis([0 tf -0.01 0.01])
   legend('Magnetic Torque1',2)
   grid on
   ylabel ('Mag_Torque1(N-m)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(Mag_TorqueM(:,1), Mag_TorqueM(:,3),'g-')
   %axis([0 tf -.005 .005])
   legend('Magnetic Torque2',2)
   grid on
   ylabel ('Mag_Torque2(N-m)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(Mag_TorqueM(:,1), Mag_TorqueM(:,4),'b-')
   %axis([0 tf -.01 .01])
   legend('Magnetic Torque3',2)
   grid on
   ylabel ('Mag_Torque3(N-m)')
   xlabel('time (sec)')
end

if endplots(1,8) == 1 % Plot left over from EXIST
   figure(8)
   clf
   hold on
   subplot(2,1,1), plot(SubVectorM(:,1), SubVectorM(:,2)*180/pi,'r-')
   legend('Gamma',2)
   grid on
   ylabel ('Gamma(deg)')
   xlabel('time (sec)')
   subplot(2,1,2), plot(SubVectorM(:,1), SubVectorM(:,6),'k-')
   legend('Flux Density Mag (alt method)',2)
   grid on
   ylabel ('Flux Density(Tesla)')
   xlabel('time (sec)')
```

```
end

if endplots(1,9) == 1 % Plots Magnetic Flux Density
   figure(9)
   clf
   hold on
   subplot(3,1,1), plot(SubVectorM(:,1), SubVectorM(:,3),'r-')
   legend('FluxComp1(G frame)',2)
   grid on
   ylabel ('FluxDensity1(Tesla)')
   xlabel('time (sec)')
   subplot(3,1,2), plot(SubVectorM(:,1), SubVectorM(:,4),'g-')
   legend('FluxComp2(G frame)',2)
   grid on
   ylabel ('FluxDensity2(Tesla)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(SubVectorM(:,1), SubVectorM(:,5),'b-')
   legend('FluxComp3(G frame)',2)
   grid on
   ylabel ('FluxDensity3(Tesla)')
   xlabel('time (sec)')
end

if endplots(1,10) == 1 % Plots Magnetic Dipole Moment
   figure(10)
   clf
   hold on
   subplot(3,1,1), plot( MagDipoleM(:,1),  MagDipoleM(:,2),'r-')
   %axis([0 tf -130 130])
   legend('MagDipole1',1)
   grid on
   xlabel('time (sec)')
   subplot(3,1,2), plot(MagDipoleM(:,1),  MagDipoleM(:,3),'g-')
   %axis([0 tf -130 130])
   legend('MagDipole2',1)
   grid on
   ylabel ('Magnetic Dipole (A-m^2)')
   xlabel('time (sec)')
   subplot(3,1,3), plot(MagDipoleM(:,1),  MagDipoleM(:,4),'b-')
   %axis([0 tf -130 130])
   legend('MagDipole3',1)
   grid on
   xlabel('time (sec)')
end
```

```
if endplots(1,11) == 1 % Plots Attitude Quaternion
   figure(11)
   clf
   hold on
   subplot(4,1,1), plot( QuatM(:,1),  QuatM(:,2),'r-')
   legend('q1',2)
   grid on
   ylabel ('q1 (rad)')
   xlabel('time (sec)')
   subplot(4,1,2), plot(QuatM(:,1),  QuatM(:,3),'g-')
   legend('q2',2)
   grid on
   ylabel ('q2 (rad)')
   xlabel('time (sec)')
   subplot(4,1,3), plot(QuatM(:,1),  QuatM(:,4),'b-')
   legend('q3',2)
   grid on
   ylabel ('q3 (rad)')
   xlabel('time (sec)')
   subplot(4,1,4), plot(QuatM(:,1),  QuatM(:,5),'k-')
   legend('q4',2)
   grid on
   ylabel ('q4 (rad)')
   xlabel('time (sec)')
end

if endplots(1,12) == 1 % Plots Spacecraft Body Rate Error
   figure(12)
   clf
   hold on
   subplot(4,1,1), plot( QuatM(:,1),  OEr(:,1),'r-')
%   axis([0 tf -.005 .005])
   legend('Omega Error 1 (rad/sec)',1)
   grid on
   xlabel('time (sec)')
   subplot(4,1,2), plot(QuatM(:,1),  OEr(:,2),'g-')
%   axis([0 tf -.005 .005])
   legend('Omega Error 2 (rad/sec)',1)
   grid on
   xlabel('time (sec)')
   ylabel('error (radians/second)')
   subplot(4,1,3), plot(QuatM(:,1),  OEr(:,3),'b-')
%   axis([0 tf -.005 .005])
```

