

MAD-1701



Design and Analysis of the Sphinx-NG CubeSat

A Major Qualifying Project
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In Partial Fulfilment of the Requirements for The
Degree of Bachelor of Science
By

Jack Agolli

James Gadoury

Andrew Rathbun

March 3rd, 2017

Approved by:

A blue ink signature of Professor Michael A. Demetriou, Advisor to the Aerospace Engineering Program.

Professor Michael A. Demetriou, Advisor
Aerospace Engineering Program

WPI

Abstract

This project continues the development of the Attitude Determination and Control (ADC) subsystem for a three-unit CubeSat in a high-altitude, polar, sun-synchronous orbit mission to perform solar and extraterrestrial X-ray spectroscopy. The previously outlined list of sensors and actuators were evaluated and updated. The performance of algorithms used for detumble, initial attitude determination, and attitude maintenance was improved by using MATLAB® and Systems Tool Kit (STK) simulations. Additionally, a low-cost, prototype ADC test bed was constructed which will be used by future teams to test and verify the selected ADC hardware. Finally, a detailed MATLAB® code guide, derivations of ADC algorithms, and recommendations were developed to assist future CubeSat ADC teams.

"Certain materials are included under the fair use exemption of the U.S. Copyright Law and have been prepared according to the fair use guidelines and are restricted from further use."

Acknowledgments

We would like to primarily thank our advisor:

Professor Michael Demetriou, Ph.D

Professor, Aerospace Engineering Program

Department of Mechanical Engineering, Worcester Polytechnic Institute

We would also like to extend our thanks to the members of the 2011, 2012, and 2013 ADCS team.

We also want to thank James Loiselle, Senior Instructional Laboratory Technician at the Higgins Lab facility, for his immense help in machining the first test bed prototype.

In addition, we would like to thank the advisors and members of the other CubeSat subsystem design teams:

Thermal, Power, Telecommunications, and Command and Data Handling Subsystem,

Professor John J. Blandino, Ph.D.

Associate Professor, Aerospace Engineering Program

Department of Mechanical Engineering, Worcester Polytechnic Institute

Structural and Mission Analysis Subsystem,

Professor Nikolaos Gatsonis, Ph.D.

Director, Aerospace Engineering Program

Department of Mechanical Engineering, Worcester Polytechnic Institute

Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 CubeSat Background	1
1.1.1 The WPI CubeSat Mission	1
1.1.2 CubeSat Subsystems	3
1.2 Previous MQP Review	5
1.3 Project Objectives, Approach, and Methods	6
1.4 Motivation	9
2 Observation Models	11
2.1 Geomagnetic Reference Field Model (IGRF 12)	11
2.2 Sun Vector Model	12
3 Attitude Determination and Control	14
3.1 Subsystem Overview	14
3.2 Sensor and Actuator Selection	15
3.2.1 Fine Sun Sensor	15
3.2.2 Coarse Sun Sensor	17
3.2.3 Magnetometer	18
3.2.4 Gyroscope	19
3.2.5 Magnetic Torquer	21
3.2.6 Global Positioning System (GPS)	22
3.3 ADC Algorithms and Methods	23
3.3.1 Spacecraft Detumble	23
3.3.2 Initial Attitude Determination	26
3.3.3 The TRIAD Method	26
3.3.4 Wahba's Problem	28
3.3.5 Attitude Maintenance	32

4 ADC Simulations and Results	37
4.1 Spacecraft Detumble Simulations	37
4.1.1 Single-Fixed Gain	37
4.1.2 Adaptive Gain	39
4.2 Attitude Determination and Maintenance Simulations	40
4.2.1 Extended Kalman Filter Simulation	41
4.2.2 Exact Angles Simulation	41
4.3 Power and Current Requirements	42
4.3.1 Power Calculation Overview	42
4.3.2 Current Calculation Overview	44
4.3.3 Power and Current Profiles for Detumble Phase	44
4.4 Systems Tool Kit (STK) Simulations	47
4.4.1 STK Attitude Coverage	48
4.4.2 STK Attitude Control Simulation	50
5 ADC Test Bed	52
5.1 Test Bed Design and Prototype Construction	52
5.2 Prototype Testing	57
6 Conclusions and Recommendations	60
6.1 Project Summary	60
6.1.1 ADC Subsystem Overview	60
6.1.2 ADC Test-Bed Overview	63
6.2 Recommendations	65
6.2.1 Understanding the ADC Algorithms and Methods	65
6.2.2 Improving the ADC Simulations	65
6.2.3 Rewriting the Extended Kalman Filter	66
6.2.4 Continuation of STK Simulations	66
6.2.5 Future Work on the Test Bed	67
References	70
Appendices	71
Appendix A - Quaternions	72
Appendix B - Magnetic Torquer Overview	74
Appendix C - Spacecraft Detumble Phase	78
Appendix D - Initial Attitude Determination	82
Appendix E - Attitude Maintenance Phase	87
Appendix F - MATLAB Simulation Guide	94
Appendix G - STK Data	112
Appendix H - STK Setup Screen shots	113

Appendix I - Fine Sun Sensor Data Sheet	120
Appendix J - Coarse Sun Sensor Data Sheet	122
Appendix K - Magnetometer Data Sheet	125
Appendix L - Gyroscope Data Sheet	146
Appendix M - Magnetic Torquer Data Sheet	177
Appendix N - GPS Data Sheet	179

List of Figures

1.1	Side-view and perspective view of the Sphinx-NG.	2
1.2	P-POD coordinate system.	3
1.3	CubeSat coordinate system.	3
1.4	Iterative ADCS design process.	9
3.1	Mission phases of the ADCS.	15
3.2	NSS CubeSat sun sensor.	16
3.3	Space Micro CSS-01,02.	17
3.4	Honeywell HMC5883L magnetometer.	19
3.5	ADXRS453 and EVAL board in the vertical (left) and SOIC (right) configurations.	20
3.6	ZARM-Technik MTO-5.1 magnetic torquer.	22
3.7	Surrey SGR-05U GPS.	23
4.1	Detumble time versus fixed gain.	38
4.2	Detumble simulation with single-fixed gain in seconds.	38
4.3	Detumble simulation with single-fixed gain in orbital periods.	39
4.4	Detumble simulation with adaptive gain in seconds.	40
4.5	Detumble simulation with adaptive gain in orbital periods.	40
4.6	Simulation of EKF.	41
4.7	Simulation of exact angles.	42
4.8	Current plot of fixed gain controller.	45
4.9	Current plot of adaptive gain controller.	45
4.10	Power plot of fixed gain controller.	46
4.11	Power plot of adaptive gain controller.	46
4.12	STK sun alignment with nadir constraint.	48
4.13	STK Attitude Sphere.	48
4.14	Sun sensor conical FOV in STK.	49
4.15	STK attitude coverage of the +X facing fine sun sensor.	50
4.16	STK MATLAB simulation process, configuration 2.	50
4.17	STK detumble simulation in 3D graphics window.	51
4.18	STK detumble simulation (with attitude sphere shown).	51

5.1	Three general types of ADC test beds.	52
5.2	(left to right) WPI 2013 MQP Team, Hawaii Space Flight Lab, Naval Postgraduate School.	53
5.3	Induced moment when center of mass is not balanced.	53
5.4	Center of mass balancing.	54
5.5	Spherical air bearing pressure chamber design.	56
5.6	Spherical air bearing assembly.	56
5.7	Fully rendered test bed design.	57
5.8	Adafruit 9-DOF Fusion IMU Breakout.	58
5.9	Adafruit Feather HUZZAH with ESP8266 WiFi.	58
5.10	EFOSHM Ultra Slim Battery.	58
5.11	Raspberry Pi 3.	59
6.1	Block diagram of the final ADCS.	63
6.2	Test bed prototype.	64
6.3	Air bearing.	64

List of Tables

1.1	Steps in attitude subsystem design.	7
3.1	Fine sun sensor technical specifications.	16
3.2	Space Micro CSS-01,02 technical specifications.	18
3.3	Honeywell HMC5883L technical specifications.	19
3.4	Gyroscope technical specifications.	20
3.5	ZARM-Technik MTO-5.1 technical specifications.	22
3.6	Surrey SGR-05U technical specifications.	23
4.1	Sensor data used in simulation.	41
4.2	Description of variables used during magnetic torquer analysis.	43
4.3	Sun sensor placement in STK.	49
5.1	Parts ordered for initial construction of test bed.	55
6.1	Overview of ADCS hardware.	61

Chapter 1

Introduction

1.1 CubeSat Background

Dr. Jordi Puig-Suari of California Polytechnic State University San Luis Obispo (Cal Poly) and Dr. Bob Twiggs of Stanford University established the CubeSat standard in 1999 in order to facilitate access to space for students. This standard defines a one unit (1U) CubeSat as a 10 cm cube with a mass of approximately 1.33 kg that must be electronically off until it has been deployed by the Poly Picosatellite Orbital Deployer (P-POD) [1]. The P-POD, also designed at Cal Poly, is the interface between the launch vehicle and the CubeSat. All CubeSats must interfacing requirements with P-POD. The P-POD has room for up to 3U of CubeSats, so in general satellites are designed to be 3U in size to make use of all available space. CubeSats are generally secondary payloads, so the satellite deployer is needed to insert the CubeSat into its desired orbit. Launch opportunities are frequently available on most launch vehicles, but the desired orbit may not always be reachable [2]. The CubeSat standard has allowed universities like WPI an affordable way to design and operate missions to space, which provides students with invaluable hands-on experience. In addition, reduced cost of space missions and the increased demand for spacecraft has allowed smaller companies and organizations to develop, test, and sell satellite technology, resulting in an influx of small startup companies supporting CubeSats and full-scale satellites.

1.1.1 The WPI CubeSat Mission

The WPI CubeSat mission goal is to place a 3U CubeSat into a sun-synchronous 600km polar orbit for space weather observation. The scientific goal will be achieved by the Sphinx-NG, an instrument in development at the Space Research Centre in Warsaw, Poland. The Space Research Centre is a research institute that is a part of the Polish Academy of Sciences with a primary goal of researching terrestrial space. The weather observation will primarily be done by solar and terrestrial X-ray spectroscopy, which can be broken down into a) solar X-ray monitoring of long-term flux variability, nonactive corona, active regions, solar flares, temperature and differential emissions, and plasma abundances and b) terrestrial X-ray and particle observations, including X-ray signatures of terrestrial gamma-ray flashes (TGFs), auroral X-ray spectra, and orbital particle fluctuations [3]. Figure 1.1 below depicts the Sphinx-NG

science instrument [3]:

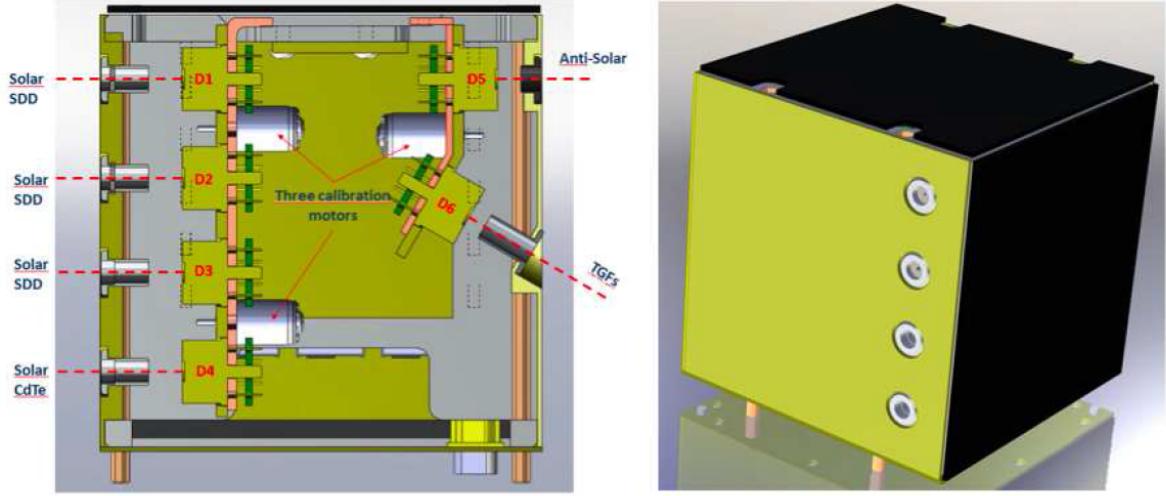


Figure 1.1: Side-view and perspective view of the Sphinx-NG.

As can be seen in the diagram above, the Sphinx-NG has four solar detectors out of six total multi-channel X-ray detectors. These four must face directly towards the sun and have an accuracy range of ± 1 degree. Silicon drift detectors are used to detect the soft energy domains, which are X-rays of approximately 0.8-1.5 keV and Schottky diode detectors are used for the 5-150 keV domains [3]. These accuracy requirements require a well-developed attitude control system to maintain such a precise view of the sun. Since the satellite is a picosatellite, we have extremely limited space for sensors and actuators and highly stringent power, thermal, and computational requirements for them. Due to this, the relatively high accuracy requirements for our mission must be obtained with relatively low accuracy sensors. Our project is a continuation of three years of previous work to determine the best way to design a system that meets these requirements.

The primary mission requirement for the ADC subsystem is to align the $+x$ body axis of the spacecraft with the center of the sun to within 1-2 degrees of accuracy during lighting periods. This satisfies the Sphinx-NG's pointing requirement, and it also satisfies the pointing requirement of the CubeSat for power generation, since the deployed solar-panels were designed to also face the $+x$ direction of the spacecraft body-fixed coordinate system. Figures 1.2 and 1.3 below demonstrate the body-fixed reference frame.

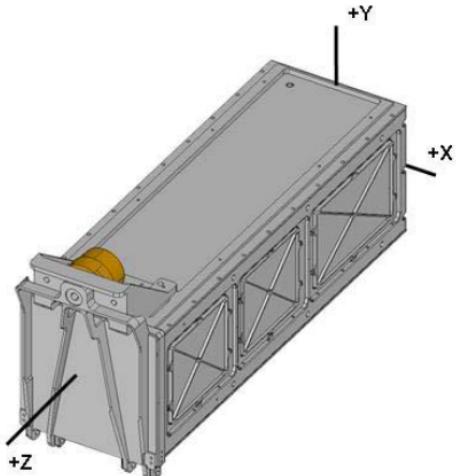


Figure 1.2: P-POD coordinate system.

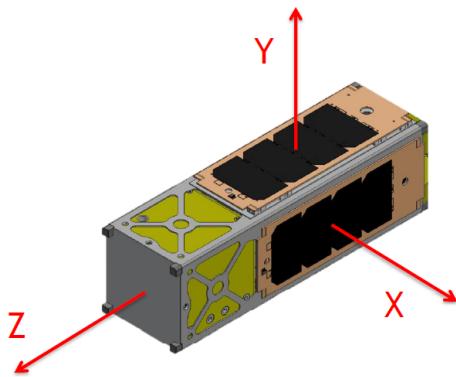


Figure 1.3: CubeSat coordinate system.

This CubeSat coordinate system was chosen by the Mechanical and Structural team in order to align with both the P-POD coordinate system and the Systems Tool Kit (STK) coordinate system. Simulating the mission in STK was a significant portion of the work done in this year's project across all the subsystems as it was the first time WPI had licensing access to it. The software and the work done with it will be discussed in depth in a later chapter.

1.1.2 CubeSat Subsystems

This MQP is a part of a large group that performs a conceptual design of a 3U CubeSat which carries the Solar Photometer in X-rays-Next Generation (Sphinx-NG) instrument. The Space Research Center (SRC) at the Polish Academy of Sciences is leading the design of the Sphinx-NG instrument as a miniaturized version of the Sphinx which flew onboard the Coronas-Photon spacecraft [4].

The organizational structure of the CubeSat MQP was divided into three subsystems, the Attitude Determination and Control (ADC) Subsystem, the Orbital Analysis and Mechanical Subsystem, and the Thermal, Telecommunication, Command and Data Handling, and Power Subsystem and an inclusive,

Systems Engineering Group. The subsystems and their duties are broken up as follows:

The Attitude Determination and Control team contained three members. Details of the ADC subsystem design are given in this report. The team had the following responsibilities:

1. Sensor and actuator hardware selection
2. Attitude determination method selection
3. Attitude control method selection
4. Software simulations of ADC subsystem
5. Hardware testing of ADC subsystem

The Orbital Analysis and Structural team contained four members. For a detailed review of the Orbital Analysis and Structural team's objectives and results, refer to MQP Report NAG-1104 [5]. The team had the following responsibilities:

1. Define Orbital Parameters
2. Mechanical Design
3. Analyze Potential Electromagnetic Interference

The Thermal, Power, Telecommunications, and Command and Data Handling team contained five members. For a detailed review of the Thermal, Power, Telecommunications, and Command and Data Handling team's objectives and results, refer to MQP Report JB17-01 [6]. The team had the following responsibilities:

1. Thermal Analysis
2. Power Distribution System Design and Power Usage Tracking
3. Telecommunication Link Budgeting
4. Data Command and Handling, On-Board Computer System Design

Lastly, the project contained a Systems Engineering Group (SEG) that was the combination of all three subsystems into one engineering group working on the design and analysis of the entire WPI CubeSat mission. the SEG had weekly meetings of all three subsystems where the teams received updates on the progress of each respective subsystem. This allowed easy resolution of conflicts, set and update action items, view results, and advise across all subsystems. This structure gave each team an organized and overarching view of the entire project and its progress throughout. In addition to this collective meeting, the ADC subsystem provided weekly updates to the project advisor.

1.2 Previous MQP Review

During the 2011-12 academic year, WPI began its relationship with the Polish Academy of Sciences in Poland and NASA's Goddard Space Flight Center to develop a 3U CubeSat which could house the Sphinx-NG [3]. The 2011-12 group consisted of 16 students divided into three MQP teams. Dopart et al. [3] presents orbital and decay analysis using Systems Tool Kit (STK), the selection of the GPS and the magnetometer, ambient and induced environment analysis using COMSOL, and a preliminary discussion on command and data handling and the on-board computer. Farhead et al. [7] presents the hardware selection of the gyroscope, sun sensors, and magnetic torquers, attitude determination algorithms, and control policies. Bauer et al. [8] presents environmental and component-induced thermal analysis, component and assembly design, preliminary stress analysis, and power generation and management. The most recent version of the CubeSat was developed by sixteen students separated into three groups during the 2012-13 academic year. Billings et al. [9] presents the mechanical design, orbital analysis using STK, and an analysis of electromagnetic interference induced by the magnetic torquers. Dawson et al. [10] presents the selection of sensor, actuator, and processor hardware, the attitude control algorithm using MATLAB, and the preliminary design of an attitude control test-bed stand. Hanley et al. [11] presents the analysis of the CubeSat power budget, design of the wiring diagram, thermal analysis using STK and COMSOL, and telecommunications analysis using STK.

The first iteration in 2011 of the Attitude Determination and Control Subsystem (ADCS) team was a pair of students led by Professor Michael Demetriou. These students, Andrew Bigelow and Cyle Hawkins, did the base-line work of researching, deriving, and proving the validity of the various control methods that would work with our mission requirements. They explained why certain types of sensors and actuators should be chosen, along with preliminary testing of the methods they designed. Their work was significantly more theoretical in nature than later stages of this project will be. The most significant contribution of this team was the development of the MATLAB® simulation Graphical User Interface (GUI) that the later teams used for the testing of control methods, determination, and hardware [12].

The 2012 MQP team, comprised of Elizabeth Dawson, Nell Nassiff, and Dianna Velez, provided detailed explanations of the control methods that will be used in this project, along with alternative methods. A significant portion of their project was dedicated to research and examination of control methods used on pre-existing CubeSat missions. Towards the end of their project, they began preliminary research and development for an ADCS test bed [7].

The most recent work on the ADC subsystem was done in 2013 by Assand Farhant, Jighjigh Ivase, Ye Lu, and Alan Snapp. The first part of their project was dedicated to finalizing the hardware selection from the wider range of devices that were introduced in the 2012 ADCS MQP report. They introduced a system architecture and an iterative simulation procedure from IEEE report [13]. The team then briefly outlined observation models for the Earth's magnetic field and alternative attitude determination methods. Afterwards, they dedicated a significant portion of their report to running MATLAB® simulations of the control policies and trying to optimize them using the proposed simulation procedure. Lastly, the 2013 MQP team outlined a structural design and electronic setup for a test bed for hardware-in-the-loop

testing [10].

1.3 Project Objectives, Approach, and Methods

This project began with looking at ADCS from a broad, sub-system level perspective of spacecraft mission design. After determining which point in this design process the project was currently (based on the work done by projects of previous years), the next steps were chosen to continue the process. The final method used was choosing a simulation procedure for testing the control modes (the same process as the project done in 2013 was used). Shown below in Table 1.1, adapted from [14], is a general procedure for attitude system design:

Step	Inputs	Outputs
1a) Define Control Modes 1b) Define or Derive System-Level Requirements by Control Mode	Mission requirements, mission profile, type of insertion for launch vehicle	List of different control modes during mission. Requirements and constraints.
2) Quantify Disturbance Environment	Spacecraft geometry, orbit, solar/magnetic models, mission profile	Values for torques from external and internal sources
3) Select Type of Spacecraft Control by Attitude Control Mode	Payload, thermal & power needs Orbit, pointing direction Disturbance environment Accuracy requirements	Method for stabilization & control: three-axis, spinning, gravity gradient, etc.
4) Select and Size ADCS Hardware	Spacecraft geometry and mass properties, required accuracy, orbit geometry, mission lifetime, space environment, pointing direction, slew rates. Failure detection and redundancy	Sensor suite: Earth, Sun, inertial, or other sensing devices. Control actuators: reaction wheels, thrusters, magnetic torquequers, etc. Data processing avionics, if any, or processing requirements for other subsystems or ground computer.
5) Define Determination and Control Algorithms	Performance considerations (stabilization method(s), attitude knowledge & control accuracy, slew rates) balanced against system-level limitations (power and thermal needs, lifetime, jitter sensitivity, spacecraft processor capability)	Algorithms and parameters for each determination and control mode, and logic for changing from one mode to another.
6) Iterate and Document	All of above	Refined mission and subsystem requirements. More detailed ADCS design. Subsystem and component specifications.

Table 1.1: Steps in attitude subsystem design.

Based on a review of previous work, the design process when this project began fell into steps four, five, and six. Therefore, this put the overall progress of the project at the later stages of the design process. In order to accomplish these steps (selecting hardware, defining algorithms, and then iterating and revising them) the approach decided on was as follows:

1. Finalize hardware list

- (a) The sensors and actuators for the ADC system were filtered down through the years since this project began. It started with a wide range of devices and as of 2013 there was a finalized list of one device for each role. The beginning of the project was dedicated to examining why these particular sensors and actuators were chosen, and ensuring that they were still up-to-date and suitable for the mission. If this was found to not be the case, then replacement devices were to be chosen.

2. Finalize attitude determination algorithms

- (a) Research and examine determination methods outlined by previous MQPs and previous CubeSat missions
- (b) Simulate these methods by following simulation procedure and obtaining verifiable results
- (c) Finalize primary and backup determination methods based on results for different power and computing scenarios

3. Finalize attitude control policies

- (a) Examine and simulate control methods outlined in previous MQPs and previous CubeSat missions
- (b) Finalize primary and backup control methods for different power and computing scenarios that satisfy our performance requirement

4. Determine possible ways to improve attitude determination and control policies

5. Develop MATLAB[®] code for simulation

- (a) Edit and debug existing MATLAB[®] code
- (b) Write new MATLAB[®] scripts that employ more effective control methods and power/computing scenarios

6. Design and begin development of test bed for testing the ADC system with actual hardware

- (a) Examine designs by other CubeSat projects and the previous MQP
- (b) Design a prototype and build it
- (c) Verify and update design of final structure
- (d) Begin building test bed

Figure 1.4 below, adapted from [15], demonstrates the iterative ADCS design process used throughout this project:

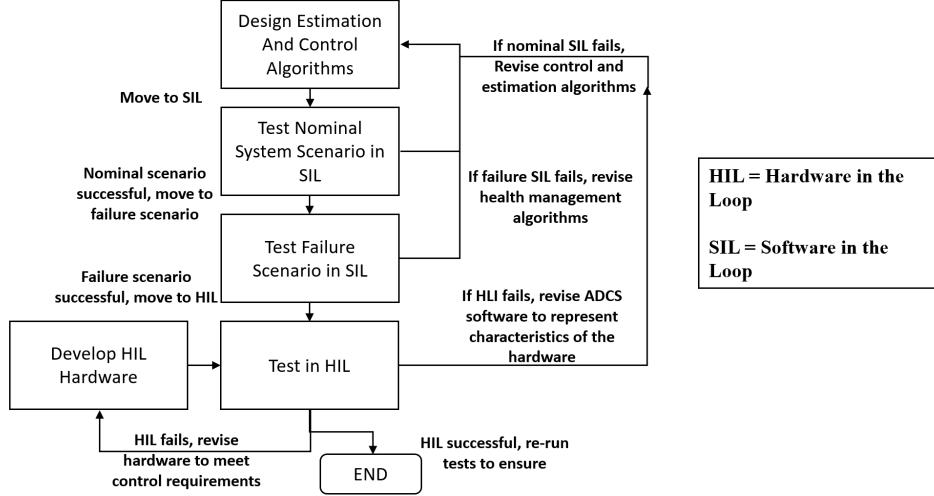


Figure 1.4: Iterative ADCS design process.

1.4 Motivation

The CubeSat mission allows aerospace engineering students at WPI to apply the skills they have spent several years developing in the classroom to producing deliverable proposals and blueprints for an actual satellite. This offers invaluable engineering experience for working in teams on spacecraft systems and subsystems. The ADCS subsystem is particularly useful to the entire mission for several reasons.

Generally, when CubeSats are launched from the P-POD, they are not stabilized and are tumbling at random, but constant, angular velocities along multiple axes. Without stabilization and the ability to point the satellite to a desired orientation, the satellite would be unable to be powered. Without power, all other functions of the satellite would not function, including the Sphinx-NG payload. If this were the case, the mission would automatically fail because the scientific instrument would not be able to gather the required data. Two of the biggest limitations to CubeSats are known to be communication bandwidth and power [1] and without a means of orienting the satellite, neither of these functions would be available. The telemetry knowledge that comes from ADC and our determination knowledge is also invaluable to many of the other subsystems. The ADC system is thereby essential to achieving our overall mission requirements.

In an IEEE report on cell efficiency dependence on solar incidence angle, the author determines a relation between solar cell efficiency and the incidence angle, using linear regression techniques [16]:

$$\text{efficiency} = 1.81\% \sin \theta + 27.07\% \cos \theta - 1.86\% \tan \theta \quad (1.1)$$

In equation (1.1), the angle θ is the solar incidence angle, which is the angle between the sun vector and the surface of the CubeSat. The current mission requirements can be written in terms of an error angle, which is defined as the angle between the $+x$ axis of the CubeSat and the sun vector in the body frame. This angle is related to the incidence angle. From the solar cell efficiency equation above, it can then be seen that it is possible to improve the efficiency of power generation by maintaining a high pointing accuracy. If the angle is less than or equal to 50° , for example, the equivalent efficiency of body-mounted solar panels is achieved [1].

The control accuracy requirements for this mission are comparatively not as precise as those of larger satellites in that only one-axis control is required rather than three-axis control. As long as the Sphinx-NG instrument and the solar panels are in full view of the sun, the ADC requirements are met. Since the only necessary control is to align one axis of the satellite with the sun, the use of highly accurate but more structurally complex actuators (that can be prone to mechanical failure), like reaction wheels, can be sacrificed. Instead the system can be restricted to using only magnetic torquers. Smaller but less accurate sensors can also be used to determine position. This saves the space and weight expenses that more precise sensors like star trackers would require. It is possible that the other subsystems would deem control of other axes necessary to aid in meeting their mission requirements. For instance, the telecommunications subsystem could desire a nadir constraint for alignment of the antenna. If this is the case, the control methods will have to be adjusted accordingly.

Chapter 2

Observation Models

The attitude determination method used in this project requires two reference vectors in cohesion with two observation vectors in order to determine the satellite's orientation. The determination methods will be discussed more in the next chapter, but the two vectors that will be used are the local magnetic field vector and the sun vector. These will be read by onboard sensors and will be used in attitude calculations alongside the reference magnetic field vector and sun vector. To obtain these two reference vectors, a sun vector model and a magnetic field model will be used.

2.1 Geomagnetic Reference Field Model (IGRF 12)

IGRF 12 is the 12th and latest generation of the International Geomagnetic Reference Field developed by the International Association of Geomagnetism and Aeronomy (IAGA) [17]. The model is by far the most widely used geomagnetic field model to date. It is a series of mathematical models describing the large scale internal part of the Earth's magnetic field between epochs 1900 A.D. and the present (or in other words, it is valid from 1900-2020). The current model, IGRF 12, was published in December 2014 and is valid up through 2020. If this project is taken up again at a date later than this period, the next generation model will have to be used. It is necessary to use the most recent published version of the model to maintain accuracy because each model is revised to reflect temporal changes of the geomagnetic field generated in the Earth's outer core.

The IGRF model described in [17] defines the internal geomagnetic field vector as $\mathbf{B}(r, \theta, \phi, t)$ and its annual rate of change. It is in spherical polar coordinates where r is the radial distance from the Earth's center, θ is the geocentric co-latitude, ϕ is the east longitude, and t is the current time. The GPS should be able to provide the corresponding position information, and the on-board-computer (OBC) should give the current time. It then defines the magnetic field above the Earth's surface in terms of a magnetic scalar potential V , such that:

$$\mathbf{B} = -\nabla V \tag{2.1}$$

In the same coordinate frame, V is approximate by a series expansion:

$$V(r, \theta, \phi, t) = a \sum_{n=1}^N \sum_{m=0}^n \left(\frac{a}{r}\right)^{n+1} \times [g_n^m(t) \cos(m\phi) + h_n^m(t) \sin(m\phi) P_n^m(\cos\theta)] \quad (2.2)$$

where $a = 6,371.2\text{km}$ (the geomagnetic conventional Earth's mean reference spherical radius), $P_n^m(\cos\theta)$ are the Schmidt quasi-normalized associated Legendre functions of degree n and order m and g_n^m and h_n^m are the Gauss coefficients in nanotesla (nT) of degree n and order m and are a function of time.

This series calculation will be computationally intensive for the on-board computer, and though this is the most reliable and accurate magnetic field model, if it is deemed too intensive in future iterations of this project, there are other options. The IGRF model has been implemented into code in many different programming languages, however, and is available online.

In [18], equation (2.2) is simplified. This is found by converting the Schmidt quasi-normalized Legendre functions to Gauss normalized functions. He makes this assumption because both the Gauss coefficients and the Legendre functions do not depend on position and so the calculations can be simplified by calculating them only once or at smaller intervals. This method can be used and researched further if the normal IGRF model is too computationally intensive.

If even further computational requirements are imposed on the ADC team, precomputed lookup tables can be used for the magnetic field reference vector.

2.2 Sun Vector Model

In order to model what the sun vector should be at any given point during the mission, a series of equations from the Astronomical Almanac will be used. These equations are supposed to give an accuracy within 0.01° , which is well below the attitude control requirement of less than 0.1° error. The equations are valid up through 2050, since the Earth's orbit around the sun is constantly changing due to gravitational forces between the Earth and the Sun and other disturbances. The derivation begins with equation (2.3), shown below:

$$n = JD - 2451545.0 \quad (2.3)$$

where n equals the number of days since J2000, or Greenwich noon on January 1st 2000, JD = Julian Date (or Day) of the desired time, or the continuous count of days since the beginning of the Julian Period used by astronomers. Once n is determined the mean longitude of the sun, L , the mean anomaly of the Earth in its orbit, denoted by g , and the obliquity of the ecliptic, ϵ can be calculated. These are given by the equations:

$$L = 280.460^\circ + 0.9856474^\circ * n \quad (2.4)$$

$$g = 357.528^\circ + 0.9856003^\circ * n \quad (2.5)$$

$$\epsilon = 23.439^\circ + 0.0000004^\circ * n \quad (2.6)$$

The mean longitude is corrected for the aberration of light, which is the viewing error caused by the

Earth's velocity relative to the sun. Using the mean anomaly, g , the distance from the Sun to the Earth, R , can be calculated:

$$R = 1.00014 - 0.01671\cos(g) - 0.00014\cos(2g) \quad (2.7)$$

The ecliptic longitude λ is given by:

$$\lambda = L + 1.915^\circ\sin(g) + 0.020^\circ\sin(2g) \quad (2.8)$$

and the ecliptic latitude β is neglected and assumed to be 0° . Finally, we can then calculate the x, y, and z coordinates of the sun in the right-handed rectangular equatorial frame. In this frame the x axis is in the direction of the vernal point, and the y axis is 90° to the east, in the plane of the celestial equator, and the z axis is directed toward the north celestial pole:

$$x = R\cos\lambda \quad (2.9)$$

$$y = R\cos\lambda\sin\lambda \quad (2.10)$$

$$z = R\cos\lambda\cos\lambda \quad (2.11)$$

This model makes use of relatively computationally simple calculations, compared to the series expansion in the IGRF model for instance, that should be well under any computational requirements imposed on the ADC team by the C&DH subsystem. Most of the equations are linear in nature as they are derived from a set of approximations and assumptions that simplify the overall calculation.

Chapter 3

Attitude Determination and Control

3.1 Subsystem Overview

The purpose of the Attitude Determination and Control (ADC) subsystem is to determine the orientation of a spacecraft and, if necessary, reposition the spacecraft to point in a desired direction. An ADCS may be used to ensure solar panels are always pointed towards the sun, or to reorient a spacecraft such that a scientific instrument will obtain more accurate readings.

The specific goal of the WPI ADC subsystem is to ensure that the CubeSat maintains a polar, sun-synchronous orbit with the face of the satellite containing the payload instrument pointing towards the sun with an accuracy of 1 to 2 degrees. Successfully following these requirements will ensure that the Sphinx-NG payload can take accurate measurements for the maximum amount of time. Unfortunately, despite the fact that our desired orbit is sun-synchronous, it is not guaranteed that we will achieve this exact orbit, as CubeSats are typically launched as secondary payloads. It is more likely that we will achieve a similar but not identical orbit to the one we desire. This means that realistically, we will have lighting periods and then periods of eclipse, where our ADC would go into a standby mode.

In most cases, an ADCS uses a combination of control and determination algorithms, sensors, and actuators to orient the spacecraft. The attitude determination and control of a satellite is generally broken down into three phases: detumbling, initial attitude determination, and attitude maintenance. Figure 3.1, created by the 2012 ADCS team [7], provides an overview of the three ADC phases:

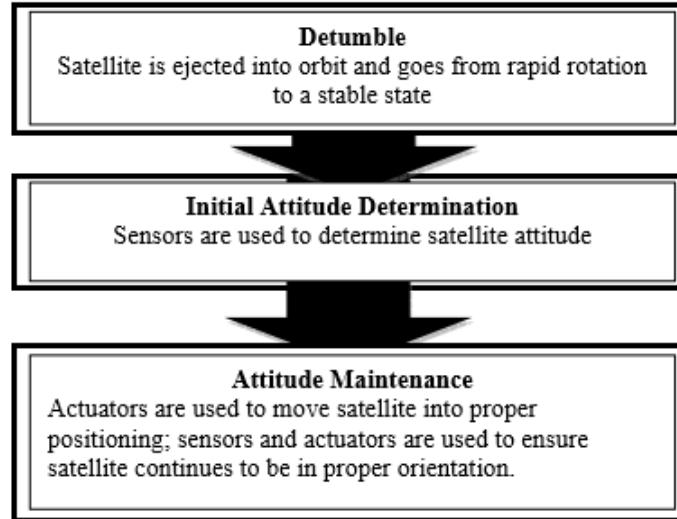


Figure 3.1: Mission phases of the ADCS.

Appropriate sensors and actuators were reviewed and chosen to reflect cost, power, and mission requirements. In addition, various spacecraft control and determination algorithms were researched to properly execute the three ADCS phases listed in Figure 3.1. The final selection of sensors, actuators, and algorithms have been detailed in the following sections.

3.2 Sensor and Actuator Selection

The hardware for the ADC subsystem was originally determined by the ADCS team in 2011 [12] and the magnetometer by the Instrument and Mission Analysis team in 2012 [3]. Since then, the ADC subsystem reports have provided an updated list of these sensors and actuators to account for new hardware models and updated specifications.

The following sections provide an overview of the sensors and actuators discussed in the previous CubeSat ADC reports and any hardware updates made during this project. Each section discusses the technical specifications of the respective sensor or actuator, how the data produced by each sensor will be conditioned (i.e. read by the on-board computer), and justification for any changes made to the previous ADC hardware list.

3.2.1 Fine Sun Sensor

As discussed in the mission statement, the WPI CubeSat requires a high degree of sun pointing accuracy such that the Sphinx-NG instrument can take proper X-ray measurements from the sun. This means that the sun sensor mounted on the satellite face containing the payload sensor must have a resolution greater than the required pointing accuracy. High resolution sun vector data will allow the ADC algorithms to make minute attitude adjustments to ensure that the ADCS satisfies mission requirements.

In 2013, the previous ADCS team chose a recently developed fine sun sensor designed specifically for use in a CubeSat. This specific sensor had been used in two CubeSat missions, Ukube-1 and TDS-1 [7].

The sensor is small, lightweight, and low power which is perfect for a CubeSat. The sensor produces four analog voltages that vary based on the incident angle of sunlight in both the horizontal and vertical directions. With a reported 0.5 degree accuracy, this sensor surpasses the 1-2 degree mission pointing requirement [7].

Since the 2013 ADCS report the sensor manufacturer, SSBV Aerospace, has closed. Fortunately the aerospace company NewSpace Systems (NSS) has a similar version of the sensor recommended by the previous ADCS team. The only physical difference between these two sensors is the required power, which increases to 10mA from 5mA [19]. Since the NSS CubeSat Sun Sensor is nearly identical to the SSBV CubeSat Fine Sun Sensor, the NSS model was deemed a suitable replacement. The sensor is pictured in Figure 3.2 below [20]:

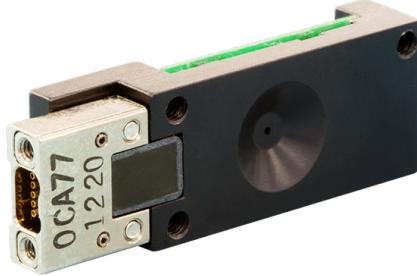


Figure 3.2: NSS CubeSat sun sensor.

As mentioned previously, the NSS sun sensor produces an analog voltage, which inherently requires data conditioning so the on-board computer can interpret the meaning of these voltages. According to the manufacturer, each sensor is provided with a calibration algorithm that calculates a sun vector from the four voltage measurements [20]. Since the manufacturer provides the necessary conversion algorithms, it is not required for ADCS teams to create any additional means of data conditioning. Table 3.1 below provides the technical specifications of both the NSS and SSBV (from report [10]) sun sensors [19]:

Parameter	SSBV Sensor	NSS Sensor
Mass	< 5g	< 5g
Power	< 5mA	< 10mA
Size	33 x 11 x 6mm	33 x 11 x 6mm
Field of View	120°	114°
Accuracy	0.5°	0.5°
Supply	3.3V, 5V	5V
Connector	9 way female Nano-D	9 way female Nano-D
Price	5,000.00USD	3,300.00USD

Table 3.1: Fine sun sensor technical specifications.

The technical data sheet for the NewSpace Systems CubeSat Sun Sensor can be found in the appendix

of this report. The technical data sheet for the SSBV model of this sun sensor appears in the 2013 ADCS report [10].

3.2.2 Coarse Sun Sensor

After the initial detumbling phase, the spacecraft may be oriented in such a way that the field of view of the fine sun sensor is out of range of the sun. If this occurs, the ADCS does not have adequate information to reorient the satellite to point at the sun. To prevent this situation additional smaller, lower accuracy sun sensors can be placed on the CubeSat faces that are not sun-pointing. Positioning a sun sensor on each satellite face will allow the ADCS to determine a general sun direction regardless of prior orientation.

The research for which less accurate, coarse sun sensor (CSS) should be used on the remaining four faces (one face contains the fine sun sensor and one face is inaccessible due to the Sphinx-NG instrument) was done by the ADCS team in 2012. Through extensive research of technical specifications and previously developed CubeSats, the team determined that the CSS-01,02 Coarse Sun Sensor was the best option due to its flight heritage and simplicity [7]. The CSS-01,02 is pictured in Figure 3.3 below [21]:



Figure 3.3: Space Micro CSS-01,02.

Originally developed by Comtech AeroAstro and recently by the company Space Micro, this coarse sun sensor design boasts 20+ years of flight heritage on multiple spacecraft, including ALEXIS, HETE, MOST, CHIPSat, and STPSAT-1. The CSS-01,02 is a passive sensor, requiring no power to operate. When in view of the sun, the CSS can produce up to 3.5mA (typical) of current. Due to the simplicity of the sensor design, the manufacturer states that the CSS-01,02 only has a ± 5 degree accuracy over a 120 degree circular FOV (over one axis) [21]. The low resolution on the CSS-01,02 is adequate since these sensors are on faces that are not sun-pointing. Sun sensors on non sun-pointing faces will be used to determine the general direction of the sun such that the ADCS can reorient the satellite so the fine sun sensor is pointing towards the sun.

Since the Space Micro CSS-01,02 produces an analog voltage signal and does not come with any software, additional data conditioning is required for these sensors. Coarse sun sensors estimate the angle of the sun by using an approximate cosine. Which means that ratio of the sensor output voltage,

V_{out} , over the output voltage at zero sun incidence angle, V_0 , is approximately equal to the cosine of the single axis sun angle, θ_{sun} . Mathematically stated:

$$\frac{V_{out}}{V_0} = \cos(\theta_{sun}) \quad (3.1)$$

In the equation above any $\frac{V_{out}}{V_0}$ produces both $\cos(\theta_{sun})$ and $\cos(-\theta_{sun})$ across the sensor FOV. To solve this issue the data from the CSS on the bottom face of the spacecraft will be used to determine whether the sun angle of incidence (AOI) is positive or negative. Since the bottom face of the spacecraft has a CSS and the top face does not (due to the payload instrument), the sign of the AOI will be determined by whether the sensor at the base shows a voltage output [10]. Table 3.2 provides the technical specifications of the Space Micro CSS-01, 02 [21]:

Parameter	CSS-01,02
Mass	10g
Power	None (Passive)
Sizing	
- Housing Diameter	1.270cm
- Flange Diameter	2.286cm
- Sensor Height	0.899cm
Field of View	120° (One Axis)
Accuracy	5°
Connector	Flying Leads
Price	2,600.00USD

Table 3.2: Space Micro CSS-01,02 technical specifications.

Additional information on this sensor can be found in the technical data sheet for the Space Micro CSS-01,02 Coarse Sun Sensor in the appendix of this report.

3.2.3 Magnetometer

The purpose of placing a magnetometer in a CubeSat is to measure the strength of Earth's magnetic field relative to the body frame of the spacecraft. Access to data on the Earth's magnetic field is integral to the functionality of the ADCS; this information will be used as part of the actuator control laws and the initial attitude determination algorithm (to be discussed in Section 2.3).

The magnetometer was originally chosen by the Instrument Mission Analysis team in report NAG-1102. The team researched magnetometers used in past CubeSat missions and chose the HMC5883L by Honeywell for three reasons; the sensor had past flight heritage, measured the magnetic field strength along all three body axes, and operated at 3.3 Volts (within bus capacity) [3]. To ensure that the HMC5883L was still the best option for a magnetometer the previous ADCS team analyzed the Freescale FCXOS8700CQ 6-axis Xtrinsic sensor, a competing combined magnetometer and accelerometer. The

team continued to recommend the HMC5883L after they determined that the Freescale sensor was heavier, less accurate, and required more power than the Honeywell magnetometer [10]. The Honeywell HMC5883L is pictured in Figure 3.4 [22]:

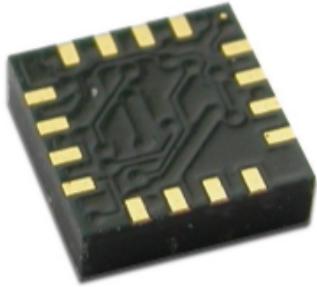


Figure 3.4: Honeywell HMC5883L magnetometer.

The data transmitted by the magnetometer is conditioned using a printed breakout board (PCB). The schematic for the PCB is provided in the HMC5883L technical documentation [22]. In addition, the magnetometer can be purchased with a pre-installed PCB from sites such as Adafruit or Sparkfun [23]. With the PCB the magnetometer can interface directly with the on-board computer through a dedicated connector, requiring no additional data conditioning. The technical specifications for the Honeywell HMC5883L are summarized in Table 3.3 below [22]:

Parameter	HMC5883L
Mass	18mg
Power	2.16-3.6V
Size	17.8 x 17.8 x 0.9mm
Linearity	$\pm 0.1\%$
Heading Accuracy	1-2°
Connector	Direct Interface
Price	10.00USD

Table 3.3: Honeywell HMC5883L technical specifications.

Additional information on the magnetometer can be found in the technical data sheet for the Honeywell HMC5883L in the appendix of this report.

3.2.4 Gyroscope

In a CubeSat ADCS angular velocity measurements from a gyroscope are used to help detumble, determine initial attitude, and maintain the desired attitude of the satellite. After launch into orbit, the gyroscope will be used to determine the initial spin of the satellite and to tell the ADCS when the spacecraft has stopped spinning. During sun-pointing periods data produced from the gyroscope will be used to detect any small rotations created by external torques which will then be utilized by the attitude

maintenance control algorithms to ensure the spacecraft continues to meet pointing requirements.

The previous ADCS team in 2013 recommended the ADXRS450 single-axis gyroscope by Analog Devices [10]. Since the previous report, Analog Devices has released an updated version of this sensor called the ADXRS453. Both the ADXRS450 and ADXRS453 are designed for high performance platform stabilization, which fits the requirements of the WPI CubeSat mission. The major difference between the two models is that the newer ADXRS453 takes more accurate reading in high vibration environments [24]. Since the gyroscope is a single-axis sensor, the ADXRS453 comes in two different configurations: a SOIC package for z axis (yaw) sensing and a vertical mount package designed for x axis and y axis (pitch and roll) sensing [25]. To measure the angular velocity about all three principal axes, two vertically mounted and one SOIC gyroscopes are required. The ADXRS453, mounted in both the vertical and SOIC configuration, is pictured below in Figure 3.5 [24]:

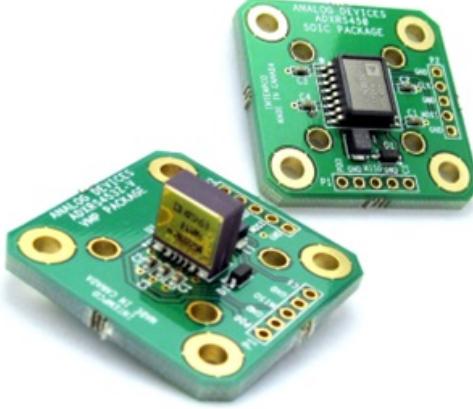


Figure 3.5: ADXRS453 and EVAL board in the vertical (left) and SOIC (right) configurations.

The data produced by the ADXRS453 is conditioned by the EVAL-ADXRS453Z breakout board. This breakout board can be purchased directly from Analog Devices and allows for easy connection to the satellite on-board computer [24]. The technical specifications of the ADXRS450 (from report [10]) and the ADXRS453 are summarized in Table 3.4 below [25]:

Parameter	ADXRS450	ADXRS453
Max Measurement	$\pm 300^\circ/\text{sec}$ - $\pm 400^\circ/\text{sec}$	$\pm 300^\circ/\text{sec}$ - $\pm 400^\circ/\text{sec}$
Sensitivity	80 LSB	80 LSB
Bandwidth	80Hz	77.5Hz
Supply	3.15V to 5.25V	3.15V to 5.25V
Connector	Direct Interface	Direct Interface
Price		
- Sensor	33.40USD	48.24USD
- EVAL Board	50.00USD	70.00USD

Table 3.4: Gyroscope technical specifications.

The technical data sheet for the Analog Devices ADXRS453 can be found in the appendix of this report. The technical data sheet for the Analog Devices ADXRS450 appears in the 2013 ADCS report [10].

3.2.5 Magnetic Torquer

The role of an actuator is to enact any control signal output by a controller algorithm. In other words, an actuator is the part of the ADC system that is responsible for physically reorienting the spacecraft. The type of actuator chosen for the WPI CubeSat ADCS was the magnetic torquer, due to its simplicity and reliability.

A magnetic torquer is an active device that creates a torque using an external magnetic field and a magnetic dipole moment. Unlike a momentum wheel or a reaction wheel this actuator has no moving parts and is less prone to malfunction. The construction of this device is fundamentally simple; a magnetic torquer consists of a metal rod wrapped in copper wire and is connected to a power source by two leads. Running a current through the wire induces a magnetic moment, denoted by μ , normal to the magnetic torque and in the presence of an external magnetic field, denoted by B , creates a torque perpendicular to the two vectors as given by the following relationship:

$$\tau = \mu \times B \quad (3.2)$$

Where μ consists of the magnetic moment components along each principal axis and B consists of the magnetic field components along each principal axis, written in the spacecraft body frame. To obtain complete three-axis control of the spacecraft three magnetic torque rods are required: one aligned with the body x axis, another aligned with the y axis, and the third aligned with the z axis.

