

ATTITUDE CONTROL SYSTEM DESIGN FOR ION, THE ILLINOIS
OBSERVING NANOSATELLITE

BY

BRYAN SCOTT GREGORY

B.S., Marquette University, 2001

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
1 INTRODUCTION	1
2 CUBESAT OVERVIEW	2
2.1 History of the CubeSat Program.....	2
2.2 The History of ION	3
2.3 ION's Dimensions	3
2.4 ION's Payload	4
3 THE ATTITUDE CONTROL HARDWARE.....	6
3.1 Torque Coils	6
3.1.1 Overview of torque coil theory	6
3.1.2 Sizing the torque coils.....	8
3.2 The Magnetometer.....	9
3.3 The Flight Computer	10
4 ATTITUDE CONTROL BASICS.....	12
4.1 Coordinate Systems	12
4.1.1 The geocentric celestial inertial coordinate system	13
4.1.2 The geocentric fixed coordinate system	13
4.1.3 The satellite fixed body coordinate system.....	15
4.1.4 The satellite orbital reference coordinate system.....	16
4.2 Attitude Systems.....	17
4.2.1 The attitude or rotation matrix	18
4.2.2 Quaternions	20
4.2.3 Euler angles.....	22
4.3 Angular Velocity	23
4.4 Angular Momentum	24
4.5 Modeling the Satellite Dynamics	25
4.5.1 Rotational dynamics.....	26
4.5.2 Rotational kinematics.....	26
5 ATTITUDE CONTROL ALGORITHMS	28
5.1 Position Determination	28
5.1.1 The orbital elements.....	29
5.1.2 Kepler's second law	31
5.1.3 Orbit propagation algorithm	32
5.2 The Geomagnetic Field	33
5.2.1 Generating the magnetic field.....	34
5.2.2 Magnetic field algorithm.....	36
5.3 Attitude Estimation.....	36
5.3.1 Extended Kalman filter	38

5.3.2	ION's attitude estimation via the extended Kalman filter	40
5.4	Convergence Issues	44
5.5	Extended Kalman Filter Algorithm	45
5.6	The Linear Quadratic Regulator Control Law.....	45
5.6.1	Linear quadratic regulator theory.....	46
5.6.1.1	Asymptotic periodic linear quadratic regulator	47
5.6.2	ION's asymptotic periodic linear quadratic regulator design.....	48
5.6.2.1	Integral control.....	50
5.6.2.2	Calculating the C matrix	51
5.6.2.3	Control algorithms	51
5.6.2.4	Results.....	52
6	FUTURE WORK AND OPEN ISSUES	55
6.1	Extended Kalman Filter Convergence Issues.....	55
6.2	Implementation Issues	55
6.3	Detumbling Mode.....	56
6.4	Simulation With Real Magnetometer Data	56
6.5	Reducing the Order of Harmonics in the Magnetic Field Model.....	57
6.6	Flight Torque Coils.....	57
6.7	Testing	57
	APPENDIX A : TORQUE COIL IMPLEMENTATION	59
A.1	Torque Coil Circuitry	59
A.2	Calculating Inputs for Torque Coil Circuitry	61
	APPENDIX B : MAGNETOMETER IMPLEMENTATION.....	63
B.1	Magnetometer Circuitry	63
B.1.1	Hardware.....	63
B.1.2	Signals.....	63
B.2	Calibration	65
B.2.1	Temperature and local magnetic effects	66
B.2.2	Hard and soft metal calibration.....	66
B.3	Normal Operation Procedure.....	68
	APPENDIX C : OVERVIEW OF THE ATTITUDE CONTROL SYSTEM	70
	APPENDIX D : CONVERSION BETWEEN THE RADIUS AND VELOCITY VECTORS AND THE ORBITAL ELEMENTS	72
D.1	From the Radius and Velocity to the Orbital Elements.....	72
D.2	From the Orbital Elements to the Radius and Velocity.....	73
	APPENDIX E : CONVERSIONS BETWEEN COORDINATE SYSTEMS	75
E.1	The Attitude Matrix for Converting Between the Geocentric Celestial Inertial and Orbital Reference Coordinate Systems	75
E.2	Obtaining the Angular Velocity of the Reference Frame With Respect to the Inertial Frame.....	76

E.3 The Attitude Matrix for Converting Between the Geocentric Celestial Inertial and Geocentric Fixed Coordinate Systems	77
E.4 The Satellite Fixed Body Coordinate System	78
APPENDIX F : CONVERSIONS BETWEEN ATTITUDE REPRESENTATIONS	79
F.1 Conversion from the Quaternion Vector to the Attitude Matrix	79
F.2 Conversion from the Attitude Matrix to the Quaternion Vector	80
F.3 Conversion from the Euler Angles to the Attitude Matrix	81
F.4 Conversion from the Attitude Matrix to the Euler Angles	81
APPENDIX G : QUATERNION MATHEMATICS	83
APPENDIX H : SIMULATION OVERVIEW	85
H.1 Flow Diagram of the Simulations	85
H.2 Magnetometer Simulation	85
H.3 Modeling Disturbance Torques	87
H.4 Gravity Gradient Torque	87
H.5 Aerodynamic Torque	88
H.6 Disturbance Torque Results	92
H.7 Attitude Propagation Algorithm	93
APPENDIX I : DERIVATION OF THE LINEARIZATIONS FOR THE EXTENDED KALMAN FILTER AND LINEAR QUADRATIC REGULATOR	94
I.1 Linear Representation of the Extended Kalman Filter	94
I.2 Linear System for the Linear Quadratic Regulator	98
REFERENCES	104

LIST OF TABLES

TABLE 1: ION's Numerical Parameters	5
TABLE 2: Torque Coil Parameters	59

LIST OF FIGURES

Figure 1: Torque Coil.....	7
Figure 2: Intuitive Torque Coil Operation.....	8
Figure 3: HMC2003 Magnetometer.....	10
Figure 4: Small Integrated Data Logger (SID).....	11
Figure 5: Geocentric Celestial Inertial Coordinate System.....	14
Figure 6: Geocentric Fixed Coordinate System.....	14
Figure 7: The Fixed Body Coordinate System.....	15
Figure 8: Satellite Reference Frame Coordinate System.....	16
Figure 9: Yaw Pitch and Roll Angles.....	23
Figure 10: The Three-Dimensional Orbit.....	29
Figure 11: Circumscribed Two-Dimensional Elliptical Plane.....	31
Figure 12: Orbit Propagation Algorithm.....	33
Figure 13: The Magnetic Field Algorithm.....	36
Figure 14: Attitude Ambiguity.....	37
Figure 15: Attitude Estimation Algorithm.....	45
Figure 16: P_{ss} Calculation.....	51
Figure 17: Attitude Control Algorithm.....	52
Figure 18: Typical Steady State Yaw, Pitch, and Roll Angles for Linear Quadratic Regulator Control.....	53
Figure 19: Typical Yaw, Pitch, and Roll Angles for an Initial 90° Attitude Deviation.....	54
Figure 20: Torque Coil Schematic.....	60
Figure 21: Magnetometer Circuit Schematic.....	64
Figure 22: Signal Specifications for “Set,” “Reset,” and “S/R+”.....	64
Figure 23: Magnetometer Coordinate System.....	65
Figure 24: Attitude Control System Flow Diagram.....	71
Figure 25: The Earth’s Equatorial Plane.....	78
Figure 26: Simulation Flow Diagram.....	86
Figure 27: Attitude Propagation Algorithm.....	93

1 INTRODUCTION

Wertz defines an *attitude control system* as “both the process and the hardware by which the attitude [of a spacecraft] is controlled” [1]. This thesis consists of the attitude control system design for the CubeSat class satellite designed and built by students at the University of Illinois, which is known as ION, Illinois Observing Nanosatellite. As such, both of these aspects (process and hardware) of the attitude control system will be explored in detail.

The attitude control hardware consists of a three-axis magnetometer to sample the Earth’s magnetic field, three torque coils mounted on each of ION’s three axes, and the microprocessor upon which the algorithms are implemented. The process by which this hardware is used by the attitude control system will be explained in detail throughout the second section of the paper.

This thesis will begin with a short introduction to CubeSat satellites in general. It will then provide a short overview of ION in particular. In the second section of the thesis, the attitude control hardware will be reviewed in detail. The third section of the thesis will introduce some of the basic theory necessary to address attitude control problems. The final section will include the algorithms that are implemented in the attitude control design.

2 CUBESAT OVERVIEW

The section provides a brief overview of the CubSat program. It continues by discussing the history of the CubeSat at the University of Illinois.

2.1 History of the CubeSat Program

The construction of smaller satellites is currently a significant trend within the aerospace community. The CubeSat program is part of this trend. Traditionally, satellites have been built on a grand scale, designed over the course of many years (sometimes even up to a decade). Typical masses for these large satellites would often exceed 1000 kg, and budgets required to build them were also enormous. Due to the accelerating advances of technology, many of the parts used on these satellites were obsolete before they were even launched. However, since the 1990s, satellite designs have gotten smaller. Nanosatellites, or satellites with a mass between 1 and 10 kg, have become more common. The advantage of these small satellites is that they can be developed much faster and at considerably less expense. Their small size also presents many more launch opportunities. Nanosatellites are often launched as secondary payloads on various space missions.

The CubeSat Project started in 1999 as a collaborative effort between California Polytechnic State University and Stanford University. The project's goal was to standardize the design of these nanosatellites. The CubeSat standard specifies each satellite as a 10-cm cube of 1-kg maximum mass. The standard also provides additional guidelines for the location of a diagnostic port, a remove-before-flight pin, and deployment switches. With this standardization, a number of satellites can be launched from a single deployment unit. Thus, it is possible to drastically reduce the launch cost for any single satellite. On June 30, 2003, the first six CubeSat

class satellites were launched into space. Currently more than 30 high schools, colleges, and universities around the world are developing CubeSats. For more information on the CubeSat program, see <http://cubesat.calpoly.edu/index.html>.

2.2 The History of ION

The CubeSat Program at the University of Illinois dates back to August of 2001. The program was initiated under the guidance of Professor Gary Swenson in the Department of Electrical and Computer Engineering and Professor Victoria Coverstone in the Department of Aerospace Engineering. CubeSat is a fundamental component of ITSI: the Illinois Tiny Satellite Initiative. Since its conception, numerous students from various disciplines have contributed to the design and implementation of ION. The California Polytechnic State University has been in negotiation with the Russian launch provider, Kosmotras, to launch a number of CubeSat satellites from various universities. Kosmotras converts old Soviet SS-18 missiles into satellite launching vehicles. This is known as the Dnepr space launch system. ION is currently slated to be included with a number of CubeSat satellites to go into orbit in the fall of 2004. The projected orbit is a 700-km circular orbit at 98° inclination with a 9:30-10:30 or 22:00-23:00 local launch time. For more information on Kosmotras and the Dnepr space launch system, see <http://www.kosmotras.ru/>.

2.3 ION's Dimensions

ION is a CubeSat class satellite, which means it must abide by the requirements set down by Cal-Poly and Stanford. These requirements govern the physical dimensions, mass, and materials from which ION may be built. However, one characteristic of ION that distinguishes it from most CubeSats is that it is a two-cube design. In other words, the satellite consists of two

10-cm cubes stacked on top of each other. Thus, the physical dimensions of ION are 10 cm x 10 cm x 21.5 cm. It has a mass constraint of 2 kg.

2.4 ION's Payload

ION has several important components in its mission. It will be flying two sensors as payload. The first is a H7155 photomultiplier tube (PMT) that was manufactured by Hamamatsu. This is a sensor that is used to measure an atmospheric emission known as airglow at altitudes of approximately 90 km. This data will be collected and processed under the direction of Dr. Gary Swenson at the University of Illinois. For the PMT to acquire its data, there is a pointing requirement that the sensor should be nadir (Earth) pointing with a deviation angle of less than 10° . The pitch rate must be under $0.12^\circ/\text{s}$ so that the angle deviation between scans is negligible. The sampling rate is a function of the altitude, but it will never be more than 7 s per scan. Thus, only small changes in pitch can be made every 7 s. These are the most stringent requirements for the attitude control system. The secondary sensor is a complementary metal-oxide semiconductor (CMOS) camera, the pb330 from Photobit. It will be used to take snapshots of the Earth.

Besides the sensor payloads, ION will also be space-qualifying hardware for a number of its primary components. ION's microprocessor is known as SID, the Small Integrated Data-logger. This is a single board computer designed around the Hitachi 32-bit RISC SH7045F microcontroller. SID is credit card sized and runs at 15 MHz on less than 0.5 W. SID is a prototype designed by Tether Applications. The other hardware being space-qualified by ION is an electric propulsion system. This is a custom system that was designed and built by Alameda Applied Sciences Corporation in cooperation with students at the University of Illinois. The propulsion system consists of a power-processing unit that controls four vacuum arc micro-

thrusters. These thrusters are small, pulsed electric thrusters, designed for attitude control of small satellites. It is important to note that these thrusters represent a new technology with exciting possibilities. However, since this technology is untested, the attitude control system developed in this paper does not make use of the propulsion system as an actuation device. Only the torque coils are used for attitude control. See TABLE 1 for a table of some of ION's important parameters relating to the attitude control system. For more information regarding ION, see <http://courses.ece.uiuc.edu/cubesat>.

TABLE 1: ION's Numerical Parameters

Orbit Properties	
Altitude	700km
Eccentricity	≈ 0
Inclination	98°
Satellite Dimensions	
X	10 cm
Y	10 cm
Z	21.5 cm
Deviation From Centroid to Center of Mass	
X	5.498 mm
Y	-1.079 mm
Z	4.760 mm
Moments of Inertia	
I_x	7.380 g-m ²
I_y	7.475 g-m ²
I_z	2.155 g-m ²
I_{xy}	-0.03156 g-m ²
I_{xz}	-0.09591 g-m ²
I_{yz}	-0.03867 g-m ²

3 THE ATTITUDE CONTROL HARDWARE

There are three main hardware components that are used in ION's attitude control implementation: the magnetometer, the torque coils, and the flight computer. The first two are concerned with magnetic fields. The first component, the magnetometer, is used to measure the Earth's magnetic field. The second component, the torque coils, is the actuator that changes the orientation of the satellite in space. This is done by generating a magnetic field, which in turn interacts with the naturally occurring geomagnetic field of the Earth. The final hardware component is the flight computer upon which the algorithms and data processing are carried out.

3.1 Torque Coils

It is a well-known fact that running a current through a wire loop will generate a magnetic field. It is also well known that magnetic fields interact to generate torque. These are the governing principles behind all modern motors. ION utilizes these principles in its attitude control system through the use of torque coils or magnetorquers. These torque coils consist of wire wound inside the satellite to form a large coil, as seen in Figure 1. This coil is used to generate a magnetic field, which interacts with the geomagnetic field to change the orientation of the satellite. For details on the specifics of the torque coils and their required circuitry, see APPENDIX A. The development of the remainder of this section can be found in any standard text on electromagnetism, such as [2].

3.1.1 Overview of torque coil theory

A current flowing through the torque coils generates a magnetic dipole moment, according to the following equation:

$$\mathbf{m} = NIA\mathbf{a}_n \quad (1)$$



Figure 1: Torque Coil

The parameter A is the cross sectional area of the coil, N is the number of turns of the coil, and I is the current running through the coil. The direction of the dipole moment is normal to the coil, and is determined by the right-hand rule according to the direction the current is moving in the coil. This is illustrated in Figure 2. The magnetic field at the geometric center of this rectangle, \mathbf{b}_m , with length a and width b is given by the following equation:

$$\mathbf{b}_m = \frac{2\mu_0 NI \sqrt{(a^2 + b^2)}}{\pi A} \mathbf{a}_n \quad (2)$$

The parameter μ_0 is known as the permeability of free space, which is $4\pi \times 10^{-7} \text{H/m}$. The interaction of this magnetic dipole moment with the Earth's magnetic field generates a torque, \mathbf{t}_m , according to the following equation:

$$\mathbf{t}_m = \mathbf{m} \times \mathbf{b}_e \quad (3)$$

The parameter \mathbf{m} is as defined in Equation (1), and \mathbf{b}_e is the earth’s magnetic field at a particular location.

For an intuitive understanding behind generating magnetic torques, see Figure 2. As can be seen in this figure, the torque generated in this magnetic field tends to align the magnetic dipole moment with the geomagnetic field. Note that there is no generated torque if the magnetic dipole moment is already aligned with the geomagnetic field.

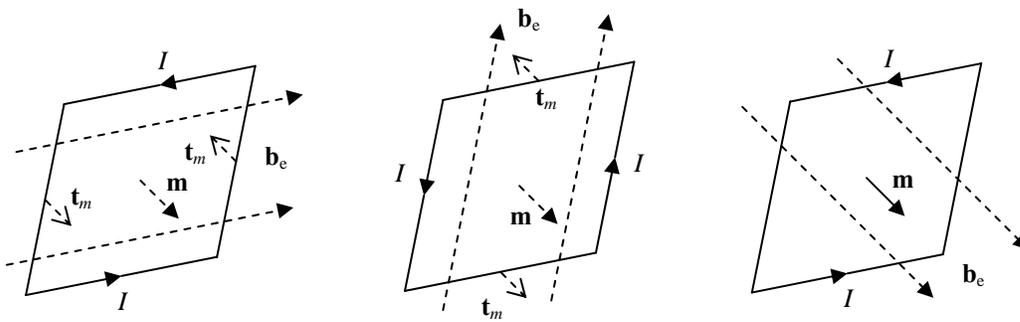


Figure 2: Intuitive Torque Coil Operation

3.1.2 Sizing the torque coils

According to Equations (1) and (3), the generated torque is proportional to the area of the coil. So, ideally, the coils should be as large as possible. However, the physical dimensions of the satellite and the locations of the various components within the satellite limit the area of the coils. Nonetheless, the coils are sized as large as possible to meet these constraints. The other constraint limiting the number of turns of the coil is that ION has only a finite amount of power available for attitude control. After analysis of the spacecraft’s power budget, it was decided that 0.1 W would be the maximum power available for each of the torque coils. As such, the coils may have as many turns as necessary as long as their total power dissipation is less than 0.1 W.

The total power dissipated by the coil is given according to the well-known equation

$$P = \frac{V_{bus}^2}{R_{total}} \quad (4)$$

The resistance R_{total} is the total resistance of the coil and the H-bridge, which is found by

$$R_{total} = R_{bridge} + R_{coil} \quad (5)$$

The resistance R_{bridge} is the resistance of the H-bridge. The resistance of the coil is given by the formula

$$R_{coil} = \frac{l}{\sigma S} \quad (6)$$

The parameter σ is the conductivity of the wire, l is the length of the wire, and S is the wire's cross-sectional area. Finally, the length of the coil is related to the dimensions of the coil according to simple geometry:

$$l = 2N(a + b) \quad (7)$$

From these equations, and the given power constraint of 0.1W for each of the torque coils, it is possible to determine the number of turns in each of the coils. Combining Equations (4) through (7), we find the total number of turns as

$$N = \frac{\sigma S}{2(a + b)} \left(\frac{V_{bus}^2}{P} - R_{bridge} \right) \quad (8)$$

3.2 The Magnetometer

The second major component of the attitude control system is the magnetometer. This is a sensor that measures magnetic fields. ION employs its magnetometer to sample the

geomagnetic field. These field measurements are then used by the attitude control system to obtain estimates of ION's orientation in space. Although it may be obvious, it is essential that the magnetometer should not be sampled while operating the torque coils. Otherwise, the magnetic field created by the torque coils will skew the measurement of the geomagnetic field. The particular magnetometer flown aboard ION is the HMC2003, which is manufactured by Honeywell. See Figure 3 for a picture of this magnetometer. As can be seen, this model is a solid-state circuit mounted on a 20-pin hybrid DIP package. This magnetometer is a three-axis magnetic sensor, which is capable of independently measuring three orthogonal components of the magnetic field using magnetoresistive transducers. This sensor can detect fields up to plus or minus 2 gauss, with an accuracy of 40 μ gauss. Detailed information regarding the implementation of this magnetometer may be found in APPENDIX B.



Figure 3: HMC2003 Magnetometer

3.3 The Flight Computer

The flight computer is a commercial, off-the-shelf part from Tether Applications (TAI), Chula Vista, CA. They have developed the Small Integrated Data-logger, known as SID, with an eye towards its use aboard small satellites. See Figure 4 for a photograph of SID. ION will be

the first satellite to fly SID into space, and will be providing TAI with feedback on SID's performance and the development experience.



Figure 4: Small Integrated Data Logger (SID)

SID is based on the Hitachi SH7045 microprocessor, a member of the SH-2 processor family. SID is only 10 x 55 x 85 mm and weighs a meager 35 grams. The microprocessor is combined with many peripherals on a very small, heavily integrated board. In addition to its 256 kB of onboard FLASH and 4 kB of onboard RAM, SID has 1024 kB external RAM, and an 8 MB serial FLASH. Because SID has been custom designed with space applications in mind, it has a variety of extra features included in its computational ability including high-power switches, an over current detector, clocks, temperature sensors, 22 analog inputs, 4 TTL serial interfaces, 24 general purpose digital I/O pins available and 28 output only pins available. As can be seen, SID is a robust computing system with many extra features for increased computational power and efficiency. For more details on SID, see http://www.nts.gov/events/symp_rec/proceedings/authors/carroll.pdf.

4 ATTITUDE CONTROL BASICS

Before the attitude control system can utilize any of the hardware to control the satellite, it is necessary to introduce a number of standard definitions necessary to solve attitude control problems. This section will lay the groundwork upon which the rest of the thesis will be built. First, the fundamental coordinate systems will be defined. Next, the intuitive concept of attitude will be formalized. Several different attitude schemes will be introduced. The concept of angular velocity will also be introduced. The fundamental equations governing the satellite dynamics will be presented. Finally, the necessary disturbance torques which influence satellite motion in the space environment will be introduced and mathematical models will be presented.

4.1 Coordinate Systems

A coordinate system, or reference frame, is simply a set of three mutually orthogonal basis vectors defining a grid in three-dimensional space. Basis vectors for the coordinate system A are denoted as \mathbf{x}_A , \mathbf{y}_A , and \mathbf{z}_A , respectively. All coordinate systems which will be introduced are both right handed and orthonormal. Thus, the following equations hold:

$$\mathbf{v}_A \bullet \mathbf{w}_A = \begin{cases} 1, & \mathbf{v} = \mathbf{w} \\ 0, & \mathbf{v} \neq \mathbf{w} \end{cases} \forall \mathbf{v}, \mathbf{w} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\} \quad (9)$$

$$\mathbf{x}_A \times \mathbf{y}_A = \mathbf{z}_A \quad (10)$$

$$\mathbf{y}_A \times \mathbf{z}_A = \mathbf{x}_A \quad (11)$$

$$\mathbf{z}_A \times \mathbf{x}_A = \mathbf{y}_A \quad (12)$$

In this thesis, the bases for various coordinate systems are defined according to various terrestrial or celestial objects. Some problems are more conveniently addressed and solved in

different coordinate systems. Thus, the following group of coordinate systems will be introduced to address these problems more conveniently.

4.1.1 The geocentric celestial inertial coordinate system

The first coordinate system which will be introduced is the Geocentric Celestial Inertial Coordinate System, which is denoted by the letter I . This system is located at the Earth's center of mass. In this system, \mathbf{x}_I is taken as the unit vector in the direction of the first point in Aries, which incidentally is the line common to both the earth's equatorial plane and ecliptic planes. The vector \mathbf{z}_I is typically chosen to be the line passing through the Earth's center and the North Pole. The vector \mathbf{y}_I is chosen to complete the right-handed system. It is important to note that this system does not take into account the rotation of the earth. Rather, the Earth rotates about the vector \mathbf{z}_I . Also, axes are defined relative to the earth's position at the chosen epoch time. The epoch time that is used for ION's attitude control system is J2000, which means t_0 is January 1st, 2000 at 12:00:00 noon. See Figure 5 for a picture of the Geocentric Celestial Inertial Coordinate System.