```
   legend('Omega Error 3 (rad/sec)',1)
   grid on
   xlabel('time (sec)')
end

if endplots(1,13) == 1 % Plots Attitude Control Mode
   figure(13)
   clf
   hold on
   plot(ModeM(:,1),ModeM(:,2), 'r-')
   legend ('Alignment Mode')
   ylabel ('Mode')
   xlabel('time (sec)')
   grid on
end
```

Appendix B [ODE45 Subroutine]

```
% Subroutine ExistSub2 sets up the differential equations that ODE 45 needs to solve
%  numerically.  It also calculates the Gravity Gradient Torques, calls the subroutine
%  MagField in order to calculate the flux density vector due to the Earth's magnetic
%  field, and calculates a torque available for reaction wheel desaturation. The progress
%  as of 03 February 2003 includes all magnetic formulation.

% Both the kinematic and the dynamic equations are returned in the form or a
%  10 x 1 column vector containing the solutions to the couple differential equations
%  Below is the order/scheme that they are returned in the following vector
% |Omega 1 |
% |Omega 2 |
% |Omega 3 |
% |Quat 1  |
% |Quat 2  |
% |Quat 3  |
% |Quat 4  |

function deOmQu = ExistSub2(time,State)
% ********** MAKE THE NECESSARY VARIABLES GLOBAL***************
global I Iinv incline eta_not eta rE CapOmega_not T_GG lat long
global gamma_not gamma_dot R FluxVector gamma T_Mag
global M_CmdStar lambda_not lambda_dot a Torque_Mag

% The next statement ensures that deOmQu returns a column vector
deOmQu = zeros(7,1);

% (1)  Calculate "unit" radius vector for each initial time using the IC passed to ODE45
eta = eta_not + CapOmega_not*time;      % Calculates current spacecraft position wrt
                                        %  ascending node
lambda = lambda_not + lambda_dot*time;  % Calculates current position of the longitude
                                          %  of the ascending node

% Calculates position vector using equation from Chobotov
rE(1,1) = cos(lambda)*cos(eta)-sin(lambda)*cos(incline)*sin(eta);
rE(2,1) = sin(lambda)*cos(eta)+cos(lambda)*cos(incline)*sin(eta);
rE(3,1) = sin(incline)*sin(eta);

% (2)  Calculate DCM for each new attitude to transform E1-E2-E3 axes to e1-e2-e3 axes
% Uses Equation (5) from Thesis
a(1,1) = (State(4))^2 - (State(5))^2 - (State(6))^2 + (State(7))^2;
a(1,2) = 2*(State(4)*State(5)+State(6)*State(7));
```

```
a(1,3) = 2*(State(4)*State(6)-State(5)*State(7));
a(2,1) = 2*(State(4)*State(5)-State(6)*State(7));
a(2,2) = -(State(4)^2) + State(5)^2 - (State(6)^2) + State(7)^2;
a(2,3) = 2*(State(4)*State(7)+State(5)*State(6));
a(3,1) = 2*(State(4)*State(6)+State(5)*State(7));
a(3,2) = 2*(-(State(4)*State(7))+State(5)*State(6));
a(3,3) = -(State(4))^2 - (State(5))^2 + (State(6))^2 + (State(7))^2;

% (3) Orbit unit radius vector expressed in e1-e2-e3 axes
re = a*rE;

% (4) re Telda matrix for matrix algebra
% Uses Equation (53) from Thesis
reTel(1,1) = 0;       reTel(1,2) = -re(3,1); reTel(1,3) = re(2,1);
reTel(2,1) = re(3,1); reTel(2,2) = 0;        reTel(2,3) = -re(1,1);
reTel(3,1) = -re(2,1); reTel(3,2) = re(1,1); reTel(3,3) = 0;

% (5) Gravity Gradient torques in e1-e2-e3 axes
% Uses Equation (51) from Thesis
T_GG = 3*(CapOmega_not^2)*reTel*I*re;