In 2012 the ADCS team determined that the ZARM MTO-2.1, which produces a $0.2 Am^2$ dipole moment, created a sufficient amount of torque to overcome any external gravity gradients, solar radiation, and atmospheric drag [7]. Upon contacting the manufacturer, ZARM-Technik, the 2013 ADCS discovered that the company discontinued the MTO-2.1 model. Due to the lack of available COTS magnetic torque rods, the team decided to go with the $0.5 Am^2$ MTO-5.1 model. Even though the MTO-5.1 creates more torque, the magnetic torquer is longer, heavier, and consumes more power than the MTO-2.1. The CubeSat power system is able to support this increase in power demand [10]. The ZARM MTO-5.1 is pictured in Figure 3.6 below [26]:

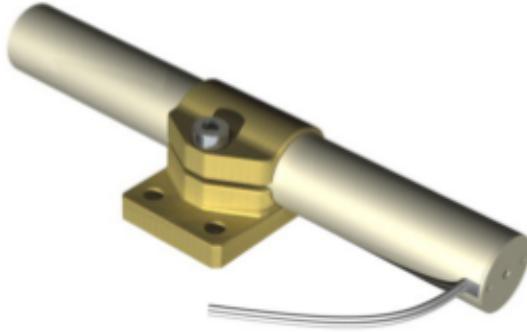


Figure 3.6: ZARM-Technik MTO-5.1 magnetic torquer.

According to the manufacturer, the MTO-5.1 is still in production. The technical specifications of the ZARM-Technik MTO-5.1 are summarized in Table 3.5 below [26]:

Parameter	MTO-5.1
Max Dipole Strength	$0.5 Am^2$
Mass	0.05kg
Max Power	0.3W
Max Current	60mA
Size	94mm (length), 12mm (radius)
Coil Resistance	83Ω
Connector	Flying Leads
Price	660.00USD

Table 3.5: ZARM-Technik MTO-5.1 technical specifications.

Additional information on the ZARM-Technik MTO-5.1 magnetic torquer can be found in the technical performance data sheet provided in the appendix of this report.

3.2.6 Global Positioning System (GPS)

In addition to the sensors and actuator detailed above, the ADCS also contains a global positioning system (GPS). Information provided by an on-board GPS receiver is used to pull data from sun and magnetic field reference models and used as part of the initial attitude determination calculation of the CubeSat after detumble.

The current GPS unit was originally researched and chosen by the Instrumentation subgroup in report NAG-1102. After comparing multiple GPS models the team decided on the SGR-05U from Surrey Satellite Technology LCC. The Surrey GPS is compact, lightweight, low power, and has flight heritage on five satellites. In addition, the SGR-05U was the only competitive GPS designed specifically for "small satellite" or CubeSat applications [3]. the Surrey SGR-05U is pictured in Figure 3.7 below [26]:

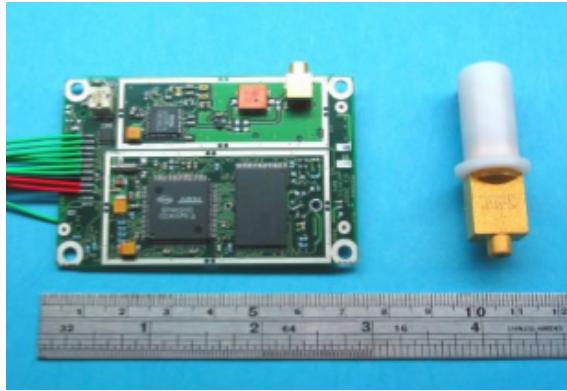


Figure 3.7: Surrey SGR-05U GPS.

The SGR-05U connects directly to the on-board computer through a serial data connector. Since the GPS receiver transmits a digital signal to the computer, no additional data conditioning is required. The technical specifications for the SGR-05U GPS are provided in Table 3.6 [27]:

Parameter	SGR-05U
Mass	20g
Power	0.8W
Size	70 x 45 x 10 mm
Accuracy	10m
First Fix	180sec (Cold), 90sec (Warm)
Connector	Serial
Price	18,000.00USD

Table 3.6: Surrey SGR-05U technical specifications.

Additional information on the Surrey SGR-05U GPS can be found in the technical data sheet provided in the appendix of this report.

3.3 ADC Algorithms and Methods

The following sections describe the attitude determination and control methods chosen to complete each of the three mission phases of the ADCS (seen in Figure 3.1). These sections justify each chosen method and provides a detailed description of each algorithm. The selection of ADC algorithms and methods was done by the original CubeSat ADCS team in 2011 [12]. The ADCS teams in 2012 and 2013 researched additional methods and improved the performance of these algorithms.

3.3.1 Spacecraft Detumble

When a CubeSat is initially sent out into orbit it will be spinning with a relatively high angular velocity and must be detumbled before the mission can begin. In general, the detumbling phase of a spacecraft is governed by a B-dot controller, derived using the Lyapunov function and Lyapunov stability analysis. The

B-dot control law commands a magnetic moment by utilizing the time derivative of the local geomagnetic field, expressed in spacecraft body frame coordinates. The resultant control torque produced by the three magnetic torquers will produce an angular velocity counteracting the spin of the spacecraft. The following equation is the vector representation of the B-dot control law where $\boldsymbol{\mu}$ is the magnetic dipole moment, \mathbf{B} is the Earth's magnetic field (\mathbf{b} is the unitized vector), and k is the controller gain:

$$\boldsymbol{\mu} = -\frac{k}{\|\mathbf{B}\|} \dot{\mathbf{b}} \quad (3.3)$$

As seen in the equation (3.3), the namesake of the B-dot controller comes from the time derivative of the magnetic field measurements provided by the magnetometer. The reason why this control law is useful for detumbling a spacecraft is given by first looking at the vector transport theorem:

$$\dot{\mathbf{V}}^B = \dot{\mathbf{R}}_A^B - \boldsymbol{\omega}_{BA}^B \times \mathbf{V}^B \quad (3.4)$$

This theorem states that the time derivative of the body frame is dependent on the relative rotation between the body and reference frames. Which means that the derivative of the Earth's magnetic field in the body frame can be written in the following form:

$$\dot{\mathbf{B}} = \mathbf{A} \dot{\mathbf{R}} - \boldsymbol{\omega} \times \mathbf{B} \quad (3.5)$$

The angular velocity in equation (3.5) is the relative velocity of the spacecraft relative to the Earth reference frame, i.e. the angular velocity of the spacecraft. This angular velocity is the term that must be driven to zero (or near zero) to detumble the spacecraft.

A common assumption made during B-dot controller design is that during the initial detumbling phase the angular velocity between the reference and body frames, given by $\boldsymbol{\omega} \times \mathbf{B}$ is much larger than the derivative of the Earth reference magnetic field vector and can be assumed to be zero for practical purposes. This means that if spacecraft angular velocity data is available (i.e. gyroscope sensor data) the B-dot control law can be rewritten as the following:

$$\boldsymbol{\mu} = \frac{k}{\|\mathbf{B}\|} \boldsymbol{\omega} \times \mathbf{b} \quad (3.6)$$

In some cases of the B-dot control law, the magnitude of the magnetic field vector \mathbf{B} is assumed constant and is factored into the gain k . For the k that appears in equation (3.6), a constant positive value can be chosen through simulation or an estimated value for k can be determined by evaluating the following expression:

$$k = \frac{4\pi}{T_{orb}} (1 + \sin(i_{orb})) J_{min} \quad (3.7)$$

Where T_{orb} is the orbital period, i_{orb} is the inclination angle of the orbit, and J_{min} is the minimum principal moment of inertia. This expression was derived by analyzing the closed loop dynamics of the component of angular velocity perpendicular to the magnetic field vector \mathbf{B} in [28].

As mentioned earlier, the stability of the B-dot controller can be proven by using Lyapunov stability analysis. Consider the following candidate Lyapunov function (V) and Lyapunov derivative where J is the moment of inertia matrix:

$$V = \frac{1}{2}\boldsymbol{\omega}^T J \boldsymbol{\omega} \quad (3.8)$$

$$\dot{V} = \boldsymbol{\omega}^T J \dot{\boldsymbol{\omega}} \quad (3.9)$$

To prove stability the derivative of the Lyapunov function must be negative definite, i.e. less than zero for any value of $\boldsymbol{\omega}$. To relate the B-dot controller to the Lyapunov function, the equation for torque and Euler's equation for rotational motion are required:

$$\boldsymbol{\tau} = \boldsymbol{\mu} \times \boldsymbol{B} \quad (3.10)$$

$$J \dot{\boldsymbol{\omega}} = -[\boldsymbol{\omega}^\times] J \boldsymbol{\omega} + \boldsymbol{\tau} \quad (3.11)$$

After combining the two above equations and substituting $\boldsymbol{\mu}$ for the B-dot control law given by equation (3.6), the Lyapunov derivative becomes:

$$\dot{V} = -\boldsymbol{\omega}^T (I_3 - \boldsymbol{b}\boldsymbol{b}^T) \boldsymbol{\omega} \quad (3.12)$$

Since the eigenvalues of $(I_3 - \boldsymbol{b}\boldsymbol{b}^T)$ must always be 0, 1, and 1, \dot{V} is negative semi-definite. This means that the B-dot control law is stable and will bring $\boldsymbol{\omega}$ to zero unless $\boldsymbol{\omega}$ is parallel to \boldsymbol{b} [28]. This is not a concern in practice and as such means that the B-dot controller is an adequate controller for the detumble phase of the spacecraft attitude control system.

Additionally, the B-dot control is often implemented as a bang-bang control law. A bang-bang controller (also known as an on-off or hysteresis controller) is a feedback controller that instead of providing a variable control output, switches between two states. In the case of detumbling a spacecraft using B-dot control, a bang-bang control law will not calculate a specific magnetic moment but rather calculate the sign of the spacecraft spin and apply the maximum allowed moment in the opposite direction. The bang-bang implementation of the B-dot controller is written as the following:

$$\mu_i = -\mu^{max} \text{sign}(\boldsymbol{u}_i \cdot \dot{\boldsymbol{B}}) \quad (3.13)$$

Where the μ^{max} is set to the maximum rated magnetic moment of each magnetic torquer and \boldsymbol{u} is the unit vector along which the magnetic moment is applied. When there are three magnetic torquers, one along each axis, the control law becomes:

$$\mu_x = -\mu^{max} \text{sign}(\boldsymbol{i} \cdot \dot{\boldsymbol{B}}) \quad (3.14)$$

$$\mu_y = -\mu^{max} \text{sign}(\mathbf{j} \cdot \dot{\mathbf{B}}) \quad (3.15)$$

$$\mu_z = -\mu^{max} \text{sign}(\mathbf{k} \cdot \dot{\mathbf{B}}) \quad (3.16)$$

Hysteresis control laws are generally used to minimize the time in which a control action is performed. Since these equations apply the maximum allowable moment, the detumble time is shortened but the total power required during this phase increases. If the power during the detumble phase is limited then the bang-bang control law cannot be used, or must be combined with the traditional B-dot control law.

In addition, another benefit of using the bang-bang implementation of the B-dot controller is that there is no need to calculate a gain or saturation condition. This means that the control law is less computationally intensive and easier to integrate into ADC algorithms than the traditional control B-dot control law.

3.3.2 Initial Attitude Determination

The previous projects settled on using TRIAD method in combination with noise removing methods to a) minimize the error of TRIAD and b) produce an optimal quaternion for initial attitude determination. More specifically, the projects outlined using the TRIAD method to produce a directional cosine matrix and then several more recent and proven methods to produce an accurate quaternion from that directional cosine matrix [10]. This project will maintain the same method because of its widespread use in many other CubeSats, picosatellites, and other spacecraft.

There are primarily two types of attitude determination; recursive and deterministic. Deterministic methods compare current readings to a reference reading and calculate an attitude based on the difference between the two. Recursive methods work by, essentially, comparing the current attitude to the most recent attitude. Assuming the current attitude quaternion is denoted by $\hat{\mathbf{q}}_n$, then a recursive method would compare the current estimate $\hat{\mathbf{q}}_n$ with $\hat{\mathbf{q}}_{n-1}$ and obtain an updated estimate.

3.3.3 The TRIAD Method

The TRIAD method is a deterministic method of procuring spacecraft attitude, which balances computational demand, hardware complexity, and accuracy. TRIAD uses two reference vectors instead of one (using one is more prevalent in attitude determination methods in larger spacecraft equipped with more accurate sensors), which over-estimates the attitude. This is done because CubeSats are usually equipped with commercial off the shelf (COTS) hardware to bring down costs, power usage, and computing needs. The result is an array of sensors that do not produce the highly accurate readings necessary to use determination methods that only require one reference vector. Using two unique observation vectors allows an over-estimation of the attitude and also simplifies the computing requirements.

TRIAD is credited to Harold Black in a 1964 paper published in [29] where he describes the method in detail. It is one of the oldest and most straightforward solutions to the attitude determination

problem for spacecraft. The method requires two sets of vectors; two observation vectors obtained by the magnetometer and the sun sensor, and two reference vectors obtained by using either a magnetometer reading or a geomagnetic model and a sun vector model to obtain vectors based on the satellite's position in the geodetic reference frame. The method in its most basic sense involves several vector and cross-product operations that give us the necessary directional cosine matrix to rotate from the body-fixed frame to the Earth-fixed inertial frame (Geodetic). TRIAD accounts for some possible sensor noise by normalizing the input vectors and using only the unit vectors in the mathematical operations [29]. Grace Wahba's later proposal of finding a rotation matrix that minimizes a least squares cost function will be looked at in order to minimize the error that results from sensor noise [30]. Then Davenport's q-method followed by QUEST (which came about later as a result of Davenport's q-method) and the Optimal Two Observation Quaternion Estimator Method (which was proposed much more recently, in 2002) will be studied for producing the solution to Wahba's problem; an optimal quaternion to estimate the spacecraft's attitude. Refer to Appendix A for more info on quaternions and their relevance to spacecraft attitude.

Begin with defining two linearly independent observation vectors \mathbf{b}_1 and \mathbf{b}_2 , corresponding to the observed vectors in the body frame, and two linearly independent reference vectors, \mathbf{r}_1 and \mathbf{r}_2 , corresponding to the inertial direction of references of the Earth's magnetic field and the Sun's direction from the Earth. Then attempt to produce a rotation matrix that allows the transformation between the two frames of reference, using these four vectors. In an effort to preserve the orthogonality condition of the direction cosine matrix and address the sensor noise, all vectors will be normalized and all work will be with unit vectors only [1]. So three rotation matrices A , A_1 , A_2 , can then be described such that $A = A_1 = A_2$, and:

$$A_1 \mathbf{r}_1 = \mathbf{b}_1 \quad (3.17)$$

$$A_2 \mathbf{b}_2 = \mathbf{r}_2 \quad (3.18)$$

This would in fact not hold true, because sensors have an uncertainty in their measurements that causes $A_1 \neq A_2$.

In the TRIAD method, the observation and reference vectors are taken and two triads created with them, M_{ref} and M_{obs} , that are composed of three orthogonal unit vectors.

$$M_{obs} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix} \quad (3.19)$$

where:

$$\mathbf{w}_1 = \mathbf{b}_1 \quad (3.20)$$

$$\mathbf{w}_2 = \mathbf{b}_x = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{|\mathbf{b}_1 \times \mathbf{b}_2|} \quad (3.21)$$

$$\mathbf{w}_3 = \mathbf{b}_1 \times \mathbf{b}_x \quad (3.22)$$

and:

$$M_{obs} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \quad (3.23)$$

where:

$$\mathbf{v}_1 = \mathbf{r}_1 \quad (3.24)$$

$$\mathbf{v}_2 = \mathbf{r}_\times = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|} \quad (3.25)$$

$$\mathbf{v}_3 = \mathbf{r}_1 \times \mathbf{r}_\times \quad (3.26)$$

Using these two triads rather than the vectors themselves, the same operation as before can be performed to obtain the direction cosine matrix:

$$A M_{ref} = M_{obs} \quad (3.27)$$

or simply:

$$\hat{A}_{TRIAD} = M_{obs} M_{ref}^T = \mathbf{b}_1 \mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_\times) (\mathbf{r}_1 \times \mathbf{r}_\times)^T + \mathbf{b}_\times \mathbf{r}_\times^T \quad (3.28)$$

This rotation matrix allows rotation from the satellite's body-fixed frame to the Earth-fixed inertial frame, and vice-versa. The TRIAD method on its own uses all of the information from the set of vectors labeled 1, while only partially using information from the set of vectors labeled 2 [1]. This essentially means that the method assumes the first vector to be entirely accurate, which is not the case, so unless the variances of the two readings are equal Wahba's problem must be used to minimize this error. For the simulations in this project however, the rotation matrix A was simply converted to a quaternion without using any of the noise reducing methods for obtaining an optimal quaternion. The only filtering was done through the extended Kalman filter.

3.3.4 Wahba's Problem

Grace Wahba attempted to address the previously described issue that arises when attempting to estimate spacecraft attitude with direction cosines of objects observed in a satellite fixed frame and of objects in a known reference frame [30]. In [30], Schuster describes "The Generalized Wahba Problem" as an extension of the original Wahba Problem.

If given two sets of n unit vectors $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ and $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$ where $n \geq 2$, find the rotation matrix A that brings the first set into the best least squares coincidence with the second. Or rather, find the matrix A that minimizes:

$$L(A) = \frac{1}{2} \sum_{i=1}^n \|\mathbf{b}_i - A\mathbf{r}_i\|^2 \quad (3.29)$$

If \mathbf{b} is a matrix of the first set of points \mathbf{b}_n and \mathbf{r} be a matrix of the second of points \mathbf{r}_n , also known as our reference and observation vectors. The formula above is the sum of the squares minimized. Further

expansion of $\|\mathbf{b}_i - A\mathbf{r}_i\|$ leads to:

$$\|\mathbf{b}_i - A\mathbf{r}_i\| = \|\mathbf{b}_i\|^2 + \|A\mathbf{r}_i\|^2 - 2\mathbf{b}_i \cdot (A\mathbf{r}_i) = 2 - 2\text{tr}(A\mathbf{r}_i\mathbf{b}_i^T) \quad (3.30)$$

where tr is the trace function and a T superscript denotes transposition. The last term is the only term dependent on A , so minimizing $L(A)$ is obtained by maximizing:

$$F(A) = \text{tr}(A\mathbf{r}_i\mathbf{b}_i^T) \quad (3.31)$$

Rewriting this in the context of this mission, with the two observation vectors and two reference vectors, the loss function below can be created:

$$L(A) = \frac{1}{2} \sum_{i=1}^2 a_i \|\mathbf{b}_i - A\mathbf{r}_i\|^2 \quad (3.32)$$

Where a_1 and a_2 are weights applied to the two sensor readings, with the subscript corresponding to which sensor reading is set as \mathbf{b}_1 and which sensor is set as \mathbf{b}_2 . The weights must be applied according to the accuracies of the sensors, with $a_i = \sigma_i^2$. This is done to relate Wahba's problem with Maximum Likelihood Estimation. There are several solutions that have been created for Wahba's problem such as Davenport's q method, Singular Value Decomposition, QUEST (QUaternion ESTimator) and ESOQ/ESOQ2 (ESTimators of the Optimal Quaternion). Davenport's q method and the SVD method have been found to be the most robust estimators, while QUEST and ESOQ are significantly faster. These methods will be discussed in detail along with what methods should be used in what specific scenario. First, however, rewrite the error as:

$$L(A) = \sum_{i=1}^2 a_i - \text{tr}(AB^T) \quad (3.33)$$

where:

$$B = \sum_{i=1}^2 a_i \mathbf{b}_i \mathbf{r}_i^T \quad (3.34)$$

Refer to [31] for a full proof of the following methods and all of the matrix operations involved in deriving them. Note that the methods described below were chosen for simulation because of the ease in setting them up in MATLAB®, as the main functions used are *eig* and *svd* in order to take eigenvalues and perform a singular value decomposition. Next, the recommended methods of cleaning the quaternion from solutions Wahba's problem will be discussed, based on decisions by the previous projects.

Davenport's q Method

Paul Davenport's solution to Wahba's problem was the first significant breakthrough in solving Wahba's problem for spacecraft attitude determination, although several other earlier solutions had already been

proposed by then. It essentially states that if \mathbf{q} is a unit quaternion such that:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \quad (3.35)$$

where:

$$\|\mathbf{q}\| = 1 \quad (3.36)$$

Then A can be parameterized by the equation:

$$A = (q_4^2 - |\mathbf{q}_{1:3}|) I + 2\mathbf{q}_{1:3}\mathbf{q}_{1:3}^T - 2q_4[\mathbf{q}_{1:3}^\times] \quad (3.37)$$

which is a homogeneous quadratic function of \mathbf{q} that can be rewritten as:

$$\text{tr } (AB^T) = \mathbf{q}^T K \mathbf{q} \quad (3.38)$$

K , a symmetric traceless matrix, is defined as:

$$K = \begin{bmatrix} S - I_3 \text{tr}B & \mathbf{z} \\ \mathbf{z}^T & \text{tr}B \end{bmatrix} \quad (3.39)$$

with:

$$S \equiv B + B^T \quad (3.40)$$

and:

$$\mathbf{z} = \begin{bmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{bmatrix} = \sum_{i=1}^2 a_i \mathbf{b}_i \times \mathbf{r}_i \quad (3.41)$$

Davenport then proves that the optimal unit quaternion is then the normalized eigenvector of K with the largest eigenvalue and given by the solution of:

$$K \hat{\mathbf{q}} = \lambda_{max} \hat{\mathbf{q}} \quad (3.42)$$

An issue arrives when the eigenvalues are symmetric, so look to QUEST and some more recent methods to address this problem. Refer to [31] for more information.

Singular Value Decomposition Method (SVD)

Singular value decomposition is a mathematical factorization of a matrix in linear algebra. Taking an $m \times n$ matrix, the singular value decomposition results in a factorization of the form $U\Sigma V^T$ where U is an $m \times m$ matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, denoted by $s_1, s_2, s_3 \dots s_n$ and V is an $n \times n$ matrix. A singular value decomposition of B

gives:

$$B = U \Sigma V^T = U \text{diag} \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} V^T \quad (3.43)$$

where U and V are orthogonal and the singular values obey the inequalities $s_1 \geq s_2 \geq s_3 \geq 0$. Recall that A is the rotation matrix obtained from TRIAD. Taking the trace of AB^T and maximizing it, assuming the $\det(A)=1$, this ends up as:

$$U^T \hat{A} V = \text{diag}([1 \ 1 \ (\det U)(\det V)]) \quad (3.44)$$

and rewriting this equation gives:

$$\hat{A} = U \text{diag}([1 \ 1 \ (\det U)(\det V)]) V^T \quad (3.45)$$

then defined:

$$s'_3 = s_3(\det U)(\det V) \quad (3.46)$$

so the attitude error covariance becomes:

$$P = U \text{diag}([(s_2 + s'_3)^{-1} \ (s'_3 + s_1)^{-1} \ (s_1 + s_2)^{-1}]) U^T \quad (3.47)$$

and the singularity condition occurs when $s_2 + s'_3 = 0$. The maximum eigenvalue of Davenport's K matrix is related to the singular values by:

$$\lambda_{max} = s_1 + s_2 + s'_3 \quad (3.48)$$

Quaternion Estimator (QUEST)

This method begins by rewriting equation (3.42) into two separate equations.

$$[(\lambda_{max} + \text{tr}B) I_3 - S] \hat{\mathbf{q}}_{1:3} = \hat{q}_4 \mathbf{z} \quad (3.49)$$

$$(\lambda_{max} - \text{tr}B) \hat{q}_4 - \hat{\mathbf{q}}_{1:3} \mathbf{z} = 0 \quad (3.50)$$

Rewriting $\hat{\mathbf{q}}$ using the classical adjoint representation gives:

$$\hat{\mathbf{q}}_{1:3} = \hat{q}_4 ((\lambda_{max} + \text{tr}B) I_3 - S)^{-1} \mathbf{z} = \hat{q}_4 (\text{adj}((\lambda_{max} + \text{tr}B) I_3 - S) \mathbf{z}) / \det((\lambda_{max} + \text{tr}B) I_3 - S) \quad (3.51)$$

Using the Cayley-Hamilton theorem for 3x3 matrices, the above equation can be rewritten in terms of the classical adjoint of G (refer to [31] for these steps). The optimal quaternion estimate given by QUEST is:

$$\hat{\mathbf{q}} = \alpha \begin{bmatrix} \text{adj}(\rho I_3 - S) \mathbf{z} \\ \det(\rho I_3 - S) \end{bmatrix} \rho = \lambda_{max} + \text{tr}B \quad (3.52)$$

where:

$$\rho = \lambda_{max} + \text{tr}B \quad (3.53)$$

The term α is determined through normalization of the resultant $\hat{\mathbf{q}}$.

Two-Observation Case

This case was studied because it applies specifically to the current mission since there are only two observation vectors measured. Many of the methods described in Markley and Mortari's report are capable of handling many more observation vectors, although that is unnecessary for this mission.

In this case of two observation vectors, the B matrix is of rank 2 and the determinant of B is 0. Therefore, the solution to the characteristic equation gives:

$$\lambda_{max} = \sqrt{a_1^2 + a_2^2 + 2a_1a_2[(\mathbf{b}_1 \cdot \mathbf{b}_2)(\mathbf{r}_1 \cdot \mathbf{r}_2) + |\mathbf{b}_1 \times \mathbf{b}_2| |\mathbf{r}_1 \times \mathbf{r}_2|]} \quad (3.54)$$

This solution would then be input into QUEST (refer to Section 3.3.4) to speed up the calculation for solving the characteristic equation. With it the optimal attitude estimate ends up as:

$$\hat{\mathbf{A}} = \mathbf{b}_3 \mathbf{r}_3^T + \left(\frac{a_1}{\lambda_{max}} \right) [\mathbf{b}_1 \mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_3)(\mathbf{r}_1 \times \mathbf{r}_3)^T] + \left(\frac{a_2}{\lambda_{max}} \right) [\mathbf{b}_2 \mathbf{r}_2^T + (\mathbf{b}_2 \times \mathbf{b}_3)(\mathbf{r}_2 \times \mathbf{r}_3)^T] \quad (3.55)$$

where:

$$\mathbf{b}_3 \equiv \frac{\mathbf{b}_1 \times \mathbf{b}_2}{|\mathbf{b}_1 \times \mathbf{b}_2|} \mathbf{r}_3 \equiv \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|} \quad (3.56)$$

Accuracy could be gained if the magnetic field of the satellite is thoroughly characterized and then magnetic field readings are corrected accordingly.

3.3.5 Attitude Maintenance

Once the spacecraft has finished detumbling and the initial attitude has been determined, the ADCS must reorient the satellite to the desired attitude, and then maintain that attitude throughout the orbit. For the attitude maintenance phase, the ADCS uses a combination of an Extended Kalman Filter (EKF) and a Proportional Derivative (PD) controller. The EKF filters noise from sensor measurements to create an accurate estimate of the spacecraft attitude. This information is used by the PD controller to determine, and correct, any error in the spacecraft's current attitude and the desired attitude. The functionality of the EKF and the PD controller are described below.

The sensors on-board the spacecraft are not perfect, as with any sensor. Which means that measurements of the spacecraft attitude have noise and errors. In a situation where a high degree of pointing accuracy or precise knowledge of the location of the spacecraft is required, removing noise is necessary to make accurate course corrections throughout the mission. A Kalman Filter creates a 'best estimate' of the state of the spacecraft by comparing the sensor measurements to a state estimate created by using previous system knowledge and a model created by using the system dynamics of the CubeSat.

A state vector consists of a set of parameters that together completely describe the state of a system. In this specific case, the rotation and orientation completely describe the state of a spacecraft. The rotation of a satellite is given by its angular velocity about the x, y, and z axes of the body frame. The orientation of the spacecraft is given by the quaternion. The angular velocity and unit quaternion of the spacecraft are written in the following forms:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.57)$$

and:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \quad (3.58)$$

where the first three \mathbf{q} values describe the vector orientation of the spacecraft and the fourth \mathbf{q} value is a scalar used to avoid singularities. Combining these two equations provides the complete state vector of the satellite system:

$$\mathbf{X} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3.59)$$

Recall that during its estimation phase the Kalman filter utilizes the system dynamics of the spacecraft, or in other words, the equations that govern how the state of the satellite changes over time. This means that the system dynamics equations of the satellite are given by taking the time derivative of the state vector. The derivative of the angular velocity vector, or angular acceleration, can be determined by using equation (3.60):

$$\boldsymbol{\alpha} = \dot{\omega}_x \mathbf{i} + \dot{\omega}_y \mathbf{j} + \dot{\omega}_z \mathbf{k} + (\boldsymbol{\Omega} \times \boldsymbol{\omega}) \quad (3.60)$$

Where the $\boldsymbol{\Omega}$ vector is the angular velocity of the moving body frame relative to the reference frame. If the moving body frame is rigidly attached to the spacecraft $\boldsymbol{\Omega}$ is equal to the angular velocity of the spacecraft, causing the cross product term to vanish. This implies that if the body frame stays constant, relative to the spacecraft, then the angular acceleration is found by taking the time derivative of each angular velocity component.

Since $\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$ are not available, an expression for the angular acceleration of the system can be derived by looking at the torque generated by a spacecraft. The net torque on a rotating spacecraft with

a rigidly attached body frame is characterized by Euler's equation:

$$\boldsymbol{\tau}_{ext} = \dot{\mathbf{H}}_{rel} + (\boldsymbol{\omega} \times \mathbf{H}) \quad (3.61)$$

Where \mathbf{H} is angular momentum vector and is related to the angular velocity of the spacecraft through the following expressions, where J_x , J_y , and J_z are the principal moments of inertia of the spacecraft:

$$\mathbf{H} = J_x \omega_x \mathbf{i} + J_y \omega_y \mathbf{j} + J_z \omega_z \mathbf{k} \quad (3.62)$$

and:

$$\dot{\mathbf{H}}_{rel} = J_x \dot{\omega}_x \mathbf{i} + J_y \dot{\omega}_y \mathbf{j} + J_z \dot{\omega}_z \mathbf{k} \quad (3.63)$$

Since the co-moving frame is rigidly attached to the spacecraft body, the equation for $\boldsymbol{\alpha}$ can be substituted in to the above equations for angular momentum:

$$\mathbf{H} = J\boldsymbol{\omega} \quad (3.64)$$

and:

$$\dot{\mathbf{H}}_{rel} = J\boldsymbol{\alpha} \quad (3.65)$$

Substituting equations (3.64) and (3.65) into Euler's equation of rotational motion provides the relationship between external torque (applied by the magnetic torquers) and angular velocity:

$$(\boldsymbol{\mu} \times \mathbf{B}) = J\boldsymbol{\alpha} + (\boldsymbol{\omega} \times J\boldsymbol{\omega}) \quad (3.66)$$

Solving for the angular velocity, $\boldsymbol{\alpha}$, provides an equation for the rate of change of the angular velocity of the spacecraft system and the first portion of the system dynamics:

$$\boldsymbol{\alpha} = J^{-1}[(\boldsymbol{\mu} \times \mathbf{B}) - (\boldsymbol{\omega} \times J\boldsymbol{\omega})] \quad (3.67)$$

The second portion of the system dynamics is given by the time derivative, or kinematics, of the unit quaternion. The equation for the quaternion kinematics is provided in [28], as the derivation is rather complex. This equation appears below:

$$\frac{d}{dt}(\mathbf{q}) = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \quad (3.68)$$

Where $\Omega(\boldsymbol{\omega})$ is the 4x4 cross matrix of the angular velocity about the x, y, and z axes of the spacecraft body frame:

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \quad (3.69)$$

Combining equations (3.67) and (3.68) provides the complete system dynamics, or the complete description of the rate of change of the spacecraft system:

$$\frac{d}{dt} \mathbf{X} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} J^{-1}[(\boldsymbol{\mu} \times \mathbf{B}) - (\boldsymbol{\omega} \times J\boldsymbol{\omega})] \\ \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \end{bmatrix} \quad (3.70)$$

The Kalman filter uses the system dynamics given by equation (3.70) to make a prediction of the updated state of the spacecraft's attitude. Since the system dynamics equations are non-linear, an extended Kalman Filter is required. Before computing the state estimate using the Kalman Filter algorithms, the system dynamics equations must be linearized. The process of linearizing these equations is detailed in the appendix of this report.

The Kalman Filter algorithms were originally designed to run in a continuous time setting. In practice, sensor measurements do not usually come continuously. To combat this issue additional Kalman Filter algorithms were developed. Two such algorithms are the Discrete Kalman Filter and the Continuous-Discrete Kalman Filter. The Discrete Kalman Filter uses the discretized system dynamics equations to perform a system update calculation once every time step. The Continuous-Discrete Kalman Filter uses the continuous system dynamics equations to propagate the system estimate in between time steps, calculating a final system update using new sensor measurements at the end of every time step. The process behind the Discrete and Continuous-Discrete Kalman Filters is discussed in the appendix.

The sensor measurements are used by the EKF to create the measured state of the spacecraft. For ease of calculation, the EKF created for the WPI CubeSat measures the state of the satellite directly, which means that $\boldsymbol{\omega}$ and \mathbf{q} are measured directly by the sensors. Since there is no sensor able to measure the quaternion of a system, a virtual quaternion is measured prior to each measurement update by using the magnetometer measurements, sun sensor measurements, and the TRIAD algorithm.

While the Kalman Filter can help detect an error, a controller is used to correct that error. The best controller for this situation would be a Proportional-Derivative (PD) controller. The PD controller corrects both the steady-state error, or orientation, and the rate of change of the error, or angular velocity, of the satellite. The spacecraft orientation, given by the quaternion, and the spacecraft angular velocity are readily available in the state estimate \mathbf{X} , so no additional calculations are required to implement this controller. This means that in this situation, the PD controller outperforms the P controller in terms

of accuracy and outperforms the PI and PID controllers in terms of computational effort. The general form of the PD controller for the CubeSat is:

$$\boldsymbol{\tau}_c = -k_p \boldsymbol{\delta}\hat{q}_{1:3} - k_d \hat{\boldsymbol{\omega}} \quad (3.71)$$

Where $\hat{\boldsymbol{\omega}}$ is the estimated angular velocity and $\boldsymbol{\delta}\hat{q}_{1:3}$ is the vector portion of the estimated error quaternion defined by $\boldsymbol{\delta}\hat{q} = \hat{q} \otimes q_{desired}^{-1}$. The estimated error quaternion and estimated angular velocity are determined by using the EKF and the desired orientation is given by the mission requirements. The proportional and derivative gain are determined through simulation and testing. The required $\boldsymbol{\mu}$ can be determined by expanding equation (3.2) and solving for μ_x , μ_y , and μ_z simultaneously.

Chapter 4

ADC Simulations and Results

4.1 Spacecraft Detumble Simulations

For the purpose of simulation, an orbit was chosen by the Orbital team and implemented in the orbital propagator package Satellite Tool Kit (STK). Using known models of Earth's magnetic field and sunlight, STK provided the data that the on board sensors would read. STK provided the magnetic field information using the IGRF model, which was used to simulate the magnetic field that the CubeSat's magnetometers would read. The sunlight that would hit the CubeSat while in orbit and eclipse times were used to simulate what the Cubesat's sun sensors would read. Other CubeSat missions were analyzed to find the initial conditions of angular rotations that the simulation would use to simulate the CubeSat's injection into orbit. These angular rotations averaged roughly 5 degrees per second about each axis and were used to test the B-dot controller. In general, the satellite detumble phase ends when the angular velocity about each axis reaches a magnitude of 0.1 degrees per second, this same condition was chosen for this simulation.

The optimization of the B-dot controller is paramount due to the necessity of the detumble phase. The length of time that it takes for the CubeSat to detumble affects how quickly the controller can change the satellite's orientation to the desired angles. The effectiveness of the utilized B-dot controller and its respective gain(s) affects the success of the mission and the resultant strains that the CubeSat will undergo. Therefore, the updated inertia matrix provided by the Cubesat's structural subsystem team was used to simulate the detumble with many different gains:

$$J = \begin{bmatrix} 0.030179 & -0.000020 & -0.003273 \\ -0.000020 & 0.030491 & 0.000407 \\ -0.003273 & 0.000407 & 0.005436 \end{bmatrix} kg - m^2 \quad (4.1)$$

4.1.1 Single-Fixed Gain

A single-fixed gain that applied to each axis in the B-dot control law was tested to find the most effective gain. Effectiveness was judged based on how quickly the Cubesat detumbled with respect to each axis.

The CubeSat's detumble was simulated using a B-dot Controller with gains ranging from -150000 to -50000. The results of this simulation of different gains appears in Figures 4.1:

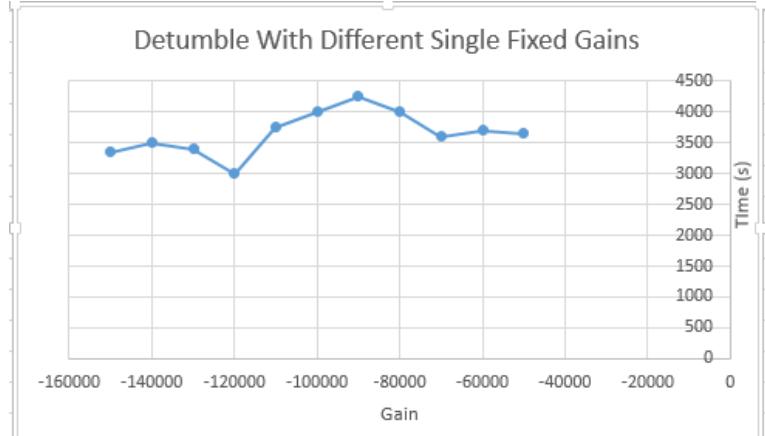


Figure 4.1: Detumble time versus fixed gain.

Through many hours of simulation, a gain of -120000 was found to be the most effective. The shortest detumble simulation using a single-fixed gain appears in Figures 4.2 and 4.3 below:

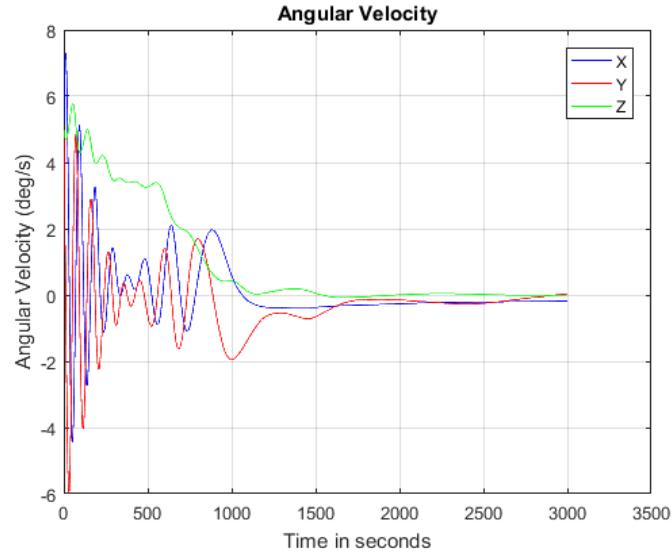


Figure 4.2: Detumble simulation with single-fixed gain in seconds.

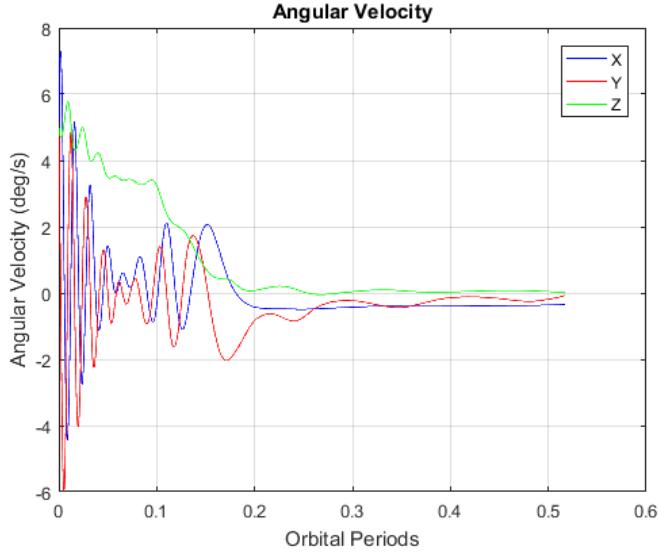


Figure 4.3: Detumble simulation with single-fixed gain in orbital periods.

Using a single-fixed gain the spacecraft stabilized in roughly 3000 seconds (0.5 Orbital Periods). The x and y axis rotations stabilized in 2000 seconds, while the z axis took significantly longer. The fixed gain simulations were performed using an initial angular velocity of 5 degrees per second about each axis. This initial rotation is typical of a CubeSat during initial orbit injection.

4.1.2 Adaptive Gain

An adaptive gain was theorized by the previous 2013 ADCS team [10] and also tested and compared to the aforementioned single-fixed gain. This involved a separate gain for each axis that changed proportionally to the magnetic field and angular rotations acting on the CubeSat over time. The gains are defined as follows:

$$C_x = -19500\left(5 - \frac{B_y}{0.00005} - \frac{B_z}{0.00005} - \frac{\omega_y}{5} - \frac{\omega_z}{5}\right) \quad (4.2)$$

$$C_y = -120000\left(5 - \frac{B_x}{0.00005} - \frac{B_z}{0.00005} - \frac{\omega_x}{5} - \frac{\omega_z}{5}\right) \quad (4.3)$$

$$C_z = -19500\left(5 - \frac{B_x}{0.00005} - \frac{B_y}{0.00005} - \frac{\omega_x}{5} - \frac{\omega_y}{5}\right) \quad (4.4)$$

The terms within the parenthesis of each equation change the gain over time. The value in front of the parenthesis of each equation was tested over many iterations and optimized to be the ones seen above. These values are dependent on the inertia matrix of the Cubesat. The detumble phase with the adaptive gain is shown in Figures 4.4 and 4.5 as follows:

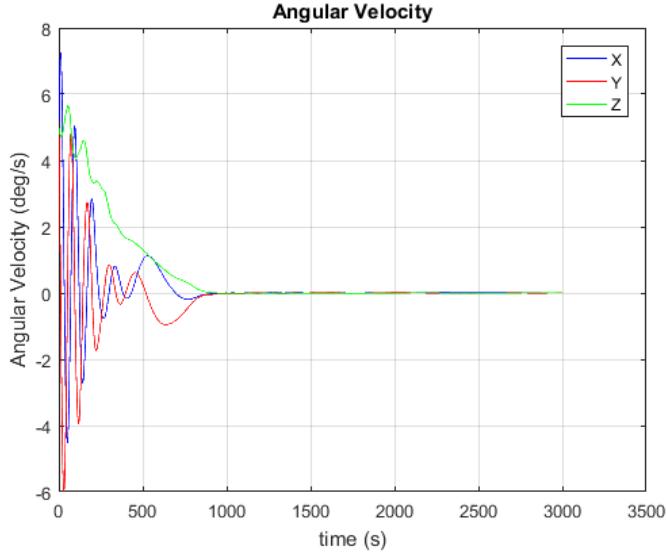


Figure 4.4: Detumble simulation with adaptive gain in seconds.

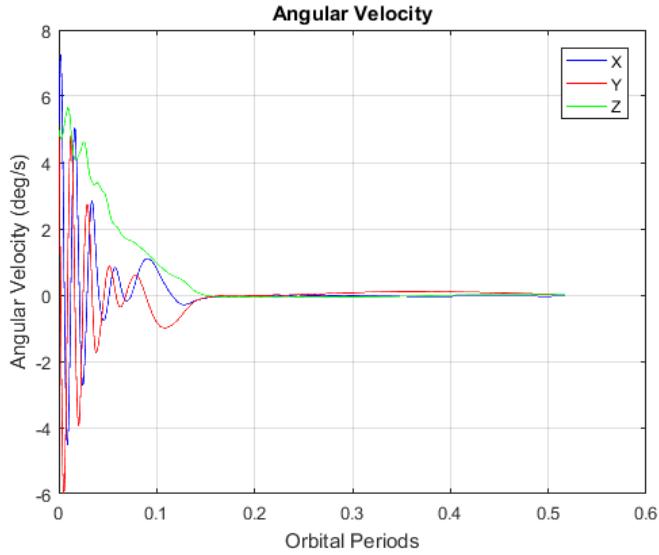


Figure 4.5: Detumble simulation with adaptive gain in orbital periods.

The addition of an adaptive gain to the B-dot control law provided a significant performance increase over using a single-fixed gain. The spacecraft detumble time was shortened to about 800 seconds (0.15 Orbital Periods), which is less than a third of the detumble time of the controller with a single-fixed gain. The adaptive gain simulations were performed using an initial angular velocity of 5 degrees per second about each axis. This initial rotation is typical of a CubeSat during initial orbit injection.

4.2 Attitude Determination and Maintenance Simulations

Two crucial components of the ADCS system are the static determination methods used to determine satellite attitude and the methods used to maintain an attitude. These methods are paramount to the success of the mission because they tell the satellite where it is and keeps it in the desired orientation. For

this reason, simulation was done to ensure that the estimation error of the attitude and determination methods were within the desired requirements for the mission. Simulation was done using the sensor standard deviations given in Table 4.1:

Sensor	Standard Deviation
Sun Sensor	0.005 deg
Magnetometer	250 nT
Gyroscope	0.00236 deg/sec

Table 4.1: Sensor data used in simulation.

4.2.1 Extended Kalman Filter Simulation

The Extended Kalman Filter is essential to the attitude maintenance of the Cubesat. Simulation was done to test its effectiveness in finding the Cubesat's orientation and attitude in orbit. This was very important, because the mission's high accuracy pointing requirements are directly dependent on the effectiveness of the EKF. Using MATLAB®, the accuracy of the Kalman Filter was characterized by using the result of detumble simulations. The angular velocity and rotation estimated by the EKF was simulated with the results shown in Figure 4.6:

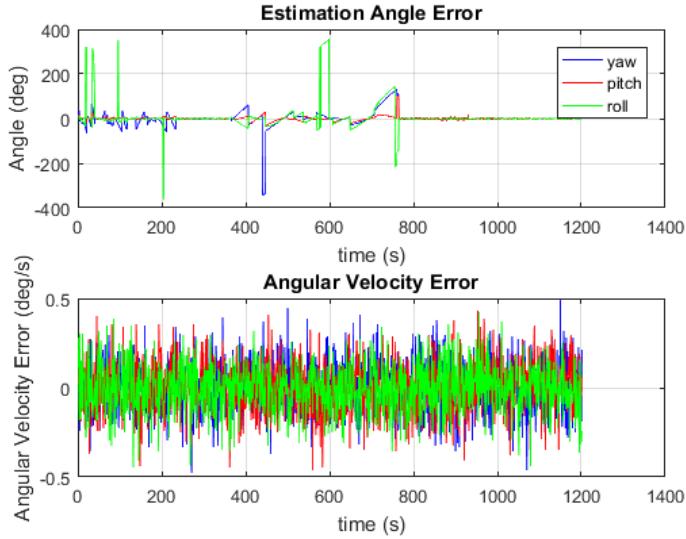


Figure 4.6: Simulation of EKF.

The simulation of the EKF found it to be relatively accurate. After the CubeSat is detumbled around 800 seconds, the TRIAD does an initial reading and then feeds information into the EKF to do recursive attitude maintenance. This is represented by the estimated angle error approaching zero in Figure 4.6.

4.2.2 Exact Angles Simulation

The exact angles of the CubeSat were calculated using Euler equations to find if the CubeSat would reach the desired body angles. This was done over a large period of time to simulate how the CubeSat

would maintain attitude while in orbit. The simulation is shown in Figure 4.7 below:

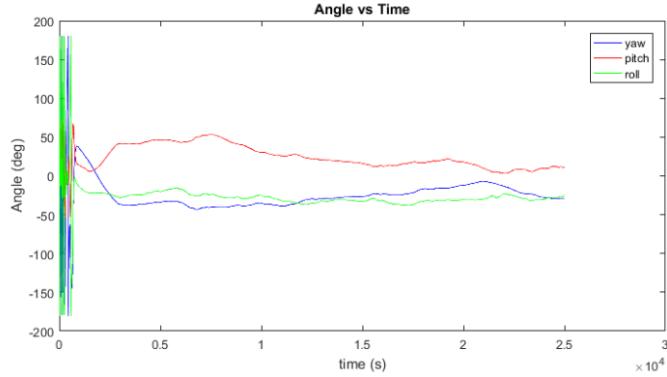


Figure 4.7: Simulation of exact angles.

After the B-dot controller stabilizes the spacecraft and the ADCS begins the attitude maintenance phase, the exact body angles do not approach the desired value of zero degrees over a reasonable period of time. This may be due to the new CubeSat configuration and resulting inertia matrix rendering the previous PD controller ineffective. To alleviate this issue re-optimization of the proportional gain k_p and the derivative gain k_d or an updated attitude maintenance control law is required. Eclipse periods during orbit also worsen this problem, since the ACDS is in standby during eclipse and disturbance torques are allowed to rotate the spacecraft freely. If an optimized gain for the current PD controller does not completely close the angle error seen in Figure 4.7, a controller for eclipse periods may be needed. Since the virtual quaternion measurements are not available during eclipse since TRIAD uses the sun vector, the control law for this potential controller must differ from the current proportional derivative design.

4.3 Power and Current Requirements

In addition to the simulating the performance of the attitude determination and attitude control algorithms for the three ADCS phases shown in Figure 3.1, the power consumption and current demand of the actuators was profiled. This information will be used, in part, to help characterize the CubeSat power subsystem requirements. Additional information on power and current calculations can be found in the appendix of this report.

4.3.1 Power Calculation Overview

Power is a very limited resource on a CubeSat. Using simulation results to estimate how much power the three magnetic torquers will consume during the detumbling phase and attitude maintenance phase is necessary for proper power subsystem management. Since the ZARM MT0.5-1 is rated for use in the linear magnetic region, power and magnetic moment are determined using the following relationships:

$$\mu = nIA \quad (4.5)$$

$$P = I^2 R \quad (4.6)$$

Where the constants n , R , and A are properties of the magnetic torquer rod as described in the table below. The notation described in Table 4.2 is consistent throughout the analysis.

Variable	Description
P	Power (W)
I	Current (A)
R	Coil Resistance (Ω)
μ	Magnetic Dipole Moment (Am^2)
n	Number of wire turns in coil
A	Area (m^2)

Table 4.2: Description of variables used during magnetic torquer analysis.

