2013

AN ATTITUDE DETERMINATION SYSTEM WITH MEMS GYROSCOPE DRIFT COMPENSATION FOR SMALL SATELLITES

Maxwell Bezold
University of Kentucky, max.bezold@gmail.com

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Maxwell Bezold, Student

Dr. James E. Lumpp, Major Professor

Cai-Cheng Lu, Director of Graduate Studies

AN ATTITUDE DETERMINATION SYSTEM WITH MEMS GYROSCOPE DRIFT COMPENSATION FOR SMALL SATELLITES

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering in the College of Engineering at the University of Kentucky

By
Maxwell Bezold
Lexington, Kentucky
Director: James E. Lumpp
Associate Professor of Electrical Engineering
Lexington, Kentucky
2013

ABSTRACT OF THESIS

AN ATTITUDE DETERMINATION SYSTEM WITH MEMS GYROSCOPE DRIFT COMPENSATION FOR SMALL SATELLITES

This thesis presents the design of an attitude determination system for small satellites that automatically corrects for attitude drift. Existing attitude determination systems suffer from attitude drift due to the integration of noisy rate gyro sensors used to measure the change in attitude. This attitude drift leads to a gradual loss in attitude knowledge, as error between the estimated attitude and the actual attitude increases.

In this thesis a Kalman Filter is used to complete sensor fusion which combines sensor observations with a projected attitude based on the dynamics of the satellite. The system proposed in this thesis also utilizes a novel sensor called the stellar gyro to correct for the drift. The stellar gyro compares star field images taken at different times to determine orientation, and works in the presence of the sun and during eclipse. This device provides a relative attitude fix that can be used to update the attitude estimate provided by the Kalman filter, effectively compensating for drift. Simulink models are developed of the hardware and algorithms to model the effectiveness of the system. The Simulink models show that the attitude determination system is highly accurate, with steady state errors of less than 1 degree.

KEYWORDS: Extended Kalman Filter, CubeSat, Attitude Determination, Stellar Gyro, Attitude Drift

Maxwell Bezold

July 17th 2013

AN ATTITUDE DETERMINATION SYSTEM WITH MEMS GYROSCOPE DRIFT COMPENSATION FOR SMALL SATELLITES

By

Maxwell Bezold

James E Lumpp PhD

Director of Thesis

Cai-Cheng Lu PhD

Director of Graduate Studies

July 17th 2013

To my Parents: Don and Jamie Bezold

ACKNOWLEDGEMENTS

I first must acknowledge Dr. James Lumpp, who took a chance on an inexperienced Chemical Engineering senior with an interest in space.

I must also thank my colleague, Samir Rawashdeh, for providing me with an orbital environment simulator, an important stepping stone, as well as a unique and novel sensor, the stellar gyro, which provides a very useful extension of my thesis.

I would also like to thank Dr. James Mohl, of Ball Aerospace and Technologies Corporation, for his advice and insight. His mentoring during my internship at the Ball Aerospace and Technologies Corporation's Boulder campus during the summer of 2012 gave me many of the crucial skills I needed to execute this thesis.

I would also like to thank all of my colleagues at the Space Systems Laboratory at the University of Kentucky. You all have been the perfect team and have made this academic adventure truly enjoyable.

I must acknowledge my parents for their love and support. Without them, I would have never made the decision to pursue this field of study which will allow me to have a career in the space industry, a lifelong dream.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS III
LIST OF FIGURES VI
ACRONYMS AND ABBREVIATIONS VIII
1 INTRODUCTION 1
1.1 ATTITUDE DETERMINATION AND CONTROL SYSTEMS 1
1.2 ABSOLUTE ATTITUDE DETERMINATION AND ATTITUDE PROPAGATION 3
1.3 THE ATTITUDE DETERMINATION PROCESS 4
1.4 ATTITUDE DRIFT 7
1.5 PROBLEM STATEMENT 9
2 BACKGROUND AND PREVIOUS WORK 10
2.1 CUBESAT FORM FACTOR 10
2.1.1 CubeSat Mission Examples 12
2.1.2 CubeSat Subsystems 14
2.2 COORDINATE REFERENCE FRAMES 15
2.2.1 Earth Centered, Earth Fixed (ECEF) 15
2.2.2 Earth Center Inertial (ECI) 16
2.2.3 Spacecraft or Body Fixed Coordinates 16
2.2.4 Roll, Pitch, and Yaw (RPY) Coordinates 16
2.3 COORDINATE ROTATION AND PARAMETERIZATIONS 18
2.3.1 Direction Cosine Matrix (DCM) 18
2.3.2 Euler Angle Parameterization of the DCM 19
2.3.3 Quaternion Parameterization of the DCM 20
2.4 AN OVERVIEW OF SPACECRAFT DYNAMICS 22
2.4.1 An Overview of Rigid Body Dynamics 22
2.4.2 Kinematics and Dynamics 23
2.4.3 Sources of Disturbance Torques 24
2.5 ATTITUDE DETERMINATION AND CONTROL SYSTEM COMPONENTS 26
2.5.1 Actuators 26
2.5.2 Sensors 29

2.6 Overview of Previous CubeSats with ADAC Systems 34
3 ATTITUDE DETERMINATION ALGORITHMS 36
3.1 Absolute Attitude Determination 36
3.1.1 Wahba's Problem 36
3.1.2 Solutions to Wahba's Problem: Davenport's q-Method 37
3.1.3 Solutions to Wahba's Problem: QUEST Algorithm 39
3.2 Attitude Propagation 41
3.2.1 State Space Models 42
3.2.2 Kalman Filter 43
3.2.3 Kalman Filter for Attitude Propagation 46
4 ATTITUDE DRIFT AND COMPENSATION 53
4.1 Attitude Drift Sources 53
4.1.1 Analog to Digital Conversion Process 53
4.1.2 Allan Variance 54
4.1.3 Gyro Resolution Limits 57
4.2 Drift Compensation through Stellar Gyro Updates 59
5 SIMULINK MODELING OF THE ATTITUDE DETERMINATION SYSTEM 61
5.1 High Level System Model 61
5.2 Simulink Sensor Models 63
5.3 Quaternion Extended Kalman Filter Subsystem 69
5.4 Absolute Attitude Determination Block 70
6 RESULTS 71
6.1 Drift Compensation with Low Roll Rate 71
6.2 Drift Compensation with Higher Roll Rate 74
7 CONCLUSION 76
7.1 Summary of Work 76
7.2 Future Work 77
REFERENCES 79
VITA 84

List of Figures

FIGURE 1: SYSTEM LEVEL BLOCK DIAGRAM OF AN ATTITUDE DETERMINATION AND CONTROL SYSTEM (ADCS) 2
FIGURE 2: SYSTEM SCHEMATIC OF A SMALL SATELLITE ATTITUDE DETERMINATION SYSTEM 5
FIGURE 3: ATTITUDE ERROR DUE TO DRIFT, GIVEN AS ROLL, PITCH, YAW EULER ANGLE ERRORS 8
FIGURE 4: A PROMOTIONAL POSTER FOR THE NASA ELANA PROGRAM SHOWING THREE 1U CUBESATS DEPLOYING FROM A P-POD. 11
FIGURE 5: THE PHARMASAT 3U CUBESAT. SOURCE: NASA 12
FIGURE 6: THE BODY FRAME, ALONG WITH RPY ANGLES ARE SHOWN, AS WELL AS THE LOCAL FRAME CONVENTION. SOURCE: CREATIVE COMMONS. 18
FIGURE 7: AN EXAMPLE OF A REACTION WHEEL. NOTE THE LARGE FLYWHEEL. SOURCE: NASA 27
FIGURE 8: AN EXAMPLE OF MAGNETORQUERS. 28
FIGURE 9: GYRO FROM A JUPITER IRBM SHOWING GIMBALS. 30
FIGURE 10.4 AN EXAMPLE OF STAR SENSORS INSTALLED ON A LARGE SATELLITE. NOTE THE LARGE BAFFLES INSTALLED ON THE THREE STAR SENSORS. 33
FIGURE 11: A SET OF THREE STAR SENSORS WITH ATTACHED BAFFLES. SOURCE: NASA 33
FIGURE 12: THE GAUSSIAN PROBABILITY DISTRIBUTION, CENTERED AT ZERO. SOURCE: WIKIMEDIA COMMONS 44
FIGURE 13: THE ANALOG TO DIGITAL CONVERSION PROCESS SHOWING A ZERO-ORDER HOLD PROCESS, WITH SAMPLING AND QUANTIZATION. 54
FIGURE 14: ALLAN DEVIATION OF THE GYROSCOPE IN AN ANALOG DEVICES' ADIS16334 IMU 55
FIGURE 15: ANGULAR RATES OF A CUBESAT SUBJECT TO ENVIRONMENTAL TORQUES AND THE +/1 LSB RESOLUTION LIMITS OF THE ADIS16334 GYRO 58
FIGURE 16: THE STELLAR GYRO PROVIDING THE KALMAN FILTER WITH PERIODIC DRIFT

FREE UPDATES. 60
FIGURE 17: HIGH LEVEL SIMULINK MODEL 62
FIGURE 18: SIMULINK SENSOR MODELS SHOWING THE GUI MENU FOR CONFIGURING SENSOR PARAMETERS. 64
FIGURE 19: THE MASKED VARIABLE CONTEXT MENU FOR THE MAGNETOMETER 65
FIGURE 20: MAGNETOMETER MODEL 65
FIGURE 21: SUN SENSOR INTERNAL MODEL 67
FIGURE 22: COARSE SUN SENSOR INTERNAL MODEL 68
FIGURE 23: QUATERNION EKF SUBSYSTEM 69
FIGURE 24: ABSOLUTE ATTITUDE DETERMINATION SUBSYSTEM 70
FIGURE 25: TRUTH ATTITUDE WITH NO INITIAL SPIN 72
FIGURE 26: ATTITUDE ESTIMATE WITH NO INITIAL SPIN 73
FIGURE 27: ATTITUDE ESTIMATE ERROR 73
FIGURE 28: TRUTH ATTITUDE WITH SOME INITIAL ANGULAR BODY RATES 74
FIGURE 29: ATTITUDE ESTIMATE WITH SOME INITIAL SPIN 75
FIGURE 30: ATTITUDE ERROR WITH SOME INITIAL SPIN 75

Acronyms and Abbreviations

AD	Attitude Determination
ADAC	Attitude Determination and Control
ADACS	Attitude Determination And Control System
ADCS	Attitude Determination and Control System
ARW	Angular Random Walk
CMG	Control Moment Gyro
C&DH	Command and Data Handling
CMOS	Complementary metal–oxide–semiconductor
CCD	Charge Coupled Device
DC	Direct Current
DCM	Direction Cosine Matrix
DoF	Degree of Freedom
ECEF	Earth Centered, Earth Fixed
ECI	Earth Centered, Inertial
ELANA	Educational Launch of Nanosatellite
E/PO	Education and Public Outreach
EPS	Electrical Power System
FOV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
ITRF	International Terrestrial Frame
IGRF	International Geomagnetic Reference Field
LSB	Least Significant Bit
MEMS	Micro Electronic Mechanical Systems
NASA	National Aeronautics and Space Administration
NORAD	North America Aerospace Defense Command
P-POD	Poly-PicoSatellite Orbital Deployer
RPY	Roll, Pitch, Yaw
SC	Spacecraft
TLE	Two Line Elements
UTIAS	University of Toronto Institute of Aerospace Studies

1 Introduction

A nontechnical description of an attitude determination and control system is given, as well as a description of the key limitations of existing systems. The problem statement of this thesis is presented.

1.1 Attitude Determination and Control Systems

An Attitude Determination and Control subsystem (ADCS) is the satellite subsystem responsible for controlling the orientation of a satellite in space. In the case of both aircraft and spacecraft, the term attitude refers to the orientation of the spacecraft in a given reference frame. Another term that is often used is pointing, as in a payload may have a pointing requirement such that the satellite must be oriented for a sensor to “point” at a target. An example of a payload with a pointing requirement is a camera that must point at a specific location on the surface of the earth [8].

A functional block diagram of an ADCS is shown in Figure 1. An ADCS system has two main functional components. The first is the Attitude Determination component, or AD component, indicated by the blue shading. The AD component’s role is to monitor several different sensors and determine the orientation of the satellite using a set of algorithms. Examples of sensors in the AD component are magnetometers that measure the earth’s magnetic field, sun sensors that measure the spacecraft to sun vector, and star trackers that determine the spacecraft’s attitude relative to the stars. Once the attitude is known, a control law is used to calculate the error between the actual and the desired attitude that is used to generate a control signal that is sent to the actuators, which then reorients the satellite back to the desired attitude. Examples of actuators include thrusters, magnetic torque rods, and reaction wheels. It is important to note that the ADCS has two closely related but distinct roles on the satellite. The first is slewing, which refers to changing the orientation of the satellite such that it points in a different direction. The other is attitude stabilization, which is the process of maintaining the satellite attitude in the

desired orientation. This is often referred to as tracking or attitude maintenance.

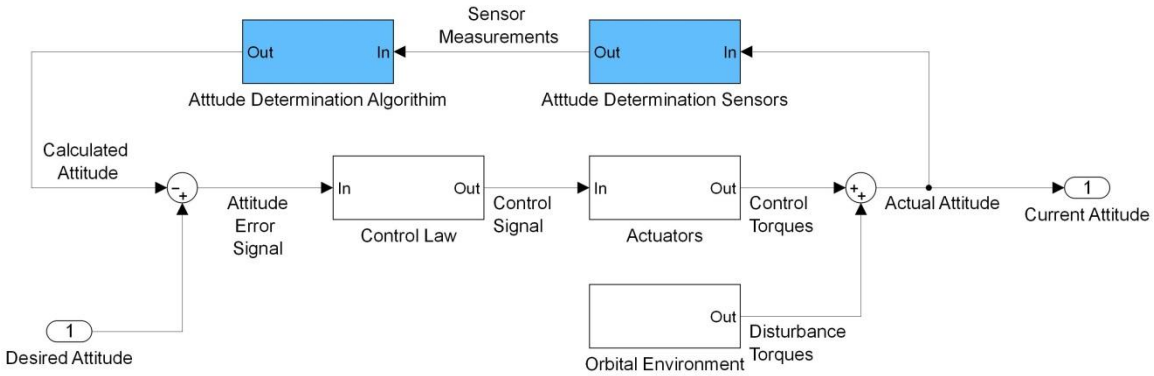

Figure 1: System Level Block Diagram of an Attitude Determination and Control System (ADCS)

ADCS systems are crucial to satellites for a number of different reasons. On some satellites, the payload may have a pointing requirement for operation. For instance, a payload like a space telescope will require that the satellite's attitude afford the telescope an unobstructed view of a certain region of space containing an astronomical feature of interest. The ADCS must maintain this attitude accurately, with high stability, for long periods of time to facilitate clear images. The satellite may also maintain its attitude such that their solar panels are always pointed to the sun. An earth observation satellite would have special requirements to reorient the satellite frequently to take observations of different areas of the earth. A communication satellite must remain pointed at a fixed location on the earth so that customers on the surface of the earth can orient their satellite dish antennas to transmit or receive radio signals from it.

An additional distinction is active versus passive attitude control systems. Active systems utilize sensors and actuators to monitor and control the attitude as described previously. They have the ability to change the attitude to arbitrary orientations. The second type, which are particularly common on smaller satellites, are passive control systems [8]. Passive control systems utilize permanent magnets to orient the satellite with the magnetic field lines of the earth. Damping of oscillations in the passive attitude control system are provided by hysteresis material that receive a temporary magnetic field dipole from the

earth's magnetic field, and subsequently provide oscillation damping with the oscillatory energy being dissipated as heat [9]. This thesis deals primarily with the attitude determination function of the ADCS.

1.2 Absolute Attitude Determination and Attitude Propagation

Attitude determination is comprised of two different, yet complementary, techniques. **Absolute attitude determination** refers to the process of determining the orientation of an object in space. This requires two different vector measurements provided by sensors on the spacecraft. These vectors include measured earth magnetic field vectors from sensors called magnetometers, sun vectors from sun sensors, or vectors indicating the location of the earth that are provided by earth horizon sensors. An algorithm called the QUEST algorithm takes two or more of these vector measurements and determines the absolute attitude.

On a spacecraft in orbit, it may not always be possible to measure two different vectors. This situation can occur for a number of different reasons. Optical type sensors such as star trackers and earth horizon sensors could be blinded by the sun, for instance. Another reason is caused by the absence of the sun due to obscuration by the earth. For most types of satellite orbits, at least some portion of the orbit will be spent on the dark side of the earth, without a line of sight to the sun. This situation is known as eclipse. In the eclipse scenario, the spacecraft loses one reference vector, the sun vector. Unless the spacecraft has large, complex, and expensive sensors such as star trackers or earth horizon sensors onboard, which are not affected by eclipse, attitude knowledge will be lost at some point during the orbit.

When two vector measurements are not available, the attitude must be propagated from the last known attitude determined using the QUEST algorithm. **Attitude propagation** is the second technique of the attitude determination process, and it requires knowledge of spacecraft angular body rates. Angular rate knowledge is provided by onboard gyroscopes. These angular rates are used to track the change from the initial absolute attitude. An algorithm known as

a Kalman filter uses the angular rate measurements from the gyroscopes to propagate the attitude.

Another motivation for implementing attitude propagation algorithms is to lower the duty cycle of a sensor. Many sensors consume large amounts of power, so it is often not possible to keep certain sensors such as star trackers and earth sensors on permanently, since most satellites have limited power generation capability. Adding an attitude propagation algorithm allows these sensors to be switched off, which is advantageous for small satellites where power can often be particularly limited.

This method of attitude propagation using a Kalman filter works fairly well, although it suffers from one key limitation. Integrating the angular rate measurements, which are noisy and suffer from other inaccuracy issues, causes a slow degradation in attitude knowledge over time. If the error is not compensated for, all attitude knowledge will eventually be lost. Building an attitude determination system that can compensate for attitude drift is a non-trivial problem.

1.3 The Attitude Determination Process

This section will present an initial attempt at designing an Attitude Determination System for small satellites. This system is particularly optimal for a class of satellites called CubeSats, which have dimensions in the 10 centimeter range. This type of satellite will be described in more detail in Chapter 2. Figure 2, below, shows an initial attempt at an attitude determination system for small satellites. This system design assumes an initial condition where the satellite has a complete lack of knowledge of its attitude in space, a situation that every satellite faces after deployment from a launch vehicle, or immediately after a full system reset. The first step of the attitude determination process is utilizing the solar panels as coarse sun sensors. The electrical power systems of satellites provide telemetry to the main spacecraft computer that identifies the voltage and current from each of the solar panels on the spacecraft. This information indicates the location of the sun relative to the different faces of the small

satellite, since only solar panels on faces illuminated by the sun will generate current. Thus, the solar panels can be used as a type of coarse sun sensor, providing inaccurate but useful information about the relative location of the sun. This method of using solar cells as coarse sun sensors has been demonstrated on Boeing CubeSat Testbed 1 [20].

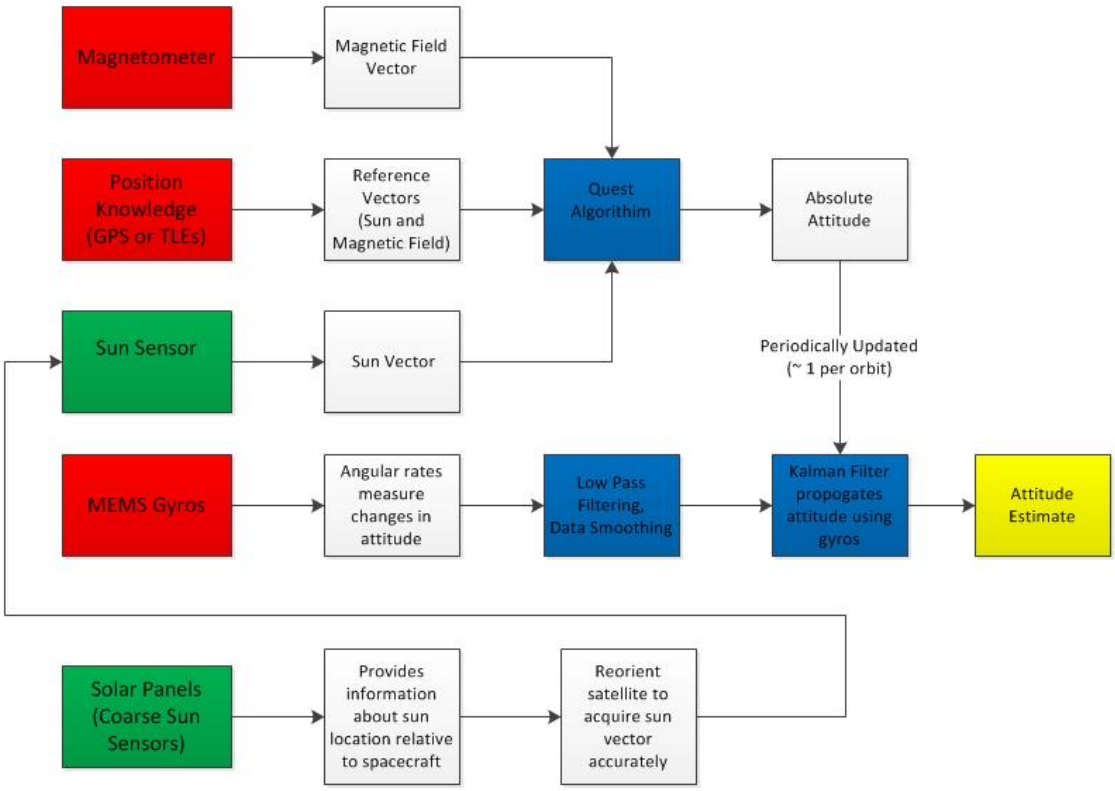

Figure 2: System Schematic of a small satellite attitude determination system

The coarse sensors are not accurate enough to provide the absolute attitude determination algorithm with one of the two vectors it needs to determine the attitude, but they do have adequate accuracy to assist the satellite in determining how it can orient itself so that the sun occupies the field of view of the highly accurate fine sun sensor. Thus, with the relative location of the sun provided by the solar panels, the satellite can command onboard actuators, such as magnetorquers and reaction wheels, to generate torque so that the satellite can rotate such that a more accurate fine sun sensor can acquire the sun vector. This maneuver does not require precision. Sun Sensors are available with fields of

view of 114 Degrees and with rapid update rates of 10 Hz, so only a slow rotation of the satellites is required to move the sun into the field of view of the sensor, a maneuver that is not extremely challenging [44].

After acquisition of a sun vector, an onboard magnetometer can provide the additional vector that QUEST needs to determine the attitude. The magnetometer is particularly useful for satellites, since the vector measurement it provides is always available, assuming a satellite is in orbit around a celestial body like the earth where a measureable magnetic field is present. In order to determine the attitude, QUEST must also be provided with reference vectors corresponding to the two measured vectors. These reference vectors must be calculated onboard the satellite using software models. These calculations require accurate knowledge of the satellite's current position. The CubeSat position can be determined in two different ways, through accurate knowledge of the spacecraft's orbit along with the current time, or through an onboard Global Positioning System (GPS) module.