% (6) Calculate the Magnetic Flux Density Vector
% Calculates current magnetic reference angle used to relate ECI & ECEF frames
gamma = gamma_not + gamma_dot*time;

% called to calculate the flux density vector (FluxVector) in body axes
[FluxVector] = magfd(2012.5,2,R,90-lat,long,a);

% Write the Spacecraft State equations in terms of the States
% The states are simply:
% Omega 1     = State 1, Omega 2     = State 2, Omega 3     = State 3,
% Quaternion 1 = State 4, Quaternion 2 = State 5, Quaternion 3 = State 6,
% Quaternion 4 = State 7

OmegaM(1,1) = 0;          OmegaM(1,2) = -State(3); OmegaM(1,3) = State(2);
OmegaM(2,1) = State(3);  OmegaM(2,2) = 0;          OmegaM(2,3) = -State(1);
OmegaM(3,1) = -State(2); OmegaM(3,2) = State(1);  OmegaM(3,3) = 0;

OmegaV(1,1) = State(1); OmegaV(2,1) = State(2); OmegaV(3,1) = State(3);

% Step #11: CALCULATE THE MAGNETIC TORQUE VECTOR
% Uses Equation (7) from Thesis
Torque_Mag = cross(M_CmdStar,FluxVector);
```

```
T_Mag = Torque_Mag;

% (8) First 3 state equations: Omega
deOmQu = (Iinv)*(T_GG + T_Mag - OmegaM*(I*OmegaV));

% Write the QuatDot equations in terms of the States
deOmQu(4) = 0.5*(0*State(4)+State(3)*State(5)-State(2)*State(6)+State(1)*State(7));
deOmQu(5) = 0.5*(-State(3)*State(4)+0*State(5)+State(1)*State(6)+State(2)*State(7));
deOmQu(6) = 0.5*(State(2)*State(4)-State(1)*State(5)+0*State(6)+State(3)*State(7));
deOmQu(7) = 0.5*(-State(1)*State(4)-State(2)*State(5)-State(3)*State(6)+0*State(7));
```

Appendix C [IGRF-10 Model]

```
function  J=magfd(DATE,ITYPE,ALT,COLAT,ELONG,a)
global SubVector
%  MAGFD [Last Updated 08 April 2011]
%  Function to compute Earths magnetic field
%  and components: X,Y,Z,T for a given latitude
%  and longitude, date and altitude.
%  Uses MATLAB MAT files sh1900.mat to sh2010.mat in 5 yr
%  intervals. ONLY uses harmonic expansion to order 10 at present.
%
%  Usage: out=magfd(DATE,ITYPE,ALT,COLAT,ELONG);
%
%  DATE = date of survey (decimal years)
%  ITYPE=1 for geodetic coordinates (usual case)
%  ITYPE=2 for geocentric coordinates
%  ALT = (for ITYPE=1) altitude of survey relative to sealevel (km +ve up)
%  ALT = (for ITYPE=2) radial distance from center of earth in km
%  COLAT=90-latitude (decimal degrees)
%  ELONG=longitude of survey (decimal degrees)
%
%  Output array out contains components X,Y,Z,T in nanoteslas
%   X north component
%   Y east component
%   Z vertical component +ve down
%   T total field magnitude
%
%  ref: IAGA, Division V, Working Group VMOD,
%   The 10th generation International Geomagnetic
%   Reference Field, Geophys. J. Int, 161, 561-565, 2005.
%
% Maurice A. Tivey March 1997
% Mod Dec 1999 (add igrf2000 and y2k compliance
% Mod Nov 2000 (use up to degree 10 sh coefficients)
% Mod Apr 2005 added 2005 coeffs
% Mod Sep 2006 some clean up and info added
% Mod Jan 2010 added 2010 coefficients
% http://deeptow.whoi.edu/matlab.html
% Copyright: Maurice A. Tivey, 2005
%  Woods Hole Oceanographic Institution

if nargin < 1
 disp('DEMO MAGFD:')
```

```
help magfd
disp('Compute magnetic field at Woods Hole from Jan 1st 1900');
disp(' every five years until 2005');
disp(' Latitude 42 N, Longitude 74W')
disp('sample command:  out=magfd(1997,1,0,90-42,-74);')
for i=1:23,
  out(i,:)=magfd(-((i-1)*5+1900),1,0,90-42,-74);
end
plot(([1:23]-1)*5+1900,out(:,4),'-r+','linewidth',2);
xlabel('Year');
ylabel('Total Magnetic Field (nT)')
title('Total Magnetic Field Intensity at Woods Hole, MA')
axis tight
return
end