All constants values required to solve the equations for power (P) and dipole moment (μ) are provided in the ZARM MT0.5-1 magnetic torquer data sheet, excluding the number of wire turns of the copper wire coil (n) [26]. This value can be estimated by using the maximum values stated for magnetic moment and current for the magnetic torquer and solving equation (4.5) for n . Since the operating range of the magnetic torquer is linear and magnetic moment scales linearly with current, this value of n is an accurate estimate of the real number of turns in the coil.

Substituting equation (4.5) in to equation (4.6) provides the relationship between magnetic moment and power. This equation can be applied to the x, y, and z axis magnetic torquers:

$$\mu_x = \sqrt{\frac{P_x}{R}} n A \quad (4.7)$$

$$\mu_y = \sqrt{\frac{P_y}{R}} n A \quad (4.8)$$

$$\mu_z = \sqrt{\frac{P_z}{R}} n A \quad (4.9)$$

Solving for P_x , P_y , and P_z provides the instantaneous power consumed by each of the three magnetic torquers controlling the spacecraft:

$$P_x = \frac{\mu_x^2 R}{n^2 A^2} \quad (4.10)$$

$$P_y = \frac{\mu_y^2 R}{n^2 A^2} \quad (4.11)$$

$$P_z = \frac{\mu_z^2 R}{n^2 A^2} \quad (4.12)$$

Summing the instantaneous power calculated by using the above equations can be used to estimate the total amount of power each magnetic torquer consumes during detumble phase and how much average power is required during the attitude maintenance phase.

4.3.2 Current Calculation Overview

The magnetic torquer does not command a torque on its own; it creates a torque through the use of magnetic dipole moments. This means that while the final control output is a torque perpendicular to the magnetic moment and external magnetic field, the control is enacted by commanding a magnetic moment using a control law (In this project this was done by using a B-dot control law and a PD controller for detumble and attitude maintenance, respectively).

Since the magnetic torquer is a simple device with no internal computer, it is unable to use a commanded magnetic moment as a control signal. To induce the desired moment, the on-board computer must send an analog current signal to the magnetic torquer. The equations relating magnetic moment to current are derived below. Equations (4.13), (4.14), and (4.15) provide the relationship between magnetic moment and the current flowing through the wire wrapped around each magnetic torque rod:

$$\mu_x = nI_x A \quad (4.13)$$

$$\mu_y = nI_y A \quad (4.14)$$

$$\mu_z = nI_z A \quad (4.15)$$

Solving these equations for I_x , I_y , and I_z respectively, provides the expressions necessary to convert a desired magnetic moment into an equivalent analog current:

$$I_x = \frac{\mu_x}{nA} \quad (4.16)$$

$$I_y = \frac{\mu_y}{nA} \quad (4.17)$$

$$I_z = \frac{\mu_z}{nA} \quad (4.18)$$

This calculated current can be sent to the on-board computer to power the magnetic torquers and enact the desired magnetic dipole moment, which will result in the proper spacecraft attitude control.

4.3.3 Power and Current Profiles for Detumble Phase

Since the detumble phase of the spacecraft requires the highest power demand, the power and current requirements of this ADCS phase were profiled using newly implemented MATLAB® scripts. The power

and current profiles for the detumble Phase were simulated using both the optimized single-fixed gain and the adaptive gain. The current required by each magnetic torquer, for both types of gain, is shown in Figures 4.8 and 4.9:

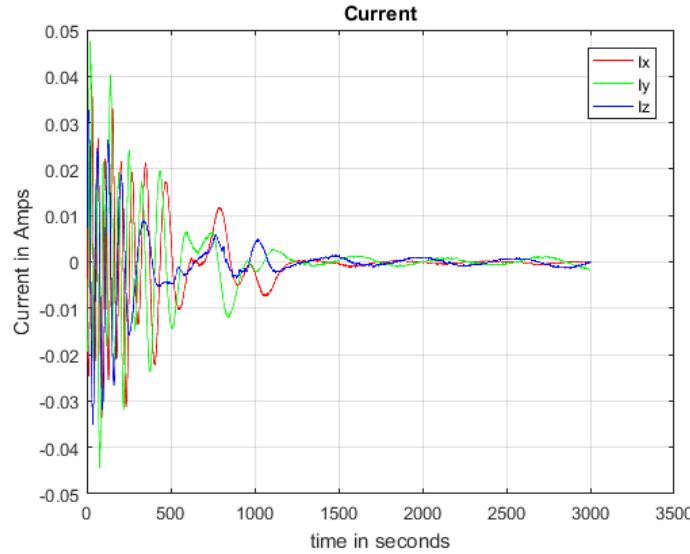


Figure 4.8: Current plot of fixed gain controller.

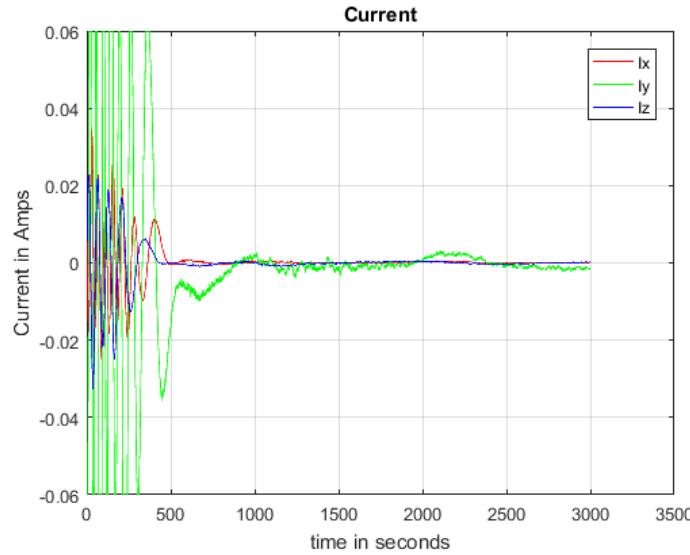


Figure 4.9: Current plot of adaptive gain controller.

For the simulation using a B-dot Controller with single-fixed gain, the current used by each magnetic torquer averaged roughly 0.025 Amps during the detumble phase. Using a B-dot Controller with adaptive gains, the current used by the magnetic torquers aligned along the x and z axes averaged roughly 0.0185 Amps. The current used by the magnetic torquer aligned along the y axis averaged roughly 0.06 Amps. The power consumed by each magnetic torquer during detumble phase, using both fixed and adaptive gain, is shown in Figures 4.10 and 4.11:

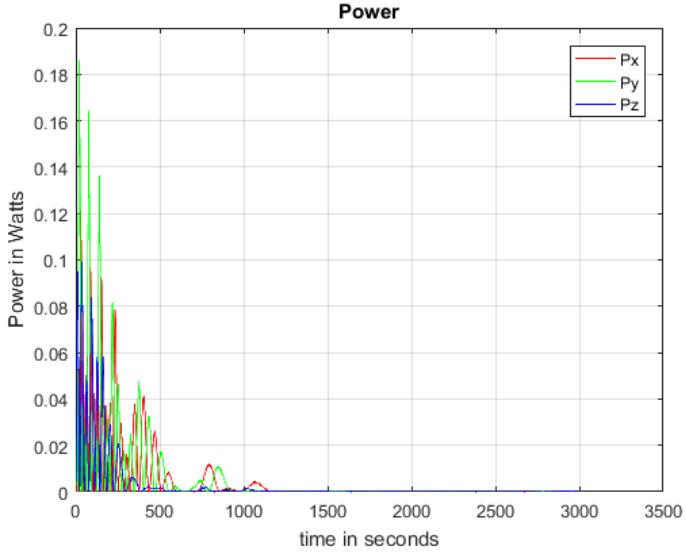


Figure 4.10: Power plot of fixed gain controller.

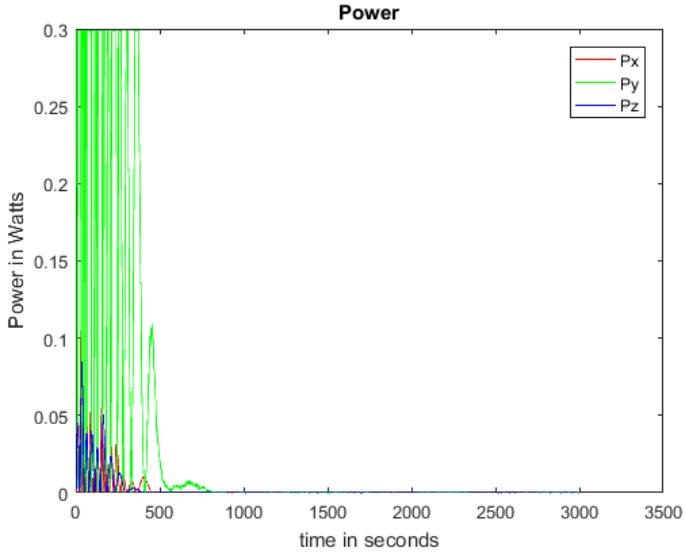


Figure 4.11: Power plot of adaptive gain controller.

For the simulation using a B-dot Controller with single-fixed gain, the maximum power used by the magnetic torquer was roughly 0.18 Watts during the detumble phase. Using a B-dot Controller with adaptive gains, the maximum power used by the magnetic torquers aligned along the x and z axes was roughly 0.05 Watts. The power used by the magnetic torquer aligned along the y axis was much higher, averaging roughly 0.3 Watts (the maximum allowable power). The heightened current and power usage of the y axis magnetic torquer utilizing the adaptive gain is due to the high multiplier used in the y axis proportional gain calculation.

The results of the simulation were reported to the Power Subsystem of the CubeSat mission. After communicating with the Power Subsystem, the total power usage was found to be well within the capability of the CubeSat's power supply. The power required, both computationally and electrically, did not affect the optimization of the CubeSat's computing and power subsystems. Therefore, the only

factors that needed to be considered for the gain(s) is how fast the gain can detumble the CubeSat. Therefore, the adaptive gain was found to be the best choice for the mission since it had a much faster detumble time than the single-fixed gain.

4.4 Systems Tool Kit (STK) Simulations

Systems Tool Kit, otherwise known as STK, is a modeling environment software that is used extensively by engineers, mission analysts, and software developers to model complex systems and visualize their dynamic data sets [32]. More specifically, it is used significantly in the aerospace industry to model satellite missions and has built-in features for a variety of subsystems, including ADC. This was the first year that WPI had licensing access to the software and so it was the first time the CubeSat project could make use of some of the tools STK had to offer. The primary goal was to set up a means of writing sets of MATLAB® code that could easily integrate into STK and be easily understood by anyone who picks up this project.

To run any sort of simulations, the STK MATLAB® Connector add-on that enabled connecting the installed version of MATLAB® to STK had to be set up. STK 11 has a built-in feature known as the “Attitude Simulator”, which allows one to choose Initialization, Simulation, and Post Processing scripts using MATLAB® and generate an attitude file in .a format. It also allows the specification of initial conditions for the quaternion and angular velocities. Communication with AGI allowed for a better understanding of how they perform their internal calculations, and whether they use Euler angles or quaternions to do them.

According to Nova Kazmi, systems engineer at STK, the software essentially does all attitude computations in quaternions. If you input Euler angles as the initial condition for the attitude simulator, then STK converts to quaternions and these values are integrated/calculated in quaternions. If you want a value at a specific time, then the time history will be used to interpolate data at the time you want (still calculating in quaternions). It then converts to the desired units.

The (*.a) format in STK is their attitude file format. It is an ASCII text file that has many formats, although the one we generally use for attitude control is the “AttitudeTimeQuatAngVels” format. It lists the values in the form:

```
<TimeInSeconds> <q1> <q2> <q3> <q4> <X> <Y> <Z>
```

where q1, q2, q3, and q4 are the four components of the quaternion and X, Y, and Z are the three angular velocities in their corresponding axes. Each row represents a new set of data at each time.

The way attitude control simulation was done was by writing MATLAB® code that influences the external torque on the satellite via variables that are recognized by STK. For instance, in the B-dot controller code that was written, the IGRF magnetic field vector was pulled directly from STK and used to calculate the required magnetic moment to detumble. After the simulation was run, it produced a .a file of the attitude during whatever time span the simulation was run for. This .a file can then be plugged back into the orbital attitude settings, which then visually models the control simulation in the STK scenario. Refer to Appendix H for more information on the filetypes.

For alignment settings in STK, the other subsystems were referred to using "Sun alignment with nadir constraint" where the satellite's X axis is aligned with the Sun direction and the Z axis is constrained in the direction of nadir [32]. This is the desired alignment of the CubeSat, ignoring the nadir constraint. This alignment is given in Figure 4.12:

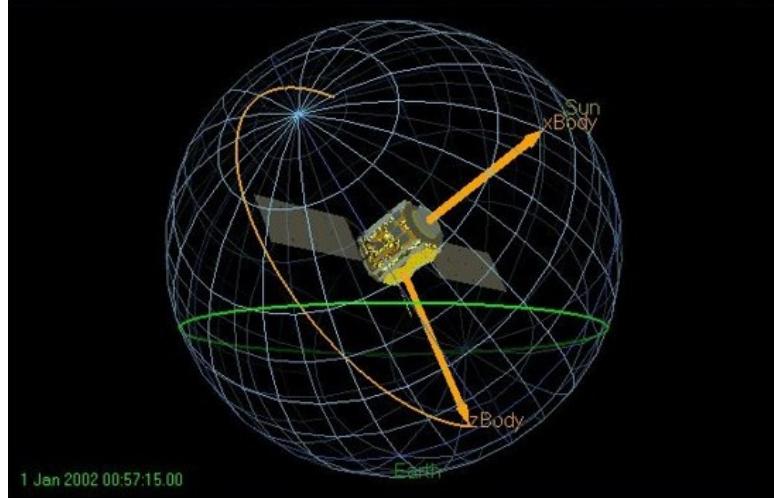


Figure 4.12: STK sun alignment with nadir constraint.

4.4.1 STK Attitude Coverage

Visual simulations of the attitude coverage in the 600km orbit were run. The initial orbital scenario was created by the Orbital Analysis subsystem, and this was expanded on to include the desired attitude features.

The first thing done was enabling the attitude sphere. The attitude sphere is STK's method of visualizing the satellite and all desired reference frames and tracking its attitude over time. By focusing on the satellite it makes it much clear to examine the orientation. The attitude sphere is shown in Figure 4.13 below:

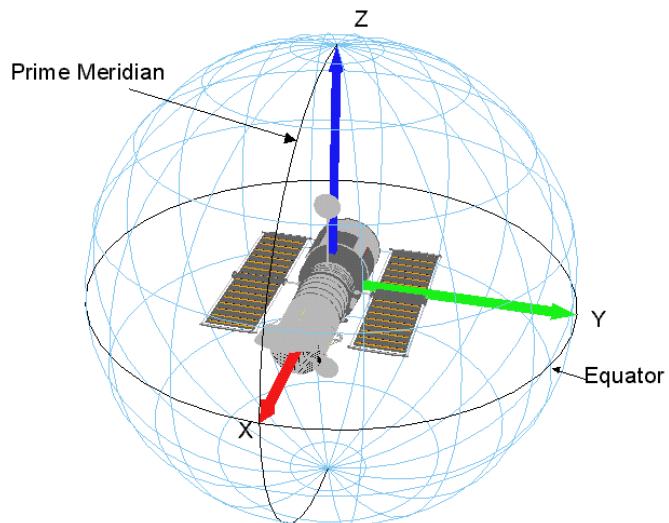


Figure 4.13: STK Attitude Sphere.

There are several parameters that can be adjusted in the attitude sphere, including colors, angle spacing, scale relative to model, and many others. See appendix H for detailed information on all the settings used to set this up.

After doing this, viewing of the sun vector relative to the spacecraft and the body axes of the spacecraft was enabled. This way the direction of the sun relative to our spacecraft coordinate system could clearly be seen. Then five sensors were placed on the spacecraft, excluding the Sphinx face, in the geometric center of each face. Table 4.3 provides the placement of each sun sensor and Figure 4.14 shows the resulting conical sun sensor FOV.

Sun Sensor Type	Direction	X (km)	Y (km)	Z (km)	Cone half angle (deg)
Fine	+X	+0.001	0	0	57
Coarse	-X	0	0	0	60
Coarse	+Y	+0.0005	+0.0005	0	60
Coarse	-Y	+0.0005	-0.0005	0	60
Coarse	-Z	+0.0005	0	-0.0015	60

Table 4.3: Sun sensor placement in STK.

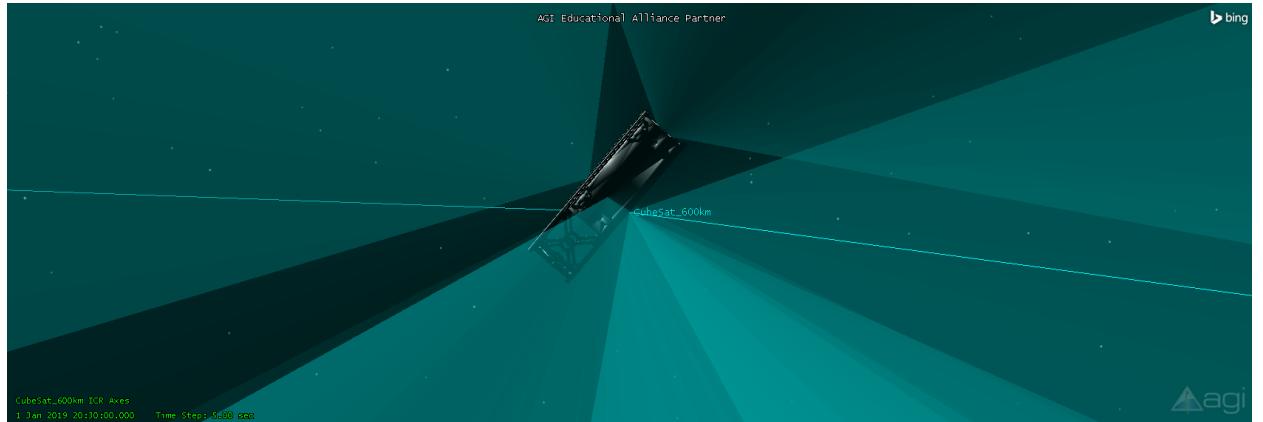


Figure 4.14: Sun sensor conical FOV in STK.

An Attitude Coverage was then set up on the satellite, which works in conjunction with the Attitude Sphere visual display in the 3D graphics window. What the feature does is allow a means of overlaying static or animated graphics onto the Attitude Sphere visualization. This is then linked to an Attitude Figures of Merit which is what mathematically analyzes the coverage. See Appendix H for detailed information on the setup. The primary asset was the Sun, and it was tied to the +X facing sun sensor for creating the visualization, seen in Figure 4.15.

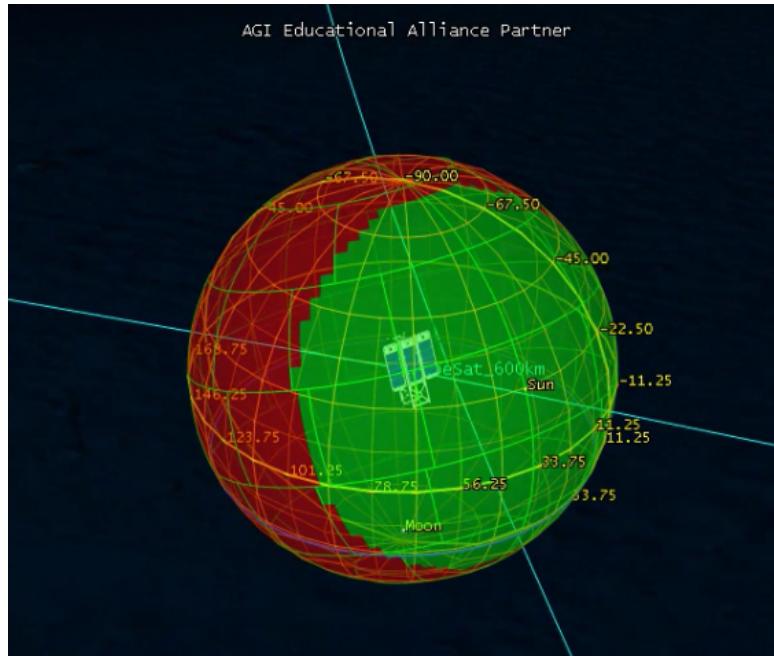


Figure 4.15: STK attitude coverage of the +X facing fine sun sensor.

In Figure 4.15, the static graphics are set on a colored ramp display system. So for values of $n = 0, 1, 2, 3, \dots$ different colors were set to each numerical value. The statistic analyzed is whether or not there is coverage. So for $n = 0$ there is no coverage, and the color is red, whereas for $n = 1$ and coverage, there is the color green.

4.4.2 STK Attitude Control Simulation

The MATLAB[®] Connector for STK was successfully installed on several of the machines in the Higgins Design Studio. This add-on was free and is the interface that enables communication between MATLAB and STK. The simulation process is shown in Figure 4.16:

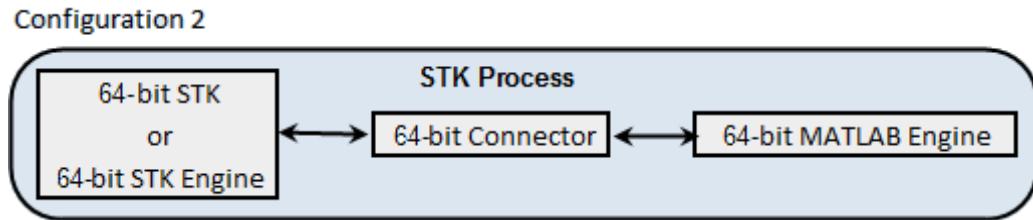


Figure 4.16: STK MATLAB simulation process, configuration 2.

Through work in conjunction with the Thermal subsystem the STK Space Environments and Effects Tool (SEET) was installed in order to ensure that STK was giving us a correct magnetic field reading. Part of this tool is the magnetic field component, which provides the total magnitude along the user-specified satellite path using a user-specified or default magnetic field model. According to STK, the component provides common magnetic field functionality with current AF-GEOSpace magnetic field models, including simple, tilted and offset dipole models based on time-interpolated moments of the full

IGRF field representation, full time-interpolated IGRF, and full IGRF plus Olson-Pfizer (1977) external field models [32].

A detumble script was written in MATLAB® meant specifically for pulling info from STK in order to simulating the first phase of our attitude control.

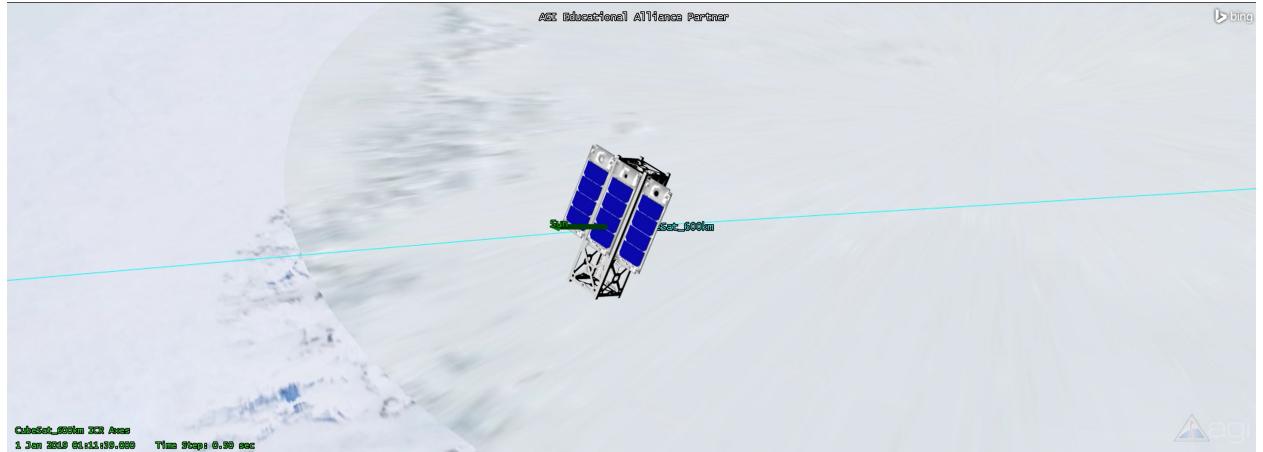


Figure 4.17: STK detumble simulation in 3D graphics window.

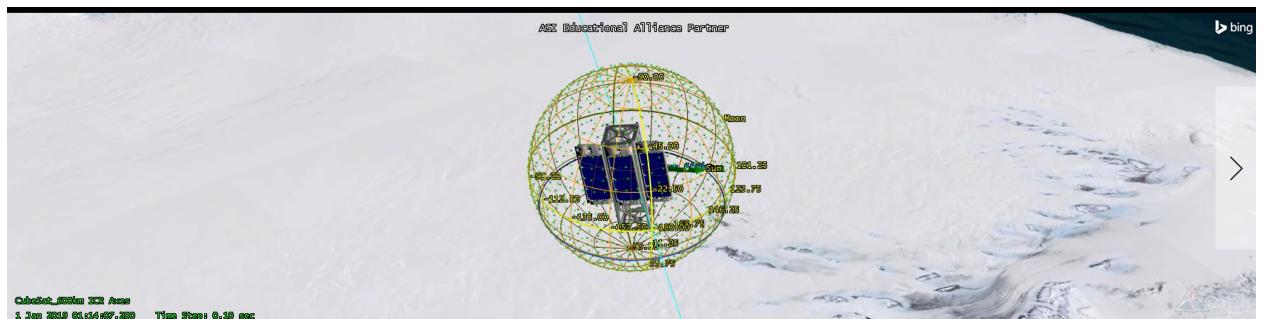


Figure 4.18: STK detumble simulation (with attitude sphere shown).

Figures 4.17 and 4.18 are screen shots of the STK graphics window during the detumble simulation. The first can be seen in orbit above Antarctica in the 600 km polar, sun-synchronous orbit without an attitude sphere graphic. The second image demonstrates the attitude sphere, showcasing both the spacecraft reference frame and the celestial reference frame. Refer to the appendix for a sample of the data output by STK.

Chapter 5

ADC Test Bed

5.1 Test Bed Design and Prototype Construction

The hardware-in-the-loop testing will be done on a test bed. Test beds are used commonly for physically testing ADC attitude control policies and their actual corresponding physical components. In its simplest form, a test bed is a combination of a spherical air bearing and a platform that combined can serve as a three degree-of-freedom system for both translational and rotational motion. A traditional bearing would allow for rotation, although there would be friction between it and the support structure it is placed on [33]. This is why a spherical air bearing is necessary as it allows for the system to be nearly frictionless. Many designs have been used in the past for various types of test beds, ranging from NASA's full scale satellite test beds to other university's CubeSat test beds. The tabletop design has been chosen for this mission, based on previous project's research and studies done during this project. Refer to [10] to view research done by the previous project on various test-bed types and their advantages. See Figure 5.1 from [34]:

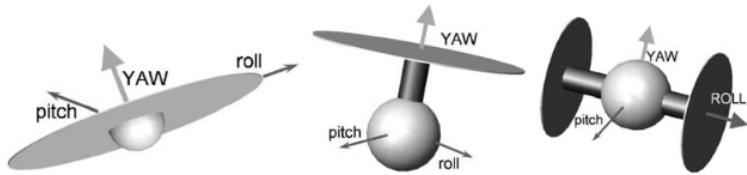


Figure 5.1: Three general types of ADC test beds.

The designs shown above are known more simply as, from left to right; the tabletop, umbrella, and dumbbell designs. They are designed according to the desired placement of components for aligning the center of mass with the center of rotation. This is a design consideration that depends on what type of satellite is being tested, and what type of system will be installed on the test bed platform. Alignment of the center of mass of the system is crucial in maintaining stability. Accurate balancing of the platform is necessary to duplicate a torque-free environment. Perfect balancing is obtained when the center of gravity coincides with the center of rotation of the platform. [35] The test bed will have various

sensors and actuators and circuit boards mounted on it, so it is necessary to design a counterweight system for adjusting the center of mass based on current configuration. Eventually the CubeSat itself will be mounted on the platform. In a simulator built by Tsiotras, Kriengsiri, Velenis, and Kim [35], an identification algorithm was developed to estimate the moment of inertia matrix and the center of gravity of the whole platform. This algorithm can also be used to estimate the inertia matrix for later use in attitude tracking and indirect adaptive control schemes.

The design chosen by the 2013 project was investigated again through research of two other CubeSat test beds. The test bed designs are shown below in Figure 5.2 below, taken from [10], [33], and [34]:

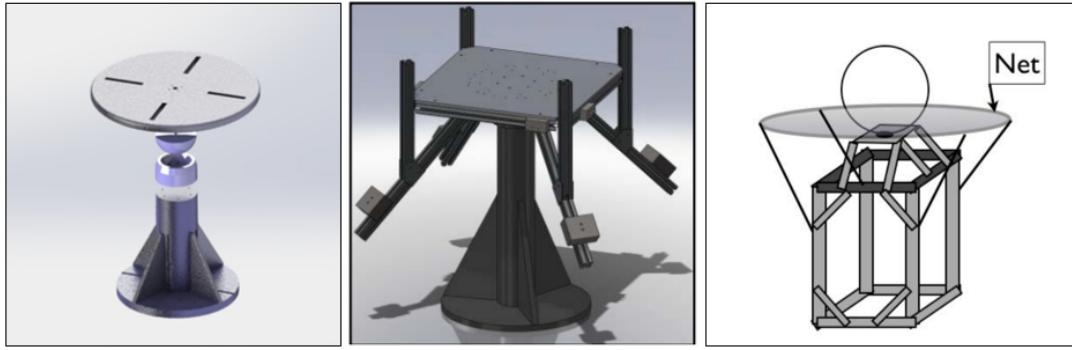


Figure 5.2: (left to right) WPI 2013 MQP Team, Hawaii Space Flight Lab, Naval Postgraduate School.

The WPI project in 2013 used a design with four milled slits in the platform meant for placing counterweights to balance center of mass along the x and y axes. The Hawaii Space Flight Laboratory (HSFL) design implemented a more complex mass balancing system that allowed for balancing the center of mass in the x, y, and z directions [33]. This is necessary because not only does it allow the platform space to be unrestricted by counterweights, but it allows lowering the center of mass to be at the center of rotation for a dynamically and statically stable system. See the Figure 5.3 below [33]:

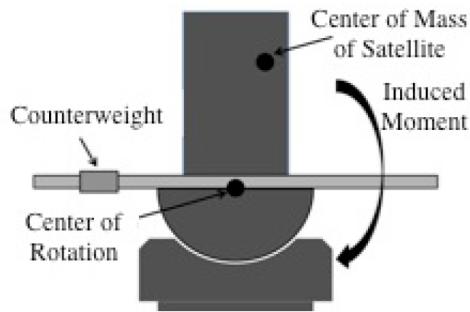


Figure 5.3: Induced moment when center of mass is not balanced.

Counterweights can be utilized to move the center of mass such that it is either at or below the center of rotation in the z direction, and at the center of rotation in the x-y plane. This is demonstrated in Figure 5.4:

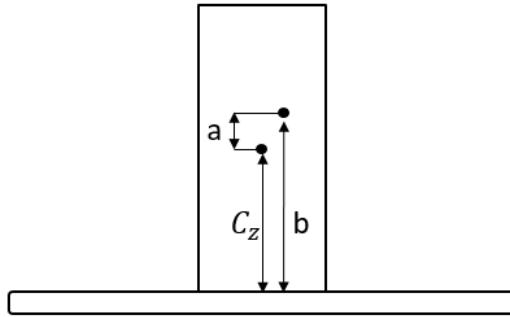


Figure 5.4: Center of mass balancing.

Let C_z be geometric center (subscript denotes z direction) and σ denote the maximum variance (in percentage) of center of mass from geometric center. According to P-POD requirements, the structural subsystem is required to keep the geometric center within twenty percent of the center of mass. Knowing mass, height, width, C_z location, and then calculating a , b , and c the system for center of mass balancing can be designed:

$$a = (C_z)(s) = 0.2(C_z) \quad (5.1)$$

$$b = a + C_z \quad (5.2)$$

$$c = 0.5(\text{width})(s) \quad (5.3)$$

The design by Meissner in [34] placed all testing components inside a hemisphere that sat on a spherical air bearing. It was found that the previous year's design was suitable as the support fin structure design was seen to have been used by both universities and companies alike, including in the HSFL design. The stress analysis performed in [10] demonstrated a structurally sound support design for the bearing that could withstand loads significantly higher than the actual test setup. The top platform was redesigned to be a chamfered square to make use of as much area as possible for mounting components and to be realistically machinable.

The previous project in 2013 completed a full CAD design of their proposed test bed design. They also performed a structural analysis of their design in Solidworks as stated above and proposed several electronic devices including a computer and some simple sensors to simulate the actual devices that will be on the spacecraft [10]. After validation of their design choice, their CAD design was modified into something that was machinable, through consultation with the machine shop. There were several issues with the design that arose during this process. First, problems arose with the spherical air bearing. As previously discussed, it is the center of the entire structure and is the most important component. Most test bed designs generally start with an air bearing and then everything else is constructed around it because it is by far the most mechanically complex component of test beds, and everything is attached to it. Without the spherical air bearing, the test bed fails to achieve its design purpose.

The previous project suggested the Specialty Components SRA250-R45 spherical air bearing. Specialty Components and Nelson Air Corp are two major suppliers of spherical air bearings for small

satellite applications. Both companies were contacted and two models were given that would suit the maximum CubeSat mass of 3 kg and a maximum test setup mass of about 30 kg. The SRA100-R45 is much smaller and cheaper than the SRA250-R45 recommended by the 2013 project and still supports more than enough load for the test requirements (69 lbs) [36]. Therefore, the SRA100-R45 is suggested as the best option for purchasing a spherical air bearing, alongside Nelson Air Corp's A-65X Series. The problem with purchasing these devices, however, is they are quite expensive. So, the idea of designing a custom air bearing was put forth during this project. Granted, these companies ensure certain high standards and high quality finish that cannot be replicated by students machining scrap metal, but a crude design was still built that allowed for some form of frictionless rotation.

A mount for the bearing was designed that was two parts. The bottom half consisted of an air inlet pipe leading into a pressure reservoir. It also had a groove for placing an O-ring in order to seal it with the top half afterwards, following. The top half had four exit pipes to create an even distribution of air pressure around the hemisphere and a spherical surface cut out of the cylindrical shape. The bearing itself is a sphere with the top portion machined off. The design was then reconstructed to revolve around this new bearing, including the platform and the support structure. Through consultation with the machine shop in Higgins and significant help from the machine shop contact James Loiselle, the following components were ordered for the initial build of the test bed. More features to machine will be discussed further in the recommendations, but some of these are; counterweight system, support fins, more precise air bearing. All parts are listed in Table 5.1 below:

Type	Component	Part Number	Description
Structural Components	Platform	MSC #02255875	1" thick, 18" x 18" aluminum 6061 plate
	Bearing	MSC #02629517	2.25" diameter, 12" long aluminum 6061 round rod
	Support cylinder	MSC #02629517	2.25" diameter, 12" long aluminum 6061 round rod
	Support fins	MSC #02255743	0.5" thick, 12"x 12" aluminum 6061 plate
	Base	McMaster-Carr #1610T77	14" diameter, 1" long aluminum 6061 disk
Attachment Components	Tube Fitting	MSC #84426121	0.25" OD, 0.125" NPTF thread push to connect metal tube fitting
	O-ring	See Marco Rubber. Number not available.	Size 223 O-ring, Viton® (fluorocarbon) material
	Screws	McMaster-Carr #90044A289	Alloy steel socket head screws, 2" length, 0.14" head diam.

Table 5.1: Parts ordered for initial construction of test bed.

First a simple inlet-outlet pipe for the air bearing was proposed as a crude initial design. After consultation with Professor John Blandino regarding a more sophisticated design, the result shown in Figure 5.5 was created:

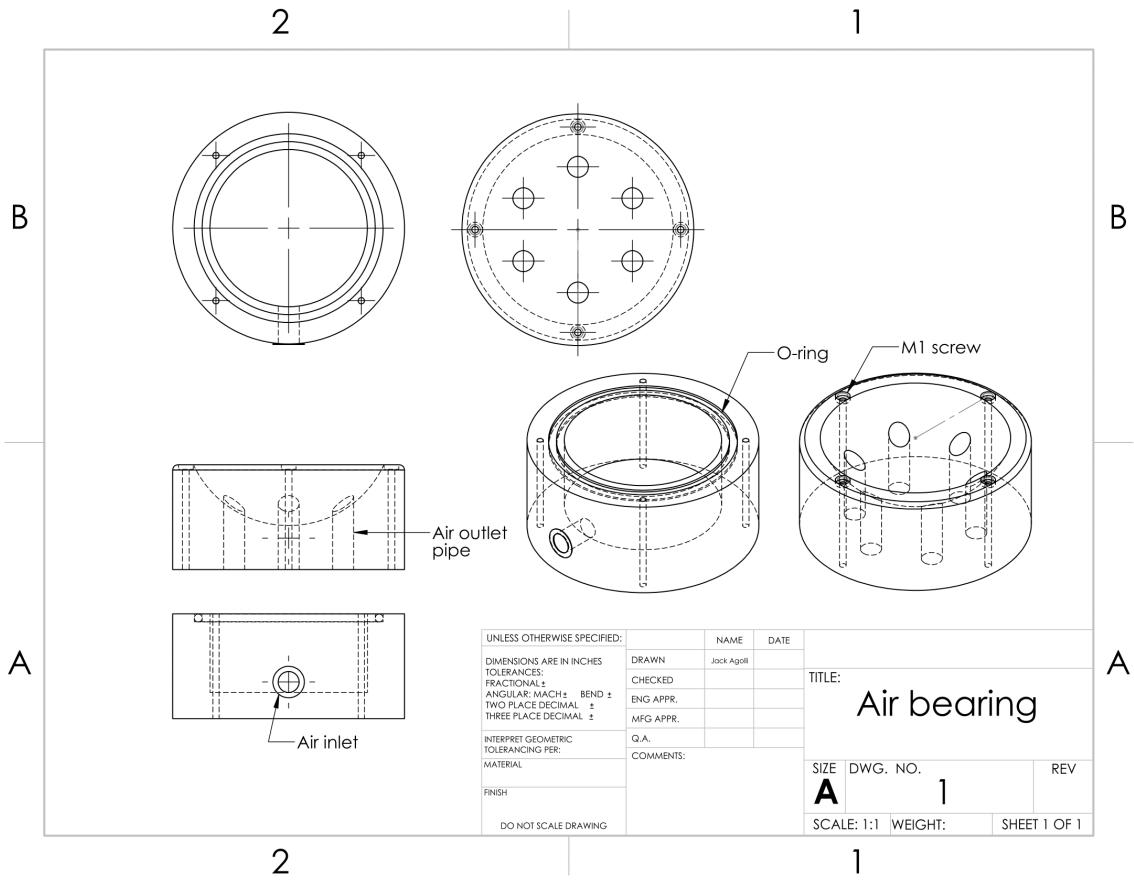


Figure 5.5: Spherical air bearing pressure chamber design.

This design consists of the entire bearing mount split into two halves. The top half contains the hemisphere for the sphere to sit, and the exit pipes. The bottom half contains the pressure chamber and the air inlet pipe, along with an O-ring groove. Both contain 2-56 counterbore screw holes. The idea was that using machine tools like mills, drills, and lathes, this was the most realistic way to create the desired air flow under the sphere. The size of the pipes was determined based on the principle of conservation of mass and the inlet air characteristics. The entire bearing is shown in Figure 5.6 below:

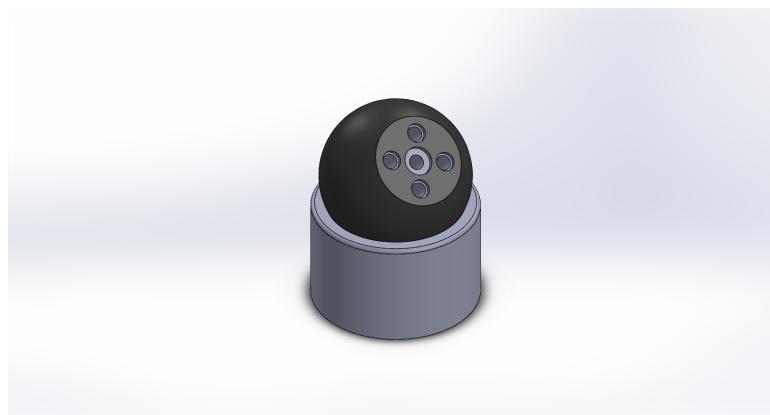


Figure 5.6: Spherical air bearing assembly.

The final, rendered test bed CAD design is shown in Figure 5.7. This design includes a model of the

WPI CubeSat for size comparison.

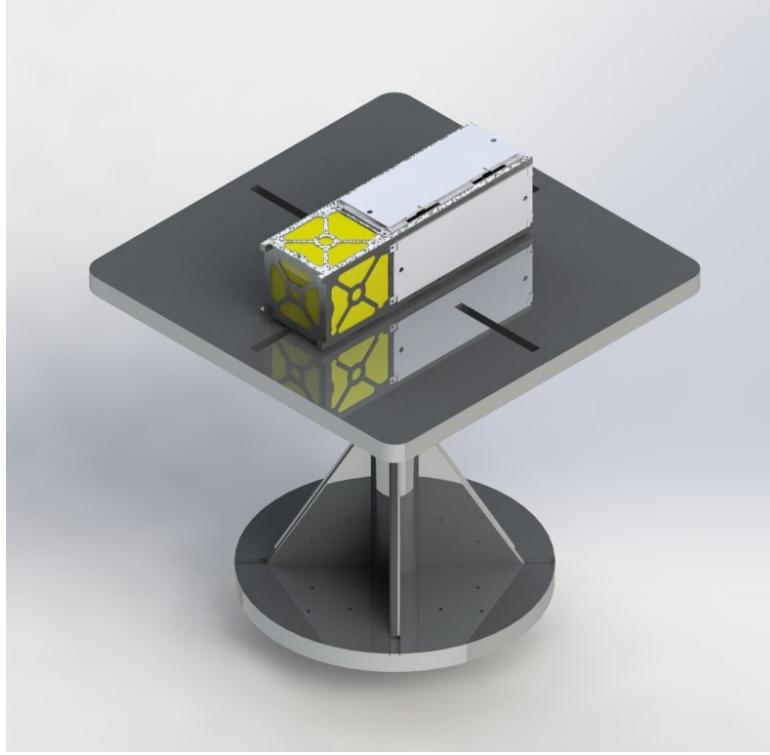


Figure 5.7: Fully rendered test bed design.

The Solidworks models were loaded in ESPRIT and then the machining operations were set up for use in the CNC. A range of drills, lathes, and end mills were used to cut aluminum stock and after various designs and tests, the first prototype for WPI's CubeSat test bed was built in a relatively short time. There are several adjustments and additions that need to be made, along with future improvements, and these will be described in the Recommendations section.

5.2 Prototype Testing

To test and demonstrate the functionality of the prototype test-bed, equipment for a simple experiment was purchased. The goal of this experiment is to use the prototype test-bed to obtain angular velocity readings from a 3D printed CubeSat model. An IMU, a microprocessor board with WiFi capability, and a battery are required for this experiment, since wires will interfere with the rotation of the platform. The Adafruit 9-DOF Fusion was chosen as the IMU. The IMU is pictured in Figure 5.8 [37]:

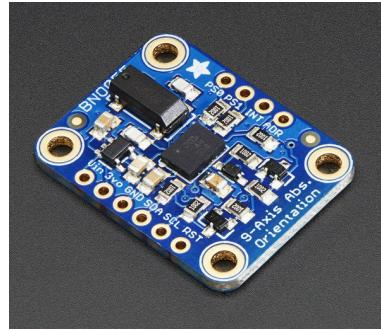


Figure 5.8: Adafruit 9-DOF Fusion IMU Breakout.

The microprocessor for this experiment is the Adafruit Feather HUZZAH with ESP8266 WiFi. The board, pre-installed with stacking headers, is shown below in Figure 5.9 [38]:

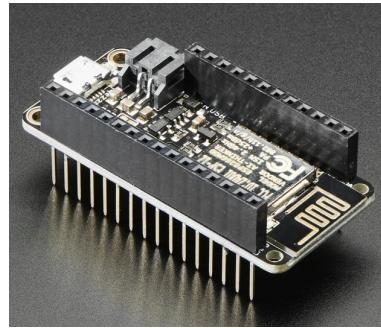


Figure 5.9: Adafruit Feather HUZZAH with ESP8266 WiFi.

To power the microprocessor the EFOSHM Ultra Slim battery was chosen. This battery is small, thin, and light which will allow it to fit into the CubeSat model. This battery is pictured in Figure 5.10 [39]:



Figure 5.10: EFOSHM Ultra Slim Battery.

The Feather HUZZAH microprocessor is programmed using the Arduino suite. The IMU will be directly connected to the Feather HUZZAH using the pre-installed headers which will wirelessly transmit

the measured angular velocity data to a nearby computer. One downside to the Feather HUZZAH is that the microprocessor only has enough computing power to process raw data. To implement an EKF in this experiment and filter sensor noise, a more powerful computer must be used. For this application a Raspberry Pi 3 was purchased, seen in Figure 5.11 [40]:



Figure 5.11: Raspberry Pi 3.

The Raspberry Pi has enough computing power to implement an EKF and has built-in WiFi capability, but is much larger than the Feather HUZZAH microprocessor and harder to program. Either processor has the capability to demonstrate the functionality of the test-bed. The equipment required to test the ADCS hardware described in Chapter 3 is outlined in [10].

Chapter 6

Conclusions and Recommendations

6.1 Project Summary

The goal of the WPI CubeSat mission is to place a 3U CubeSat into a sun-synchronous 600km polar orbit for space weather observation, which will primarily be done by solar and terrestrial X-ray spectroscopy. The scientific goal will be achieved by the Sphinx-NG, an instrument in development at the Space Research Centre, a research institute in the Polish Academy of Sciences with a primary goal of researching terrestrial space. To ensure accurate measurements, the Sphinx-NG requires a one to two degree sun pointing accuracy during the lighting period of the satellite orbit.

The purpose of this project was to continue developing and testing an attitude determination and control subsystem that is capable of properly orienting the spacecraft towards the sun and maintaining the required pointing accuracy for the duration of the mission. The following sections summarize the development of the ADC subsystem and ADC test bed. In addition, these sections highlight the improvements made to the existing ADCS during this project.

6.1.1 ADC Subsystem Overview

As discussed in Chapter 3, the attitude determination and control subsystem consists of a combination of sensors, actuators, and algorithms. Sensors are used to gather relevant data about the current state of the spacecraft. These data are then used by the ADC algorithms determine the attitude of the satellite, decide whether it must be reoriented, and calculate the necessary control inputs. Finally, the actuators use the control input calculated by the ADC algorithms to create a torque and rotate the spacecraft to the desired orientation.

The ADCS is broken down into three phases; detumble, initial attitude determination, and attitude maintenance. When the spacecraft is initially dropped into orbit it is, in general, spinning with a very high angular velocity. Before the mission begins the satellite must be detumbled. Once the satellite has stopped spinning, the initial attitude of the spacecraft is determined and the spacecraft is reoriented to face the sun. After the spacecraft has stopped spinning and has reoriented to face the sun, the desired attitude is maintained by filtering out sensor noise and making small corrections for errors.

Hardware Selection

Due to the high required pointing accuracy of the mission, the selection of sensors and actuators for the satellite was considered carefully. Each sensor was chosen based on previous use in CubeSat missions, on the space flight heritage, and on how well the sensor or actuator can assist in completing each of the three ADCS phases. The sun sensor and magnetometer are able to provide information on the sun direction and strength of the Earth's magnetic field relative to the spacecraft, which provides an adequate amount of data to determine the satellite's attitude. In addition, the GPS provides the satellite's current location in orbit. The gyroscope is provides the angular velocity of the spacecraft, which is used to detumble the satellite and correct any attitude errors. In addition, the GPS provides the satellite's current location in orbit. Three orthogonally placed magnetic torquers allow for complete three-axis control of the satellite to enact any necessary attitude adjustments. Table 6.1 below summarizes the relevant sensors and actuators:

Sensor/Actuator	Model	Manufacturer	No.	Total Cost	Power Required
Coarse Sun Sensor	CSS-01,02	Space Micro	4	10,400.00USD	NA
Fine Sun Sensor	CubeSat	NewSpace	1	3,300.00USD	0.280W
Magnetometer	HMC5883L	Honeywell	1	10.00USD	0.030W
Gyroscope	ADXRS453	Analog Devices	3	250.20USD	0.09W
GPS	SGR-05U	Surrey	1	18,000.00USD	0.800W
Magnetic Torquer	MTO-5.1	ZARM-Technik	3	1,980.00USD	0.900W

Table 6.1: Overview of ADCS hardware.

Due to the extensive research done in the previous ADCS projects, many of the sensors chosen by past ADCS teams remained unchanged. The only changes between the sensors and actuators selected by the 2013 project and the sensors and actuators selected for this project are the fine sun sensor and gyroscope. The fine sun sensor due to a change in manufacturer and the gyroscope was changed to an updated model.

ADC Algorithms

After the CubeSat is ejected from the P-POD into orbit it will have some random angular velocity, estimated to be around five to seven degrees per second. Prior to determining the CubeSat's attitude, the satellite will be detumbled using a B-dot stabilizing controller. The purpose of a B-dot controller is to bring the angular velocity of a rotating spacecraft to zero. The control law uses measurements of the Earth's magnetic field from the magnetometer and measurements of the angular velocity from the gyroscope to calculate a magnetic dipole moment vector. The magnetic dipole moment along each axis is then commanded to each of the three respective orthogonal magnetic torquers, generating a torque that counteracts the spin of the spacecraft.

Since the satellite has stopped spinning a static determination method can be used to determine the attitude of the spacecraft. The WPI ADCS uses the TRIAD method to calculate the current attitude

matrix of the satellite, which is then converted to a unit quaternion. The TRIAD uses two body frame vectors and two reference frame vectors to over determine the solution to satellite attitude calculation. An overdetermined solution limits the number of required computations. The two body frame vectors used in the TRIAD calculation are given by the sun sensor and the magnetometer. The two corresponding reference sun and magnetic field vectors are taken from the on-board observation models. The relevant entry in the observation models is determined by the satellite location data provided by the GPS. The TRIAD algorithm has a high degree of accuracy and is not computationally demanding, which makes this algorithm ideal for a CubeSat ADCS.