4.1.2 The geocentric fixed coordinate system

The second coordinate system necessary for attitude control problems is known as the Geocentric Fixed coordinate system, which is denoted by the letter E . This system is shown in Figure 6. Like the Geocentric Inertial Coordinate System, this reference frame has its origin at the Earth's center of mass. But, this coordinate system is fixed to the Earth, and this rotates in time. The vector \mathbf{x}_E is defined as the unit vector in the equatorial plane from the Earth's center to the Prime Meridian. The vector \mathbf{z}_E is the unit vector from the center of the Earth to the

geographic North Pole. And, y_E is defined to make the system a right-handed system. As such, it points from the Earth's center to the 90th longitude line.

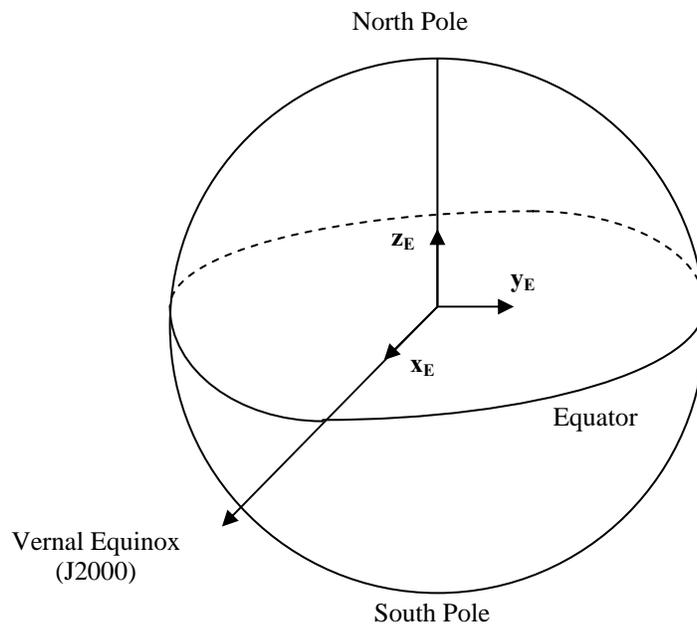


Figure 5: Geocentric Celestial Inertial Coordinate System

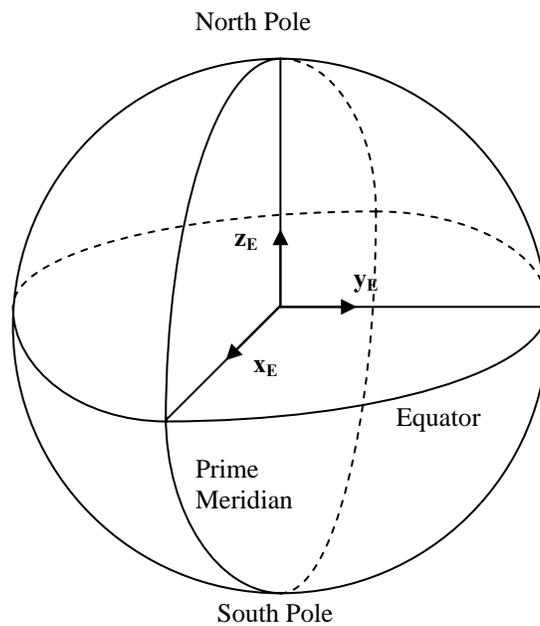


Figure 6: Geocentric Fixed Coordinate System

4.1.3 The satellite fixed body coordinate system

The third important coordinate system is Satellite Fixed Body coordinate system, which is denoted with the letter B . As its name suggests, this reference frame is defined in relation to the satellite itself. The reference frame is geographically located at the center of mass of the satellite as found in Figure 7. The unit vectors that comprise the axis for this system are denoted \mathbf{x}_B , \mathbf{y}_B , and \mathbf{z}_B , respectively. The first two are located in the plane parallel to the small face of the satellite. The final vector, \mathbf{z}_B , is located parallel to the long faces of the satellite.

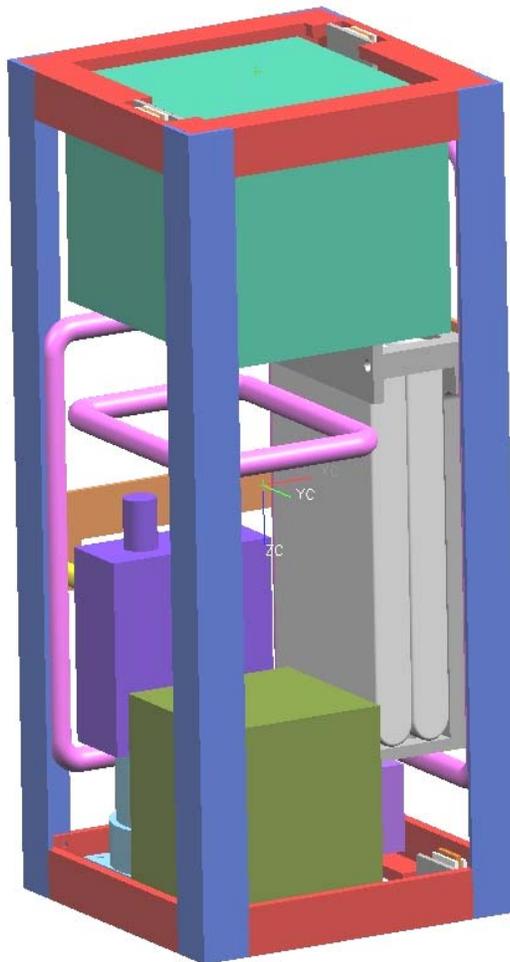


Figure 7: The Fixed Body Coordinate System

4.1.4 The satellite orbital reference coordinate system

The final coordinate system is the Satellite Orbital Reference coordinate system, which is denoted by the letter R . It is located at the satellite's center of mass. This coordinate system defines the nominal orientation of the satellite. Thus, the vector \mathbf{z}_R is the unit vector which points from the satellite's center of mass to the Earth's center of mass. The vector \mathbf{x}_R is defined as the unit vector in the direction of the component of the velocity vector that is orthogonal to the radius vector. This vector is in the orbital plane, and is in the same general direction as the velocity vector. However, it is not exactly in the direction of the velocity vector, since the radius is not orthogonal to the radius vector, except at isolated points in the orbit. Finally, \mathbf{y}_R is then defined as the cross product of \mathbf{z}_R and \mathbf{x}_R . See Figure 8 for a graphical representation of the satellite reference frame coordinate system. Note that in this figure, the vector \mathbf{y}_R is pointing into the page.

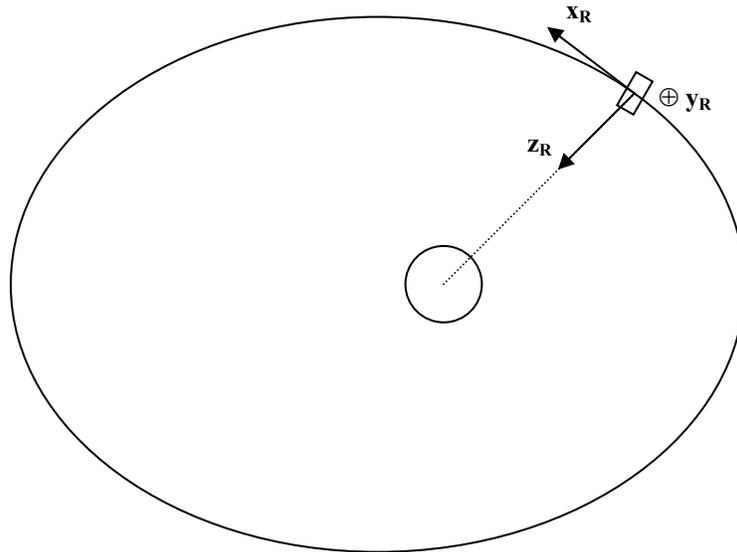


Figure 8: Satellite Reference Frame Coordinate System

As the satellite changes its position within an orbit, the orientation of the satellite reference frame changes also. It is important to note that the actual orientation of the satellite has no bearing on the reference frame coordinate system. If the satellite were to change its attitude at a particular place in the orbit, the Satellite Fixed Body coordinate system would not change. Also, it is important to note that if the satellite achieves its desired orientation within the orbital plane, both the fixed body and reference coordinate systems will be aligned with each other. That is to say, $\mathbf{x}_R \approx \mathbf{x}_B$, $\mathbf{y}_R \approx \mathbf{y}_B$, and $\mathbf{z}_R \approx \mathbf{z}_B$.

4.2 Attitude Systems

The previous sections defined coordinate systems in three-dimensional space. The next sections are concerned with the relationships between coordinate systems. Obtaining different systemic ways of describing these relationships is a means of formalizing the intuitive concept of attitude, or orientation. In other words, the basic problem of describing attitude is to specify the orientation of the one coordinate system in terms of another coordinate system.

Clearly enough, if one of the coordinate systems which is being specified is the Satellite Fixed Body coordinate system, then the attitude which is specified is the attitude of the satellite. In some contexts, it is desirable to know how the satellite's fixed body coordinate system is oriented with respect to the inertial coordinate system. In other instances, it is important to know the satellite's attitude with respect to the fixed body coordinate system. However, it is in general possible to relate any two coordinate systems to each other. As such, the theory will be developed to relate any arbitrary right-hand coordinate systems C and D , with basis vectors \mathbf{x}_C , \mathbf{y}_C , and \mathbf{z}_C ; and \mathbf{x}_D , \mathbf{y}_D , and \mathbf{z}_D . All of the results may be applied to any arbitrary coordinate system as necessary.

Describing attitude of coordinate systems relative to each other is not intuitively obvious. Therefore, many different schemes have been developed to try to solve this problem. These schemes trade off intuition with ease of implementation. Three of the more popular schemes are described below. Four good texts describing these and other attitude parameterizations are [1], [3], [4], and [5].

4.2.1 The attitude or rotation matrix

Since all of the basis vectors are unit vectors in space, it is possible to write each as a linear combination of the vectors of the other coordinate system. For example, \mathbf{x}_C may be written as

$$\mathbf{x}_C = x_x \mathbf{x}_D + x_y \mathbf{y}_D + x_z \mathbf{z}_D = \begin{bmatrix} x_x & x_y & x_z \end{bmatrix} \begin{bmatrix} \mathbf{x}_D \\ \mathbf{y}_D \\ \mathbf{z}_D \end{bmatrix} \quad (13)$$

Likewise, it is possible to write similar equations for \mathbf{y}_C and \mathbf{z}_C . Grouping these three equations together, it is possible to obtain the following matrix equation:

$$\begin{bmatrix} \mathbf{x}_C \\ \mathbf{y}_C \\ \mathbf{z}_C \end{bmatrix} = \begin{bmatrix} x_x & x_y & x_z \\ y_x & y_y & y_z \\ z_x & z_y & z_z \end{bmatrix} \begin{bmatrix} \mathbf{x}_D \\ \mathbf{y}_D \\ \mathbf{z}_D \end{bmatrix} = \mathbf{A}^{C/D} \begin{bmatrix} \mathbf{x}_D \\ \mathbf{y}_D \\ \mathbf{z}_D \end{bmatrix} \quad (14)$$

The matrix $\mathbf{A}^{C/D}$ is typically known as the attitude, or direct cosine matrix. The nine elements of $\mathbf{A}^{C/D}$ define the orientation of coordinate system C with respect to coordinate system D . However, these nine elements are a redundant description of the attitude because each of the body vectors is orthonormal, as specified in Equation (9). Therefore, there are six possible constraint equations on the elements of the direct cosine matrix, which are obtained from the various permutations of Equation (9). As such, there are three degrees of freedom necessary to

parameterize the attitude of a satellite. This is an important fact which will be used in the attitude estimation algorithms later on. Grouping all six possible combinations of these constraint equations together yields the important result that $\mathbf{A}^{C/D}$ is a unitary rotation matrix. In other words:

$$\mathbf{A}^{C/D}(\mathbf{A}^{C/D})^T = \mathbf{I} \quad (15)$$

Thus, the rotation matrix has its transpose as merely its inverse, which is always guaranteed to exist.

Clearly, any vector may be written as a weighted sum of the basis vectors of either coordinate system C or D . Thus, an arbitrary vector \mathbf{v} may be written as

$$\mathbf{v} = [\mathbf{x}_C \quad \mathbf{y}_C \quad \mathbf{z}_C] \mathbf{v}_C = [\mathbf{x}_D \quad \mathbf{y}_D \quad \mathbf{z}_D] \mathbf{v}_D \quad (16)$$

Both \mathbf{v}_C and \mathbf{v}_D are three-element column vectors. After substituting Equation (14) into Equation (16), the following result is obtained:

$$[\mathbf{x}_D \quad \mathbf{y}_D \quad \mathbf{z}_D] (\mathbf{A}^{C/D})^T \mathbf{v}_C = [\mathbf{x}_D \quad \mathbf{y}_D \quad \mathbf{z}_D] \mathbf{v}_D \quad (17)$$

Thus, it must follow that

$$\mathbf{v}_D = (\mathbf{A}^{C/D})^T \mathbf{v}_C \quad (18)$$

But, since $\mathbf{A}^{C/D}$ is unitary, when it multiplies both sides, the following result is obtained:

$$\mathbf{v}_C = \mathbf{A}^{C/D} \mathbf{v}_D \quad (19)$$

Thus, besides a method of quantifying the satellite's attitude, $\mathbf{A}^{C/D}$ can also be interpreted as a transformation between coordinate systems. The matrix $\mathbf{A}^{C/D}$ is a mapping from coordinate system D to coordinate system C , and $(\mathbf{A}^{C/D})^T$ can be interpreted as its adjoint operator. Note

that the coordinate systems C and D were defined arbitrarily. If they are relabeled oppositely, all results still follow. Thus, the following equation must hold:

$$\left(\mathbf{A}^{C/D}\right)^T = \mathbf{A}^{D/C} \quad (20)$$

One final property of attitude matrices is that it is possible to obtain a parameterization of an attitude of two coordinate systems through an intermediary coordinate system. In addition to coordinate systems C and D , defined above, let E be a coordinate system with basis vectors \mathbf{x}_E , \mathbf{y}_E , and \mathbf{z}_E . Clearly \mathbf{v} from Equation (16) may be written as a linear combination of the basis vectors of coordinate system E :

$$\mathbf{v} = [\mathbf{x}_E \quad \mathbf{y}_E \quad \mathbf{z}_E] \mathbf{v}_E \quad (21)$$

But, following the discussion leading up to Equation (19), it is possible to relate \mathbf{v}_D to \mathbf{v}_E and \mathbf{v}_C to \mathbf{v}_E through attitude matrices as follows:

$$\mathbf{v}_D = \mathbf{A}^{D/E} \mathbf{v}_E \quad (22)$$

$$\mathbf{v}_C = \mathbf{A}^{C/E} \mathbf{v}_E \quad (23)$$

Substituting Equations (19) and (22) yields the following result:

$$\mathbf{v}_C = \mathbf{A}^{C/D} \mathbf{A}^{D/E} \mathbf{v}_E \quad (24)$$

After comparing Equations (23) and (24), the final result regarding attitude matrices is obtained:

$$\mathbf{A}^{C/E} = \mathbf{A}^{C/D} \mathbf{A}^{D/E} \quad (25)$$

4.2.2 Quaternions

As mentioned above, the attitude matrix describes the attitude of the satellite using nine parameters, six of which are redundant. Other methods have been derived to describe the

orientation between coordinate systems to one another with less redundancy. One of the most useful systems is the method of quaternions, or the Euler Symmetric Parameters. This system uses four parameters to describe the attitude of the satellite.

$$\mathbf{q}^{C/D} = \begin{bmatrix} q_1^{C/D} \\ q_2^{C/D} \\ q_3^{C/D} \\ q_4^{C/D} \end{bmatrix} = \begin{bmatrix} \mathbf{q}^{C/D} \\ q_4^{C/D} \end{bmatrix} \quad (26)$$

The vector $\mathbf{q}^{C/D}$ is a quaternion. Like $\mathbf{A}^{C/D}$, it describes the attitude of the coordinate system C with respect to D . For rules regarding mathematics of quaternions, see APPENDIX G. Since four parameters are used in the quaternion, there is an extra constraint equation:

$$\|\mathbf{q}^{C/D}\| = 1 \quad (27)$$

There are a number of analogous relationships between the quaternion and the attitude matrix. First of all, since they are both means to specify attitude, it is only natural that it is possible to convert directly between them. These conversions are given in APPENDIX F. Although these parameters lack the physical insight of the direction cosine matrix, they are useful because their kinematics equations are simple to integrate numerically and because they are more computationally efficient than using the nine parameters of the attitude matrix.

A second similarity between the quaternion and the attitude matrix is if the quaternion $\mathbf{q}^{C/D}$ is known, it is easy to find $\mathbf{q}^{D/C}$. This is found by the following equation:

$$\mathbf{q}^{D/C} = \mathbf{q}^{C/D} * \quad (28)$$

The “*” in (28) is associated with the conjugation operation, as specified by Equation (161). Another important similarity between the quaternion and the attitude matrix is that it is

possible to perform conversions by multiplications. Thus, Equation (25) may be represented in quaternion form as follows:

$$\mathbf{q}^{C/E} = \mathbf{q}^{D/E} \otimes \mathbf{q}^{C/D} \quad (29)$$

The multiplication “ \otimes ” in (29) is a quaternion multiplication. Note that the order of multiplication is reversed from the matrix multiplication associated with the attitude matrices.

One final similarity between the quaternion and attitude matrix is that a vector rotation for a given attitude matrix may be expressed as a quaternion multiplication. Thus, Equation (19) may be equivalently represented as follows:

$$\mathbf{v}_C = (\mathbf{q}^{C/D})^* \otimes \begin{bmatrix} \mathbf{v}_D \\ 0 \end{bmatrix} \otimes \mathbf{q}^{C/D} \quad (30)$$

4.2.3 Euler angles

The final system used to represent attitudes is the Euler angles. The Euler angles are three angles; ψ , θ , and ϕ ; which are rotated around intermediate coordinate axes to obtain an attitude matrix. There are twelve possible sets of body-axis rotations. Thus, there are 12 possible sets of Euler angles. The angles used in this development are a “3-2-1” rotation sequence. Thus, the first rotation of angle ψ is made about the z-axis, the second of angle θ about the new y-axis, and the final rotation of angle ϕ about the new x-axis to obtain the new coordinate system.

The Euler angles are not nearly as mathematically elegant as either the attitude matrix or the quaternions. Also, they have a famous singularity point known as “gimble lock” which presents problems for representing all attitudes. However, they do have two advantages. First of all, only three elements are necessary to characterize the attitude. As such, no constraint

equation is necessary. Also, the Euler angles have an intuitive advantage that when they are used to represent the relationship between Satellite Fixed Body coordinate system to the Orbital Reference System, and the angles are small, then ϕ , θ , and ψ approximate the roll, pitch, and yaw angles of the satellite. These angles are represented in Figure 9.

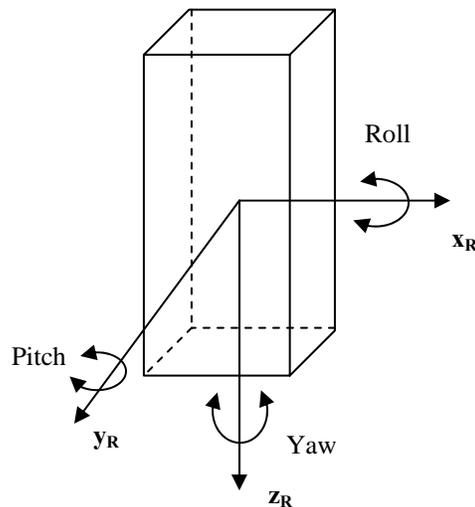


Figure 9: Yaw Pitch and Roll Angles

4.3 Angular Velocity

The previous section introduced the concept of attitude as a static relationship between coordinate systems. However, often the orientation of various coordinate systems will change with respect to each other. This is expressed in the concept of angular velocity. Angular velocity describes how quickly one coordinate system rotates around another coordinate system. Thus, $\omega^{C/D}$ is the notation to describe the angular velocity of the coordinate system C with respect to coordinate system D .

There are a few important properties regarding angular velocity. Note that if coordinate system C rotates with an angular velocity with respect to coordinate system D , then coordinate

system D will seem to rotate the opposite direction from the perspective of coordinate system C .

Thus, the following law holds:

$$\boldsymbol{\omega}^{C/D} = -\boldsymbol{\omega}^{D/C} \quad (31)$$

Another of the interesting facts regarding angular velocity is its additive property.

Angular velocities of various vector frames add according to the following rule:

$$\boldsymbol{\omega}^{C/E} = \boldsymbol{\omega}^{C/D} + \boldsymbol{\omega}^{D/E} \quad (32)$$

Thus, it is possible to obtain an angular velocity in terms of intermediate angular velocities.

The final important component of angular velocity involves taking derivatives in various reference frames. Rates of change are clearly relative to the perspective from which they are computed. Thus, derivatives of vectors are relative to the coordinate system in which they are taken. To take the derivative of an arbitrary vector \mathbf{v} in two coordinate systems C and D , the following rule applies:

$$\left\{ \frac{d\mathbf{v}}{dt} \right\}_C = \left\{ \frac{d\mathbf{v}}{dt} \right\}_D + \boldsymbol{\omega}^{D/C} \times \mathbf{v} \quad (33)$$

The brackets and subscripts in Equation (33) designate the perspective from which the derivatives are evaluated.

4.4 Angular Momentum

The concept of angular velocity is closely related to the concept of angular momentum. If ION's Satellite Fixed Body coordinate system has an angular velocity with respect to the inertial frame of $\boldsymbol{\omega}^{B/I}$ then its angular momentum \mathbf{h} is defined as

$$\mathbf{h} = \tilde{\mathbf{I}} \boldsymbol{\omega}^{B/I} \quad (34)$$

The matrix $\tilde{\mathbf{I}}$ is the 3 x 3 inertial matrix which is defined as

$$\tilde{\mathbf{I}} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \quad (35)$$

Each of the elements of $\tilde{\mathbf{I}}$ is defined by the following integrals taken in the Satellite Fixed Body coordinate system:

$$I_x = \int_B (y^2 + z^2) dm \quad I_y = \int_B (x^2 + z^2) dm \quad I_z = \int_B (x^2 + y^2) dm \quad (36)$$

$$I_{xy} = \int_B xy dm \quad I_{xz} = \int_B xz dm \quad I_{yz} = \int_B yz dm \quad (37)$$

Due to the nature of Equations (35) through (37), any real objects with some finite density and length in all directions of three-dimensional space will have a nonsingular inertial matrix $\tilde{\mathbf{I}}$. Thus, its inverse is guaranteed to exist for real objects.

4.5 Modeling the Satellite Dynamics

After introducing these concepts of attitude and angular velocities in the previous two sections, it is possible to put these concepts to use to model how a satellite's orientation changes. The satellite's system dynamics are specified by two vector differential equations. The first equation controls the evolution of the satellite's angular momentum, and thus the rate of change of the angular velocity. The second equation uses the angular velocity to describe the change of the satellite's attitude. The fundamentals of satellite dynamics are available in many standard texts. Three useful texts are [1], [3], and [4].

4.5.1 Rotational dynamics

It is a well-known fact that the derivative of the angular momentum is equal to the total applied torque. Thus, the differential equation taken in the inertial reference frame may be written as

$$\left\{ \frac{d\mathbf{h}}{dt} \right\}_I = \mathbf{t}_T \quad (38)$$

The vector \mathbf{h} is the angular momentum, and \mathbf{t}_T is the total applied torque. Euler's moment equation is obtained by following the derivative rule outlined in Equation (33) to Equation (38) as follows:

$$\left\{ \frac{d\mathbf{h}}{dt} \right\}_B = \mathbf{t}_T - \boldsymbol{\omega}^{B/I} \times \mathbf{h} \quad (39)$$

This is the dynamics equation, which governs the evolution of the satellite's angular momentum. Since the inverse of the inertial matrix is guaranteed to exist, it is possible to multiply both sides of Equation (34) by the inverse of I and substitute this result into Equation (39) to obtain an explicit formula for the evolution of the angular velocity as follows:

$$\left\{ \frac{d\boldsymbol{\omega}^{B/I}}{dt} \right\}_B = \tilde{\mathbf{I}}^{-1} \left(\mathbf{t}_T - \boldsymbol{\omega}^{B/I} \times (\tilde{\mathbf{I}} \boldsymbol{\omega}^{B/I}) \right) \quad (40)$$

4.5.2 Rotational kinematics

The kinematics equation is used to determine the rate of change of the orientation of the satellite as a function of its angular velocity. Clearly, as covered previously, there are a variety of systems available to represent orientation. Each of these various systems has its own

differential equation. Since the quaternions are the primary attitude representation used by ION, this section will only cover the kinematics of this system.