DGRF=[1000:5:2010];
igrfyear=2010;
igrffile='sh2010';
pl=0;
if DATE < 0, pl=1; end
DATE=abs(DATE);
% Determine year for base DGRF to use.
 if DATE < igrfyear,
  BASE=fix(DATE-DGRF(1));
  i=fix(BASE/5)+1;
  BASE=DGRF(i);
  if pl==0,
     fprintf('Using DGRF base year %f \n',BASE);
  end
  eval(['load sh',num2str(BASE)])
  % loads agh and agh41 but now need to get next epoch
   iagh=agh;iagh41=agh41;
  % figure out next epoch to load
  if BASE < 1900, % a check to get pre-1900 estimates of gauss coeffs
   eval(['load sh',num2str(BASE+25)])
  else
   eval(['load sh',num2str(DGRF(i+1))])
  end
   eagh=agh;eagh41=agh41;
   dgh=(eagh-iagh)./5;dgh41=(eagh41-iagh41)./5;
   agh=iagh;agh41=iagh41;
   clear iagh iagh41 eagh eagh41
   T = DATE - BASE;
```

```
 else
  if pl==0,
     %fprintf('Using IGRF base year %f \n',igrfyear);
  end
  eval(['load ',igrffile])   % load in igrf data file
  T    = DATE - igrfyear;
 end
% combine spherical harmonic coefficients from first 8 degrees
% with degrees 9 thru 13
 agh=[agh,agh41];
 dgh=[dgh,dgh41];
%
     D2R   = pi/180;
     R     = ALT;
     SLAT  = cos(COLAT*D2R);
     CLAT  = sin(COLAT*D2R);
     CL(1) = cos(ELONG*D2R);
     SL(1) = sin(ELONG*D2R);
     X     = 0.0;
     Y     = 0.0;
     Z     = 0.0;
     CD    = 1.0;
     SD    = 0.0;
     L     = 1;
     M     = 1;
     N     = 0;
     RE    = 6371.2; % Earth's mean radius
if ITYPE == 1  % CONVERT FROM GEODETIC TO GEOCENTRIC COORDINATES
   %A2    = 40680925.;  % squared semi major axis
   %B2    = 40408588.;  % squared semi minor axis
   % WGS84
   A2    = 40680631.59; % 6378.137^2; % squared semi major axis
   B2    = 40408299.98; % 6356.7523142^2; % squared semi minor axis
   ONE   = A2*CLAT*CLAT;
   TWO   = B2*SLAT*SLAT;
   THREE = ONE + TWO;
   FOUR  = sqrt(THREE);
   R     = sqrt(ALT*(ALT + 2.0*FOUR) + (A2*ONE + B2*TWO)/THREE);
   CD    = (ALT + FOUR)/R;
   SD    = (A2 - B2)/FOUR*SLAT*CLAT/R;
   ONE   = SLAT;
   SLAT  = SLAT*CD - CLAT*SD;
   CLAT  = CLAT*CD +  ONE*SD;
end
```

```
% if geocentric coordinates desired then only need to define the following
   RATIO = RE/R;
%
%    COMPUTATION OF SCHMIDT QUASI-NORMAL COEFFICIENTS  P & X(=Q)
%
     P(1)  = 2.0*SLAT;
     P(2)  = 2.0*CLAT;
     P(3)  = 4.5*SLAT*SLAT - 1.5;
     P(4)  = sqrt(27)*CLAT*SLAT;
     Q(1)  = -CLAT;
     Q(2)  =  SLAT;
     Q(3)  = -3.0*CLAT*SLAT;
     Q(4)  = sqrt(3)*(SLAT*SLAT - CLAT*CLAT);