Finally, the desired CubeSat orientation will be maintained by using a combination of a PD controller and an Extended Kalman Filter. The EKF compares sensor measurements to previous knowledge of the spacecraft attitude to calculate a best estimate of the true state of the satellite, effectively filtering out noise from sensor measurements. The current state of the CubeSat is given by the angular velocity and unit quaternion. To simplify the EKF design, the spacecraft state is measured directly; the angular velocity measurements are given by the gyroscope and the unit quaternion is given by a virtual measurement determined through the TRIAD algorithm (using magnetometer and sun sensor measurements). The state estimate is then passed on to the PD controller which determines the error between the actual and desired quaternion in addition to determining if the spacecraft is rotating. If there is an orientation error or residual angular velocity the controller calculates a commanded magnetic dipole moment to close the error. The PD controller has a performance benefit over a Proportional controller since it considers the change in error (angular velocity) in the control law. In addition, the PD controller requires significantly less computational effort than a PID controller since both the quaternion (proportional) and angular velocity (derivative) are measured, so no additional calculations are required to implement the controller.

Additional information on these algorithms appear in Chapter 3 of this report. The final ADCS design is summarized in Figure 6.1, created by the 2012 ACDS team [7]:

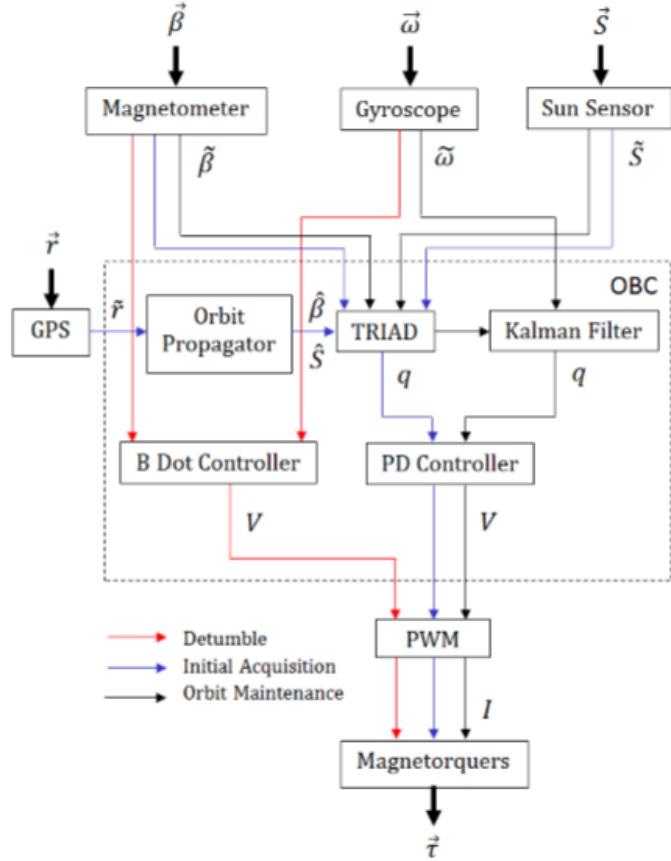


Figure 6.1: Block diagram of the final ADCS.

The aforementioned ADC algorithms were simulated using MATLAB[®]. To increase the realism of these simulations noise was injected into the simulated sensor readings. The magnitude of noise added to each sensor measurement was determined from each sensor's respective measurement standard deviation. Through the addition of adaptive gain to the B-dot controller the detumble time of the satellite was reduced to 800 seconds, which is one third of the detumble time of the fixed gain B-dot designed by the previous ADCS teams (using the new inertial matrix). In addition, the accuracy of the TRIAD algorithm and EKF were tested. Errors in the algorithms and simulation code were corrected and it was verified that the simulations produced accurate results.

Alongside simulations, detailed documentation on the theory behind the algorithms and control policies used in the three ADC phases was developed. This documentation was used to verify (and correct, if necessary) the theory discussed in the previous ADCS project papers. This documentation appears in the appendix of this report.

6.1.2 ADC Test-Bed Overview

Using the initial design developed by the previous ADCS team in [10], this project designed and constructed a prototype test bed. For the prototype design, a polycarbonate material was used to construct the platform of the test bed, in a 12 inch by 12 inch square, rather than the larger aluminum plate. 10-24 screws holes were drilled and tapped for attaching to the air bearing. Two spheres were cut with a lathe;

the first was a sphere that was milled off near the top (0.4 inch depth) to allow for greater degrees of rotation. This would put the center of the mass of the platform significantly above the center of rotation. Next, a hemisphere was cut with a lathe, in an attempt to match the center of mass with the center of rotation. A counterweight system is still necessary however, and the surface of both the bearing mount and the sphere need to be finely cut to ensure the air film thickness is the required size. The rest of the structure was built out of aluminum, and the prototype was put together. The O-ring seal in the bearing structure worked effectively, and the air tubing fit correctly. The completed test bed prototype is pictured in Figure 6.2:

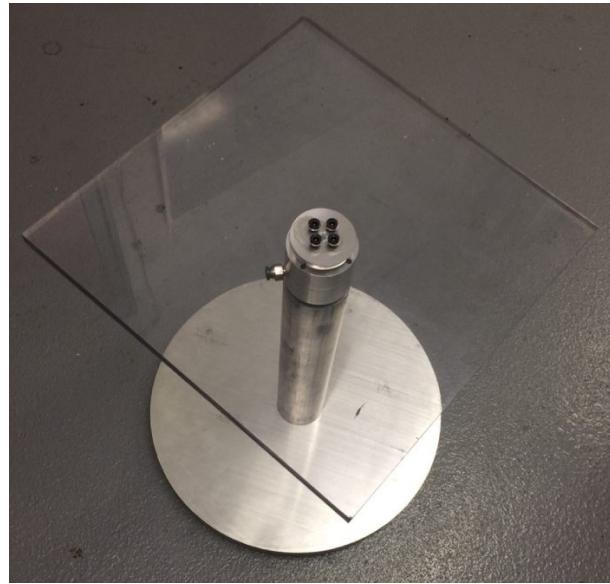


Figure 6.2: Test bed prototype.

and the machined air bearing can be seen in Figure 6.3:



Figure 6.3: Air bearing.

The machined air bearing proved to be fully functional and capable of rotating with minimal friction when compressed air is fed into the cup. While the platform is able to rotate when connected to the bearing assembly, additional work must be done on the platform to ensure the full functionality of the test bed. The next steps are detailed in the recommendations.

6.2 Recommendations

This section discusses recommendations to improve the current design of the WPI CubeSat attitude determination and control subsystem. These recommendations are separated into multiple subsections, found below.

6.2.1 Understanding the ADC Algorithms and Methods

Many of the attitude determination and control methods and concepts are not generally taught until the graduate level at most institutions. Students entering this project may not have the necessary background to design and evaluate an ADCS that meets the mission requirements. To assist future projects with the understanding of the relevant ADC theory, ADCS teams should create extensive documentation on new algorithms or methods and update any previous documentation as necessary. In addition, all students involved with the CubeSat ADCS project should consult the relevant sections in [28]. This textbook provides easy to follow explanations of spacecraft attitude control and static determination methods. Decreasing the time it takes for students to learn the theory increases the amount of time available for improving the CubeSat ADCS and increases the quality of the project results.

6.2.2 Improving the ADC Simulations

Even though the ADCS simulations were improved during this project, many more improvements can be made in the future. While the adaptive gain on the B-dot controller provided a significant increase in detumble performance, this control law can be optimized even further. Since the inertia matrix was calculated late into the project period, there was not sufficient time to properly optimize the adaptive gain equations for the updated satellite configuration. In addition, the B-dot control law can also be implemented as a hysteresis or Bang-Bang control law, which may increase performance.

Calculation of the computational requirements of the ADCS system could also be integrated within simulation code. The Computer subsystem team reported that computing requirements should not be a problem, but proper optimization of the On-Board Computer (OBC) will require these calculations. Specifically, the computational power that the detumble process, the TRIAD method, and the Extended Kalman Filter each use must be calculated. If simulations determine that the OBC has sufficient computing power, then a higher computational effort, but more accurate determination method may be implemented. While the TRIAD method has proven to have a high degree of accuracy in calculating the true attitude matrix, additional accuracy will improve the overall quality of the ADCS. These higher computational effort determination methods (described in Chapter 3) should be integrated into the existing ADCS simulations to determine their benefits.

After the detumble phase the PD controller used in the attitude maintenance phase was unable to close the quaternion error seen in Figure 4.7. This may have been due to the new inertia matrix for the spacecraft configuration. The following ADCS project should perform additional simulations to re-optimize the proportional and derivative gains calculated in [10] to increase the effectiveness of the controller. If the improved gain does not improve attitude maintenance performance to a sufficient level

then a controller during eclipse periods may be necessary. Since the virtual quaternion measurements calculated using the TRIAD algorithm are not available during eclipse, this controller should use only the ω measurements provided by the gyroscope. A simple controller of this nature will prevent disturbance torques from rotating the spacecraft when the remainder of the ADCS is in standby. This will decrease the change in error quaternion during eclipse. If future ADCS projects would like to implement a better, more complex controller in place of the current PD controller, Chapter 7 in [28] provides several possible control laws to consider.

In addition to improving the quality of ADC algorithms, future ADCS projects should continue annotating and documenting the MATLAB® simulation code. Proper documentation will allow proceeding ADCS teams to better understand, debug, and edit the simulation code.

6.2.3 Rewriting the Extended Kalman Filter

While the current EKF provides accurate state estimation, a simpler and more accurate EKF could be implemented within the simulation code. The current EKF was designed using the discrete method, which may be difficult for undergraduate students to implement because this method involves deriving the linearized, discrete equations of motion. This is a concern since for hardware testing the EKF will must be rewritten in python. Since it will be difficult to rewrite the current, discrete EKF in python future ADCS teams should design a simpler, continuous-discrete EKF, which utilizes the continuous equations of motion. This new EKF should first be integrated into the MATLAB® simulation code to verify the functionality of the filter. The theory for a continuous-discrete EKF is detailed in Appendix E.

6.2.4 Continuation of STK Simulations

For future work in STK, future ADCS teams should start with the detumble script we were able to create based on STK examples and mission requirements. There were issues with the results obtained that are shown in the sample data in the appendix, where the angular velocities were staying constant even though the 3D graphics window demonstrated a clear detumble (the initial velocity clearly decreased to near-zero). The first step should be to resolve this issue. This issue may have been caused by a) an error in exporting the .a attitude file, in that the angular velocities in STK did not correspond to the outputted text file or b) the script was not reading the correct magnetic field reading, which should have been pulling directly from IGRF within the software.

If these issues are resolved, ideally it would be best to then integrate the MATLAB® code for TRIAD, the EKF, and the PD controller into STK. Due to the time constraints and the errors that were discovered when using STK for the first time, the transfer between MATLAB® to STK was not completed during this project. Our recommendation is that this be done while maintaining the MATLAB® code, as this is crucial for the future task of propagating this into the language used by the OBC on the CubeSat.

The work on demonstrating the attitude coverage by the sun sensors was completed, although this can be looked at more in terms of producing raw data rather than just a graphic simulation. A graphical

representation of the sun vector in the satellite's orbit would demonstrate useful results for the ADCS.

6.2.5 Future Work on the Test Bed

For the test bed, the full platform can be constructed out of something stronger than the polycarbonate if it is deemed necessary. The polycarbonate was used because the time constraint of the project meant there was not enough time to machine a counterweight system for use with an aluminum plate, although it can support the loads of the CubeSat and other devices. With aluminum's density of 2.7 g/cm³ and an 18" x 18" frame, this led to a roughly 38 pound platform with a center of mass about 1.4" above the center of rotation of the air bearing. This would lead to instability if any rotation was applied anywhere other than on the z axis (using the Solidworks reference frame). Future ADCS teams should look at the design used by HSFL that was studied in this report. This design allowed for balancing in the x, y, and z directions while leaving the entire platform face free for placing components. James Loiselle, from senior lab technician Higgins Machine Shop, should be asked about continuing the work that was done in this area for next year.

In [35], the full nonlinear equations of motion of the entire simulator system are derived. If the center of mass of the platform is located at S^* , the angular velocity of the platform with respect to the inertial frame be denoted by ${}^I\omega^S$, the velocity of dm is given by $\mathbf{v} = {}^I\omega^S \times \mathbf{r}$, where \mathbf{r} is the position vector from the bearing center B^* . The angular momentum of the whole system is:

$$\mathbf{H} = \mathbf{I}^{(S+W)/B^*} \cdot {}^I\omega^S + \sum_{l=1}^3 \mathbf{I}^{W_l/W_{l^*}} \cdot {}^S\omega^{W_{l^*}} \quad (6.1)$$

where $\mathbf{I}^{W_l/W_{l^*}}$ is the inertia dyadic of the l^{th} wheel (assuming a wheel is placed on the platform) with respect to the center of mass of the wheel. The external torque caused by gravity is given by:

$$\mathbf{G} = \mathbf{r}_s \times mg\mathbf{K} = \mathbf{I} \cdot {}^I\dot{\omega}^S + {}^I\omega^S \times \mathbf{I} \cdot {}^I\omega^S + \sum_{l=1}^3 (\mathbf{I}^l \cdot {}^S\dot{\omega}^{W_l} + ({}^I\omega^S + {}^S\omega^{W_l}) \times \mathbf{I}^l \cdot {}^S\omega^{W_l}) \quad (6.2)$$

where both of these calculations are assuming three reaction wheels are placed on the assembly. It is recommended that [35] is referred to for calculating the affects of a moving center of mass on the stability of the platform. Preliminary testing can be done by ensuring the platform is parallel to the x-y plane and spinning only about the z axis. This would only require having the center of mass be aligned with the center of rotation.

In addition, initial testing of the prototype test-bed should be done using the recommended sensors and equipment purchased during this project. The experimental setup described in Chapter 5 is capable of wirelessly reading angular velocity measurements, which provides a simple demonstration on the functionality and purpose of the test bed.

References

- [1] Bowen JA. On-board orbit determination and 3-axis attitude determination for picosatellite applications. Master's Theses and Project Reports. 2009;p. 131.
- [2] California Polytechnic State University, San Luis Obispo, Stanford University's Space Systems Development Lab. The CubeSat Program;. Available from: <http://www.cubesat.org/about/>.
- [3] Dopart C, Morlath R, Oliver E, Schomaker J. "Design and Analysis for a CubeSat Mission". Worcester, MA: Worcester Polytechnic Institute; 2012.
- [4] Sylwester J, Kowalinski M, Gburek S, Siarkowski M, Kuzin S, Farnik F, et al. SphinX measurements of the 2009 solar minimum X-ray emission. *The Astrophysical Journal*. 2012;751(2):111.
- [5] Curci E, Jacobson J, Schlack W, Slabinski K. "Design and Analysis for a CubeSat Mission". Worcester, MA: Worcester Polytechnic Institute; 2017.
- [6] Ko D, Laudage S, Murphy M, Pelgrift D, Young S. "Thermal, Telecommuncation, Command and Data Handling, and Power Subsystem for a CubeSat". Worcester, MA: Worcester Polytechnic Institute; 2017.
- [7] Dawson E, Nassiff N, Velez D. "Attitude Determination and Control Subsystem Design for a CubeSat". Worcester, MA: Worcester Polytechnic Institute; 2012.
- [8] Bauer J, Carter M, Kelley K, Mello E, Neu S, Orphanos A, et al. "Mechanical, Power, and Thermal Subsystem Design for a CubeSat Mission". Worcester, MA: Worcester Polytechnic Institute; 2017.
- [9] Billings D, Gradel I, Hoey F, Lavallee P, Martinez N. "Design and Analysis for a CubeSat Mission". Worcester, MA: Worcester Polytechnic Institute; 2013.
- [10] Farhat A, Ivase J, Lu Y, Snapp A. "Attitude Determination and Control System for CubeSat". Worcester, MA: Worcester Polytechnic Institute; 2013.
- [11] Hanley J, Joseph B, Miller M, Monte S, Trudeau J, Weinrick R. "Thermal, Telecommunication and Power Systems for a CubeSat. Worcester, MA: Worcester Polytechnic Institute; 2017.
- [12] Bigelow A, Hawkins C. "Attitude Determination and Control, On Board Computing and Communication Subsystem Design for a CubeSat Mission". Worcester, MA: Worcester Polytechnic Institute; 2011.

- [13] Ure NK, Kaya YB, Inalhan G. The development of a Software and Hardware-in-the-Loop Test System for ITU-PSAT II nano satellite ADCS. In: Aerospace Conference, 2011 IEEE. IEEE; 2011. p. 1–15.
- [14] Wertz JR, Everett DF, Puschell JJ. Space mission engineering: the new SMAD. Microcosm Press; 2011.
- [15] Ure NK, Kaya YB, Inalhan G. The development of a Software and Hardware-in-the-Loop Test System for ITU-PSAT II nano satellite ADCS. In: Aerospace Conference, 2011 IEEE. IEEE; 2011. p. 1–15.
- [16] Seshan C. Cell efficiency dependence on solar incidence angle. In: Photovoltaic Specialists Conference (PVSC), 2010 35th IEEE. IEEE; 2010. p. 002102–002105.
- [17] Thébault E, Finlay CC, Beggan CD, Alken P, Aubert J, Barrois O, et al. International geomagnetic reference field: the 12th generation. Earth, Planets and Space. 2015;67(1):1–19.
- [18] Larson WJ, Wertz JR. Space mission analysis and design. Microcosm, Inc., Torrance, CA (US); 1992.
- [19] Fine Sun Sensor Datasheet;. Accessed: 2016-09-10. <https://www.cubesatshop.com/wp-content/uploads/2016/06/NSS-Cubesat-Sun-Sensor.pdf>.
- [20] Fine Sun Sensor Product Page;. Accessed: 2016-09-10. <https://www.cubesatshop.com/product/nss-cubesat-sun-sensor/>.
- [21] SpaceMicro. CSS-01,02 Coarse Sun Sensors;. Accessed: 2016-09-10. <http://www.spacemicro.com/assets/datasheets/guidance-and-nav/CSS.pdf>.
- [22] Honeywell. 3-Axis Digital Compass IC HMC5883L;. Accessed: 2016-09-20. https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf.
- [23] Triple-axis Magnetometer (Compass) Board - HMC5883L;. Accessed: 2016-09-20. <https://www.adafruit.com/product/1746>.
- [24] Analog-Devices. ADXRS453 Datasheet and Product Info;. Accessed: 2016-09-15. <http://www.analog.com/en/products/mems/gyroscopes/adxrs453.html#product-overview>.
- [25] Analog-Devices. ADXRS453 Datasheet;. Accessed: 2016-09-15. <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXRS453.pdf>.
- [26] ZARM-Technik. MT0.5-1 Technical Performance Datasheet; September 2016.
- [27] Surrey. SGR-05U Satellite GPS Datasheet; December 2014.
- [28] Markley FL, Crassidis JL. Fundamentals of spacecraft attitude determination and control. vol. 33. Springer; 2014.

- [29] Black HD. A passive system for determining the attitude of a satellite. *AIAA journal*. 1964;2(7):1350–1351.
- [30] Shuster MD. The generalized Wahba problem. *The Journal of the Astronautical Sciences*. 2006;54(2):245–259.
- [31] Markley FL, Mortari D. How to estimate attitude from vector observations. 1999;.
- [32] AGI. Systems Tool Kit (STK);. Available from: <https://www.agi.com/products/stk/>.
- [33] Wukelic DE. Air Bearing ADCS Test Bed. Hawaii Space Flight Laboratory; 2009.
- [34] Meissner DM. A three degrees of freedom test-bed for nanosatellite and Cubesat attitude dynamics, determination, and control. DTIC Document; 2009.
- [35] Kim B, Velenis E, Kriengsiri P, Tsiotras P. Designing a low-cost spacecraft simulator. *IEEE control systems*. 2003;23(4):26–37.
- [36] Components S. SRA100-R45 Drawings and Specifications;. Accessed: 2016-10-19. <http://www.specialtycomponents.com/Products/sra100-r45/>.
- [37] Adafruit. 9-DOF Absolute Orientation IMU Fusion Breakout Product Page;. Accessed: 2017-02-15. <https://www.adafruit.com/products/2472>.
- [38] Adafruit. Feather HUZZAH with ESP8266 WiFi Product Page;. Accessed: 2017-02-15. <https://www.adafruit.com/products/3213>.
- [39] EFOSHM Ultra Slim Portable Power Bank Product Page;. Accessed: 2017-02-10. <https://www.amazon.com/EFOSHM-Portable-External-Charging-Smartphones/dp/B01CQGEXQC>.
- [40] Foundation RP. Raspberry Pi 3 Product Page;. Accessed: 2017-02-20. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [41] Großekathöfer K, Yoon Z. Introduction into quaternions for spacecraft attitude representation. TU Berlin. 2012;16.
- [42] Singla P, Mortari D, Junkins JL. How to avoid singularity when using Euler angles. *Advances in the Astronautical Sciences*. 2005;119(II):1409–1426.
- [43] Markley FL. Attitude determination using vector observations: A fast optimal matrix algorithm. 1993;.

Appendices

Appendix A - Quaternions

Quaternions are often used as an attitude representation parameter of rigid bodies like spacecraft because they are computationally less intense to use than other attitude parameters (like Euler angles or a directional cosine matrix) [41]. They also negate the singularity issue that occurs when describing attitude kinematics in terms of Euler angles [42]. For instance, take the three basic rotation matrices for Euclidean space.

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (\text{A.1})$$

$$R_2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (\text{A.2})$$

$$R_3 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3})$$

Assuming that \mathbf{i} , \mathbf{j} , \mathbf{k} are the coordinate axes that the numbers 1, 2, 3 represent. Now despite being a relatively simple way to visualize rotations, we see a geometric singularity occur when $\theta_2 = \pm \frac{\pi}{2}$ or 0 for the asymmetric Euler angle sets, and when $\theta_2 = \pm \pi$ or 0 for the symmetric Euler angle sets. It therefore becomes a mathematical fact that singularities cannot be eliminated in any three dimensional representation of orientation using Euler angles. A resulting direction cosine matrix, denoted by A , from the three rotation matrices can be formulated as

$$A_{ijk} = R_1 R_2 R_3 \quad (\text{A.4})$$

with the same singularity issue present. This is where quaternions can be used instead, as an improved method of describing attitude for spacecraft.

Quaternions can be used to parametrize the spacecraft's attitude with respect to a reference coordinate system, perform coordinate transformations (like we will be when we calculate a vector in the body-fixed frame based on a known vector in the inertial frame), and to propagate the attitude from one orientation to the next by integrating the spacecraft equations of motion.

In essence, a quaternion is a 4x1 matrix that consists of a scalar part as its first element, and a vector part as the next three elements.

Let our quaternion be notated by the letter \mathbf{q} . Let s be the scalar part, and \mathbf{v} be the vector part.

$$\mathbf{q} = \begin{bmatrix} \mathbf{v} \\ s \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ s \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (\text{A.5})$$

A quaternion that represents a coordinate transformation from system A to system B, $q_{A \rightarrow B}$, can be defined by the following equation, where \mathbf{e} is the unitized rotational axis and θ is the transformation angle:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{e} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} \quad (\text{A.6})$$

Appendix B - Magnetic Torquer Overview

Overview

A magnetic torquer is an active device that creates a torque using an external magnetic field and a magnetic dipole moment. Unlike a momentum wheel or a reaction wheel this actuator has no moving parts and is less prone to malfunction. The construction of this device is fundamentally simple; a magnetic torquer consists of a metal rod wrapped in copper wire and is connected to a power source by two leads. Running a current through the wire induces a magnetic moment, denoted by μ , normal to the magnetic torque and in the presence of an external magnetic field, denoted by B , creates a torque perpendicular to the two vectors as given by the following relationship:

$$\tau = \mu \times B \quad (\text{B.1})$$

Where μ consists of the magnetic moment components along each principal axis and B consists of the magnetic field components along each principal axis, written in the spacecraft body frame. To obtain complete three-axis control of the spacecraft three magnetic torque rods are required: one aligned with the body x axis, another aligned with the y axis, and the third aligned with the z axis.

Power Consumption of a Magnetic Torque Rod

Power is a very limited resource on a cube/nano satellite. Using simulation results to estimate how much power the three magnetic torquers will consume during the detumbling phase and attitude maintenance phase is necessary for proper power subsystem management. Since the ZARM MT0.5-1 is rated for use in the linear magnetic region, power and magnetic moment are determined using the following relationships:

$$\mu = nIA \quad (\text{B.2})$$

$$P = I^2R \quad (\text{B.3})$$

Where the constants n , R , and A are properties of the magnetic torquer rod as described in the table below. The notation described in the table below is consistent throughout the analysis.

Variable	Description
P	Power (W)
I	Current (A)
R	Coil Resistance (Ω)
μ	Magnetic Dipole Moment (Am^2)
n	Number of wire turns in coil
A	Area (m^2)

Description of variables used during this analysis.

All constants values required to solve equations (B.2) and (B.3) are provided in the ZARM MT0.5-1 magnetic torquer data sheet excluding the number of wire turns of the copper wire coil (n). This value can be estimated by using the maximum values stated for magnetic moment and current for the magnetic torquer and solving equation (B.2) for n . Since the operating range of the magnetic torquer is linear and magnetic moment scales linearly with current, this value of n is an accurate estimate of the real number of turns in the coil.

Substituting equation (B.2) in to equation (B.3) provides the relationship between magnetic moment and power. This equation can be applied to the x, y, and z axis magnetic torquers:

$$\mu_x = \sqrt{\frac{P_x}{R}} n A \quad (\text{B.4})$$

$$\mu_y = \sqrt{\frac{P_y}{R}} n A \quad (\text{B.5})$$

$$\mu_z = \sqrt{\frac{P_z}{R}} n A \quad (\text{B.6})$$

Solving for P_x , P_y , and P_z provides the instantaneous power consumed by each of the three magnetic torquers:

$$P_x = \frac{\mu_x^2 R}{n^2 A^2} \quad (\text{B.7})$$

$$P_y = \frac{\mu_y^2 R}{n^2 A^2} \quad (\text{B.8})$$

$$P_z = \frac{\mu_z^2 R}{n^2 A^2} \quad (\text{B.9})$$

Summing the instantaneous power can be used to estimate the total amount of power each magnetic torquer consumes during detumble phase and how much average power is required during the attitude maintenance phase.

Determining Current Given a Commanded Magnetic Moment

As stated earlier, the magnetic torquer does not command a torque on its own; it creates a torque through the use of magnetic moments. This means that while the final control output is a torque perpendicular to the magnetic moment and external magnetic field, the control is enacted by commanding a magnetic moment using a control law.

Since the magnetic torquer is a simple device with no internal computer, it is unable to use a commanded magnetic moment as a control signal. To induce the desired moment, the on-board computer must send an analog current signal to the magnetic torquer. The equations relating magnetic moment to current are derived below.

Equations (B.10), (B.11), and (B.12) provide the relationship between magnetic moment and the current flowing through the wire wrapped around each magnetic torque rod:

$$\mu_x = nI_x A \quad (\text{B.10})$$

$$\mu_y = nI_y A \quad (\text{B.11})$$

$$\mu_z = nI_z A \quad (\text{B.12})$$

Solving equations (B.10), (B.11), and (B.12) for I_x , I_y , and I_z respectively, provides the expressions necessary to convert a desired magnetic moment into an equivalent analog current. This calculated current can be sent to the on-board computer to power the magnetic torquers and enact the desired spacecraft attitude control.

$$I_x = \frac{\mu_x}{nA} \quad (\text{B.13})$$

$$I_y = \frac{\mu_y}{nA} \quad (\text{B.14})$$

$$I_z = \frac{\mu_z}{nA} \quad (\text{B.15})$$

Determining Torque from Commanded Moment and Current

Using equations (B.1), (B.2), and the commanded magnetic moment it is possible to determine the resultant control torque vector and current required to command a desired torque. The expanded scalar representation of equation (B.1) is provided below:

$$\tau_x = \mu_x B_z - \mu_z B_y \quad (\text{B.16})$$

$$\tau_y = \mu_z B_x - \mu_x B_z \quad (\text{B.17})$$

$$\tau_z = \mu_x B_y - \mu_y B_x \quad (\text{B.18})$$

By substituting equation (B.2) into equations (B.13), (B.14), and (B.15) control torque can be related to the instantaneous current of each magnetic torquer:

$$\tau_x = nA(I_x B_z - I_z B_y) \quad (\text{B.19})$$

$$\tau_y = nA(I_zB_x - I_xB_z) \quad (\text{B.20})$$

$$\tau_z = nA(I_xB_y - I_yB_x) \quad (\text{B.21})$$

And in matrix form:

$$\boldsymbol{\tau} = nA \begin{bmatrix} I_xB_z - I_zB_y \\ I_zB_x - I_xB_z \\ I_xB_y - I_yB_x \end{bmatrix} \quad (\text{B.22})$$

Which can be rewritten as:

$$\boldsymbol{\tau} = nA \begin{bmatrix} 0 & B_z & -B_y \\ -B_z & 0 & B_x \\ B_y & -B_x & 0 \end{bmatrix} \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} \quad (\text{B.23})$$

In this derivation the constant 3x3 matrix in equation (B.23) is singular and cannot be inverted, there is no closed form solution to solve for current in terms of control torque. The resultant torque induced by the magnetic torquer can only be calculated after the commanded moment (or the commanded current) has been determined.

Appendix C - Spacecraft Detumble Phase

B-dot Controller Introduction and Description

When a cube/nano satellite is initially sent out into orbit it will be spinning with a relatively high angular velocity and must be detumbled before the mission can begin. In general, the detumbling phase of a spacecraft is governed by a B-dot controller, derived using the Lyapunov function and Lyapunov stability analysis. The B-dot control law commands a magnetic moment by utilizing the time derivative of the local geomagnetic field, expressed in spacecraft body frame coordinates. The resultant control torque produced by the three magnetic torquers will produce an angular velocity counteracting the spin of the spacecraft. The following equation is the vector representation of the B-dot control law where μ is the magnetic dipole moment, \mathbf{B} is the Earth's magnetic field (\mathbf{b} is the unitized vector), and k is the controller gain:

$$\boldsymbol{\mu} = -\frac{k}{\|\mathbf{B}\|} \dot{\mathbf{b}} \quad (\text{C.1})$$

As seen in equation (C.1), the namesake of the B-dot controller comes from the time derivative of the magnetic field measurements provided by the magnetometer. The reason why this control law is useful for detumbling a spacecraft is given by first looking at the vector transport theorem:

$$\dot{\mathbf{V}}^B = \dot{\mathbf{R}}_A^B - \boldsymbol{\omega}_{BA}^B \times \mathbf{V}^B \quad (\text{C.2})$$

This theorem states that the time derivative of the body frame is dependent on the relative rotation between the body and reference frames. Which means that the derivative of the Earth's magnetic field in the body frame can be written in the following form:

$$\dot{\mathbf{B}} = \mathbf{A} \dot{\mathbf{R}} - \boldsymbol{\omega} \times \mathbf{B} \quad (\text{C.3})$$

The angular velocity in equation (C.3) is the relative velocity of the spacecraft relative to the Earth reference frame, i.e. the angular velocity of the spacecraft. This angular velocity is the term that must be driven to zero (or near zero) to detumble the spacecraft.

A common assumption made during B-dot controller design is that during the initial detumbling phase the angular velocity between the reference and body frames, given by $\boldsymbol{\omega} \times \mathbf{B}$ is much larger than the derivative of the Earth reference magnetic field vector and can be assumed to be zero for practical purposes. This means that if spacecraft angular velocity data is available (i.e. gyroscope sensor data) the B-dot control law can be rewritten as the following:

$$\boldsymbol{\mu} = \frac{k}{\|\mathbf{B}\|} \boldsymbol{\omega} \times \mathbf{b} \quad (\text{C.4})$$

In some cases of the B-dot control law, the magnitude of the magnetic field vector \mathbf{B} is assumed constant and is factored into the gain k . For the k that appears in equation (C.1), a constant positive value can be chosen through simulation or an estimated value for k can be determined by evaluating the

following expression:

$$k = \frac{4\pi}{T_{orb}}(1 + \sin(i_{orb}))J_{min} \quad (\text{C.5})$$

Where T_{orb} is the orbital period, i_{orb} is the inclination angle of the orbit, and J_{min} is the minimum principal moment of inertia. This expression was derived by Avanzini and Giulietti in *Magnetic Detumbling of a Rigid Spacecraft* by analyzing the closed loop dynamics of the component of angular velocity perpendicular to the magnetic field vector \mathbf{B} .

As mentioned previously, the stability of the B-dot controller can be proven by using Lyapunov stability analysis. Consider the following candidate Lyapunov function (V) and Lyapunov derivative where J is the moment of inertia matrix:

$$V = \frac{1}{2}\boldsymbol{\omega}^T J \boldsymbol{\omega} \quad (\text{C.6})$$

$$\dot{V} = \boldsymbol{\omega}^T J \dot{\boldsymbol{\omega}} \quad (\text{C.7})$$

To prove stability the derivative of the Lyapunov function must be negative definite, i.e. less than zero for any value of $\boldsymbol{\omega}$. To relate the B-dot controller to the Lyapunov function, the equation for torque and Euler's equation for rotational motion are required:

$$\boldsymbol{\tau} = \boldsymbol{\mu} \times \mathbf{B} \quad (\text{C.8})$$

$$J \dot{\boldsymbol{\omega}} = -[\boldsymbol{\omega}^\times] J \boldsymbol{\omega} + \boldsymbol{\tau} \quad (\text{C.9})$$

After combining equations (C.8) and (C.9) and substituting $\boldsymbol{\mu}$ for the B-dot control law in equation (C.4), the Lyapunov derivative becomes:

$$\dot{V} = -\boldsymbol{\omega}^T (I_3 - \mathbf{b} \mathbf{b}^T) \boldsymbol{\omega} \quad (\text{C.10})$$

Since the eigenvalues of $(I_3 - \mathbf{b} \mathbf{b}^T)$ must always be 0, 1, and 1, \dot{V} is negative semi-definite. This means that the B-dot control law is stable and will bring $\boldsymbol{\omega}$ to zero unless $\boldsymbol{\omega}$ is parallel to \mathbf{b} . This is not a concern in practice and as such means that the B-dot controller is an adequate controller for the detumble phase of the spacecraft attitude control system.

Control Torque during Detumble

An expression for determining the control torque of the B-dot controller can be determined by some simple algebraic manipulations. Evaluating the cross product term in the B-dot control law provides the following:

$$\boldsymbol{\mu} = -\frac{k}{\|\boldsymbol{B}\|^2} \begin{bmatrix} B_y\omega_z - B_z\omega_y \\ B_z\omega_x - B_x\omega_z \\ B_x\omega_y - B_y\omega_x \end{bmatrix} \quad (\text{C.11})$$

Substituting equation (C.11) into equation (C.1) provides an expression for torque in terms of the Earth's magnetic field in the body frame of the spacecraft and the angular velocity of the spacecraft.

$$\boldsymbol{\tau} = -\frac{k}{\|\boldsymbol{B}\|^2} \begin{bmatrix} (B_y^2 + B_z^2)\omega_x - B_x B_y \omega_y - B_x B_z \omega_z \\ -B_y B_x \omega_x + (B_x^2 + B_z^2)\omega_y - B_y B_z \omega_z \\ -B_z B_x \omega_x - B_z B_y \omega_y + (B_x^2 + B_y^2)\omega_z \end{bmatrix} \quad (\text{C.12})$$

Which can be rewritten in the following form:

$$\boldsymbol{\tau} = -\frac{k}{\|\boldsymbol{B}\|^2} \begin{bmatrix} (B_y^2 + B_z^2) & -B_x B_y & -B_x B_z \\ -B_y B_x & (B_x^2 + B_z^2) & -B_y B_z \\ -B_z B_x & -B_z B_y & (B_x^2 + B_y^2) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (\text{C.13})$$

The expression above can be used to express the total control torque produced by all three magnetic torque rods at an instance in time, by commanding magnetic moment using \boldsymbol{B} and $\boldsymbol{\omega}$.

Implementing B-dot as a Bang-Bang Control Law

B-dot control is often implemented as a bang-bang control law. A bang-bang controller (also known as an on-off or hysteresis controller) is a feedback controller that instead of providing a variable control output, switches between two states. In the case of detumbling a spacecraft using B-dot control, a bang-bang control law will not calculate a specific magnetic moment but rather calculate the sign of the spacecraft spin and apply the maximum allowed moment in the opposite direction. The bang-bang implementation of the B-dot controller is written as the following:

$$\mu_i = -\mu^{max} \text{sign}(\boldsymbol{u}_i \cdot \dot{\boldsymbol{B}}) \quad (\text{C.14})$$

Where the μ^{max} is set to the maximum rated magnetic moment of each magnetic torquer and \boldsymbol{u} is the unit vector along which the magnetic moment is applied. When there are three magnetic torquers, one along each axis, the control law becomes:

$$\mu_x = -\mu^{max} \text{sign}(\boldsymbol{i} \cdot \dot{\boldsymbol{B}}) \quad (\text{C.15})$$

$$\mu_y = -\mu^{max} \text{sign}(\boldsymbol{j} \cdot \dot{\boldsymbol{B}}) \quad (\text{C.16})$$

$$\mu_z = -\mu^{max} \text{sign}(\boldsymbol{k} \cdot \dot{\boldsymbol{B}}) \quad (\text{C.17})$$

Hysteresis control laws are generally used to minimize the time in which a control action is performed.

Since these equations apply the maximum allowable moment, the detumble time is shortened but the total power required during this phase increases. If the power during the detumble phase is limited then the bang-bang control law cannot be used, or must be combined with the traditional B-dot control law.

In addition, another benefit of using the bang-bang implementation of the B-dot controller is that there is no need to calculate a gain or saturation condition. This means that the control law is less computationally intensive and easier to integrate into ADC algorithms than the traditional control B-dot control law.

Appendix D - Initial Attitude Determination

Once the detumble phase is complete, the next step is to determine the orientation, or attitude, of the satellite. Euler's theorem states that motion of a rigid body about a point is described by a rotation about some axis. In other words, to determine the orientation of the satellite a body axis, a reference axis, and the rotation between them is required. Since vectors contain only two independent pieces of attitude information (two of the three axes are linearly independent, while the third is constrained to the cross product of the other two axes), a second vector is required to completely define the axis of rotation. So, the initial attitude calculation of the satellite requires two unit vectors measured in the body frame and two unit vectors measured in a reference frame. Unit vectors are used during these calculations because the rotation angle between the body and reference frames is not dependent on the magnitude of each vector, but rather the direction of each axis.

For this application, the sun vector and magnetic field vector are chosen as the two body unit vectors for the initial attitude calculations. These vectors are determined from sun sensor and magnetometer measurements, respectively, and then normalized. In some cases, the sun and magnetic field vectors may be replaced by star tracker measurements. The respective reference vectors are determined by first acquiring the satellite's orbital radius and true anomaly by using a GPS and a model of the predetermined orbit. The corresponding sun and magnetic field vectors can be determined by referring to the satellite's estimated position in sun and magnetic field models. The following relationship describes how the measured body frame vectors relate to the reference frame vectors:

$$A\mathbf{r}_1 = \mathbf{b}_1 \quad (\text{D.1})$$

$$A\mathbf{r}_2 = \mathbf{b}_2 \quad (\text{D.2})$$

The 3x3 attitude matrix A, known as a direction cosine matrix, describes the three rotation angles between the body and reference vectors. Since the rotation matrix A holds true for any vector rotated between the body and reference frame, the angles determined from the attitude matrix also describe the rotation between the body and references axes. The rotation between these two sets of axes provides a complete picture of the orientation of the satellite. Since the attitude matrix is the same in both equations, this implies that the following relationship between the two pairs of body and reference vectors must hold true:

$$\mathbf{b}_1 \cdot \mathbf{b}_2 = (A\mathbf{r}_1) \cdot (A\mathbf{r}_2) = \mathbf{r}_1^T A^T A \mathbf{r}_2 = \mathbf{r}_1 \cdot \mathbf{r}_2 \quad (\text{D.3})$$

In the presence of noise or measurement errors equation (D.3) does not hold true. This means that it is not possible to satisfy both equations (D.1) and (D.2) in a real-world environment. To combat this issue, numerous algorithms have been developed to determine a best estimate of the attitude matrix A with the understanding that equations (D.1) and (D.2) cannot be satisfied simultaneously. In a CubeSat environment, computing power tends to be the limiting factor on which of the many determination algo-

rithms is chosen. The two least computationally intensive algorithms are known as the TRIAD method, which calculates an attitude matrix, and the direct quaternion method, which directly determines a system quaternion. These two algorithms are detailed in the following sections.

Asymmetrical TRIAD Method

The traditional TRIaxial Attitude Determination algorithm, also known as the TRIAD algorithm, assumes that one of the two reference and body unit vectors has less noise or is more accurate than the other and asymmetrically weighs the chosen vector more heavily during calculations. For the sake of simplicity, the more accurate unit vector for the reference and body frames will be denoted as \mathbf{r}_1 and \mathbf{b}_1 , respectively. This assumption implies that the attitude matrix calculated using the TRIAD method provides an exact solution to equation (D.1) but only provides an approximate solution to equation (D.2). The justification for this assumption is that since the estimated attitude matrix was calculated using the more accurate unit vectors, then the exact solution for equation (D.1) is closer to the true attitude matrix than the exact solution for equation (D.2). The TRIAD estimate of the attitude matrix is given by the following algorithm:

$$A = [\mathbf{w}_1 \mathbf{w}_2 \mathbf{w}_3][\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3]^T = \mathbf{w}_1 \mathbf{v}_1^T + \mathbf{w}_2 \mathbf{v}_2^T + \mathbf{w}_3 \mathbf{v}_3^T \quad (\text{D.4})$$

The w and v matrices are known as triads and consist of three orthogonal unit vectors. The w triad is the spacecraft body frame triad and is determined from the two measured body frame unit vectors discussed earlier. Similarly, the v triad is the reference frame triad and is determined from the two reference frame unit vectors. The components of the reference frame triad are calculated as follows:

$$\mathbf{v}_1 = \mathbf{r}_1 \quad (\text{D.5})$$

$$\mathbf{v}_2 = \mathbf{r}_\times = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|} \quad (\text{D.6})$$

$$\mathbf{v}_3 = \mathbf{r}_1 \times \mathbf{r}_\times \quad (\text{D.7})$$

The components of the body frame triad are calculated in a similar fashion:

$$\mathbf{w}_1 = \mathbf{b}_1 \quad (\text{D.8})$$

$$\mathbf{w}_2 = \mathbf{b}_\times = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{|\mathbf{b}_1 \times \mathbf{b}_2|} \quad (\text{D.9})$$

$$\mathbf{w}_3 = \mathbf{b}_1 \times \mathbf{b}_\times \quad (\text{D.10})$$

Expanding equation 4 provides:

$$\hat{A}_{TRIAD} = \mathbf{b}_1 \mathbf{r}_1^T + (\mathbf{b}_1 \times \mathbf{b}_\times)(\mathbf{r}_1 \times \mathbf{r}_\times) + \mathbf{b}_\times \mathbf{r}_\times^T \quad (\text{D.11})$$

Notice that with some manipulation, the attitude matrix given by equation (D.11) satisfies equation (D.1). It should also be noted that in the ideal case (with no measurement error) where equation (D.3) is satisfied, then the attitude matrix then also satisfies equation (D.2). In systems where a quaternion is preferred over a direction cosine matrix, there are several methods to transform the estimated attitude matrix given by the TRIAD method into an estimated unit quaternion.

Asymmetrical Direct Quaternion Method

The direct quaternion method of calculating initial attitude was developed to bypass the calculation of the attitude matrix in cases where a quaternion is the desired form of initial attitude determination. This algorithm is useful because it requires fewer floating point calculations compared to the TRIAD method, thus saving computing power. The algorithm behind the Direct Quaternion Method is based on the knowledge that a unit quaternion can be used to rotate a vector, or in other words, used as an attitude matrix. A vector rotated by a quaternion is denoted by the following:

$$A(\mathbf{q})\mathbf{v} = (q_4^2 - |\mathbf{q}|^2)\mathbf{v} + 2(\mathbf{q} \cdot \mathbf{v})\mathbf{q}_{1:3} - 2q_4(\mathbf{q} \times \mathbf{v}) \quad (\text{D.12})$$

and the quaternion corresponding to a rotation matrix about some arbitrary angle and axis is:

$$\mathbf{q} = [e \sin(\frac{\theta}{2}), \cos(\frac{\theta}{2})] \quad (\text{D.13})$$

Similar to the Asymmetrical TRIAD method the Asymmetrical Direct Quaternion method utilizes the body and reference unit vectors determined to be the most accurate out of the pair, which are denoted by a subscript 1 in the calculations. The first step in the derivation is to determine the quaternion that relates the chosen reference vector on to the frame of the chosen body vector through the minimum angle of rotation. This quaternion is given by:

$$\hat{\mathbf{q}}_{min} = \frac{1}{\sqrt{2(1 + \mathbf{b}_1 \cdot \mathbf{r}_1)}} [\mathbf{b}_1 \times \mathbf{r}_1, 1 + \mathbf{b}_1 \cdot \mathbf{r}_1] \quad (\text{D.14})$$

The rotated quaternion can now be determined by using the minimum-angle rotation and the quaternions of the body and reference frame unit vectors determined through equation (D.13) The general form of the rotated quaternion represented as:

$$\hat{\mathbf{q}}_1 = \mathbf{q}(\mathbf{b}_1, \theta_{body}) \otimes A(\mathbf{q}_{min}) \otimes \mathbf{q}(\mathbf{r}_1, \theta_{ref}) \quad (\text{D.15})$$

Similarly, if the calculations were performed on the second set of unit vectors:

$$\hat{\mathbf{q}}_2 = \mathbf{q}(\mathbf{b}_2, \phi_{body}) \otimes A(\mathbf{q}_{min}) \otimes \mathbf{q}(\mathbf{r}_2, \phi_{ref}) \quad (\text{D.16})$$

Through some manipulation it is easy to notice that the vector portion of $\hat{\mathbf{q}}_1$ is perpendicular to

$\mathbf{b}_1 - \mathbf{r}_1$ and likewise for the vector portion of $\hat{\mathbf{q}}_2$. Using this observation, Reynolds related the results of equations (D.15) and (D.16) by choosing a rotation angle that made the two quaternions perpendicular to both $\mathbf{b}_1 - \mathbf{r}_1$ and $\mathbf{b}_2 - \mathbf{r}_2$. With this new condition proposed by Reynolds $\hat{\mathbf{q}}_1$ and $\hat{\mathbf{q}}_2$ can now be written as:

$$\hat{\mathbf{q}}_1 = c_1^{-0.5}[(\mathbf{b}_1 - \mathbf{r}_1) \times (\mathbf{b}_2 - \mathbf{r}_2), (\mathbf{b}_1 + \mathbf{r}_1) \cdot (\mathbf{b}_2 - \mathbf{r}_2)] \quad (\text{D.17})$$

$$\hat{\mathbf{q}}_2 = c_2^{-0.5}[(\mathbf{b}_1 - \mathbf{r}_1) \times (\mathbf{b}_2 - \mathbf{r}_2), (\mathbf{b}_2 + \mathbf{r}_2) \cdot (\mathbf{b}_1 - \mathbf{r}_1)] \quad (\text{D.18})$$

Where the constant c is normalizes the quaternion. A more detailed derivation of these equations can be found in [43]. In addition, if equation (D.3) is satisfied, then both of the quaternion estimates are the same and are equal to:

$$\hat{\mathbf{q}}_3 = c_3^{-0.5}[(\mathbf{b}_1 - \mathbf{r}_1) \times (\mathbf{b}_2 - \mathbf{r}_2), (\mathbf{b}_2 \cdot \mathbf{r}_1) - (\mathbf{b}_1 \cdot \mathbf{r}_2)] \quad (\text{D.19})$$

The quaternion estimate given in equation (D.19) weighs both sets of body and reference vectors equally and assumes there are no measurement errors or noise. In some cases, this quaternion is assumed to be the "best estimate" and is chosen as the initial attitude estimate of a spacecraft. For the Asymmetric Direct Quaternion Method, the "more accurate" estimate $\hat{\mathbf{q}}_1$ is chosen instead and generally leads to better accuracy.

Comparison Between the Two Methods

The main differences between the TRIAD method and the Direct Quaternion method is the number of computations required to complete each algorithm and the error in the final estimate of the satellite's attitude. Dr. F. Landis Markley, a former engineer at NASA Goddard, compared the accuracy of various two vector attitude determination methods in [43]. The total number of computations required to complete each algorithm was reported by Markley as follows:

Algorithm	A Output	q Output
Asymmetric TRIAD	143	172
Asymmetric Direct Quaternion	-	< 46

Number of computations required for each algorithm.

And the estimation errors for each algorithm was reported as:

Algorithm	All Cases	$ \mathbf{q} > 0.5$	$ \mathbf{q} < 0.5$
Asymmetric TRIAD	4.6 (12.1)	4.5 (11.3)	4.7 (12.1)
Asymmetric Direct Quaternion	13.6 (2562)	5.2 (17.8)	20.1 (2562)

Average (maximum) estimation errors (arcseconds) for star tracker data.

According to these data provided in Markley's report, the TRIAD method requires more computations and is significantly more accurate than the Direct Quaternion method. Unless computing power is severely constrained, the TRIAD method should be chosen over the Direct Quaternion method.