Let $\boldsymbol{\omega}^{\text{B/C}}$ be the angular velocity of the spacecraft with respect to an arbitrary reference frame C . Then, the following kinematics equation may be conveniently expressed in terms of the quaternion vector as follows:

$$\left\{ \frac{d\mathbf{q}^{\text{B/C}}}{dt} \right\}_C = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}^{\text{B/C}}) \mathbf{q}^{\text{B/C}} \quad (41)$$

The parameter $\mathbf{q}^{\text{B/C}}$ is the four-element quaternion vector, which represents the attitude of the satellite with respect to coordinate system C , and the matrix $\boldsymbol{\Omega}(\boldsymbol{\omega})$ is defined as follows:

$$\boldsymbol{\Omega} \left(\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \quad (42)$$

5 ATTITUDE CONTROL ALGORITHMS

This section of the thesis builds upon the topics introduced in the previous chapter to develop the necessary algorithms to execute the attitude control system. The first algorithm that is introduced is used to determine the position of the satellite in space. After the position of the satellite has been determined, it is possible to calculate the nominal geomagnetic field. This is then compared to the measured magnetic field in an extended Kalman filter to produce an estimate of the satellite's attitude. Finally, the linear quadratic regulator is used to actuate the torque coils and control ION's attitude.

5.1 Position Determination

The information in the following sections outlines position determination. It is readily available in many orbital mechanics textbooks. In particular, [6] provides a good development of the material. The approach that is taken in these sections assumes the two-body orbital motion. In other words, it is assumed that the satellite's orbit is completely determined through the laws of Newtonian physics, using the mass of the earth and satellite, along with its initial position and velocity vectors. Thus, perturbations in the orbit caused by the earth's obliqueness, atmospheric drag, gravitational effects from other celestial bodies, and other miscellaneous effects are completely ignored. The deviations in the orbit from these effects are quite small compared to the contribution of the earth itself. Thus, the two-body solution provides a reasonable first-order approximation, which is useful in designing the attitude control system. It should be noted that if more accurate orbit propagation techniques are necessary, there are many commercial products available that can perform orbit propagation to determine the satellites position while accounting for some of these other effects.

5.1.1 The orbital elements

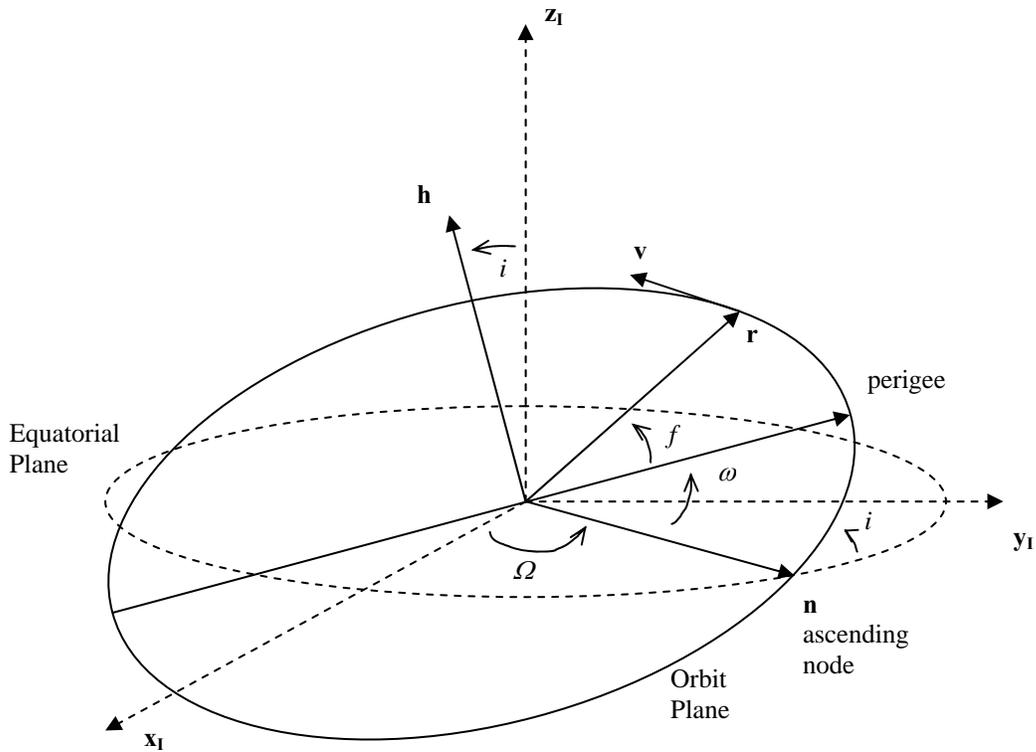


Figure 10: The Three-Dimensional Orbit

As can be seen in Figure 10, a satellite orbits the earth in an elliptical plane, with the earth at one focus of the ellipse. This ellipse can be conveniently located in three-dimensional space using Geocentric Celestial Inertial coordinate system. It is possible to completely specify a satellite's orbit at any time from a set of six numbers. There are many valid six number sets that contain the necessary orbital information. For example, the three components of the position and velocity vectors in a given reference plane are sufficient to determine the orbit. However, for tasks such as orbit propagation, it is inconvenient to work with this set of numbers. A more convenient set of parameters are the orbital elements. This set includes five numbers to locate

the orbit in three-dimensional space and one number to locate the satellite's position within the orbit. They are defined as follows:

a : The semimajor axis gives the length from the center of the ellipse to the furthest point on the ellipse.

e : The eccentricity is a number from 0 to 1 which determines the shape of the ellipse. If the eccentricity is 0, the orbit is circular. For eccentricities near 1, the orbit is nearly hyperbolic or parabolic. In other words, the satellite in this orbit is nearly escaping the central body's attraction and heading into space.

i : The inclination is an angle from 0 to π . It is the angle of the orbit plane compared to the fixed equatorial reference plane.

Ω : The longitude of the ascending node is an angle that takes values from 0 to 2π . It is the angle from the axis \mathbf{x}_1 to the ascending intersection of the reference and orbital plane.

ω : The argument of perigee is an angle taking values from 0 to 2π . It is the angle from the longitude of the ascending node to the perigee, which is the point of the orbit where the satellite is closest to the Earth.

f : The true anomaly is the angle taking values from 0 to 2π , which measures the angle between the perigee and the radius vector from the Earth to the satellite's current position.

See Figure 10 for a pictorial representation of the orbital elements. Note that since the first five of the orbital elements determine the orbit's location in space, they remain fixed in the two-body model, while the sixth element determines the satellite's location within the orbit.

Also note that conversions can be made between the orbital elements and the Cartesian radius and velocity vectors. These conversions are available in APPENDIX D.

5.1.2 Kepler's second law

Let Figure 11 represent the circumscribed orbital plane. If T is the orbital period and Δt is the time it takes the satellite to get from the perigee located at point P, to its current position at point C, then Kepler's Law may be stated as

$$\frac{\Delta t}{T} = \frac{\text{Area of sector FCP}}{\text{Area of Ellipse}} \quad (43)$$

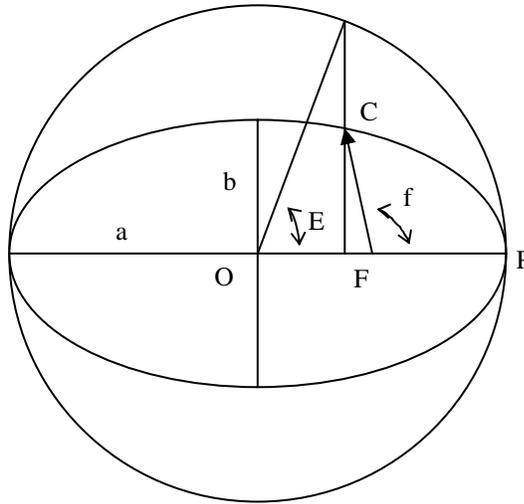


Figure 11: Circumscribed Two-Dimensional Elliptical Plane

The following results have been derived (see[6] for example):

$$\text{Area of sector FCP} = \frac{1}{2} ab(E - e \sin E) \quad (44)$$

$$\text{Area of an ellipse} = \pi ab \quad (45)$$

$$T = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (46)$$

The parameter μ is the gravitational constant that has the value of $398,600 \text{ km}^3/\text{s}^2$ for the Earth, b is the semi minor axis of the ellipse, and E is the eccentric anomaly. The rest of the parameters

are either defined in Figure 11 or are included in the orbital elements. It should be noted that it is quite easy to convert between E and f using the following equation:

$$\tan\left(\frac{E}{2}\right) = \tan\left(\frac{f}{2}\right) \sqrt{\frac{1-e}{1+e}} \quad (47)$$

Combining Equations (43) through (46), the following result is obtained:

$$\Delta t = \sqrt{\frac{a^3}{\mu}} (E - e \sin E) \quad (48)$$

Equation (48) is the key equation that is used for orbit propagation. This is explained in the following section.

5.1.3 Orbit propagation algorithm

Due to the nature of Newtonian two-body problem, the satellite always remains in the same known orbit. Thus, once the position and velocity are known at some initial time t_0 , it is possible to propagate the orbit to determine the radius and velocity of the satellite at any desired time t_d by following the method described in Figure 12. It should be pointed out that step 6 of this algorithm is not trivial, as it requires solving a transcendental equation. Thus, finding an analytic solution to this equation is not possible. However, iterative methods may be used to find a solution to within the required level of accuracy. Thus, using this procedure, it is possible to determine a satellite's exact position at any time. This completes the first necessary algorithm for ION's attitude control system.

Orbit Propagation Algorithm
Inputs: Initial time, t_0 ; desired time, t_d ; either the radius and velocity at the initial time, or the orbital elements, including f_0 , the initial true anomaly.
Step 1. If the initial known parameters are the position and velocity of the satellite, convert them to the orbital elements as outlined in APPENDIX D.
Step 2. Convert f_0 to E_0 , the eccentric anomaly at the initial time using Equation (47).
Step 3. Find Δt_0 from Equation (48). This represents the time that has passed since the satellite was last at perigee.
Step 4. Let $t_p = t_0 - \Delta t_0$. This is the time when the satellite was last at perigee.
Step 5. Let $\Delta t_d = t_d - t_p$. This is the time that passed from the perigee time to the desired time.
Step 6. Using Δt_d , solve Equation (48) for E_d . This is the eccentric anomaly at the desired time.
Step 7. Convert E_d to f_d using Equation (47). This is the true anomaly at the desired time.
Step 8. If necessary, convert the orbital elements to the desired position and velocity vectors. This may be done using APPENDIX D.

Figure 12: Orbit Propagation Algorithm

5.2 The Geomagnetic Field

The previous section dealt with determining ION's location in space. The second major task for the attitude control system is to calculate the nominal geomagnetic field at this location.

The geomagnetic field is a naturally occurring phenomenon, which has been measured at a variety of positions on the Earth's surface. For general information regarding the geomagnetic field see [7] or [8]. From the various measurements, algorithms have been developed to obtain accurate estimates of the geomagnetic field for a given position. One such algorithm developed by the International Association of Geomagnetism and Aeronomy (IAGA) is the International Geomagnetic Reference Field (IGRF). Information is available at <http://nssdc.gsfc.nasa.gov/space/model/magnetos/igrf.html>.

5.2.1 Generating the magnetic field

Since the geomagnetic field is a function of position relative to the earth, it is only natural to use the Geocentric Fixed coordinate system to compute it. It is possible to generate the Earth's magnetic field by writing it as a sum of spherical harmonics, as shown in [9]:

$$\mathbf{b}_c = \sum_{n=1}^{\infty} \sum_{m=0}^n \mathbf{b}_{n,m} \quad (49)$$

The vector \mathbf{b}_c is the calculated magnetic field in the Geocentric Fixed coordinate system, and $\mathbf{b}_{n,m}$ is the spherical harmonic of degree n and order m . The components of $\mathbf{b}_{n,m}$ are found by the following equation:

$$\mathbf{b}_{n,m} = \frac{K_{n,m} a^{n+2}}{r^{n+m+1}} \left\{ \begin{array}{l} \frac{g_{n,m} C_m + h_{n,m} S_m}{r} \left[(u A_{n,m+1}(u) + (n+m+1) A_{n,m}(u)) \hat{\mathbf{r}} - A_{n,m+1}(u) \mathbf{z}_E \right] \\ -m A_{n,m}(u) \left[(g_{n,m} C_{m-1} + h_{n,m} S_{m-1}) \mathbf{x}_E + (h_{n,m} C_{m-1} - g_{n,m} S_{m-1}) \mathbf{y}_E \right] \end{array} \right\} \quad (50)$$

The parameters $g_{n,m}$ and $h_{n,m}$ are the Gauss coefficients of degree n and order m , as published by the IGRF. The parameter a is the mean radius of the Earth (6371 km) as defined by the IGRF. The parameter r is the magnitude of \mathbf{r} , the desired location at which to find the magnetic field in the Geocentric Fixed coordinate system. The unit vector $\hat{\mathbf{r}}$ is in the direction of \mathbf{r} . The parameter u is the third component of $\hat{\mathbf{r}}$ which may be found as follows:

$$u = \hat{\mathbf{r}} \cdot \mathbf{z}_E \quad (51)$$

The remaining terms in (50) may be determined recursively through the following recursions:

$$\begin{aligned}
K_{n,0} &= K_{1,1} = 1, \forall n \in \{1, \dots, \infty\} \\
K_{n,m} &= \left(\frac{n-m}{n+m} \right)^{1/2} K_{n-1,m}, \forall m \in \{1, \dots, \infty\}, n > m \\
K_{n,m} &= ((n+m)(n-m+1))^{-1/2} K_{n,m-1}, \forall m \in \{2, \dots, \infty\}, n \geq m
\end{aligned} \tag{52}$$

$$\begin{aligned}
A_{0,0}(u) &= 1 \\
A_{n,n}(u) &= (1)(3)(5) \cdots (2n-1) \forall n \in \{1, \dots, \infty\} \\
A_{n,m}(u) &= \frac{1}{n-m} \left[(2n-1)u A_{n-1,m}(u) - (n+m-1)A_{n-2,m}(u) \right], \forall m \in \{0, \dots, \infty\}, n > m
\end{aligned} \tag{53}$$

$$\begin{aligned}
S_0 &= 0 \\
S_1 &= \mathbf{r} \bullet \mathbf{y}_E
\end{aligned} \tag{54}$$

$$\begin{aligned}
C_0 &= 1 \\
C_1 &= \mathbf{r} \bullet \mathbf{x}_E
\end{aligned} \tag{55}$$

$$S_m = S_1 C_{m-1} + C_1 S_{m-1} \tag{56}$$

$$C_m = C_1 C_{m-1} - S_1 S_{m-1} \tag{57}$$

$K_{n,m}$ are known as the Schmitt coefficients, while $A_{n,m}$ are derived Legendre polynomials. S_m and C_m are related to the Schmitt coefficients, but are separated as a means of simplifying the recursions.

It is important to note that the Schmitt coefficients are independent of the desired position. Thus, if the magnetic field is to be calculated for a number of points, the Schmitt coefficients need only be calculated once, and may be reused for each subsequent calculation. However, the rest of the parameters are location dependent, and must be recalculated for each desired location. The interested reader may see [1] or [9] for a more detailed explanation of the development of Equations (50) through (57).

5.2.2 Magnetic field algorithm

Although the true value of \mathbf{b}_c is obtained by an infinite summation, good results may be found by merely truncating this infinite sum, as all the terms of high degree and order rapidly approach zero. Tests have shown that good results may be obtained using order 10 for both m and n . Using the above equations, it is possible to obtain an algorithm to determine the magnetic field in relation to the GCI coordinate system. The algorithm is depicted in Figure 13.

Magnetic Field Algorithm
Inputs: The desired position, \mathbf{r} .
Step 1. If \mathbf{r} is not in the Geocentric Fixed coordinate system, convert it as outlined in APPENDIX E.
Step 2. Use Equations(50) through (57) to calculate the magnetic field, \mathbf{b}_c as a function of this position vector.
Step 3. If necessary, convert \mathbf{b}_c from the Geocentric Fixed coordinate system to the desired coordinate system as outlined in APPENDIX E.

Figure 13: The Magnetic Field Algorithm

5.3 Attitude Estimation

After calculation of the geomagnetic field, it can be used to estimate the attitude of the spacecraft. It is important to note that it is not possible to use a “snapshot” methodology to obtain an estimate of the attitude directly by comparing the calculated and measured vectors at each sample time independent of previous information. The reason this is true ultimately has to do with the number of degrees of freedom necessary to parameterize the attitude of the satellite. As stated in the previous chapter, there are three degrees of freedom necessary to parameterize the attitude. A single reading of the magnetometer provides three pieces of information regarding magnetic field (its direction in each of the three directions of the body frame of the satellite). However, it provides only two independent pieces of information regarding the satellite’s attitude. This is because the generated and measured magnetic field vectors have the

same magnitude. Thus, knowing the first two elements of the measured vector also gives the third element (assuming there is no noise in the measurement). Thus, only two degrees are specified at each reading.

This may be understood intuitively by looking at the following example. Assume that the earth's magnetic field at a particular location is simply \mathbf{x}_R . Also, assume the magnetometer takes a reading of $[1;0;0]$ for the local magnetic field strength. This does *not* imply the body axis is in the correct orientation relative to the reference axis. It only implies the x axis is correctly aligned. However, the satellite may have the incorrect rotation about this axis while still providing the same reading. See Figure 14 for a picture of this example.

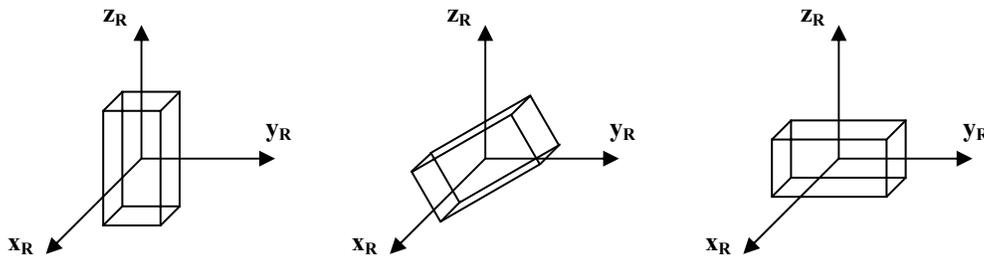


Figure 14: Attitude Ambiguity

To avoid this ambiguity, historically satellites have made comparisons of two reference vectors in attitude estimation. For example, satellites have employed a sun, moon, or horizon sensor to obtain another vector measurement. The problem of obtaining an attitude estimate from two vector measurements is known as *Wahba's Problem* [10]. As long as these two measurements are not collinear (which is almost always the case except for isolated instances), it is possible to obtain an attitude estimate at each measurement reading.

5.3.1 Extended Kalman filter

However, ION incorporates only magnetometer readings to obtain its attitude estimates. To overcome the ambiguities stated above, it has been shown that it is possible to obtain correct attitude estimates using only magnetometer data by using an extended Kalman filter [11]. For a good introduction to extended Kalman filtering and state estimation, see [12]. Reference [13] is a seminal work relating extended Kalman filtering to the problem of spacecraft attitude estimation.

The extended Kalman filter assumes a system of the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_d, t) + \mathbf{u} \quad (58)$$

The vector \mathbf{x} is the unknown state which is to be estimated, \mathbf{u}_d is the control input, and \mathbf{u} is assumed to be a white-noise process with covariance matrix \mathbf{R}_k . In general Equation (58) is a nonlinear, continuous time differential equation. Along with the state dynamics, there is an output process that is observed of the following form:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, t) + \mathbf{v} \quad (59)$$

The vector \mathbf{z} is the observed output, while \mathbf{v} is a white noise process with covariance matrix \mathbf{Q}_k . The processes \mathbf{v} and \mathbf{u} are assumed to be uncorrelated. It is assumed that both \mathbf{f} and \mathbf{h} are known, nonlinear functions. Thus, it is possible to linearize both of these equations as follows. Let the true state vector be equal to the estimated vector $\hat{\mathbf{x}}$ plus a perturbation $\Delta\mathbf{x}$:

$$\mathbf{x} = \hat{\mathbf{x}} + \Delta\mathbf{x} \quad (60)$$

Then, the equation is linearized about the estimated trajectory to obtain the following continuous time following linear equations:

$$\Delta \dot{\mathbf{x}} = \mathbf{F}(t, \mathbf{u}_d) \Delta \mathbf{x} + \mathbf{u} \quad (61)$$

$$\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}, t) = \mathbf{H}(t) \Delta \mathbf{x} + \mathbf{v} \quad (62)$$

The matrices \mathbf{F} and \mathbf{H} are found by taking the gradient of \mathbf{f} and \mathbf{h} and evaluating them along the estimated state:

$$\mathbf{F}(t, \mathbf{u}_d) = \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}_d, t) \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (63)$$

$$\mathbf{H}(t) = \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}, t) \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (64)$$

To implement the extended Kalman filter, \mathbf{z} is sampled periodically every ΔT s. At some time k , a measurement of \mathbf{z} is taken, which is referred to as \mathbf{z}_k . Also, assume there is an a priori estimate of \mathbf{x} immediately prior to k , $\hat{\mathbf{x}}_k^-$. From this estimate, it is possible to obtain an estimate of the output according to the following equation:

$$\hat{\mathbf{z}}_k^- = \mathbf{h}(\hat{\mathbf{x}}_k^-, t) \quad (65)$$

The new information gained from this estimate is known as the innovations process or measurement residual. It is simply:

$$\mathbf{v}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k^- \quad (66)$$

From this innovations process, it is possible to obtain an update for the best estimate at time k , $\hat{\mathbf{x}}_k$, by the following equation:

$$\hat{\mathbf{x}}_k = \Delta \hat{\mathbf{x}}_k + \hat{\mathbf{x}}_k^- \quad (67)$$

The perturbation $\Delta \hat{\mathbf{x}}$ is found by the equation

$$\Delta \hat{\mathbf{x}}_k = \mathbf{K}_k \mathbf{v}_k \quad (68)$$

The matrix \mathbf{K} is known as the Kalman gain. It may be computed according to the following formula:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (69)$$

The matrix \mathbf{P}_k turns out to be the error covariance matrix, which may be computed from the following equation:

$$\mathbf{P}_{k+1}^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^T + \boldsymbol{\Phi}_k \mathbf{Q}_k \quad (70)$$

$$\mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (71)$$

The matrix $\boldsymbol{\Phi}_k$ is the state transition matrix. If $\mathbf{F}(k)$ is assumed constant over the sampling period, $\boldsymbol{\Phi}_k$ is simply found by performing the following matrix exponentiation:

$$\boldsymbol{\Phi}_k = e^{\mathbf{F}(k)\Delta T} \quad (72)$$

To complete the state estimation it is necessary to project the state vector ahead to the next time step. This is done through the following equation:

$$\hat{\mathbf{x}}_{k+1}^- = \hat{\mathbf{x}}_k + \int_k^{k+1} \mathbf{f}(\hat{\mathbf{x}}_k, \mathbf{u}_d, t) dt \quad (73)$$

5.3.2 ION's attitude estimation via the extended Kalman filter

The extended Kalman filter ION employs is similar to the presentation given in the previous section, although, there are a few changes based on those which were introduced in references [11] and [13]. The goal of this filter is to estimate the ION's attitude and angular velocity. Also, a disturbance torque is estimated to increase the robustness of the filter. Thus, the state is the following 10 element vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}^{B/I} \\ \boldsymbol{\omega}^{B/I} \\ \mathbf{t}_d \end{bmatrix} \quad (74)$$

The four-element quaternion $\mathbf{q}^{B/I}$ represents the attitude of the satellite with respect to the inertial reference frame. The vector $\boldsymbol{\omega}^{B/I}$ is the angular velocity of the body frame with respect to the reference frame. The vector \mathbf{t}_d is the disturbance torque. All vectors are in the fixed body reference frame.