NMAX=13; % Max number of harmonic degrees
NPQ=(NMAX*(NMAX+3))/2;
for K=1:NPQ,
 if N < M
   M    = 0;
   N    = N + 1;
   RR   = RATIO^(N + 2);
   FN   = N;
 end
 FM   = M;
 if K >= 5 %8,5,5
   if (M-N) == 0 %,7,6,7
      ONE  = sqrt(1.0 - 0.5/FM);
      J    = K - N - 1;
      P(K)  = (1.0 + 1.0/FM)*ONE*CLAT*P(J);
      Q(K)  = ONE*(CLAT*Q(J) + SLAT/FM*P(J));
      SL(M) = SL(M-1)*CL(1) + CL(M-1)*SL(1);
      CL(M) = CL(M-1)*CL(1) - SL(M-1)*SL(1);
   else
      ONE  = sqrt(FN*FN - FM*FM);
      TWO  = sqrt((FN - 1.0)^2 - FM*FM)/ONE;
      THREE = (2.0*FN - 1.0)/ONE;
      I    = K - N;
      J    = K - 2*N + 1;
      P(K)  = (FN + 1.0)*(THREE*SLAT/FN*P(I) - TWO/(FN - 1.0)*P(J));
      Q(K)  = THREE*(SLAT*Q(I) - CLAT/FN*P(I)) - TWO*Q(J);
   end
%
%    SYNTHESIS OF X, Y AND Z IN GEOCENTRIC COORDINATES
%
```

```
end
ONE   = (agh(L) + dgh(L)*T)*RR;

if M == 0 %10,9,10
  X    = X + ONE*Q(K);
  Z    = Z - ONE*P(K);
  L    = L + 1;
else
  TWO   = (agh(L+1) + dgh(L+1)*T)*RR;
  THREE = ONE*CL(M) + TWO*SL(M);
  X    = X + THREE*Q(K);
  Z    = Z - THREE*P(K);
  if CLAT > 0 %12,12,11
    Y = Y+(ONE*SL(M)-TWO*CL(M))*FM*P(K)/((FN + 1.0)*CLAT);
  else
    Y = Y + (ONE*SL(M) - TWO*CL(M))*Q(K)*SLAT;
  end
  L    = L + 2;
end
  M    = M + 1;
end

% Calulate latitude and longitude in radians
latr  = (90-COLAT)*pi/180;
longr = ELONG*pi/180;

% Calculate rotation matrix for LNED Equation (2) from Thesis
LNED(1,1) = -sin(latr)*cos(longr); LNED(1,2) = -sin(longr);
LNED(1,3) = -cos(latr)*cos(longr);
LNED(2,1) = -sin(latr)*sin(longr); LNED(2,2) =  cos(latr);
LNED(2,3) = -cos(latr)*sin(longr);
LNED(3,1) =  cos(latr);          LNED(3,2) = 0;          LNED(3,3) = -sin(latr);

%    CONVERSION TO COORDINATE SYSTEM SPECIFIED BY ITYPE
ONE   = X;
X    = X*CD +  Z*SD;
Z    = Z*CD - ONE*SD;

X    = X * 10^-9;
Y    = Y * 10^-9;
Z    = Z * 10^-9;
T    = sqrt(X*X + Y*Y + Z*Z);
Fixed = [X,Y,Z];       % Mag field in LNED frame
SubVector = Fixed';    % Mag field in LNED frame
```