Appendix E - Attitude Maintenance Phase

The sensors onboard the spacecraft are not perfect, as with any sensor. Which means that measurements of the spacecraft attitude have noise and errors. In a situation where a high degree of pointing accuracy or precise knowledge of the location of the spacecraft is required, a filter to remove noise is necessary to make accurate course corrections throughout the mission. One such filter is a Kalman Filter. A Kalman Filter creates a "best estimate" of the state of the spacecraft by comparing the sensor measurements to a state estimate created by using previous system knowledge and a model created by using the system dynamics of the CubeSat. The CubeSat system dynamics, Kalman Filter algorithm, and how a Kalman Filter is used to maintain spacecraft attitude are described in the following sections.

State Equation and System Dynamics

A state vector consists of a set of parameters that together completely describe the state of a system. In this specific case, the rotation and orientation completely describe the state of a spacecraft. The rotation of a satellite is given by its angular velocity about the x, y, and z axes of the body frame. The orientation of the spacecraft is given by the quaternion, which is a representation of the three Euler angles which also includes a fourth scalar term to avoid singularities at specific orientations. The angular velocity and unit quaternion of the spacecraft are written in the following forms:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (\text{E.1})$$

and:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \quad (\text{E.2})$$

Where the first three \mathbf{q} values describe the vector orientation of the spacecraft and the fourth q value is a scalar used to avoid singularities. Combining equations (E.1) and (E.2) provides the complete state vector of the satellite system:

$$\mathbf{X} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (\text{E.3})$$

Recall that during its estimation phase the Kalman filter utilizes the system dynamics of the spacecraft, or in other words, the equations that govern how the state of the satellite changes over time. This

means that the system dynamics equations of the satellite are given by taking the time derivative of the state vector. The derivative of the angular velocity vector, or angular acceleration, can be determined by using the equation (E.4):

$$\boldsymbol{\alpha} = \dot{\omega}_x \mathbf{i} + \dot{\omega}_y \mathbf{j} + \dot{\omega}_z \mathbf{k} + (\boldsymbol{\Omega} \times \boldsymbol{\omega}) \quad (\text{E.4})$$

Where the $\boldsymbol{\Omega}$ vector is the angular velocity of the moving body frame relative to the reference frame. If the moving body frame is rigidly attached to the spacecraft $\boldsymbol{\Omega}$ is equal to the angular velocity of the spacecraft, causing the cross product term to vanish. This implies that if the body frame stays constant, relative to the spacecraft, then the angular acceleration is found by taking the time derivative of each angular velocity component.

Since $\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$ are not available, an expression for the angular acceleration of the system can be derived by looking at the torque generated by a spacecraft. The net torque on a rotating spacecraft with a rigidly attached body frame is characterized by Euler's equation:

$$\boldsymbol{\tau}_{ext} = \dot{\mathbf{H}}_{rel} + (\boldsymbol{\omega} \times \mathbf{H}) \quad (\text{E.5})$$

Where \mathbf{H} is angular momentum vector and is related to the angular velocity of the spacecraft through the following expressions, where J_x , J_y , and J_z are the principal moments of inertia of the spacecraft:

$$\mathbf{H} = J_x \omega_x \mathbf{i} + J_y \omega_y \mathbf{j} + J_z \omega_z \mathbf{k} \quad (\text{E.6})$$

and:

$$\dot{\mathbf{H}}_{rel} = J_x \dot{\omega}_x \mathbf{i} + J_y \dot{\omega}_y \mathbf{j} + J_z \dot{\omega}_z \mathbf{k} \quad (\text{E.7})$$

Since the co-moving frame is rigidly attached to the spacecraft body, the equation for $\boldsymbol{\alpha}$ can be substituted in to the above equations for angular momentum:

$$\mathbf{H} = J\boldsymbol{\omega} \quad (\text{E.8})$$

and:

$$\dot{\mathbf{H}}_{rel} = J\boldsymbol{\alpha} \quad (\text{E.9})$$

Substituting equations (E.8) and (E.9) into Euler's equation of rotational motion provides the relationship between external torque (applied by the magnetic torquers) and angular velocity:

$$(\boldsymbol{\mu} \times \mathbf{B}) = J\boldsymbol{\alpha} + (\boldsymbol{\omega} \times J\boldsymbol{\omega}) \quad (\text{E.10})$$

Solving for the angular velocity, $\boldsymbol{\alpha}$, provides an equation for the rate of change of the angular velocity of the spacecraft system and the first portion of the system dynamics:

$$\boldsymbol{\alpha} = J^{-1}[(\boldsymbol{\mu} \times \mathbf{B}) - (\boldsymbol{\omega} \times J\boldsymbol{\omega})] \quad (\text{E.11})$$

The second portion of the system dynamics is given by the time derivative, or kinematics, of the unit quaternion. The equation for the quaternion kinematics is provided in [28], as the derivation is rather complex. This equation appears below:

$$\frac{d}{dt}(\mathbf{q}) = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \quad (\text{E.12})$$

Where $\Omega(\boldsymbol{\omega})$ is the 4x4 cross matrix of the angular velocity about the x, y, and z axes of the spacecraft body frame:

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \quad (\text{E.13})$$

Combining equations (E.11) and (E.12) provides the complete system dynamics, or the complete description of the rate of change of the spacecraft system:

$$\frac{d}{dt}\mathbf{X} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} J^{-1}[(\boldsymbol{\mu} \times \mathbf{B}) - (\boldsymbol{\omega} \times J\boldsymbol{\omega})] \\ \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \end{bmatrix} \quad (\text{E.14})$$

The Kalman filter uses these system dynamics equations to make a prediction of the updated state of the spacecraft's attitude. A description of the Kalman Filter and the associated algorithms is provided in the following sections.

Linearization of the System Dynamics

The equations describing the Kalman Filter estimation algorithm only hold when the predictive system model is linear. As seen in equation (E.14), the satellite system is nonlinear. To incorporate nonlinear systems into the estimation equations, a variation of the Kalman Filter called the Extended Kalman Filter (EKF) must be used. The EKF uses the nonlinear system dynamic model to make predictive updates of the state estimate, while it uses linear approximations of the dynamical equations to calculate the Kalman gain and error covariance matrix.

In many cases, nonlinear system equations are linearized by using the first two terms in the Taylor series of said function (assuming that higher order terms have little effect on the approximation). The general form of the Taylor series approximation of a function (using the first two terms) is given by the

following:

$$f(x) = f(x_0) + \frac{df(x_0)}{dx}(x - x_0) \quad (\text{E.15})$$

where the derivative term can also be written as:

$$f(x) = f(x_0) + \Delta_x f(x_0)(x - x_0) \quad (\text{E.16})$$

As stated previously, to apply the Kalman Filter to the satellite system equation (E.14) must be made linear by using a Taylor series approximation. The expanded vector form of equation (E.14) appears below:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{u}) = \begin{bmatrix} -\frac{1}{I_x}(B_y\mu_z - B_z\mu_y - I_y\omega_y\omega_z + I_z\omega_y\omega_z) \\ \frac{1}{I_y}(B_x\mu_z - B_z\mu_x - I_x\omega_x\omega_z + I_z\omega_x\omega_z) \\ -\frac{1}{I_z}(B_x\mu_y - B_y\mu_x - I_x\omega_x\omega_y + I_y\omega_x\omega_y) \\ \frac{1}{2}(q_4\omega_x - q_3\omega_y + q_2\omega_z) \\ \frac{1}{2}(q_3\omega_x + q_4\omega_y - q_1\omega_z) \\ \frac{1}{2}(-q_2\omega_x + q_1\omega_y + q_4\omega_z) \\ \frac{1}{2}(-q_1\omega_x - q_2\omega_y - q_3\omega_z) \end{bmatrix} \quad (\text{E.17})$$

Where \mathbf{X} is the state vector of the system given by equation (E.14) and \mathbf{u} is the control input of the system given by the magnetic moment vector $\boldsymbol{\mu}$. The Taylor series approximation for the satellite system dynamics equation can be written in the following form:

$$f(x) = f(x_0, u_0) + \Delta_x f(x_0, u_0)(x - x_0) \quad (\text{E.18})$$

Assuming that the external torque is independent of the system state (i.e. $\boldsymbol{\mu} \times \mathbf{B}$ does not depend on $\boldsymbol{\omega}$ or \mathbf{q}), this expression is expanded into the following form, which is known as the Jacobian of a matrix:

$$\begin{aligned} \Delta_x f(\mathbf{X}, \mathbf{u}) &= \begin{bmatrix} -J^{-1}[\boldsymbol{\omega}^\times]J & 0_{3 \times 4} \\ \frac{1}{2}\Xi(\mathbf{q}) & \frac{1}{2}\Omega(\boldsymbol{\omega}) \end{bmatrix} \\ &= \begin{bmatrix} 0 & \frac{\omega_z(I_y - I_z)}{I_x} & \frac{\omega_y(I_y - I_z)}{I_x} & 0 & 0 & 0 & 0 \\ -\frac{\omega_z(I_x - I_z)}{I_y} & 0 & -\frac{\omega_x(I_x - I_z)}{I_y} & 0 & 0 & 0 & 0 \\ \frac{\omega_y(I_x - I_y)}{I_z} & \frac{\omega_x(I_x - I_y)}{I_z} & 0 & 0 & 0 & 0 & 0 \\ \frac{q_4}{2} & -\frac{q_3}{2} & \frac{q_2}{2} & 0 & \frac{\omega_z}{2} & -\frac{\omega_y}{2} & \frac{\omega_x}{2} \\ \frac{q_3}{2} & \frac{q_4}{2} & -\frac{q_1}{2} & -\frac{\omega_z}{2} & 0 & \frac{\omega_x}{2} & \frac{\omega_y}{2} \\ -\frac{q_2}{2} & \frac{q_1}{2} & \frac{q_4}{2} & \frac{\omega_y}{2} & -\frac{\omega_x}{2} & 0 & \frac{\omega_z}{2} \\ -\frac{q_1}{2} & -\frac{q_2}{2} & -\frac{q_3}{2} & -\frac{\omega_x}{2} & -\frac{\omega_y}{2} & -\frac{\omega_z}{2} & 0 \end{bmatrix} \quad (\text{E.19}) \end{aligned}$$

When the Jacobian matrix is linearized about the current state estimate \mathbf{X} the matrix in equation

(E.19) becomes known as the A matrix. Depending on the control law chosen for the external control torque, τ_c (denoted as $\mu \times B$ in previous equations) may depend on ω and q . This relationship will change the A matrix seen in equation (E.19).

Design of the Extended Kalman Filter

As stated in the previous section, since the system dynamics equations are nonlinear, a linear Kalman Filter cannot be used for this application. An Extended Kalman Filter will be used to incorporate the nonlinear characteristics of the system instead. The first step to designing an Extended Kalman is to determine the nonlinear plant equation, also known as the ideal dynamic model of the system. The nonlinear plant equation for the system state is written in the following form:

$$\dot{\mathbf{X}}^{true} = f(\mathbf{X}^{true}, \mathbf{u}, \mathbf{w}, t) \quad (\text{E.20})$$

The \mathbf{X} vector is the ideal value of the state vector given by equation (E.14), \mathbf{u} is the control input vector, and \mathbf{w} is the process noise. The state measurements are modeled using the plant equation in the following manner:

$$\mathbf{y} = h(\mathbf{X}^{true}) + \boldsymbol{\nu} \quad (\text{E.21})$$

The vector \mathbf{y} models the measurements provided by the satellite sensors. The function h is (generally) a nonlinear function that relates the state vector variables to the sensor measurements and the $\boldsymbol{\nu}$ represents the measurement noise. While in general the relationship between measurements and the state vector may be nonlinear, this is not the case for this project. The sensors on the spacecraft measure the state vector directly. The three measurements for the angular velocity are given by the gyroscope and a virtual quaternion measurement is calculated by performing the TRIAD algorithm on the current sun sensor and magnetometer measurements. This means that:

$$\mathbf{y} = C\mathbf{X}^{true} + \boldsymbol{\nu} \quad (\text{E.22})$$

Where the matrix C is a 7x7 identity matrix. This also means that the measurement model is a linear function. In MATLAB® simulations, the values for \mathbf{w} and $\boldsymbol{\nu}$ are modeled as $\sigma * randn$.

Now that the plant and measurement model are determined, the filter cycle of the EKF can be characterized. The following equations describe the change in the Kalman gain (K) and estimation error covariance (P) for a continuous time EKF, where Q and R are process and measurement error covariance matrices:

$$\dot{P} = AP + PA^T + Q \quad (\text{E.23})$$

$$K = PC^TR^{-1} \quad (\text{E.24})$$

The estimation error covariance is found by solving equation (E.23) for P using an initial condition $P(t_0) = P_0$ over the measurement time interval. The initial estimation error covariance is given by the equation $E(\epsilon\epsilon^T)$ where ϵ is $\mathbf{X}^{true} - \hat{\mathbf{X}}_0$. Once the Kalman gain is calculated an updated state estimate can be calculated:

$$\dot{\hat{\mathbf{X}}} = f(\hat{\mathbf{X}}, \hat{\mathbf{u}}) + K(\mathbf{y} - C\hat{\mathbf{X}}) \quad (\text{E.25})$$

In practice, measurements are not taken continuously. So, it is common to use a discrete Kalman Filter instead. For a discrete time system, the linear equations determined previously are not necessarily valid. A linear approximation of a discrete time system requires the discrete equations of motion, which in many cases can be quite difficult to determine analytically. Fortunately, these equations can be avoided by using a continuous-discrete Extended Kalman Filter in favor of a true discrete and autonomous Kalman Filter.

A continuous-discrete Kalman filter updates the state estimate, error covariance, and system measurements after a specified time interval t has passed. Between the time t_0 and t^- the previous state estimate and error covariance are propagated by numerically solving differential equations. Before the start of the time interval, the initial conditions of the system are quantified. At the beginning of the first time interval, the initial state estimate is taken from the output of the initial TRIAD calculation and used to calculate the initial error covariance. These values are then propagated throughout the specified time interval using the following equations:

$$\dot{\hat{\mathbf{X}}} = f(\hat{\mathbf{X}}, \hat{\mathbf{u}}, \mathbf{w}, t) \quad (\text{E.26})$$

$$\dot{P} = AP + PA^T + Q \quad (\text{E.27})$$

At the instant before the end of the interval (i.e. the instant before system measurements are updated), the final values of the propagation of equations (E.26) and (E.27) above are used to calculate the current Kalman gain, K , of the system. These estimates are denoted with a (-) since they were determined prior to receiving updated measurement data.

$$K = P^- C^T [CP^- C^T + R]^{-1} \quad (\text{E.28})$$

The final state estimate and error covariance for the time interval are now calculated using the Kalman gain, K , and the system measurements, \mathbf{y} . These values, known as the updated system estimates, are denoted by a (+). The (+) indicates that these estimates were made using updated system measurement data, an instant after the end of the time interval, t .

$$\hat{\mathbf{X}}^+ = \hat{\mathbf{X}}^- + K[\mathbf{y} - C\hat{\mathbf{X}}^-] \quad (\text{E.29})$$

$$P^+ = [I - KC]P^- \quad (\text{E.30})$$

The estimated state and covariance determined from equations (E.29) and (E.30) are the best estimates of the system at time t and are the final outputs of the Extended Kalman Filter. In addition to serving as the output, these values are set as the initial system estimates for the next time interval and the process repeats.

Using an Extended Kalman Filter to Maintain Spacecraft Attitude

The CubeSat mission requirements state that the solar panels and payload must be pointed towards the sun with a high degree of accuracy. This requirement is achievable only if the on-board sensors provide a nearly noise free measurement of the attitude and state of the satellite. Since, in reality, sensors have noise and errors in their measurements a Kalman Filter is used to remove noise and provide an accurate estimation of the true state of the spacecraft. The state estimate form the Kalman Filter is compared to the desired attitude of the spacecraft to determine the error in attitude.

While the Kalman Filter can help detect an error, a controller is used to correct that error. The best controller for this situation would be a Proportional-Derivative (PD) controller. The PD controller corrects both the steady-state error, or orientation, and the rate of change of the error, or angular velocity, of the satellite. The spacecraft orientation, given by the quaternion, and the spacecraft angular velocity are readily available in the state estimate \mathbf{X} , so no additional calculations are required to implement this controller. This means that in this situation, the PD controller outperforms the P controller in terms of accuracy and outperforms the PI and PID controllers in terms of computational effort. The general form of the PD controller for the CubeSat is:

$$\tau_c = -k_p \delta \hat{\mathbf{q}}_{1:3} - k_d \hat{\omega} \quad (\text{E.31})$$

Where $\hat{\omega}$ is the estimated angular velocity and $\delta \hat{\mathbf{q}}_{1:3}$ is the vector portion of the estimated error quaternion defined by $\delta \hat{\mathbf{q}} = \hat{\mathbf{q}} \otimes \mathbf{q}_{desired}^{-1}$. The estimated error quaternion and estimated angular velocity are determined by using the EKF and the desired orientation is given by the mission requirements. The proportional and derivative gain are determined through simulation and testing. The required μ can be determined by expanding equation (3.2) and solving for μ_x , μ_y , and μ_z simultaneously.

Appendix F - MATLAB® Simulation Guide

The MATLAB® simulation guide developed throughout this project begins on the following page.

MATLAB SIMULATION GUIDE

Simulation NOTES:

- 1. MATLAB ADCS 2016 code currently works with MATLAB 2016a**
- 2. SatSim is used to run simulation**
- 3. Do not edit any files within GUI folder unless you have experience with editing GUI's**
 - a. If you do have experience, then you might consider adding controls to do graphs utilizing the power and current scripts**
- 4. Power and Current Scripts utilize a counter "j", to simulate the power and current usage:**
 - a. Go to stabilizer script and follow directions**
 - b. Go to end of Mag script and uncomment the appropriate plots**
 - c. Every time the power and current plots are made, type initialize into command window to re-set the counter "j"**
- 5. The sensors are simulated using data collected from the orbit in Systems Tool Kit (STK)**
 - a. Therefore, if the orbital team makes any changes to orbit, this data has to be recollected for proper simulation**
 - i. This data has to be integrated into the models folder**

- b. Noise is introduced using standard deviation. If any of the sensors are changed, then the standard deviations of the new sensors have to be integrated into the SimProps.m file
- 6. Mag.m is responsible for most of the simulation
 - a. Study this file extensively as early as possible, it will help you figure out how the simulation is done
- 7. Any changes to the structure will result in changes to the Inertia Matrix
 - a. The Inertia Matrix is crucial to simulation, so this needs to be updated as soon as possible for accurate results
- 8. The TRIAD (TRIAD_est.m) is used as the initial estimate
 - a. Results from this are fed into the attitude maintenance methods including EKF.m and LowPass.m
- 9. Any new changes to actuators will affect simulation
 - a. Specifically, if using only magnetometers:
 - i. Change the max moment and power usage in GUI to match that of new magnetometer
 - b. If other actuators are added, then the simulation code has to be edited to adjust for this

- i. The moment produced by the new control laws for other actuator systems has to be added to the existing B_dot controller (Stabilizer.m)

10. Plots within the GUI

- a. The plots of the angle error are used to test the determination methods
- b. The plots of ang velocity are used to test the controllers
- c. The plots of exact angles are used to simulate the body angles of the CubeSat

11. The time step used within the GUI represents time in between sensor readings

Brief Explanation of all scripts:

Globals:

- **IGRF**=comes from the **MAG.csv** spreadsheet (simulated magnetic field data from STK).
- **SUNLIGHT** =comes from **Sunlight.csv** –Sunlight file is the raw sun data without noise added.
- **NADIR** = comes from the **NADIR.csv** file generated by STK
s0= initial state vector. This is a 7 x 1 matrix, with the first 3 entries being angular velocity components $\omega_1, \omega_2, \omega_3$ and the last 4 entries being the quaternion components $q1, q2, q3$, and $q4$.
- **Mu**=3 x 1 vector containing magnetorquer moment values.
- “**B**”=magnetic field vector given by **sensor_magnet** (t0, ‘real’, ‘body’)
- “**W**”= angular velocity given by **sensor_gyro** (‘real’) (3 x 1 matrix)
- “**S**”= sun vector given by **sensor_sun** (t0, ‘real’, ‘body’)
- **DT**= 3 x 1 matrix defined in **Mag.m**-desired torques from magnetorquer.
- **Kds**=3 x 1 matrix defined in **Mag.m**
- **InSun**=0 matrix defined in **Mag.m**
- **Props**

- **S_est**= estimated state vector

Control:

PD.m

Gives **mu**, the 3 x 1 vector containing magnetorquer moment values. Uses Globals “**B**”, **Props**, **DT**, **Kds**, **InSun** and inputs **t**, current time in simulation, **des_q**, the desired quaternion, and “**s**”, the current state of the system. Uses a proportional constant **Kp** and a derivative constant **Kd**. Defines desired torques from magnetorquer **DT**. **Kp** value can be changed in the Satsim PD GUI, **Kd** is currently a variable gain with the applicable range only capable of being changed through hard coding, the **Kd** value from the Satsim PD Gui only changes the **Kd** used during eclipse.

Stabilizer.m

The stabilizing controller aims to slow the satellite during de-tumbling. Globals used are **Props**. Produces an equation
mu=stabilizer(“s”, “B”)=Props.K_Stab.*cross(“B”, “w”) where **K_Stab** is the stabilizing constant, and “**w**” is the 3 x 1 vector containing magnetorquer moment values.

Determination:

EKF.m

Globals: **s_est**, **P_ekf**, **mu**, **Props**, “**B**”, **F_ekf**, “**S**”, **s0**, **InSun**.

Uses the inertia tensor set, the current quaternion and angular velocity to set up the predict and filter cycles of the EKF outputting the new state estimate.

Defines **w** as rows 1-3 of **s_est**, and **q** as rows 4-7.

Uses a predict cycle utilizing “**B**” and **mu** for use in the Euler equation “**E**” with **qd**, **s_est**, and **P_ekf**. Uses a filter cycle which uses the TRIAD estimate **y=TRIAD_est(t)**, **s_est**, **InSun** to output a new vector “**s**”, **s=s_est**

intEKF.m

Uses the state vector estimate from TRIAD **s_est=TRIAD_est(t)** and acceleration “**a**” from **gyro_std**, along with **P_ekf** as a diagonal matrix. Used **F_ekf=jacobian()** to produce a new **F_ekf**. Uses the initial information from triad and the angular acceleration to begin the EKF knowledge.

LowPass.m

Gives updated estimated state vector **est_s_new** from the old state vector estimate **est_s_old=est_s** using the Low Pass Filter
est_s_new=LowPass(count, est_s) to filter the noise introduced by sensors.

TRIAD_est.m

Produces the state vector estimate $s=TRIAD_est(t)$ based on “B” and “S” for the body vector triad and **model_B=model_magnet(t,’ideal’)** as the magnetic field model and **model_S=model_sun(t,’ideal’)** as the sun vector model for the inertial vector triad. These two triads are utilized by the direction cosine matrix **q=q_from_dcm(A)** to produce the 7 x 1 state vector S, where the first three rows are “W”.

Math:

Angle_diff.m

Quantifies the change in the body angle experienced.

Angular_error.m

Determines unit vector angular error **uv=angular_error(v,rtype,angle)** .

Where the direction vector for axes are defined for angular error estimations.

Dcm_from_q.m

Calculates the direction cosine matrix **Q=dcm_from_q(q)** from the quaternion values $q1, q2, q3$, and $q4$, where $q4$ is scalar.

Delta_q.m

Calculates the difference in quaternion values **q=delta_q(des,act)** from the last reading based in acceleration.

q_from_dcm.m

Calculates the quaternion **q=q_from_dcm(Q)** from the direction cosine matrix **Q**.

q_to_ypr.m

Converts quaternion q to yaw, pitch, and roll angles using **ypr=q_to_ypr(q)**

QXx_ypr.m

Converts yaw, pitch, and roll angles to components of the direction cosine matrix **Q=QXx_ypr(y,p,r)** from the inertial frame to the body frame

Wx_to_wypr.m

Calculates body frame velocities (**wx**, **wy**, **wz**) from angular velocity measurement **w** and yaw, pitch, and roll (**ypr**) 3 x 1 matrix.

Ypr_to_q.m

Converts yaw, pitch, and roll angles to direction cosine matrix **Q** using
Q=yrp_to_q(ypr)

Models:

Model_desq.m

Gives the model for the Desired Quaternion (**dq**) using the sun model (**s=model_sun**) and the nadir model (**n=model_nadir**).

Model_magnet.m

Gives function modeling magnetic field (**m=model_magnet(t, mode)**) using the **IGRF** magnetic field data and added noise based on the magnetometer specifications.

Model_nadir.m

Gives function modeling nadir (**q=model_nadir(t, mode, frame, form)**) using the **NADIR** data, adding noise, and converting data to the satellite body frame using **QXx=dcm_from_q**.

Model_sun.m

Gives function modeling sun vector (**s=model_nadir(t, mode)**) using the **SUNLIGHT** data and added noise from the sun sensor specifications.

NADIR.csv

NADIR data generated by STK. Used for the Global NADIR.

Sensors:

MAG.csv

Magnetic Field data generated by STK. Used for the **IGRF** Global.

Sensor_gyro.m

Input the initial angular velocity vector **w0**, which is the first three entries of the initial state vector **s0**. Adding noise to **w0** for the real case gives **w**, 3 x 1 vector containing magnetorquer moment values. There should be some initial angular velocity for detumble simulations and none for attitude maintenance simulations.

Sensor_magnet.m

Uses **IGRF** and **s0** to generate magnetic field vector **m** using the function **m=sensor_magnet(t,mode,frame)** and noise added based on the magnetometer specifications. Input simulation time **t**, the mode (either ‘ideal’ or ‘real’, and the ‘inertial’ or ‘body’ frame. Conversion to body frame uses **QXx=dcm_from_q**.

Sensor_sun.m

Uses **SUNLIGHT** and **s0** to generate sunlight vector **s** using the function **s=sensor_sun(t,mode,frame)** and noise added based on the sun sensor specifications. Input simulation time **t**, the mode (either ‘ideal’ or ‘real’, and

the ‘**inertial**’ or ‘**body**’ frame. Conversion to body frame uses
QXx=dcm_from_q.

Sunlight.csv

Sunlight data generated by STK.

Other:

Inertia.m

Input **u** of CubeSat (1,2, or 3), **cx**, **cy**, **cz** (x, y, and z positions of the center of gravity) to get the inertia tensor **I** about a point.

Initialize.m

Initializes magnetic moment (**mu**) and loads data to STK where:

IGRF reads **MAG.csv** –MAG file becomes the raw IGRF data without noise added.

SUNLIGHT reads **Sunlight.csv** –Sunlight file is the raw sun data without noise added.

NADIR reads **NADIR.csv** –NADIR becomes the nadir reading.

Jacobian.m

Gives the Jacobian matrix for desired state vector **s**. Outputs the derivative of the state matrix.

Mag.m

This is the function for solving magnetorquer simulation equations. Outputs are **yc** (state positions yaw, pitch, roll, yaw_dot, pitch_dot, roll_dot), **tc** (time vector corresponding to **yc**), **Uvals** (values of the magnetorquers (**mu**)), and **E** (energy of the maneuver). Input is **desired** (desired yaw, pitch, roll

position), and **s0**. This handles most of the simulation, so first study this script if you want to have a grasp on how the simulation is done.

Magsystem.m

Calculates the time derivative of the state vector **s_dot**

Model.m

Displays rotating model on GUI.

Rot_mod.m

Rotates model on GUI.

Satsim.m

Run “satsim” to display GUI/run sims. Must have all paths added in the directory to run satsim GUIs

SimProps.m

Describes property inputs for GUI

GUI:

All GUI files are written to create the GUI input options. “Save as Default” chosen in GUI does NOT change the hard coded values in the related file.

EKFEdit.m

Creates a new EKFEDIT from the EKFEdit for DCM and Rotation matrix.

Not to be edited.

IntertiaMatrix.m

Creates new inertia matrix and updates/pulls from the set GUI inertia matrix

InitialConditions.m

Not to be edited, sets up, keeps and distributes the inertia matrix

LPFEedit.m

Not to be edited, controls the LPF GUI and distributes the information.

Magnetotorquers.m

Not to be edited, controls the magnetotorquer GUI and distributes the information.

Main.m

Not to be edited, controls the main satsim GUI and distributes the information.

PDEdit.m

Not to be edited, controls the PD GUI and distributes the information.

RandomError.m

Not to be edited, controls the random error GUI and distributes the information.

SimTime.m

Not to be edited, controls the simulation time GUI and distributes the information. NOTE: for best results the determination and control simulations must be run with no more than a 1 second time step.

Appendix G - STK Data

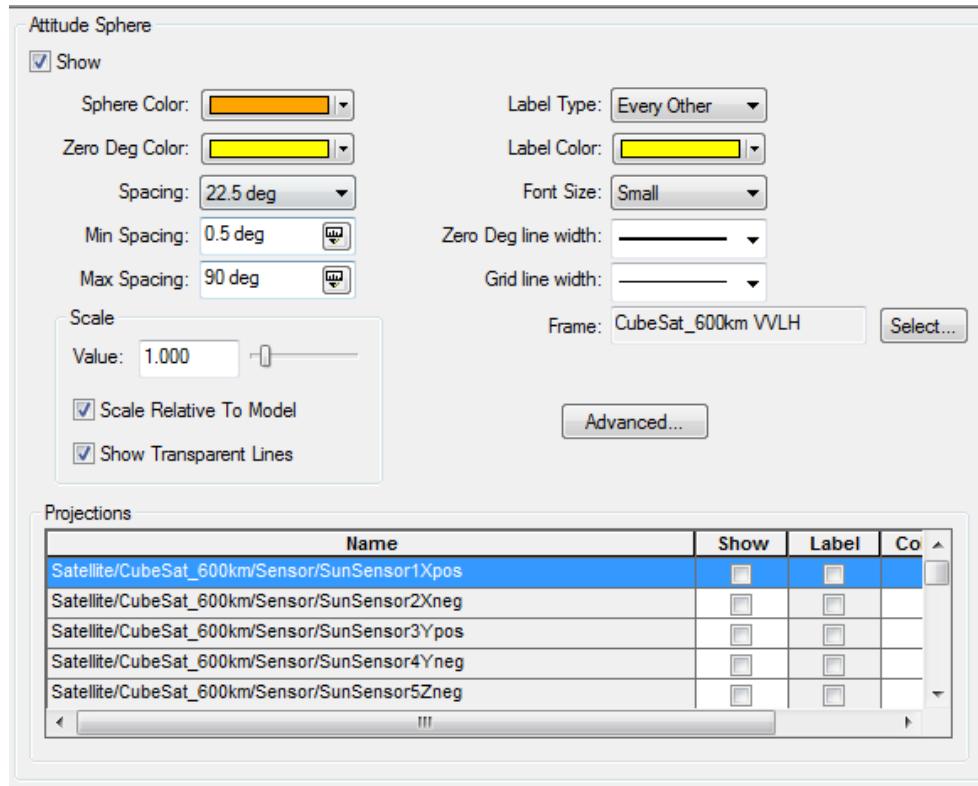
The table below demonstrates the data exported by STK into text format, which was then imported into Excel and reorganized.

Time (s)	q1	q2	q3	qs	Wx	Wy	Wz
4209.00	0.99097838	-0.00000051	-0.00000008	0.13402180	4.99998235	-0.00000002	-0.00000001
4225.54	0.83242203	-0.00000043	0.00000027	-0.55414219	4.99998219	-0.00000001	0.00000001
4260.85	-0.52856966	0.00000026	0.00000044	-0.84888993	4.99998187	0.00000000	0.00000000
4295.09	-0.88688888	0.00000046	-0.00000022	0.46198280	4.99998156	-0.00000001	0.00000000
4329.28	0.39062443	-0.00000018	-0.00000047	0.92055014	4.99998126	-0.00000001	0.00000001
4362.94	0.95552513	-0.00000049	0.00000014	-0.29490968	4.99998097	0.00000000	-0.00000001
4396.07	-0.17312086	0.00000008	0.00000050	-0.98490059	4.99998070	-0.00000002	0.00000000
4428.30	-0.99995442	0.00000051	0.00000000	0.00954748	4.99998045	0.00000001	0.00000000
4450.59	-0.55523018	0.00000028	-0.00000042	0.83169673	4.99998029	-0.00000001	-0.00000002
4476.48	0.51496589	-0.00000026	-0.00000043	0.85721067	4.99998011	-0.00000002	0.00000002

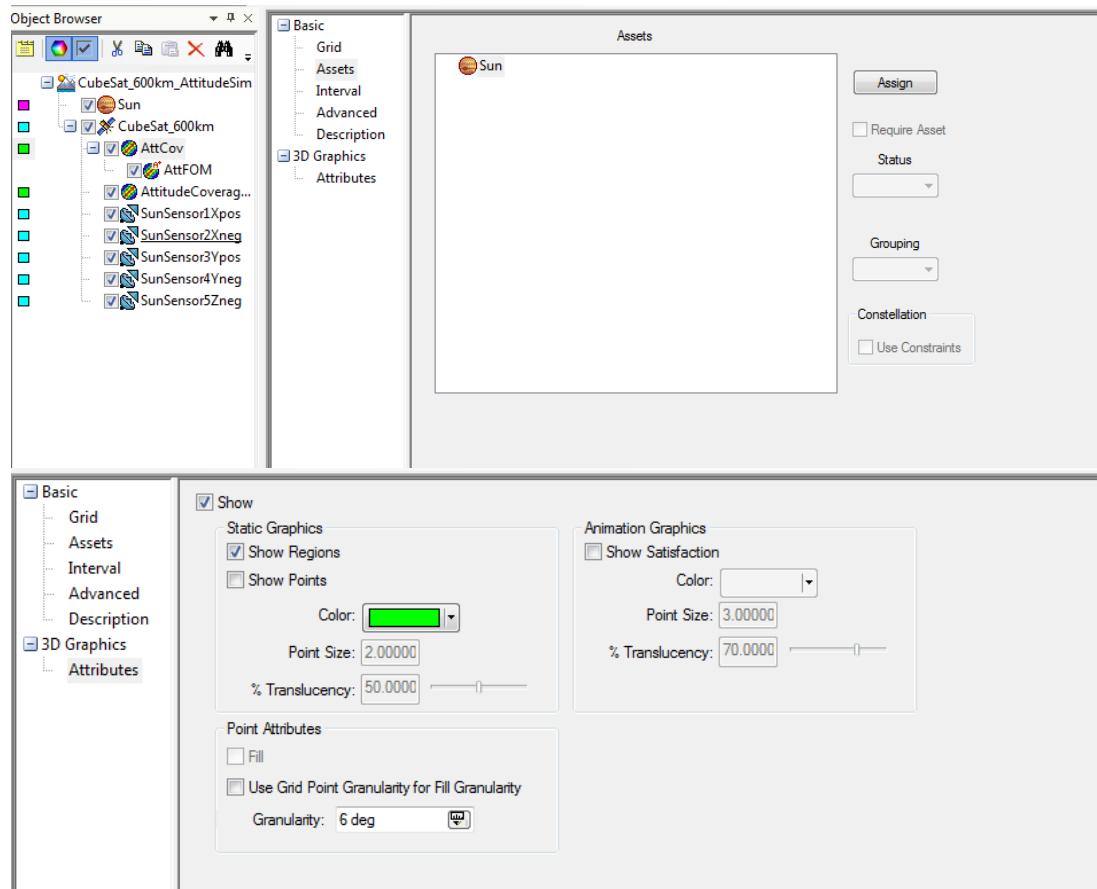
Appendix H - STK Setup Screen Shots

Screen shots of the settings required for STK simulation appear on the following pages.

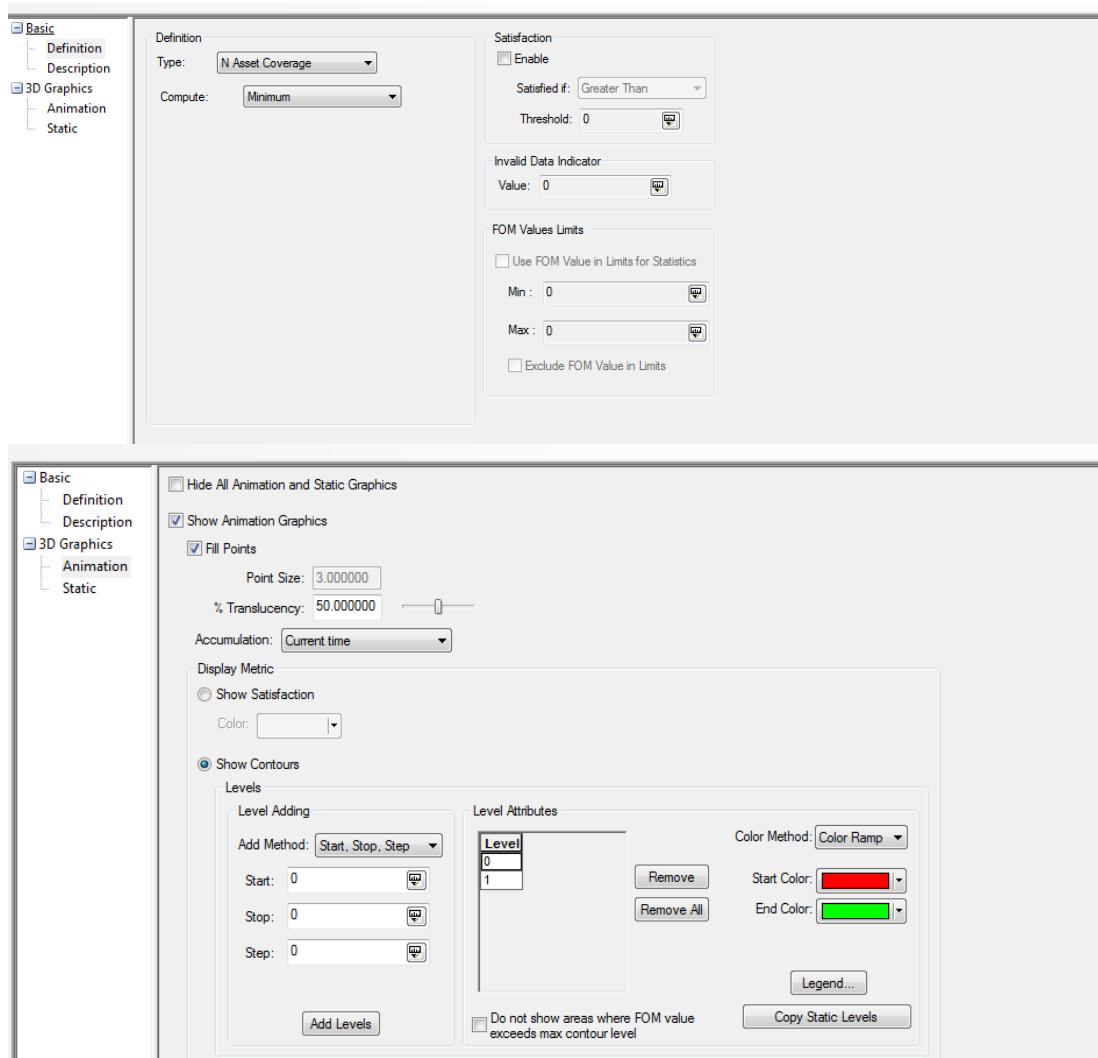
STK attitude sphere settings.

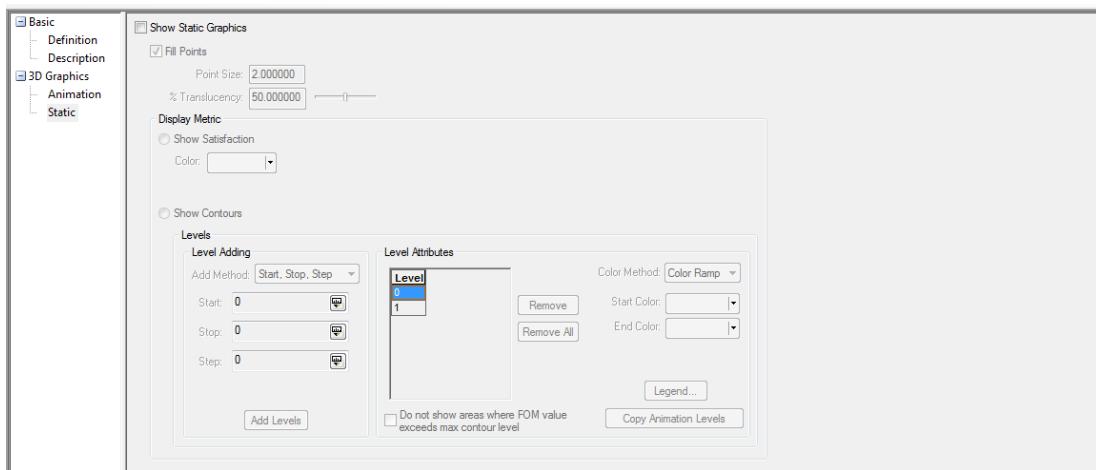


STK attitude coverage scenario settings shown below.

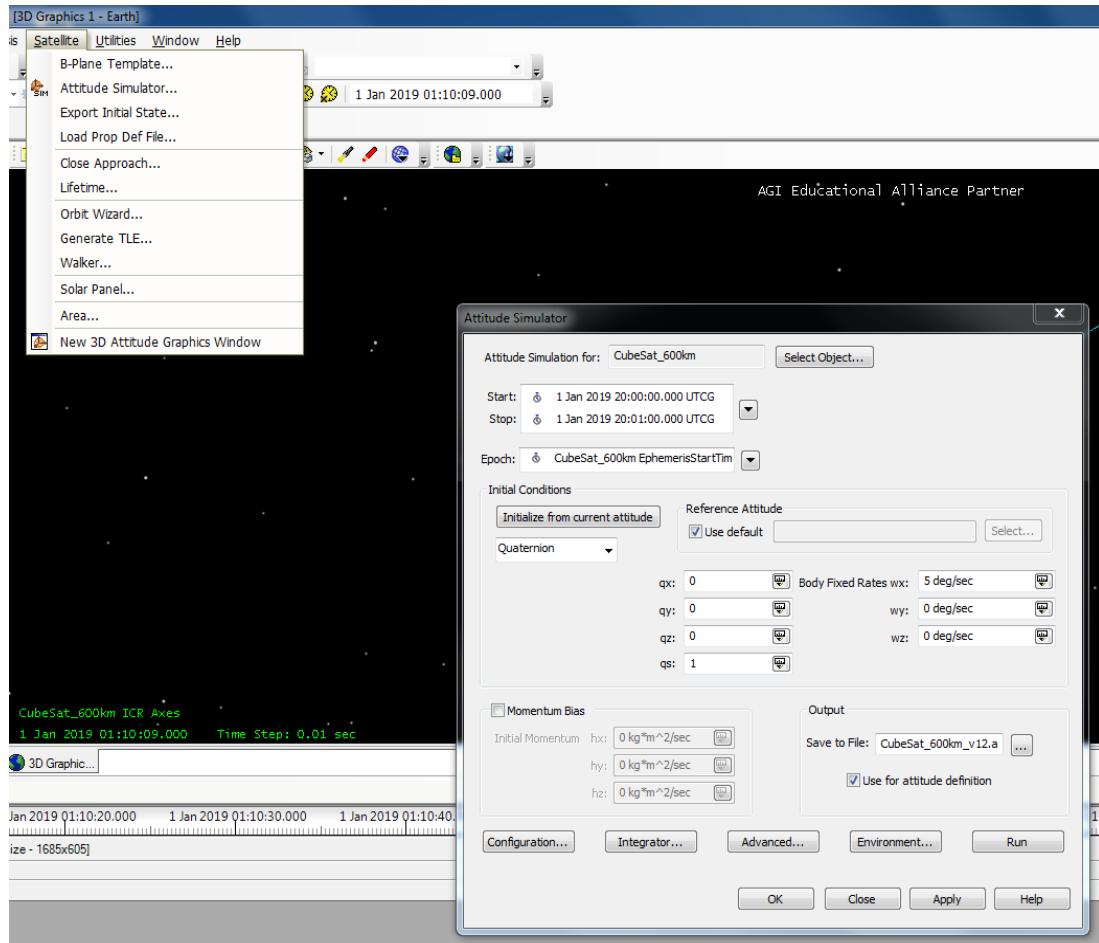


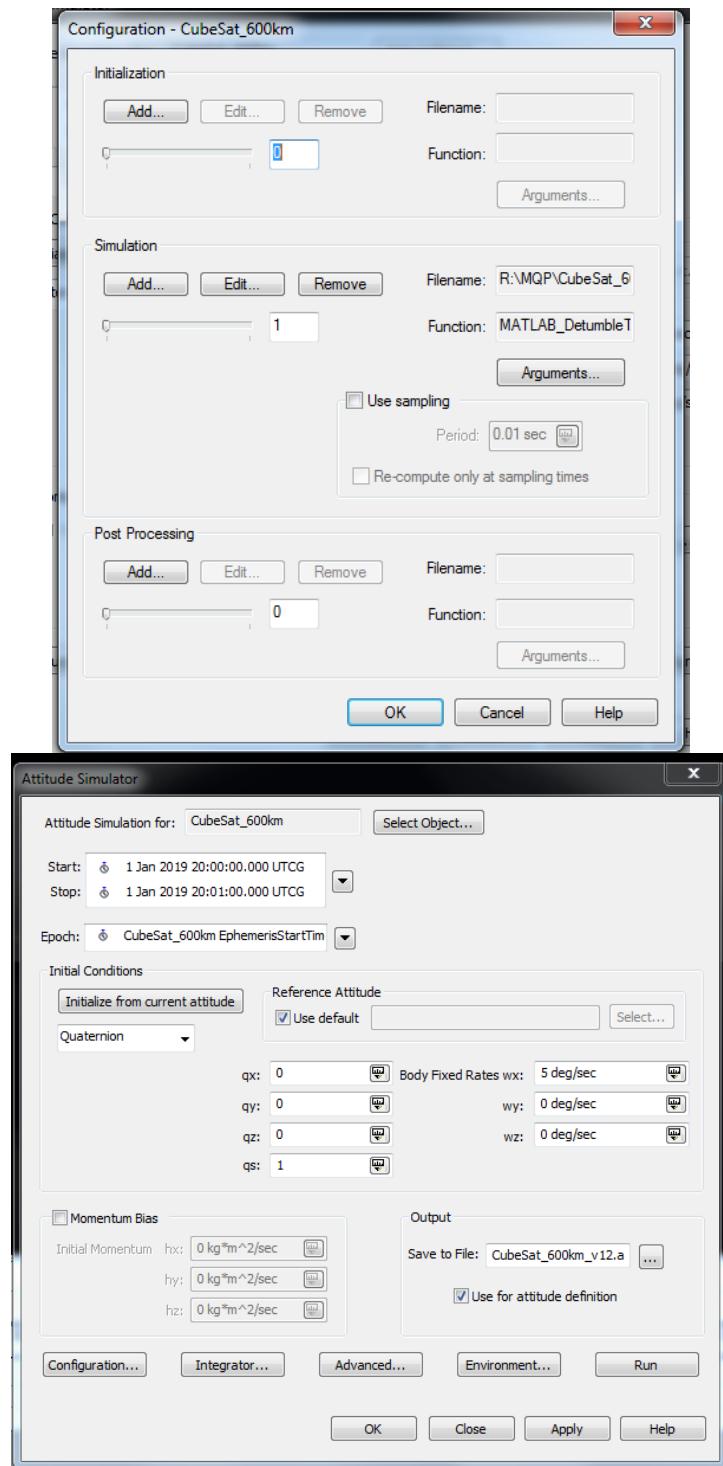
Attitude Figures of Merit (FOM) setup.





STK simulation procedure.

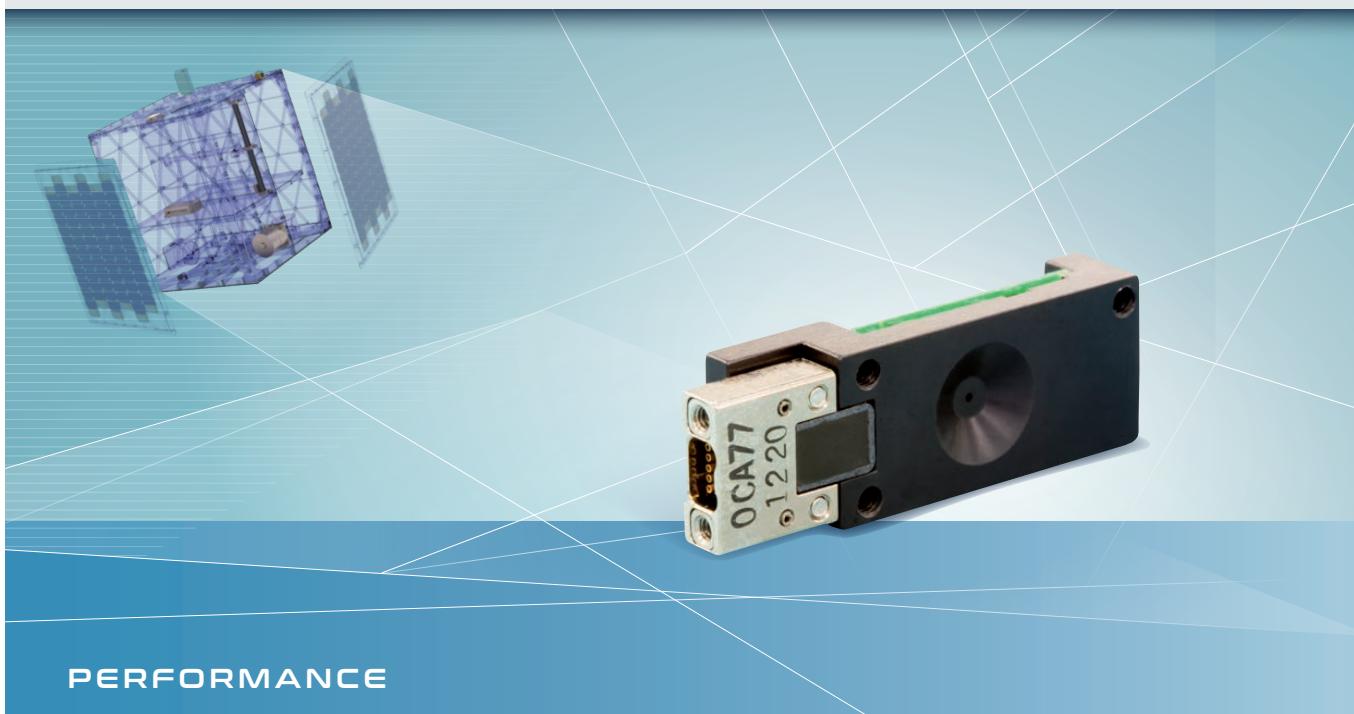




Appendix I - Fine Sun Sensor Data Sheet

The data sheet for the fine sun sensor appears on the next page.