The orbit of the satellite is described through Keplerian elements [8], which are a series of parameters that fully describe the orbit of the satellite. They are provided to satellite operators shortly after deployment by the North American Aerospace Defense Command (NORAD), a joint command of the United States Air Force and the Canadian Forces. NORAD operates a network of ground tracking radars that can accurately determine the orbits of satellites. NORAD supplies the Keplerian Elements in a format called Two Line Elements, or TLEs. These TLEs can then be uplinked to the satellite along with the current time, where a suitable algorithm can then determine the location of the satellite [21].

An alternative method for determining the location of the satellite is through an onboard GPS receiver. This method is more complicated from a hardware perspective, since the satellite must have a GPS receiver installed onboard. This oftentimes requires an additional set of external antennas so that the GPS receiver can detect the weak signals from the GPS constellation satellites. Additionally, the GPS unit can often have high power requirements, a problem for

satellites such as CubeSats with extremely limited onboard power generation [8]. In the experience of the author, the GPS option should be strongly considered, since accurate TLEs often take a long time to acquire from NORAD.

Once this location is known, the magnetic field reference vector and the sun reference vector can be calculated through the International Geomagnetic Reference Field (IGRF) magnetic field model and the VSOP 87 sun vector model, respectively [11] [44].

Providing a set of two vector measurements, along with corresponding reference vectors, allows QUEST to determine the absolute attitude. This absolute attitude is then provided to the Kalman filter. The Kalman filter propagates this absolute attitude to maintain attitude knowledge. In order to propagate the attitude, the satellite must include a set of gyroscopes to provide angular rotation rate measurements. These measurements must first be conditioned using a low pass filter to compensate for the random noise present in gyroscope outputs [45]. This process is known as smoothing [36]. The Kalman filter than integrates these angular rates to propagate the attitude.

This system has the advantage of requiring a minimal set of hardware. It can be implemented using a three-axis gyroscope, a sun sensor, and a magnetometer for a total of 3 pieces of hardware. This design is effective on paper. In practice however, it has a key limitation. Any system that relies on integration of gyro measurements will experience the attitude drift described in section 1.2. Any practical system must find a way to compensate for attitude drift, which will be described next.

1.4 Attitude Drift

Attitude drift is degradation in the propagated attitude solution, as time increases, since the last absolute attitude fix using the QUEST algorithm. This means that the error steadily accumulates in the attitude solution over time. This degradation is shown in Figure 3, below.

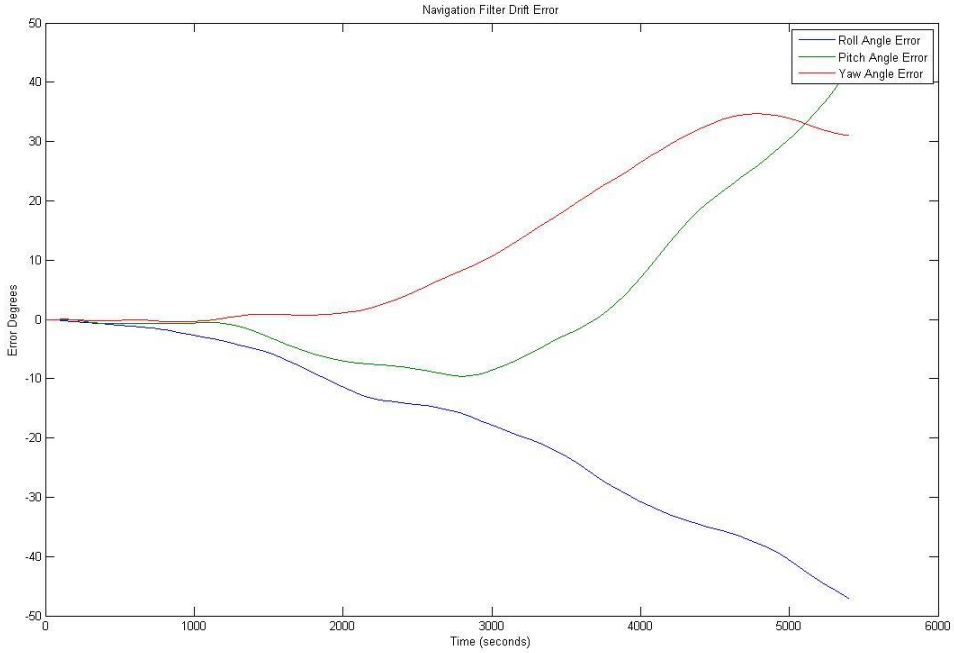

Figure 3: Attitude Error Due to Drift, given as Roll, Pitch, Yaw Euler Angle Errors

Figure 3 shows the drift error in a simulated attitude determination system over the course of an orbit. This system accurately models a gyro system and a Kalman Filter. At time 0, an absolute attitude is received, and the Kalman filter begins integrating the gyroscope outputs. The error is minimal at first, but begins to increase rapidly after 1000 seconds, or about 16 minutes. Eventually, the attitude error grows so great that the attitude knowledge is effectively lost. The conventional method for dealing with this problem on larger satellites is to update the attitude estimate periodically using the QUEST algorithm. This solution requires additional sensors and works well for larger satellites, which have less restrictive power requirements, as well as a large surface area and external structure to mount sensors. Larger satellites could install multiple sun sensors to cover a huge swath of the sky to ensure that a sun vector can always be measured, provided it is not in eclipse. Additionally, the large satellite would include multiple star trackers or earth sensors to provide an external reference when the sun is not visible, during eclipse for instance. This brute force approach

is not feasible on smaller satellites, such as CubeSats, due to power and form factor limitations. It is extremely difficult to mount an adequate number of sensors on a CubeSat to solve the drift problem in this manner. Every sensor installed on a CubeSat reduces the power generation capability of the CubeSat since the sensors occupy space that would normally be allocated for solar panels. At the same time, additional sensors require additional power. Clearly, this solution is not feasible on a CubeSat or other types of smaller satellites.

In order to solve this problem, it is necessary to take a closer look at the drift mechanism and identify a more targeted solution. Drift has a number of sources; all related to different types of errors in the gyro measurements. These errors, when integrated by the Kalman filter, produce the drift in the estimated attitude. The solution to this problem is presented in Chapter 4.

1.5 Problem Statement

This thesis describes the development of an attitude determination algorithm appropriate to small satellites, and the sensors commonly used on small satellites. This thesis will also demonstrate a solution to the problem of spacecraft attitude determination in eclipse, which is a significant problem for both large and small satellites. To this end, an attitude determination system is developed that utilizes a novel new sensor, a stellar gyroscope that uses images of star fields to determine the orientation of a satellite in space. Success of the final algorithm will be demonstrated through studies of the accuracy of the algorithm with models of actual sensors. The resulting system produces a continuous attitude error of less than 1 degree.

2 Background and Previous Work

This chapter presents background information about the hardware, mathematics, and techniques in spacecraft attitude determination and control field. Specific examples of hardware and system implementations will be presented.

2.1 CubeSat Form Factor

The CubeSat form factor was first proposed in 2001 by Jordi Puig-Suari of the California Polytechnic University and Bob Twiggs of Stanford University. The motivation was to dramatically lower the barriers to entry of space exploration such that non-nation entities with limited resources such as Universities and small companies could launch satellites. The CubeSat form factor is based around the basic volume and mass unit of the “U.” A “U” is a 10cm Cube with a mass of 1 kilogram and a center of gravity within 1 centimeter of the center of volume. A single CubeSat can be 1, 2, or 3 Us in length, but no longer. This constraint is a result of the design of the standard CubeSat deployer, which is called a Poly-PicoSatellite Orbital Deployer, or P-POD. The P-POD can only accept up to 3 1U CubeSats. The P-Pod provides a standard bolt pattern and standard deployment system for all CubeSats. The P-POD itself is not large, with dimension of approximately 3 feet long and 10 inches tall and wide. It was designed specifically to bolt to large space rockets, and allow up to 3 1U CubeSats to share a ride as secondary payloads on a larger mission. Most rockets launching large satellites have extra mass and volume left over after the main satellite is integrated with the launch vehicle. In the past, in order to make the dynamics of the rocket predictable, this volume and mass allocation was taken up by ballast. The P-POD and the CubeSat standard thus put this extra volume and mass to good use, by providing low cost access to space to organizations without the access to large budgets [1].

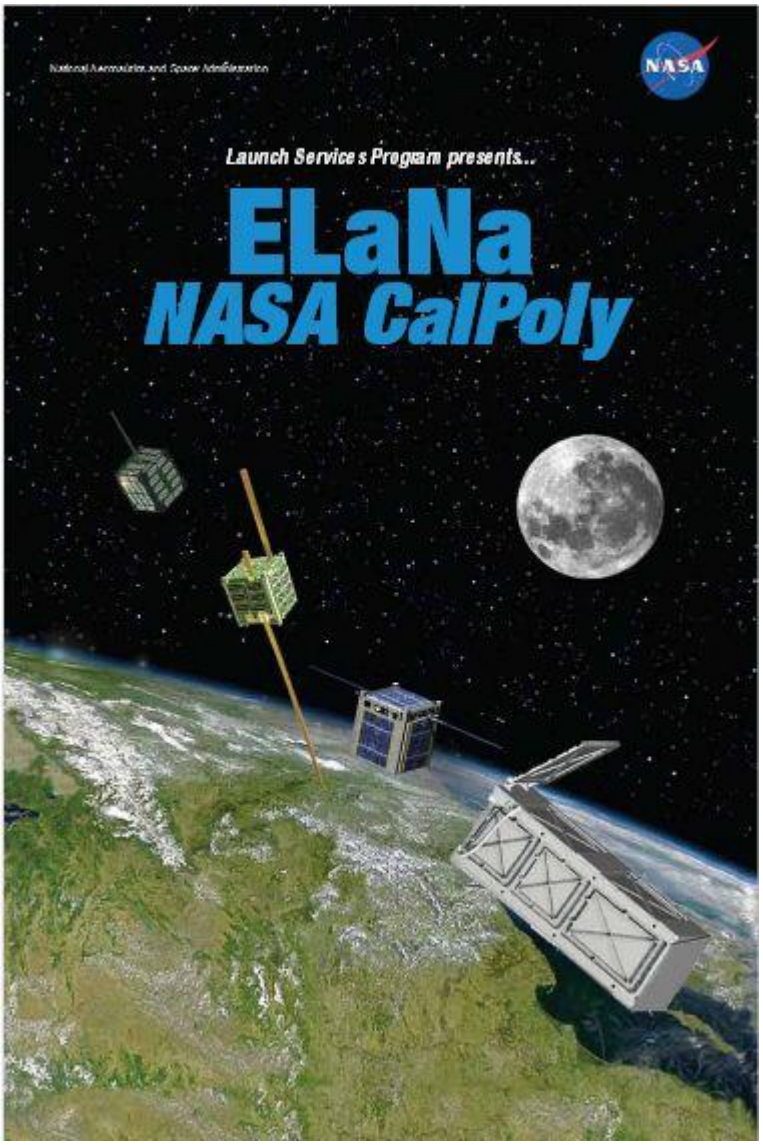

Figure 4: A promotional poster for the NASA ElaNa Program showing three 1U CubeSats deploying from a P-Pod. In reality the P-Pod would be mounted on a rocket. Source: NASA

2.1.1 CubeSat Mission Examples

To date, many CubeSats have been built, conducting a variety of different mission types. Mission types conducted include biological experiments, engineering technology demonstration, space weather research, astrophysics, and education and public outreach (E/PO).

NASA's Ames research center has flown a series of 3U biological research CubeSats. An example of one of these missions is Pharmasat, which launched in 2009. Pharmasat contained a temperature controlled payload with a life support system for a colony of yeast microbes. The health and viability of the

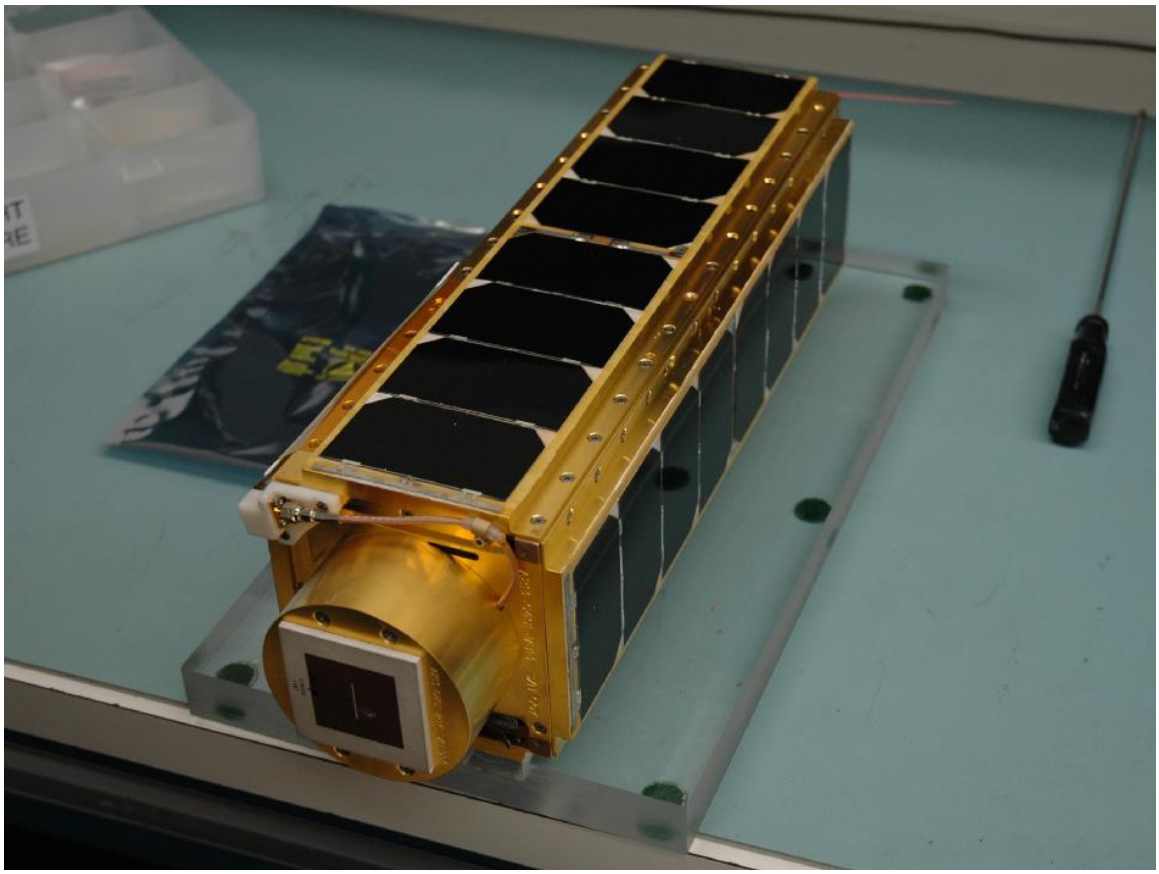

Figure 5: The Pharmasat 3U CubeSat Source: NASA

yeast in microgravity was monitored, as well as the effect of drugs on the yeast in the space environment [2].

RAX-2, a 3U CubeSat built by the University of Michigan and funded by the National Science Foundation, studies space weather phenomena, an

important area of study in aeronautics due to the serious disruptions that space weather, such as plasma and solar flares, can cause for satellite missions. In the RAX-2 experiments, ground-based radar transmits RF energy into plasma clouds in low earth orbit. On orbit, RAX-2 then measures the scattering caused by the plasma clouds, which provides important information about the distribution and the formation mechanism of these clouds, which can cause communication outages for spacecraft in earth orbit [3].

Pico Satellite Solar Cell Testbed-2 (PSSCT-2) is an example of an engineering technology demonstration mission. Developed by the Aerospace Corporation, PSSCT-2 was ejected from the Space Shuttle Atlantis during the final STS-135 mission on July 20, 2011. The purpose of this satellite was to test new solar cell technology. PSSCT-2 is notable for having an active control system with sensors and actuators that allowed the CubeSat to track the sun [4].

The Cosmic X-Ray Background NanoSat (CXBN) is an example of a CubeSat that had an astrophysics mission. CXBN, built at Morehead State University in Morehead, Kentucky with a sensor provided by University of California-Berkeley, was meant to study cosmic background radiation present in space spread by the big bang. It was a 3U CubeSat with deployable solar cells and an active control system that slowly rotates the CubeSat to provide gyroscope stabilization and allows the sensor to view the full sky [5].

KySat-1 was an example of an education and public outreach (E/PO) mission. Initiated in 2006 by the Kentucky Science and Technology Corporation, KySat-1 was meant to jump start aerospace development in the state of Kentucky and provide K-12 educators with a unique teaching tool through a network of mobile ground stations that would allow K-12 students to interact with the satellite. KySat-1 launched as part of the first NASA ELANA (Educational Launch of Nanosatellite) Mission in March 2011. Unfortunately the Taurus XL carry rocket failed to reach orbital velocity, and KySat-1 never made it to orbit. A follow-on mission with the same education objective, KySat-2, is scheduled to launch from Wallops Island in the fall of 2013 [6] [7].

2.1.2 CubeSat Subsystems

Although CubeSats are small, they are still complex spacecraft that contain the same subsystems as large satellites. All of these subsystems must be designed together so that they can be correctly integrated together to achieve the satellite's mission. If any of them should fail, the satellite will no longer be able to achieve its mission. These subsystems include a Payload, an Electrical Power System, a Communication System, a Command and Data Handling System, and an Attitude Determination and Control System, which has been introduced in Chapter 1.

The satellite payload is usually the subsystem that is the justification for the mission. All of the other subsystems support the operation of the payload. A payload could be a scientific instrument taking a measurement of space, an earth imaging camera that is used to take imagery of the earth, or a communication transponder that receives a radio signal from earth and then retransmits it to earth, as seen on a communication satellite. A satellite's design requirements greatly depend on the payload's operational requirements, and other subsystems must be designed appropriately to facilitate operation of this payload on orbit.

The Electrical Power System, commonly known as the EPS, provides power to the rest of the subsystems. For a short duration, CubeSat missions of a few days or a few weeks, this system can be as simple as a bank of batteries. Most CubeSats, however, are more complex and have solar cells which can recharge the batteries using the energy from the sun. The EPS also must contain circuits to regulate the voltage in the batteries, inhibiting battery charging when they are at full charge, and protecting against short circuit conditions. Oftentimes the EPS contains a DC to DC power converter to increase or decrease the battery voltage level as required by the other subsystems.

The Communication System provides a method for controllers on the ground to communicate with the satellite, and for the satellite to transmit telemetry back to the ground. On CubeSats, the communication system consists of a radio, antennas, and antenna deployment systems. Larger satellites also

include encryption systems to protect transmissions from interception.

The Command and Data Handling (C&DH) subsystem is the main flight computer used to control the satellite. It executes commands sent from the ground, performs system-wide maintenance tasks, and controls the operations of the other subsystems. It can be considered the brain of the satellite.

The final subsystem, the Attitude Determination and Control Subsystem (ADCS) is the subsystem responsible for controlling the orientation of the satellite in space, and has been discussed in Chapter 1.

2.2 Coordinate Reference Frames

Coordinate reference frames are one of the fundamental concepts in attitude determination and control. Coordinate reference frames refer to the convention by which origin and the axes of an <x,y,z> Cartesian coordinate system are defined. In spacecraft design, 4 common sensor frames exist. See Figure 2.1 following the descriptions for a comprehensive diagram of these coordinate systems.

2.2.1 Earth Centered, Earth Fixed (ECEF)

ECEF coordinates have an origin point (0,0,0) at the center of the earth, with an axis collinear with the rotation axis of the earth and with other axes orthogonal to this one and each other. Several different conventions exist for ECEF coordinates, the main difference being that the points at which the two axes orthogonal to the Earth's rotation axis intersect the surface of the earth. Latitude and longitude are an example of ECEF coordinates, but one based on spherical coordinates and not the familiar <x,y,z> Cartesian coordinate system. In the Cartesian representation, the Z-axis is collinear with the Earth's rotation axis; the X-axis intersects the surface of the earth at the point where the equator and the prime meridian (the 0 degree longitude line) intersect. The third axis, the Y-axis, is orthogonal to the X and Z axis. It is important to note that the coordinate system rotates with the earth. Thus, a given point on the earth has a constant ECEF coordinate representation, which is what makes it earth fixed. Another name for ECEF is International Terrestrial Reference Frame, or ITRF. [8] [10]

2.2.2 Earth Center Inertial (ECI)

ECI coordinates have their origin at the center of the earth, but unlike the ECEF system, the axes are not referenced to a fixed position on the earth like the intersection of the equator to the prime meridian, but rather to this position at a fixed moment in time. In this thesis, the reference frame is the Z-axis that is collinear with the rotation axis through the North Pole, and the X-axis is orthogonal to the rotation axis, and passes through the point of intersection of the equator and the prime meridian at a specific instant in time: the vernal equinox, the longest day of the year. At this precise instant, ECEF and ECI coordinate systems are the same. As the earth continues to rotate however the ECEF moves with it, but the ECI frame, being inertial, does not. The ECI frame is often used to describe the local frame or orbital frame. At any instance in time, ECI coordinates will locate a local frame for the spacecraft [8].

2.2.3 Spacecraft or Body Fixed Coordinates

One of the most important coordinate systems in the ADCS field is spacecraft or body fixed coordinates. The origin of this system is the center of the spacecraft. The positive X-axis is in the nominal velocity vector direction, which is the forward flight direction of the spacecraft. The positive Z-axis is in the nominal nadir direction, which is the direction pointing towards the center of the earth. The positive Y- axis is the nominal orbital anti-normal, which is a vector that is negatively perpendicular to the orbital plane. The nominal stipulation refers to the fact that, in flight, these directions might not match the actual trajectory of the spacecraft if it is tumbling, for instance, but rather when it is flying in a normal trajectory, these are where the axes are oriented. Attitude determination is essentially the process of describing the rotation between the body fixed coordinates of the spacecraft and the local frame [8].

2.2.4 Roll, Pitch, and Yaw (RPY) Coordinates

Roll, Pitch, and Yaw (RPY) coordinates are widely used in aviation. In this system, angles represent the parameters of the coordinate. It shares the same

axes as the nominal body fixed orientation, and represents the orientation of the spacecraft in the local orbital frame as rotations around the body axis. The positive roll is defined as a clockwise rotation around the X, or velocity axis, as viewed by an observer at the body coordinate origin looking down the positive velocity axis. Pitch and Yaw are identically defined for Y (orbit anti normal) and Z (nadir vector) respectively. In RPY coordinates, the order of the rotations must be specified as well. A 123 sequence indicates a roll axis rotation, followed by a pitch axis rotation, and finally a yaw axis rotation. Many different conventions exist for the rotation sequences. Another term for roll, pitch, and yaw angles are Euler angle rotations [10] [8]. .

Sensor frame- Another frame of reference that is used in this paper is the sensor frame. In this frame, the origin is the detector of a sensor, with the x axis down the so-called bore sight, which can also be described as a line orthogonal to the surface of a detector, and the Y and Z axes being orthogonal to it by a given convention. This frame is useful in a simulation environment, where a vector measurement in ECI could be rotated into body frame, and subsequently rotated into the sensor frame, to provide a simulated sensor measurement.