Unfortunately, there is a problem with this choice for the state vector. Since the quaternion consists of three independent parameters and one dependent parameter, there is a redundant parameter in the state. One alternative is to define a small quaternion $\Delta\mathbf{q}$ by the following quaternion multiplication:

$$\mathbf{q}^{B/I} = \begin{bmatrix} \Delta\mathbf{q} \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}^{B/I} \quad (75)$$

Using this definition, an alternative choice for the state vector is

$$\tilde{\mathbf{x}} = \begin{bmatrix} \Delta\mathbf{q} \\ \boldsymbol{\omega}^{B/I} \\ \mathbf{t}_d \end{bmatrix} \quad (76)$$

The equations driving ION's extended Kalman filter are based upon the nine-element implementation where only the small quaternions are estimated. Yet, all 10 elements are stored for bookkeeping purposes. This is done through a creative implementation of the Kalman gain in the measurement update equations. Equation (68) is implemented normally as

$$\Delta\hat{\tilde{\mathbf{x}}}_k = \begin{bmatrix} \Delta\hat{\mathbf{q}} \\ \Delta\hat{\boldsymbol{\omega}}^{B/I} \\ \Delta\hat{\mathbf{t}}_d \end{bmatrix} = \mathbf{K}_k \mathbf{v}_k \quad (77)$$

However, instead of using Equation (67) to obtain the new estimate, the following two equations are used:

$$\begin{bmatrix} \hat{\boldsymbol{\omega}}^{B/I} \\ \hat{\mathbf{t}}_d \end{bmatrix} = \begin{bmatrix} \Delta \hat{\boldsymbol{\omega}}^{B/I} \\ \Delta \hat{\mathbf{t}}_d \end{bmatrix} + \begin{bmatrix} \hat{\boldsymbol{\omega}}^{B/I^-} \\ \hat{\mathbf{t}}_d^- \end{bmatrix} \quad (78)$$

But the quaternion update makes use of a quaternion multiplication:

$$\hat{\mathbf{q}}^{B/I} = \hat{\mathbf{q}}^{B/I^-} \otimes \begin{bmatrix} \Delta \hat{\mathbf{q}} \\ \sqrt{1 - \|\Delta \hat{\mathbf{q}}\|^2} \end{bmatrix} \quad (79)$$

Then the new 10-element state estimate is simply

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{q}}^{B/I} \\ \hat{\boldsymbol{\omega}}^{B/I} \\ \hat{\mathbf{t}}_d \end{bmatrix} \quad (80)$$

The observation vector is simply the three-dimensional magnetic field vector that is taken by the magnetometer, as outlined in APPENDIX B. This vector is in the satellite's fixed body system. Thus:

$$\mathbf{z} = \mathbf{b}_m \quad (81)$$

The control input is the torque produced by the magnetorquers, as outlined in Equation (3) and APPENDIX A.

$$\mathbf{u}_d = \mathbf{t}_m \quad (82)$$

The state vector evolves according to Equations (40) and (41). Nevertheless, it is restated below for consistency:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}_d, t) = \begin{bmatrix} \dot{\mathbf{q}}^{B/I} \\ \dot{\boldsymbol{\omega}}^{B/I} \\ \dot{\mathbf{t}}_d \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}^{B/I}) \mathbf{q}^{B/I} \\ \tilde{\mathbf{I}}^{-1} (\mathbf{t}_m + \mathbf{t}_d - \boldsymbol{\omega}^{B/I} \tilde{\mathbf{I}}^{-1} \times (\tilde{\mathbf{I}} \boldsymbol{\omega}^{B/I})) \\ \mathbf{0} \end{bmatrix} \quad (83)$$

The measurement equation \mathbf{h} is merely the magnetic field in the fixed body coordinate system \mathbf{b}_m . It is given as

$$\mathbf{h}(\mathbf{x}, t) = \mathbf{b}_m = \mathbf{A}^{B/R} \mathbf{b}_r \quad (84)$$

The vector \mathbf{b}_r is the calculated magnetic field vector in the orbital reference frame. Note that $\mathbf{A}^{B/R}$ is a function of the quaternion $\mathbf{q}^{B/R}$, so \mathbf{h} is clearly a function of \mathbf{x} , even though this relationship is not explicitly stated in (84).

One of the major changes between the standard and ION's extended Kalman filter is in the definition of the innovation process. ION's innovation process is as recommended in [11]. Instead of defining this process as in (66), the following definition is used:

$$\mathbf{v}_k = \frac{\mathbf{z}_k \times \hat{\mathbf{z}}_k^-}{\|\mathbf{z}_k\| \|\hat{\mathbf{z}}_k^-\|} \quad (85)$$

This nonstandard innovations equation is chosen because it effectively eliminates differences in length between the calculated and measured magnetic field while concentrating on the differences in their directions. This is strongly desired, since the attitude of the satellite is not dependent upon the length of these vectors at all. If the state equation were truly linear, then there would be no advantage to using the nonstandard innovations process over the typical one. However, the atypical one performs better for estimating the nonlinear system.

The linearized state and observation equations are derived in APPENDIX I. Only the results are presented below. The linearized matrix for the state dynamics is

$$\mathbf{F}(t, \mathbf{u}_d) = \begin{bmatrix} -\mathbf{X}(\hat{\boldsymbol{\omega}}^{B/I}) & \frac{1}{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \Gamma(\hat{\boldsymbol{\omega}}^{B/I}) & \tilde{\mathbf{I}}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (86)$$

The matrix Γ is as restated below:

$$\Gamma(\boldsymbol{\omega}) = -\tilde{\mathbf{I}}^{-1} \begin{bmatrix} -I_{xz}\omega_y + I_{xy}\omega_z & -2I_{yz}\omega_y + (I_z - I_y)\omega_z - I_{xz}\omega_x & 2I_{yz}\omega_z + (I_z - I_y)\omega_y + I_{xy}\omega_x \\ 2I_{xz}\omega_x + (I_x - I_z)\omega_z + I_{yz}\omega_y & I_{yz}\omega_x - I_{xy}\omega_z & -2I_{xz}\omega_z + (I_x - I_z)\omega_x - I_{xy}\omega_y \\ -2I_{xy}\omega_x + (I_y - I_x)\omega_y - I_{yz}\omega_z & 2I_{xy}\omega_y + (I_y - I_x)\omega_x + I_{xz}\omega_z & -I_{yz}\omega_x + I_{xz}\omega_y \end{bmatrix} \quad (87)$$

The matrix \mathbf{X} is the skew symmetric matrix associated with the cross product:

$$\mathbf{X}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (88)$$

The linear output equation for the matrix \mathbf{H} is

$$\mathbf{H}(t) = \begin{bmatrix} 2\mathbf{I} - \frac{2\hat{\mathbf{b}}\hat{\mathbf{b}}_m^T}{\|\hat{\mathbf{b}}_m\| \|\hat{\mathbf{b}}\|} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (89)$$

The remainder of the equations necessary to implement the Kalman filter are the state and measurement time update, the Kalman gain, and the time and measurement covariance matrices.

These are identical to the standard equations for the extended Kalman filter.

5.4 Convergence Issues

The extended Kalman filter has a number of issues regarding its convergence. First of all, it is hard to obtain an accurate estimate of the state and measurement noise covariance matrices. Also, the initial conditions on the state and covariance of the state are crucial factors in obtaining convergence. For the standard Kalman filter, these conditions are not as critical.

However the nonlinearities associated with the extended Kalman filter cause severe problems with the convergence if the parameters differ greatly from the true values.

5.5 Extended Kalman Filter Algorithm

The extended Kalman filtering algorithm is given in Figure 15:

Extended Kalman Filtering Attitude Estimation Algorithm
Inputs: initial conditions $\hat{\mathbf{x}}_0^-$ and \mathbf{P}_0^- .
Step 1. Find $\hat{\mathbf{b}}_k$ according to Equation (84).
Step 2. Make the measurement at time k. This is \mathbf{b}_k
Step 3. Make the measurement updates according to steps 3a through 3f.
Step 3a. Find \mathbf{v}_k according to Equation (85).
Step 3b. Find \mathbf{H}_k according to Equation (89).
Step 3c. Find \mathbf{K}_k according to Equation (69).
Step 3d. Compute $\Delta\tilde{\mathbf{x}}$ according to Equation (77).
Step 3d. Compute $\hat{\mathbf{x}}_k$ according to Equations (78) through (80).
Step 3e. Compute \mathbf{P}_k according to Equation (71).
Step 3f. Compute \mathbf{F}_k according to Equation (86).
Step 4. Make the time update for the next time step according to steps 4a and 4b.
Step 4a. Predict $\hat{\mathbf{x}}_{k+1}^-$ according to Equation (73).
Step 4b. Compute \mathbf{P}_{k+1}^- according to Equation (70).
Step5. Set k = to k+1. Wait for the next sampling time. Then, go to step 2.

Figure 15: Attitude Estimation Algorithm

5.6 The Linear Quadratic Regulator Control Law

Magnetorquers have been used to control spacecraft in a variety of ways. Traditionally, magnetorquers have been used in conjunction with reaction-wheel based systems. The reaction wheels are used for precision attitude control, while the torquers are used to “dump” excess momentum when the wheel rates get too high. Recently, various control laws have been developed to implement three-axis attitude control without the use of reaction wheels. The first controller was a projection based attitude control system. More sophisticated techniques such as

linear quadratic regulators, fuzzy control, and sliding mode control systems have been simulated. Thus, there are a variety of control laws available to the designer to implement magnetorquers.

Magnetic control is not a trivial problem. This is because, as given in Equation (3), the control torque is simply the cross product of the magnetorquers' magnetic moment and the geomagnetic field. Thus, the control torque is orthogonal to both of these vectors. Thus, it is never possible to generate any control torque in the direction of the geomagnetic field. This results in an uncontrollable subspace. Luckily enough, this subspace is time varying, due to the time varying nature of the geomagnetic field as the satellite changes its orbit. Thus, the satellite is controllable for highly inclined orbits.

ION's attitude control system will utilize an asymptotic quasi periodic linear quadratic regulator. This is similar to the control law proposed in [14]. This section will proceed with discussion of some of the theory associated with the linear quadratic regulator. It will then proceed to describe the actual design necessary for ION's control law.

5.6.1 Linear quadratic regulator theory

The linear quadratic control problem is an optimal control problem. It is covered in most modern control texts, such as [15]. In this problem, it is assumed that there is a system in which the state dynamics are constant, but the input dynamics vary with time:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(t)\mathbf{u}, \text{ given } \mathbf{x}(t_0) \quad (90)$$

For this problem, the following cost function is assigned:

$$\mathbf{J} = \frac{1}{2} \int_{t_0}^{t_f} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) d\tau + \frac{1}{2} \mathbf{x}(t_f)^T \mathbf{P}_T \mathbf{x}(t_f), \text{ given } \mathbf{x}(t_0) = \mathbf{x}_0 \quad (91)$$

The matrix \mathbf{Q} may be thought of as a penalty on the state vector to discourage the state from deviating too large, \mathbf{R} as a matrix to penalize using too much control, and \mathbf{P}_T as a penalty on the final state. All of these are constant matrices. From the point of view of the designer, unless these matrices have some a priori designation, they may be viewed as a means of changing the characteristics of the system dynamics. Thus, these matrices provide a quantitative way of trading off state deviation with control energy. All of \mathbf{Q} , \mathbf{R} , and \mathbf{P}_T are assumed to be positive definite.

The goal of the linear quadratic regulator is to find the optimal control to minimize the cost function given these various matrices and an initial state condition. It is a well-known result that the optimal solution to this problem is a full state feedback controller of the following form:

$$\mathbf{u}^* = \mathbf{F}\mathbf{x} = -\mathbf{R}^{-1}\mathbf{B}(t)^T \mathbf{P}(t)\mathbf{x} \quad (92)$$

The matrix $\mathbf{P}(t)$ is found by the following differential equation:

$$\dot{\mathbf{P}}(t) = \mathbf{Q} + \mathbf{P}(t)\mathbf{A} + \mathbf{A}^T \mathbf{P}(t) - \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}^{-1}\mathbf{B}(t)^T \mathbf{P}(t), \text{ given } \mathbf{P}(t_f) = \mathbf{P}_T \quad (93)$$

In principle, this equation may be evaluated in closed form or analytically, to find the time varying matrix $\mathbf{P}(t)$, which in turn specifies \mathbf{u}^* .

5.6.1.1 Asymptotic periodic linear quadratic regulator

A special case of the quadratic linear regulator is where the matrix $\mathbf{B}(t)$ is periodic. In this case, for some value of T and all t ,

$$\mathbf{B}(t) = \mathbf{B}(t + T) \quad (94)$$

In this case, if \mathbf{P}_T is properly chosen, it can be shown that the feedback gain matrix is also periodic in T . This periodicity in the feedback matrix can be used to show that, in the case of

some very general conditions on the penalty matrix \mathbf{Q} , $\mathbf{P}(t)$ approaches a steady state matrix \mathbf{P}_{SS} as the minimum eigenvalue of \mathbf{R} approaches infinity. Thus, for large values of \mathbf{R} , it can be expected that this \mathbf{P}_{SS} matrix may be used as a reasonable approximation for $\mathbf{P}(t)$ for all values of t . In this case, the optimal control law becomes

$$\mathbf{u}^* = \mathbf{F}\mathbf{x} - \mathbf{R}^{-1}\mathbf{B}(t)^T \mathbf{P}_{SS} \mathbf{x} \quad (95)$$

To find \mathbf{P}_{SS} , note that the following long average over one period can be made:

$$\mathbf{B}(t)\mathbf{R}^{-1}\mathbf{B}(t)^T \approx \mathbf{C} = \frac{1}{T} \int_0^T \mathbf{B}(\tau)\mathbf{R}^{-1}\mathbf{B}(\tau)^T d\tau \quad (96)$$

These approximations greatly simplify the implementation for the linear quadratic regulator. Inserting this approximation into the differential equation found in Equation (93), it is possible to solve the following algebraic Riccati equation:

$$\mathbf{0} = \mathbf{Q} + \mathbf{P}_{SS}\mathbf{A} + \mathbf{A}^T\mathbf{P}_{SS} - \mathbf{P}_{SS}\mathbf{C}\mathbf{P}_{SS} \quad (97)$$

5.6.2 ION's asymptotic periodic linear quadratic regulator design

To apply the linear quadratic regulator to ION's attitude control system, the state vector consists of the attitude and angular velocity of the fixed body coordinate system with respect to the orbital reference frame. Only three elements are necessary to represent the attitude as the fourth element is redundant. Thus, the six-element state is as follows:

$$\mathbf{x}_{lqr} = \begin{bmatrix} \mathbf{q}^{\mathbf{B}/\mathbf{R}} \\ \boldsymbol{\omega}^{\mathbf{B}/\mathbf{R}} \end{bmatrix} \quad (98)$$

The subscript lqr is used to make a distinction between the state vector used by the linear quadratic regulator and the state vector used by the extended Kalman filter. The first three

elements of the state are the three element attitude vector. The final three elements are the angular velocity vectors. The input is the magnetic moment generated by the magnetorquers, as outlined in APPENDIX A and Equation (3). It is as follows:

$$\mathbf{u} = \mathbf{m} \quad (99)$$

The system dynamics are clearly nonlinear. Furthermore, the state dynamics are time varying. Thus, to apply the linear quadratic regulator theory, it is first necessary to linearize the system about a nominal trajectory. This derivation is available in APPENDIX I. Only the results are presented below. The system takes the form of Equation (90) with the following definitions:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \frac{1}{2} \mathbf{I} \\ \omega_0 \tilde{\mathbf{I}}^{-1} \begin{bmatrix} 8\omega_0(I_z - I_y) & -6\omega_0 I_{xy} & 2I_{xz} & 0 & -2I_{yz} & I_x - I_y + I_z \\ -8\omega_0 I_{xy} & 6\omega_0(I_z - I_x) & -2\omega_0 I_{yz} & 2I_{yz} & 0 & -2I_{xy} \\ 8\omega_0 I_{xz} & 6\omega_0 I_{yz} & 2\omega_0(I_x - I_y) & -I_x + I_y - I_z & 2I_{xy} & 0 \end{bmatrix} \end{bmatrix} \quad (100)$$

The parameter ω_0 is the magnitude of the orbital velocity of the reference coordinate system with respect to the inertial coordinate system. It should be noted that this is assumed to be a constant throughout the derivations. There will be some slight deviation of the angular velocity for noncircular orbits. However, the average orbital velocity may be used in the calculations. Also, the above system dynamics take into account the gravity gradient effects when computing the linearized system.

The contribution from the input can be taken into account:

$$\mathbf{B}(t) = \begin{bmatrix} \mathbf{0} \\ -\tilde{\mathbf{I}}^{-1} \mathbf{X}(\mathbf{b}(t)) \end{bmatrix} \quad (101)$$

The vector \mathbf{b} is the geomagnetic field vector in the fixed body coordinate system. The matrix \mathbf{X} is the normal skew symmetric matrix associated with the cross product. It is defined as

$$\mathbf{X} \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{bmatrix} 0 & -b_z & b_y \\ b_z & 0 & -b_x \\ -b_y & b_x & 0 \end{bmatrix} \quad (102)$$

Note that if the spacecraft remained in its nominal attitude, and the changes in the geomagnetic field due to the Earth's rotation are ignored, then the linear matrix $\mathbf{B}(t)$ would be periodic. Thus, it is reasonable to assume that a steady state matrix solving the algebraic Riccati equation would be a reasonable approximation to the optimal solution.

5.6.2.1 Integral control

Integral control has many known benefits. Notable are its stabilizing tendencies and noise reduction. To introduce integral control into this design, the above system is augmented as follows. Introducing a new state vector consisting of the integral of the three-element quaternion and the previous state:

$$\tilde{\mathbf{x}}_{lqr} = \begin{bmatrix} \int \mathbf{q}^{B/R} \\ \mathbf{q}^{B/R} \\ \boldsymbol{\omega}^{B/R} \end{bmatrix} \quad (103)$$

The modified system is

$$\dot{\tilde{\mathbf{x}}}_{lqr} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{lqr} + \tilde{\mathbf{B}}(t)\mathbf{u} \quad (104)$$

The modified matrices are

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \end{bmatrix} \quad (105)$$

$$\tilde{\mathbf{B}}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}(t) \end{bmatrix} \quad (106)$$

5.6.2.2 Calculating the C matrix

In the discussion above, \mathbf{C} was calculated by averaging the effect of the matrix $\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$ over one orbital period. However, this only works if the \mathbf{B} matrix is periodic. Since the real matrix will not be exactly periodic, this technique must be modified slightly by performing the average over several orbits to include approximately one day's worth of data. This amounts to approximately 15 orbits worth of data. Performing this longer integration will average out the effect of the rotation of the earth. Thus, \mathbf{C} is calculated as:

$$\mathbf{C} = \frac{1}{15T} \int_0^{15T} \tilde{\mathbf{B}}(\tau)\mathbf{R}^{-1}\tilde{\mathbf{B}}(\tau)^T d\tau \quad (107)$$

In this calculation, the magnetic field vector is calculated in the reference frame, since the calculation assumes the satellite is in the nominal orientation. It should be noted that it is only necessary to calculate \mathbf{C} once. This matrix is then used to solve the algebraic Riccati equation found in Equation (97) for \mathbf{P}_{ss} .

5.6.2.3 Control algorithms

There are two control algorithms. The first is an offline calculation to find the matrix \mathbf{P}_{ss} . This algorithm, which is seen in Figure 16, must be run only once, prior to implementing the control algorithm.

Off Line Algorithm to Calculate \mathbf{P}_{ss}
Inputs. The penalty matrices \mathbf{R} and \mathbf{Q} , as well as the average orbital angular velocity, ω_0 .
Step 1. Calculate nominal magnetometer values, $\mathbf{b}_r(t)$, for 15 orbits.
Step 2. Calculate \mathbf{C} from Equation (107).
Step 3. Calculate $\tilde{\mathbf{A}}$ from Equation (105).
Step 4. Calculate \mathbf{P}_{ss} from Equation (97).

Figure 16: \mathbf{P}_{ss} Calculation

Before stating the main control algorithm, it is necessary to convert the extended Kalman filter's state vector to the one required by the linear quadratic regulator. The extended Kalman filter estimates the quaternion and angular velocity of the satellite with respect to the inertial frame, but the linear quadratic regulator requires them to be with respect to the orbital reference frame. To convert the angular velocity, the rule stated in Equation (32) is used.

$$\boldsymbol{\omega}^{B/R} = \boldsymbol{\omega}^{B/I} - \boldsymbol{\omega}^{R/I} \quad (108)$$

The angular velocity $\boldsymbol{\omega}^{R/I}$ is obtained from Equation (139). To convert between the different quaternions, Equations (27) and (29) may be used as follows:

$$\boldsymbol{q}^{B/R} = (\boldsymbol{q}^{R/I} *) \otimes \boldsymbol{q}^{B/I} \quad (109)$$

The quaternion $\boldsymbol{q}^{R/I}$ must be found by converting the attitude matrix $\mathbf{A}^{R/I}$, as defined in Equation (138). This conversion is found in APPENDIX F. After addressing these conversions, it is possible to state the linear quadratic regulator control algorithm as shown in Figure 17.

Linear Quadratic Regulator Control Algorithm
Inputs. State estimate from extended Kalman filter, \mathbf{x} ; penalty matrix \mathbf{R} , and the steady state matrix \mathbf{P}_{ss}
Step 1. Calculate $\tilde{\mathbf{x}}_{lqr}$ from \mathbf{x} , as outlined in Equations (108) and (109).
Step 2. Obtain state estimate \mathbf{x} and magnetometer reading \mathbf{b} .
Step 3. Find $\tilde{\mathbf{B}}$ according to Equation (106).
Step 2c. Find \mathbf{u} according to Equation (95).
Step 2d. Find torque coil circuitry inputs as outlined in APPENDIX A.

Figure 17: Attitude Control Algorithm

5.6.2.4 Results

The simulations show that the linear quadratic control law performs satisfactorily in some circumstances. Through the process of trial and error, the following were found to be good values of the penalty matrices.

$$\mathbf{R} = 5.7\mathbf{I} \quad (110)$$

$$\mathbf{Q} = \text{diag}([1.8 \times 10^{-8} \quad 1.8 \times 10^{-8} \quad 1.8 \times 10^{-8} \quad 1 \quad 1 \quad 1 \quad .1 \quad .1 \quad .1]) \quad (111)$$

Using these matrices, a number of tests have been run. A typical plot of the satellite in steady-state operation is presented in Figure 18. The time scale for this figure is in hours. As can be seen, the satellite typically stays within a 5° error for the yaw, pitch, and roll angles. This is clearly within the specifications required by the photomultiplier tube.