CUBESAT SUN SENSOR



PERFORMANCE

CubeSat Sun Sensor	
FUNCTIONAL CHARACTERISTICS	
Field of view	114°
Update rate	> 10 Hz (limited by customer ADC)
Accuracy	< 0.5° (ms error over Fov)
PHYSICAL CHARACTERISTICS	
Mass	< 5 grams
Power	< 10 mA
Size	33mm x 11mm x 6mm
ENVIRONMENTAL CHARACTERISTICS	
Thermal (operational)	-25°C to +50°C
Vibration (qualification)	20g rms random
INTERFACES	
Power Supply	5 V
Data	5 analogue channels
Connector	9 way female Nano-D
Mechanical	3x M2 threaded holes

ACCEPTANCE TESTING: All parts undergo random vibration (10 rms) as well as thermal cycling (four cycle ambient pressure) to five degrees beyond operational thermal specifications. However, NewSpace can perform additional environmental testing if required by a client.

FEATURES

- PSD architecture
- Good accuracy
- Wide field of view
- Ultra-small size and low mass
- Low power
- Simple analogue interface

APPLICATIONS

- Accurate determination of sun-angle
- Six sensors can achieve full sky coverage
- Used in conjunction with a magnetometer for simple attitude control
- Can be used as safe-mode sensors on gyro or star-mapper controlled systems
- High performance CubeSat ACS systems

QUALIFICATION

More than one hundred CubeSat Sun Sensors have been delivered to a multitude of international satellite programmes.



Appendix J - Coarse Sun Sensor Data Sheet

The data sheet for the coarse sun sensor appears on the next page.

CSS-01,02 Coarse Sun Sensors

Space Micro celebrates its 13-year anniversary in 2015 and continues to support the Space Industry with innovative, affordable and high performance Digital/Image Processing, RF Communication and Attitude Determination Sensor Products.

Space Micro's Coarse Sun Sensor is a very low-cost, high reliability device with >20 years of flight heritage. They are designed to provide attitude determination information during various mission modes including launch vehicle separation, initial attitude acquisition, emergencies and off-nominal modes. They can also provide coarse sun vector determination as a backup to higher resolution sensors.

The Coarse Sun Sensor contains a single photodiode, with the housing assembly also serving as an aperture. The inexpensive, lightweight sensor (only 20g) draws no power and has an accuracy of better than ± 5 degrees over a full angle 120 degree field of view. Sensors are available in panel mount or through-hole configurations.

The Coarse Sun Sensors have flown successfully on multiple spacecraft, including ALEXIS, HETE, MOST, CHIPSat, STPSat-1. and are compatible with virtually all launch environments.

FEATURES

- >20 Years of Flight Heritage
- Very low size and weight
- Compatible with all launch environments
- Standard lead time and pricing
- Back (CSS-01) and front (CSS-02) mount configurations
- Simple reliable design

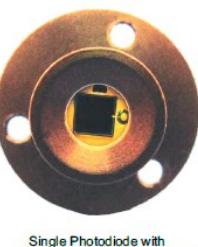
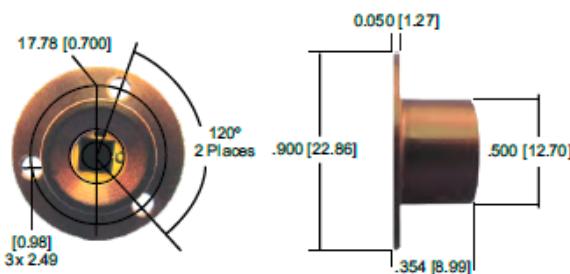


CSS-01,02 Coarse Sun Sensors

SPECIFICATIONS

Field of View	120° full-angle circular field of view
Accuracy	±5° of 1-axis knowledge
Temperature Range	-40 to +93°C
Vibration Test Levels	14.1 grms
Shock Test Levels	60g protoqual
Interface	0 to 3.5 mA (typical) current sources on two flying leads: 50" (1.27m) in length, M22759/ 33-26, 26 AWG wire
Mounting	Three #2 through holes, 120° apart on a .700" (1.78 cm) diameter pattern (See Figure below).
Power	None required
Size	
Housing diameter	.500" (1.27cm)
Flange diameter	.900" (2.286 cm)
Sensor height	0.354" (.899 cm)
Volume	0.500" (1.27 cm) diameter × .354" (0.90 cm) height
Mass	0.022 lbs (10g) with 1.27 cm flying leads

DIMENSIONS



Single Photodiode with
Housing Serving as Aperture

10237 Flanders Court
San Diego, CA 92121
Tel: 858.332.0700
Email: sales@spacemicro.com
www.spacemicro.com

 **SPACE MICRO**

Appendix K - Magnetometer Data Sheet

The data sheet for the magnetometer appears on the next page.

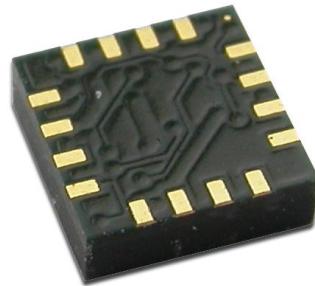
3-Axis Digital Compass IC

HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I²C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magnetoresistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES

BENEFITS

- | | |
|--|--|
| ► 3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package | ► Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors
Designed for High Volume, Cost Sensitive OEM Designs
Easy to Assemble & Compatible with High Speed SMT Assembly |
| ► 12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields | ► Enables 1° to 2° Degree Compass Heading Accuracy |
| ► Built-In Self Test | ► Enables Low-Cost Functionality Test after Assembly in Production |
| ► Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA) | ► Compatible for Battery Powered Applications |
| ► Built-In Strap Drive Circuits | ► Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation |
| ► I ² C Digital Interface | ► Popular Two-Wire Serial Data Interface for Consumer Electronics |
| ► Lead Free Package Construction | ► RoHS Compliance |
| ► Wide Magnetic Field Range (+/- 8 Oe) | ► Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy |
| ► Software and Algorithm Support Available | ► Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available |
| ► Fast 160 Hz Maximum Output Rate | ► Enables Pedestrian Navigation and LBS Applications |

HMC5883L

SPECIFICATIONS (* Tested at 25°C except stated otherwise.)

Characteristics	Conditions*	Min	Typ	Max	Units
Power Supply					
Supply Voltage	VDD Referenced to AGND VDDIO Referenced to DGND	2.16 1.71	2.5 1.8	3.6 VDD+0.1	Volts Volts
Average Current Draw	Idle Mode Measurement Mode (7.5 Hz ODR; No measurement average, MA1:MA0 = 00) VDD = 2.5V, VDDIO = 1.8V (Dual Supply) VDD = VDDIO = 2.5V (Single Supply)	- -	2 100	- -	µA µA
Performance					
Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss
Digital Resolution	VDD=3.0V, GN=0 to 7, 1-LSb, 12-bit ADC	0.73		4.35	milli-gauss
Noise Floor (Field Resolution)	VDD=3.0V, GN=0, No measurement average, Standard Deviation 100 samples (See typical performance graphs below)		2		milli-gauss
Linearity	±2.0 gauss input range			0.1	±% FS
Hysteresis	±2.0 gauss input range		±25		ppm
Cross-Axis Sensitivity	Test Conditions: Cross field = 0.5 gauss, Happlied = ±3 gauss		±0.2%		%FS/gauss
Output Rate (ODR)	Continuous Measurement Mode Single Measurement Mode	0.75		75 160	Hz Hz
Measurement Period	From receiving command to data ready		6		ms
Turn-on Time	Ready for I ² C commands Analog Circuit Ready for Measurements		200 50		µs ms
Gain Tolerance	All gain/dynamic range settings		±5		%
I ² C Address	8-bit read address 8-bit write address		0x3D 0x3C		hex hex
I ² C Rate	Controlled by I ² C Master			400	kHz
I ² C Hysteresis	Hysteresis of Schmitt trigger inputs on SCL and SDA - Fall (VDDIO=1.8V) Rise (VDDIO=1.8V)		0.2*VDDIO 0.8*VDDIO		Volts Volts
Self Test	X & Y Axes Z Axis		±1.16 ±1.08		gauss
	X & Y & Z Axes (GN=5) Positive Bias X & Y & Z Axes (GN=5) Negative Bias	243 -575		575 -243	LSb
Sensitivity Tempco	T _A = -40 to 125°C, Uncompensated Output		-0.3		%/°C
General					
ESD Voltage	Human Body Model (all pins) Charged Device Model (all pins)			2000 750	Volts
Operating Temperature	Ambient	-30		85	°C
Storage Temperature	Ambient, unbiased	-40		125	°C

HMC5883L

Characteristics	Conditions*	Min	Typ	Max	Units
Reflow Classification	MSL 3, 260 °C Peak Temperature				
Package Size	Length and Width	2.85	3.00	3.15	mm
Package Height		0.8	0.9	1.0	mm
Package Weight			18		mg

Absolute Maximum Ratings (* Tested at 25°C except stated otherwise.)

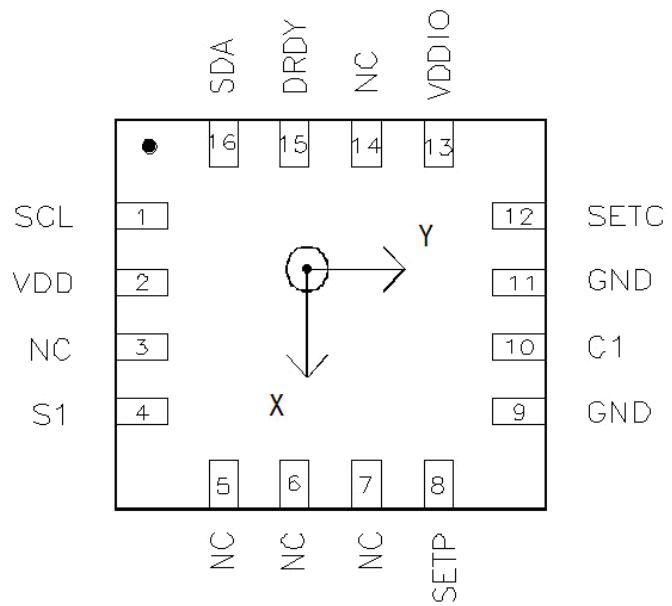
Characteristics	Min	Max	Units
Supply Voltage VDD	-0.3	4.8	Volts
Supply Voltage VDDIO	-0.3	4.8	Volts

PIN CONFIGURATIONS

Pin	Name	Description
1	SCL	Serial Clock – I ² C Master/Slave Clock
2	VDD	Power Supply (2.16V to 3.6V)
3	NC	Not to be Connected
4	S1	Tie to VDDIO
5	NC	Not to be Connected
6	NC	Not to be Connected
7	NC	Not to be Connected
8	SETP	Set/Reset Strap Positive – S/R Capacitor (C2) Connection
9	GND	Supply Ground
10	C1	Reservoir Capacitor (C1) Connection
11	GND	Supply Ground
12	SETC	S/R Capacitor (C2) Connection – Driver Side
13	VDDIO	IO Power Supply (1.71V to VDD)
14	NC	Not to be Connected
15	DRDY	Data Ready, Interrupt Pin. Internally pulled high. Optional connection. Low for 250 usec when data is placed in the data output registers.
16	SDA	Serial Data – I ² C Master/Slave Data

Table 1: Pin Configurations

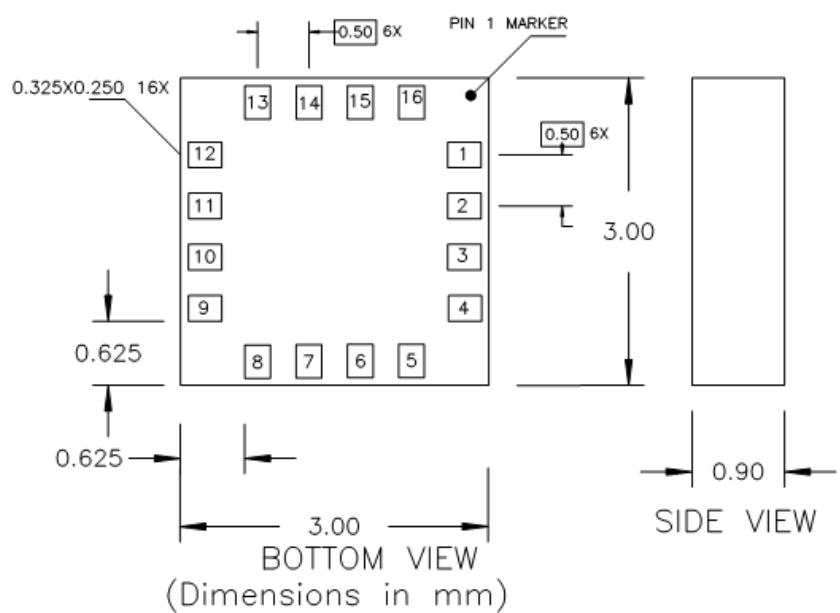
HMC5883L



Arrow indicates direction of magnetic field that generates a positive output reading in Normal Measurement configuration.

PACKAGE OUTLINES

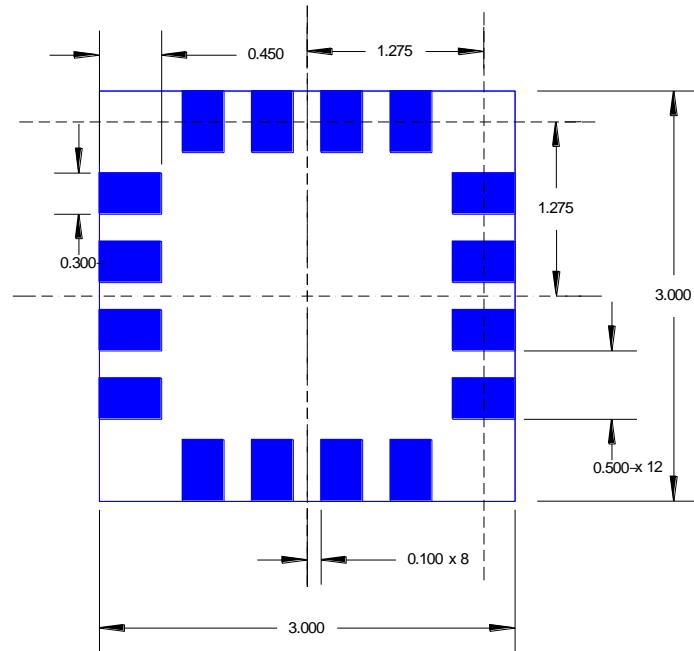
PACKAGE DRAWING HMC5883L (16-PIN LPCC, dimensions in millimeters)



MOUNTING CONSIDERATIONS

The following is the recommended printed circuit board (PCB) footprint for the HMC5883L.

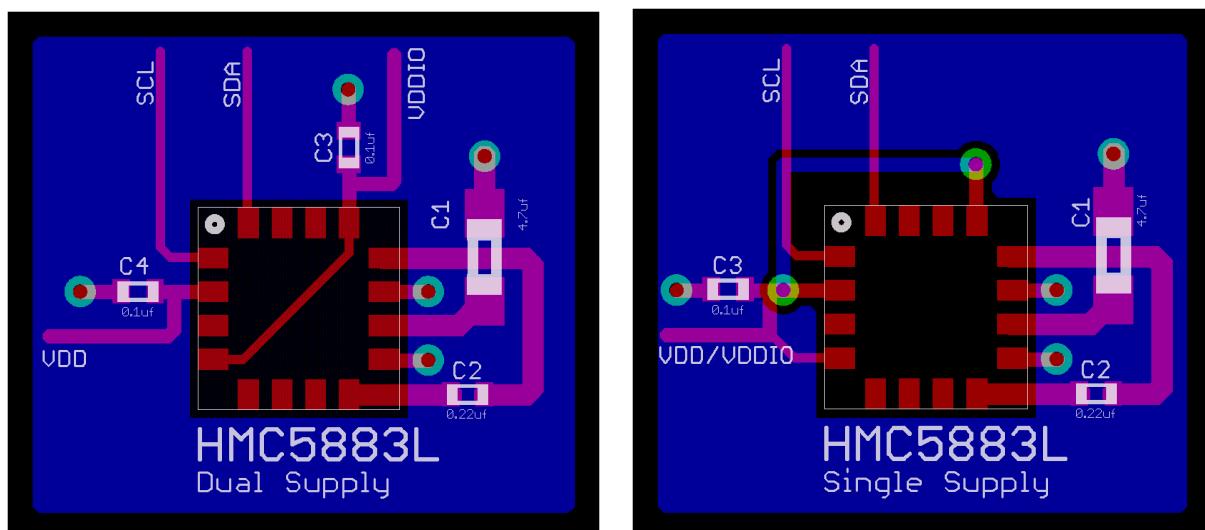
HMC5883L



HMC5883 Land Pad Pattern
(All dimensions are in mm)

LAYOUT CONSIDERATIONS

Besides keeping all components that may contain ferrous materials (nickel, etc.) away from the sensor on both sides of the PCB, it is also recommended that there is no conducting copper under/near the sensor in any of the PCB layers. See recommended layout below. Notice that the one trace under the sensor in the dual supply mode is not expected to carry active current since it is for pin 4 pull-up to VDDIO. Power and ground planes are removed under the sensor to minimize possible source of magnetic noise. For best results, use non-ferrous materials for all exposed copper coding.



HMC5883L

PCB Pad Definition and Traces

The HMC5883L is a fine pitch LCC package. Refer to previous figure for recommended PCB footprint for proper package centering. Size the traces between the HMC5883L and the external capacitors (C1 and C2) to handle the 1 ampere peak current pulses with low voltage drop on the traces.

Stencil Design and Solder Paste

A 4 mil stencil and 100% paste coverage is recommended for the electrical contact pads.

Reflow Assembly

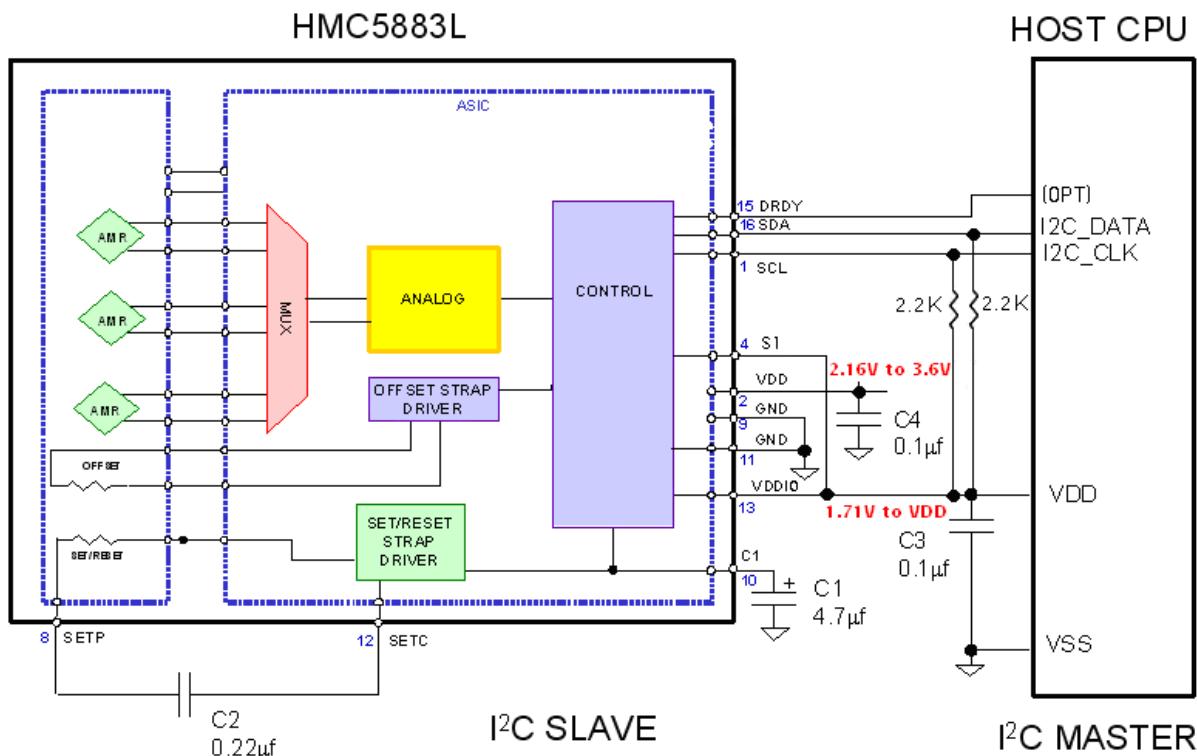
This device is classified as MSL 3 with 260°C peak reflow temperature. A baking process (125°C, 24 hrs) is required if device is not kept continuously in a dry (< 10% RH) environment before assembly. No special reflow profile is required for HMC5883L, which is compatible with lead eutectic and lead-free solder paste reflow profiles. Honeywell recommends adherence to solder paste manufacturer's guidelines. Hand soldering is not recommended. Built-in self test can be used to verify device functionalities after assembly.

External Capacitors

The two external capacitors should be ceramic type construction with low ESR characteristics. The exact ESR values are not critical but values less than 200 milli-ohms are recommended. Reservoir capacitor C1 is nominally 4.7 µF in capacitance, with the set/reset capacitor C2 nominally 0.22 µF in capacitance. Low ESR characteristics may not be in many small SMT ceramic capacitors (0402), so be prepared to up-size the capacitors to gain Low ESR characteristics.

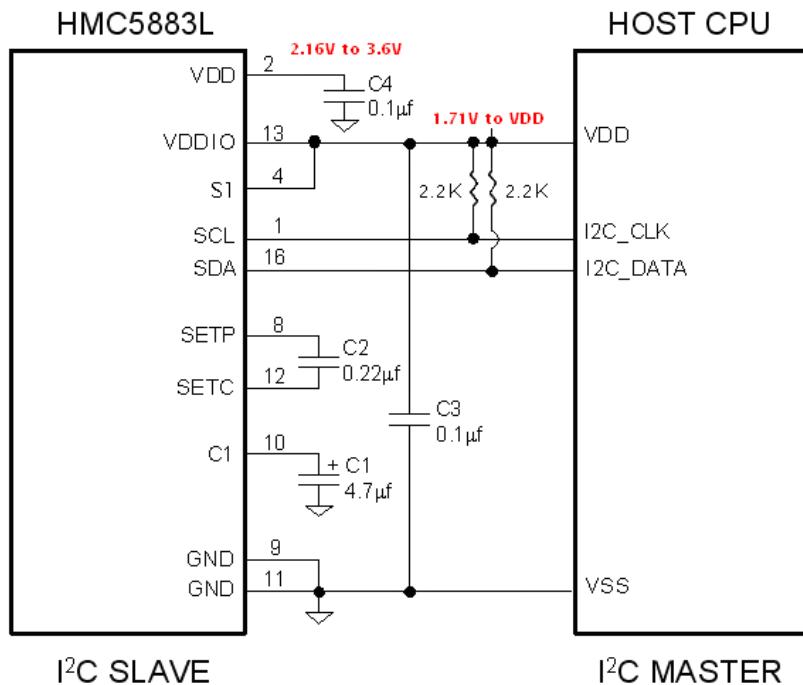
INTERNAL SCHEMATIC DIAGRAM

HMC5883L

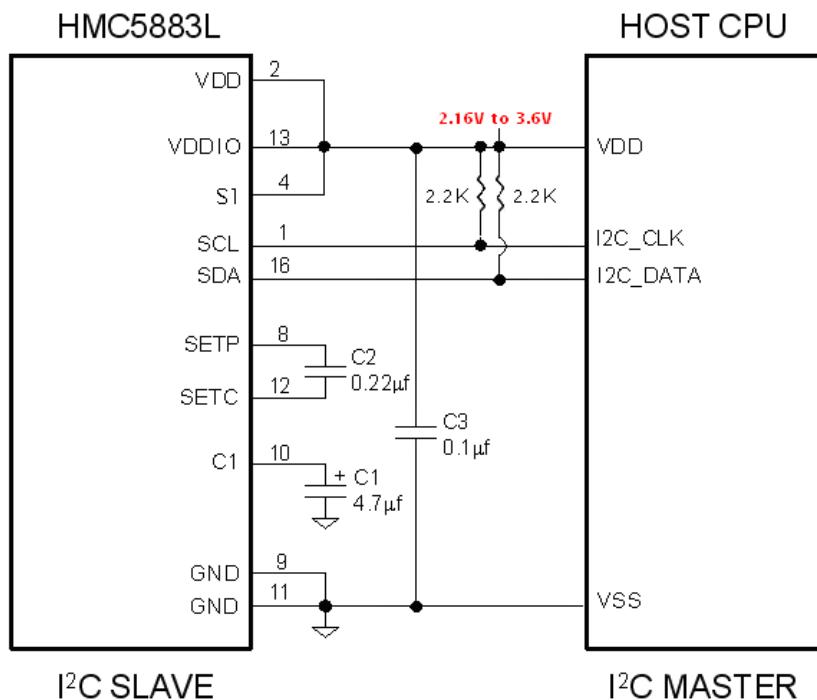


HMC5883L

DUAL SUPPLY REFERENCE DESIGN



SINGLE SUPPLY REFERENCE DESIGN

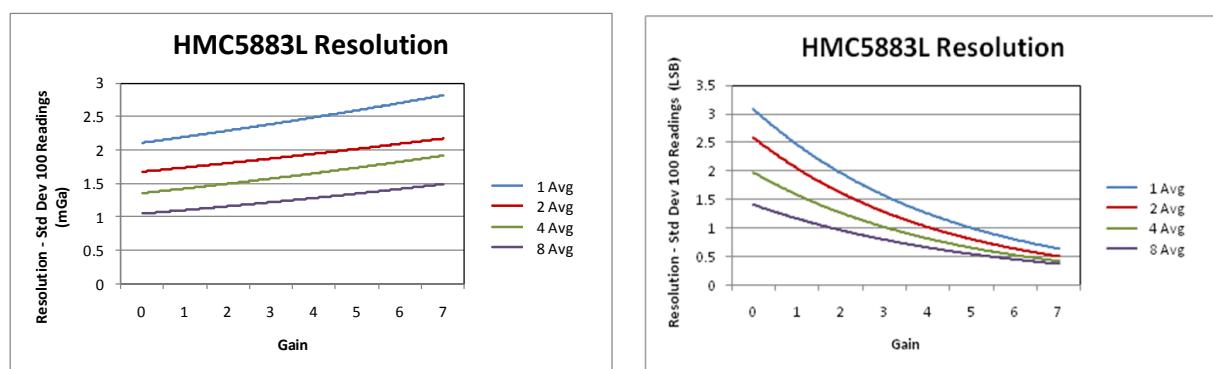


HMC5883L

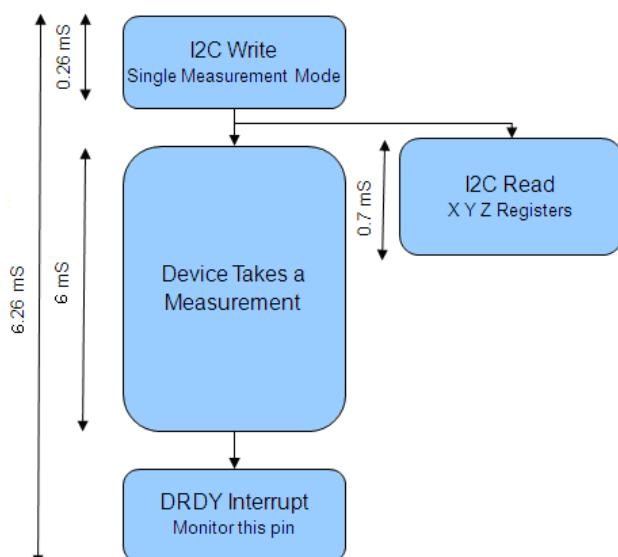
PERFORMANCE

The following graph(s) highlight HMC5883L's performance.

Typical Noise Floor (Field Resolution)



Typical Measurement Period in Single-Measurement Mode



* Monitoring of the DRDY Interrupt pin is only required if maximum output rate is desired.

HMC5883L

BASIC DEVICE OPERATION

Anisotropic Magneto-Resistive Sensors

The Honeywell HMC5883L magnetoresistive sensor circuit is a trio of sensors and application specific support circuits to measure magnetic fields. With power supply applied, the sensor converts any incident magnetic field in the sensitive axis directions to a differential voltage output. The magnetoresistive sensors are made of a nickel-iron (Permalloy) thin-film and patterned as a resistive strip element. In the presence of a magnetic field, a change in the bridge resistive elements causes a corresponding change in voltage across the bridge outputs.

These resistive elements are aligned together to have a common sensitive axis (indicated by arrows in the pinout diagram) that will provide positive voltage change with magnetic fields increasing in the sensitive direction. Because the output is only proportional to the magnetic field component along its axis, additional sensor bridges are placed at orthogonal directions to permit accurate measurement of magnetic field in any orientation.

Self Test

To check the HMC5883L for proper operation, a self test feature is incorporated in which the sensor is internally excited with a nominal magnetic field (in either positive or negative bias configuration). This field is then measured and reported. This function is enabled and the polarity is set by bits MS[n] in the configuration register A. An internal current source generates DC current (about 10 mA) from the VDD supply. This DC current is applied to the offset straps of the magnetoresistive sensor, which creates an artificial magnetic field bias on the sensor. The difference of this measurement and the measurement of the ambient field will be put in the data output register for each of the three axes. By using this built-in function, the manufacturer can quickly verify the sensor's full functionality after the assembly without additional test setup. The self test results can also be used to estimate/compensate the sensor's sensitivity drift due to temperature.

For each "self test measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (M1)
3. Sends the (~10 mA) offset current to generate the (~1.1 Gauss) offset field and takes another measurement (M2)
4. Puts the difference of the two measurements in sensor's data output register:

$$\text{Output} = [M2 - M1] \quad (\text{i.e. output} = \text{offset field only})$$

See SELF TEST OPERATION section later in this datasheet for additional details.

Power Management

This device has two different domains of power supply. The first one is VDD that is the power supply for internal operations and the second one is VDDIO that is dedicated to IO interface. It is possible to work with VDDIO equal to VDD; Single Supply mode, or with VDDIO lower than VDD allowing HMC5883L to be compatible with other devices on board.

I²C Interface

Control of this device is carried out via the I²C bus. This device will be connected to this bus as a slave device under the control of a master device, such as the processor.

This device is compliant with *I²C-Bus Specification*, document number: 9398 393 40011. As an I²C compatible device, this device has a 7-bit serial address and supports I²C protocols. This device supports standard and fast modes, 100kHz and 400kHz, respectively, but does not support the high speed mode (Hs). External pull-up resistors are required to support these standard and fast speed modes.

Activities required by the master (register read and write) have priority over internal activities, such as the measurement. The purpose of this priority is to not keep the master waiting and the I²C bus engaged for longer than necessary.

Internal Clock

The device has an internal clock for internal digital logic functions and timing management. This clock is not available to external usage.

HMC5883L

H-Bridge for Set/Reset Strap Drive

The ASIC contains large switching FETs capable of delivering a large but brief pulse to the Set/Reset strap of the sensor. This strap is largely a resistive load. There is no need for an external Set/Reset circuit. The controlling of the Set/Reset function is done automatically by the ASIC for each measurement. One half of the difference from the measurements taken after a set pulse and after a reset pulse will be put in the data output register for each of the three axes. By doing so, the sensor's internal offset and its temperature dependence is removed/cancelled for all measurements. The set/reset pulses also effectively remove the past magnetic history (magnetism) in the sensor, if any.

For each "measurement", the ASIC:

1. Sends a "Set" pulse
2. Takes one measurement (M_{set})
3. Sends a "Reset" pulse
4. Takes another measurement (M_{reset})
5. Puts the following result in sensor's data output register:

$$\text{Output} = [M_{set} - M_{reset}] / 2$$

Charge Current Limit

The current that reservoir capacitor (C_1) can draw when charging is limited for both single supply and dual supply configurations. This prevents drawing down the supply voltage (V_{DD}).

MODES OF OPERATION

This device has several operating modes whose primary purpose is power management and is controlled by the Mode Register. This section describes these modes.

Continuous-Measurement Mode

During continuous-measurement mode, the device continuously makes measurements, at user selectable rate, and places measured data in data output registers. Data can be re-read from the data output registers if necessary; however, if the master does not ensure that the data register is accessed before the completion of the next measurement, the data output registers are updated with the new measurement. To conserve current between measurements, the device is placed in a state similar to idle mode, but the Mode Register is not changed to Idle Mode. That is, $MD[n]$ bits are unchanged. Settings in the Configuration Register A affect the data output rate (bits $DO[n]$), the measurement configuration (bits $MS[n]$), when in continuous-measurement mode. All registers maintain values while in continuous-measurement mode. The I²C bus is enabled for use by other devices on the network in while continuous-measurement mode.

Single-Measurement Mode

This is the default power-up mode. During single-measurement mode, the device makes a single measurement and places the measured data in data output registers. After the measurement is complete and output data registers are updated, the device is placed in idle mode, and the Mode Register is changed to idle mode by setting $MD[n]$ bits. Settings in the configuration register affect the measurement configuration (bits $MS[n]$) when in single-measurement mode. All registers maintain values while in single-measurement mode. The I²C bus is enabled for use by other devices on the network while in single-measurement mode.

Idle Mode

During this mode the device is accessible through the I²C bus, but major sources of power consumption are disabled, such as, but not limited to, the ADC, the amplifier, and the sensor bias current. All registers maintain values while in idle mode. The I²C bus is enabled for use by other devices on the network while in idle mode.

HMC5883L

REGISTERS

This device is controlled and configured via a number of on-chip registers, which are described in this section. In the following descriptions, *set* implies a logic 1, and *reset* or *clear* implies a logic 0, unless stated otherwise.

Register List

The table below lists the registers and their access. All address locations are 8 bits.

Address Location	Name	Access
00	Configuration Register A	Read/Write
01	Configuration Register B	Read/Write
02	Mode Register	Read/Write
03	Data Output X MSB Register	Read
04	Data Output X LSB Register	Read
05	Data Output Z MSB Register	Read
06	Data Output Z LSB Register	Read
07	Data Output Y MSB Register	Read
08	Data Output Y LSB Register	Read
09	Status Register	Read
10	Identification Register A	Read
11	Identification Register B	Read
12	Identification Register C	Read

Table2: Register List

Register Access

This section describes the process of reading from and writing to this device. The devices uses an address pointer to indicate which register location is to be read from or written to. These pointer locations are sent from the master to this slave device and succeed the 7-bit address (0x1E) plus 1 bit read/write identifier, i.e. 0x3D for read and 0x3C for write.

To minimize the communication between the master and this device, the address pointer updated automatically without master intervention. The register pointer will be incremented by 1 automatically after the current register has been read successfully.

The address pointer value itself cannot be read via the I²C bus.

Any attempt to read an invalid address location returns 0's, and any write to an invalid address location or an undefined bit within a valid address location is ignored by this device.

To move the address pointer to a random register location, first issue a "write" to that register location with no data byte following the command. For example, to move the address pointer to register 10, send 0x3C 0x0A.

HMC5883L

Configuration Register A

The configuration register is used to configure the device for setting the data output rate and measurement configuration. CRA0 through CRA7 indicate bit locations, with CRA denoting the bits that are in the configuration register. CRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. CRA default is 0x10.

CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Table 3: Configuration Register A

Location	Name	Description
CRA7	CRA7	Bit CRA7 is reserved for future function. Set to 0 when configuring CRA.
CRA6 to CRA5	MA1 to MA0	Select number of samples averaged (1 to 8) per measurement output. 00 = 1(Default); 01 = 2; 10 = 4; 11 = 8
CRA4 to CRA2	DO2 to DO0	Data Output Rate Bits. These bits set the rate at which data is written to all three data output registers.
CRA1 to CRA0	MS1 to MS0	Measurement Configuration Bits. These bits define the measurement flow of the device, specifically whether or not to incorporate an applied bias into the measurement.

Table 4: Configuration Register A Bit Designations

The Table below shows all selectable output rates in continuous measurement mode. All three channels shall be measured within a given output rate. Other output rates with maximum rate of 160 Hz can be achieved by monitoring DRDY interrupt pin in single measurement mode.

DO2	DO1	DO0	Typical Data Output Rate (Hz)
0	0	0	0.75
0	0	1	1.5
0	1	0	3
0	1	1	7.5
1	0	0	15 (Default)
1	0	1	30
1	1	0	75
1	1	1	Reserved

Table 5: Data Output Rates

MS1	MS0	Measurement Mode
0	0	Normal measurement configuration (Default). In normal measurement configuration the device follows normal measurement flow. The positive and negative pins of the resistive load are left floating and high impedance.
0	1	Positive bias configuration for X, Y, and Z axes. In this configuration, a positive current is forced across the resistive load for all three axes.
1	0	Negative bias configuration for X, Y and Z axes. In this configuration, a negative current is forced across the resistive load for all three axes..
1	1	This configuration is reserved.

Table 6: Measurement Modes

HMC5883L

Configuration Register B

The configuration register B for setting the device gain. CRB0 through CRB7 indicate bit locations, with *CRB* denoting the bits that are in the configuration register. CRB7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. CRB default is 0x20.

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

Table 7: Configuration B Register

Location	Name	Description
CRB7 to CRB5	GN2 to GN0	Gain Configuration Bits. These bits configure the gain for the device. The gain configuration is common for all channels.
CRB4 to CRB0	0	These bits must be cleared for correct operation.

Table 8: Configuration Register B Bit Designations

The table below shows nominal gain settings. Use the “Gain” column to convert counts to Gauss. The “Digital Resolution” column is the theoretical value in term of milli-Gauss per count (LSb) which is the inverse of the values in the “Gain” column. The effective resolution of the usable signal also depends on the noise floor of the system, i.e.

Effective Resolution = Max (Digital Resolution, Noise Floor)

Choose a lower gain value (higher GN#) when total field strength causes overflow in one of the data output registers (saturation). Note that the very first measurement after a gain change maintains the same gain as the previous setting. **The new gain setting is effective from the second measurement and on.**

GN2	GN1	GN0	Recommended Sensor Field Range	Gain (LSb/Gauss)	Digital Resolution (mG/LSb)	Output Range
0	0	0	± 0.88 Ga	1370	0.73	0xF800–0x07FF (-2048–2047)
0	0	1	± 1.3 Ga	1090 (default)	0.92	0xF800–0x07FF (-2048–2047)
0	1	0	± 1.9 Ga	820	1.22	0xF800–0x07FF (-2048–2047)
0	1	1	± 2.5 Ga	660	1.52	0xF800–0x07FF (-2048–2047)
1	0	0	± 4.0 Ga	440	2.27	0xF800–0x07FF (-2048–2047)
1	0	1	± 4.7 Ga	390	2.56	0xF800–0x07FF (-2048–2047)
1	1	0	± 5.6 Ga	330	3.03	0xF800–0x07FF (-2048–2047)
1	1	1	± 8.1 Ga	230	4.35	0xF800–0x07FF (-2048–2047)

Table 9: Gain Settings

HMC5883L

Mode Register

The mode register is an 8-bit register from which data can be read or to which data can be written. This register is used to select the operating mode of the device. MR0 through MR7 indicate bit locations, with *MR* denoting the bits that are in the mode register. MR7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit. Mode register default is 0x01.

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
HS(0)	(0)	(0)	(0)	(0)	(0)	MD1 (0)	MD0 (1)

Table 10: Mode Register

Location	Name	Description
MR7 to MR2	HS	Set this pin to enable High Speed I ₂ C, 3400kHz.
MR1 to MR0	MD1 to MD0	Mode Select Bits. These bits select the operation mode of this device.

Table 11: Mode Register Bit Designations

MD1	MD0	Operating Mode
0	0	Continuous-Measurement Mode. In continuous-measurement mode, the device continuously performs measurements and places the result in the data register. RDY goes high when new data is placed in all three registers. After a power-on or a write to the mode or configuration register, the first measurement set is available from all three data output registers after a period of $2/f_{DO}$ and subsequent measurements are available at a frequency of f_{DO} , where f_{DO} is the frequency of data output.
0	1	Single-Measurement Mode (Default). When single-measurement mode is selected, device performs a single measurement, sets RDY high and returned to idle mode. Mode register returns to idle mode bit values. The measurement remains in the data output register and RDY remains high until the data output register is read or another measurement is performed.
1	0	Idle Mode. Device is placed in idle mode.
1	1	Idle Mode. Device is placed in idle mode.

Table 12: Operating Modes

HMC5883L

Data Output X Registers A and B

The data output X registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel X. Data output X register A contains the MSB from the measurement result, and data output X register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DXRA0 through DXRA7 and DXRB0 through DXRB7 indicate bit locations, with *DXRA* and *DXRB* denoting the bits that are in the data output X registers. DXRA7 and DXRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DXRA7	DXRA6	DXRA5	DXRA4	DXRA3	DXRA2	DXRA1	DXRA0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DYRB7	DYRB6	DYRB5	DYRB4	DYRB3	DYRB2	DYRB1	DYRB0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 13: Data Output X Registers A and B

Data Output Y Registers A and B

The data output Y registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Y. Data output Y register A contains the MSB from the measurement result, and data output Y register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DYRA0 through DYRA7 and DYRB0 through DYRB7 indicate bit locations, with *DYRA* and *DYRB* denoting the bits that are in the data output Y registers. DYRA7 and DYRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

DYRA7	DYRA6	DYRA5	DYRA4	DYRA3	DYRA2	DYRA1	DYRA0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DYRB7	DYRB6	DYRB5	DYRB4	DYRB3	DYRB2	DYRB1	DYRB0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 14: Data Output Y Registers A and B

Data Output Z Registers A and B

The data output Z registers are two 8-bit registers, data output register A and data output register B. These registers store the measurement result from channel Z. Data output Z register A contains the MSB from the measurement result, and data output Z register B contains the LSB from the measurement result. The value stored in these two registers is a 16-bit value in 2's complement form, whose range is 0xF800 to 0x07FF. DZRA0 through DZRA7 and DZRB0 through DZRB7 indicate bit locations, with *DZRA* and *DZRB* denoting the bits that are in the data output Z registers. DZRA7 and DZRB7 denote the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

In the event the ADC reading overflows or underflows for the given channel, or if there is a math overflow during the bias measurement, this data register will contain the value -4096. This register value will clear when after the next valid measurement is made.

HMC5883L

DZRA7	DZRA6	DZRA5	DZRA4	DZRA3	DZRA2	DZRA1	DZRA0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
DZRB7	DZRB6	DZRB5	DZRB4	DZRB3	DZRB2	DZRB1	DZRB0
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Table 15: Data Output Z Registers A and B

Data Output Register Operation

When one or more of the output registers are read, new data cannot be placed in any of the output data registers until all six data output registers are read. This requirement also impacts DRDY and RDY, which cannot be cleared until new data is placed in all the output registers.

Status Register

The status register is an 8-bit read-only register. This register is used to indicate device status. SR0 through SR7 indicate bit locations, with SR denoting the bits that are in the status register. SR7 denotes the first bit of the data stream.

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
(0)	(0)	(0)	(0)	(0)	(0)	LOCK (0)	RDY(0)

Table 16: Status Register

Location	Name	Description
SR7 to SR2	0	These bits are reserved.
SR1	LOCK	Data output register lock. This bit is set when: 1. some but not all of the six data output registers have been read, 2. Mode register has been read. When this bit is set, the six data output registers are locked and any new data will not be placed in these registers until one of these conditions are met: 1. all six bytes have been read, 2. the mode register is changed, 3. the measurement configuration (CRA) is changed, 4. power is reset.
SR0	RDY	Ready Bit. Set when data is written to all six data registers. Cleared when device initiates a write to the data output registers and after one or more of the data output registers are written to. When RDY bit is clear it shall remain cleared for a 250 µs. DRDY pin can be used as an alternative to the status register for monitoring the device for measurement data.

Table 17: Status Register Bit Designations

HMC5883L

Identification Register A

The identification register A is used to identify the device. IRA0 through IRA7 indicate bit locations, with */RA* denoting the bits that are in the identification register A. IRA7 denotes the first bit of the data stream. The number in parenthesis indicates the default value of that bit.

The identification value for this device is stored in this register. This is a read-only register.
Register values. ASCII value *H*

IRA7	IRA6	IRA5	IRA4	IRA3	IRA2	IRA1	IRA0
0	1	0	0	1	0	0	0

Table 18: Identification Register A Default Values

Identification Register B

The identification register B is used to identify the device. IRB0 through IRB7 indicate bit locations, with */RB* denoting the bits that are in the identification register A. IRB7 denotes the first bit of the data stream.

Register values. ASCII value *4*

IRB7	IRB6	IRB5	IRB4	IRB3	IRB2	IRB1	IRB0
0	0	1	1	0	1	0	0

Table 19: Identification Register B Default Values

Identification Register C

The identification register C is used to identify the device. IRC0 through IRC7 indicate bit locations, with */RC* denoting the bits that are in the identification register A. IRC7 denotes the first bit of the data stream.

Register values. ASCII value *3*

IRC7	IRC6	IRC5	IRC4	IRC3	IRC2	IRC1	IRC0
0	0	1	1	0	0	1	1

Table 20: Identification Register C Default Values

I²C COMMUNICATION PROTOCOL

The HMC5883L communicates via a two-wire I²C bus system as a slave device. The HMC5883L uses a simple protocol with the interface protocol defined by the I²C bus specification, and by this document. The data rate is at the standard-mode 100kbps or 400kbps rates as defined in the I²C Bus Specifications. The bus bit format is an 8-bit Data/Address send and a 1-bit acknowledge bit. The format of the data bytes (payload) shall be case sensitive ASCII characters or binary data to the HMC5883L slave, and binary data returned. Negative binary values will be in two's complement form. The default (factory) HMC5883L 8-bit slave address is 0x3C for write operations, or 0x3D for read operations.

The HMC5883L Serial Clock (SCL) and Serial Data (SDA) lines require resistive pull-ups (*R_p*) between the master device (usually a host microprocessor) and the HMC5883L. Pull-up resistance values of about 2.2K to 10K ohms are recommended with a nominal VDDIO voltage. Other resistor values may be used as defined in the I²C Bus Specifications that can be tied to VDDIO.

The SCL and SDA lines in this bus specification may be connected to multiple devices. The bus can be a single master to multiple slaves, or it can be a multiple master configuration. All data transfers are initiated by the master device, which is responsible for generating the clock signal, and the data transfers are 8 bit long. All devices are addressed by I²C's unique 7-bit address. After each 8-bit transfer, the master device generates a 9th clock pulse, and releases the SDA line. The receiving device (addressed slave) will pull the SDA line low to acknowledge (ACK) the successful transfer or leave the SDA high to negative acknowledge (NACK).

HMC5883L

Per the I²C spec, all transitions in the SDA line must occur when SCL is low. This requirement leads to two unique conditions on the bus associated with the SDA transitions when SCL is high. Master device pulling the SDA line low while the SCL line is high indicates the Start (S) condition, and the Stop (P) condition is when the SDA line is pulled high while the SCL line is high. The I²C protocol also allows for the Restart condition in which the master device issues a second start condition without issuing a stop.

All bus transactions begin with the master device issuing the start sequence followed by the slave address byte. The address byte contains the slave address; the upper 7 bits (bits7-1), and the Least Significant bit (LSb). The LSb of the address byte designates if the operation is a read (LSb=1) or a write (LSb=0). At the 9th clock pulse, the receiving slave device will issue the ACK (or NACK). Following these bus events, the master will send data bytes for a write operation, or the slave will clock out data with a read operation. All bus transactions are terminated with the master issuing a stop sequence.

I²C bus control can be implemented with either hardware logic or in software. Typical hardware designs will release the SDA and SCL lines as appropriate to allow the slave device to manipulate these lines. In a software implementation, care must be taken to perform these tasks in code.

OPERATIONAL EXAMPLES

The HMC5883L has a fairly quick stabilization time from no voltage to stable and ready for data retrieval. The nominal 56 milli-seconds with the factory default single measurement mode means that the six bytes of magnetic data registers (DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB) are filled with a valid first measurement.

To change the measurement mode to continuous measurement mode, after the power-up time send the three bytes:

0x3C 0x02 0x00

This writes the 00 into the second register or mode register to switch from single to continuous measurement mode setting. With the data rate at the factory default of 15Hz updates, a 67 milli-second typical delay should be allowed by the I²C master before querying the HMC5883L data registers for new measurements. To clock out the new data, send:

0x3D, and clock out DXRA, DXRB, DZRA, DZRB, DYRA, and DYRB located in registers 3 through 8. The HMC5883L will automatically re-point back to register 3 for the next 0x3D query. All six data registers must be read properly before new data can be placed in any of these data registers.

Below is an example of a (power-on) initialization process for “continuous-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default, normal measurement)
 2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
 3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
 4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
 5. Loop
 - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
 - Convert three 16-bit 2's compliment hex values to decimal values and assign to X, Z, Y, respectively.
 - Send **0x3C 0x03** (point to first data register 03)
 - Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin
- End_loop

Below is an example of a (power-on) initialization process for “single-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default or any other rate, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. For each measurement query:
 - Write Mode (02) – send **0x3C 0x02 0x01** (Single-measurement mode)
 - Wait 6 ms or monitor status register or DRDY hardware interrupt pin
 - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
 - Convert three 16-bit 2's compliment hex values to decimal values and assign to X, Z, Y, respectively.