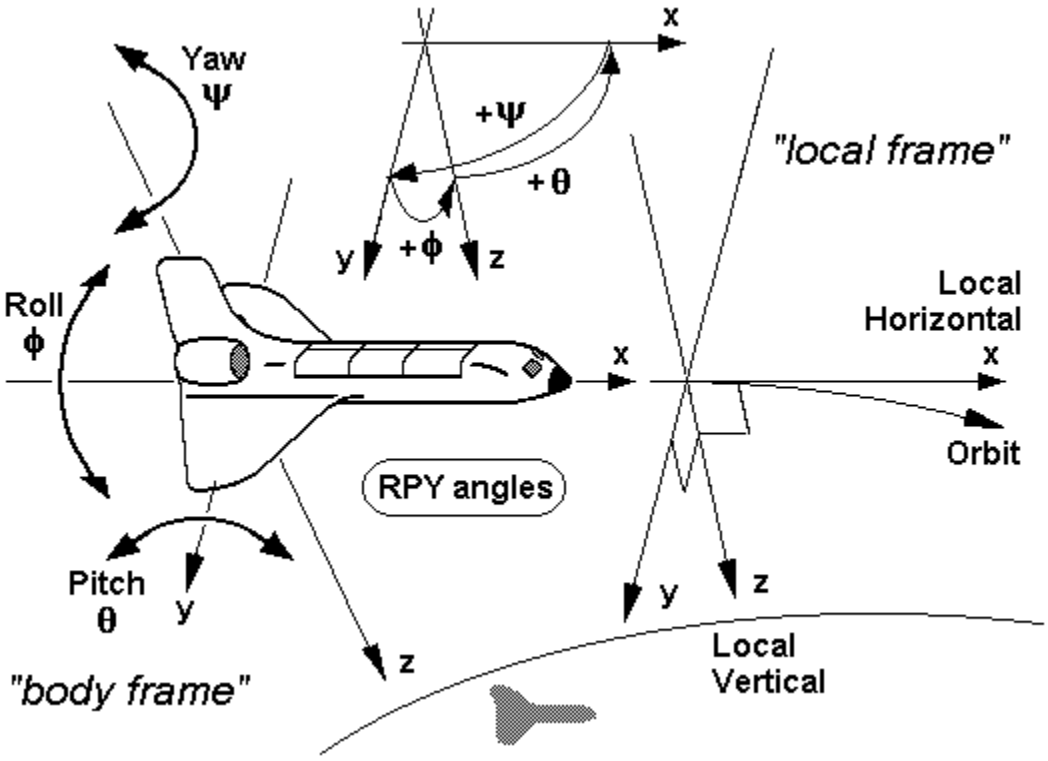

Figure 6: The body frame, along with RPY angles are shown, as well as the local frame convention. Source: Creative Commons.

2.3 Coordinate Rotation and Parameterizations

Switching between different coordinate frames is an integral part of ADCS. The entire process of determining satellite attitude is essentially determining a coordinate rotation between two different reference frames, usually the ECI orbital frame and satellite body frame. Orientation of a body in space, which is referred to as attitude, is a coordinate rotation. The primary tool for rotating between different frames is the Direction Cosine Matrix.

2.3.1 Direction Cosine Matrix (DCM)

The Direction Cosine Matrix (DCM) is a 3x3 matrix that is used to rotate a vector from one frame to another. The equation below shows the DCM being used to rotate a vector in the ECI frame to the body frame. A, in the equation below, is the DCM.

$$\begin{Bmatrix} x_{body} \\ y_{body} \\ z_{body} \end{Bmatrix} = [A_{eci-body}] \begin{Bmatrix} x_{eci} \\ y_{eci} \\ z_{eci} \end{Bmatrix}$$

If the matrix inverse is taken of the DCM, the DCM can be used to rotate a vector in the opposite direction, as shown below.

$$\begin{Bmatrix} x_{eci} \\ y_{eci} \\ z_{eci} \end{Bmatrix} = [A_{eci-body}]^{-1} \begin{Bmatrix} x_{body} \\ y_{body} \\ z_{body} \end{Bmatrix} = [A_{body-eci}] \begin{Bmatrix} x_{body} \\ y_{body} \\ z_{body} \end{Bmatrix}$$

The DCM has an additional property of being an orthogonal matrix. With an orthogonal matrix, the transpose of the matrix is also the inverse. This relationship, and its utility, is shown below [11].

$$[A_{eci-body}]^{T} = [A_{eci-body}]^{-1} = [A_{body-eci}]$$

To carry out sequential rotations between different frames, DCMs can be multiplied together. For instance, to rotate from ECI to Body Frame to Sensor Frame, it's possible to calculate the following DCM [11].

$$[A_{body-sensor}] = [A_{body-sensor}] x [A_{eci-body}]$$

$$\begin{Bmatrix} x_{sensor} \\ y_{sensor} \\ z_{sensor} \end{Bmatrix} = [A_{eci-body}] \times \begin{Bmatrix} x_{eci} \\ y_{eci} \\ z_{eci} \end{Bmatrix}$$

2.3.2 Euler Angle Parameterization of the DCM

A variety of different parameterizations exist for the DCM. The most intuitive is the Euler angle parameterization. The Roll, Pitch, and Yaw angles are examples of Euler angles. Euler angles describe a rotation in terms of three parameters: roll (φ), pitch (θ), and yaw (ψ). To fully describe the DCM as parameterized by Euler Angles, it is also necessary to describe the rotation order. An Euler angle is not fully described unless the order in which the rotations are carried out as specified. For instance, R-P-Y, or 1-2-3, describe a roll rotation followed by a pitch, followed by a yaw. A DCM for a 1-2-3 rotation sequence is

shown below [12].

$$[A_{123}] = [A_{\varphi\theta\psi}]$$

$$= \begin{bmatrix} \sin\psi\cos\theta & \cos\psi\sin\theta + \sin\psi\cos\varphi & -\cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ -\sin\psi\cos\theta & -\sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi + \cos\psi\sin\varphi \\ \sin\theta & -\cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix}$$

Although Euler Angles are useful and intuitive, they have a critical limitation. Due to the trigonometric functions in the DCM, singularities are encountered in certain orientations, such as 0 or 90 degrees. In these orientations, sine and cosine functions respectively equal zero. This leads to a loss of attitude knowledge, as terms in the DCM go to zero, whenever a term is the product of sine or cosine [13].

2.3.3 Quaternion Parameterization of the DCM

The DCM could also be parameterized in terms of quaternions. Quaternions, also known as Euler symmetric parameters, are very useful in ADCS, although they are a much more abstract representation than the Euler angles. Quaternions are particularly useful because they do not contain the singularities of an Euler Angle representation, due to the lack of trigonometric functions. Quaternions originate from the concept of the Euler axis, which holds that any rotation or sequence of rotations can be represented as a single rotation about an axis called the Euler axis. A single quaternion is shown below.

$$\boldsymbol{q} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} q_v \\ q_4 \end{Bmatrix}$$

A quaternion has two distinct parts. The first 3 elements, the components of the vector q_v, represent an Euler axis of rotation. The fourth element, q_4, represents the magnitude of the rotation around the Euler axis, Each element in the quaternion can be calculated given knowledge of the Euler axis and the angle of rotation around this axis.

$$q_1 = e_1 \sin\left(\frac{\alpha}{2}\right)$$

$$q_2 = e_2 \sin\left(\frac{\alpha}{2}\right)$$

$$q_3 = e_3 \sin\left(\frac{\alpha}{2}\right)$$

$$q_4 = \cos\left(\frac{\alpha}{2}\right)$$

In these equations, e_1, e_2, and e_3, are the elements of a unit vector that is collinear with the Euler axis. The parameter α represents the angle of the rotation around this vector [12].

Quaternions are subject to the unit norm constraint. This constraint holds that the norm or length of a quaternion must equal unity, as shown below.

$$|\boldsymbol{q}| = \sqrt{{q_1}^2 + {q_2}^2 + {q_3}^2 + {q_4}^2} = 1$$

Another useful relationship for the quaternion is the quaternion conjugate or inverse. This relationship is shown below.

$$q^* = \begin{vmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{vmatrix}$$

A conjugate or inverse is calculated simply by negating the vector components of the quaternion. Physically, the inverse represents the same rotation as the original quaternion, but in the opposite direction [11].

In order to rotate a vector between coordinate frames, the direction cosine matrix can be parameterized in terms of a quaternion, as shown below.

$$[A(\mathbf{q})] = \begin{bmatrix} {q_1}^2 + {q_2}^2 + {q_3}^2 + {q_4}^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -{q_1}^2 + {q_2}^2 + {q_3}^2 + {q_4}^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 + q_1q_4) & -{q_1}^2 - {q_2}^2 + {q_3}^2 + {q_4}^2 \end{bmatrix}$$

With quaternions, it's much easier to carry out the sequential rotations. In terms of the DCM, a sequential rotation is represented below.

$$[A(\mathbf{q_k})] = \left[A\left(\mathbf{q_j}\right)\right] * [A(\mathbf{q_i})]$$

Instead of multiplying the DCM directly, it is possible to multiply the quaternion as shown below [12].

$$\mathbf{q_k} = \begin{Bmatrix} q_{k1} \\ q_{k2} \\ q_{k3} \\ q_{k4} \end{Bmatrix} = \begin{bmatrix} q_{j4} & q_{j3} & -q_{j2} & q_{j1} \\ -q_{j3} & q_{j4} & q_{j1} & q_{j2} \\ q_{j2} & -q_{j1} & q_{j4} & q_{j3} \\ -q_{j1} & -q_{j2} & -q_{j3} & q_{j4} \end{bmatrix} \begin{Bmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \\ q_{i4} \end{Bmatrix}$$

2.4 An Overview of Spacecraft Dynamics

The modeling of spacecraft dynamics is extremely important in ADCS. Spacecraft dynamics equations are given by a set of ordinary differential equations that predict the effect of applied torques from actuators and the environment on the attitude of the spacecraft. Applying a torque to a spacecraft generates angular acceleration, which increases the rotation rates of the spacecraft. The dynamics equations permit determination of spacecraft orientation at a given time if the applied torques are known, along with the current angular momentum in the system and the angular rates.

2.4.1 An Overview of Rigid Body Dynamics

The motion of a spacecraft is analyzed using rigid body dynamics. In rigid body dynamics, a rotating body has angular momentum that is proportional to the angular velocity as shown below.

$$L = I\boldsymbol{\omega}$$

Each element in the angular momentum vector, L, represents the angular momentum in the x, y, or z body axes. Similarly, each component in the angular velocity vector, $\boldsymbol{\omega}$, represents the angular rates in the x, y, or z body axes. They are directly proportional through the constant matrix, I. I has a special name. It is called the moment of inertia tensor, which is a matrix that translates the angular velocity in each of the body axes to the resulting body axis angular momentum vector component. If our rigid body is perfectly symmetric, the moment of inertia tensor is simply the identity matrix. When this is true, the angular momentum in each axis is simply a scalar multiple of the angular velocity in this axis. In most real world situations, however, this is not the case, and the other components of angular velocity will factor into determining the angular momentum component in a given body axis.

The rate of change of the angular momentum vector is a quantity called torque. Thus applying a torque, N, leads to angular acceleration. [14].

$$\frac{d}{dt}[L] = N$$

2.4.2 Kinematics and Dynamics

Dynamics equations have two distinct components: the dynamics equations themselves and the kinematics equations. Kinematics equations describe the motion of the spacecraft in terms of angular rates which are assumed to be known. These angular rates, imparted by applied torques, do not have a distinct origin in the equations. Instead the equations incorporate angular rates which are provided with no knowledge of how they were generated. In other words, a given sum of torques produces some angular rates on the spacecraft, which are used to predict the orientation of the spacecraft [14]. In order to determine these angular rates, the dynamics equations are used to account for all torques on the spacecraft and all existing angular momentum. Integrating the dynamics equation provides angular rates that can used to solve the kinematics equation, allowing us to propagate the spacecraft orientation. The dynamics equation is given below [15].

$$\frac{d}{dt}[I\omega] = N_{disturbance} + N_{control} - \omega \times I\omega - [\omega \times h + N_{internal}]$$

In this dynamics equation, I is the angular momentum tensor, a 3 x 3 matrix. $N_{disturbance}$ and $N_{control}$ are disturbance and control torques, respectively. The cross product operation is indicated by the $\times$ operator. h is the total angular momentum of the any internal rotating mechanisms, and $N_{internal}$ is the torque generated by accelerations of any internal rotating mechanisms in the spacecraft. After using this equation to find the angular velocity vector, the kinematics equation is used to calculate the current attitude of the spacecraft. This differential equation is shown below. Attitude is parameterized in terms of the quaternion [11].

$$\frac{d\boldsymbol{q}}{dt} = \frac{1}{2}\Omega\boldsymbol{q}$$

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}$$

2.4.3 Sources of Disturbance Torques

A spacecraft in orbit is affected by several different types of torques from the space environment, which cause deviations from the desired attitude. One of the main purposes of an ADCS is to maintain the attitude of the satellite in spite of these so called disturbance torques. Disturbance torques fall into two categories: cyclic and constant. Cyclic torques are periodic over the course of an orbit. In other words, they have a pattern of minima and maxima over the course of an orbit that repeat each orbital period. On the other hand, constant torques do not change over the course of an orbit. They have the same value during orbit 10 as they have on orbit 100 or 1000. Several different types of disturbance torques exist [8].

Aerodynamic torques are caused by atmospheric drag. Although the atmosphere as experienced on the earth's surface doesn't exist in space, trace amounts of oxygen and nitrogen are present in the orbital environment. These trace amounts, taken collectively, can induce noticeable drag on a spacecraft. When a spacecraft is not symmetrical, the imbalance of drag forces across the surface of the spacecraft induces these torques on the surface of the spacecraft. The equation for aerodynamic torque is given below [8].

$$T_a = F(c_{pa} - cg)$$

$$F = \frac{1}{2}(\rho C_d A V^2)$$

In this equation, ρ is the atmospheric density, c_{pa} is the center of aerodynamic presentation, A is the surface area, V is the forward velocity, and cg is the center of gravity. This equation shows that as the velocity or surface area increases, the aerodynamic torque also increases, but only as long as a difference exists between the center of pressure and the center of gravity. If this

quantity is zero, implying a balanced symmetrical spacecraft, then surface area and velocity can be very large and not induce aerodynamic torques. Aerodynamic torques are constant given a constant orbit altitude. As orbit altitude increases, the atmospheric density decreases, leading to a subsequent decrease in atmospheric drag. Thus, variable altitude orbits, such as a Molniya orbits, are cyclical, due to the periodic changes of the atmospheric density throughout the orbital period.

Solar radiation represents another type of disturbance torque. Solar radiation, hitting the surface of a spacecraft, can generate drag in much the same way as residual atmosphere does with aerodynamic torques. The equation is shown below.

$$T_{sp} = F(c_{ps} - cg)$$

$$F = \frac{F_s}{c} A_s (1 + q) \cos\theta$$

F_s is the solar constant, c is the speed of light, A_s is the surface area of the spacecraft, cg is the center of gravity, q is the surface reflectance factor, ranging from 0 to 1, and θ is the angle to the sun. The greater the angle to the sun, the lower the solar radiation torques, due to the cosine relationship. Larger surface area increases the torque as well, as does reduced reflectance of the spacecraft surface. As with the aerodynamic torque, the quantity represented by the difference between the center of solar pressure and the center of gravity is crucial. If this quantity is very small, indicating that these two points closely coincide, solar radiation will not generate a large disturbance torque on the spacecraft. Solar radiation torque is cyclic given a nadir or zenith pointing spacecraft, due to the periodic presence or absence of the sun [8].

The Earth's magnetic fields also lead to disturbance torques. These occur due to interactions between magnetic field dipoles contained within the spacecraft and the earth's magnetic field. These magnetic dipoles arise from permanent magnets, electric motor windings, torque coils, and current loops in the spacecraft electronics [8].

Gravity gradient torque is a fourth source of disturbance torque, which can

be used to provide stabilization of certain spacecraft designs. Gravity gradient torques result from the differential effect of gravity on an object with a center of mass that is offset from the center of volume. Given an object with a high aspect ratio, a term that refers to an object that is much longer in one dimension than any other, gravity gradient torques will cause such an object to orient itself such that the long axis will point towards the center of the earth. It will tend to remain in this position unless acted on by other torques. Given certain spacecraft geometries, this can provide a method of stabilization. If it is not desirable to have the long axis oriented towards the center of the earth, then an ADCS must compensate for this disturbance torque [11].

2.5 Attitude Determination and Control System Components

An Attitude Determination and Control subsystem has two main hardware components. These components are actuators, which generate torques to change the attitude, and sensors that provide indirect measurements of the attitude that an algorithm can process to determine the attitude.

2.5.1 Actuators

Thrusters are gas jets that produce thrust by ejecting propellant. The thrust from this propellant ejection generates an external torque that changes the spacecraft attitude. Thrusters fall into two different categories. Cold gas thrusters provide thrust through the expansion of a propellant that is compressed inside a storage tank. This thrust is called an impulse. Hot gas thrusters, on the other hand, generate thrust through a chemical reaction. Hot gas thrusters can be further classified as monopropellant or bipropellant. Bipropellant thrusters combine two different chemicals that react to produce thrust. Monopropellant thrusters use just one chemical that undergoes a reaction to produce thrust. Hot gas systems generally produce higher thrust than cold gas systems. Cold gas thrusters are generally better for fine control, since they produce smaller thrusts. The major limitation of thrusters is that they depend on a finite fuel supply. Once this fuel is depleted, the attitude control capability is lost on the spacecraft unless

other types of actuators are present [11].

Reaction wheels are often used to produce torques to slew a spacecraft. A reaction wheel is an electric motor attached to a flywheel with significant inertia. Accelerating the rotation of the flywheel generates an external torque on the spacecraft due to the principle of conservation of angular momentum. Reaction wheels are often placed in a three-axis configuration corresponding to the three body rotation axes of the spacecraft. Torque can then be generated in each body axis of the satellite. They are particularly useful on spacecraft, since they run on electricity that can be generated onboard the spacecraft using solar panels. Thus, they don't depend on consumable resources like propellant in the case of thrusters. During normal operations, reaction wheels periodically need to be spun down when they reach their maximum speed. When they reach this maximum speed, they are saturated and can no longer be spun up to provide a torque. In order to be used again, they must be spun down in a technique known as momentum dumping. To complete a momentum dumping operation, an additional set of actuators that can provide compensating torque is required [8].

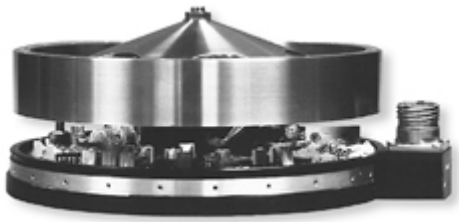

Figure 7: An example of a reaction wheel. Note the large flywheel. Source: NASA

Magnetorquers are electromagnets that interact with the earth's magnetic field to produce a torque to rotate a spacecraft. Since they are electromagnets, the magnetic field will disappear when power is removed by the control system. The torque will then disappear. Torque rods are typically arranged in a three-axis configuration to provide torque in all three axes. In a three-axis stabilized system using reaction wheels, it is common to use a set of torque rods for momentum dumping. A spacecraft must perform momentum dumping when its reaction wheels reach their maximum rotational speed. This stored momentum must be

“dumped” by decelerating, or spinning down, the wheels. Due to conservation of angular momentum, this deceleration would normally cause the spacecraft to rotate back to its original orientation. In the momentum dumping process, magnetorquers are turned on immediately before the reaction wheels are spun down, which holds the spacecraft in its original orientation as the reaction wheels decelerate [13] [16].

Figure 8: An example of magnetorquers. Note that they are essentially tightly wound coils of wire that become an electromagnet when energized. Source: Wikimedia Commons

Control moment gyros are similar to reaction wheels since they both have electric motors that accelerate a mass. A control moment gyro (CMG) differs from a reaction wheel in that this wheel and motor is mounted on a set of gimbals that can rotate the apparatus. The wheel motor combination produces an angular moment that has a constant orientation in inertial space. A CMG takes advantage of this fact, by using the gimbals to move the spacecraft body around this constant angular moment vector. Unlike reaction wheels, CMGs do not require momentum dumping, since they can spin at a constant rate to produce the crucial angular moment. Additionally, one CMG can take the place of a three-axis reaction wheel system, which leads to power saving efficiency, although this comes at the cost of the additional complexity of the CMG system. CMGs have

not yet been flown on CubeSats, although their efficiency has been compared to three-axis reaction wheels in the research [17], and units have been demonstrated on the laboratory bench top [18].

2.5.2 Sensors

Attitude determination systems utilize several different types of sensors. One of the most crucial is the gyroscope. Gyroscopes measure the angular rotation of the spacecraft in the 3 body axes. Gyroscopes are used for attitude propagation. Propagation involves tracking the changes in orientation from a known point. By measuring the changes in orientation from a starting point using a gyroscope, it's possible to maintain full attitude knowledge through propagation. Traditionally gyroscopes have been mechanical devices that contain a spinning mass that rotates in 3 or more axes. The joint where rotation occurs is called a gimbal. The spinning mass maintains a constant angular momentum vector in inertial space aligned with the spin axis. As the spacecraft orientation changes, it is possible to measure the angular change by the rotation of the gimbals. A unique problem with gyroscope is gimbal lock. It is possible for the spacecraft to enter certain orientations where the 2 or more gimbal axes are aligned. When this occurs, rotations in either aligned axis appear identical in the other axis, and as a result, a degree of attitude knowledge is lost [19]. Due to the power and mass limitations of CubeSats, it is not feasible to use large mechanical gyroscopes. Instead, MEMS (Microelectronic Mechanical Systems) gyroscopes must be used. MEMS gyros provide an angular rate output that must be integrated to determine the orientation of the spacecraft. MEMS gyros are small enough to fit inside a single integrated circuit package and are low power. Instead of using a spinning mass, MEMS gyros use a miniature piezoelectric oscillating mass. Motion caused by a centrifugal force due to motion disturbs the mass, and this disturbance can be correlated with angular motion. This is commonly known as the Coriolis Effect [13].