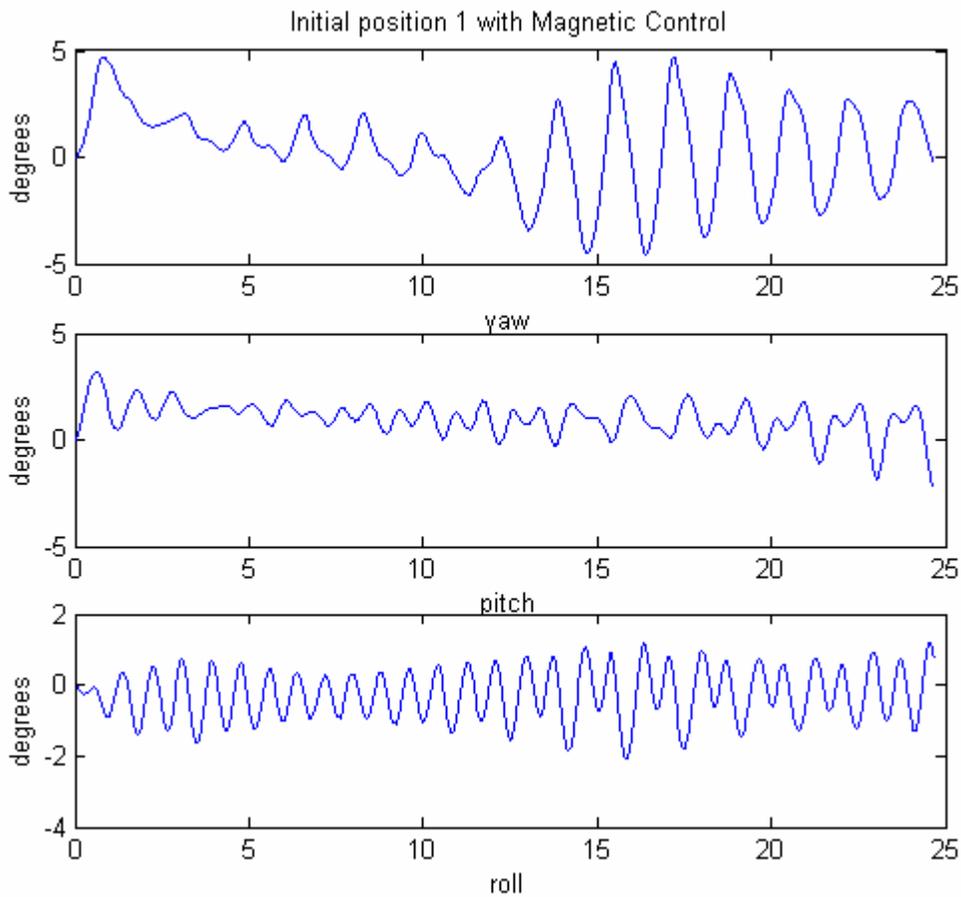


Figure 18: Typical Steady State Yaw, Pitch, and Roll Angles for Linear Quadratic Regulator Control

A second plot is included to show how the satellite responds from some initial deviation. The satellite starts off at a 90° rotation about the y-axis. This response is shown in Figure 19.

The time scale in this figure is in hours. The pitch angle is the slowest of the three to converge, but it finally does so in approximately 8 hours.

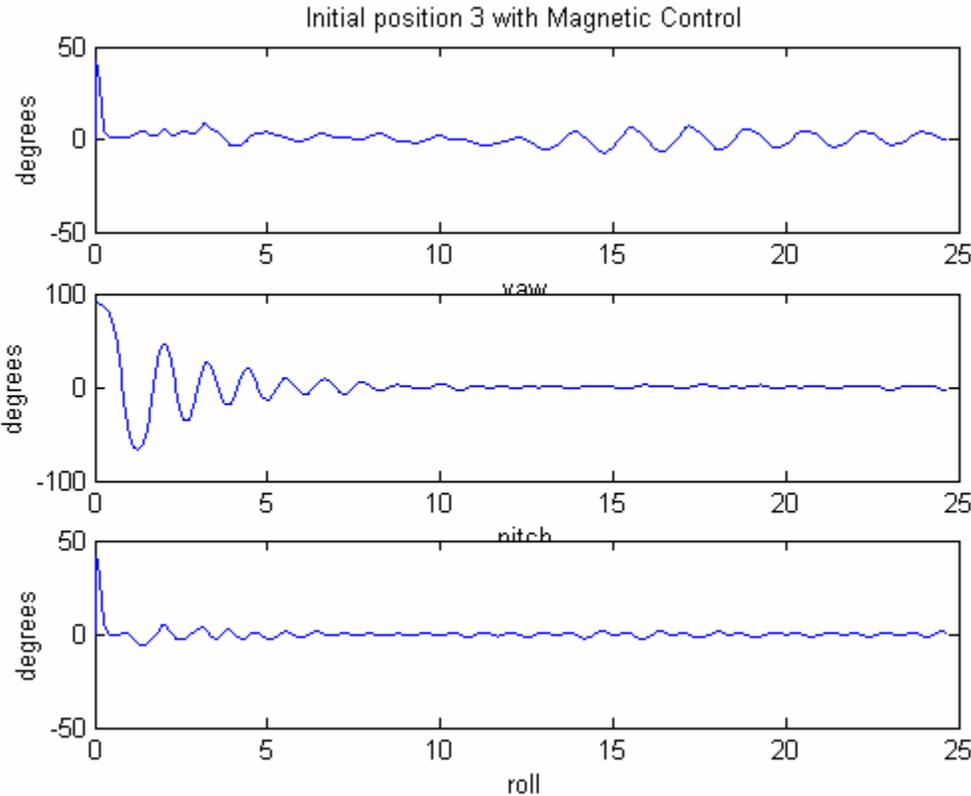


Figure 19: Typical Yaw, Pitch, and Roll Angles for an Initial 90° Attitude Deviation.

6 FUTURE WORK AND OPEN ISSUES

There are a considerable number of open issues and future work that could be done to improve ION's attitude control system. These issues shall be addressed in order of priority.

6.1 Extended Kalman Filter Convergence Issues

The greatest concern with the attitude control system lies in the attitude estimation problem. The extended Kalman filter rarely converges to the correct attitude. This is a major obstacle that needs to be addressed. One means of doing so has been suggested in [16]. This paper addresses using data available from solar panels specifically to address the convergence of the extended Kalman filter. This technique could provide a simple fix to the problem without adding any more sensors.

Another possibility involves adding additional sensors to ION. It has recently been identified that the SID may easily be configured to incorporate rate gyros. These would be used to measure the angular velocity of the satellite with respect to the inertial coordinate system. This would greatly simplify the filter, since effectively half the unknown states would be removed. Both possibilities of incorporating solar panel measurements and rate gyro measurements could be investigated further in parallel as a means to solve these convergence issues.

6.2 Implementation Issues

A second open issue at this time lies in implementation of the attitude control system. It turns out that the processing capabilities of the SID are not as extensive as was originally hoped. To address these issues, much of the attitude control system must be implemented off-line on the ground station. Then, commands may be sent through the communication system to ION as to

how to correctly use the torque coils. This will add huge amounts of delay to the attitude control system. This could cause severe problems for the linear quadratic regulator, which is a closed loop state feedback control law. One way to address this issue would be in implementing another control law. In [14], a number of alternative attitude control laws have been listed. Some of the alternative methods include fuzzy control, sliding mode control, and a projection based control law. However, further research may provide additional control laws which might be more suited to an open loop approach.

6.3 Detumbling Mode

In addition to the normal control law, many satellites utilize a detumbling control law. This secondary control law is used initially to remove angular momentum in the case that the satellite is deployed with an appreciable spin rate. It might be worthwhile to consider adding a detumbling control law to initialize the attitude control system, since simulations have shown that the linear quadratic control law does not perform well for initial conditions with large angular momentum.

6.4 Simulation With Real Magnetometer Data

Magnetometers have been used in space applications for many years. Testing the attitude control system with real magnetometer measurements (as opposed to merely the IGRF data) could prove invaluable as a means of verifying the control system. In [16], the authors mention testing their extended Kalman filtering system with data taken by the Danish Ørsted satellite. It would be beneficial if this data could be obtained to test ION's attitude control system.

6.5 Reducing the Order of Harmonics in the Magnetic Field Model

One of the most mathematically intensive components of the attitude control system is calculating the magnetic field. Currently the IIRGF field model calculates the spherical harmonics n and m of order 10 for each. It might be possible to reduce the number of components necessary in the calculation and drastically reduce the computational burden without losing much accuracy in the magnetic field model. This is one possible technique that could be explored.

6.6 Flight Torque Coils

At this point, the final flight torque coils have yet to be constructed. The initial torque coil design has been completed with the estimated space available for each coil. These results are presented in APPENDIX A. However, these results may change slightly due to the actual volume constraints inside of the satellite. After the final available volume has been determined, Equation (8) can be used to find the correct number of turns for the flight torque coils. Then, it will be necessary to construct these final coils.

6.7 Testing

Besides the open issues that are listed above, ION's entire attitude control system design has been finalized. However, at this point extensive testing is still necessary to verify the functionality of the design. Although all of the necessary circuitry to run the magnetometer and torque coils has been designed, these circuits require extensive testing to verify correct functionality. It will also be necessary to measure the important parameters from each of the torque coils. These parameters are listed later in TABLE 2 in APPENDIX A. It will be necessary to ensure that the magnetometer calibration functions correctly and accurately samples the magnetic field. This functionality is described in APPENDIX B. The magnetometer's

functionality may also be verified from measurements by a Gaussmeter. All of these tests must be completed to ensure the hardware functions properly.

APPENDIX A: TORQUE COIL IMPLEMENTATION

The actual implementation of the torque coils in ION is relatively straightforward. The only required elements are coils themselves and their necessary circuitry. The torque coils are made of Belden heavy armored poly-thermaleze 38 AWG magnet wire. For the preliminary parameters relating to ION's torque coils, see TABLE 2.

TABLE 2: Torque Coil Parameters

Torque Coil Parameters			
Coil	X	Y	Z
H bridge Resistance	.4 Ω	.4 Ω	.4 Ω
Cross Sectional Area	1.32x10 ⁻⁸ m ²	1.32x10 ⁻⁸ m ²	1.32x10 ⁻⁸ m ²
Conductivity of Copper	5.8x10 ⁷ Ω m	5.8x10 ⁷ Ω m	5.8x10 ⁷ Ω m
Nominal Unregulated Voltage	11.1V	11.1V	11.1V
Length of First Side of Coil	.184 m	.111 m	.075 m
Length of Second Side of Coil	.083 m	.072 m	.075 m
Number of turns	1084	1582	1930
Coil Resistance	1232 Ω	1232 Ω	1232 Ω
Max Current	9.01x10 ⁻³ A	9.01x10 ⁻³ A	9.01x10 ⁻³ A
Max Magnetic Dipole Moment	.149 Am ²	.114 Am ²	.0978 Am ²
Max Magnetic Field	1.03 G	1.89 G	2.62 G

A.1 Torque Coil Circuitry

The circuitry necessary to interface the torque coils and the SID is depicted in the schematic in Figure 20. Since the three torque coils have identical circuitry, explanation given in this discussion will be limited only to the x torque coil. Besides the torque coil, the only other components in the circuit are two pull-up resistors and two ICs: the Texas Instruments TPIC0108 is the H-bridge logic which determines the direction of the current flow of the torque coil. The Maxim MAX319 serves as an interface between the control signals and the H-bridge.

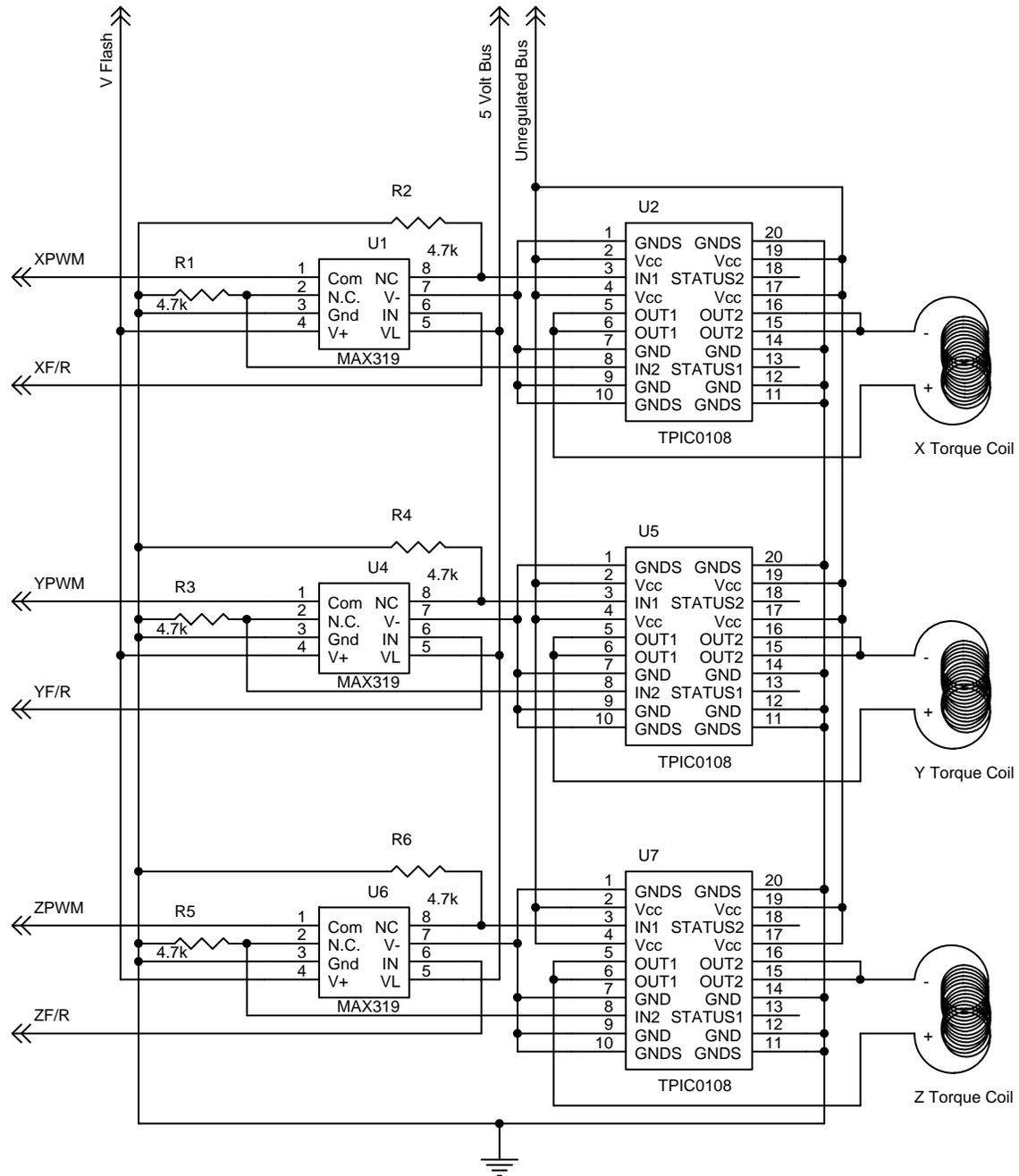


Figure 20: Torque Coil Schematic

Besides the various voltage buses necessary to power the chips, two inputs are required from the microprocessor. “XF/R” is a signal which controls the direction of current flow in the x torque coil. If the voltage is + 5 V, then the current in the coil flows in the forward direction, from the positive to the negative terminal. If the voltage is 0 V, then the current in the coil flows

in the reverse direction, from the negative to the positive terminal. The coil should be oriented in ION in such a way that when “XF/R” is +5 V, the magnetic field generated is in the \mathbf{x}_B direction, as defined in Figure 7. The other signal, “XPWM,” is a pulse width modulation signal which alternates between 0 and + 5 V at a specified frequency and duty cycle. The frequency of this signal is irrelevant. Whenever the voltage of “XPWM” is +5 V, the torque coils are firing. When the voltage is 0 V, they are not in operation. Thus, when the coils are in operation, the duty cycle of “XPWM” acts as a throttle by controlling the percentage of time the coils are in use. For more information on the torque coil circuitry, please see the documentation provided by the manufacturers.

A.2 Calculating Inputs for Torque Coil Circuitry

The control law uses the magnetic moment generated by the torque coils as the control. To generate the correct torque called for by the control law, it is necessary to convert between this magnetic moment and the six inputs to the torque coil circuitry. This calculation is outlined below. First, in case of torque coil saturation effects, the scale factor β is calculated as

$$\beta = \max \left\{ \frac{|u_j|}{N_j I_j A_j}, 1 : j \in \{x, y, z\} \right\} \quad (112)$$

The parameter u_j is the j^{th} component of the requested input. The other parameters, N_j , I_j , and A_j , are the number of turns, torque coil current, and coil area, respectively, of the j^{th} torque coil. The parameter β takes on values greater than one in the event of torque coil saturation. In other words, the computations are requesting more torque than the coils are able to provide. If there is no saturation, then β will be one. The j^{th} duty cycle output is calculated as

$$JPWM = 5 \frac{|u_j|}{\beta N_j I_j A_j}, j \in \{x, y, z\} \quad (113)$$

The j^{th} direction output is simply

$$JF / R = 2.5 \operatorname{sgn}(u_j) + 2.5, j \in \{x, y, z\} \quad (114)$$

APPENDIX B: MAGNETOMETER IMPLEMENTATION

As mentioned above, the particular magnetometer which is being flown aboard ION is the Honeywell HMC2003. For this sensor to function properly, it must be implemented correctly and integrated into ION. This implementation consists of three major parts: the additional circuitry necessary to interface the SID with the magnetometer, the calibration procedure for the magnetometer, and the procedure to take accurate readings.

B.1 Magnetometer Circuitry

See Figure 21 for the complete magnetometer circuit schematic. The magnetometer circuit consists of three ICs, three resistors, eight capacitors, two diodes, and a transistor. The circuitry comes right from the example circuits found in the Honeywell documentation.

B.1.1 Hardware

The Maxim MAX662A chip and its associated circuitry are used to generate a 20-V signal. The International Rectifier IRF7105 chip with its associated circuitry uses this 20-V signal, along with the “Set” and “Reset” signals from the SID to generate the “SR+” signal to calibrate the magnetometer circuit. Finally, the Honeywell HMC2003 is the magnetometer that senses the magnetic field.

B.1.2 Signals

The only two inputs to the system are the “SET” and “RESET” signals. They are generated by the microprocessor and are used to calibrate the magnetometer. See Figure 22 for these signal specifications. T_{RS} and T_{SR} must be greater than 5 μ s. T_{PWS} and T_{PWRS} should each be approximately 2 μ s. The duration of T_{SET} is not nearly as critical, but should be long enough

to ensure a sufficient delay between the “S/R+” signal pulses, while also providing ION with enough time to sample the magnetometer. A typical value for this duration is approximately 25 μ s.

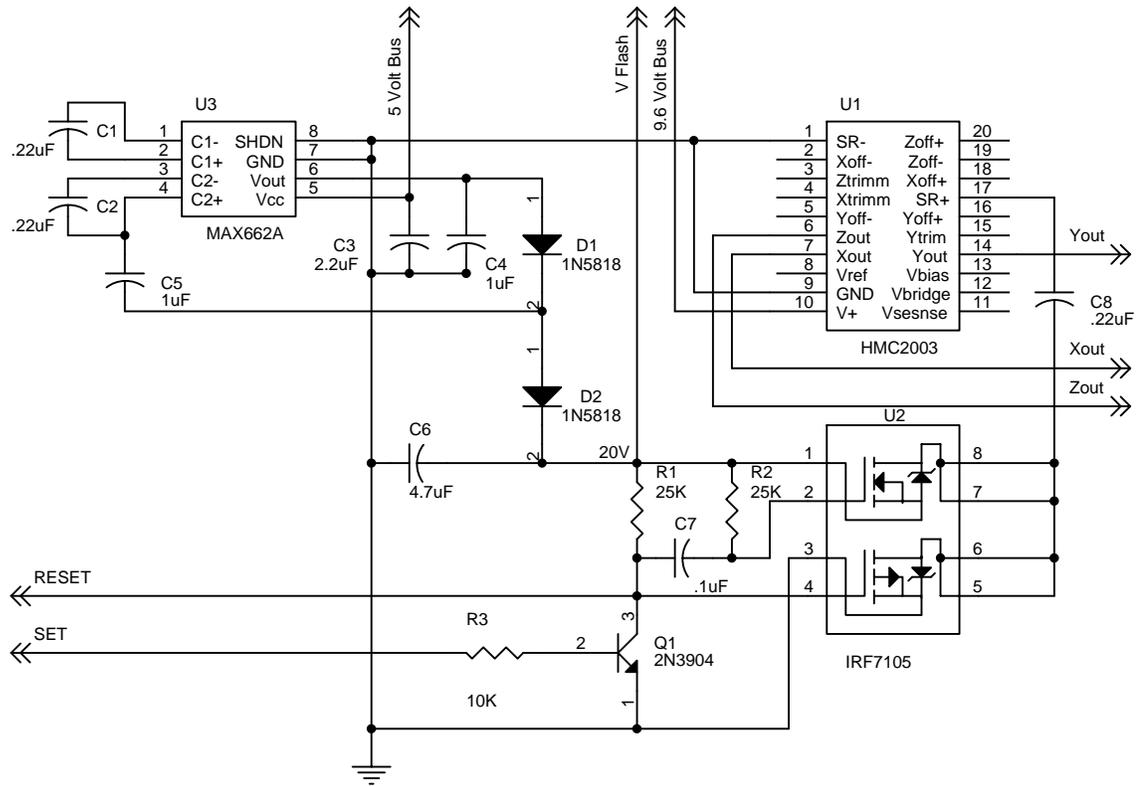


Figure 21: Magnetometer Circuit Schematic

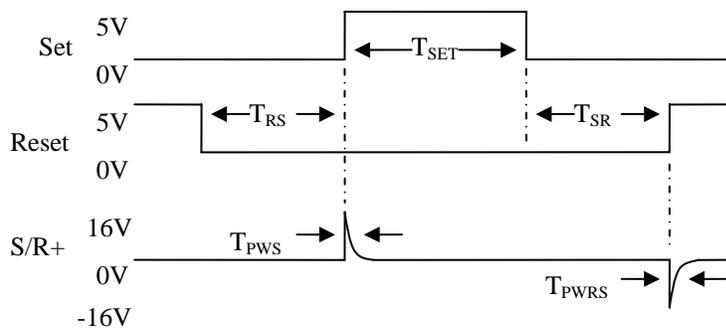


Figure 22: Signal Specifications for “Set,” “Reset,” and “S/R+”

The circuit contains three analog outputs: “Xout”, “Yout” and “Zout”. These are the measurements of the magnetic field in magnetometer coordinate system, as seen in Figure 23. Thus, it is essential that the magnetometer be mounted correctly so that its coordinate system is properly aligned with the fixed body coordinate system of the satellite, as seen in Figure 7. Thus, $x_M = x_B$, $y_M = y_B$, and $z_M = z_B$.

For further information regarding the magnetometer circuit, see the documentation provided by the manufacturers. It is available at <http://www.ssec.honeywell.com/magnetic/datasheets.html>.

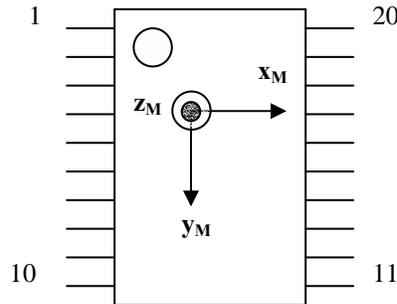


Figure 23: Magnetometer Coordinate System

B.2 Calibration

The calibration of the three-axis magnetic sensor takes place in two steps. The calibration for hard and soft metals is done initially before ION is launched. The calibration for temperature offsets and local magnetic effects must be performed each time the sensor is saturated. Since the sensor is saturated each time the torque coils are fired, this calibration will have to be performed at every reading.

B.2.1 Temperature and local magnetic effects

The Honeywell HMC2003 contains special internal circuitry to compensate for temperature and local magnetic effects. When the set-reset sequence described above is activated, these effects are compensated for. Thus, there is no necessary additional calibration besides running the set-reset sequence described above.

B.2.2 Hard and soft metal calibration

The effects associated with the hard and soft metal disturbances depend on the location and distance of any metal on the spacecraft in relation to the magnetometer. Thus, this calibration may not take place until the satellite has been assembled. It should then be taken to a remote area and calibrated to remove the hard and soft metal effects.

The theory behind the calibration is fairly straightforward. The hard and soft metal interference tends to cause an offset and a warping to the measured values. Thus, the magnetometer can be properly adjusted with a linear model of the following form:

$$B_w = a_w V_w + b_w, w \in \{x, y, z\} \quad (115)$$

The parameter B_w is the magnetic field in the w_{th} direction, V_w is the output of the magnetometer of the w_{th} axis. The parameters a_w and b_w are the scale factors and offset caused by the hard and soft metal effects.

When measuring magnetic fields prior to launch, they can be broken into components parallel and perpendicular to the surface of the Earth. Only the component parallel to the Earth's surface is used in the calibration. The perpendicular component is ignored. In the calibration, the orientation of the satellite is changed to measure the parallel component of the Earth's field in the positive and negative x , y , and z directions. To do this, the satellite is brought to a remote

area with no metallic interference and ION is then placed on its z face. The set-reset sequence described above is used to initially calibrate the sensor for the temperature and local magnetic effects. The satellite is slowly rotated one revolution, and the maximum and minimum magnetometer voltages from the x and y sensors are recorded. The satellite may then be placed on either the x or y face, and the procedure is repeated, measuring only the maximum and minimum voltages from the z sensor.