HMC5883L

SELF TEST OPERATION

To check the HMC5883L for proper operation, a self test feature is incorporated in which the sensor offset straps are excited to create a nominal field strength (bias field) to be measured. To implement self test, the least significant bits (MS1 and MS0) of configuration register A are changed from 00 to 01 (positive bias) or 10 (negative bias).

Then, by placing the mode register into single or continuous-measurement mode, two data acquisition cycles will be made on each magnetic vector. The first acquisition will be a set pulse followed shortly by measurement data of the external field. The second acquisition will have the offset strap excited (about 10 mA) in the positive bias mode for X, Y, and Z axes to create about a 1.1 gauss self test field plus the external field. The first acquisition values will be subtracted from the second acquisition, and the net measurement will be placed into the data output registers.

Since self test adds ~1.1 Gauss additional field to the existing field strength, using a reduced gain setting prevents sensor from being saturated and data registers overflowed. For example, if the configuration register B is set to 0xA0 (Gain=5), values around +452 LSb (1.16 Ga * 390 LSb/Ga) will be placed in the X and Y data output registers and around +421 (1.08 Ga * 390 LSb/Ga) will be placed in Z data output register. To leave the self test mode, change MS1 and MS0 bit of the configuration register A back to 00 (Normal Measurement Mode). Acceptable limits of the self test values depend on the gain setting. Limits for Gain=5 is provided in the specification table.

Below is an example of a “positive self test” process using continuous-measurement mode:

1. Write CRA (00) – send **0x3C 0x00 0x71** (8-average, 15 Hz default, positive self test measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5)
3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
 - Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
 - Convert three 16-bit 2's compliment hex values to decimal values and assign to X, Z, Y, respectively.
 - Send **0x3C 0x03** (point to first data register 03)
 - Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin
6. Check limits –
 - If all 3 axes (X, Y, and Z) are within reasonable limits (243 to 575 for Gain=5, adjust these limits basing on the gain setting used. See an example below.) Then
 - All 3 axes pass positive self test
 - Write CRA (00) – send **0x3C 0x00 0x70** (Exit self test mode and this procedure)
 - Else
 - If Gain<7
 - Write CRB (01) – send **0x3C 0x01 0x_0** (Increase gain setting and retry, skip the next data set)
 - Else
 - At least one axis did not pass positive self test
 - Write CRA (00) – send **0x3C 0x00 0x70** (Exit self test mode and this procedure)
- End If

Below is an example of how to adjust the “positive self” test limits basing on the gain setting:

1. If Gain = 6, self test limits are:
Low Limit = $243 * 330/390 = 206$
High Limit = $575 * 330/390 = 487$
2. If Gain = 7, self test limits are:
Low Limit = $243 * 230/390 = 143$
High Limit = $575 * 230/390 = 339$

HMC5883L

SCALE FACTOR TEMPERATURE COMPENSATION

The built-in self test can also be used to periodically compensate the scaling errors due to temperature variations. A compensation factor can be found by comparing the self test outputs with the ones obtained at a known temperature. For example, if the self test output is 400 at room temperature and 300 at the current temperature then a compensation factor of (400/300) should be applied to all current magnetic readings. A temperature sensor is not required using this method.

Below is an example of a temperature compensation process using positive self test method:

1. If self test measurement at a temperature "when the last magnetic calibration was done":

$$X_{STP} = 400$$

$$Y_{STP} = 410$$

$$Z_{STP} = 420$$

2. If self test measurement at a different temperature:

$$X_{STP} = 300 \text{ (Lower than before)}$$

$$Y_{STP} = 310 \text{ (Lower than before)}$$

$$Z_{STP} = 320 \text{ (Lower than before)}$$

Then

$$X_{TempComp} = 400/300$$

$$Y_{TempComp} = 410/310$$

$$Z_{TempComp} = 420/320$$

3. Applying to all new measurements:

$$X = X * X_{TempComp}$$

$$Y = Y * Y_{TempComp}$$

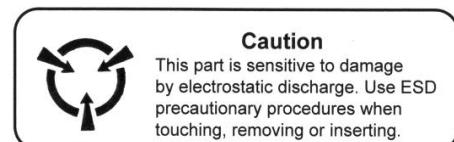
$$Z = Z * Z_{TempComp}$$

Now all 3 axes are temperature compensated, i.e. sensitivity is same as "when the last magnetic calibration was done"; therefore, the calibration coefficients can be applied without modification.

4. Repeat this process periodically or, for every Δt degrees of temperature change measured, if available.

ORDERING INFORMATION

Ordering Number	Product
HMC5883L-T	Cut Tape
HMC5883L-TR	Tape and Reel 4k pieces/reel



Caution

This part is sensitive to damage by electrostatic discharge. Use ESD precautionary procedures when touching, removing or inserting.

CAUTION: ESDS CAT. 1B

FIND OUT MORE

For more information on Honeywell's Magnetic Sensors visit us online at www.magneticsensors.com or contact us at 1-800-323-8295 (763-954-2474 internationally).

The application circuits herein constitute typical usage and interface of Honeywell product. Honeywell does not warranty or assume liability of customer-designed circuits derived from this description or depiction.

Honeywell reserves the right to make changes to improve reliability, function or design. Honeywell does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

U.S. Patents 4,441,072, 4,533,872, 4,569,742, 4,681,812, 4,847,584 and 6,529,114 apply to the technology described

Honeywell
12001 Highway 55
Plymouth, MN 55441
Tel: 800-323-8295
www.magneticsensors.com

Form # 900405 Rev E
February 2013
©2010 Honeywell International Inc.

Honeywell

Appendix L - Gyroscope Data Sheet

The data sheet for the gyroscope appears on the next page.

FEATURES

- Complete rate gyroscope on a single chip
- $\pm 300^\circ/\text{sec}$ angular rate sensing
- Ultrahigh vibration rejection: $0.01^\circ/\text{sec}/\text{g}$
- Excellent $16^\circ/\text{hour}$ null bias stability
- Internal temperature compensation
- 2000 g powered shock survivability
- SPI digital output with 16-bit data-word
- Low noise and low power
- 3.3 V to 5 V operation
- 40°C to +105°C operation
- Ultrasmall, light, and RoHS compliant
- Two package options
 - Low cost SOIC_CAV package for yaw rate (z-axis) response
 - Innovative ceramic vertical mount package (LCC_V) for pitch and roll response

APPLICATIONS

- Rotation sensing in high vibration environments
- Rotation sensing for industrial and instrumentation applications
- High performance platform stabilization

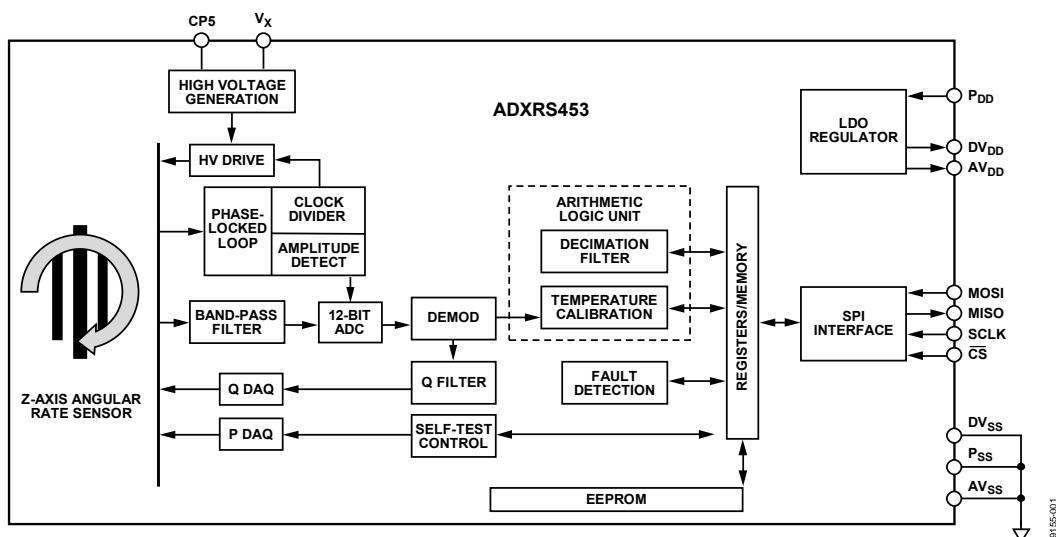
GENERAL DESCRIPTION

The ADXRS453 is an angular rate sensor (gyroscope) intended for industrial, instrumentation, and stabilization applications in high vibration environments. An advanced, differential, quad sensor design rejects the influence of linear acceleration, enabling the ADXRS453 to offer high accuracy rate sensing in harsh environments where shock and vibration are present.

The ADXRS453 uses an internal, continuous self-test architecture. The integrity of the electromechanical system is checked by applying a high frequency electrostatic force to the sense structure to generate a rate signal that can be differentiated from the baseband rate data and internally analyzed.

The ADXRS453 is capable of sensing an angular rate of up to $\pm 300^\circ/\text{sec}$. Angular rate data is presented as a 16-bit word that is part of a 32-bit SPI message.

The ADXRS453 is available in a 16-lead plastic cavity SOIC (SOIC_CAV) and an SMT-compatible vertical mount package (LCC_V), and is capable of operating across a wide voltage range (3.3 V to 5 V).

FUNCTIONAL BLOCK DIAGRAM

Figure 1.
Rev. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2011 Analog Devices, Inc. All rights reserved.

ADXRS453* Product Page Quick Links

Last Content Update: 08/30/2016

[Comparable Parts](#)

View a parametric search of comparable parts

[Evaluation Kits](#)

- ADXRS453Z Evaluation Board
- Real Time Eval System for Digital Output Sensor

[Documentation](#)

Application Notes

- AN-1073: Soldering Recommendations for the Ceramic Vertical Mount Package

Data Sheet

- ADXRS453: High Performance, Digital Output Gyroscope Data Sheet

User Guides

- UG-246: Evaluation Board for the $\pm 300^\circ/\text{sec}$, Single-Axis Digital Output Rate Gyroscope
- UG-260: ADXRS450/ADXRS453 Sensor Evaluation System

[Software and Systems Requirements](#)

- ADXRS453 - No-OS Driver for Microchip Microcontroller Platforms
 - ADXRS453 - No-OS Driver for Renesas Microcontroller Platforms
 - ADXRS453 Pmod Xilinx FPGA Reference Design
-

[Reference Materials](#)

Technical Articles

- MS-2158: Gyro Mechanical Performance: The Most Important Parameter

[Design Resources](#)

- ADXRS453 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

[Discussions](#)

View all ADXRS453 EngineerZone Discussions

[Sample and Buy](#)

Visit the product page to see pricing options

[Technical Support](#)

Submit a technical question or find your regional support number

* This page was dynamically generated by Analog Devices, Inc. and inserted into this data sheet. Note: Dynamic changes to the content on this page does not constitute a change to the revision number of the product data sheet. This content may be frequently modified.

TABLE OF CONTENTS

Features	1	ADXRS453 Signal Chain Timing.....	13
Applications.....	1	SPI Communication Protocol.....	14
General Description.....	1	Command/Response	14
Functional Block Diagram	1	Device Data Latching.....	15
Revision History	2	SPI Timing Characteristics	16
Specifications.....	3	Command/Response Bit Definitions.....	17
Absolute Maximum Ratings.....	4	Fault Register Bit Definitions	18
Thermal Resistance	4	Recommended Start-Up Sequence with CHK Bit Assertion.....	20
Rate Sensitive Axis	4	Rate Data Format.....	21
ESD Caution.....	4	Memory Map and Registers.....	22
Pin Configurations and Function Descriptions	5	Memory Map	22
Typical Performance Characteristics	7	Memory Register Definitions	23
Theory of Operation	9	Package Orientation and Layout Information.....	25
Continuous Self-Test.....	9	Solder Profile.....	26
Mechanical Performance	10	Package Marking Codes	27
Noise Performance	11	Outline Dimensions.....	28
Applications Information	12	Ordering Guide	29
Calibrated Performance.....	12		
Mechanical Considerations for Mounting	12		
Application Circuits	12		

REVISION HISTORY

12/11—Rev. A to Rev. B

Changes to Features Section.....	1
Changes to Rate Sensitive Axis Section	4
Deleted Endnote 1, Table 3.....	4
Deleted Figure 5; Renumbered Sequentially.....	6
Changes to Figure 4.....	6
Changes to Figure 32.....	25
Deleted Figure 36.....	26

6/11—Rev. 0 to Rev. A

Changes to Bit 30 and Bit 31 in Table 9	14
Updated Outline Dimensions	29
Changes to Ordering Guide	30

1/11—Revision 0: Initial Version

SPECIFICATIONS

$T_A = T_{MIN}$ to T_{MAX} , $P_{DD} = 5$ V, angular rate = $0^\circ/\text{sec}$, bandwidth = $f_0/200$ (~ 77.5 Hz), ± 1 g, continuous self-test on.

Table 1.

Parameter	Test Conditions/Comments	Symbol	Min	Typ	Max	Unit
MEASUREMENT RANGE	Full-scale range	FSR	± 300		± 400	$^\circ/\text{sec}$
SENSITIVITY	See Figure 2			80		LSB/ $^\circ/\text{sec}$
Nominal Sensitivity			-3		+3	%
Sensitivity Tolerance	$T_A = -40^\circ\text{C}$ to $+105^\circ\text{C}$			0.05		% FSR rms
Nonlinearity ¹	Best fit straight line			-3	+3	%
Cross-Axis Sensitivity ²						
NULL ACCURACY	$T_A = 25^\circ\text{C}$ $T_A = -40^\circ\text{C}$ to $+105^\circ\text{C}$			± 0.4		$^\circ/\text{sec}$
				± 0.5		$^\circ/\text{sec}$
NOISE PERFORMANCE						
Rate Noise Density	$T_A = 25^\circ\text{C}$ $T_A = 105^\circ\text{C}$			0.015		$^\circ/\text{sec}/\sqrt{\text{Hz}}$
				0.023		$^\circ/\text{sec}/\sqrt{\text{Hz}}$
LOW-PASS FILTER						
Cutoff (-3 dB) Frequency	$f_0/200$	f_{LP}		77.5		Hz
Group Delay ³	$f = 0$ Hz	t_{LP}	3.25	4	4.75	ms
SENSOR RESONANT FREQUENCY		f_0	13	15.5	19	kHz
SHOCK AND VIBRATION IMMUNITY						
Sensitivity to Linear Acceleration	DC to 5 kHz			0.01		$^\circ/\text{sec}/g$
Vibration Rectification				0.0002		$^\circ/\text{sec}/g^2$
SELF-TEST	See the Continuous Self-Test section					
Magnitude				2559		LSB
Fault Register Threshold	Compared to LOCSTx register data		2239		2879	LSB
Sensor Data Status Threshold	Compared to LOCSTx register data		1279		3839	LSB
Frequency	$f_0/32$	f_{ST}		485		Hz
ST Low-Pass Filter						
Cutoff (-3 dB) Frequency	$f_0/8000$			1.95		Hz
Group Delay ³			52	64	76	ms
SPI COMMUNICATIONS						
Clock Frequency				8.08		MHz
Voltage Input High	MOSI, \overline{CS} , SCLK			$0.85 \times P_{DD}$		V
Voltage Input Low	MOSI, \overline{CS} , SCLK			-0.3		$P_{DD} \times 0.15$
Voltage Output Low	MISO, current = 3 mA				0.5	V
Voltage Output High	MISO, current = -2 mA			$P_{DD} - 0.5$		V
Pull-Up Current	$\overline{CS}, P_{DD} = 3.3$ V, $\overline{CS} = P_{DD} \times 0.15$			60	200	μA
	$\overline{CS}, P_{DD} = 5$ V, $\overline{CS} = P_{DD} \times 0.15$			80	300	μA
MEMORY REGISTERS	See the Memory Register Definitions section					
Temperature Register				0		LSB
Value at 45°C				5		$LSB/^\circ\text{C}$
Scale Factor				80		$LSB/^\circ/\text{sec}$
Quadrature, Self-Test, and Rate Registers						
Scale Factor						
POWER SUPPLY						
Supply Voltage		P_{DD}	3.15		5.25	V
Quiescent Supply Current		I_{DD}		6.0	8.0	mA
Turn-On Time	Power-on to $0.5^\circ/\text{sec}$ of final value			100		ms

¹ Maximum limit is guaranteed by Analog Devices, Inc., characterization.

² Cross-axis sensitivity specification does not include effects due to device mounting on a printed circuit board (PCB).

³ Minimum and maximum limits are guaranteed by design.

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, 0.5 ms)	
Unpowered	2000 g
Powered	2000 g
Supply Voltage (P_{DD})	-0.3 V to +6.0 V
Output Short-Circuit Duration (Any Pin to Ground)	Indefinite
Operating Temperature Range	
LCC_V Package	-55°C to +125°C
SOIC_CAV Package	-40°C to +125°C
Storage Temperature Range	
LCC_V Package	-65°C to +150°C
SOIC_CAV Package	-40°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

θ_{JA} is specified for the worst-case conditions, that is, for a device soldered in a printed circuit board (PCB) for surface-mount packages.

Table 3. Thermal Resistance

Package Type	θ_{JA}	θ_{JC}	Unit
16-Lead SOIC_CAV (RG-16-1)	191.5	25	°C/W
14-Lead Ceramic LCC_V (EY-14-1)	185.5	23	°C/W

RATE SENSITIVE AXIS

The ADXRS453 is available in two package options.

- The SOIC_CAV package is for applications that require z-axis (yaw) rate sensing.
- The LCC_V (vertical mount) package is for applications that require x-axis or y-axis (pitch or roll) rate sensing. The package has terminals on two faces. However, the terminals on the back are for internal evaluation only and should not be used in the end application. The terminals on the bottom of the package incorporate metallization bumps that ensure a minimum solder thickness for improved solder joint reliability. These bumps are not present on the back terminals and, therefore, poor solder joint reliability can be encountered if the back terminals are used in the end application. For the outline dimensions of this package, see Figure 38.

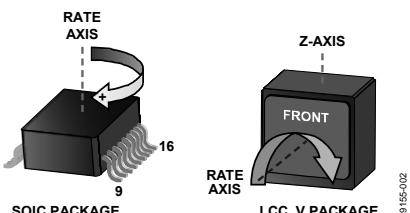


Figure 2. Rate Signal Increases with Clockwise Rotation

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS

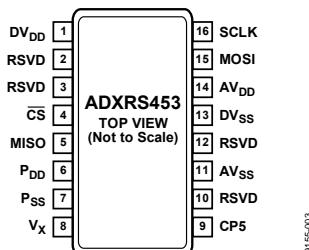
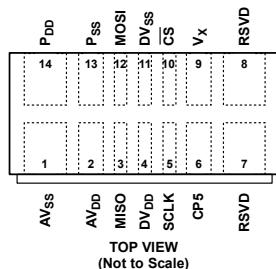


Figure 3. Pin Configuration, 16-Lead SOIC_CAV

Table 4. Pin Function Descriptions, 16-Lead SOIC_CAV

Pin No.	Mnemonic	Description
1	DV _{DD}	Digital Regulated Voltage. See Figure 25 for the application circuit diagram.
2	RSVD	Reserved. This pin must be connected to DV _{ss} .
3	RSVD	Reserved. This pin must be connected to DV _{ss} .
4	CS	Chip Select.
5	MISO	Master In/Slave Out.
6	P _{DD}	Supply Voltage.
7	P _{SS}	Switching Regulator Ground.
8	V _x	High Voltage Switching Node. See Figure 25 for the application circuit diagram.
9	CP5	High Voltage Supply. See Figure 25 for the application circuit diagram.
10	RSVD	Reserved. This pin must be connected to DV _{ss} .
11	AV _{ss}	Analog Ground.
12	RSVD	Reserved. This pin must be connected to DV _{ss} .
13	DV _{ss}	Digital Signal Ground.
14	AV _{DD}	Analog Regulated Voltage. See Figure 25 for the application circuit diagram.
15	MOSI	Master Out/Slave In.
16	SCLK	SPI Clock.

**NOTES**

1. THE PACKAGE HAS TERMINALS ON TWO FACES. HOWEVER, THE TERMINALS ON THE BACK ARE FOR INTERNAL EVALUATION ONLY AND SHOULD NOT BE USED IN THE END APPLICATION. THE TERMINALS ON THE BOTTOM OF THE PACKAGE INCORPORATE METALLIZATION BUMPS THAT ENSURE A MINIMUM SOLDER THICKNESS FOR IMPROVED SOLDER JOINT RELIABILITY. THESE BUMPS ARE NOT PRESENT ON THE BACK TERMINALS AND, THEREFORE, POOR SOLDER JOINT RELIABILITY CAN BE ENCOUNTERED IF THE BACK TERMINALS ARE USED IN THE END APPLICATION. FOR THE OUTLINE DIMENSIONS OF THIS PACKAGE, SEE FIGURE 38.

0915-5-004

Figure 4. Pin Configuration, 14-Terminal LCC_V

Table 5. Pin Function Descriptions, 14-Terminal LCC_V

Pin No.	Mnemonic	Description
1	AV _{SS}	Analog Ground.
2	AV _{DD}	Analog Regulated Voltage. See Figure 26 for the application circuit diagram.
3	MISO	Master In/Slave Out.
4	DV _{DD}	Digital Regulated Voltage. See Figure 26 for the application circuit diagram.
5	SCLK	SPI Clock.
6	CP5	High Voltage Supply. See Figure 26 for the application circuit diagram.
7	RSVD	Reserved. This pin must be connected to DV _{SS} .
8	RSVD	Reserved. This pin must be connected to DV _{SS} .
9	V _X	High Voltage Switching Node. See Figure 26 for the application circuit diagram.
10	CS	Chip Select.
11	DV _{SS}	Digital Signal Ground.
12	MOSI	Master Out/Slave In.
13	P _{SS}	Switching Regulator Ground.
14	P _{DD}	Supply Voltage.

TYPICAL PERFORMANCE CHARACTERISTICS

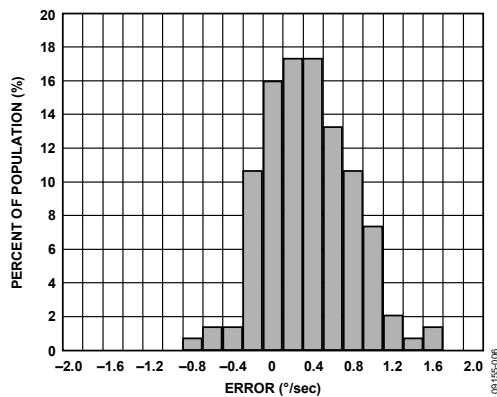


Figure 5. SOIC_CAV Null Accuracy at 25°C

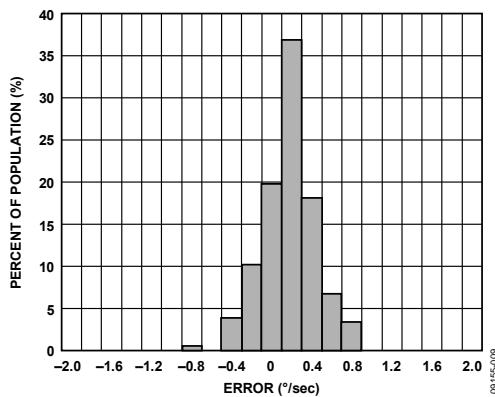


Figure 8. LCC_V Null Accuracy at 25°C

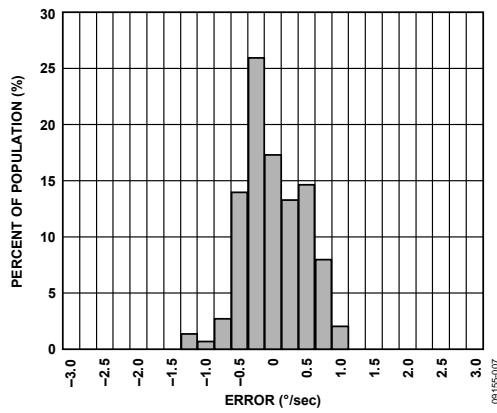


Figure 6. SOIC_CAV Null Drift over Temperature

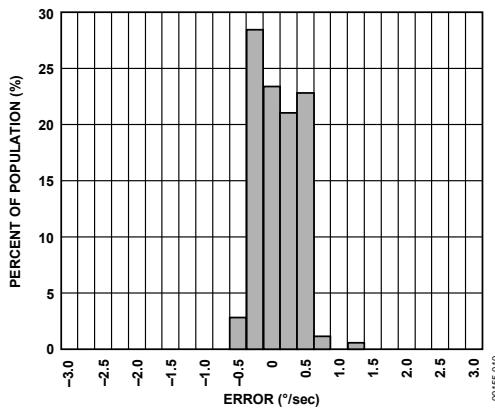


Figure 9. LCC_V Null Drift over Temperature

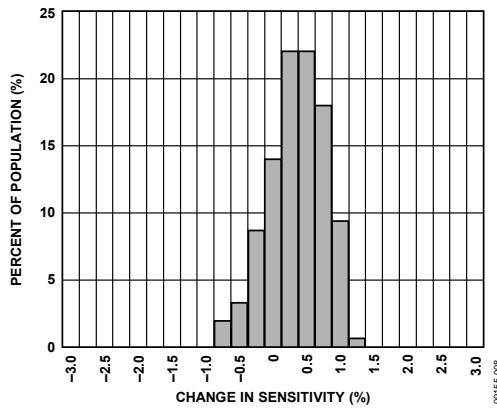


Figure 7. SOIC_CAV Sensitivity Error at 25°C

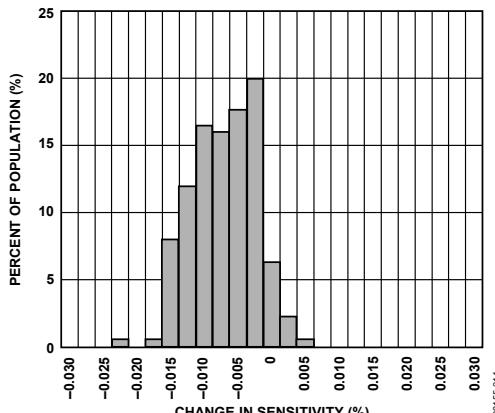


Figure 10. LCC_V Sensitivity Error at 25°C

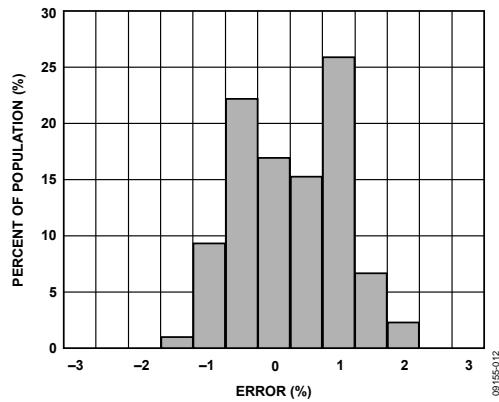


Figure 11. SOIC_CAV Sensitivity Drift over Temperature

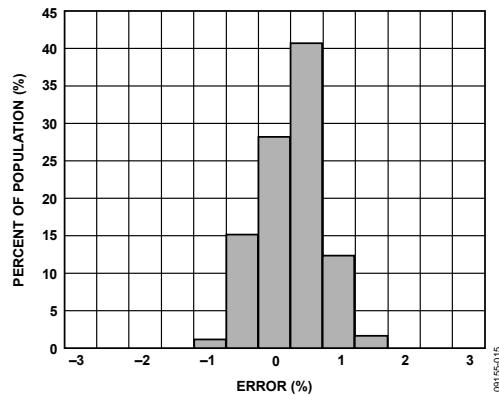


Figure 14. LCC_V Sensitivity Drift over Temperature

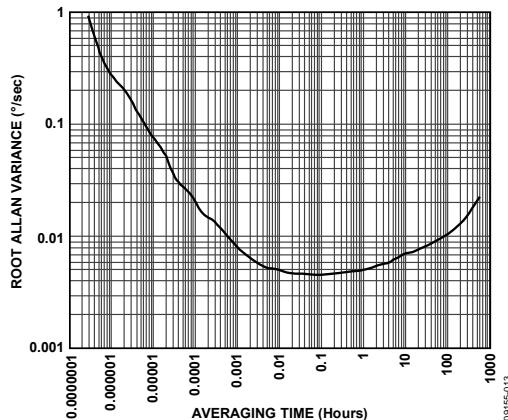


Figure 12. Typical Root Allan Variance at 40°C

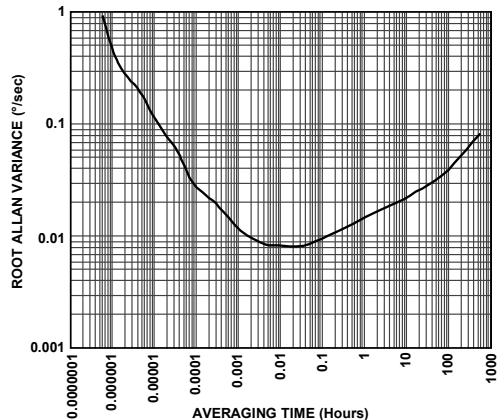


Figure 15. Typical Root Allan Variance at 105°C

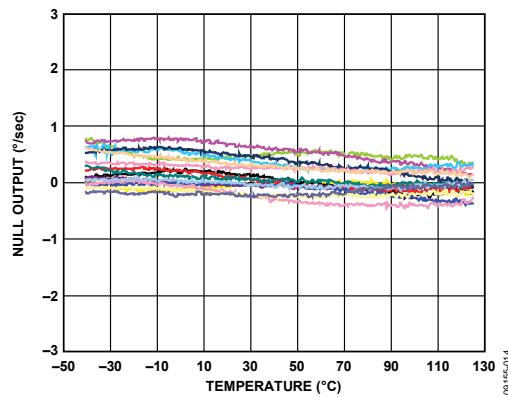


Figure 13. Null Output over Temperature, 16 Devices Soldered on PCB

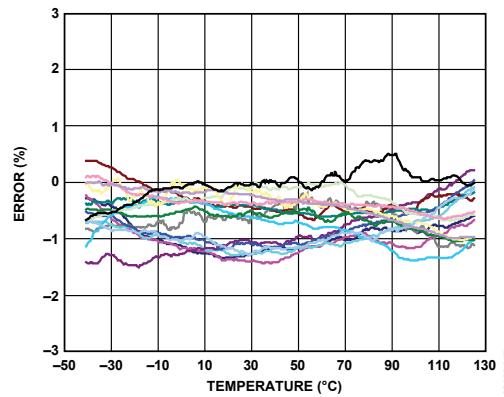


Figure 16. Sensitivity over Temperature, 16 Devices Soldered on PCB

THEORY OF OPERATION

The ADXRS453 operates on the principle of a resonator gyroscope. Figure 17 shows a simplified version of one of four polysilicon sensing structures. Each sensing structure contains a dither frame that is electrostatically driven to resonance. This produces the necessary velocity element to produce a Coriolis force when the device experiences angular rate. In the SOIC_CAV package, the ADXRS453 is designed to sense a z-axis (yaw) angular rate; the LCC_V vertical mount package orients the device such that it can sense pitch or roll angular rate on the same PCB.

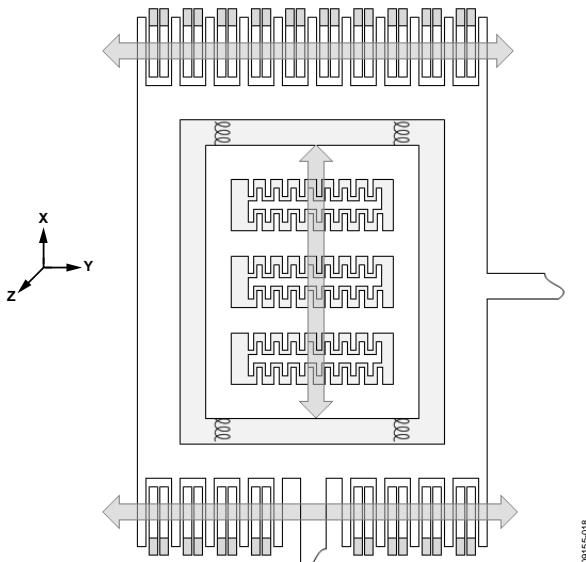


Figure 17. Simplified Gyroscope Sensing Structure

When the sensing structure is exposed to angular rate, the resulting Coriolis force couples into an outer sense frame, which contains movable fingers that are placed between fixed pickoff fingers. This forms a capacitive pickoff structure that senses Coriolis motion. The resulting signal is fed to a series of gain and demodulation stages that produce the electrical rate signal output. The quad sensor design rejects linear and angular acceleration, including external g-forces and vibration. This is achieved by mechanically coupling the four sensing structures such that external g-forces appear as common-mode signals that can be removed by the fully differential architecture implemented in the ADXRS453.

The resonator requires 22.5 V (typical) for operation. Because only 5 V is typically available in most applications, a switching regulator is included on chip.

CONTINUOUS SELF-TEST

The ADXRS453 gyroscope implements a complete electro-mechanical self-test. An electrostatic force is applied to the gyroscope frame, resulting in a deflection of the capacitive sense fingers. This deflection is exactly equivalent to deflection that occurs as a result of external rate input. The output from the beam structure is processed by the same signal chain as a true rate output signal, providing complete coverage of both the electrical and mechanical components.

The electromechanical self-test is performed continuously during operation at a rate higher than the output bandwidth of the device. The self-test routine generates equivalent positive and negative rate deflections. This information can then be filtered with no overall effect on the demodulated rate output.

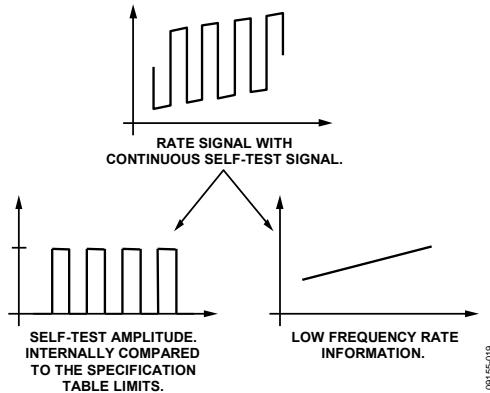


Figure 18. Continuous Self-Test Demodulation

The difference amplitude between the positive and negative self-test deflections is filtered to $f_0/8000$ (~1.95 Hz) and is continuously monitored and compared to hard-coded self-test limits. If the measured amplitude exceeds these limits (listed in Table 1), one of two error conditions is asserted, depending on the magnitude of the self-test error.

- For less severe self-test error magnitudes, the CST bit of the fault register is asserted. However, the status bits (ST[1:0]) in the sensor data response remain set to 01 for valid sensor data.
- For more severe self-test errors, the CST bit of the fault register is asserted and the status bits (ST[1:0]) in the sensor data response are set to 00 for invalid sensor data.

Table 1 lists the thresholds for both of these failure conditions. If desired, the user can access the self-test information by issuing a read command to the self-test memory register (Address 0x04). See the SPI Communication Protocol section for more information about error reporting.

MECHANICAL PERFORMANCE

The ADXRS453 has excellent shock and vibration rejection. Figure 19 shows the output noise response of the ADXRS453 in a vibration free environment. Figure 20 shows the response of the same device to 15 g rms random vibration (50 Hz to 5 kHz). As shown in Figure 20, no frequencies are particularly sensitive to vibration. Response to vibration in all axes is similar.

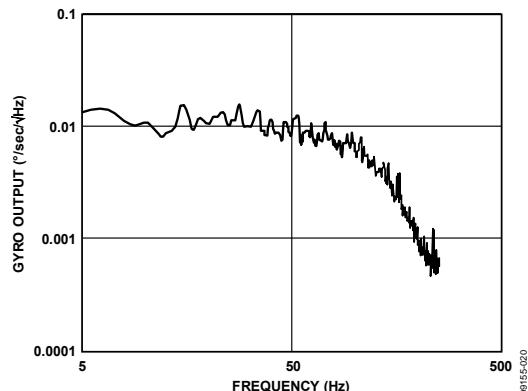


Figure 19. ADXRS453 Output Noise Response with No Vibration Applied

Shock response is also excellent, as shown in Figure 21 and Figure 22. Figure 21 shows a 99 g input stimulus applied to each axis, and Figure 22 shows the typical response to this shock in each axis. Shock response of 0.01°/sec/g is apparent.

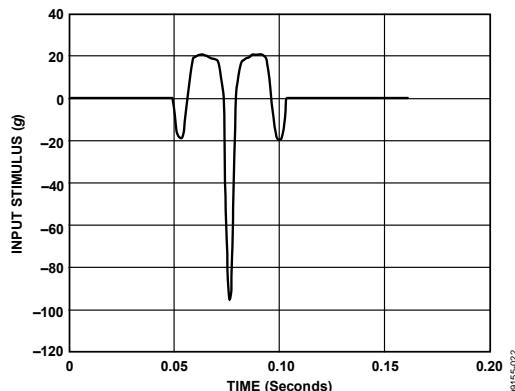


Figure 21. 99 g Shock Input

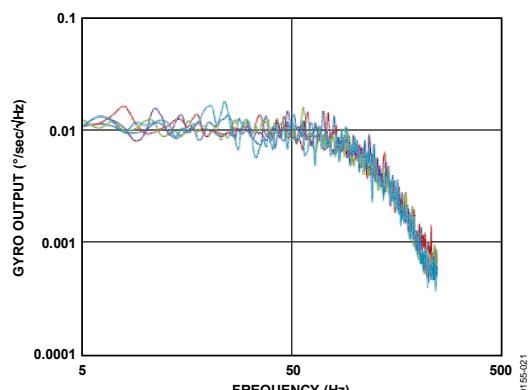


Figure 20. ADXRS453 Output Noise Response with 15 g RMS Random Vibration (50 Hz to 5 kHz) Applied

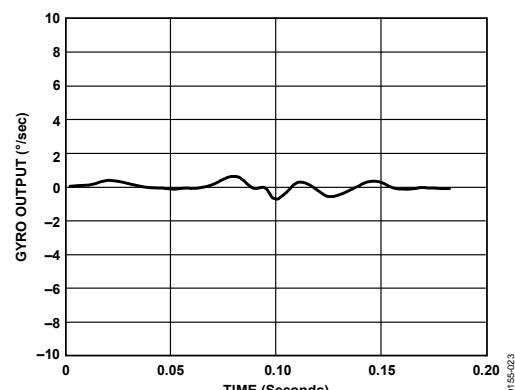


Figure 22. Typical Output Response Due to 99 g Shock (see Figure 21)

NOISE PERFORMANCE

The ADXRS453 noise performance is very consistent from device to device and varies very predictably with temperature. Table 6 contains statistical noise data at three temperature points for a large population of ADXRS453 devices (more than 3000 parts from several manufacturing lots).

Table 6. Statistical Noise Data

Temperature	Noise ($^{\circ}/\text{sec}/\sqrt{\text{Hz}}$)	
	Mean	Standard Deviation
-40°C	0.0109	0.0012
+25°C	0.0149	0.0015
+105°C	0.0222	0.0019

Noise increases fairly linearly with temperature, as shown in Figure 23.

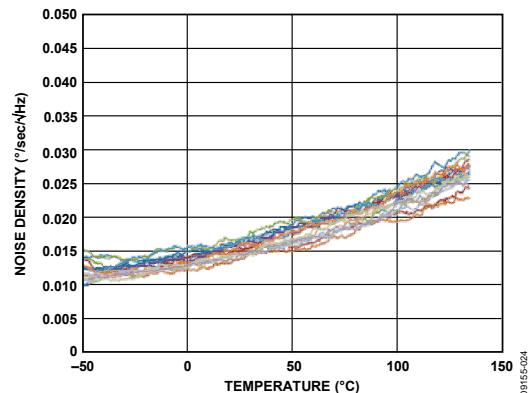


Figure 23. Noise Density vs. Temperature, 16 Devices

09155/024

APPLICATIONS INFORMATION

CALIBRATED PERFORMANCE

The ADXRS453 gyroscope uses internal EEPROM memory to store its temperature calibration information. The calibration information is encoded into the device during factory test. The calibration data is used to perform offset, gain, and self-test corrections over temperature. By storing this information internally, the ADXRS453 eliminates the need for the customer to perform system level temperature calibration.

MECHANICAL CONSIDERATIONS FOR MOUNTING

Mount the ADXRS453 in a location close to a hard mounting point of the PCB. Mounting the ADXRS453 at an unsupported PCB location (that is, at the end of a lever or in the middle of a trampoline, as shown in Figure 24) can result in apparent measurement errors because the gyroscope is subject to the resonant vibration of the PCB. Locating the gyroscope near a hard mounting point helps to ensure that any PCB resonances at the gyroscope are above the frequency at which harmful aliasing with the internal electronics can occur. To ensure that aliased signals do not couple into the baseband measurement range, design the module so that the first system level resonance occurs at a frequency higher than 800 Hz.

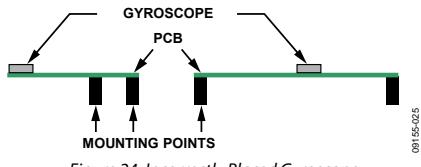


Figure 24. Incorrectly Placed Gyroscope

APPLICATION CIRCUITS

Figure 25 and Figure 26 show the recommended application circuits for the ADXRS453 gyroscope. These application circuits provide a connection reference for the available package types. Note that DV_{DD}, AV_{DD}, and P_{DD} are all individually connected to ground through 1 μ F capacitors; do not connect these supplies together. In addition, an external diode and inductor must be connected for proper operation of the internal shunt regulator (see Table 7). These components allow the internal resonator drive voltage to reach its required level.

Table 7. Components for ADXRS453 Application Circuits

Component	Qty	Description
Inductor	1	470 μ H
Diode	1	>24 V breakdown voltage
Capacitor	3	1 μ F
Capacitor	1	100 nF

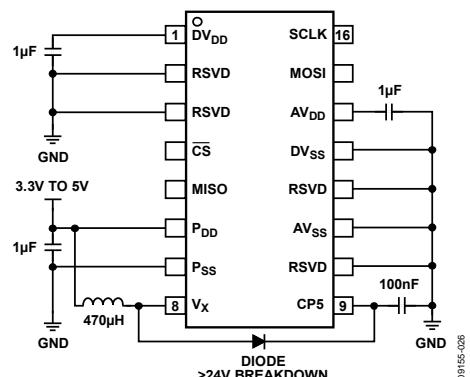


Figure 25. Recommended Application Circuit, SOIC_CAV Package

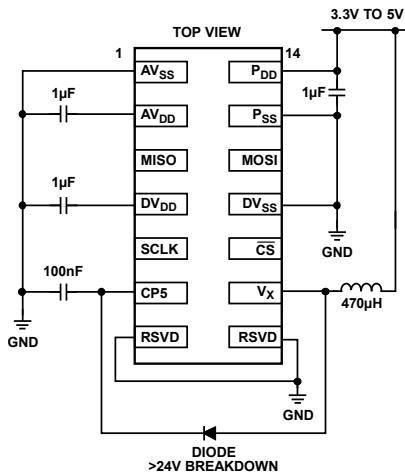


Figure 26. Recommended Application Circuit, LCC_V Package

ADXRS453 SIGNAL CHAIN TIMING

The ADXRS453 primary signal chain is shown in Figure 27. The signal chain is the series of necessary functional circuit blocks through which the rate data is generated and processed. This sequence of electromechanical elements determines how quickly the device can translate an external rate input stimulus to an SPI word that is sent to the master device.

The group delay, which is a function of the filter characteristic, is the time required for the output of the low-pass filter to be within 10% of the external rate input. In Figure 27, the group delay is shown to be ~4 ms. Additional delay can be observed due to the timing of SPI transactions and the population of the rate data into the internal device registers. Figure 27 illustrates this delay through each element of the signal chain.

The transfer function for the rate data LPF is given as

$$\left[\frac{1 - Z^{-64}}{1 - Z^{-1}} \right]^2$$

where:

$$T = \frac{1}{f_0} = \frac{1}{16 \text{ kHz (typ)}}$$

(f_0 is the resonant frequency of the ADXRS453.)

The transfer function for the continuous self-test LPF is given as

$$\frac{1}{64 - (63 \times Z^{-1})}$$

where:

$$T = \frac{16}{f_0} = 1 \text{ ms (typ)}$$

(f_0 is the resonant frequency of the ADXRS453.)

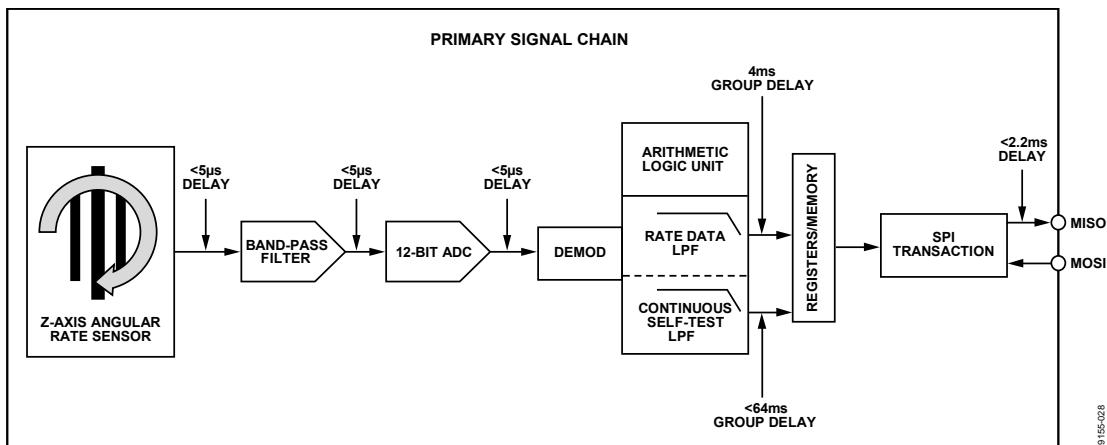


Figure 27. Primary Signal Chain and Associated Delays

09155-028

SPI COMMUNICATION PROTOCOL

COMMAND/RESPONSE

Input/output is handled through a 32-bit command/response SPI interface. With the command/response SPI interface, the response to a command is issued during the next sequential SPI exchange (see Figure 28).

The format for the interface is defined as follows:

Clock Phase = Clock Polarity = 0

Table 9 shows the commands that can be sent from the master device to the gyroscope. Table 10 shows the responses to these commands from the gyroscope. For descriptions of the bits in the commands and responses, see the Command/Response Bit Definitions section and the Fault Register Bit Definitions section.

The device response to the initial command is 0x00000001. This response prevents the transmission of random data to the master device upon the initial command/response exchange.

The SPI interface uses the ADXRS453 pins described in Table 8.

Table 8. SPI Signals

Signal	Pin	Description
Serial Clock	SCLK	Exactly 32 clock cycles during \overline{CS} active
Chip Select	\overline{CS}	Active low chip select pin
Master Out/ Slave In	MOSI	Input for data sent to the gyroscope (slave) from the main controller (master)
Master In/ Slave Out	MISO	Output for data sent to the main controller (master) from the gyroscope (slave)

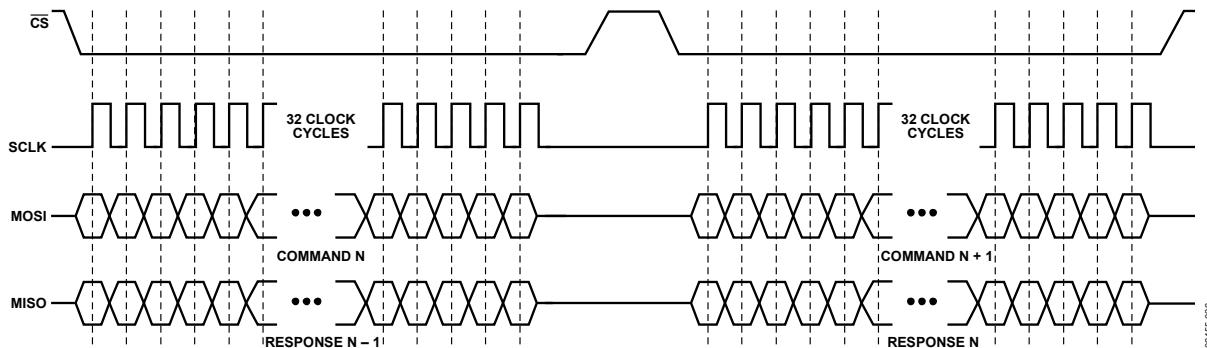


Figure 28. SPI Protocol

09155-029

Table 9. SPI Commands

Command	Bit																																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Sensor Data	SQ1	SQ0	1	SQ2																														CHK P
Read	1	0	0	SM2	SM1	SM0	A8	A7	A6	A5	A4	A3	A2	A1	A0																	P		
Write	0	1	0	SM2	SM1	SM0	A8	A7	A6	A5	A4	A3	A2	A1	A0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	P		

Table 10. SPI Responses

Command	Bit																														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Sensor Data	SQ2	SQ1	SQ0	P0	ST1	ST0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0			PLL Q	NVM	POR	PWR	CST	CHK P1	
Read	0	1	0	P0	1	1	1	0	SM2	SM1	SM0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0			P1	
Write	0	0	1	P0	1	1	1	0	SM2	SM1	SM0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0			P1	
R/W Error	0	0	0	P0	1	1	1	0	SM2	SM1	SM0	0	0	SPI RE	DU										PLL Q	NVM	POR	PWR	CST	CHK P1	

DEVICE DATA LATCHING

To allow for rapid acquisition of data from the ADXRS453, device data latching is implemented, as shown in Figure 29. When the chip select pin is asserted (\overline{CS} goes low), the data in the device is latched into memory. When the full MOSI command is received and the chip select pin is deasserted (\overline{CS} goes high), the data is shifted into the SPI port registers in preparation for the next sequential command/response exchange. Device data latching allows for an extremely fast sequential transfer delay of 0.1 μ s (see Table 11).

Note that the transmitted data is only as recent as the sequential transmission delay implemented by the system. Conditions that result in a sequential transfer delay of several seconds cause the next sequential device response to contain data that is several seconds old.