```
Fixed = a*LNED*Fixed';  % Mag field in body frame
J = [Fixed];
%  END
```

Appendix D [IGRF-10 Coefficients]

Table 2: IGRF-10 Main Field Coefficients for 2010

-29496.5	-1585.9	4945.1	-2396.6	3026	-2707.7	1668.6	-575.4	1339.7	-2326.3
-160.5	1231.7	251.7	634.2	-536.8	912.6	809	286.4	166.6	-211.2
-357.1	164.4	89.7	-309.2	-231.1	357.2	44.7	200.3	188.9	-141.2
-118.1	-163.1	0.1	-7.7	100.9	72.8	68.6	-20.8	76	44.2
-141.4	61.5	-22.9	-66.3	13.1	3.1	-77.9	54.9	80.4	-75
-57.8	-4.7	-21.2	45.3	6.6	14	24.9	10.4	7	1.6
-27.7	4.9	-3.4	24.3	8.2	10.9	-14.5	-20	-5.7	11.9
-19.3	-17.4	11.6	16.7	10.9	7.1	-14.1	-10.8	-3.7	1.7
5.4	9.4	-20.5	3.4	11.6	-5.3	12.8	3.1	-7.2	-12.4
-7.4	-0.8	8	8.4	2.2	-8.4	-6.1	-10.1	7	-2
-6.3	2.8	0.9	-0.1	-1.1	4.7	-0.2	4.4	2.5	-7.2
-0.3	-1	2.2	-4	3.1	-2	-1	-2	-2.8	-8.3
3	-1.5	0.1	-2.1	1.7	1.6	-0.6	-0.5	-1.8	0.5
0.9	-0.8	-0.4	0.4	-2.5	1.8	-1.3	0.2	-2.1	0.8
-1.9	3.8	-1.8	-2.1	-0.2	-0.8	0.3	0.3	1	2.2
-0.7	-2.5	0.9	0.5	-0.1	0.6	0.5	0	-0.4	0.1
-0.4	0.3	0.2	-0.9	-0.8	-0.2	0	0.8	-0.2	-0.9
-0.8	0.3	0.3	0.4	1.7	-0.4	-0.6	1.1	-1.2	-0.3
-0.1	0.8	0.5	-0.2	0.1	0.4	0.5	0	0.4	0.4
-0.2	-0.3	-0.5	-0.3	-0.8	-0.2				

Table 3: IGRF-10 Secular-Variation Model for 2010 to 2015

11.4	16.7	-28.8	-11.3	-3.9	-23	2.7	-12.9	1.3	-3.9
8.6	-2.9	-2.9	-8.1	-2.1	-1.4	2	0.4	-8.9	3.2
4.4	3.6	-2.3	-0.8	-0.5	0.5	0.5	-1.5	1.5	-0.7
0.9	1.3	3.7	1.4	-0.6	-0.3	-0.3	-0.1	-0.3	-2.1
1.9	-0.4	-1.6	-0.5	-0.2	0.8	1.8	0.5	0.2	-0.1
0.6	-0.6	0.3	1.4	-0.2	0.3	-0.1	0.1	-0.8	-0.8
-0.3	0.4	0.2	-0.1	0.1	0	-0.5	0.2	0.3	0.5
-0.3	0.4	0.3	0.1	0.2	-0.1	-0.5	0.4	0.2	0.4

Appendix E [LPMT Design Code]

```
% 10-19-2009 // From Thesis Reference [2] converted into MATLAB
% This sim is used to get results for AlaskaSat Low-Power Magnetic Torquers
%  Assumes ALNICO 1

% 1/16" core with bobbin for Cubesat analysis
  d=0.001587;
  r=0.61;
  xk=135.0; % Baseline slope of M-H curve
  xms=5.7296E+05; % Saturation intensity of magnetization (a/m)
  xhc=37000; % Coercivity (a/m)
  e=7.5; % Input voltage (volts)

rch=1000.0;  % Charging Resistor (Ohms)
wo=0.0;
wo1=0.0;
iii=0;
xn=360;  % Number of Turns
pi;  % 3.14159... (radians)
rs=.75;  % Shortening Ratio (baseline=0.75)
u=12.57E-07;  % Permeability of Free Space (4*pi*1e-7)
c=0.0005;  % Discharge Capacitor (Farads), c=0.004 is baseline
xl=0.0254;  % Length of Torquer (meters)
ro=1.724E-08;  % Resistivity of Copper (Ohm-meters)
a=0.82307E-06;  % Cross sectional area of the solenoid wire (meters) -- 18 AWG
a=2.588E-07;  % Cross sectional area of the solenoid wire (meters) -- 23 AWG
a=1.281007E-07;  % Cross sectional area of the solenoid wire (meters) -- 26 AWG
%a=5.067E-08;  % Cross sectional area of the solenoid wire (meters) -- 30 AWG
%a=3.243E-08;  % Cross sectional area of the solenoid wire (meters) -- 32 AWG
%a=1.2477E-08;  % Cross sectional area of the solenoid wire (meters) -- 36 AWG
xnb=.005;  % Nb -- Ballistic Demagnetization Factor (baseline=0.005)
tmax=0.001;

t1=0.02;
t2=9.9999;
t3=9.99994;
dt1=0.02E-05; %(baseline=0.02E-05)
dt2=0.1E-03;