The key insight behind the calibration is as follows: when the measured voltage is at a maximum or a minimum in any given direction, the component of the Earth's magnetic field parallel to the Earth's surface is completely contained in that reading. This reduces the two-component vector into a scalar. Thus, the six measurements made above result in six independent values of the parallel component of the Earth's magnetic field. These may be represented by the following six resulting equations:

$$B_{parallel} = a_w V_w^{max} + b_w = -(a_w V_w^{min} + b_w), w \in \{x, y, z\} \quad (116)$$

In these six equations, there are six unknowns: the scale factors and the offsets in the x , y , and z directions. $B_{parallel}$ may be obtained by a computer software model, or from a chart, in [7]. After obtaining this value, the solutions to the equations are

$$a_w = \frac{2B_{parallel}}{V_w^{max} - V_w^{min}}, w \in \{x, y, z\} \quad (117)$$

$$b_w = -a_w (V_w^{max} + V_w^{min}), w \in \{x, y, z\} \quad (118)$$

The magnetometer readings always contain some noise, and there is always some local variation in the magnetic field at particular locations. Therefore, in order to ensure accuracy in

calibration, it is necessary to complete this calibration numerous times at various locations and then to average the results.

B.3 Normal Operation Procedure

For ease of explanation, let the three voltages from the outputs of the magnetometer be represented as a vector as follows:

$$\mathbf{v} = \begin{bmatrix} X_{out} \\ Y_{out} \\ Z_{out} \end{bmatrix} \quad (119)$$

Under normal operation, the “Set” and “Reset” signals described in Figure 22 must be sent from the SID to the magnetometer circuitry in order to obtain an accurate reading. During this sequence, SID will store two values of \mathbf{v} . The first is to be recorded following T_{PWS} and before the expiration of T_{SET} . This reading will be referred to as \mathbf{v}_{SET} . The second reading, which will be referred to as \mathbf{v}_{RESET} will be made after T_{PWRS} . To obtain the sample voltage, the following calculation is made:

$$\mathbf{v} = \frac{\mathbf{v}_{SET} - \mathbf{v}_{RESET}}{2} \quad (120)$$

Then the magnetic field estimates may be calculated as

$$\mathbf{b}_m = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & a_z \end{bmatrix} \mathbf{v} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (121)$$

The parameters used in this calculation were obtained from the hard and soft metal calibration, as described above. Clearly, \mathbf{b}_m is the magnetic field measured in the magnetometer fixed coordinate system, as seen in Figure 23. However, since the magnetometer has been

correctly mounted, this is also the same as the fixed body coordinate system, which is seen in Figure 7. So, \mathbf{b}_m is equivalently the geomagnetic field reading taken with respect to the fixed body coordinate system.

APPENDIX C: OVERVIEW OF THE ATTITUDE CONTROL SYSTEM

This appendix consists of a flow diagram for the entire attitude control system, which is found in Figure 24. In this diagram, rounded boxes represent variables, while square boxes represent algorithms. All algorithms include references to portions of the thesis that contain the necessary details to perform them.

To implement this system, there are a number of necessary initial conditions. It is necessary to know the initial time t_0 , sampling time T , and final time t_f . To perform the attitude determination, it is necessary to know the initial velocity \mathbf{v}_0 and radius \mathbf{r}_0 . To begin the attitude estimation algorithm, it is necessary to have initial estimates for ION's attitude $\hat{\mathbf{q}}_0^{\text{B/I}}$ and angular velocity $\hat{\boldsymbol{\omega}}_0^{\text{B/I}}$. It is also necessary to estimate the initial disturbance torque \mathbf{t}_{d0} and the initial covariance matrix \mathbf{P}_0 .

The entire system is implemented in software with the exception of reading the magnetometer and outputting values to torque coil hardware. The magnetometer is sampled every T seconds, beginning with the initial sample time. At each sample, the orbit propagation algorithm computes the location of ION. This information is used to calculate the nominal magnetic field value. The extended Kalman filter compares these values to improve the estimate of the attitude of the system. Finally, the attitude control law computes the necessary parameters to run the torque coils.

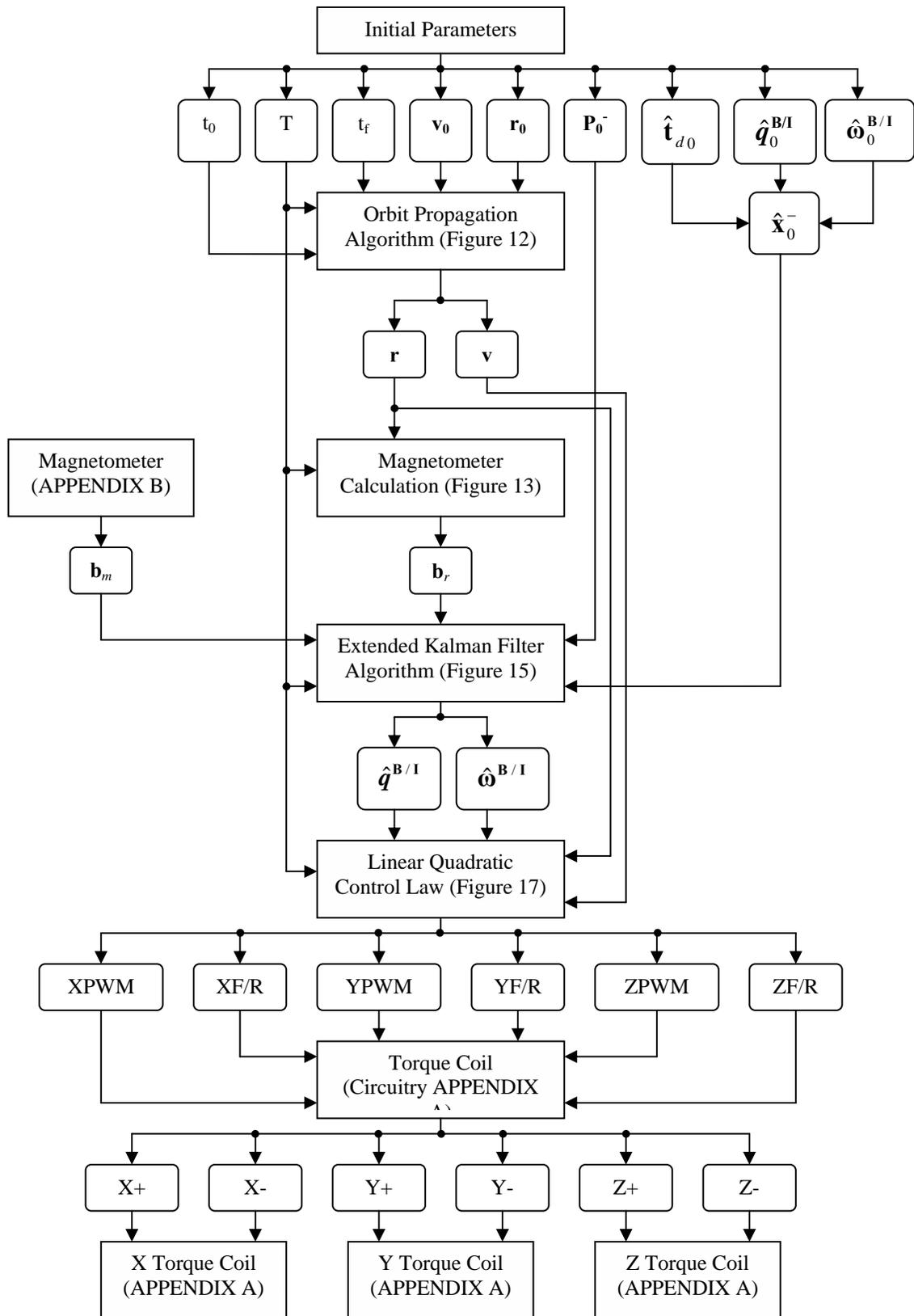


Figure 24: Attitude Control System Flow Diagram

APPENDIX D: CONVERSION BETWEEN THE RADIUS AND VELOCITY VECTORS AND THE ORBITAL ELEMENTS

Conversion between the radius and velocity vectors to the classical orbital elements is a fairly straightforward process. In this paper, only the fundamental equations will be given. For examples of their development, see [6]. See Figure 10 for a picture of the orbit for the location of appropriate vectors defined below.

D.1 From the Radius and Velocity to the Orbital Elements

Given the radius and velocity vectors in the Geocentric Celestial Inertial coordinate system, \mathbf{r} and \mathbf{v} , the first step in finding the orbital elements lies in obtaining the vector \mathbf{h} , which is defined as

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \quad (122)$$

The parameter \mathbf{h} is a constant vector for all position and velocity vectors in a given orbit, and defines the orbit plane. Next, the nodal vector is defined as

$$\mathbf{n} = \mathbf{z}_1 \times \hat{\mathbf{h}} \quad (123)$$

The nodal vector defines the ascending intersection of the reference plane and the orbital plane.

Once these two parameters have been defined, it is possible to directly solve for the orbital elements through the following equations:

$$a = \left(\frac{2}{r} - \frac{v^2}{\mu} \right)^{-1} \quad (124)$$

$$\mathbf{e} = \left((v^2 - \frac{\mu}{r})\mathbf{r} - (\mathbf{r} \bullet \mathbf{v})\mathbf{v} \right) / \mu \quad (125)$$

$$i = \cos^{-1}\left(\frac{h_z}{h}\right) \quad (126)$$

$$\Omega = \begin{cases} \cos^{-1}\left(\frac{n_x}{n}\right) & \text{if } n_y \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{n_x}{n}\right) & \text{if } n_y < 0 \end{cases} \quad (127)$$

$$\omega = \begin{cases} \cos^{-1}\left(\frac{\mathbf{n} \bullet \mathbf{e}}{ne}\right) & \text{if } e_z \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{n} \bullet \mathbf{e}}{ne}\right) & \text{if } e_z < 0 \end{cases} \quad (128)$$

$$f = \begin{cases} \cos^{-1}\left(\frac{\mathbf{e} \bullet \mathbf{r}}{er}\right) & \text{if } \mathbf{r} \bullet \mathbf{v} \geq 0 \\ 2\pi - \cos^{-1}\left(\frac{\mathbf{e} \bullet \mathbf{r}}{er}\right) & \text{if } \mathbf{r} \bullet \mathbf{v} < 0 \end{cases} \quad (129)$$

A few points must be made about these equations. First of all, the eccentricity e is simply the norm of \mathbf{e} as computed in Equation (125). Since the inclination takes on values only from 0 to π , Equation (126) may be used to directly calculate i . However, the angles defined in Equations (127) through (129) take on values of 0 to 2π . Thus, the principle value of the inverse cosine will not guarantee the correct solution. As such, there are two possible solutions depending upon the given conditions to determine whether the principle value solution is correct or not.

D.2 From the Orbital Elements to the Radius and Velocity

The radius vector \mathbf{r} in the inertial reference frame may be found by the following equation:

$$\mathbf{r} = \frac{a(1-e^2)}{1+e\cos f} \begin{bmatrix} \cos(\omega+f)\cos\Omega - \sin(\omega+f)\cos i \sin\Omega \\ \cos(\omega+f)\sin\Omega + \sin(\omega+f)\cos i \cos\Omega \\ \sin(\omega+f)\sin i \end{bmatrix} \quad (130)$$

The velocity vector \mathbf{v} in the inertial reference frame is obtained as follows:

$$\mathbf{v} = \left(\frac{\mu}{a(1-e^2)} \right)^{1/2} \begin{bmatrix} -\cos\Omega(\sin(\omega+f) + e\sin\omega) - \sin\Omega\cos i(\cos(\omega+f) + e\cos\omega) \\ -\sin\Omega(\sin(\omega+f) + e\sin\omega) + \cos\Omega\cos i(\cos(\omega+f) + e\cos\omega) \\ \sin i(\cos(\omega+f) + e\cos\omega) \end{bmatrix} \quad (131)$$

APPENDIX E: CONVERSIONS BETWEEN COORDINATE SYSTEMS

As outlined in the third section, it is possible to convert between various coordinate systems by obtaining the attitude matrix. This section will give the various attitude matrices necessary for these conversions. Then, performing the conversions is as simple as performing the appropriate matrix multiplication. For example, to convert between coordinate systems C and D , the same vector \mathbf{v} may be written as a linear combination of both bases:

$$\mathbf{v} = [\mathbf{x}_C \quad \mathbf{y}_C \quad \mathbf{z}_C] \mathbf{v}_C = [\mathbf{x}_D \quad \mathbf{y}_D \quad \mathbf{z}_D] \mathbf{v}_D \quad (132)$$

Both \mathbf{v}_C and \mathbf{v}_D are the appropriate three-dimensional vectors necessary to obtain \mathbf{v} in the bases of coordinate systems C and D , respectively. If $\mathbf{A}^{C/D}$ is the attitude matrix representing the orientation of coordinate system C to D , then the following equations provide the conversions from \mathbf{v}_C to \mathbf{v}_D :

$$\mathbf{v}_C = \mathbf{A}^{C/D} \mathbf{v}_D \quad (133)$$

$$\mathbf{v}_D = \mathbf{A}^{D/C} \mathbf{v}_C = (\mathbf{A}^{C/D})^T \mathbf{v}_C \quad (134)$$

E.1 The Attitude Matrix for Converting Between the Geocentric Celestial Inertial and Orbital Reference Coordinate Systems

See Figure 5 for a view of the Geocentric Celestial Inertial coordinate system and Figure 8 for the Orbital Reference coordinate system. The attitude matrix defining these coordinate systems is a function of the radius and velocity vectors at the satellite's precise location. Let \mathbf{r} and \mathbf{v} be the radius and velocity vectors in the inertial reference frame. The vector \mathbf{z}_R is the unit vector in the opposite direction of the radius vector:

$$\mathbf{z}_R = -\frac{\mathbf{r}}{r} = -\frac{1}{r} \begin{bmatrix} \mathbf{x}_I & \mathbf{y}_I & \mathbf{z}_I \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (135)$$

The vector \mathbf{x}_R is the unit vector in the direction of the projection of the velocity vector into the orthogonal subspace of the radius vector. This can be computed as

$$\mathbf{x}_R = \frac{\mathbf{v} - \frac{\mathbf{r} \cdot \mathbf{v}}{r^2} \mathbf{r}}{\left\| \mathbf{v} - \frac{\mathbf{r} \cdot \mathbf{v}}{r^2} \mathbf{r} \right\|} = \begin{bmatrix} \mathbf{x}_I & \mathbf{y}_I & \mathbf{z}_I \end{bmatrix} \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix} \quad (136)$$

The vector \mathbf{y}_R is defined by the cross product of \mathbf{z}_R and \mathbf{x}_R :

$$\mathbf{y}_R = \mathbf{z}_R \times \mathbf{x}_R = \begin{bmatrix} \mathbf{x}_I & \mathbf{y}_I & \mathbf{z}_I \end{bmatrix} \left(-\frac{1}{r} \begin{bmatrix} x_x \\ x_y \\ x_z \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right) = \begin{bmatrix} \mathbf{x}_I & \mathbf{y}_I & \mathbf{z}_I \end{bmatrix} \begin{bmatrix} y_x \\ y_y \\ y_z \end{bmatrix} \quad (137)$$

Substituting these equations yields the appropriate attitude matrix, $\mathbf{A}^{R/I}$:

$$\begin{bmatrix} \mathbf{x}_R \\ \mathbf{y}_R \\ \mathbf{z}_R \end{bmatrix} = \begin{bmatrix} x_x & x_y & x_z \\ y_x & y_y & y_z \\ -\frac{r_x}{r} & -\frac{r_y}{r} & -\frac{r_z}{r} \end{bmatrix} \begin{bmatrix} \mathbf{x}_I \\ \mathbf{y}_I \\ \mathbf{z}_I \end{bmatrix} = \mathbf{A}^{R/I} \begin{bmatrix} \mathbf{x}_I \\ \mathbf{y}_I \\ \mathbf{z}_I \end{bmatrix} \quad (138)$$

E.2 Obtaining the Angular Velocity of the Reference Frame With Respect to the Inertial Frame.

Since the fixed body coordinate system is not dependent upon the orientation of the satellite, its angular velocity relative to the geocentric inertial coordinate system is only a function of the radius and velocity vectors. It is simply computed as follows:

$$\boldsymbol{\omega}^{R/I} = -\frac{\|\mathbf{v} \times \mathbf{r}\|}{r^2} \mathbf{y}_R \quad (139)$$

Note that for a circular orbit, \mathbf{r} and \mathbf{v} are always perpendicular to each other and have the same magnitude. Thus, in this case, the angular velocity is constant for all points in the orbit.

E.3 The Attitude Matrix for Converting Between the Geocentric Celestial Inertial and Geocentric Fixed Coordinate Systems

See Figure 5 for a picture of the Geocentric Celestial Inertial coordinate system and Figure 6 for the Geocentric Fixed coordinate system. Since both the Geocentric Fixed and Geocentric Celestial Inertial coordinate systems have the third axis from the center of the Earth to the north pole, transformation between the two systems is simply a rotation around this axis of an angle θ as shown in Figure 25. This angle is known as Greenwich Mean Sidereal Time. It is a well-known fact that the Earth rotates about the axis from its center to the North Pole at a rate of one rotation for every 23 h, 56 min, and 4.1 s. Since the Geocentric Fixed coordinate system's axes are fixed within the earth, they also rotate at this same rate. Thus, Greenwich Mean Sidereal Time also rotates at this same rate. It can be found as a simple function of time according to [17] as

$$\theta = 4.89496121274 + 6.30038809899 * \Delta d \quad (140)$$

The parameter Δd is the total time measured in days, including partial days as decimal equivalents, which has passed since the epoch time. The epoch is J2000, which means d_0 is January 1st, 2000, at 12:00:00 noon. Equation (140) does not take into account the precession or nutation of the Earth's equatorial axis. However, since these terms are small, this will provide a reasonable approximation in the calculation of θ .

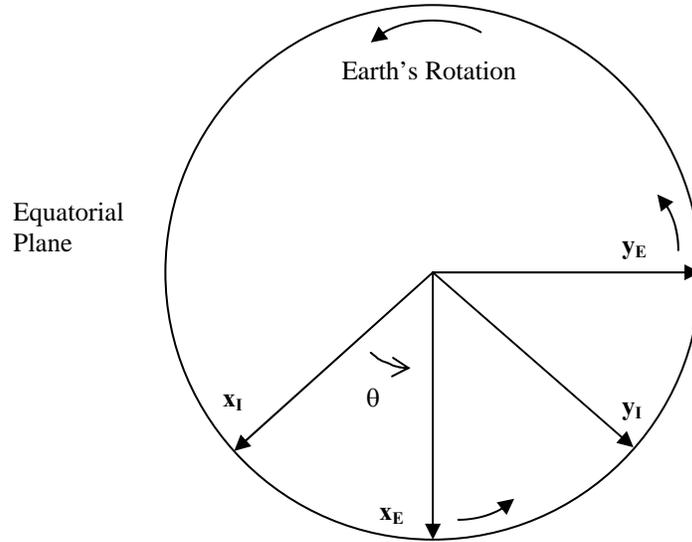


Figure 25: The Earth's Equatorial Plane

Once θ has been calculated, it is trivial to calculate the matrix representing the attitude of the Geocentric Fixed coordinate system with respect to the Geocentric Celestial Inertial coordinate system. It is as follows:

$$\mathbf{A}^{E/I} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (141)$$

E.4 The Satellite Fixed Body Coordinate System

The attitude of the satellite representing the orientation of the Satellite Fixed Body Coordinate System is usually represented with a quaternion vector. Thus, to make conversions involving fixed body coordinate system, it is necessary to either obtain the attitude matrix from the quaternion as outlined in APPENDIX F or use the quaternion multiplication rule in Equation (30).

APPENDIX F: CONVERSIONS BETWEEN ATTITUDE REPRESENTATIONS

It is more convenient to use various attitude representations for different tasks. The attitude matrix is good for vector conversions. The quaternion is ION's chief attitude representation, primarily because of its ease for attitude propagation. The Euler angles provide an intuitive way to visualize results. Thus, it is only natural to want to convert between these various systems when performing different functions. This section contains the necessary information to convert between these three representations. This information is available in any standard orbital mechanics text such as [1], [4], or [18].

Throughout this section, the following notation will be used. Let $\mathbf{A}^{C/D}$ be the attitude matrix, $\mathbf{q}^{C/D}$ be the quaternion, and ϕ , θ , and ψ be the Euler angles associated with the attitude of coordinate system C with respect to coordinate system D . Let $\mathbf{A}^{C/D}$ be represented as

$$\mathbf{A}^{C/D} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad (142)$$

Let $\mathbf{q}^{C/D}$ be represented as

$$\mathbf{q}^{C/D} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (143)$$

F.1 Conversion from the Quaternion Vector to the Attitude Matrix

It is straightforward to convert from the quaternion to the attitude matrix. This calculation is as follows:

$$\mathbf{A}^{C/D} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 - q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (144)$$

F.2 Conversion from the Attitude Matrix to the Quaternion Vector

Conversion from the attitude matrix to the quaternion vector is not as trivial as the previous conversion. One way to make this conversion is as follows:

$$q_4 = \pm \frac{1}{2} \sqrt{1 + A_{11} + A_{22} + A_{33}} \quad (145)$$

$$q_1 = (4q_4)^{-1} (A_{23} - A_{32}) \quad (146)$$

$$q_2 = (4q_4)^{-1} (A_{31} - A_{13}) \quad (147)$$

$$q_3 = (4q_4)^{-1} (A_{12} - A_{21}) \quad (148)$$

The sign ambiguity in Equation (145) is irrelevant. Either answer provides a correct representation of the quaternion vectors. For convenience, the positive sign may always be chosen such that q_4 is positive. The only drawback from this method is that if q_4 is very small, the results obtained from Equations (146) through (148) will yield high numerical errors.

However, in this case, it is possible to solve directly for one of the other parameters as follows:

$$q_1 = \pm \frac{1}{2} \sqrt{1 + A_{11} - A_{22} - A_{33}} \quad (149)$$

$$q_2 = \pm \frac{1}{2} \sqrt{1 - A_{11} + A_{22} - A_{33}} \quad (150)$$

$$q_3 = \pm \frac{1}{2} \sqrt{1 - A_{11} - A_{22} + A_{33}} \quad (151)$$

Each of Equations (149) through (151) will have its own resulting set of equations similar to Equations (146) through (148). However, deriving these equations by manipulating Equation (144) is trivial; therefore, the results are omitted from this appendix for the sake of brevity.

F.3 Conversion from the Euler Angles to the Attitude Matrix

Euler Angles are based upon taking successive rotations about three axes. There are different sets of Euler angles depending upon which axes are chosen and what order they are rotated about. For each different set of Euler angles, there is a different conversion. The results presented here will be for the “3-2-1” rotation order which is discussed in the third section. It is a straightforward process to convert from these angles to the attitude matrix:

$$\mathbf{A}^{C/D} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (152)$$

F.4 Conversion from the Attitude Matrix to the Euler Angles

Unfortunately, converting from the attitude matrix to the Euler angles is not as trivial as vice versa. The conversions can be obtained by manipulating Equation (152) and using trigonometric inverses to obtain the necessary angles. The chief problem with this method is that the inverse trigonometric functions have quadrant ambiguities. Since the Euler angles are used primarily for reference purposes, it is probably sufficient for most cases merely to ignore the ambiguities. If this is done, the following equations may be used to solve for the Euler angles.