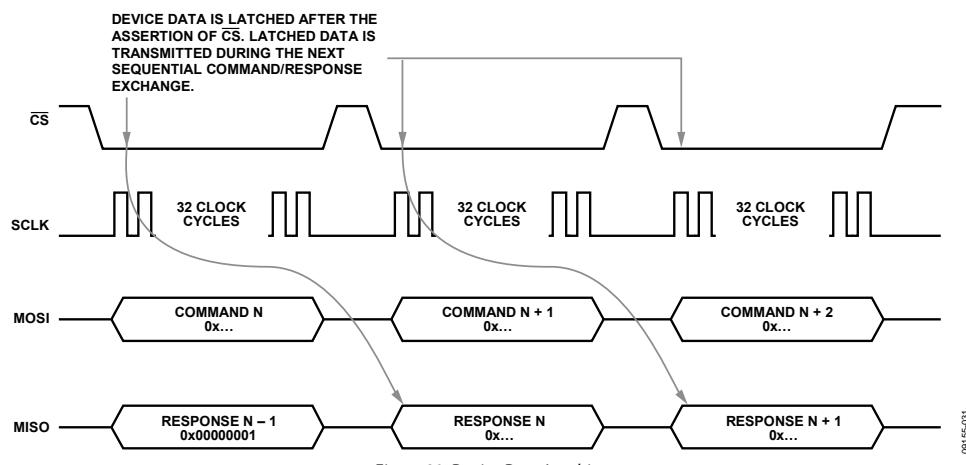


Figure 29. Device Data Latching

SPI TIMING CHARACTERISTICS

The following conditions apply to the SPI command/response timing characteristics in Table 11:

- All timing parameter are guaranteed through characterization.
- All timing is shown with respect to 10% DV_{DD} and 90% of the actual delivered voltage waveform.

- Parameters are valid for $3.0 \text{ V} \leq DV_{DD} \leq 5.5 \text{ V}$.
- Capacitive load for all signals is assumed to be $\leq 80 \text{ pF}$.
- Ambient temperature is $-40^\circ\text{C} \leq T_A \leq +105^\circ\text{C}$.
- The MISO pull-up is $47 \text{ k}\Omega$ or $110 \mu\text{A}$.

Table 11. SPI Command/Response Timing Characteristics

Symbol	Min	Max	Unit	Description
f _{OP}		8.08	MHz	SPI operating frequency
t _{SCLKH}	$1/2 \times t_{SCLK} - 13$		ns	SCLK high time
t _{SCLKL}	$1/2 \times t_{SCLK} - 13$		ns	SCLK low time
t _{SCLK}	123.7		ns	SCLK period
t _F	5.5	13	ns	SCLK fall time
t _R	5.5	13	ns	SCLK rise time
t _{SU}	37		ns	Data input (MOSI) setup time
t _{HIGH}	49		ns	Data input (MOSI) hold time
t _A		20	ns	Data output (MISO) access time
t _V		40	ns	Data output (MISO) valid after SCLK
t _{LAG_MISO}	0		ns	Data output (MISO) lag time
t _{DIS}		40	ns	Data output (MISO) disable time
t _{LEAD}	$1/2 \times t_{SCLK}$		ns	Enable (\overline{CS}) lead time
t _{LAG_CS}	$1/2 \times t_{SCLK}$		ns	Enable (\overline{CS}) lag time
t _{TD}	0.1		μs	Sequential transfer delay

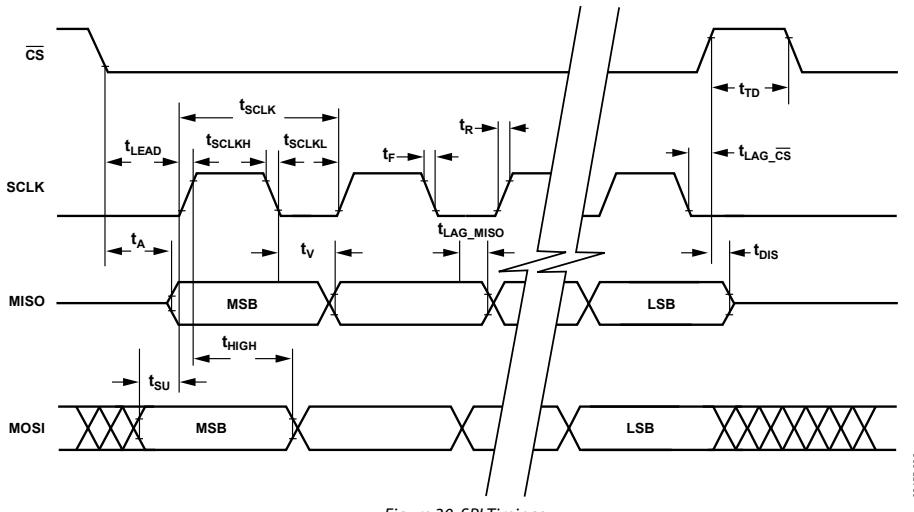


Figure 30. SPI Timings

00155030

COMMAND/RESPONSE BIT DEFINITIONS**Table 12. SPI Interface Bit Definitions**

Bits	Description
SQ2 to SQ0	Sequence bits (from master)
SM2 to SM0	Sensor module bits (from master)
A8 to A0	Register address
D15 to D0	Data
P	Command odd parity
SPI	SPI command/response
RE	Request error
DU	Data unavailable
ST1, ST0	Status bits
P0	Response, odd parity, Bits[31:16]
P1	Response, odd parity, Bits[31:0]

SQ2 to SQ0 Bits

The SQ2 to SQ0 bits provide the system with a means of synchronizing the data samples that are received from multiple sensors. To facilitate correct synchronization, the ADXRS453 gyroscope includes the SQ[2:0] bits in the response sequence as they were received in the request.

SM2 to SM0 Bits

The SM2 to SM0 bits are the sensor module bits from the master device. These bits are not implemented in the ADXRS453 and are hard-coded to 000 for all occurrences.

A8 to A0 Bits

The A8 to A0 bits represent the memory address for data read or data write. These bits should be supplied by the master when the memory registers are being accessed; these bits are ignored for all sensor data requests. For a complete description of the available memory registers, see the Memory Register Definitions section.

D15 to D0 Bits

The D15 to D0 bits are the 16-bit device data, which can contain any of the following:

- Data from the master to be written to a memory register, as specified by the A8 to A0 bits.
- Sensor rate output data from the slave.
- Device data from the slave read from the memory register specified by the A8 to A0 bits, as well as the data from the next sequential register.
- Following a write command, the 16-bit data that is written to the specified memory register in the ADXRS453 and is reflected back to the master device for correlation.

P Bit

A parity bit (P) is required for all master-to-slave data transmissions. The communication protocol requires one parity bit to achieve odd parity for the entire 32-bit command. “Don’t care” bits are also factored into the parity calculation.

SPI Bit

The SPI bit is set when either of the following occurs:

- Too many or not enough bits were transmitted.
- A message from the control module contains a parity error.

A SPI error causes the device to issue a R/W error response regardless of the SPI command type issued by the master device (see Table 10). In addition, any error during a sensor data request results in the device issuing a read/write error.

RE Bit

The request error (RE) bit is the communication error bit transmitted from the ADXRS453 device to the control module. Request errors can occur when

- An invalid command is sent from the control module.
- A read/write command specifies an invalid memory register.
- A write command attempted to write to a nonwritable memory register.

DU Bit

After the chip select pin is deasserted (\overline{CS} goes high), the user must wait 0.1 μ s before reasserting the \overline{CS} pin to initiate another command/response frame with the device. Failure to adhere to this timing specification may result in a data unavailable (DU) error.

ST1 and ST0 Bits

The status bits (ST1 and ST0) are used to signal to the master device the type of data contained in the response message (see Table 13).

Table 13. Status Bit Code Definitions

ST[1:0]	Contents of Bits[D15:D0]
00	Invalid data for sensor data response
01	Valid sensor data
10	Sensor self-test data
11	Read/write response

Either of the following conditions can result in the ST[1:0] bits being set to 00 during a sensor data response:

- The self-test response is sufficiently different from its nominal value (see the Specifications section for the appropriate limits).
- The PLL fault bit is active (see the PLL Bit section).

P0 Bit

P0 is the parity bit that establishes odd parity for Bits[31:16] of the device response.

P1 Bit

P1 is the parity bit that establishes odd parity for the entire 32-bit device response.

FAULT REGISTER BIT DEFINITIONS

Table 14 describes the bits available for signaling faults to the user. The individual bits of the fault registers are updated asynchronously, depending on their respective detection criteria; however, it is recommended that the fault registers be read at a rate of at least 250 Hz. When asserted, an individual status bit is not deasserted until it is read by the master device. If the error persists after a fault register read, the status bit is immediately reasserted and remains asserted until the next sequential command/response exchange. The bits in the FAULT0 register are appended to every sensor data response (see Table 10). Both fault registers can be accessed by issuing a read command to Address 0x0A.

Table 14. Fault Register Bit Definitions

Register	Bit Name	Description
FAULT1	Fail	Failure that sets the ST[1:0] bits to 00
	AMP	Amplitude detection failure
	OV	Regulator overvoltage
	UV	Regulator undervoltage
FAULT0	PLL	Phase-locked loop failure
	Q	Quadrature error
	NVM	Nonvolatile memory fault
	POR	Power-on or reset failed to initialize
	PWR	Power regulation failed due to over-voltage or undervoltage condition
	CST	Continuous self-test failure or amplitude detection failed
	CHK	Check: generate faults

Fail Bit

The fail flag is asserted when the ST[1:0] bits are set to 00 (see the ST1 and ST0 Bits section). Assertion of the fail bit indicates that the device has experienced a gross failure and that the sensor data could be invalid.

AMP Bit

The AMP fault bit is asserted when the measured amplitude of the silicon resonator has been significantly reduced. This condition can occur if the voltage supplied to CP5 falls below the requirements of the internal voltage regulator. This fault bit is OR'ed with the CST fault bit; therefore, during a sensor data request, the CST bit position represents either an AMP failure or a CST failure. The full fault register can be read from memory to determine the specific failure.

OV Bit

The OV fault bit is asserted if the internally regulated voltage (nominally 3 V) is observed to exceed 3.3 V. This measurement is low-pass filtered to prevent artifacts such as noise spikes from asserting a fault condition. When an OV fault occurs, the PWR fault bit is asserted simultaneously. Because the OV fault bit is not transmitted as part of a sensor data response, it is recommended that the user read back the FAULT1 and FAULT0 memory registers upon the assertion of a PWR error to determine the specific error condition.

UV Bit

The UV fault bit is asserted if the internally regulated voltage (nominally 3 V) is observed to be less than 2.77 V. This measurement is low-pass filtered to prevent artifacts such as noise spikes from asserting a fault condition. When a UV fault occurs, the PWR fault bit is asserted simultaneously. Because the UV fault bit is not transmitted as part of a sensor data response, it is recommended that the user read back the FAULT1 and FAULT0 memory registers upon the assertion of a PWR error to determine the specific error condition.

PLL Bit

The PLL bit indicates that the device has experienced a failure in the phase-locked loop functional circuit block. This occurs when the PLL fails to achieve synchronization with the resonator structure. If the PLL status flag is active, the ST[1:0] bits of the sensor data response are set to 00, indicating that the response contains potentially invalid rate data.

Q Bit

A Q fault is asserted based on two independent quadrature calculations.

- The quad memory register (Address 0x08) contains a value corresponding to the total instantaneous quadrature present in the device. If this value exceeds 4096 LSB, a Q fault is issued.
- An internal quadrature accumulator records the amount of quadrature correction performed by the ADXRS453. A Q fault is issued when the quadrature error present in the device has contributed to an equivalent of 4°/sec (typical) of rate offset.

NVM Bit

An NVM error is transmitted to the control module when the internal nonvolatile memory data fails a checksum calculation. This check is performed once every 50 µs and does not include the PIDx memory registers.

POR Bit

An internal check is performed on device startup to ensure that the volatile memory of the device is functional. This is accomplished by programming a known value from the device ROM into a volatile memory register. This value is then continuously compared to the known value in ROM every 1 µs for the duration of the device operation. If the value stored in the volatile memory changes or does not match the value stored in ROM, the POR error flag is asserted. The value stored in ROM is rewritten to the volatile memory upon a device power cycle.

PWR Bit

The device performs a continuous check of the internal 3 V regulated voltage level. If either an overvoltage (OV) or undervoltage (UV) fault is asserted, the PWR bit is also asserted. This condition occurs if the regulated voltage is observed to be either above 3.3 V or below 2.77 V. An internal low-pass filter removes high frequency glitching effects to prevent the PWR bit from being asserted unnecessarily. To determine whether the fault is a result of an overvoltage or undervoltage condition, the OV and UV fault bits must be read.

CST Bit

The ADXRS453 is designed with continuous self-test functionality. The measured self-test amplitudes are compared to the limits presented in Table 1. Deviations from these values result in reported self-test errors. The two thresholds for a self-test failure are as follows:

- Self-test value $> \pm 512$ LSB from nominal results in the assertion of the self-test flag in the fault register.
- Self-test value $> \pm 1856$ LSB from nominal results in the assertion of the self-test flag in the fault register and the setting of the ST[1:0] bits to 00, indicating that the rate data contained in the sensor data response is potentially invalid.

CHK Bit

The CHK bit is transmitted by the control module to the ADXRS453 as a method of generating faults. By asserting the CHK bit, the device creates conditions that result in the generation of all faults represented in the fault registers. For example, the self-test amplitude is deliberately altered to exceed the fault detection threshold, resulting in a self-test error. In this way, the device is capable of checking both its ability to detect a fault condition and its ability to report that fault condition to the control module.

The fault conditions are initiated nearly simultaneously; however, the timing for receiving fault codes when the CHK bit is asserted depends on the time required to generate each unique fault. It takes no more than 50 ms for all internal faults to be generated and the fault register to be updated to reflect the condition of the device. Until the CHK bit is cleared, the status bits (ST[1:0]) are set to 10, indicating that the data should be interpreted by the control module as self-test data. After the CHK bit is deasserted, an additional 50 ms are required for the fault conditions to decay and for the device to return to normal operation. See the Recommended Start-Up Sequence with CHK Bit Assertion section for the proper methodology for asserting the CHK bit.

RECOMMENDED START-UP SEQUENCE WITH CHK BIT ASSERTION

Figure 31 illustrates a recommended start-up sequence that can be implemented by the user. Alternate start-up sequences can be used, but the response from the ADXRS453 must be handled correctly. If the start-up sequence is implemented immediately after power is applied to the device, the total time to implement the following fault detection routine is approximately 200 ms.

As described in the Device Data Latching section, the data present in the device upon the assertion of the CS signal is used

in the next sequential command/response exchange. This results in an apparent one-transaction delay before the data resulting from the assertion of the CHK bit is reported by the device. For all other read/write interactions with the device, no such delay exists, and the MOSI command is serviced during the next sequential command/response exchange.

Note that if the CHK bit is deasserted and the user tries to obtain data from the device before the CST fault flag clears, the device reports the data as error data.

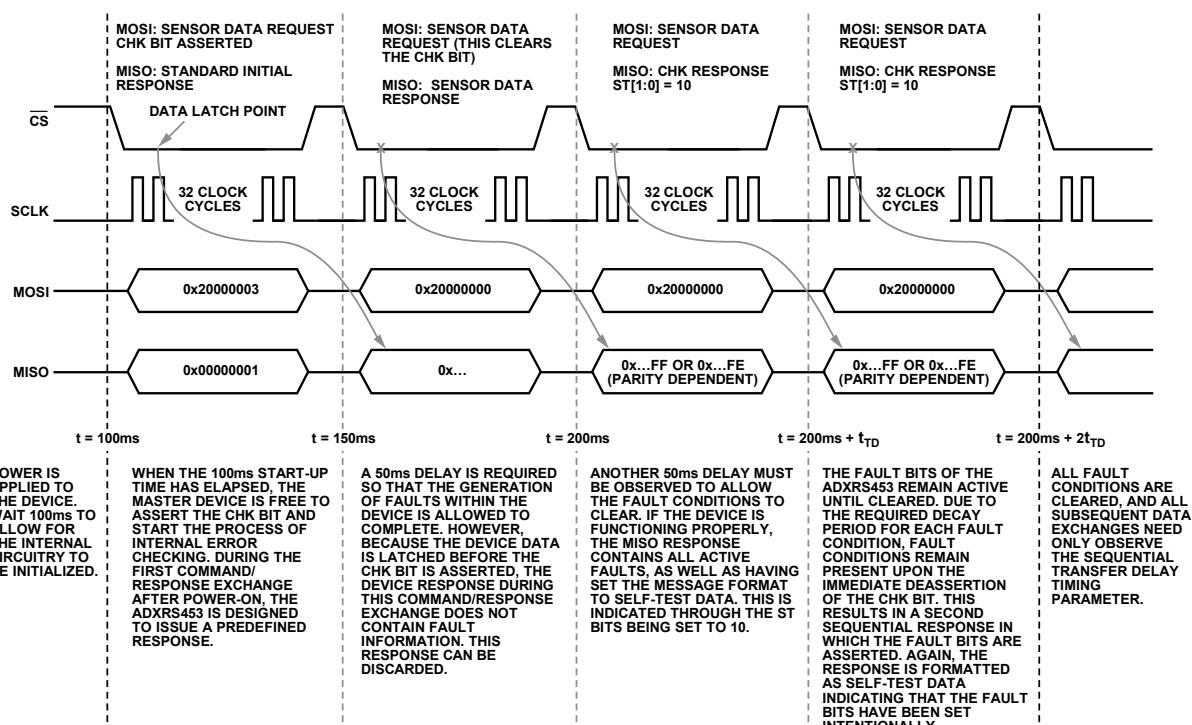


Figure 31. Recommended Start-Up Sequence

09155-02

RATE DATA FORMAT

The ADXRS453 gyroscope transmits rate data in a 16-bit format as part of a 32-bit SPI data frame. See Table 10 for the full 32-bit format of the sensor data response. The rate data is transmitted MSB first, from D15 to D0.

The data is formatted as a two's complement number with a scale factor of 80 LSB/°/sec. Therefore, the highest obtainable value for positive (clockwise) rotation is 0x7FFF (decimal +32,767), and the highest obtainable value for negative (counterclockwise) rotation is 0x8000 (decimal -32,768). Performance of the device is not guaranteed above $\pm 24,000$ LSB (± 300 °/sec).

Table 15. Rate Data

16-Bit Rate Data		Description
Decimal (LSBs)	Hex (D15:D0)	
+32,767	0x7FFF	Maximum possible positive data value (not guaranteed)
...
+24,000	0x5DC0	+300°/sec rotation (positive FSR)
...
+160	0x00A0	+2°/sec rotation
...
+80	0x0050	+1°/sec rotation
...
+40	0x0028	+0.5°/sec rotation
...
+20	0x0014	+0.025°/sec rotation
...
0	0x0000	Zero rotation value
...
-20	0xFFEC	-0.025°/sec rotation
...
-40	0xFFD8	-0.5°/sec rotation
...
-80	0xFFB0	-1°/sec rotation
...
-160	0xFF60	-2°/sec rotation
...
-24,000	0xA240	-300°/sec rotation (negative FSR)
...
-32,768	0x8000	Maximum possible negative data value (not guaranteed)

MEMORY MAP AND REGISTERS

MEMORY MAP

Table 16 provides a list of the memory registers that can be read from or written to by the user. See the SPI Communication Protocol section for the proper input sequence to read from or write to a specific memory register. Each memory register has eight bits of data; however, when a read request is performed, the data always returns as a 16-bit message. This is accomplished by appending the data from the next sequential register to the memory address that was specified.

Data is transmitted MSB first. For proper acquisition of data from the memory register, make the read request to the even-numbered register address only; for example, to read the LOCST_x registers, Address Register 0x04, but not Register 0x05. For a description of each memory register listed in Table 16, see the Memory Register Definitions section.

Table 16. Memory Register Map

Address	Register Name	D7 (MSB)	D6	D5	D4	D3	D2	D1	D0 (LSB)
0x00	RATE1	RTE15	RTE14	RTE13	RTE12	RTE11	RTE10	RTE9	RTE8
0x01	RATE0	RTE7	RTE6	RTE5	RTE4	RTE3	RTE2	RTE1	RTE0
0x02	TEM1	TEM9	TEM8	TEM7	TEM6	TEM5	TEM4	TEM3	TEM2
0x03	TEM0	TEM1	TEM0	Unused	Unused	Unused	Unused	Unused	Unused
0x04	LOCST1	LCST15	LCST14	LCST13	LCST12	LCST11	LCST10	LCST9	LCST8
0x05	LOCST0	LCST7	LCST6	LCST5	LCST4	LCST3	LCST2	LCST1	LCST0
0x06	HICST1	HCST15	HCST14	HCST13	HCST12	HCST11	HCST10	HCST9	HCST8
0x07	HICST0	HCST7	HCST6	HCST5	HCST4	HCST3	HCST2	HCST1	HCST0
0x08	QUAD1	QAD15	QAD14	QAD13	QAD12	QAD11	QAD10	QAD9	QAD8
0x09	QUAD0	QAD7	QAD6	QAD5	QAD4	QAD3	QAD2	QAD1	QAD0
0x0A	FAULT1	Unused	Unused	Unused	Unused	Fail	AMP	OV	UV
0x0B	FAULT0	PLL	Q	NVM	POR	PWR	CST	CHK	0
0x0C	PID1	PIDB15	PIDB14	PIDB13	PIDB12	PIDB11	PIDB10	PIDB9	PIDB8
0x0D	PID0	PIDB7	PIDB6	PIDB5	PIDB4	PIDB3	PIDB2	PIDB1	PIDB0
0x0E	SN3	SNB31	SNB30	SNB29	SNB28	SNB27	SNB26	SNB25	SNB24
0x0F	SN2	SNB23	SNB22	SNB21	SNB20	SNB19	SNB18	SNB17	SNB16
0x10	SN1	SNB15	SNB14	SNB13	SNB12	SNB11	SNB10	SNB9	SNB8
0x11	SN0	SNB7	SNB6	SNB5	SNB4	SNB3	SNB2	SNB1	SNB0

MEMORY REGISTER DEFINITIONS

The SPI-accessible memory registers are described in this section. As noted in the Memory Map section, when requesting data from a memory register, only the first sequential memory address should be addressed. The data returned by the device contains 16 bits of memory register information. Bits[15:8] contain the MSB of the requested information, and Bits[7:0] contain the LSB.

Rate (RATEx) Registers

Addresses: 0x00 (RATE1)

0x01 (RATE0)

Register update rate: $f_0/32$ (~485 Hz)

Scale factor: 80 LSB/°/sec

The RATEx registers contain the temperature compensated rate output of the device, filtered to $f_0/200$ (~77.5 Hz). This data can also be accessed by issuing a sensor data read request to the device. The data is presented as a 16-bit, two's complement number.

MSB								LSB							
D15	D14	D13	D12	D11	D10	D9	D8	LCST15	LCST14	LCST13	LCST12	LCST11	LCST10	LCST9	LCST8
RTE15	RTE14	RTE13	RTE12	RTE11	RTE10	RTE9	RTE8	D7	D6	D5	D4	D3	D2	D1	D0
RTE7	RTE6	RTE5	RTE4	RTE3	RTE2	RTE1	RTE0	LCST7	LCST6	LCST5	LCST4	LCST3	LCST2	LCST1	LCST0

Temperature (TEMx) Registers

Addresses: 0x02 (TEM1)

0x03 (TEM0)

Register update rate: $f_0/32$ (~485 Hz)

Scale factor: 5 LSB/°C

The TEMx registers contain a value corresponding to the temperature of the device. The data is presented as a 10-bit, two's complement number. 0 LSB corresponds to a temperature of approximately 45°C (see Table 17).

MSB								LSB									
D15	D14	D13	D12	D11	D10	D9	D8	HCST15	HCST14	HCST13	HCST12	HCST11	HCST10	HCST9	HCST8		
TEM9	TEM8	TEM7	TEM6	TEM5	TEM4	TEM3	TEM2	D7	D6	D5	D4	D3	D2	D1	D0		
TEM1	TEM0	Unused															

Table 17. Sample Temperatures and TEMx Register Contents

Temperature	Value of TEM1 and TEM0 Registers ¹
45°C	0000 0000 00XX XXXX
85°C	0011 0010 00XX XXXX
0°C	1100 0111 11XX XXXX

¹ X = don't care.

Low CST (LOCSTx) Registers

Addresses: 0x04 (LOCST1)

0x05 (LOCST0)

Register update rate: $f_0/16$ (~970 Hz)

Scale factor: 80 LSB/°/sec

The LOCSTx registers contain the value of the temperature compensated and low-pass filtered continuous self-test delta. This value is a measure of the difference between the positive and negative self-test deflections and corresponds to the values presented in Table 1. The device issues a CST error if the value of the self-test exceeds the established self-test limits. The self-test data is filtered to $f_0/8000$ (~1.95 Hz) to prevent false triggering of the CST fault bit. The data is presented as a 16-bit, two's complement number, with a scale factor of 80 LSB/°/sec.

MSB								LSB							
D15	D14	D13	D12	D11	D10	D9	D8	LCST15	LCST14	LCST13	LCST12	LCST11	LCST10	LCST9	LCST8
D7	D6	D5	D4	D3	D2	D1	D0	LCST7	LCST6	LCST5	LCST4	LCST3	LCST2	LCST1	LCST0

High CST (HICSTx) Registers

Addresses: 0x06 (HICST1)

0x07 (HICST0)

Register update rate: $f_0/16$ (~970 Hz)

Scale factor: 80 LSB/°/sec

The HICSTx registers contain the unfiltered self-test information. The HICSTx data can be used to supplement fault diagnosis in safety critical applications because sudden shifts in the self-test response can be detected. However, the CST bit of the fault register is not set when the HICSTx data is observed to exceed the self-test limits. Only the LOCSTx memory registers, which are designed to filter noise and the effects of sudden temporary self-test spiking due to external disturbances, control the assertion of the CST fault bit. The data is presented as a 16-bit, two's complement number.

MSB								LSB							
D15	D14	D13	D12	D11	D10	D9	D8	HCST15	HCST14	HCST13	HCST12	HCST11	HCST10	HCST9	HCST8
D7	D6	D5	D4	D3	D2	D1	D0	HCST7	HCST6	HCST5	HCST4	HCST3	HCST2	HCST1	HCST0

Quad Memory (QUADx) Registers

Addresses: 0x08 (QUAD1)
0x09 (QUAD0)

Register update rate: $f_0/64$ (~240 Hz)

Scale factor: 80 LSB/ $^{\circ}$ /sec equivalent

The QUADx registers contain a value corresponding to the amount of quadrature error present in the device at a given time. Quadrature can be likened to a measurement of the error of the motion of the resonator structure and can be caused by stresses and aging effects. The quadrature data is filtered to $f_0/200$ (~77.5 Hz) and can be read frequently to detect sudden shifts in the level of quadrature. The data is presented as a 16-bit, two's complement number.

MSB	LSB							
D15	D14	D13	D12	D11	D10	D9	D8	
QAD15	QAD14	QAD13	QAD12	QAD11	QAD10	QAD9	QAD8	
D7	D6	D5	D4	D3	D2	D1	D0	
QAD7	QAD6	QAD5	QAD4	QAD3	QAD2	QAD1	QAD0	

Fault (FAULTx) Registers

Addresses: 0x0A (FAULT1)
0x0B (FAULT0)

Register update rate: Not applicable

Scale factor: Not applicable

The FAULTx registers contain the state of the error flags in the device. The FAULT0 register is appended to the end of every device data transmission (see Table 10); however, this register can also be accessed independently through its memory location. The individual fault bits are updated asynchronously, requiring $<5\ \mu s$ to activate, as soon as the fault condition exists on chip. When toggled, each fault bit remains active until the fault register is read or a sensor data command is received. If the fault is still active after the bit is read, the fault bit is immediately reasserted.

MSB	LSB							
D15	D14	D13	D12	D11	D10	D9	D8	
Unused				Fail	AMP	OV	UV	
D7	D6	D5	D4	D3	D2	D1	D0	
PLL	Q	NVM	POR	PWR	CST	CHK	0	

Part ID (PIDx) Registers

Addresses: 0x0C (PID1)
0x0D (PID0)

Register update rate: Not applicable

Scale factor: Not applicable

The (PIDx) registers contain a 16-bit number that identifies the version of the ADXRS453. Combined with the serial number, this information allows for a higher degree of device individualization and tracking. The initial product ID is R01 (0x5201), with subsequent versions of silicon incrementing this value to R02, R03, and so on.

MSB	LSB							
D15	D14	D13	D12	D11	D10	D9	D8	
PIDB15	PIDB14	PIDB13	PIDB12	PIDB11	PIDB10	PIDB9	PIDB8	
D7	D6	D5	D4	D3	D2	D1	D0	

Serial Number (SNx) Registers

Addresses: 0x0E (SN3)
0x0F (SN2)
0x10 (SN1)
0x11 (SN0)

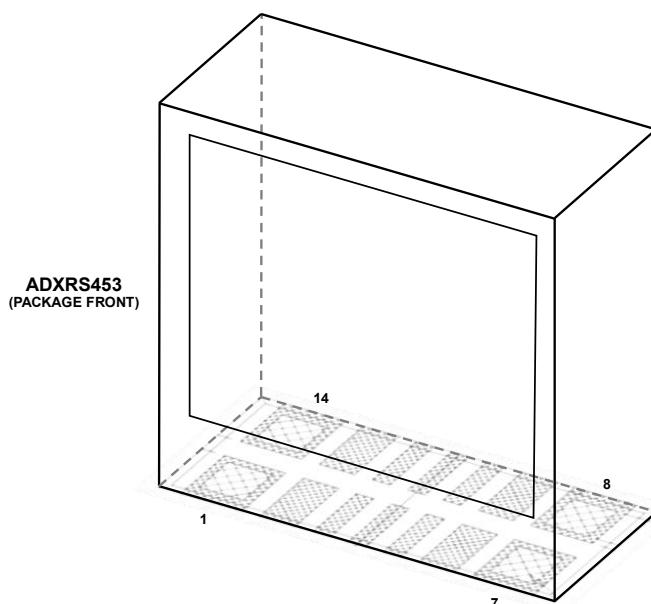
Register update rate: Not applicable

Scale factor: Not applicable

The SNx registers contain a 32-bit identification number that uniquely identifies the device. To read the entire serial number, two memory read requests must be initiated. The first read request to Address 0x0E returns the upper 16 bits of the serial number, and the following read request to Address 0x10 returns the lower 16 bits of the serial number.

MSB	LSB							
D31	D30	D29	D28	D27	D26	D25	D24	
SNB31	SNB30	SNB29	SNB28	SNB27	SNB26	SNB25	SNB24	
D23	D22	D21	D20	D19	D18	D17	D16	
SNB23	SNB22	SNB21	SNB20	SNB19	SNB18	SNB17	SNB16	
D15	D14	D13	D12	D11	D10	D9	D8	
SNB15	SNB14	SNB13	SNB12	SNB11	SNB10	SNB9	SNB8	
D7	D6	D5	D4	D3	D2	D1	D0	
SNB7	SNB6	SNB5	SNB4	SNB3	SNB2	SNB1	SNB0	

PACKAGE ORIENTATION AND LAYOUT INFORMATION



NOTES

1. THE PACKAGE HAS TERMINALS ON TWO FACES. HOWEVER, THE TERMINALS ON THE BACK ARE FOR INTERNAL EVALUATION ONLY AND SHOULD NOT BE USED IN THE END APPLICATION. THE TERMINALS ON THE BOTTOM OF THE PACKAGE INCORPORATE METALLIZATION BUMPS THAT ENSURE A MINIMUM SOLDER THICKNESS FOR IMPROVED SOLDER JOINT RELIABILITY. THESE BUMPS ARE NOT PRESENT ON THE BACK TERMINALS AND, THEREFORE, POOR SOLDER JOINT RELIABILITY CAN BE ENCOUNTERED IF THE BACK TERMINALS ARE USED IN THE END APPLICATION. FOR THE OUTLINE DIMENSIONS OF THIS PACKAGE, SEE FIGURE 38.

09155-033

Figure 32. 14-Lead Ceramic LCC_V, Vertical Mount

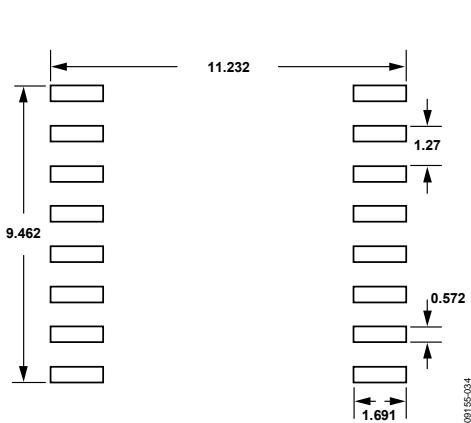


Figure 33. Sample SOIC_CAV Solder Pad Layout (Land Pattern), Dimensions Shown in Millimeters, Not to Scale

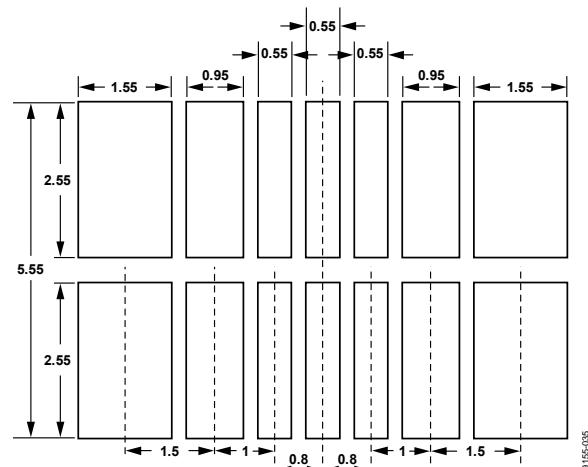


Figure 34. Sample LCC_V Solder Pad Layout (Land Pattern), Dimensions Shown in Millimeters, Not to Scale

SOLDER PROFILE

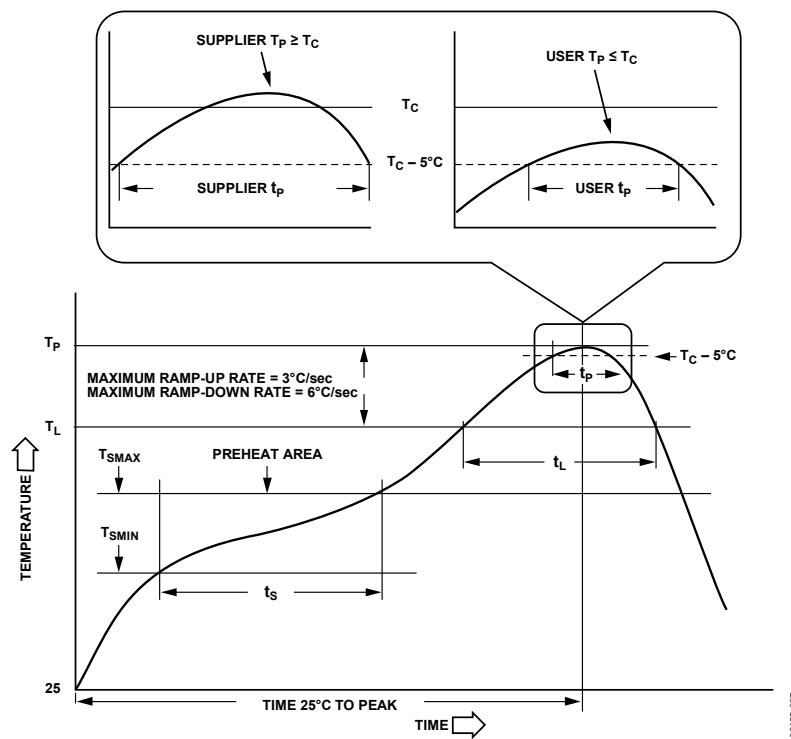


Figure 35. Recommended Soldering Profile

09155-037

Table 18. Recommended Soldering Profile Limits

Profile Feature	Sn63/Pb37	Pb-Free
Average Ramp Rate (T_L to T_p)	3°C/sec max	3°C/sec max
Preheat		
Minimum Temperature (T_{SMIN})	100°C	150°C
Maximum Temperature (T_{SMAX})	150°C	200°C
Time (T_{SMIN} to T_{SMAX}), t_s	60 sec to 120 sec	60 sec to 120 sec
Ramp-Up Rate (T_{SMAX} to T_L)	3°C/sec max	3°C/sec max
Time Maintained Above Liquidous (t_L)	60 sec to 150 sec	60 sec to 150 sec
Liquidous Temperature (T_L)	183°C	217°C
Classification Temperature (T_c) ¹	220°C	250°C
Peak Temperature (T_p)	$T_c + 0^\circ\text{C}/-5^\circ\text{C}$	$T_c + 0^\circ\text{C}/-5^\circ\text{C}$
Time Within 5°C of Actual Peak Temperature (t_p)	10 sec to 30 sec	20 sec to 40 sec
Ramp-Down Rate (T_p to T_L)	6°C/sec max	6°C/sec max
Time 25°C to Peak Temperature	6 minutes max	8 minutes max

¹ Based on IPC/JEDEC J-STD-020D.01 for SnPb and Pb-free processes. Package volume < 350 mm³, package thickness > 2.5 mm.

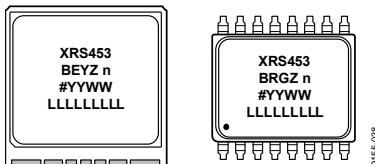
PACKAGE MARKING CODES

Figure 36. LCC_V and SOIC_CAV Package Marking Codes

Table 19. Package Code Designations

Marking	Meaning
XRS	Angular rate sensor
453	Series number
B	Temperature grade (-40°C to +105°C)
RG	Package designator (SOIC_CAV package)
EY	Package designator (LCC_V package)
Z	RoHS compliant
n	Revision number
#	Pb-free designation
YYWW	Assembly date code
LLLLLLLL	Assembly lot code (up to nine characters)

OUTLINE DIMENSIONS

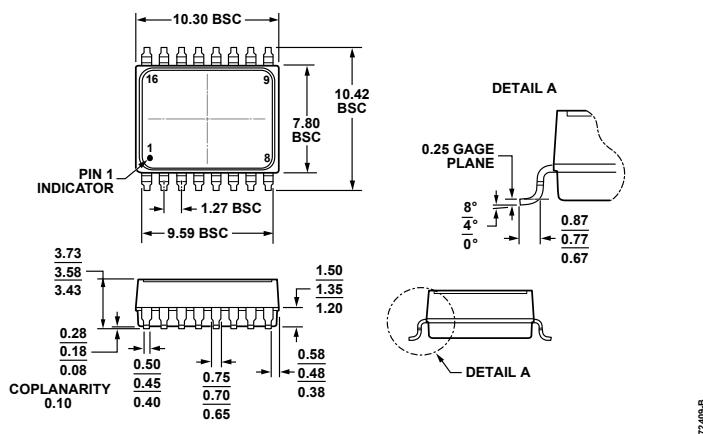


Figure 37. 16-Lead Small Outline, Plastic Cavity Package [SOIC_CAV]
(RG-16-1)
Dimensions shown in millimeters

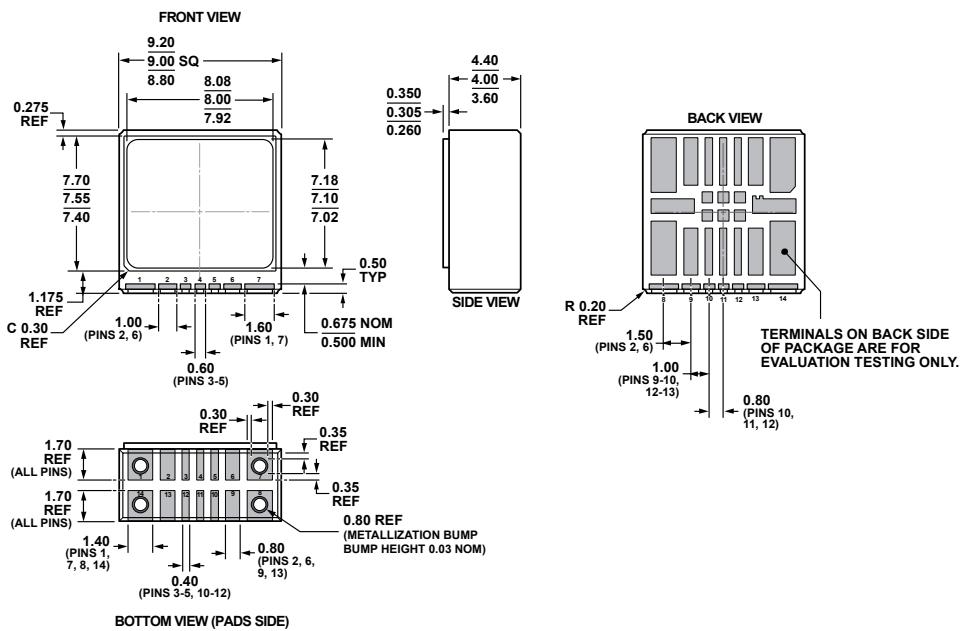


Figure 38. 14-Terminal Ceramic Leadless Chip Carrier, Vertical Form [LCC_V]
(EY-14-1)
Dimensions shown in millimeters

04-08-2010-A

Data Sheet

ADXRS453

ORDERING GUIDE

Model ¹	Temperature Range	Package Description	Package Option
ADXRS453BEYZ	–40°C to +105°C	14-Terminal Ceramic Leadless Chip Carrier, Vertical Form [LCC_V]	EY-14-1
ADXRS453BEYZ-RL	–40°C to +105°C	14-Terminal Ceramic Leadless Chip Carrier, Vertical Form [LCC_V]	EY-14-1
ADXRS453BRGZ	–40°C to +105°C	16-Lead Small Outline, Plastic Cavity Package [SOIC_CAV]	RG-16-1
ADXRS453BRGZ-RL	–40°C to +105°C	16-Lead Small Outline, Plastic Cavity Package [SOIC_CAV] Evaluation Board, SOIC_CAV	RG-16-1
EVAL-ADXRS453Z		Evaluation Board, LCC_V	
EVAL-ADXRS453Z-V		Analog Devices Inertial Sensor Evaluation System (Includes ADXRS453 Satellite)	
EVAL-ADXRS453Z-M		ADXRS453 Satellite, Standalone, to be used with Inertial Sensor Evaluation System	
EVAL-ADXRS453Z-S			

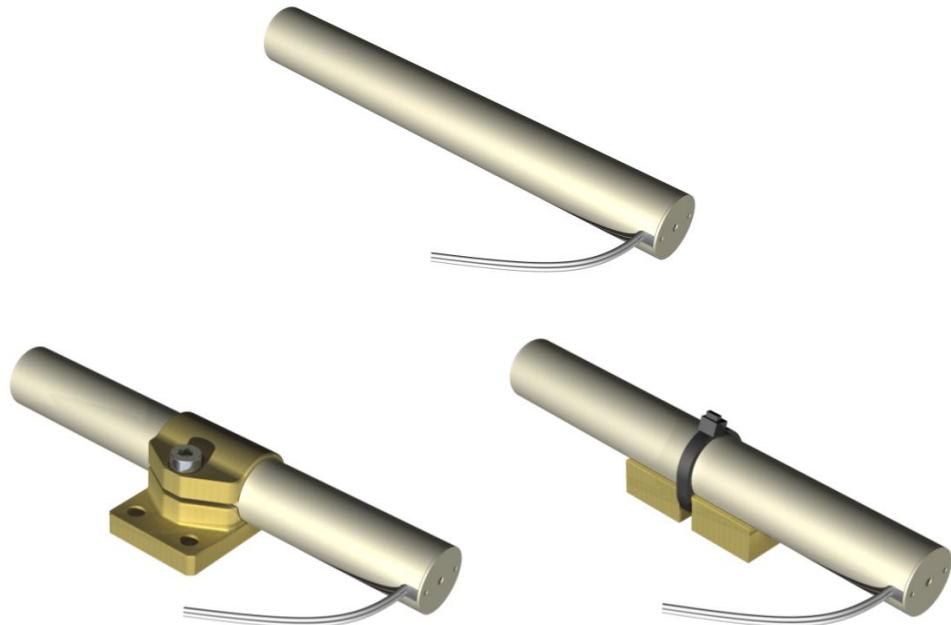
¹ Z = RoHS Compliant Part.

Appendix M - Magnetic Torquer Data Sheet

The data sheet for the magnetic torquer appears on the next page.



MT0.5-1 Technical Performance Data Sheet



Performance Data	MT0.5-1-01	Tolerance, remark
Maximum linear dipole range	$> \pm 0.5 \text{ Am}^2$	at max. linear current
Linear dipole current range	$\pm 60 \text{ mA}$	at maximum linear dipole moment
Linear dipole power consumption	0.3 W	$\pm 5\% (20.0 \text{ }^\circ\text{C})$
Linear dipole supply voltage range	$\pm 5 \text{ V DC}$	$\pm 5\% (20.0 \text{ }^\circ\text{C})$
Linearity error	< 1 %	within linear dipole moment range
Residual dipole	< 0.5 %	of maximum linear dipole moment
Coil resistance	83 Ω	$\pm 5\% (20.0 \text{ }^\circ\text{C})$
Magnetic dipole step response time constant τ (63% value)	10 ms	$\pm 20\% (20.0 \text{ }^\circ\text{C})$
Connector	Flying Leads	
Overall unit mass	0.05 kg	$\pm 5\%$
Unit Length	$\leq 94 \text{ mm}$	overall unit length
Unit Height / Width	16 mm / 33 mm	with heritage bracket
Unit Height / Width	20 mm / 12 mm	with V-bracket

Appendix N - GPS Data Sheet

The data sheet for the GPS appears on the next page.

❖ SGR-05U

Positioning and timing information can be processed to obtain orbital information. Surrey can provide expertise on orbit determination solutions using the Surrey Space GPS Receiver.

Applications

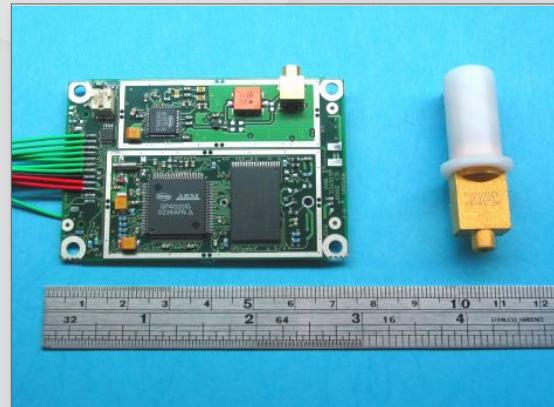
- ❖ Navigation for LEO missions
- ❖ Position, velocity, time determination
- ❖ Post-maneuver orbit determination
- ❖ Payload data time stamping
- ❖ Accurate timing and synchronization
- ❖ Suitable for cubesat and nanosatellite applications

Features

- ❖ 12 channel L1 C/A code space GPS receiver
- ❖ Low-cost commercial based unit
- ❖ OEM PCB for integration in host module
- ❖ Low power and mass
- ❖ Radiation tolerant design
- ❖ Includes active quadrifilar antenna (while supplies last)

Interfaces

- ❖ Power and control
- ❖ TM/TC serial data
- ❖ 50 ohm antenna (MCX)
- ❖ Pulse-per-second



Typical Performance

- ❖ Position to 10 m (95%)
- ❖ Velocity to 15 cm/s (95%)
- ❖ Synchronization to UTC 500 ns
- ❖ Typical time-to-first-fix (warm) 90 s
- ❖ Typical time-to-first-fix (cold) 180 s
- ❖ 5 V supply, 0.8 W
- ❖ 70 x 45 x 10 mm

Option

- ❖ Interface module for ground testing

Heritage

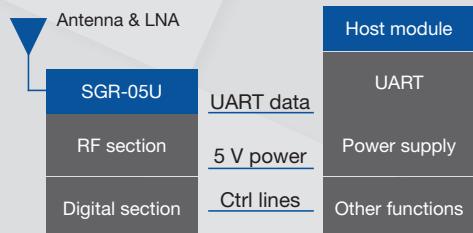
- ❖ 16 receiver units delivered to-date
- ❖ Flown on 5 satellites

❖ SGR-05U

Subsystem

- ❖ Radiation: Core components tested to TID greater than 10 kRads(Si)
- ❖ Antenna: Active quadrifilar antenna weighing 12 g with 13 x13 x 40 mm dimensions
- ❖ Performance: Based on circular polar low-Earth orbit with typical ionospheric and ephemeris error levels on signals

Typical Use



Specifications

	Typical (95%)	Max (95%)
Orbital Position (3D)	10 m	20 m
Orbital Velocity (3D)	0.15 m/s	0.25 m/s
Time	0.5 µs	1 µs
Time to First Fix (mean)	90 s	180s
Mass	40 g	
Dimensions	70 x 45 x 10 mm	
Power	0.8 W at 5 V	
Temperature	-20°C to +50°C (operating)	
Random Vibration	15 g _{rms} in all axes	
Radiation Tolerance	>5 kRad (Si)	

Product specification subject to change without notification

Typical Measurement Precision

Pseudo Range	0.9 m
Carrier-Smoothed Range	0.15 m
Carrier Phase Noise	2 mm
Doppler Velocity	0.5 m/s
Carrier Range Rate Velocity	0.03 m/s

The small satellite revolution started 30 years ago with Surrey Satellite Technology—the world's premier provider of operational and commercial satellite programs with over 40 satellites launched successfully and 240 years of on-orbit experience gained.

From its Englewood, Colorado, facilities, Surrey supplies complete in-house design, manufacture, launch, and operation of small satellites, to include remote sensing, navigation, and communications payloads, avionics suites and subsystems, ground infrastructure, and training and consulting services.

Surrey Satellite Technology US LLC 345 Inverness Drive South, Suite 100 | Englewood, CO 80112 | 303.790.0653 | www.surreysatellite.com

Surrey US is AS 9100:2009-certified | GSA Contract Holder | Shop Surrey online at www.surreysatellite.com/shop

December 2014