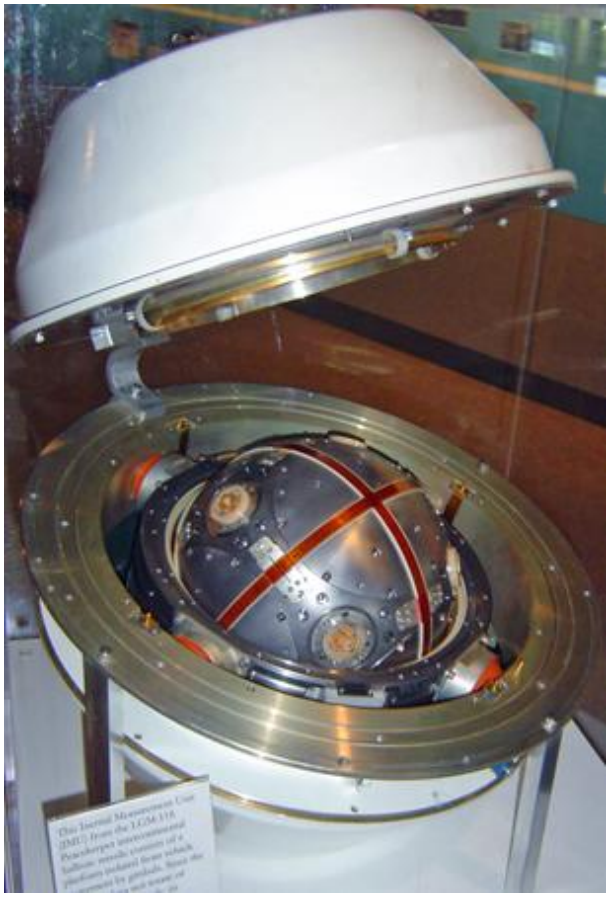

Figure 9: Gyro from a Jupiter IRBM showing gimbals. Source: USAF

Although MEMs gyros have the advantage of compactness and low power consumption, they suffer from a problem known as drift. Drift occurs due to thermal variations that cause the vibrating thermal mass's mechanical properties to fluctuate over time, leading to a systematic error.

Magnetometers are another widely used sensor in attitude determination. A magnetometer is essentially a compass that measures the current direction of the earth's magnetic field. Assuming the spacecraft knows it's location in space, through either an onboard GPS or accurate knowledge of the time and the spacecraft's orbital parameters, the earth's magnetic field can be calculated. Comparing the measured magnetic field to the calculated field provides attitude knowledge. Ambiguity remains in the attitude knowledge, however, because although it's possible to establish the spacecraft orientation in terms of this

magnetic field vector, the spacecraft can still rotate around this vector without the vector measurement changing. Thus, another vector measurement is needed to fully establish attitude knowledge [11] [12].

A good candidate for a second observation is the spacecraft to sun vector, a vector that is determined using a sun sensor. As with the magnetometer, using a sun sensor requires accurate knowledge of the spacecraft's current position, so that the sun vectors' position in the local ECI frame can be determined and subsequently correlated with the measured sun vector. Sun sensors are particularly useful because the sun's intensity or luminosity is often constant over the course of a spacecraft's orbit, and it is always the brightest object in the sky, a situation the makes it very distinct in the sky. Sun sensors can be either analog or digital devices. Analog sensors have a slit and a photo diode that detects the sun. The output current from the photo diode generally varies with a cosine relationship to the incident angle with the sun. Digital sun sensors contain a slit with an array of photo detectors beneath, with each pixel representing a single digital bit [12]. Sun sensor implementations on CubeSats are often staring type sensors. These sensors normally involve multiple discrete photo diodes on each face of the CubeSat. These photodiodes have overlapping fields of view, and thus overlapping response curves. Comparing the outputs from each sensor on a face mathematically establishes the sun's position. This sensor type was demonstrated on orbit on Boeing's CubeSat Testbed 1 in 2007 [20]. Sun sensors suffer an obvious limitation in that they require the sun to be visible in order to work. When a spacecraft is in eclipse, they do not provide meaningful data.

Star sensors are widely used on larger satellites, and are particularly useful because they provide a complete attitude determination solution and additional angular rate measurements. A typical star sensor output would be a quaternion that represents the rotation between the ECI frame and the star sensor frame [21]. Star sensors are one of the more complex attitude determination sensors. They essentially use a CMOS or CCD sensor array to image the sky, and then use image processing techniques to pick out certain stars. Every star sensor has

an onboard star catalog that it uses to identify the orientation of the spacecraft in inertial space by comparing the visible star pattern to the onboard catalog. The sun must be well out of the star sensor's field of view to prevent potential damage to the CCD or CMOS sensors. A light shade, also known as a baffle, must be installed to protect the sensor from the sun [22]. This device, although necessary for effective operation of the Star sensor, is very large and must protrude from the spacecraft body. Given the constraints of the CubeSat form factor driven by the P-Pod, large protruding objects can be difficult to implement on the CubeSat. Thus using a star sensor on a CubeSat can be complicated. An additional issue with the star sensor is that they tend to be power hungry, due both to the computational requirements of the star sensor algorithms, and the fact that the sensors must often be cooled using thermo electric coolers to decrease inaccuracies associated with dark current flow in the semiconductor pixels [23].

Figure 11: A set of three Star sensors with attached baffles.
Source: NASA

Earth Sensors are another type of attitude determination sensor. These work by detecting the presence of the earth in the sensor's field of view. This normally involves detecting the contrast between the earth's warm surface and the approximate 0 Kelvin temperature of space. Earth sensors are implemented in a number of different ways. Horizon sensors detect the earth's horizon, and based on the orientation of the earth's crescent in the field of view, determine the spacecraft orientation in roll and pitch axes [14]. The specific sensor inside the earth sensor are often thermopile devices that output a DC current proportional to the infrared energy absorbed by the sensor. The sensor design is complicated due to the varying infrared emission from the earth that tends to change according to the presence of clouds or whether the portion of the Earth's surface in view is desert, ocean, or temperate. Existing earth sensors are typically very large and power hungry, although devices have been proposed that are appropriate for the CubeSat form factor [24] [25]. One of the most useful characteristics of earth sensors is that they work independent of eclipse. To date, in-flight demonstration of an earth sensor for CubeSats has not been described

in the literature.

A novel attitude determination device, that has great potential for CubeSat applications, is the stellar gyro system. From a hardware perspective, the stellar gyro is a CMOS or CCD device that can image star fields. Using onboard processing, the self-contained stellar gyro sensor uses image processing algorithms to identify stars that appear in two different star field images. By comparing the movement of the stars between images, it is possible to determine the relative attitude of the spacecraft between the frames. The measurement is relative. In other words, the output is a quaternion that reflects a rotation between the two star field images. In order for the stellar gyro to be used, an initial attitude must be known. The stellar gyro can then provide a relative orientation from this known starting attitude. A major advantage of the stellar gyro is that it works equally well in eclipse or in the presence of the sun. Additionally, it doesn't require detection of specific stars or patterns of stars to match those found in a database [26] [27] [28].

2.6 Overview of Previous CubeSats with ADAC Systems

Previous work on CubeSat ADCS normally presents results and designs of systems for specific spacecraft. Papers on the Aeneas bus show the capabilities of the three-axis stabilized CubeSat in Low Earth Orbit [16]. A sun tracking, three-axis stabilized CubeSat is shown in the paper by the Aerospace corporation [4]. A description of attitude determination using sun vectors and magnetometer vectors in a flown implementation is presented in the paper about the Boeing CubeSat validation efforts [20]. The simulation work in this thesis work builds heavily off of the thesis work of Samir Rawashdeh, completed in 2009. In this thesis, the Smart Nanosatellite Attitude Propagator is developed. The underlying dynamic model and environmental torque models were leveraged in completing this thesis [9]. The thesis completed by Orlando Diaz presents a comparison of different attitude propagation algorithms that helped to select the attitude propagation algorithm in this thesis [29]. Theses completed at the University of Toronto Institute of Aerospace Studies (UTIAS) present very good

coverage of the capabilities and the data provided by attitude determination sensors. This provided much insight towards developing the sensor simulator models in this paper [21] [15]. Finally, previous work on the stellar gyroscope completed at the University of Kentucky Space Systems Lab laid the groundwork for development of the system utilizing this sensor [27] [26].

3 Attitude Determination Algorithms

Up to now, the discussion has revolved around attitude determination and control systems in general, with some specific discussions of sensors and actuators. The algorithms involved have only been mentioned in passing. The crux of this work is integration of attitude determination algorithms to build a complete attitude determination system, so some in depth discussion of the mathematics is required. The mathematics are then implemented in software running on a spacecraft computer to complete the attitude determination system.

Two complementary areas comprising attitude determination that will be described are absolute attitude determination and attitude propagation. The algorithms described here rely on these two different types of algorithms to work. An absolute attitude determination algorithm provides a measurement of the absolute attitude based on vector measurements. An attitude propagation algorithm is then used to maintain attitude knowledge using gyroscope measurements. This propagation will periodically need to be reset, since it will tend to diverge from the true attitude, a situation known as drift. Resetting the algorithm involves providing it with a new initial estimate through the absolute attitude determination algorithm. This strategy is used because over the course of an orbit, multiple vector measurements are not always available to execute an absolute attitude determination fix. During these lapses, attitude knowledge must be propagated.

3.1 Absolute Attitude Determination

This section will introduce the problem and the subsequent solution to the absolute attitude determination problem. This problem was first identified in the 1960s, and finding solutions to this problem is still an active area of research.

3.1.1 Wahba's Problem

The basic attitude determination problem was first posed in 1965 by Grace Wahba, an applied mathematician working for NASA. The problem is stated as

the following: given a series of vector measurements in spacecraft frame and a representation of the same vectors in a different frame such as ECI, minimize the loss function given below.

$$L(A) = \frac{1}{2}\sum a_i \, |\boldsymbol{b_i} - A\boldsymbol{r}_i|^2$$

In this equation, $\mathbf{b_i}$ is a vector in the spacecraft body frame, and $\mathbf{r_i}$ is the same vector in a reference frame such as ECI, *A* is a direction cosine matrix parameterized by a quaternion, and a_i is a weighting factor. When the quantity *L* is minimized through manipulating the quaternion that parameterizes the attitude matrix *A*, the resulting sum of squares will provide a statistically optimal estimate of the true spacecraft orientation. Explained in a more intuitive way, if a given quaternion is close to the actual attitude quaternion, $\mathbf{r_i}$ will be rotated such that it is very close to $\mathbf{b_i}$, and the resulting quantity will be very small when the body and the reference frames are subtracted. This process is repeated for each vector measurement. The resulting sum of squares will thus be minimized, which provides the quaternion that is closest to the actual system state. This problem description is very realistic for spacecraft attitude determination because the output from most types of attitude determination sensors is a vector measurement in the sensor frame. Another term for this type of attitude determination is statistical attitude determination. [11] [30]

3.1.2 Solutions to Wahba's Problem: Davenport's q-Method

Several different solutions to Wahba's problem exist. One of the most widely used is Davenport's q-Method. This method is a starting point for other solutions to Wahba's problem as well. The starting point in the derivation is a reformulation of the loss function shown above. This is shown below.

$$L(A) = \lambda_0 - tr(AB^T)$$

$$\lambda_0 = \sum a_i$$

$$B = \sum a_i \boldsymbol{b_i} \boldsymbol{r_i^T}$$

A is the direction cosine matrix, a_i,$\boldsymbol{b_i}$, $\boldsymbol{r}_i$ are a weighting factor, a reference frame vector, and a body frame vector, respectively. The *T* superscript indicates a

vector transpose, and $tr(AB^T)$ is the trace, or sum of diagonal elements of the matrix outer product. In order to minimize the Loss function, the negative quantity $tr(AB^T)$ should be maximized. The term to be maximized is given below.8

$$L'(A) = -tr(AB^T)$$

The next step in the derivation is to substitute in a quaternion parameterized direction cosine matrix. The form of the direction cosine matrix for this substitution is given below [30].

$$A(q) = (q_4^2 - \boldsymbol{q} * \boldsymbol{q}) + 2\boldsymbol{q} * \boldsymbol{q^T} - 2q_4 * Q$$

$$Q = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

In this formulation of the direction cosine matrix, which is numerically identical to the previously given direction cosine matrix, $\boldsymbol{q}$ is the quaternion, q_4 is the scalar portion of the quaternion, and Q is the skew symmetric matrix of the first three quaternion elements. Substituting this direction cosine matrix into the trace of the loss function produces the following form of the new loss function $L'(A)$ [11].

$$L'(A) = \boldsymbol{q^T} K \boldsymbol{q}$$

The matrix *K* and all the constituent terms are defined below. *I* is the identity matrix.

$$K = \begin{bmatrix} S - I * tr(B) & \boldsymbol{z} \\ \boldsymbol{z^T} & tr(B) \end{bmatrix}$$

$$S = B + B^T$$

$$\boldsymbol{z} = \begin{Bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{Bmatrix}$$

Next, the maximum extrema of the new loss function $L'(A)$ is found using the method of Lagrange multipliers. The following function is derived.

$$g(\boldsymbol{q}) = \boldsymbol{q^T} K \boldsymbol{q} - \lambda \boldsymbol{q^T} \boldsymbol{q}$$

In this equation, λ represents a Lagrangian multiplier, and the assumption is made that $\boldsymbol{q^T q}$ is equal to 1. The maxima of this function is found by taking the derivative of it according to $\boldsymbol{q^T}$and setting the result equal to zero. This produces the following equation.

$$K\boldsymbol{q} = \lambda\boldsymbol{q}$$

This is the familiar eigenvalue problem. This result is then inserted into the minimal loss function as shown below.

$$L'(A) = \boldsymbol{q}^{\boldsymbol{T}} K\boldsymbol{q} = \boldsymbol{q}^{\boldsymbol{T}}\lambda\boldsymbol{q} = \lambda$$

To summarize, the modified loss function is at a maximum when the largest eigenvalue of K is chosen [11]. The optimal quaternion is found by defining the eigenvector corresponding to the maximum eigenvalue of the matrix K. This solution assumes that all eigenvectors of K are distinct. In order to ensure this, a minimum of two non-collinear vector pairs must be available, a situation that is consistent with the intuitive explanation given earlier [31].

3.1.3 Solutions to Wahba's Problem: QUEST Algorithm

The q-Method provides a robust method of attitude determination, but it is less computationally efficient than other methods, mostly due to the need to calculate exact eigenvalues. Another method that is widely used and more computationally efficient is the QUEST algorithm [31]. QUEST was developed specifically for the MAGSAT mission in 1978, a mission that mapped the earth's magnetic field. In order to achieve a high fidelity mapping of the earth's magnetic field, it was necessary to frequently measure the magnetic field. In order to do meaningful analysis of this data, it was necessary to calculate the attitude of the spacecraft each time the magnetic field state was measured. The computers at the time were not capable of calculating an attitude solution this rapidly using the q-Method, so a new, more rapid algorithm was required. QUEST, developed by Malcolm D. Shuster, was developed as a solution [32].

The development of the Quest algorithm is conveniently a modification of the q-Method algorithm, so the initial derivation is the same. It essentially modifies the q-Method by utilizing an iterative method to approximate the maximum eigenvalue. Recalling that

$$\lambda_{\boldsymbol{max}} = \boldsymbol{q}^{\boldsymbol{T}} K\boldsymbol{q} = \boldsymbol{q}^{\boldsymbol{T}}\lambda_{max}\boldsymbol{q}$$

And that the loss function can be rewritten as

$$L(A) = \lambda_0 - \ tr(AB^T) = \lambda_0 - g(A)$$

And the function $g_A(A)$, can be rewritten, when the DCM as is parameterized by a quaternion as

$$g_A(A) = \boldsymbol{q^T} K\ \boldsymbol{q}$$

Then we rewrite the maximum eigenvalue function as

$$\lambda_{\boldsymbol{max}} = \lambda_0 - \ L(A)$$

Since

$$\lambda_0 - L(A) = g_A(A) = \boldsymbol{q^T} K \boldsymbol{q}$$

The important result here is that, given an optimal quaternion selection, $\lambda_{\boldsymbol{max}}$will be very close to λ_0, the sum of weighting factors. This situation arises because the loss function $L(A)$ will be minimized and thus much smaller than λ_0, given the optimal quaternion selection. It is now possible to write

$$\lambda_{\boldsymbol{max}} = \lambda_0$$

This result is important, it means that the sum of weighting factors can serve as an initial guess later when the Newton-Raphson Method is used to solve for the result $\lambda_{\boldsymbol{max}}$.

The next task is to derive the equation to solve using the Newton-Raphson method. The starting point for this is the following equation, developed previously.

$$K\boldsymbol{q} = \lambda \boldsymbol{q}$$

Carrying out the algebra in the above equation produces the following equation.

$$\left[\left(\lambda_{max} + tr(B)\right)I - S\right]\boldsymbol{q_v} = q_4 \boldsymbol{z}$$

It is important to remember that the quaternion can be described as a scalar part, q_4, and $\boldsymbol{q_v}$, the vector part of the quaternion.

Rearranging the previous equation, gives another form

$$\boldsymbol{q_v} = q_4((\lambda_{max} + tr(B))I - S)^{-1}\ \boldsymbol{z}$$

$$= \frac{q_4}{det((\lambda_{max} + tr(B))I - S)}\, adj[((\lambda_{max} + tr(B))I - S)]\boldsymbol{z}$$

The operators *det* and *adj* indicate the determinant and the adjoint of the quantity, respectively. The $\boldsymbol{q_v}$ term can be combined with the scalar term, q_4, to

produce the optimal quaternion.

$$\boldsymbol{q_{optimal}} = \frac{1}{\sqrt{\gamma^2 + |\boldsymbol{x}|^2}} \begin{pmatrix} \boldsymbol{x} \\ \gamma \end{pmatrix}$$

The division by the quadratic term is a normalization step, meaning that the magnitude of the quaternion is one. The vector $\boldsymbol{x}$ and the scalar γ are defined below.

$$\boldsymbol{x} = adj[(\lambda_{max} + tr(B))I - S]\boldsymbol{z} = [\alpha I + (\lambda_{max} + tr(B))S + S^2]\boldsymbol{z}$$

$$\gamma = det((\lambda_{max} + tr(B))I - S) = \alpha[\lambda_{max} + tr(B)] - \det(S)$$

$$\alpha = \lambda_{max}^2 - [tr(B)]^2 + tr(adj(S))$$

The resulting form of these equations is a result of the Cayley-Hamilton Theorem. The $\boldsymbol{q_{optimal}}$ equation and the $\boldsymbol{x}$ equation are then inserted into the following equation, which results from manipulation of the $K\boldsymbol{q} = \lambda\boldsymbol{q}$ equation.

$$(\lambda_{max} - adj(B))q_4 = \boldsymbol{q_v^2}$$

The resulting equation produces the characteristic equation that can be solved for λ_{max}.

$$\Psi(\lambda_{max}) = \gamma[(\lambda_{max} - tr(B)] - \boldsymbol{z^T}[\alpha I + (\lambda_{max} - tr(B)S + S^2]\boldsymbol{z}$$

At this point, the Newton-Raphson method is used to solve for λ_{max}, with λ_0 as the initial guess. This initial guess selection generally results in a convergence to the optimal solution in a single iteration. This initial guess made the Quest approximately 1000 times faster than the q-method on the computing equipment available at the time of the MAGSAT mission [33].

3.2 Attitude Propagation

Attitude propagation refers to a technique that a spacecraft uses to maintain attitude knowledge when an absolute fix is not available. This propagation process involves measuring changes in attitude using gyroscopes, and then integrating these measurements to determine the change in attitude from an initial fix. Mathematically, propagation is carried out using a class of system models called state space models, and a special type of algorithm called an estimator. The estimator used in this case is a Kalman Filter.

3.2.1 State Space Models

Models of the spacecraft system dynamics are an integral part of the Kalman Filter design. These models are a series of differential equations that describe the evolution of system behavior given different inputs and internal conditions. An example of a differential system model is given below.

$$\frac{dx_1}{dt} = a_{11}x_1 + a_{12}x_2 + b_1u(t)$$

$$\frac{dx_2}{dt} = a_{21}x_1 + a_{22}x_2 + b_2u(t)$$

$$y = c_1x_1 + c_2x_2 + d_1u(t)$$

These equations represent a 2nd order system of differential equations, describing the dynamics of the system to be controlled [34]. The first two differential equations model how the system variables change based on the current values. These two equations are coupled. The derivative of $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ both depend on the current values of x_1 and x_2 as well as $u(t)$, a time varying input to the system. If these equations were decoupled, $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ would depend only on x_1 and x_2 [35]. Solving for x_1 and x_2, using the differential equations allow us to solve the y equation, which is the output equation. The output equation determines the system output state based on the internal state variables x_1 and x_2.

It is relatively easy to define a state space form of the system model above. The process converts the system model into a matrix form representation. First, it's necessary to define a state vector. In this case, x_1 and x_2, fully describe the system's internal states. The state vector for the equation incorporates both of these variables as shown below.

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{pmatrix}$$

The derivative of this state vector is shown below.

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{x_1}} \\ \dot{\boldsymbol{x_2}} \end{pmatrix}$$

All of the coefficient terms must be combined into vectors and matrices.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$\boldsymbol{B} = [b_1 b_2]$$

$$\boldsymbol{C} = [c_1 c_2]$$

$$\text{D}=[d_1]$$

Now that these terms are defined, it is possible to rewrite the system model in state space, as shown below [34].

$$\dot{\boldsymbol{x}} = A\,\boldsymbol{x} + \boldsymbol{B}\text{u(t)}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \text{D u(t)}$$

3.2.2 Kalman Filter

The Kalman Filter is a state estimator. A state estimator is an algorithm that estimates the value of state variables of a dynamic system when the actual system state cannot be directly measured. An optimal estimator does this by combining known system dynamics, with current sensor data, and knowledge of the random variability of the system, also known as noise [36].

The dynamics and measurement state space equations are shown below.

$$\boldsymbol{x_{k+1}} = \Phi_k \boldsymbol{x_k} + \Gamma_k \boldsymbol{u_k} + \Upsilon_k \boldsymbol{w_k}$$

$$\boldsymbol{y_k} = H_k \boldsymbol{x_k} + \boldsymbol{v_k}$$

Both of these equations are discrete time next state equations. The name next state alludes to the fact that the next value of the system state, $\boldsymbol{x_{k+1}}$, is calculated based on the previous value, $\boldsymbol{x_k}$. The k subscript indicates the discrete time indice of the state vector. The first equation is the dynamics equation. Φ_k is the discrete time state transformation matrix. This matrix defines the contribution of the previous state on the next state. Γ_k is a matrix that defines the contribution of inputs to the system,$\boldsymbol{u_k}$, to the next state equation. Υ_k is the coefficient of the current process noise vector, $\boldsymbol{w_k}$ [37] [31].The next equation describes how sensor measurements depend on system state. $\boldsymbol{y_k}$ is a vector of sensor measurements that depend on the current state vector $\boldsymbol{x_k}$. H_k is the measurement matrix, also called a sensitivity matrix [38]. The vector $\boldsymbol{v_k}$ is the measurement noise [31].

One of the underlying assumptions of the Kalman filter is that the noise terms are zero-mean Gaussian white noise processes, and uncorrelated. Figure 12 below shows a Gaussian distribution that is centered at zero. In a stochastic variable theory, a random variable tends to vary around its mean such that if all the measurements are recorded on a histogram, the resulting pattern will approximate the Gaussian distribution shown in the figure [39] [40]. The zero mean condition specifies that the mean is at zero. Thus the zero-mean Gaussian distribution means that the histogram of the random noise term must approximate the shape of a Gaussian distribution centered at zero.