The roll angle is

$$\phi = \arctan\left(\frac{A_{32}}{A_{33}}\right) \quad (153)$$

The pitch angle is

$$\theta = -\arcsin A_{13} \quad (154)$$

The yaw angle is

$$\psi = \arctan\left(\frac{A_{21}}{A_{11}}\right) \quad (155)$$

APPENDIX G: QUATERNION MATHEMATICS

Quaternions are one useful method of quantifying the attitude of a spacecraft. This appendix introduces the concept of the quaternion and outlines some of the mathematical rules necessary to work with quaternions. Simply put, a quaternion is a vector with four components which is defined as follows:

$$\mathbf{q} = [i \quad j \quad k \quad 1] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = q_4 + iq_1 + jq_2 + kq_3 \quad (156)$$

In this formulation, i , j , and k are numbers, analogous to the “ i ” of imaginary numbers, for which the following laws apply:

$$i^2 = j^2 = k^2 = -1 \quad (157)$$

$$ij = -ji = k \quad (158)$$

$$jk = -kj = i \quad (159)$$

$$ki = -ik = j \quad (160)$$

Note that multiplication between these “imaginary” parameters is not commutative. The conjugate of \mathbf{q} is denoted \mathbf{q}^* as follows:

$$\mathbf{q}^* = q_4 - iq_1 - jq_2 - kq_3 \quad (161)$$

The vector part of the quaternion is typically denoted as

$$\mathbf{q} = iq_1 + jq_2 + kq_3 \quad (162)$$

Quaternion multiplication is defined by using the rules of Equations (157) through (160)

to obtain the following result:

$$\begin{aligned} \mathbf{q}'' = \mathbf{q} \otimes \mathbf{q}' &= (q_4 + iq_1 + jq_2 + kq_3)(q'_1 + iq'_2 + jq'_3 + kq'_4) = \\ &= -q_1q'_1 - q_2q'_2 - q_3q'_3 + q_4q'_4 + i(q_1q'_4 + q_2q'_3 - q_3q'_2 + q_4q'_1) + \\ &= j(-q_1q'_3 + q_2q'_4 + q_3q'_1 + q_4q'_2) + k(q_1q'_2 - q_2q'_1 + q_3q'_4 + q_4q'_3) \end{aligned} \quad (163)$$

The norm of \mathbf{q} is defined as

$$|\mathbf{q}| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (164)$$

For computational purposes, \mathbf{q} is often denoted as a vector. The row vector containing one and the “imaginary” component is only implied. Thus, \mathbf{q} is represented alternatively as

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \quad (165)$$

In this setting, the quaternion multiplication performed in Equation (163) may be represented through a matrix multiplication as

$$\mathbf{q}'' = \mathbf{q} \otimes \mathbf{q}' = \begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} q'_1 \\ q'_2 \\ q'_3 \\ q'_4 \end{bmatrix} \quad (166)$$

APPENDIX H: SIMULATION OVERVIEW

One of the chief aspects involved in the attitude control design is verification of the design through simulation. An entire simulator was built from the ground up in the Matlab and Simulink environments for this purpose. Since one of the chief aspects of this simulation involves numerically integrating the satellite dynamics and kinematics equations, and Simulink is a great tool for performing numerical integrations, all of the attitude propagation is performed in Simulink. Orbit propagation and magnetic field models were created in the Matlab work environment.

H.1 Flow Diagram of the Simulations

The flow diagram for the simulations is seen in Figure 26. This diagram is quite similar to the diagram of the attitude control system in Figure 24. The biggest change is that it is necessary to simulate the attitude of the satellite with an attitude propagation algorithm. To increase the accuracy of these simulations, it is necessary to simulate disturbance torques that are present in the space environment. The only other difference is that it is also necessary to simulate the magnetometer readings. This is done rotating the magnetic field vector to the fixed body coordinate system.

H.2 Magnetometer Simulation

To simulate the vector the magnetometer would read, only a simple matrix multiplication is necessary.

$$\mathbf{b}_m = \mathbf{A}^{B/I} \mathbf{b}_i \quad (167)$$

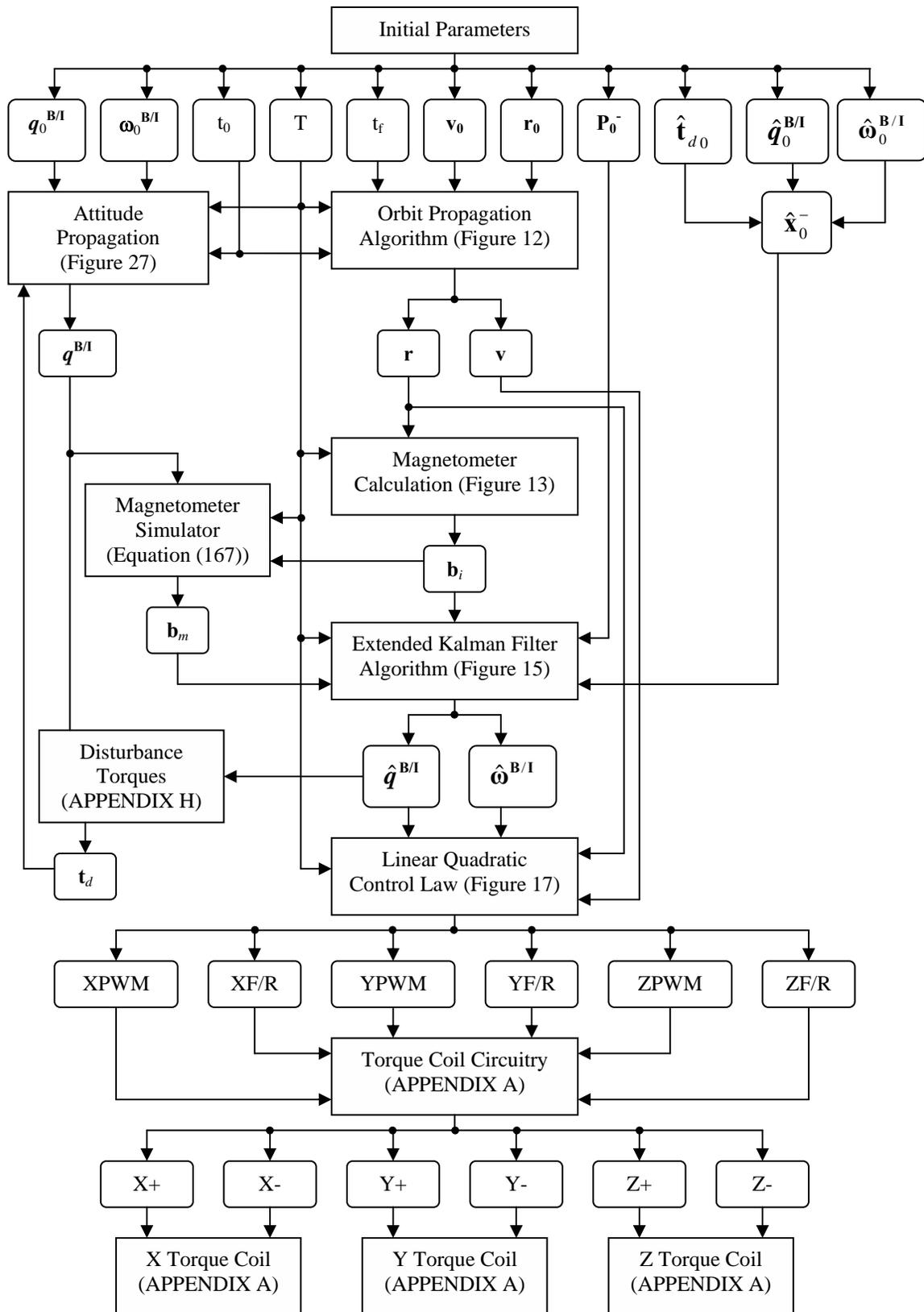


Figure 26: Simulation Flow Diagram

The vector \mathbf{b}_m is the simulated magnetometer reading, and the vector \mathbf{b}_i is the geomagnetic field in the inertial coordinate system. The matrix $\mathbf{A}^{B/I}$ can be computed directly from $\mathbf{q}^{B/I}$ as given in Equation (144).

H.3 Modeling Disturbance Torques

In the dynamics equation, Equation (39), \mathbf{t}_T is the total applied torque. This will include the control torque generated by the coils as well as disturbance torques. To increase the accuracy of simulations, it is important to model the disturbance torques as closely as possible. The major sources for disturbance torques given in [1] are gravity gradient torque, solar radiation torque, aerodynamic torque, and magnetic disturbance torque. Of these four, the magnetic disturbance torque may be considered as negligible compared to the torque produced by the coils. The solar radiation torque is difficult to model because it depends both upon the solar radiation directly from the Sun, and from reflections off the Earth, moon, and other satellites. Since these effects are minimal, due to the small size of ION, they will also be ignored. However, the other two sources of disturbance torque will be modeled to some extent.

H.4 Gravity Gradient Torque

Gravity gradient torque is caused by the variation in the Earth's gravitational field. Since the gravitational field decreases as one over the square distance from the earth, the entire satellite is not attracted to the earth with the same gravitational force. As such, over time, this torque will tend to align the satellite's long axis with the Earth. A basic first order approximation to the gravity gradient torque is as follows:

$$\mathbf{t}_{gg} = \frac{3\mu}{r_e^3} \hat{\mathbf{r}}_e \times (\tilde{\mathbf{I}} \hat{\mathbf{r}}_e) \quad (168)$$

The parameter μ is the same gravitational constant used in Equation (46) and \mathbf{r}_e is the vector from the ION's center of mass to the Earth's center of mass in the Fixed Body coordinate system. The matrix $\tilde{\mathbf{I}}$ is the inertia matrix which was introduced in Equation (36). If \mathbf{r} is the vector from the earth to the satellite in the inertial coordinate system, then \mathbf{r}_e is found by the following equation:

$$\mathbf{r}_e = -A^{B/I} \mathbf{r} \quad (169)$$

H.5 Aerodynamic Torque

The aerodynamic torque is caused by interaction between the spacecraft and the atmosphere. Since the satellite is not traveling in a perfect vacuum, there is some atmospheric drag acting on the satellite. Not surprisingly, this force also creates a torque that causes the satellite to rotate. Clearly, the aerodynamic torque is largely dependent upon the density of the atmosphere. As such, for larger orbits, this torque becomes negligible as the density of the atmosphere decreases. However, for low orbits, aerodynamic torque may be quite substantial.

The equation that governs this disturbance torque, as given in [1], is as follows:

$$\begin{aligned} \mathbf{t}_a = & \frac{1}{2} C_D \rho v_0^2 \int (\mathbf{a}_n \cdot \hat{\mathbf{v}}_0) (\hat{\mathbf{v}}_0 \times \mathbf{r}_s) dA + \\ & \frac{1}{2} C_D \rho v_0 \int \left(\mathbf{a}_n \cdot (\boldsymbol{\omega}^{B/I} \times \mathbf{r}_s) (\hat{\mathbf{v}}_0 \times \mathbf{r}_s) + (\mathbf{a}_n \cdot \hat{\mathbf{v}}_0) \{ (\boldsymbol{\omega}^{B/I} \times \mathbf{r}_s) \times \mathbf{r}_s \} \right) dA \end{aligned} \quad (170)$$

The parameter C_D is the drag coefficient, which is a number between 1 and 2, with the upper limit typically chosen when no information is available. The parameter ρ is the atmospheric density, which is a function of the altitude of the spacecraft. At 700 km, it is approximately $3.614 \times 10^{-14} \text{ kg/m}^3$. The other parameters are relatively simple: \mathbf{a}_n is the unit vector normal to the surface of the satellite; \mathbf{v}_0 is the velocity of the satellite in the fixed body

reference frame. Thus, if \mathbf{v} is the velocity of the satellite in the inertial frame, \mathbf{v}_0 is computed as follows:

$$\mathbf{v}_0 = \mathbf{A}^{B/I} \mathbf{v} \quad (171)$$

The vector \mathbf{r}_s is the vector from the center of mass of the satellite to the infinitesimal area, dA and $\boldsymbol{\omega}^{B/I}$ is the angular velocity of the satellite. Note that Equation (170) must be evaluated for each surface of the satellite where $\mathbf{a}_n \bullet \mathbf{v}_0$ is greater than 0. In other words, the atmospheric torque only acts on those sides of the satellite that are in the same general direction as the velocity.

The first term in Equation (170) governs the torque produced by the general translation of the satellite. The second integral is caused by the spacecraft's angular velocity. As long as $\omega r \ll v_0$, then this term may be ignored. This will be the case for satellites with small angular velocities, which is the case under normal attitude control.

Since ION is a cube, it is fairly straightforward to evaluate Equation (170), under the circumstances described in the previous paragraph. Letting

$$\mathbf{v}_0 = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (172)$$

$$\mathbf{r}_s = \begin{bmatrix} x - \Delta x \\ y - \Delta y \\ z - \Delta z \end{bmatrix} \quad (173)$$

The distances Δx , Δy , and Δz are the from the satellite's geometric center to its center of mass.

Using these definitions for Equations (172) and (173),

$$\mathbf{v}_0 \times \mathbf{r}_s = \begin{bmatrix} v_y(z - \Delta z) - v_z(y - \Delta y) \\ v_z(x - \Delta x) - v_x(z - \Delta z) \\ v_x(y - \Delta y) - v_y(x - \Delta x) \end{bmatrix} \quad (174)$$

Since $\mathbf{a}_n \cdot \hat{\mathbf{v}}_0$ is a constant, it may be pulled out of the integral. After substituting in Equation (174), then (170) may be written as

$$\mathbf{t}_a = \frac{1}{2} C_D \rho (\mathbf{a}_n \cdot \mathbf{v}_0) \int \begin{bmatrix} v_y(z - \Delta z) - v_z(y - \Delta y) \\ v_z(x - \Delta x) - v_x(z - \Delta z) \\ v_x(y - \Delta y) - v_y(x - \Delta x) \end{bmatrix} dA \quad (175)$$

To make the development more concrete, the results will be derived for the drag on the +X side of the craft. Then the x component of the vector is always constant and equal to $X_l/2$, which is the distance from the edge of the craft to the geometric center. Also, $\mathbf{a}_n \cdot \mathbf{v}_0$ is simply v_x , the component in the x direction. Substituting this information along with the result for the differential area, Equation (175) becomes the following double integral:

$$\mathbf{t}_{a+x} = \frac{1}{2} C_D \rho v_x \int_{-Y_l/2}^{Y_l/2} \int_{-Z_l/2}^{Z_l/2} \begin{bmatrix} v_y(z - \Delta z) - v_z(y - \Delta y) \\ v_z(\frac{X_l}{2} - \Delta x) - v_x(z - \Delta z) \\ v_x(y - \Delta y) - v_y(\frac{X_l}{2} - \Delta x) \end{bmatrix} dz dy \quad (176)$$

After evaluating the integrals the following result is obtained:

$$\mathbf{t}_{a+x} = \frac{1}{2} C_D \rho v_x \begin{bmatrix} -v_y Y_l Z_l \Delta z + v_z Y_l Z_l \Delta y \\ v_z (\frac{X_l}{2} - \Delta x) Y_l Z_l + v_x Y_l Z_l \Delta z \\ -v_x Y_l Z_l \Delta y - v_y (\frac{X_l}{2} - \Delta x) Y_l Z_l \end{bmatrix} \quad (177)$$

Notice that the vector term may be simply written as

$$\begin{bmatrix} -v_y \Delta z + v_z \Delta y \\ v_z \left(\frac{X_l}{2} - \Delta x \right) + v_x \Delta z \\ -v_x \Delta y - v_y \left(\frac{X_l}{2} - \Delta x \right) \end{bmatrix} = \begin{bmatrix} \Delta x - \frac{X_l}{2} \\ \Delta y \\ \Delta z \end{bmatrix} \times \mathbf{v}_0 \quad (178)$$

Substituting Equation (178) into Equation (177) and pulling out $Y_l Z_l$, which are common to all terms, yields the final result:

$$\mathbf{t}_{a+x} = \frac{1}{2} C_D \rho v_x Y_l Z_l \left(\begin{bmatrix} \Delta x - \frac{X_l}{2} \\ \Delta y \\ \Delta z \end{bmatrix} \times \mathbf{v}_0 \right) \quad (179)$$

To obtain the result for the $-X$ side of the craft, the only change is to replace X_l by $-X_l$ in the \mathbf{r}_s vector and $\mathbf{a}_n \bullet \mathbf{v}_0$ now becomes $-v_x$ instead of v_x . Thus, the result is

$$\mathbf{t}_{a-x} = -\frac{1}{2} C_D \rho v_x Y_l Z_l \left(\begin{bmatrix} \Delta x + \frac{X_l}{2} \\ \Delta y \\ \Delta z \end{bmatrix} \times \mathbf{v}_0 \right) \quad (180)$$

Note that the value $-v_x$ will be a positive number, since v_x must be negative for their to be a contribution to the aerodynamic torque from the $-X$ side. Combining this fact with the knowledge that there can only be a contribution to the aerodynamic torque from either the $+$ or $-X$ side, but not both, it is possible to combine Equations (179) and (180) into one final result representing the contribution from the x direction:

$$\mathbf{t}_{ax} = \frac{1}{2} C_D \rho |v_x| Y_l Z_l \left(\begin{bmatrix} \Delta x - \text{sgn}|v_x| \frac{X_l}{2} \\ \Delta y \\ \Delta z \end{bmatrix} \times \mathbf{v}_0 \right) \quad (181)$$

Similar results may be obtained for the Y and Z sides. They are as follows:

$$\mathbf{t}_{ay} = \frac{1}{2} C_D \rho |v_y| X_l Z_l \left(\begin{bmatrix} \Delta x \\ \Delta y - \text{sgn}|v_y| \frac{Y_l}{2} \\ \Delta z \end{bmatrix} \times \mathbf{v}_0 \right) \quad (182)$$

$$\mathbf{t}_{az} = \frac{1}{2} C_D \rho |v_z| X_l Y_l \left(\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z - \text{sgn}|v_z| \frac{Z_l}{2} \end{bmatrix} \times \mathbf{v}_0 \right) \quad (183)$$

Finally, the total aerodynamic torque is obtained by adding Equations (181) through (183):

$$\mathbf{t}_a = \mathbf{t}_{ax} + \mathbf{t}_{ay} + \mathbf{t}_{az} \quad (184)$$

H.6 Disturbance Torque Results

Using these results obtained for the disturbance torques, it is possible to update the dynamics equation. Define the total disturbance torque \mathbf{t}_d as

$$\mathbf{t}_d = \mathbf{t}_{gg} + \mathbf{t}_a \quad (185)$$

The vector \mathbf{t}_{gg} is defined in Equation (168), and \mathbf{t}_a is defined in Equation (184). Using these definitions, then Equation (39) may be updated as follows:

$$\left\{ \frac{d\mathbf{h}}{dt} \right\}_B = \mathbf{t}_m + \mathbf{t}_d - \boldsymbol{\omega}^{B/I} \times \mathbf{h} \quad (186)$$

The vector \mathbf{t}_m is the magnetic torque as defined in Equation (3).

H.7 Attitude Propagation Algorithm

Clearly the dynamics and kinematics equations will evolve naturally by themselves when ION is in orbit. Nevertheless, to run the simulation, it is necessary to propagate the spacecraft's attitude. This algorithm is seen in Figure 27. In some situations, it is helpful to propagate ION's attitude with respect to the inertial frame. In others, it is more helpful to propagate the attitude with respect to the orbital reference frame. For the first case, it is possible to numerically integrate Equation (41) using $\omega^{B/I}$ and $q^{B/I}$. For the later case, it is necessary to first obtain $\omega^{B/R}$ by using the rules outlined in Equations (31) and (32) as follows:

$$\omega^{B/R} = \omega^{B/I} - \omega^{R/I} \quad (187)$$

Note that $\omega^{R/I}$ is a function of the radius and velocity vectors as given in Equation (139). After obtaining $\omega^{B/I}$, Equation (41) is integrated to obtain $q^{B/I}$.

Attitude Propagation Algorithm
Inputs: Initial time t_0 , final time t_d , the initial angular velocity $\omega^{B/I}_0$, and the initial attitude q . If the attitude is with respect to the body frame, it is necessary to have the initial radius r_0 and velocity v_0 vectors as well.
Step 1. Numerically integrate Equation (186) until time t_d to obtain $\omega^{B/I}(t)$.
Step 2. If the desired attitude is with respect to the inertial frame, integrate Equation (41) to obtain $q^{B/I}$ at t_d .
Step 3. Otherwise, use the orbit propagation algorithm in Figure 12 to obtain $r(t)$ and $v(t)$ from t_0 to t_d .
Step 3. Calculate $\omega^{R/I}(t)$ according to Equation (139).
Step 4. Numerically integrate Equation (41) to obtain $q^{B/R}$ at t_d .
Step 5. If necessary, convert to a different attitude representation as outlined in APPENDIX F.

Figure 27: Attitude Propagation Algorithm

APPENDIX I: DERIVATION OF THE LINEARIZATIONS FOR THE EXTENDED KALMAN FILTER AND LINEAR QUADRATIC REGULATOR

Throughout these derivations some shorthand notation shall be employed. The angular velocity will be written as

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (188)$$

The quaternion representing the attitude of the satellite is

$$\mathbf{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (189)$$

I.1 Linear Representation of the Extended Kalman Filter

In this section, subscripts on the quaternions and angular velocities of the body frame with respect to the inertial frame are omitted for notational convenience. The equations which propagate the measurement update for the extended Kalman filter make use of the nine-element state vector defined in Equation (76). It is restated below:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \Delta \mathbf{q} \\ \boldsymbol{\omega} \\ \mathbf{t}_d \end{bmatrix} \quad (190)$$

The vector \mathbf{t}_d is the disturbance torque in the fixed body reference frame. The system dynamics are given in Equation (83). They are restated as

$$\begin{bmatrix} \dot{\mathbf{q}}^{B/I} \\ \dot{\boldsymbol{\omega}}^{B/I} \\ \dot{\mathbf{t}}_d \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}^{B/I}) \mathbf{q}^{B/I} \\ \tilde{\mathbf{I}}^{-1} (\mathbf{t}_m + \mathbf{t}_d - \boldsymbol{\omega}^{B/I} \tilde{\mathbf{I}}^{-1} \times (\tilde{\mathbf{I}} \boldsymbol{\omega}^{B/I})) \\ \mathbf{0} \end{bmatrix} \quad (191)$$

The goal of this section is to obtain the linearized system dynamics \mathbf{F} and linearized output equation \mathbf{H} . The matrix \mathbf{F} is found in Equation (63):

$$\mathbf{F}(t, \mathbf{u}_d) = \nabla_{\mathbf{x}} \dot{\tilde{\mathbf{x}}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} \quad (192)$$

Clearly to proceed, it is necessary to find the dynamics for the perturbation $\Delta \mathbf{q}$. After doing so, it is possible to take the gradients and evaluate along the estimated trajectory. The small three-element quaternion $\Delta \mathbf{q}$ is defined in Equation (75). It can be rewritten as

$$\begin{bmatrix} \Delta \mathbf{q} \\ 1 \end{bmatrix} = \mathbf{q} \otimes \hat{\mathbf{q}}^{-1} \quad (193)$$

The quaternion inverse for a quaternion \mathbf{q} is simply defined as a quaternion such that

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (194)$$

It may be easily seen by the chain rule that the derivative of a quaternion inverse is

$$\dot{\mathbf{q}}^{-1} = -\mathbf{q}^{-1} \otimes \dot{\mathbf{q}} \otimes \mathbf{q}^{-1} \quad (195)$$

After applying this result to Equation (193) and performing some algebraic manipulations, the following result is obtained:

$$\Delta \dot{\mathbf{q}} = \frac{1}{2} \Delta \mathbf{q} \times (\hat{\boldsymbol{\omega}} + \boldsymbol{\omega}) + \frac{1}{2} (\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}) \quad (196)$$

Taking the gradient yields

$$\nabla_{\hat{\mathbf{x}}}(\Delta\dot{\mathbf{q}}) = \left[-\frac{1}{2}\mathbf{X}(\hat{\boldsymbol{\omega}} + \boldsymbol{\omega}) \quad \frac{1}{2}(\mathbf{X}(\Delta\mathbf{q}) + \mathbf{I}) \quad \mathbf{0} \right] \quad (197)$$

The transform \mathbf{X} is the usual skew symmetric matrix associated with the cross product, as defined in Equation (102). Evaluating Equation (197) along the estimated trajectory yields the linear equation

$$\nabla_{\hat{\mathbf{x}}}(\Delta\dot{\mathbf{q}})|_{\mathbf{x}=\hat{\mathbf{x}}} = \left[-\mathbf{X}(\hat{\boldsymbol{\omega}}) \quad \frac{1}{2}\mathbf{I} \quad \mathbf{0} \right] \quad (198)$$

Note that when evaluating along the estimated trajectory, $\Delta\mathbf{q}$ goes to $\mathbf{0}$, as this represents the deviation between the state and the estimate.