xmsn=-xms;      % Negative Saturation condition (Amps/meter)

x1=-xl/xn;      % Length of torquer / number of turns
```

```
x2=x1*xnb;
x3=4.0/(xn*pi*d*d*rs*u);
x4=x3/c;


x5=x3*r+x3*pi*d*xn*ro/a;

x7=.25*pi*d*d;
x8=xl*rs*x7;
x9=-xn*x7;
x10=.5/c;
x12=e/rch;
x13=-1.0/(rch*c);

run=1;
while (run)

xm=-xms;
xh=-xnb*xm;

dmdh=0;
q1=0.0;
q2=e*c;
iflg=0;
kount=0;
t=0.0;
x11=x10*q1*q1+x10*q2*q2;
x14=e*q1;


i=1;
z=1;
done=0;

while (z)
   x6=1/(1+dmdh);

   if (t<t2)
      qd1=x13*q1+x12;
      qd2=x1*xh+x2*xm;
      xhd=x6*x4*q2+x6*x5*qd2;
      if (t<t1)
         dt=dt1;
         kmax=10; % changed to this from kmax=1
```

```
   else
      dt=dt2;
      kmax=10; % changed to this from kmax=1
   end
else
   qd1=-x1*xh-x2*xm;
   qd2=x13*q2+x12;
   xhd=-x6*x4*q1-x6*x5*qd1;
   if (t<t3)
      if (iii==0)
         dt=dt1;
         kmax=1;
         iii=1;
         kount=0;
         x11=x10*q1*q1+x10*q2*q2;
         x14=e*q2;
         wo1=wo;
      else
         dt=dt1;
         kmax=10; % changed to this from kmax=1
      end
   else
      dt=dt2;
      kmax=10; % changed to this from kmax=2
   end
end

if (kount==0)
   if (t<t2)
      wo=wo1+x11-x14+e*q1-x10*q1*q1-x10*q2*q2;
      xmo=x8*xm+x9*qd2;
   else
      wo=wo1+x11-x14+e*q2-x10*q1*q1-x10*q2*q2;
      xmo=x8*xm-x9*qd1;
   end

   tout(i)=t;
   q1out(i)=q1;
   q2out(i)=q2;
   xhout(i)=xh;
   xmout(i)=xm;
   dmdhout(i)=dmdh;
   qd1out(i)=qd1;
   qd2out(i)=qd2;
```

```
    xhdout(i)=xhd;
    xmoout(i)=xmo;
    woout(i)=wo;
    vcout(i) = q2/c;
    isout(i) = qd2;
    i=i+1;

    kount=kmax;
  end

  if (t>=tmax)
    z=0;
    run=0;

    if (run) hold on
    end
    done=1;

  end

  if (done==0)
    if (iflg==0)

    qd1p=qd1;
    qd2p=qd2;
    xhdp=xhd;
    q1p=q1;
    q2p=q2;
    xhp=xh;
    q1=q1p+dt*qd1p;
    q2=q2p+dt*qd2p;
    xh=xhp+dt*xhdp;
    iflg=1;
  else
  qd1p=.5*(qd1+qd1p);
  qd2p=.5*(qd2+qd2p);
  xhdp=.5*(xhd+xhdp);
  kount=kount-1;
  t=t+dt;
  iflg=0;


q1=q1p+dt*qd1p;
q2=q2p+dt*qd2p;
```

```
xh=xhp+dt*xhdp;
 dump=xk*xh+xk*xhc;

 if (xhdp<0)
      if (xm>=dump)
         if (xmsn>=dump)
            dmdh=0.0;
            xm=xmsn;
         else
            dmdh=xk;
            xm=dump;
         end
      else
         dmdh=0.0;
      end

   elseif (xhdp>0)
      dumn=xk*xh-xk*xhc;
      if(dumn>=xm)
         if (dumn>=xms)
            dmdh=0.0;
            xm=xms;
         else
            dmdh=xk;
            xm=dumn;
         end
      else
         dmdh=0.0;
      end

   end
end
end

end
close all;
figure
subplot(2,1,1),plot(tout,q1out),grid,xlabel('time (seconds)'),...
   ylabel('q1 (coulombs)')
subplot(2,1,2),plot(tout,qd1out),grid,xlabel('time (seconds)'),...
   ylabel('qdot1 (amperes)')