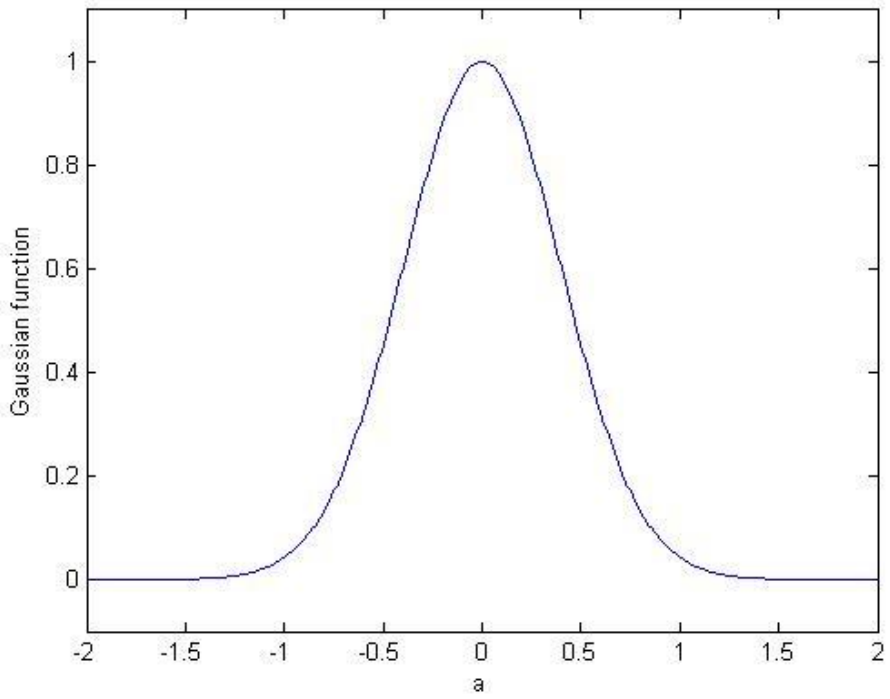

Figure 12: The Gaussian Probability distribution, centered at zero. Source: Wikimedia Commons

The other assumption of the Kalman filter is that the noise terms are uncorrelated. Physically, the elements of the noise vector are the random deviations from the ideal exhibited by a given variable. It is an additive term to the value of the state variable. If the terms are uncorrelated, they are independent of each other. If v_1 in $\boldsymbol{v_k}$ changes by some amount, it is not

expected that v_2 will change at the same time, in the same direction, or by a similar amount. No association exists between the random variables, meaning they are uncorrelated. Mathematically, the test for correlation is covariance. R_k and Q_k are the covariance matrices of the measurement noise $\boldsymbol{v_k}$ and process noise $\boldsymbol{w_k}$ as shown below [31].

$$E\{\boldsymbol{v_k}\boldsymbol{v_j^T}\} = \begin{cases} 0 & k \neq j \\ R_k & k = j \end{cases}$$

$$E\{\boldsymbol{w_k}\boldsymbol{w_j^T}\} = \begin{cases} 0 & k \neq j \\ Q_k & k = j \end{cases}$$

The $E\{\boldsymbol{w_k}\boldsymbol{w_j^T}\}$ term in both equations is called the expectation value of the quantity within the brackets. In stochastic systems, this is another name for mean. It is the weighted average of all possible values that the variable can assume [39]. Note how all elements, except the diagonal, are equal to zero. This situation results from the zero correlation assumption between variables, as previously discussed. An additional assumption of the Kalman filter is that the process noise and the measurement noise are uncorrelated. This condition is shown below [31].

$$E\{\boldsymbol{v_k}\boldsymbol{w_k^T}\} = 0$$

The Kalman filter involves several calculation steps. In the first step, the previous state is propagated in discrete time. This step is also known as the a priori update [41], or as an extrapolation [42]. The equations for this step are shown below [31].

$$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} = \Phi_k \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} + \Gamma_k \boldsymbol{u_k}$$

$$P_{\boldsymbol{k+1}}^{-} = \Phi_k \hat{\boldsymbol{P}}_{\boldsymbol{k}}^{+} \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$$

The first equation propagates the state vector in time. The second equation propagates the error covariance. The error covariance terms describe the error in the system state, which is why it includes the process noise, Q_k [31].

The next step is the calculation of the Kalman gain. Intuitively, the Kalman gain determines the weight that measurements have on the estimate of the system state [31].

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1}$$

Large measurement errors correspond to a large error covariance term, R_k. This large term leads to a small Kalman gain. Subsequently, measurements are not weighted heavily in the subsequent estimates of the state.

The next step in the Kalman filter algorithm is the update step. This step is also known as the a posteriori update [41]**.** In this step, the a priori state estimate is updated based on the measurements. The previously calculated at the Kalman gain is employed here [31].

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \widehat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} + K_k[\widehat{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{h}_{\boldsymbol{k}}(\widehat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-})]$$

$$P_{\boldsymbol{k+1}}^{+} = [I - K_k H_k] P_{\boldsymbol{k+1}}^{-}$$

In the next iteration of the algorithm, the values calculated in the update step become the new $\widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{+}$ and $\widehat{\boldsymbol{P}}_{\boldsymbol{k}}^{+}$, and the calculations are repeated.

3.2.3 Kalman Filter for Attitude Propagation

The Kalman filter for attitude propagation uses the equations outlined above, but with modifications to use quaternions and gyroscope signals. The state vector for the quaternion Kalman filter is shown below.

$$\boldsymbol{x} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

This state vector is obviously the quaternion. With the state vector defined, it is possible to define the state space model shown below.

$$\dot{\boldsymbol{x}} = \frac{1}{2}\Omega\mathbf{x}$$

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}$$

This state space model is based on the differential equation of the quaternion. As in the quaternion differential equation, the ω terms represent the body angular rates.

The next step is to discretize the continuous time model to generate the

discrete-time next state equation that is suitable for implementation on a digital computer. This discretization is carried out by finding the matrix exponential and carrying out a power series expansion [11].

Some preliminary terms are defined first.

$$\Omega^2 = -\left(\omega_x^2 + \omega_y^2 + \omega_z^2\right)I = -\omega^2 I$$

$$\Omega^{2k} = (-1)^k \omega^{2k} I$$

$$\Omega^{2k+1} = (-1)^k \omega^{2k} \Omega$$

Next, the matrix exponential is represented as a power series, and appropriated substitutions are made for the Ω^{2k} and Ω^{2k+1} terms. The resulting summation is then converted from the series representation to a corresponding trigonometric function representation. This trigonometric function is then represented as a matrix

$$exp\left[\frac{1}{2}\Omega T_s\right] = \sum_{n=0}^{\infty} \frac{\left[\frac{1}{2}\Omega T_s\right]^n}{n!}$$

$$= \sum_{k=0}^{\infty} \left[\frac{\left(\frac{1}{2}\Omega T_s\right)^{2k}}{(2k)!} + \frac{\left(\frac{1}{2}\Omega T_s\right)^{2k+1}}{(2k+1)!} \right]$$

$$= \mathbf{1} \sum_{\boldsymbol{k=0}}^{\infty} \frac{(-\mathbf{1})^{\boldsymbol{k}} \left(\frac{\mathbf{1}}{\mathbf{2}}\omega T_s\right)^{\mathbf{2k}}}{(2k)!}$$

$$+ \boldsymbol{\Omega}\omega^{-\mathbf{1}} \sum_{\boldsymbol{k=0}}^{\infty} \frac{\left(\frac{1}{2}\Omega T_s\right)^{2k+1}}{(2k+1)!} = I\cos\left(\frac{1}{2}\omega T_s\right) + \boldsymbol{\Omega}\omega^{-\mathbf{1}}\sin\left(\frac{1}{2}\omega T_s\right)$$

$$= \begin{bmatrix} \cos\left(\frac{1}{2}\omega T_s\right) & \frac{\omega_z}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & -\frac{\omega_y}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & \frac{\omega_x}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) \\ -\frac{\omega_z}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & \cos\left(\frac{1}{2}\omega T_s\right) & \frac{\omega_x}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & \frac{\omega_y}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) \\ \frac{\omega_y}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & -\frac{\omega_x}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & \cos\left(\frac{1}{2}\omega T_s\right) & \frac{\omega_z}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) \\ -\frac{\omega_x}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & -\frac{\omega_y}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & -\frac{\omega_z}{\omega}\sin\left(\frac{1}{2}\omega T_s\right) & \cos\left(\frac{1}{2}\omega T_s\right) \end{bmatrix} = \Phi_k$$

T_s is the sampling period of the attitude determination system. This value is equivalent to the reciprocal of the update frequency of the attitude determination system in Hertz. The end result of the derivation is Φ, the discrete time state transformation matrix.Φ is used to complete the a priori update of the state vector in the Kalman filter [11]. This step in the Kalman filter is also called state propagation.

The equation to complete the a priori update of the covariance requires a different Φ_k, that will be noted by the variable Φ_{Pk}. This matrix is a state transformation matrix, and is derived using a power series in a similar manner to the Φ_k matrix shown above. This equation is shown below [31].

$$\Phi_{Pk} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

$$\Phi_{11} = I_{3x3} - [\widehat{\omega}^x]\frac{\sin(\|\widehat{\omega}\|T_s)}{\|\widehat{\omega}\|^2} + [\widehat{\omega}^x]^2\frac{1-\cos(\|\widehat{\omega}\|T_s)}{\|\widehat{\omega}\|^2}$$

$$\Phi_{12} = [\widehat{\omega}^x]\frac{1-\cos(\|\widehat{\omega}\|T_s)}{\|\widehat{\omega}\|^2} - I_{3x3}T_s - [\widehat{\omega}^x]^2\frac{\|\widehat{\omega}\|T_s - \sin(\|\widehat{\omega}\|T_s)}{\|\widehat{\omega}\|^3}$$

$$\Phi_{21} = 0_{3x3}$$

$$\Phi_{22} = I_{3x3}$$

$$[\widehat{\omega}^x] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The matrix $[\widehat{\omega}^x]$ is the matrix representation of the cross product of the angular momentum vector. The term γ_k that is multiplied by the process noise covariance Q_k is shown below [31].

$$\gamma_k = \begin{bmatrix} -I_{3x3} & 0 \\ 0 & I_{3x3} \end{bmatrix}$$

Next the process noise covariance Q_k is defined below. σ_v and σ_u represent the measurement and process noise root variance respectively.

$$Q_k = \begin{bmatrix} \left(\sigma_v^2 T_s + \frac{1}{3}\sigma_u^2 T_s^3\right) I_{3x3} & \left(\frac{1}{2}\sigma_u^2 T_s^2\right) I_{3x3} \\ \left(\frac{1}{2}\sigma_u^2 T_s^2\right) I_{3x3} & (\sigma_u^2 T_s) I_{3x3} \end{bmatrix}$$

In this form of the Kalman filter, a unique form of the sensitivity matrix is used. This matrix is shown below.

$$H_k = \begin{bmatrix} 0 & -B_3 & -B_2 \\ -B_3 & 0 & -B_1 \\ -B_2 & -B_1 & 0 \end{bmatrix}$$

This sensitivity matrix is the skew symmetric cross product matrix of the magnetic field vector rotated into the body frame. This magnetic field vector is calculated using an onboard model of the earth's magnetic field. At each execution of the Kalman filter, the spacecraft computer calculates the value of the Earth's magnetic field in ECI, based on the current spacecraft position. This ECI vector is then rotated into body frame using the a priori propagated quaternion. This formulation of the sensitivity matrix is particularly useful because its usage dramatically reduces the computation requirements of the algorithm. A larger sensitivity matrix would introduce many more multiplication steps in the algorithm overall, since the sensitivity matrix is used to calculate the Kalman gain and the a posteriori update of the covariance [31].

An additional equation is part of the propagation equation. This equation compensates for the gyro drift measurement. The gyro drift model is assumed to be a first order Markov process. A Markov process exhibits the Markov property, which means that current value of the process depends only on the preceding value and not on the sequences long term history [42]. The gyro drift model is shown below [31].

$$\omega = \tilde{\omega} - \beta - \eta_v$$

$$\dot{\beta} = \eta_u$$

ω is the actual angular rate vector, and $\tilde{\omega}$ is the measured angular rate of the sensor that is provided by the sensor and visible to the user. β is the bias vector, and η_v is the Gaussian white noise process. The term $\dot{\beta}$ describes the rate of change of the bias. Since this is a first order model, it only depends on another zero mean Gaussian white noise process, η_u [31] .

The estimated angular rates equation is given below, along with the differential equation for the angular rates.

$$\widehat{\omega} = \widetilde{\omega} - \hat{\beta}$$

$$\delta\omega = \omega - \widehat{\omega}$$

Substituting the angular rate equation and the process model into the differential equation produces the following.

$$\delta\omega = -(\Delta\beta + \eta_v)$$

$$\Delta\beta = \beta - \hat{\beta}$$

A linearized model for the second order derivative of the quaternion is given below [31].

$$\delta\dot{\varrho} = -[\widehat{\omega}^x]\delta\varrho + \frac{1}{2}\delta\omega$$

$$\delta\dot{q}_4 = 0$$

Substituting the differential equation for the angular rates into the above equation produces the following.

$$\delta\dot{\varrho} = -[\widehat{\omega}^x]\delta\varrho - \frac{1}{2}(\Delta\beta + \eta_v)$$

Next, a substitution is made for $\delta\varrho = \delta\frac{\alpha}{2}$ with α being a vector representing the Euler angles of roll, pitch, and yaw. The underlying assumption for this substitution is the small angle approximation.

$$\delta\dot{\alpha} = -[\widehat{\omega}^x]\delta\alpha - \frac{1}{2}(\Delta\beta + \eta_v)$$

This representation of the quaternion error lends itself to be rewritten as the Kalman error in the state vector, since the state vector is a quaternion. This Kalman error model is shown below [31].

$$\delta\dot{\alpha} = \Delta\dot{\tilde{x}} = F\Delta\tilde{x} + G\boldsymbol{w}$$

$$F = \begin{bmatrix} -[\widehat{\omega}^x] & -I_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix}$$

$$G = \begin{bmatrix} -I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix}$$

$$\boldsymbol{\Delta\tilde{x}}_{\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{\delta\alpha_k} \\ \boldsymbol{\Delta\beta_k} \end{bmatrix}$$

The error term $\Delta\tilde{x}$ will prove useful in conducting the a posteriori update. In

this step, measurements are used to update the a priori propagated state, based on the Kalman gain. Another way of looking at this step is as a compensation step for the error in the propagated state, based on new knowledge of the system state from the measurements. In a mathematical format, the a posteriori update step can be written as

$$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} + \boldsymbol{\Delta\tilde{x}}$$

Or,

$$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} + K_k[\hat{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{h}_{\boldsymbol{k}}(\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-})]$$

Thus,

$$\boldsymbol{\Delta\tilde{x}} = K_k[\hat{\boldsymbol{y}}_{\boldsymbol{k}} - \boldsymbol{h}_{\boldsymbol{k}}(\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-})]$$

$\boldsymbol{\Delta\tilde{x}}$ must be calculated during the a posteriori update phase. This provides values for $\boldsymbol{\delta\alpha}$ and $\boldsymbol{\Delta\beta}$ that are used in the next phase. After this is done, it is possible to calculate a new value for the gyro bias. This gyro bias is subsequently used to compensate for gyro drift and calculate a new angular rate based on the gyro rate measurements. These updated equations are shown below [31].

$$\hat{\beta}_k^{+} = \hat{\beta}_k^{-} + \Delta\beta_k$$

$$\widehat{\omega} = \tilde{\omega} - \hat{\beta}_k^{+}$$

$\hat{\beta}_k^{-}$ is the previous bias value, $\hat{\beta}_k^{+}$ is the updated bias term, $\tilde{\omega}$ is the gyro measurement, and $\widehat{\omega}$ is the angular rate estimate after compensation using the bias term. This estimate of $\widehat{\omega}$ is used on the next iteration of the algorithm to propagate the quaternion using the Φ_k state transition matrix.

The a posteriori update of the state vector is shown below [31].

$$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \hat{q}_{k+1}^{+} = \hat{q}_{k+1}^{-} + \frac{1}{2}\Xi_k\,\delta\alpha$$

The term Ξ is a matrix related to the quaternion defined below [43] .

$$\Xi_k = \begin{bmatrix} e_k^x + q_4 I_{3x3} \\ -e_k^T \end{bmatrix}$$

$$e_k = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

$$e_k^x = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

e_k is a vector consisting of the vector part of the quaternion. e_k^x is the skew symmetric cross product matrix [41].

4 Attitude Drift and Compensation

This chapter will describe sources of attitude drift and methods to compensate for it. This chapter concludes with a description of an attitude determination system that utilizes the stellar gyro system to compensate for drift.

4.1 Attitude Drift Sources

This section will describe the sources of attitude drift. Sources of attitude drift include the analog to digital conversion process that includes sampling and quantization error. Other sources include random sensor noise described by the sensor Allan variance. A final source of attitude drift error is limits in the gyro resolution.

4.1.1 Analog to Digital Conversion Process

A significant source of error in the gyroscope measurements results from the analog to digital (A to D) conversion process. The A to D process is common in electronics. It involves converting a real world continuous signal, otherwise known as an analog signal, into a digital discrete signal. Figure 13 illustrates the A to D process graphically. The grey line is the underlying analog signal, and the red stair step pattern represents the resulting digital signal, produced in the A to D conversion process. The A to D process consists of two distinct activities, discretization and quantization. Discretization results from limitations in the sensor electronics' ability to switch fast enough to take a snapshot of a rapidly changing analog signal. This process is often known as a zero order hold. The analog signal is sampled, or read instantaneously, by the digital electronics at equally spaced intervals in time. This fact is apparent from the stair step pattern visible in red digital signal in Figure 13. If a signal is changing much faster than it is being sampled, any fluctuations between the vertical lines are not detectable. Quantization is another component of the A to D process and it is caused by the limited number of digital bits available to represent the amplitude of the analog signal. A real analog signal has infinite resolution that is not possible to represent with a digital signal, due to the limited number of bits. These bits must be used to

represent the full sensor range. The quantization means fixed levels for the sensor output. Values falling between these fixed quantization levels are rounded up or rounded down to the nearest quantized number. This rounding produces a type of error known as quantization error. A method to compensate for this error is through dithering. Dithering involves injecting small amplitude analog white noise into a signal, with amplitude of approximately a third of the smallest quantization value. This noise injection will cause the sampled value of the signal to toggle between the two adjacent quantization levels. When these samples are averaged using a smoother or moving average filter, the resulting number will be much closer to the analog signal [48].

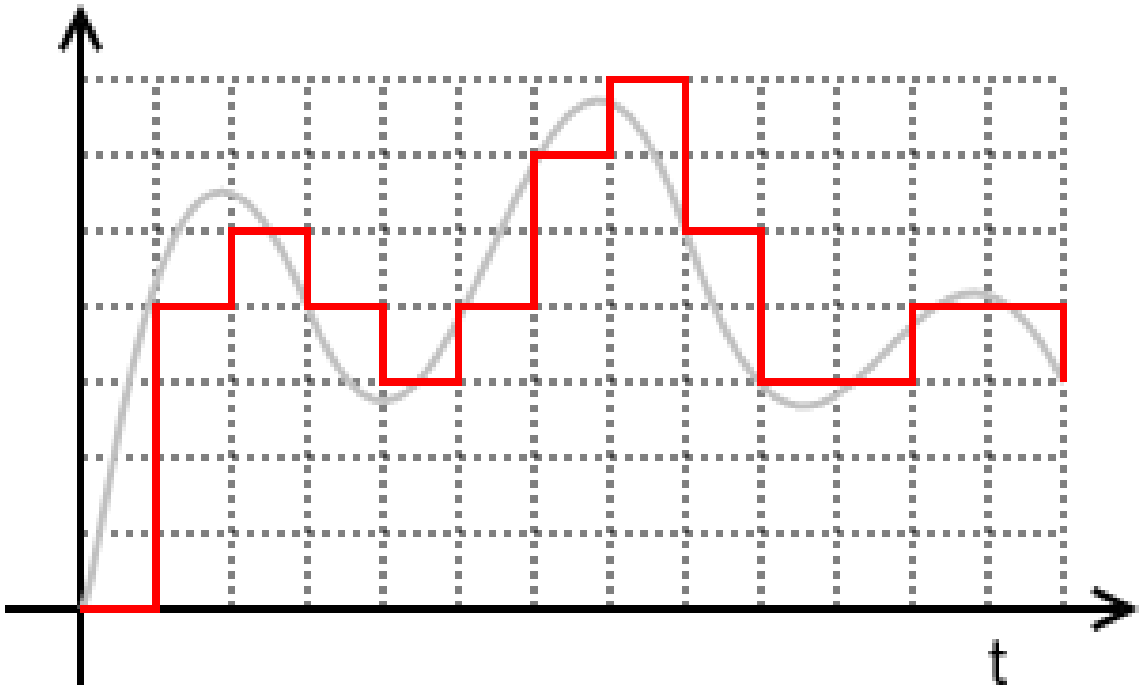

Figure 13: The Analog to Digital Conversion Process showing a zero-order hold process, with sampling and quantization. Source: Wikimedia Commons

4.1.2 Allan Variance

Gyroscopes are inherently noisy instruments. Error in gyro measurements is not easily described by a single number or specification in a data sheet. Gyro error specifications are often given in terms of an Allan Variance plot, shown in

Figure 14 below. An Allan Variance plot expresses the error in a gyro measurement in terms of integration period. This integration period represents an average of sequential measurements from the gyro. The y-axis of the Allan Variance quantifies the actual error in the measurement. In Figure 14, this value is referred to as the Root Allan Deviation, in degrees per second. The Root Allan Deviation specifies the 1 Sigma Standard deviation of the measurement from the nominal measured value [48].