Next, the differential equation for the angular velocity is examined. When substituting for the inertial matrix as defined in Equations (35) through (37) into Equation (40), the following result is obtained:

$$\dot{\boldsymbol{\omega}} = \tilde{\mathbf{I}}^{-1} \left(\mathbf{t}_m + \mathbf{t}_d - \begin{bmatrix} I_{yz}(-\omega_y^2 + \omega_z^2) + (I_z - I_y)\omega_y\omega_z - I_{xz}\omega_x\omega_y + I_{xy}\omega_x\omega_z \\ I_{xz}(\omega_x^2 - \omega_z^2) + (I_x - I_z)\omega_x\omega_z + I_{yz}\omega_x\omega_y - I_{xy}\omega_y\omega_z \\ I_{xy}(-\omega_x^2 + \omega_y^2) + (I_y - I_x)\omega_x\omega_y - I_{yz}\omega_x\omega_z + I_{xz}\omega_y\omega_z \end{bmatrix} \right) \quad (199)$$

Taking the gradient yields

$$\nabla_{\hat{\mathbf{x}}}\dot{\boldsymbol{\omega}} = \left[\mathbf{0} \quad \boldsymbol{\Gamma}(\boldsymbol{\omega}) \quad \tilde{\mathbf{I}}^{-1} \right] \quad (200)$$

The transformation defined by $\boldsymbol{\Gamma}$ is

$$\boldsymbol{\Gamma}(\boldsymbol{\omega}) = -\tilde{\mathbf{I}}^{-1} \begin{bmatrix} -I_{xz}\omega_y + I_{xy}\omega_z & -2I_{yz}\omega_y + (I_z - I_y)\omega_z - I_{xz}\omega_x & 2I_{yz}\omega_z + (I_z - I_y)\omega_y + I_{xy}\omega_x \\ 2I_{xz}\omega_x + (I_x - I_z)\omega_z + I_{yz}\omega_y & I_{yz}\omega_x - I_{xy}\omega_z & -2I_{xz}\omega_z + (I_x - I_z)\omega_x - I_{xy}\omega_y \\ -2I_{xy}\omega_x + (I_y - I_x)\omega_y - I_{yz}\omega_z & 2I_{xy}\omega_y + (I_y - I_x)\omega_x + I_{xz}\omega_z & -I_{yz}\omega_x + I_{xz}\omega_y \end{bmatrix} \quad (201)$$

Evaluating Equation (200) on the estimated trajectory yields the final result:

$$\nabla_{\hat{\mathbf{x}}}\dot{\boldsymbol{\omega}}|_{\mathbf{x}=\hat{\mathbf{x}}} = \left[\mathbf{0} \quad \boldsymbol{\Gamma}(\hat{\boldsymbol{\omega}}) \quad \tilde{\mathbf{I}}^{-1} \right] \quad (202)$$

The linear equation for the disturbance torque is trivially equal to $\mathbf{0}$. Combining these results yields the linear dynamics:

$$\mathbf{F}(t, \mathbf{u}_d) = \begin{bmatrix} -\mathbf{X}(\hat{\boldsymbol{\omega}}) & \frac{1}{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \Gamma(\hat{\boldsymbol{\omega}}) & \tilde{\mathbf{I}}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (203)$$

In dealing with the output equation, the linearization takes on a different form due to the nonstandard innovations process defined in (85). Thus, it is not possible to obtain \mathbf{H} by merely taking the gradient of the observation equation and evaluating along the nominal trajectory. Instead, \mathbf{H} is obtained by making the following observation. The measured magnetic field may be obtained by rotating the estimated magnetic field through the rotation matrix defined by the quaternion perturbation $\Delta\mathbf{q}$ as follows:

$$\mathbf{b}_m = \mathbf{A}^{B/I}(\Delta\mathbf{q})\hat{\mathbf{b}} = \begin{bmatrix} 1 & 2\Delta q_3 & -2\Delta q_2 \\ -2\Delta q_3 & 1 & 2\Delta q_1 \\ 2\Delta q_2 & -2\Delta q_1 & 1 \end{bmatrix} \hat{\mathbf{b}} = \hat{\mathbf{b}} + 2\hat{\mathbf{b}} \times \Delta\mathbf{q} \quad (204)$$

Inserting Equation (204) into Equation (85) yields the following equation:

$$\mathbf{v} = \frac{(\hat{\mathbf{b}} + 2\hat{\mathbf{b}} \times \Delta\mathbf{q}) \times \hat{\mathbf{b}}}{\|\mathbf{b}_m\| \|\hat{\mathbf{b}}\|} = 2 \left(\mathbf{I} - \frac{[\hat{\mathbf{b}}\hat{\mathbf{b}}^T]}{\|\hat{\mathbf{b}}\|^2} \right) \Delta\mathbf{q} \quad (205)$$

However, since the estimated vector is approximately equal to the calculated vector, the engineering approximation can be made to obtain the appropriate output matrix:

$$\mathbf{v} \approx \begin{bmatrix} 2\mathbf{I} - \frac{2\hat{\mathbf{b}}\hat{\mathbf{b}}_m^T}{\|\mathbf{b}_m\| \|\hat{\mathbf{b}}\|} & \mathbf{0} & \mathbf{0} \end{bmatrix} \Delta\tilde{\mathbf{x}} = \mathbf{H}\Delta\tilde{\mathbf{x}} \quad (206)$$

I.2 Linear System for the Linear Quadratic Regulator

In this section, the superscripts for the quaternion and angular velocities of the body frame with respect to the reference frame shall be omitted for a notational convenience. The linear quadratic regulator uses a different state than the extended Kalman filter. It is as follows:

$$\mathbf{x}_{lqr} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix} \quad (207)$$

The control variable \mathbf{u} is defined in (99) to simply be the magnetic moment \mathbf{m} produced by the magnetorquers. The goal of this section is to find the matrices \mathbf{A} and $\mathbf{B}(t)$ which define the linear system as follows:

$$\dot{\mathbf{x}}_{lqr} = \mathbf{A}\mathbf{x}_{lqr} + \mathbf{B}(t)\mathbf{u} \quad (208)$$

The first step in obtaining the linear approximation is to choose the nominal trajectory to linearize upon. The nominal trajectory is the typical nadir pointing attitude where the reference system is identical to the body coordinate system. In other words:

$$\mathbf{x}_{nom} = \mathbf{0} \quad (209)$$

Note that the dynamics of Equation (208) do not include the evolution of the fourth element of the quaternion q_4 . Nevertheless, these dynamics are a function of this element. Thus, the following derivation will make use of the full four-element quaternion. When evaluating along the nominal trajectory, it will be necessary to know the nominal value of the fourth element of the quaternion. From the constraint equation given in Equation (27), it is easy to show that this nominal value is simply one.

The nonlinear kinematics equation is merely the first three rows of Equation (41). It is repeated here as

$$\dot{\mathbf{q}} = [\mathbf{I} \quad \mathbf{0}] \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \mathbf{q} \quad (210)$$

To obtain the dynamics equation, the rule stated in Equation (32) is invoked:

$$\boldsymbol{\omega} = \boldsymbol{\omega}^{\text{B/I}} - \boldsymbol{\omega}^{\text{R/I}} \quad (211)$$

Taking the derivative of this equation yields

$$\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}^{\text{B/I}} - \dot{\boldsymbol{\omega}}^{\text{R/I}} \quad (212)$$

The angular velocity of the reference frame with respect to the inertial frame is defined in Equation (139):

$$\boldsymbol{\omega}^{\text{R/I}} = -\frac{\|\mathbf{r} \times \mathbf{v}\|}{\|\mathbf{r}\|^2} \mathbf{y}_R = -\omega_0 \mathbf{y}_R = -\omega_0 \mathbf{A}^{\text{B/R}} \begin{pmatrix} \mathbf{q}^{\text{B/R}} \\ 1 \end{pmatrix} = \omega_0 \begin{bmatrix} 0 \\ q_1^2 - q_2^2 + q_3^2 - q_4^2 \\ 2(q_1 q_4 - q_2 q_3) \end{bmatrix} \quad (213)$$

Assuming ω_0 is constant (which is true for circular orbits and almost true for near circular orbits), taking the derivative of Equation (213) yields

$$\dot{\boldsymbol{\omega}}^{\text{R/I}} = 2\omega_0 \begin{bmatrix} -q_2 & -q_1 & -q_4 & -q_3 \\ q_1 & -q_2 & q_3 & -q_4 \\ q_4 & -q_3 & -q_2 & q_1 \end{bmatrix} \dot{\mathbf{q}} \quad (214)$$

After substituting Equation (41) the following result is obtained:

$$\dot{\boldsymbol{\omega}}^{\text{R/I}} = 2\omega_0 \begin{bmatrix} 0 & 2(-q_1 q_4 + q_2 q_3) & q_1^2 - q_2^2 + q_3^2 - q_4^2 \\ 2(q_1 q_4 - q_2 q_3) & 0 & 2(q_1 q_2 + q_3 q_4) \\ -q_1^2 + q_2^2 - q_3^2 + q_4^2 & -2(q_1 q_2 + q_3 q_4) & 0 \end{bmatrix} \boldsymbol{\omega}^{\text{B/R}} \quad (215)$$

The angular velocity of the body frame was expanded in Equation (199):

$$\dot{\boldsymbol{\omega}}^{\mathbf{B}/\mathbf{I}} = \tilde{\mathbf{I}}^{-1} \left(\mathbf{t}_m + \mathbf{t}_{gg} + \mathbf{t}_d - \begin{bmatrix} I_{yz} \left(-\omega_y^{\mathbf{B}/\mathbf{I}^2} + \omega_z^{\mathbf{B}/\mathbf{I}^2} \right) + (I_z - I_y) \omega_y^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} - I_{xz} \omega_x^{\mathbf{B}/\mathbf{I}} \omega_y^{\mathbf{B}/\mathbf{I}} + I_{xy} \omega_x^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} \\ I_{xz} \left(\omega_z^{\mathbf{B}/\mathbf{I}^2} - \omega_y^{\mathbf{B}/\mathbf{I}^2} \right) + (I_x - I_z) \omega_x^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} + I_{yz} \omega_x^{\mathbf{B}/\mathbf{I}} \omega_y^{\mathbf{B}/\mathbf{I}} - I_{xy} \omega_y^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} \\ I_{xy} \left(-\omega_x^{\mathbf{B}/\mathbf{I}^2} + \omega_y^{\mathbf{B}/\mathbf{I}^2} \right) + (I_y - I_x) \omega_x^{\mathbf{B}/\mathbf{I}} \omega_y^{\mathbf{B}/\mathbf{I}} - I_{yz} \omega_x^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} + I_{xz} \omega_y^{\mathbf{B}/\mathbf{I}} \omega_z^{\mathbf{B}/\mathbf{I}} \end{bmatrix} \right) \quad (216)$$

Note that in this development, the gravity gradient torque is pulled out of the disturbance torque. This is because, since the gravity gradient effect is the dominant torque, it will be included in the linearized dynamics. The equation for the gravity gradient disturbance torque is given in Equation (168). It is repeated here:

$$\mathbf{t}_{gg} = \frac{3\mu}{r_e^3} \hat{\mathbf{r}}_e \times (\tilde{\mathbf{I}} \hat{\mathbf{r}}_e) \quad (217)$$

It can be shown that

$$\frac{\mu}{r_e^3} \approx \omega_0^2 \quad (218)$$

Substituting this result into Equation (217) yields the following result:

$$\mathbf{t}_{gg} = 3\omega_0^2 \begin{pmatrix} A(q) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \end{pmatrix} \times \begin{pmatrix} \tilde{\mathbf{I}} A(q) \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \end{pmatrix} = 3\omega_0^2 \begin{bmatrix} 2(q_1 q_3 - q_2 q_4) \\ 2(q_2 q_3 + q_1 q_4) \\ -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \times \begin{pmatrix} \tilde{\mathbf{I}} \begin{bmatrix} 2(q_1 q_3 - q_2 q_4) \\ 2(q_2 q_3 + q_1 q_4) \\ -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \end{pmatrix} \quad (219)$$

The complete nonlinear dynamics are obtained by substituting Equations (215), (216), and (219) into Equation (212) and combining this with Equation (210). To obtain the linear dynamics, it is necessary to take the gradient of the nonlinear dynamics and evaluate along the nominal trajectory. This will be done by dealing with each of the equations one at a time, beginning with Equation (210). Taking the gradient yields

$$\nabla_{\mathbf{x}_{lqe}} \dot{\mathbf{q}} = \frac{1}{2} [-\mathbf{X}(\boldsymbol{\omega}) \quad \mathbf{X}(\mathbf{q}) + q_4 \mathbf{I}] \quad (220)$$

The matrix \mathbf{X} is the normal skew symmetric matrix associated with the cross product as defined in Equation (102). Evaluating along the nominal trajectory yields the linear equation

$$\dot{\mathbf{q}}_{linear} = \left[\mathbf{0} \quad \frac{1}{2} \mathbf{I} \right] \mathbf{x}_{lqr} \quad (221)$$

Next the linear form of Equation (215) will be found. Taking the gradient yields

$$\nabla_{\mathbf{x}_{lqr}} \dot{\boldsymbol{\omega}}^{R/I} = \omega_0 \begin{bmatrix} 2(-\omega_y q_4 + \omega_z q_1) & 2(\omega_y q_3 - \omega_z q_2) & 2(\omega_y q_2 + \omega_z q_3) & 0 & 2(-q_1 q_4 + q_2 q_3) & q_1^2 - q_2^2 + q_3^2 - q_4^2 \\ 2(\omega_x q_4 + \omega_z q_2) & 2(-\omega_x q_3 + \omega_z q_1) & 2(-\omega_x q_2 + \omega_z q_4) & 2(q_1 q_4 - q_2 q_3) & 0 & 2(q_1 q_2 + q_3 q_4) \\ 2(-\omega_x q_1 - \omega_y q_2) & 2(\omega_x q_2 - \omega_y q_1) & 2(-\omega_x q_3 - \omega_y q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & -2(q_1 q_2 + q_3 q_4) & 0 \end{bmatrix} \quad (222)$$

Evaluating this equation along the nominal trajectory yields the linear equation

$$\dot{\boldsymbol{\omega}}_{linear}^{R/I} = \omega_0 \begin{bmatrix} 0 & 0 & -1 \\ \mathbf{0} & 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{x}_{lqr} \quad (223)$$

To find the linear form of Equation (216), first the explicit equation for the angular velocity of the body frame with respect to the inertial frame is found by substituting Equation (213) into Equation (211):

$$\boldsymbol{\omega}^{B/I} = \boldsymbol{\omega} + \omega_0 \begin{bmatrix} -2(q_1 q_2 + q_3 q_4) \\ q_1^2 - q_2^2 + q_3^2 - q_4^2 \\ 2(q_1 q_4 - q_2 q_3) \end{bmatrix} \quad (224)$$

Taking the gradient of this equation yields

$$\nabla_{\mathbf{x}_{lqr}} \boldsymbol{\omega}^{B/I} = \begin{bmatrix} -2\omega_0 q_2 & -2\omega_0 q_1 & -2\omega_0 q_4 & 1 & 0 & 0 \\ 2\omega_0 q_1 & -2\omega_0 q_2 & 2\omega_0 q_3 & 0 & 1 & 0 \\ 2\omega_0 q_4 & -2\omega_0 q_3 & -2\omega_0 q_2 & 0 & 0 & 1 \end{bmatrix} \quad (225)$$

Taking the gradient of Equation (219) requires a great deal of algebraic manipulation. It is finally obtained as the following matrix equation:

$$\nabla_{\tilde{\mathbf{x}}_{lqr}} \mathbf{t}_{gg} = 6\omega_0^2 \left[\mathbf{X} \begin{pmatrix} 2(q_1q_3 - q_2q_4) \\ 2(q_2q_3 + q_1q_4) \\ -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix} \tilde{\mathbf{I}} \begin{bmatrix} q_3 & -q_4 & q_1 \\ q_4 & q_3 & q_2 \\ -q_1 & -q_2 & q_3 \end{bmatrix} - \mathbf{X} \left(\tilde{\mathbf{I}} \begin{bmatrix} 2(q_1q_3 - q_2q_4) \\ 2(q_2q_3 + q_1q_4) \\ -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \right) \begin{bmatrix} q_3 & -q_4 & q_1 \\ q_4 & q_3 & q_2 \\ -q_1 & -q_2 & q_3 \end{bmatrix} \right] \mathbf{0} \quad (226)$$

The matrix \mathbf{X} represents the skew symmetric matrix associated with the cross product, as defined in Equation (102). Taking the gradient of Equation (216) yields

$$\nabla_{\mathbf{x}_{lqr}} \dot{\boldsymbol{\omega}}^{B/I} = \tilde{\mathbf{I}}^{-1} \left(\nabla_{\mathbf{x}_{lqr}} \mathbf{t}_{gg} + \begin{bmatrix} -I_{xz}\omega_y^{B/I} + I_{xy}\omega_z^{B/I} & -2I_{yz}\omega_y^{B/I} + (I_z - I_y)\omega_z^{B/I} - I_{xz}\omega_x^{B/I} & 2I_{yz}\omega_z^{B/I} + (I_z - I_y)\omega_y^{B/I} + I_{xy}\omega_x^{B/I} \\ 2I_{xz}\omega_x^{B/I} + (I_x - I_z)\omega_z^{B/I} + I_{yz}\omega_y^{B/I} & I_{yz}\omega_x^{B/I} - I_{xy}\omega_z^{B/I} & -2I_{xz}\omega_z^{B/I} + (I_x - I_z)\omega_x^{B/I} - I_{xy}\omega_y^{B/I} \\ -2I_{xy}\omega_x^{B/I} + (I_y - I_x)\omega_y^{B/I} - I_{yz}\omega_z^{B/I} & 2I_{xy}\omega_y^{B/I} + (I_y - I_x)\omega_x^{B/I} + I_{xz}\omega_z^{B/I} & -I_{yz}\omega_x^{B/I} + I_{xz}\omega_y^{B/I} \end{bmatrix} \nabla_{\mathbf{x}_{lqr}} \boldsymbol{\omega}^{B/I} \right) \quad (227)$$

Substituting Equations (225) and (226) and evaluating along the nominal trajectory yields the final result. Note that the nominal value for $\boldsymbol{\omega}^{B/I}$ is obtained directly from (224) as follows:

$$\boldsymbol{\omega}_{nom}^{B/I} = \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix} \quad (228)$$

Using this value in the above-mentioned substitutions yields the final result for the angular velocity of the body frame with respect to the inertial frame:

$$\dot{\boldsymbol{\omega}}_{linear}^{B/I} = \omega_0 \tilde{\mathbf{I}}^{-1} \begin{bmatrix} 8\omega_0(I_z - I_y) & -6\omega_0 I_{xy} & 2I_{xz} & -I_{xz} & -2I_{yz} & I_z - I_y \\ -8\omega_0 I_{xy} & 6\omega_0(I_z - I_x) & -2\omega_0 I_{yz} & I_{yz} & 0 & -I_{xy} \\ 8\omega_0 I_{xz} & 6\omega_0 I_{yz} & 2\omega_0(I_x - I_y) & I_y - I_x & 2I_{xy} & I_{xz} \end{bmatrix} \mathbf{x}_{lqr} \quad (229)$$

Substituting Equations (223) and (229) into Equation (212) yields the final linear equation for the angular velocity of the satellite with respect to the reference frame:

$$\dot{\boldsymbol{\omega}}_{linear} = \omega_0 \tilde{\mathbf{I}}^{-1} \begin{bmatrix} 8\omega_0(I_z - I_y) & -6\omega_0 I_{xy} & 2I_{xz} & 0 & -2I_{yz} & I_x - I_y + I_z \\ -8\omega_0 I_{xy} & 6\omega_0(I_z - I_x) & -2\omega_0 I_{yz} & 2I_{yz} & 0 & -2I_{xy} \\ 8\omega_0 I_{xz} & 6\omega_0 I_{yz} & 2\omega_0(I_x - I_y) & -I_x + I_y - I_z & 2I_{xy} & 0 \end{bmatrix} \mathbf{x}_{lqr} \quad (230)$$

Stacking the matricides from Equations (221) and (232) yields the \mathbf{A} matrix for the linear system dynamics:

$$\mathbf{A} = \left[\begin{array}{c} \mathbf{0} \quad \frac{1}{2}\mathbf{I} \\ \omega_0 \tilde{\mathbf{I}}^{-1} \left[\begin{array}{ccc} 8\omega_0(I_z - I_y) & -6\omega_0 I_{xy} & 2I_{xz} \\ -8\omega_0 I_{xy} & 6\omega_0(I_z - I_x) & -2\omega_0 I_{yz} \\ 8\omega_0 I_{xz} & 6\omega_0 I_{yz} & 2\omega_0(I_x - I_y) \end{array} \right] \end{array} \quad \begin{array}{ccc} 0 & -2I_{yz} & I_x - I_y + I_z \\ 2I_{yz} & 0 & -2I_{xy} \\ -I_x + I_y - I_z & 2I_{xy} & 0 \end{array} \right] \quad (231)$$

The component from the control comes directly from Equation (216). It is stated in Equation (3). Note that this is already a linear equation. When this linear equation is evaluated along the nominal trajectory, the time varying \mathbf{B} matrix is obtained as:

$$\mathbf{B}(t) = \left[\begin{array}{c} \mathbf{0} \\ -\tilde{\mathbf{I}}^{-1} \mathbf{X}(\mathbf{b}(t)) \end{array} \right] \quad (232)$$

The vector \mathbf{b} is simply the Earth's magnetic field in the fixed body coordinate system.

REFERENCES

- [1] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Boston: Dordrecht Reidel, 1978.
- [2] D. K. Cheng, *Field and Wave Electromagnetics*. Reading: Addison-Wesley, 1989.
- [3] M. H. Kaplan. *Modern Spacecraft Dynamics & Control*. New York: John Wiley & Sons, 1976.
- [4] B. Wie, *Space Vehicle Dynamics and Control*. Reston: AIAA, 1998.
- [5] P. C. Hughes, *Spacecraft Attitude Dynamics*. New York: John Wiley & Sons, 1986.
- [6] J. E. Prussing and B. A. Conway, *Orbital Mechanics*. New York: Oxford, 1993.
- [7] A. S. Jursa, *Handbook of Geophysics and the Space Environment*, Air Force Laboratory, Air Force Systems Command USAF, NTIS Document ADA 167000, 1985.
- [8] W. H. Campbell, *Introduction to Geomagnetic Fields*. New York: Cambridge, 1997.
- [9] C. Roithmayr, "Contributions of Spherical Harmonics to Magnetic and Gravitational Fields," NASA Johnson Space Center, Tech. Rep. EG2-96-02, Jan. 23, 1996.
- [10] G. Wahba, "Problem 65-1: A least squares estimate of satellite attitude." *SIAM Review*, vol. 7, p. 409, July 1965.
- [11] M. L. Psiaki, F. Martel, and P. K. Pal, "Three-axis attitude determination via kalman filtering of magnetometer data," *Journal of Guidance, Control, and Dynamics*, vol. 13, pp. 506-514, May-June 1990.
- [12] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: John Wiley, 1997.
- [13] E. J. Lefferts et al., "Kalman filtering for spacecraft attitude estimation", *Journal of Guidance and Control*, Vol. 5, pp. 417-429, Sep. 1982.
- [14] M. L. Psiaki, "Magnetic torquer attitude control via asymptotic periodic linear quadratic regulation," *Journal of Guidance, Control, and Dynamics*, vol. 24, pp. 386-394, March-April 2001.
- [15] W. L. Brogan, *Modern Control Theory*. Englewood Cliffs: Prentice Hall, 1991.

- [16] T. E. Humphreys, "Attitude determination for small satellites with modest pointing constraints," in *16th Annual AIAA/USU Conference on Small Satellites*, paper SSC02-vi-1, 2002.
- [17] J. Meeus, *Astronomical Algorithms*, Richmand: Willmann-Bell, 1991
- [18] M. J. Sidi, *Spacecraft Dynamics and Control*. New York: Cambridge, 1997.