figure
subplot(2,1,1),plot(tout,q2out),grid,xlabel('time (seconds)'),...
```

```
   ylabel('q2 (coulombs)')
subplot(2,1,2),plot(tout,qd2out),grid,xlabel('time (seconds)'),...
   ylabel('qdot2 (amperes)')

figure
subplot(2,1,1),plot(tout,vcout),grid,xlabel('time (seconds)'),...
   ylabel('Vc (volts)')
subplot(2,1,2),plot(tout,isout),grid,xlabel('time (seconds)'),...
   ylabel('Is (amperes)')

I = getframe(gcf);
imwrite(I.cdata, 'Figure5.png');

figure
subplot(2,1,1),plot(tout,xhout),grid,xlabel('time (seconds)'),...
   ylabel('Have/mcs (amperes/meter)')
subplot(2,1,2),plot(tout,xmout),grid,xlabel('time (seconds)'),...
   ylabel('Mave/mcs (amperes/meter)')

I = getframe(gcf);
imwrite(I.cdata, 'Figure20.png');


figure
plot(tout,xmoout),grid,xlabel('time (seconds)'),...
   ylabel('M (ampere-meters^2)')

I = getframe(gcf);
imwrite(I.cdata, 'Figure7.png');
end
```

Appendix F [Rate Calculation with Magnetometer Noise]

```
% Magnetometer rate calculartions in presence of noise using the data file
%   field_dat.daq provided by Jesse Frey.
% Data was recorded at 10 Hz which is the proposed collection rate for ARC

% Written by Maj Donald B. Mentch //  Current as of 24 April 2011

close all;
clear;
clc;

k = 1;  % Used as an ndex to record the body rate data

[data,time] = daqread('field_dat.daq');  % Reads in the magnetometer data

x=data(:,1);    % Seperates the data file into a vector for each axis
y=data(:,2);
z=data(:,3);

for j = 1:31;
   td(j)=time(j); % The time vector // 0 to 3 seconds in 0.1 sec interval
end;

i = 0;
while i < 36766    % Stops program at the end of the data
                   %  in this case about 1 hour
   for j = 1:31
      xd(j)=x(i+j);  % Collects the data into 3 second blocks used to
                     %  curve fit the data
      if real(xd(j)) ~= xd(j);  % The data file contained NaN entries
         xd(j) = x(i+j-1);      %  these lines remove those entries
         x(i+j) = x(i+j-1);     %  in the real code non-number should
      end;                      %  be rejected
      yd(j)=y(i+j);
      if real(yd(j)) ~= yd(j);
         yd(j) = y(i+j-1);
         y(i+j) = y(i+j-1);
      end;
      zd(j)=z(i+j);
      if real(zd(j)) ~= zd(j);
         zd(j) = z(i+j-1);
         z(i+j) = z(i+j-1);
```

```
    end;
  end;

[xp,xs]=polyfit(td,xd,1);  % Determines polynomial coefficients (1st order)
[yp,ys]=polyfit(td,yd,1);
[zp,zs]=polyfit(td,zd,1);
i=i+31;

B=data(i,:);              % The last magnetometer reading
B_dot(1)=xp(1);           % Flux field derivative
B_dot(2)=yp(1);
B_dot(3)=zp(1);

omega(k,:) = (cross(B_dot,B)/dot(B,B))*180/3.14*10; % Equation (48) from thesis
        % converted into degrees and multiplied by 10 seconds to show
        % total movement during a complete charge / discharge cycle

k=k+1;     % Increases the data collection index

i=i+31;    % Jumps to the next 3 second block of data

end;

figure(1)  % Plots results for each axis
clf
hold on
subplot(3,1,1), plot(omega(:,1),'r-')
legend('Omega1',2)
grid on
ylabel ('Omega1(deg/10 sec)')
xlabel('time (sec)')
subplot(3,1,2), plot(omega(:,2),'g-')
legend('Omega2',2)
grid on
ylabel ('Omega2(deg/10 sec)')
xlabel('time (sec)')
subplot(3,1,3), plot(omega(:,2),'b-')
legend('Omega3',2)
grid on
ylabel ('Omega3(deg/10 sec)')
xlabel('time (sec)')
```