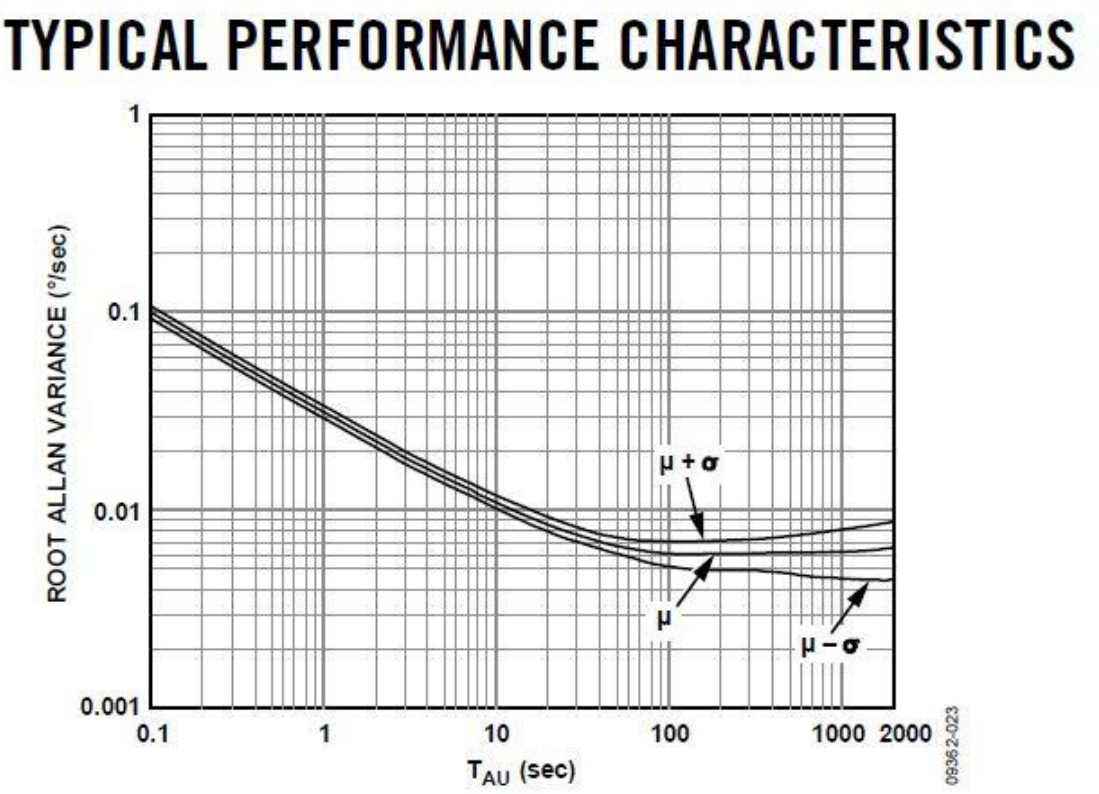

Figure 14: Allan Deviation of Gyroscope Source: Analog Devices ADIS16334 IMU Datasheet

The underlying assumption of the Allan Variance is that by averaging sequential gyro measurements, an error minimum is reached. At shorter integration periods, the measurement error is high, reaching a maximum error at the minimum integration time. Error in this region is referred to as Angular Random Walk (ARW). The formula for the measurement variance due to ARW is shown below [49].

$$\sigma^2 = \frac{N^2}{\tau}$$

In this equation, σ^2 represents the measurement variance, which is square of the root Allan deviation. N^2 is a squared constant, and τ is the integration period. Through inspection of the ARW equation, it is clear that the ARW noise is essentially high frequency noise. Since it is high frequency noise, it can be effectively eliminated through data smoothing techniques, such as a moving average filter. This is consistent with the equation above, since the noise decreases with the integration period.

Figure 14 shows that increasing the integration period greatly reduces the noise, but it does not fully eliminate it. Another noise source is present in the statistical model of the gyro. It is referred to as Bias Instability. The equation for Bias Instability is shown below [49].

$$\sigma^2 = \frac{2B^2}{\pi}\left(ln(2) - \frac{\sin^3 x}{2x^2}(\sin x + 4x\cos x) + C_i(2x) - C_i(4x)\right)$$

$$x = 2\pi f_0\,\tau$$

Here, σ^2 represents the measurement variance, B is the bias instability coefficient, C_i is the cosine integral, f_0 is the sampling frequency of the gyroscope, and τ represents the integration period. This equation is rather complicated, but fortunately it is possible to apply a simplification that will reduce the number of terms in the equation. It is reasonable to assume that τ, the integration period, is much larger that the sampling period of the gyro $\frac{1}{f_0}$. When this assumption is made, it's possible to reduce the Bias Instability error equation to the following [49].

$$\sigma^2 = \left(\frac{B}{0.6648}\right)^2$$

In this equation, B is the previously discussed Bias Instability. This simplification also eliminates the dependence on the integration time, τ. This means that Bias Instability error becomes an error signal with constant variance when τ is much larger than the sampling period of the gyro, $\frac{1}{f_0}$. The validity of this assumption

appears in the Allan Variance plot. In the region where Bias Instability is dominant, the variance curve is flat, representing the constant Bias Instability error. Thus, the bias instability is modeled as a white noise process with constant variance, assuming a τ that is much greater than the sampling period of the sensor.

4.1.3 Gyro Resolution Limits

The quantization inherent in the A to D process leads to another issue with MEMS gyros beyond quantization error. Quantization also leads to resolution limits. Resolution limits mean that only angular rates above a certain minimum value can be measured, otherwise the output appears to be zero. Resolution limits are essentially an effect of quantization since, during quantization, values below half the lowest quantization level are rounded to zero. In practice, this means that angular rates that are less than the number can be represented by the least significant bit of the sensor output fall below the resolution of the sensor and are not measureable. Figure 15 shows the angular rates of the body axes of a 2U CubeSat subject to environmental torques in low earth orbit. For reference, the resolution limits of the Analog Devices ADIS16334 IMU [50] are shown as well. The gyro in this IMU represent the angular rates in a 24 bit, two's complement number. The bold horizontal lines represent the minimum angular rate, (+/-) 0.015 Degrees/Second, that the gyro can display using the least significant bit (LSB).

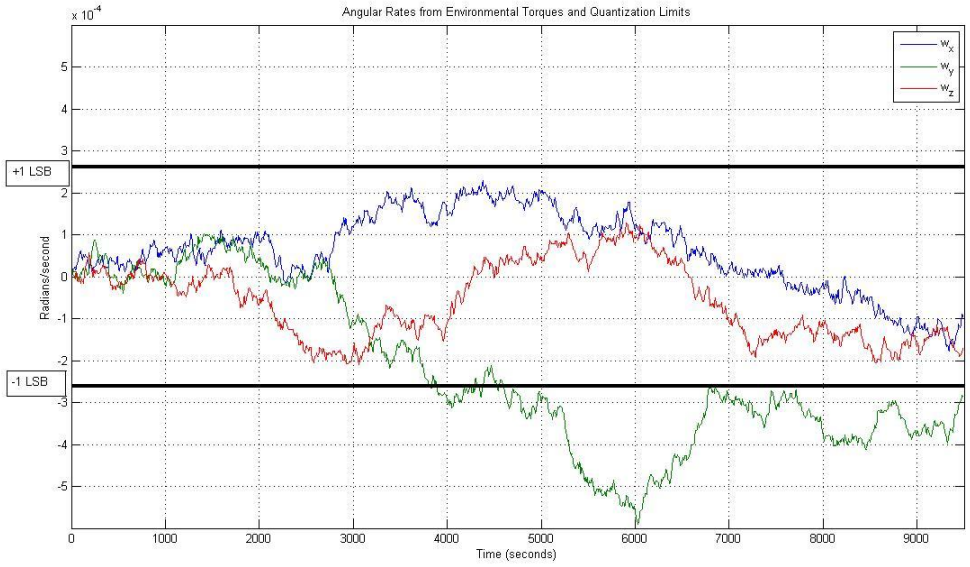

Figure 15: Angular Rates of a CubeSat Subject to Environmental Torques and the +/1 LSB Resolution Limits of the ADIS16334 Gyro

As discussed in Chapter 2, a satellite in Low Earth Orbit is subject to a variety of disturbance torques from the space environment. The magnitude of many of these disturbance torques, such as solar pressure torque and aerodynamic torque, are proportional to the surface area of the satellite. For a CubeSat with its small external surface area, these disturbance torques are small. For reference, a 2U CubeSat has a surface area of 0.1 meters squared. Using the equations given in Chapter 2, the magnitudes of the torques are on the order of 1x10-6 Newton-Meters. These environmental torques generate the angular rates shown in Figure 15. These angular rates fall below the least significant bit resolution of the sensor for much of the orbit. The conclusion here is that the practical limits in the resolution of even high end rate sensors, such as the ADIS16638, prevent the sensors from measuring the angular rates experienced on orbit. This limitation handicaps a Kalman Filter from accurately estimating the attitude of the satellite, since it is essentially integrating zero.

Some method of compensating for this lack of resolution should be attempted. One possibility is using analog rate sensors and an op-amp circuit

with appropriate low pass filtering to amplify the low angular rates. This amplified signal can then be sampled by an A to D converter. The op-amp circuit should have a gain selected that will amplify the voltage corresponding to the maximum angular rate up to the maximum voltage allowed by the A to D converter. In this way, it's possible to take advantage of the full dynamic range of the sensors [52] . This approach may not be ideal due to the additional complexity of the analog circuitry involved.

Another approach is to use oversampling and noise shaping sigma delta modulation techniques to enhance the resolution of the A to D conversion process. This approach involves sampling at above the Nyquist frequency, and employing filtering techniques to reduce the quantization noise to produce much higher resolutions. Oversampling a signal by a factor of 16 can produce an extra 2 bits of resolution of the A to D [53].

4.2 Drift Compensation through Stellar Gyro Updates

The stellar gyro provides a convenient solution to the attitude drift problem. The stellar gyro, after being provided with an absolute attitude fix by QUEST, can provide a drift free attitude estimate and update to the Kalman filter by measuring the attitude changes by comparing the changes in position of stars in the field of view. This allows a drift free propagation of the attitude. In terms of implementation, the stellar gyro requires a small, externally mounted CMOS detector to image the star field. If two of these CMOS sensors are oriented with a 180 degree angular offset, it's possible to ensure that a star field image is always visible, regardless of the presence of the earth in the field of view of an individual sensor. Figure 16 on the following page shows the updated system diagram with the stellar gyro installed. Note that the stellar gyro provides an attitude update directly to the Kalman filter, and that the MEMS gyros are still part of the system. MEMs gyros still must be present to measure rapid changes in attitude.

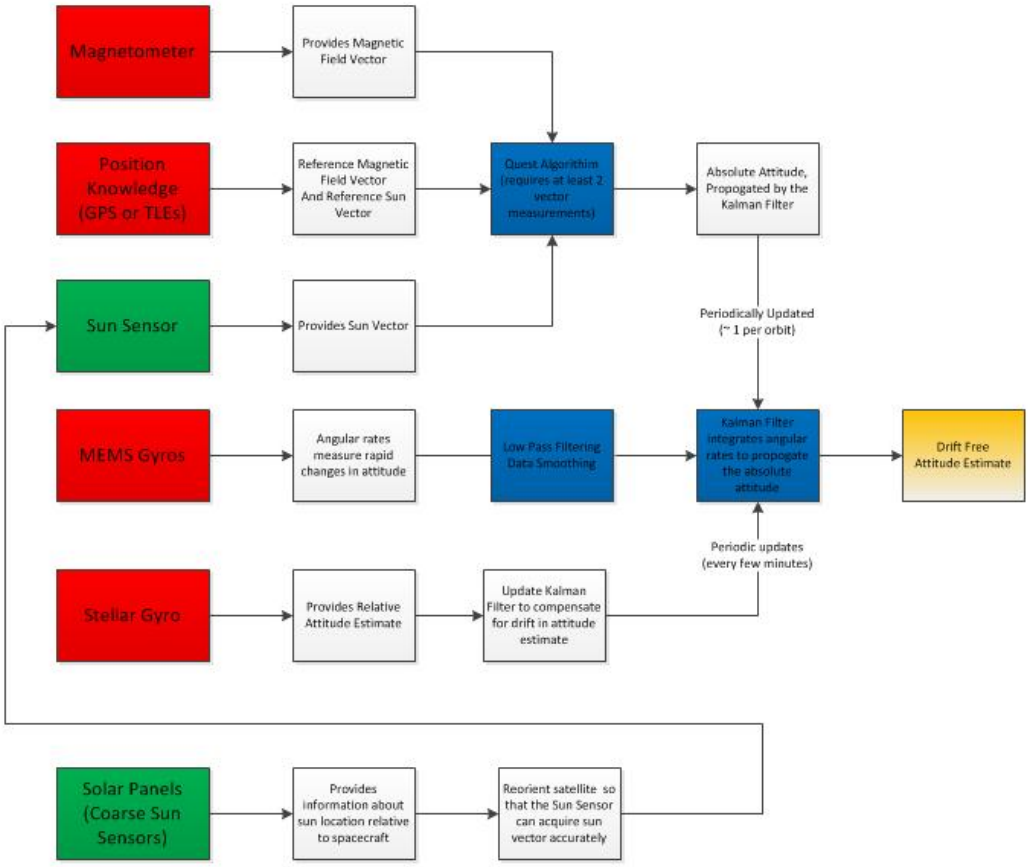

Figure 16: The stellar gyro providing the Kalman filter with periodic drift free updates.

5 Simulink Modeling of the Attitude Determination System

This section will describe the Simulink models developed to model a CubeSat attitude determination system. This Simulink model includes models of common attitude determination sensors. It also includes dynamics models to simulate the effects of torques on the CubeSat. Finally, the Simulink model includes models of the unified stellar gyro and Kalman filter system.

5.1 High Level System Model

Figure 17 shows the highest level of the Simulink model. The blocks in the Simulink model represent different subsystems. The lines between the different subsystems represent different signals moving between the subsystems. The block in the upper middle is the 6 Degree of Freedom (DoF) dynamics model. It receives torque as an input and outputs velocity, position, attitude and angular rates. The magnetic field model also appears at this level, below and to the right of the 6 DoF dynamics model. It receives position as an input, indicated by the Green 'P,' and outputs a magnetic field in Teslas. On the upper left of the 6 DoF is a gravity model. This model takes position as an input, and outputs a gravitational force, which the 6 DoF model uses to model the effect of gravity gradient torques on the CubeSat. To the right and below the 6 DoF model is the Sun Vector calculation block. This block determines the ECI sun vector, using a position input signal and a calendar date. This model utilizes a VSOP 87 model to calculate the sun vector [44]. At the lower left of the Simulink model is the block containing the attitude determination sensor models. The Kalman Filter and attitude determination algorithms are contained within the blue subsystem. The complex series of subsystems and signals on the lower right are used to calculate the error in the attitude estimate from the Kalman filter.

Error is determined by calculating the difference between the attitude estimate provided by the Kalman filter and the truth attitude. In simulation terms, the truth attitude is the output from the 6 DoF model, which is considered reality, or truth, in the simulation.

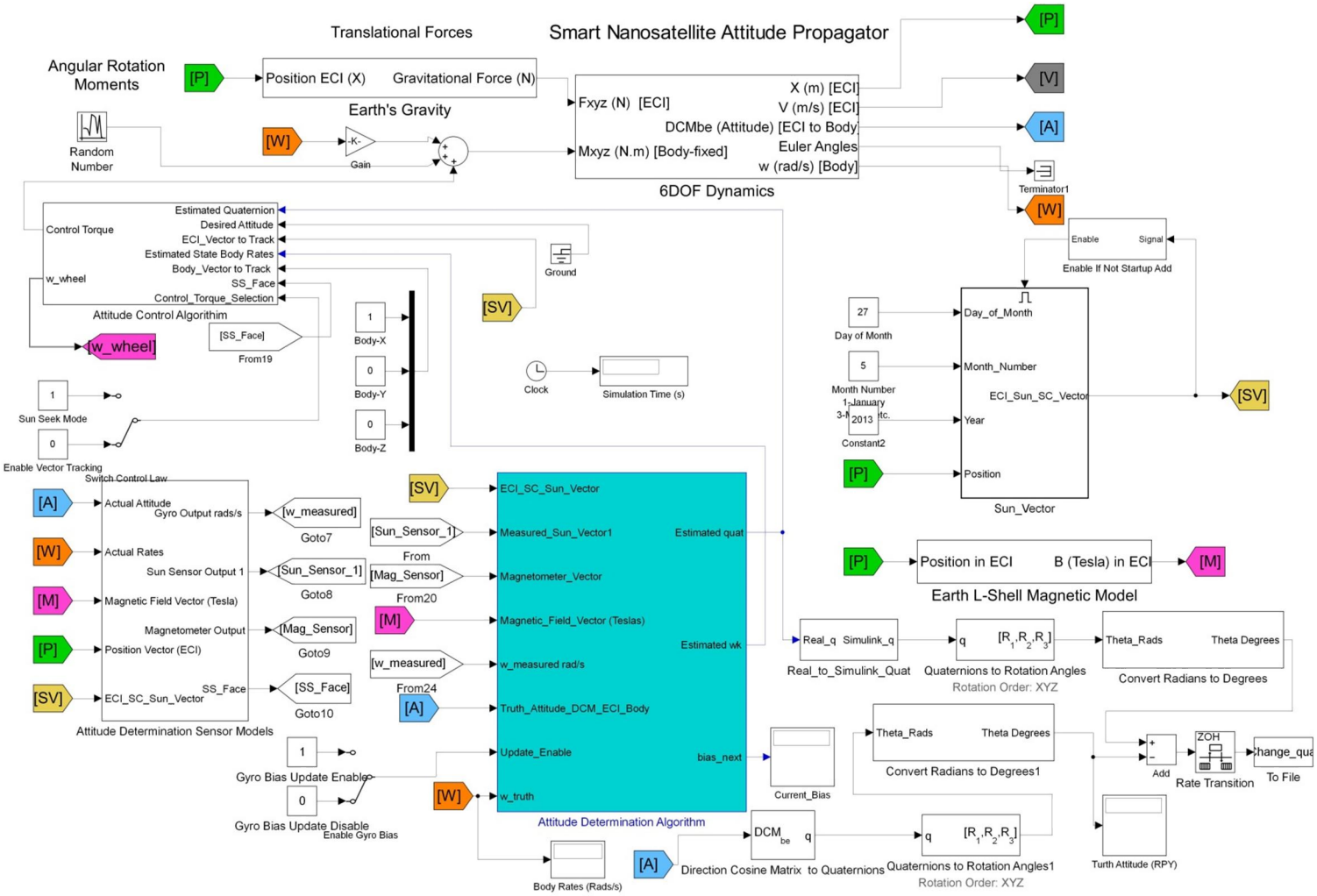

Figure 17: High Level Simulink Model

5.2 Simulink Sensor Models

Sensor models were an important part of the Simulink attitude determination system. Models were created for MEMS gyros, magnetometers, sun sensors, and solar panel based coarse sun sensors. The Simulink models of the sensors are shown in Figure 18. These sensors simulate hardware by modeling quantization and the analog to digital conversion process, and additive system noise. For the optical type sensors, such as sun sensors and star trackers, a field of view feature checks to see if a reference object appears in the sensor field of view, and blocks the output if an object is not visible. These sensor models also include masked variables, which allow configuration of various system parameters through a graphical user interface (GUI) based popup menu. The GUI based menu for the magnetometer is shown in Figure 19. Noise characteristics, field of view, quantization levels, and sampling rates can all be specified through this GUI menu. This menu also allows the user to specify the orientation of the sensor in the spacecraft body frame using Euler angles.

Figure 20 shows the internal framework of the magnetometer model. The first step in this model is rotation of the ECI magnetic field vector to the body frame. This rotation is completed using the DCM matrix that is input into this block through the actual attitude port. Next this body frame vector must be rotated into the sensor frame. The Roll, Pitch, and Yaw Euler angles, specified by the user in the mask, are represented as Simulink constants R, P, and Y within the Simulink block. The black vertical bar that these constants connect to is a Simulink mux block that combines these constants into a vector. In the lower left, these Euler Angles are fed into a Simulink block called Rotation Angles to DCM. This block generates a DCM that is used to rotate the sensor vector measurement from the body frame to the sensor frame. The important Euler angle rotation order must be specified for this block, which is shown below the block as Z-X-Y, corresponding to a Yaw-Roll-Pitch, or 3-1-2 rotation order. This rotation order must be consistent across the different Simulink blocks. After rotating into the sensor frame, the signal passes into a summing block, where Gaussian white

noise is added. The variance of this noise is also specified in the mask by the user. The signal then passes into the portion of the block simulating the A to D conversion process. The rate transition block converts the continuous time signal to a discrete representation, with frequency specified in the mask. The quantizer similarly converts this discrete signal to a digital signal with quantization levels specified in the mask. For simulation purposes, the mask parameters allow the sensor models to be configured to have the same characteristics of actual hardware. The magnetometer hardware simulated in this thesis was a surface mount Honeywell HMC512 magnetometer.

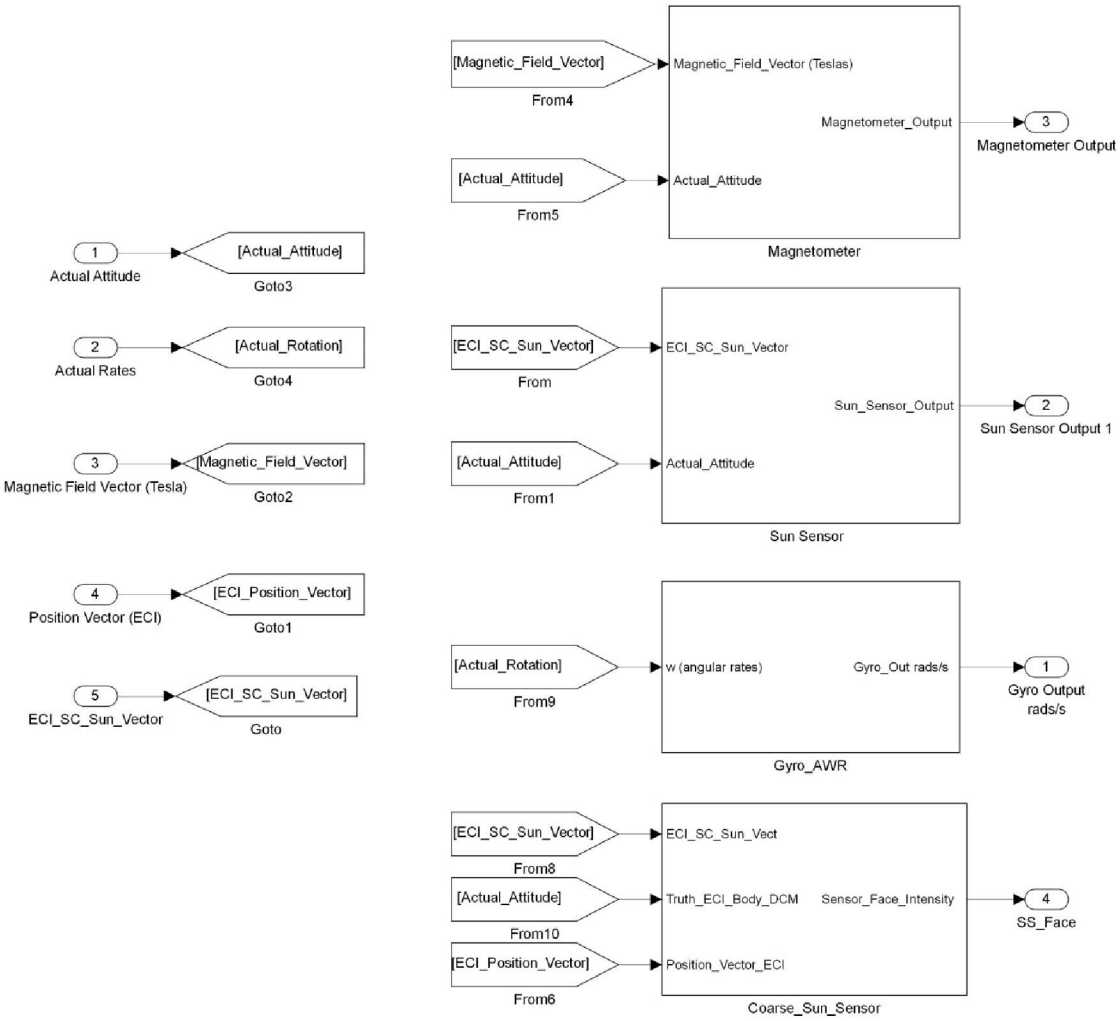

Figure 18: Simulink Sensor Models Showing the GUI Menu for configuring sensor parameters.

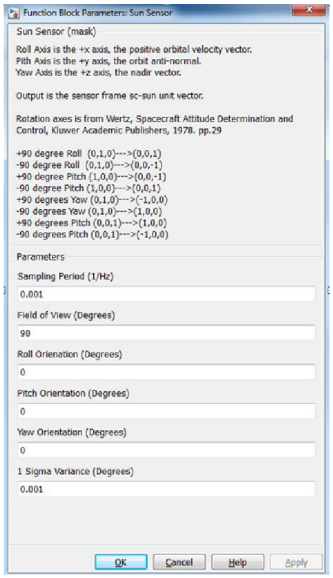

Figure 19: The Masked Variable Context Menu for the Magnetometer

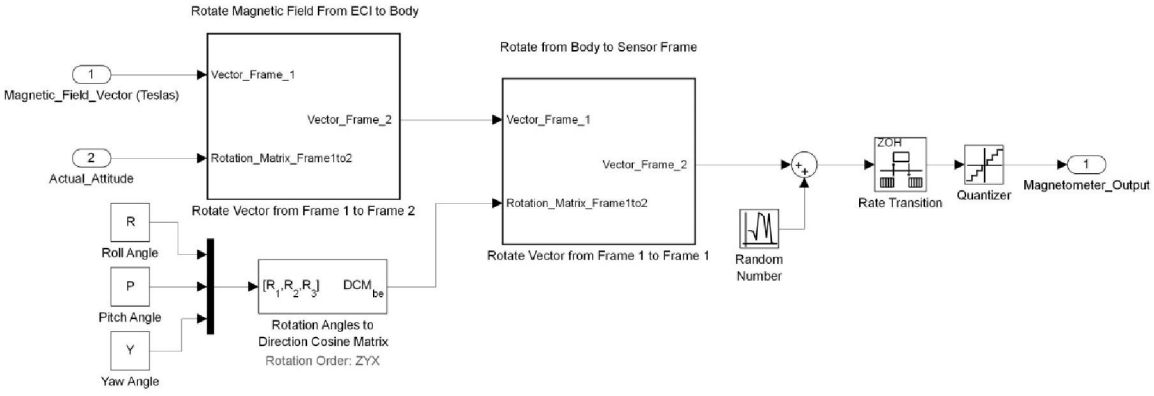

Figure 20: Magnetometer Model

Figure 21 shows the internals of the Simulink sun sensor model. This sensor model contains the same ECI-Body-Sensor Frame rotation as the magnetometer, as well as the A to D modeling process and noise injection as described previously in the description of the magnetometer model. In order to most accurately simulate a sun sensor, this model also includes functionality to determine if the sun is in the sensor field of view. To complete this logical check, the angle between the sun vector in the sensor frame and the sensor bore sight are compared. If this angle is less than the half cone angle of the sun sensor field of view (FOV), then the sun is visible to the sensor, and the sun vector is passed out of the block. If it is not visible, the sensor passes a vector of NaN values. The passing of the NaN or the actual vector occurs at the Switch 1 block, while the logical checking occurs upstream, using the Angle Between two Vectors blocks and a series of logical blocks. This sensor model is based on an SSBV Fine Sun sensor model, a highly accurate small form factor sun sensor appropriate for CubeSats and larger small satellites.

Figure 22 shows a coarse sun sensor model. This model simulates the use of solar panels as coarse sun sensors, generating an output that indicates which faces of the CubeSat are illuminated by the sun. The specific output is a 6 element vector, with 1 element for each face of the CubeSat. If a face sees the sun, the corresponding element in vector is set to 1, otherwise it is 0. This model works in a similar manner to the fine sun sensor model above. The angle between the sun vector and the normal vectors for each face are compared to a fixed field of view. If the angle is less than the half cone angle of the field of view, then the sun can illuminate the face of the CubeSat, and the vector element is set to 1. This comparison is completed for all faces of the CubeSat. The individual outputs of these logical checks are then combined using muxes to generate the output vector.

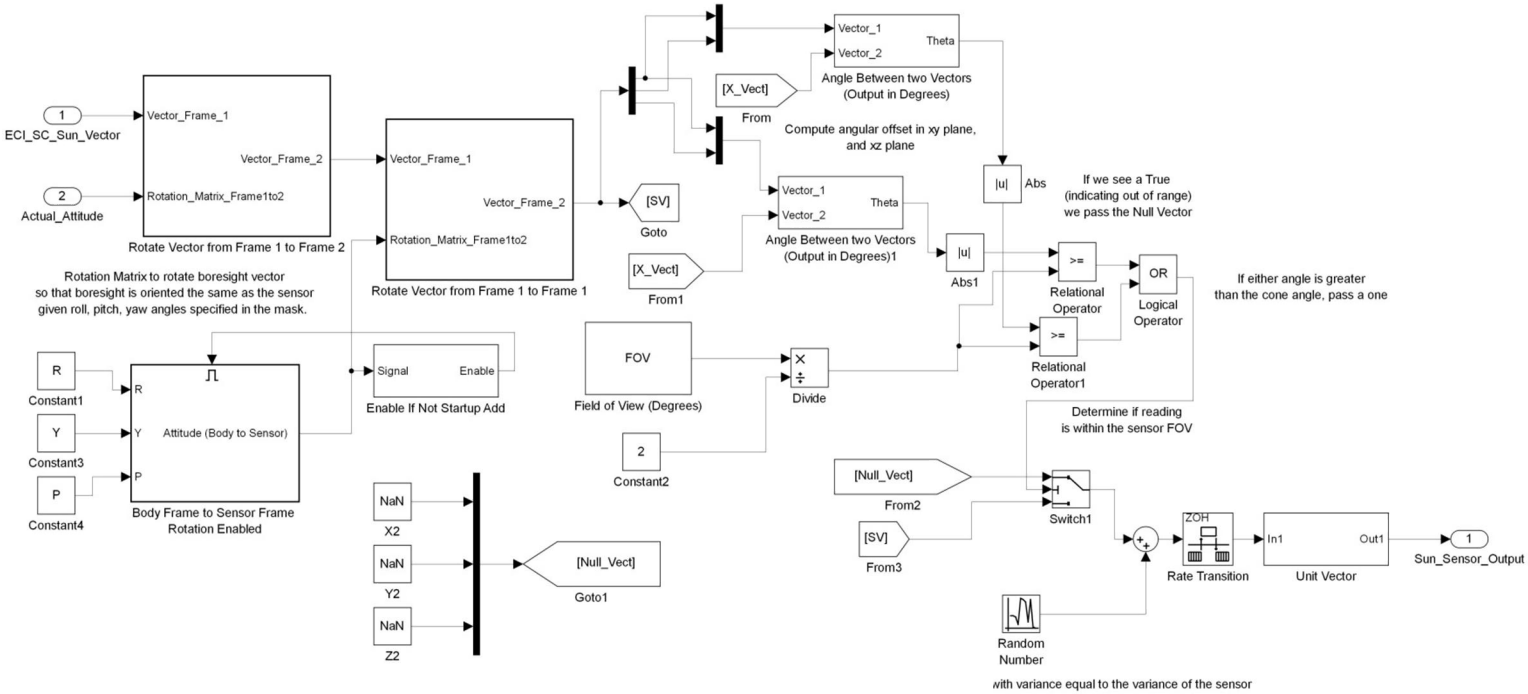

Figure 21: Sun Sensor

Internal Model

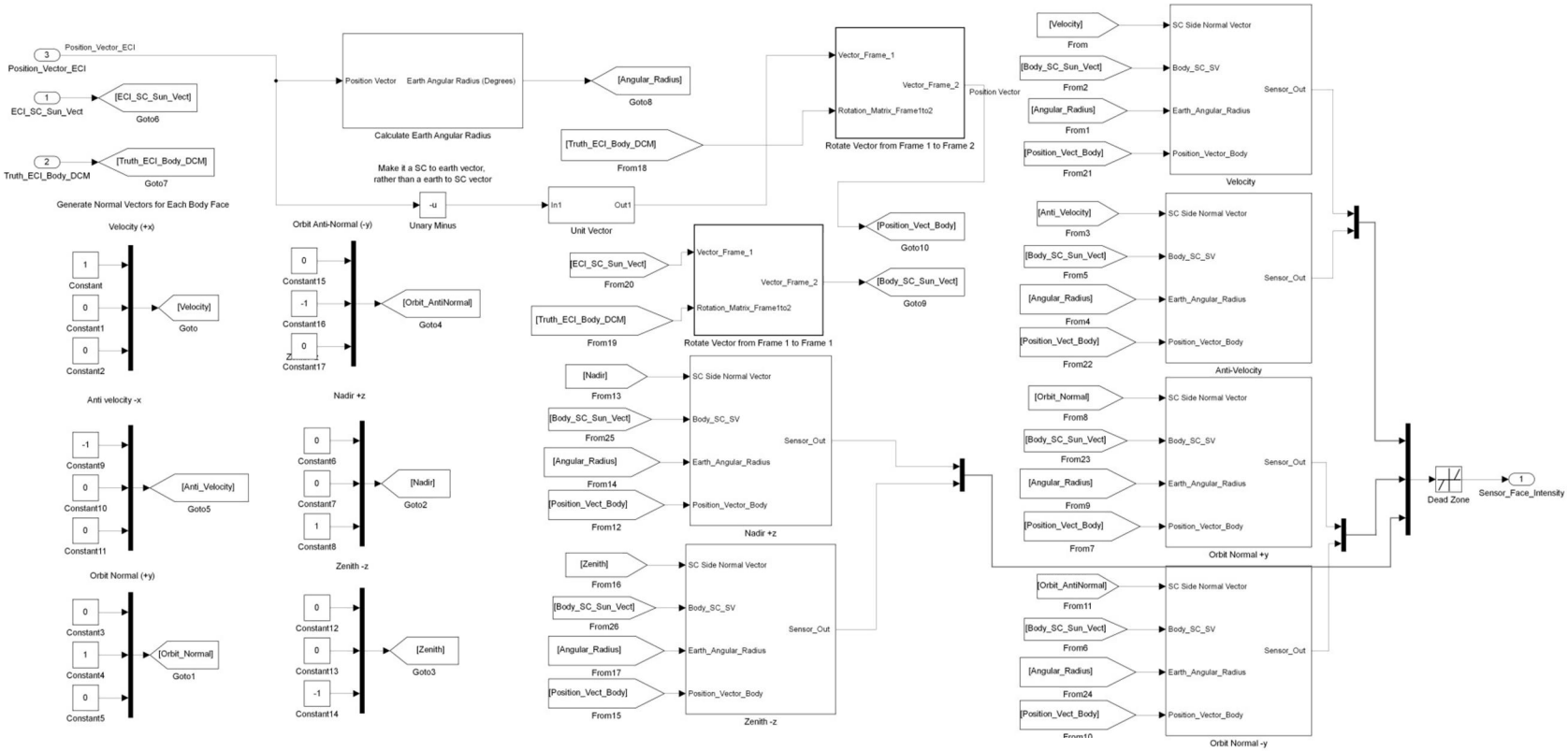

Figure 22: Coarse Sun Sensor Internal Model

5.3 Quaternion Extended Kalman Filter Subsystem

Figure 23 below shows the Quaternion Extended Kalman Filter subsystem. The actual code to implement the Extended Kalman Filter is on the far right. This block provides an output of estimated angular rates, along with the estimated quaternion. This subsystem also contains several other subsystems to support the Kalman Filter. The block on the top left is the Check_NaN block. This block checks for valid sensor inputs, indicated by the actual numbers and not the NaN data type that MATLAB uses to represent undefined data types. These inputs are not subsequently used in later calculations. The block to the right of the Check_NaN measurements is the Rotate Measurements block. This block rotates sensor readings from the sensor frame into the body frame. This step is required by the Kalman filter and the absolute attitude determination algorithms. Finally, the subsystem in the middle-bottom is the absolute attitude determination block, containing the QUEST algorithm and the stellar gyro. This subsystem is connected directly to the Kalman Filter, and provides a method of updating the quaternion that the Kalman filter is propagating.

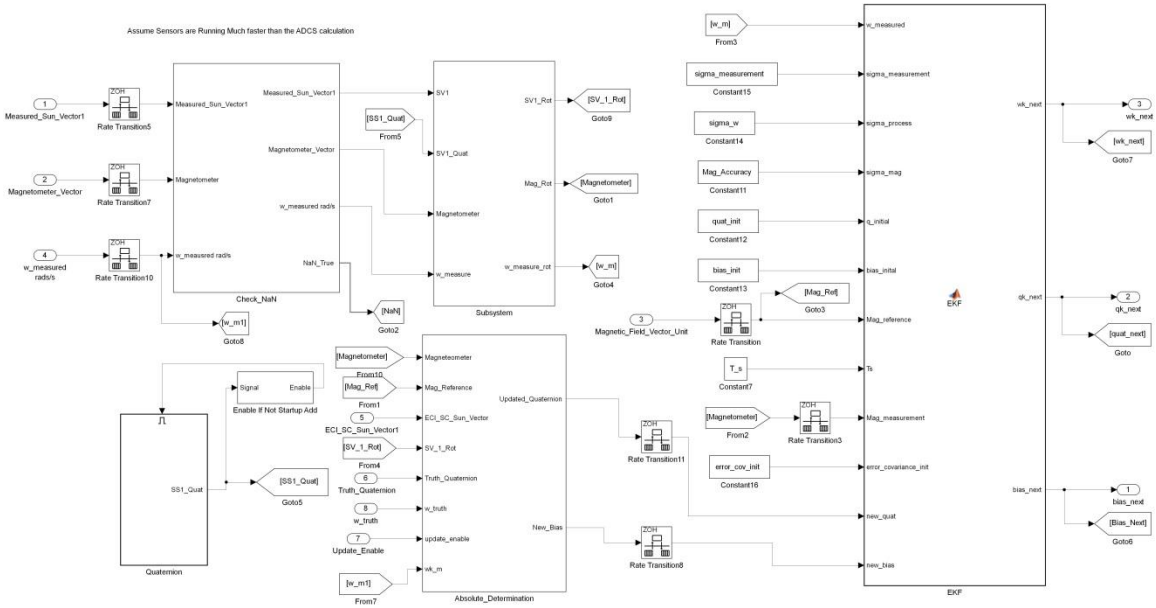

Figure 23: Quaternion EKF Subsystem

5.4 Absolute Attitude Determination Block

The absolute attitude determination block is shown below in Figure 24. It contains the QUEST algorithm and the stellar gyro and supporting subsystems. On the upper left, vector measurements and reference vectors are passed into the Form Vectors subsystem. This block combines the reference vectors and the measurement vectors into two different matrices. These matrices are arranged such that each vector measurement and reference vector pair has the same column number in each matrix. This format is appropriate for the QUEST algorithm that subsequently uses these two matrices to determine the absolute attitude. The quaternion determined using QUEST gets passed into the Stellar Gyro system block. The Stellar Gyro then uses this as a starting point to update the quaternion. A system clock input is also passed into the stellar gyro block, and the stellar gyro uses this to determine when it should update the quaternion.

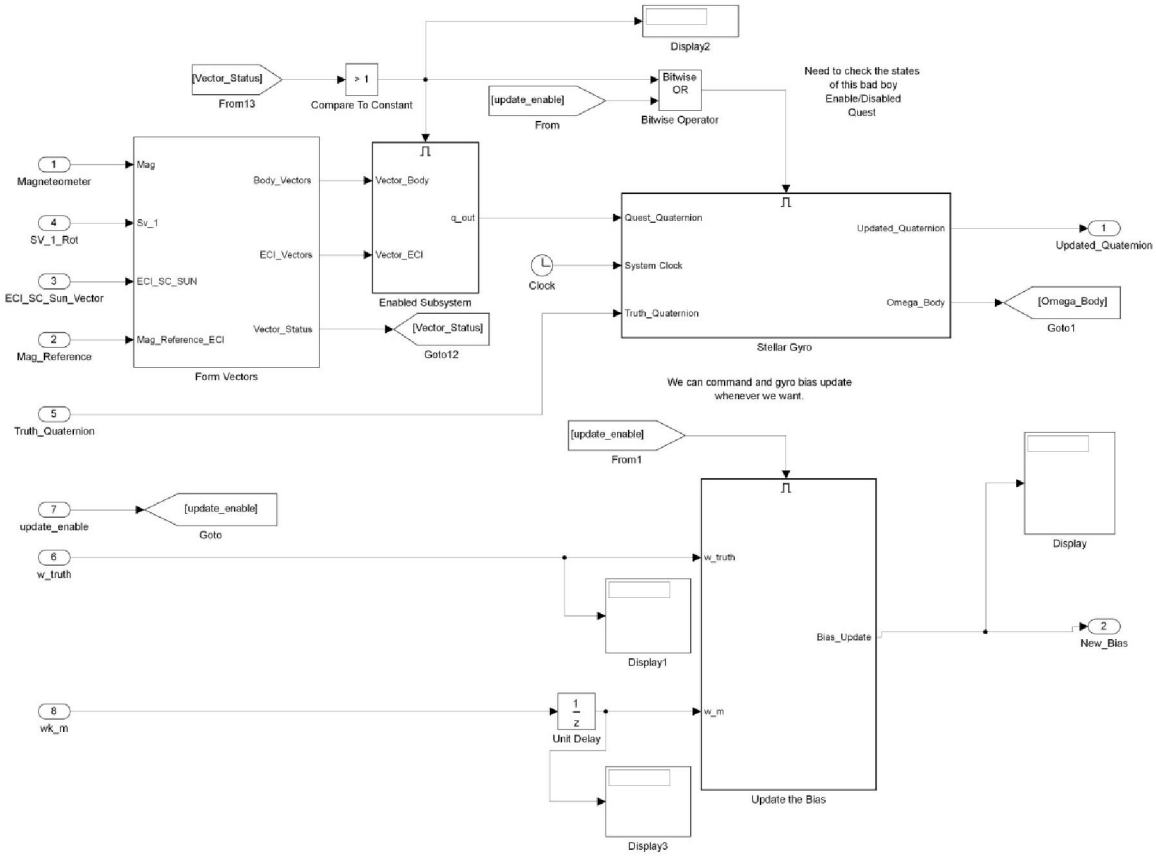

Figure 24: Absolute Attitude Determination Subsystem

6 Results

This chapter will demonstrate drift free attitude estimation through stellar gyro updates. Two different situations are used to demonstrate Kalman Filter. The first is a CubeSat with an initial angular rate of 0 degrees per second in each body axis, and accelerated by the environmental torques. The second situation is a CubeSat with 0.1 degrees per second roll rate in each body axis at the start of the simulation. The estimate of the attitude for both scenarios is compared to the truth attitude that is the output of the 6 Degree of Freedom dynamics block. Errors are calculated between the estimated attitude and the truth attitude.

6.1 Drift Compensation with Low Roll Rate

The next case demonstrates the effectiveness of the attitude determination system in a situation where the satellite is slowly rolling, subject to only environmental torques. This situation was previously discussed in the context of gyro resolution limits, where the slow roll rates imparted by environmental torques are not measureable by the MEMS gyros. Figure 25 below shows the truth attitude. This data is taken directly from the output of the 6 Degree of Freedom dynamics block. A significant characteristic of this plot are the discontinuities present. These discontinuities are visible in the roll angle at 8000 seconds and 1100 seconds, and pitch and yaw at 8000 seconds. These discontinuities are due to singularities in the Euler angle representation of the attitude that occur at 180 and -180 degrees in the roll angle, and -90 and 90 degrees in the pitch and yaw angles. The output of the Kalman filter is a quaternion that is subsequently converted to an Euler angle representation. The quaternion representation is useful because it does not experience these singularities. Thus, these discontinuities do not represent an issue with the Kalman Filter itself, but rather with the way the data is displayed. Figure 26 shows the attitude estimate that is the output of the Kalman filter. Inspection of the two figures reveals that the estimate and the truth attitude closely correspond. Figure 27 shows the estimate error. This error is calculated by subtracting the Euler angle representation of the estimate from the corresponding

truth attitude. The error in the estimate is less than 2 degrees over the course of the simulation. The large increases in error that occur around 8000 seconds and 1000 seconds are due to discontinuities in the Euler angle representation of the attitude, as previously discussed. Stellar gyro updates were completed at 60 second intervals.

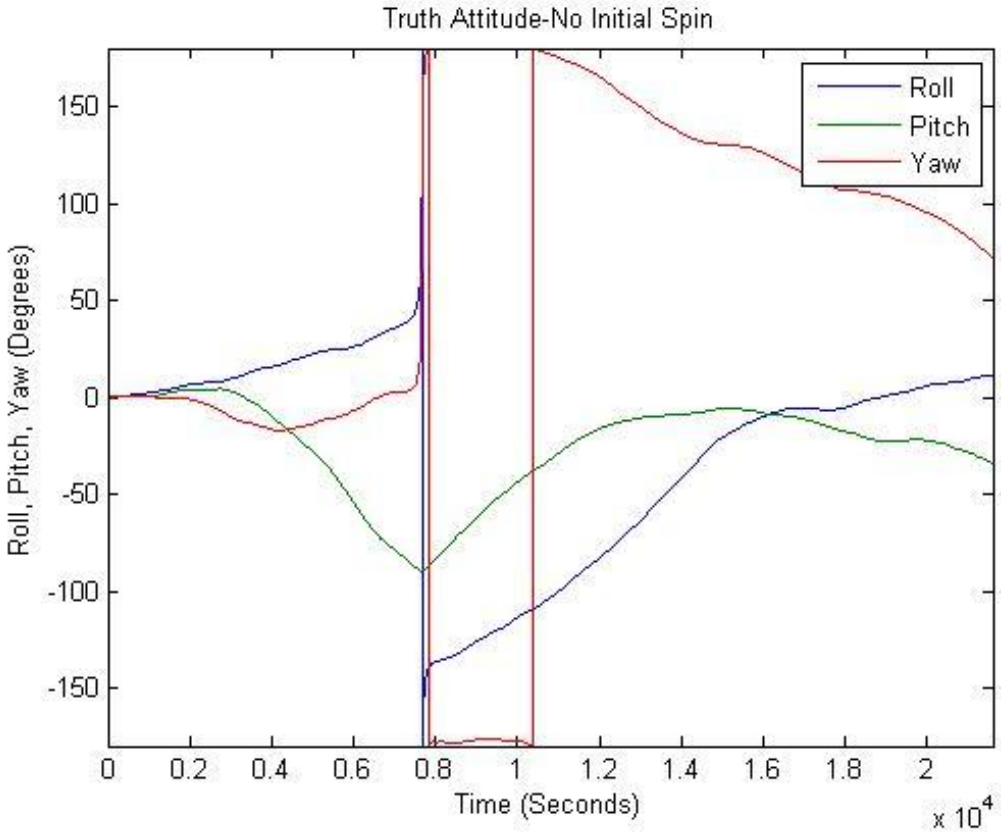

Figure 25: Truth Attitude with no initial spin

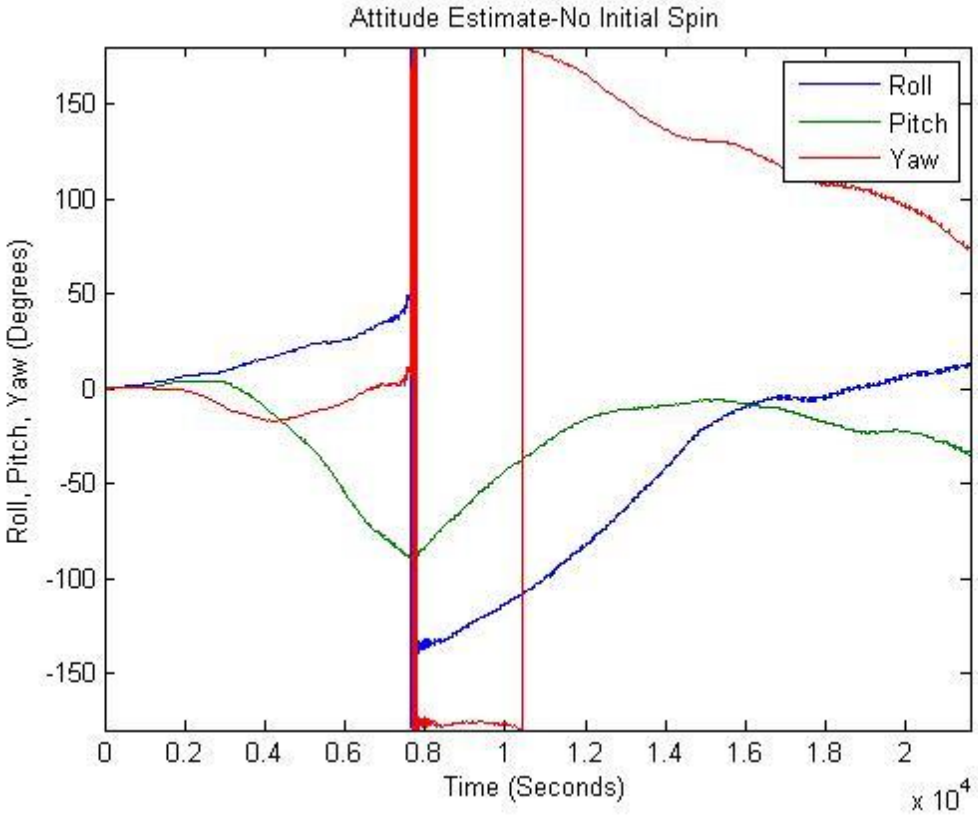

Figure 26: Attitude Estimate with no initial spin

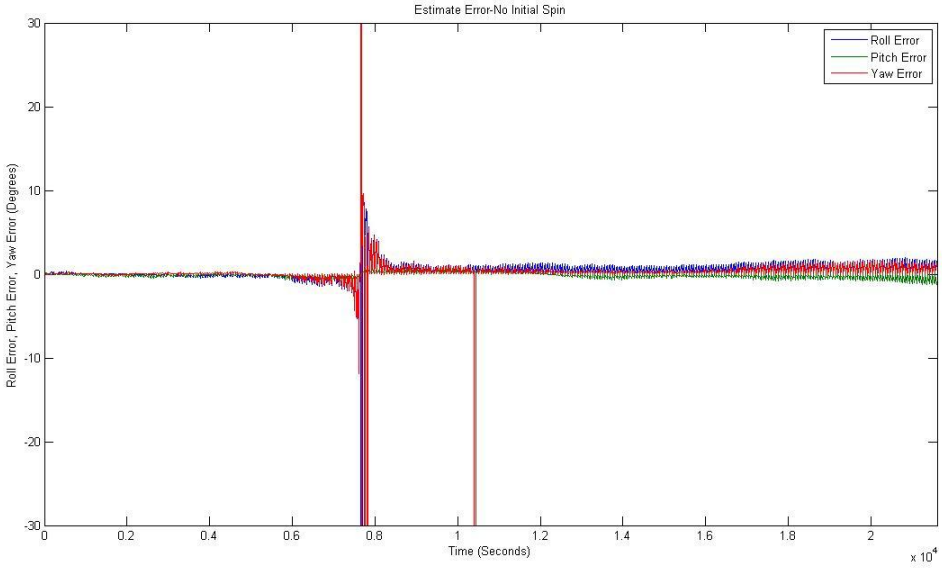

Figure 27: Attitude Estimate Error

6.2 Drift Compensation with Higher Roll Rate

The attitude determination system works equally well with higher roll rates. The roll rates in this simulation are on the order of 0.1 degrees/second in each second. This rate was selected because it is substantially higher than the resolution limits of a high performance, off the shelf MEMS gyro. Figure 28 shows the truth attitude. One of the differences between this simulation and the lower angular rates is the increased number of singularities associated with the more rapid role rates. Figure 29 shows the estimated attitude. Comparison of Figure 28 and Figure 29 demonstrates a close correlation between the estimated attitude and the truth attitude. Figure 30 shows the error in the estimate. The same singularities appear as with the lower angular rate. This error is very low, less than 1 degree, which is due to the availability of reliable gyro measurements to integrate, since the initial roll rates are within the gyros dynamic range.

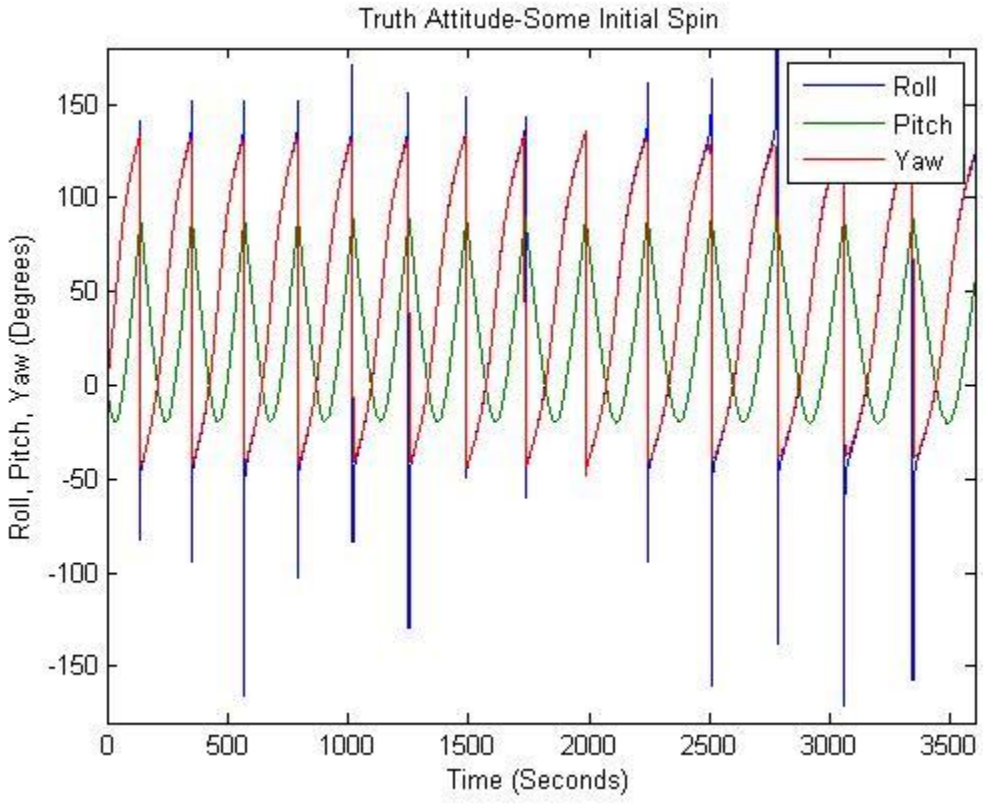

Figure 28: Truth attitude with some initial angular body rates

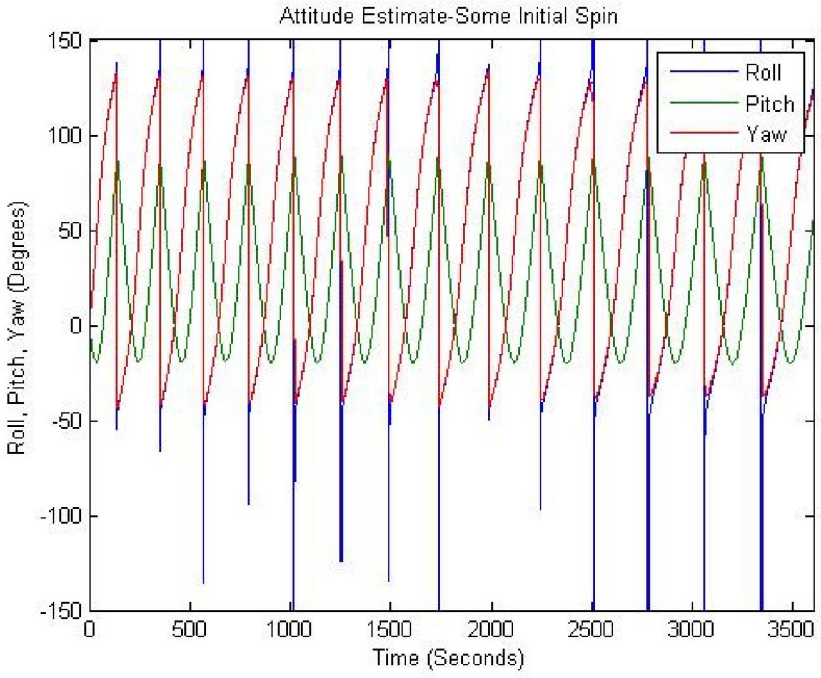

Figure 29: Attitude estimate with some initial spin

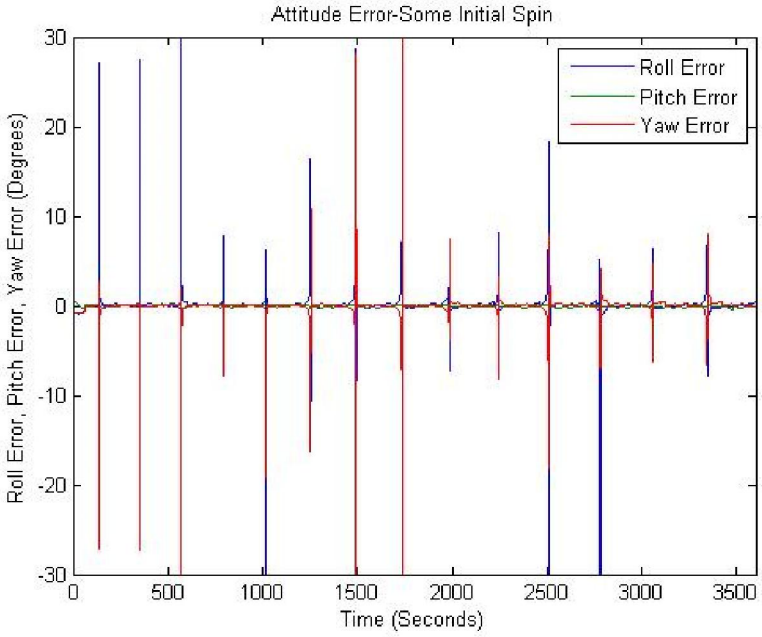

Figure 30: Attitude error with some initial spin

7 Conclusion

This section summarizes the major results of this thesis. The underlying theory and the results are discussed. Future work to build a flight ready system is also described.

7.1 Summary of Work

This thesis proposes a new attitude determination system for small satellites. This system is particularly suitable for CubeSats, since it eliminates the need for a large complement of hardware and can be realized using small and relatively low power sensors. The attitude drift problem is effectively dealt with using the new stellar gyro sensor.

The attitude determination system utilizes the solar panels as course sun sensors to provide an approximate fix on the sun. The satellite is subsequently rotated so that the fine sun sensor can acquire the sun. This sun sensor provides an accurate sun vector that, when combined with a magnetometer vector, enables the QUEST algorithm to determine the absolute attitude. This absolute attitude is then propagated by a quaternion Kalman filter that maintains attitude knowledge.

This system employs gyroscopes to provide high frequency updates of the system attitude. The use of gyroscopes inevitably leads to drift in the attitude estimate when the rate information from the gyroscopes is integrated by the Kalman filter. This drift leads to a steadily increasing error in the attitude estimate, eventually leading to a total loss in attitude knowledge over time.

Several sources of the gyro error are described. Errors result from the analog to digital conversion process and the Allan variance noise. These issues are partially compensated for using a low pass or moving average filter. The gyro resolution also introduces an additional source of error. The gyro resolution is a result of the quantization limits in the analog to digital conversion process. The resolution error results from the small angular rates generated by environmental torques. These angular rates are not measureable by the gyros.

The stellar gyro provides effective compensation for the drift errors. Using

image processing techniques to compare star field images, the stellar gyro is able to determine the drift free change in attitude. This provides a low frequency update of the attitude that is used to update the Kalman filter attitude.

This system is then implemented in the Simulink environment. Multiple sensor models are created in Simulink, including sun sensors, magnetometers, and gyroscopes. The noise characteristics, quantization errors, and sampling of these sensors are simulated. The QUEST algorithm and the quaternion Kalman Filter are implemented as well.

The results from the Simulink model show a marked improvement in the accuracy of the attitude estimate. These simulations are run with two different initial angular rates. 0.1 degree per second rotation was tested, as well as 0 degree per second initial rates. In both cases, the estimated attitude is compared to a truth model that represents the actual attitude as simulated by the Simulink dynamics model. The estimated attitude accurately tracks the truth attitude, and error is less than 1 degree in all three Euler angles for both initial conditions.

7.2 Future Work

Advancing this system from the simulation stage to a prototype system involves a significant amount of hardware and software work. Actuators to execute slewing maneuvers currently exist on the market, so off the shelf components can provide the maneuvering capability. In terms of sensors, at a minimum, this system requires a sun sensor, a magnetometer, and a MEMS gyroscope, as well as the stellar gyroscope. The first 3 sensors are also readily available, off the shelf commercial products. The stellar gyro, on the other hand, is still in development, with a proof of concept unit completed. Further work is required to reduce the power consumption of this unit and ruggedize it for the space environment.

Another focus needs to be selections of a low power microprocessor that can do the math required by the attitude determination algorithms. These calculations include implementations of a 400 point moving average filter, the QUEST algorithm, magnetic field models, sun vector reference model calculation

algorithm, and the Kalman Filter. An excellent candidate processor for this application is the ARM Cortex M4. Numerous implementations of this processor exist, but all processors containing the Cortex M4 core contain a floating point unit that is capable of completing IEEE 754 Standard Single Precision Floating Point Operations such as multiplication and addition in 1 processor clock cycle [47]. This is a huge advantage for an attitude determination processor, since single precision floating point is adequate for the attitude determination algorithms described in this thesis.

References

[1] J Puig-Suari, C Turner, and W Ahlgren, "Development of the standard CubeSat deployer and the Cubesat class PicoSatellite," *IEEE Aerospace Conference Proceedings*, pp. 347-353, 2001.

[2] C Kitts, K Ronzano, and R Rasay, "Initial Flight Results from the PharmaSat Biological Microsatellite Mission," in *23nd Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2009.

[3] James Cutler, Matthew Bennett, Andrew Klesh, Hasan Bahcivan, and Rick Doe, "The Radio Aurora Experlor - A Bistatic Radar Mission to Measure Space Weather Phenomenon," in *USU Small Satellite Conference*, Logan, Utah, 2010.

[4] S Janson, B Hardy, and A Chin, "Attitude Control on Pico Satellite Solar Cell Testbed-2," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[5] Kevin Brown, Tyler Rose, and Ben Malphrus, "The Cosmic X-Ray Background NanoSat (CXBN): Measuring the Cosmic X-Ray Background using the CubeSat Form Factor," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[6] Daniel Erb, Twyman Clements, Benjamin Malphrus, and James Lumpp, "Kentucky Space: A Multi-University Small Satellite Enterprise," in *23rd Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2009.

[7] G Skrobot, ""ELaNa – Educational Launch of Nanosatellite Enhance Education through Space Flight," in *25th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2011.

[8] James Wertz and Wiley Larson, *Space Mission Analysis and Design*. Hawthorne, California: Microcosm Press, 1999.

[9] Samir Rawashdeh, Passive Attitude Stabilization for Small Satelllites-Thesis, January 2010, University of Kentucky Libraries.

[10] David Vallado, *Fundamentals of Astrodynamics and Applications*. Hawthorne, California: Microcosm Press, 2007.

[11] James Wertz, Ed., *Spacecraft Attitude Determination and Control*. Dordrecht, Netherlands: Kluwer Academic Publishers, 1978.

[12] Marcel J. Sidi, *Spacecraft Dynamics and Control: A Practical Engineering Approach*. London, United Kingdom: Cambridge University Press, 1997.

[13] Dave Doody, *Deep Space Craft: An Overview of Interplanetary Flight*. New York: Springer Praxis Books, 2010.

[14] Bong Wie, *Space Vehicle Dynamics and Control*. Reston, Virginia: AIAA Education Series, 2008.

[15] Bryan Johnston-Lemke, High Performance Attitude Determination and Control for Nanosatellites, December 2011.

[16] M Aherne and M Barrett, "Aeneas-Colony I Meets Three-Axis Pointing," in *25th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2011.

[17] R Votel and D Sinclair, "A Comparison of Control Moment Gyros and Reaction Wheels for Small Earth-Observing Satellites," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[18] C Clark and K Worrall, "A Control Moment Gyro for Dynamic Attitude Control of Small Satellites," in *24th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2010.

[19] Frank O'Brien, *The Apollo Guidance Computer: Architecture and Operation*. New York: Springer Praxis Books, 2010.

[20] M Taraba, C Rayburn, and A Tsuda, "Boeing's CubeSat TestBed 1 Attitude Determination Design and On-Orbit Experience," in *23rd Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2009.

[21] Michael Greene, The Attitude Determination and Control System of the Generic-Thesis, 2009, University of Toronto T-Space.

[22] J Enright and D Sinclair, "The Things You Can't Ignore: Evolving a Sub-Arcsecond Star Tracker," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[23] J Enright and D Sinclair, "Towards Star Tracker Only Attitude Estimation," in *24th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2010.

[24] G Falbel, "A low weight/power/cost infrared Earth sensor," in *IEEE Aerospace Conference Proceedings*, 2004, pp. 2716 – 2722.

[25] A.W. van Herwaarden, "Low-Cost Satellite Attitude Control Sensors Based Control Sensors Based on Integrated Detector Arrays," *IEEE Transactions on Instrumentation and Measurement*, pp. 1524-1529, 2001.

[26] Samir Rawashdeh, "Development of a Drift-Free Stellar Gyroscope," in *25th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2011.

[27] Samir Rawashdeh, "A Stellar Gyroscope for Small Satellite Attitude Determination," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[28] Samir Rawashdeh, William Danhauer, and James Lumpp, "Stellar Gyroscope: Visual Spacecraft Attitude Propagation from Apparent Motion of Stars," in *33rd IEEE Aerospace Conference*, Big Sky, Montana, 2012, pp. 1-9.

[29] Orlando Diaz, Analysis and comparison of extended and unscented Kalman filtering methods for spacecraft attitude determination, December 2010.

[30] F. L. Markley and D Mortari, "Quaternion Attitude Estimation Using Vector Observations," *The Journal of the Astronautical Sciences*, pp. 359-380, 2000.

[31] John Crassidis and John Junkins, *Optimal Estimation of Dynamic Systems*. Boca Raton, Florida : CRC Press, 2012.

[32] M. D Shuster, "The Quest for Better Attitudes," *The Journal of the Astronautical Sciences*, pp. pp. 657–683, 2006.

[33] Y Cheng and M Shuster, "Robustness and Accuracy of the QUEST Algorithm," in *AAS/AIAA 17th Space Flight Mechanics*, Sedona, Arizona, 2007, pp. 41-61.

[34] Norman Nise, *Control Systems Engineering*. Hoboken, New Jersey: John Wiley & Sons, 2008.

[35] Richard Dorf and Robert Bishop, *Modern Control Systems*. Upper Saddle River, New Jersey: Pearson Prentic-Hall , 2005.

[36] Arthur Gelb, *Applied Optimal Estimation*. Cambridge, Massachusetts: MIT Press, 1974.

[37] Benjamin Kuo, *Digital Control Systems*. New York: Saunders College Publishing, 1992.

[38] Mohinder Grewal and Angus Andrews, *Kalman Filtering: Theory and Practice using MATLAB*. Hoboken, New Jersey: John Wiley & Sons, 2008.

[39] Alan Oppenheim, Ronald Schafer, and John Buck, *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Pearson Prentice-Hall, 1999.

[40] Steven Smith, *The Scientists & Engineer's Guide to Digital Signal Processing*. San Diego, California: California Technical Publications , 1997.

[41] Angelo Sabatini, "Quaternion-Based Extended Kalman Filter for," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, pp. 1346-1356, 2006.

[42] Robert Stengel, *Optimal Control and Estimation*. New York: Dover, 1994.

[43] D Choukron, I Y Bar-Itzhack, and Y Oshman, "Novel Quaternion Kalman Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174-190, January 2006.

[44] Daniel Erb, Twyman Clements, Benjamin Malphrus, and James E Lumpp, "Kentucky Space: A Multi-University Small Satellite Enterprise," in *23rd Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2009.

[45] Garrett Chandler et al., "Development of an Off-the-Shelf Bus for Small Satellites," in *IEEE Aerospace Conference*, Big Sky, Montana, 2007.

[46] James E Lumpp, Daniel M Erb, Twyman S Clements, Jason T Rexroat, and Johnson Michael D, "The CubeLab Standard for Improved Access to the International Space Station," in *IEEE Aerospace Conferencec*, Big Sky Montanna, 2011.

[47] Armen Toorian, Ken Diaz, Simon Lee, and NASA Jet Propulsion Laboratory, "The CubeSat Approach to Space Access," in *IEEE Aerospace Conference*, Big Sky, Montana, 2008.

[48] Alan N Gent, *Engineering with Rubber: How to Design Rubber Components*, 2nd ed. Cincinnati, United States of America: HanserGardner Publications, 2001.

[49] J Springmann, B Kempke, and J Cutler, "Initial Flight Results of the RAX-2 Satellite," in *26th Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2012.

[50] Malcom Shuster, "The Quest for Better Attitudes," *The Journal of Astronautical Sciences*, vol. 54, no. 3-4, pp. 657-683, July-December 2006.

VITA

Maxwell Otto Bezold
Fort Thomas, Kentucky

Education: University of Kentucky
Bachelors of Science in Chemical Engineering, Graduated December 2010
Cumulative undergraduate GPA: 3.25
Cumulative graduate GPA: 3.67

Awards and Activities:
Presidential Full Tuition Scholarship
Kentucky Governor's Scholar
University of Kentucky Honor's Program

Experience:
Space System Laboratory**,** Lexington, Kentucky,
September 2009 – August 2013 *Graduate Research Assistant*
Ball Aerospace, Boulder, Colorado,
May 2012- August 2012 *Embedded Software Intern*
NASA AMES Research Center, Mountain View, California,
July 2011 – August 2011 *Research Intern*
Patheon Pharmaceuticals, Cincinnati, Ohio,
January 2007- August 2008 *Embedded Software Intern*