3-26-2015

Design and Test of an Attitude Determination and Control System for a 6U CubeSat using AFIT's CubeSat Testbed

Michael L. Tibbs

Recommended Citation

Tibbs, Michael L., "Design and Test of an Attitude Determination and Control System for a 6U CubeSat using AFIT's CubeSat Testbed" (2015). *Theses and Dissertations*. 185.
https://scholar.afit.edu/etd/185

DESIGN AND TEST OF AN ATTITUDE DETERMINATION AND CONTROL SYSTEM FOR A 6U CUBESAT USING AFIT'S CUBESAT TESTBED

THESIS

Michael L. Tibbs, 2nd Lieutenant, USAF

AFIT-ENY-MS-15-M-240

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-MS-15-M-240

DESIGN AND TEST OF AN ATTITUDE DETERMINATION AND CONTROL SYSTEM FOR A 6U CUBESAT USING AFIT'S CUBESAT TESTBED

THESIS

Presented to the Faculty
Department of Aeronautics and Astronautics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Astronautical Engineering

Michael L. Tibbs, B.S.A.E.
2nd Lieutenant, USAF

March 2015

AFIT-ENY-MS-15-M-240

DESIGN AND TEST OF AN ATTITUDE DETERMINATION AND CONTROL SYSTEM FOR A 6U CUBESAT USING AFIT'S CUBESAT TESTBED

THESIS

Michael L. Tibbs, B.S.A.E.
2nd Lieutenant, USAF

Committee Membership:

Eric D. Swenson, PhD
Chair

Richard G. Cobb, PhD
Member

Carl R. Hartsfield, PhD
Member

AFIT-ENY-MS-15-M-240

Abstract

The design and test of a 6U CubeSat Attitude Determination and Control System (ADCS) are explored to establish single-axis control using AFIT's CubeSat testbed consisting of a Helmholtz cage and hemi-spherical air bearing. The Helmholtz cage produces a near-uniform magnetic field inside the cage while the air bearing provides a near-frictionless surface for ADCS testing. The ADCS testbed includes a four wheel pyramid reaction wheel array (RWA) for actuation and an inertial measurement unit (IMU) for attitude determination. Along with the ADCS hardware, the ADCS testbed also includes an Electrical Power System (EPS) and Command and Data Handling (CDH) for power and wireless telemetry, respectively. Attitude estimation is performed using the QUEST algorithm with magnetometer and accelerometer sensor data to estimate a current quaternion. A Proportional-Integral-Derivative (PID) controller is used for control of the ADCS testbed while each reaction wheel motor is controlled by a proportional gain. After calibration, the static estimation accuracy improved from $\pm 3^o$ to $\pm 0.02^o$ (3σ). The RWA is characterized as a 1st order system with a systemic 0.2 second time delay. The ADCS testbed controller demonstrates linear time response characteristics at small angle slews. Through this research effort, the pointing accuracy was improved from $\pm 20^o$ to $\pm 0.07^o$ (3σ). External torques caused by air currents and inherent magnetic moments were explored and characterized. The inherent magnetic moment of the ADCS testbed was calculated. Simulated time response characteristics of the model showed similar behavior to the ADCS testbed results.

AFIT-ENY-MS-15-M-240

To my wife, this research effort would not have been possible without your ceaseless motivation and support. Although we lived far apart from each other, the precious times we had together recharged and inspired me to stay diligent. And to my parents, who never forgot to call and to constantly encourage me to do my best.

Acknowledgements

I would first like to thank my research advisor, Dr. Eric Swenson, for not only the invaluable academic and research guidance, but also for the lessons in officership that will not be forgotten. Thank you for also giving me constant feedback on my research and thesis document that always pointed me in the right direction.

I would also like to thank my thesis committee members, Dr. Richard Cobb and Dr. Carl Harstfield. Dr. Cobb, thank you for teaching me the fundamentals of control theory and signal processing, and always being available to answer my questions about modeling systems and other various topics. Dr. Hartsfield, although you were only at AFIT for the summer design course, thank you for helping me get started with working on the ADCS and reaction wheels.

Next I would like to thank Matt Lippert for helping me with the ADCS software and C code algorithms. I will always appreciate your patience through the many months of me hunting you down and asking you for help. I could not have accomplished the work described in this thesis without your mentorship. I also want to thank the whole mechanical team, especially Philip Smith for his mechanical knowledge and Randall Sharp for the hand crank and help with the ADCS testbed assembly.

Lastly, I would like to thank my fellow classmates that have endured this journey with me. Through all the late nights and weekends spent here at AFIT working on our research, I couldn't have asked for a better bunch to keep the mood bright and the motivation to work hard.

Michael L. Tibbs

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol		Page
R^{bi}	Inertial to Body Frame Rotation Matrix	12
$\vec{v}_i$	3x1 Vector in Inertial Frame	12
$\vec{v}_b$	3x1 Vector in Body Frame	12
R^{bs}	Sensor to Body Frame Rotation Matrix	12
R^{si}	Inertial to Sensor Frame Rotation Matrix	12
$\vec{\omega}^{bi}$	Angular Velocity of Body Frame with Respect to the Inertial Frame	14
$\dot{\vec{\theta}}$	3x1 Euler Angle Rate Vector	14
θ	3x1 Euler Angle Array	14
$\hat{a}$	Euler Axis	15
ϕ	Euler Angle	15
a^x	Skew Symmetric Matrix	15
$\bar{q}$	4x1 Quaternion Vector	16
$\dot{q}$	4x1 Quaternion Rate Vector	17
$\vec{H}$	3x1 Angular Momentum Vector	18
$\mathbf{I}$	3x3 Moment of Inertia Tensor	18
$\vec{M}$	3x1 External Torque Vector	19
$\dot{\vec{H}}$	3x1 Angular Momentum Rate	19
$\dot{\vec{\omega}}^{bi}$	Body Frame Angular Acceleration with Respect To Inertial Frame	19
$\vec{H}_{net}$	3x1 Total Net Angular Momentum Vector	20
$\vec{h}_{body}$	3x1 Object Angular Momentum Vector	20
$\vec{h}_{act}$	3x1 Internal Actuator Angular Momentum Vector	20

$\mathbf{D}$	Reaction Wheel Moment of Inertia	20
$\vec{\psi}$	3x1 Reaction Wheel Angular Velocity Vector	20
$\dot{\vec{\psi}}$	3x1 Reaction Wheel Angular Acceleration Vector	21
I	Identity Matrix	29
p	Rodriguez Parameters	30
$\bar{q}_e$	4x1 Quaternion Error Vector	31
$\bar{q}_c$	4x1 Commanded Quaternion Vector	31
$\bar{q}_p$	Current Quaternion	31
$\tilde{\mathbf{M}}(\bar{q}_c)$	4x4 Transmuted Quaternion Matrix	31
K_p	Proportional Gain	35
K_i	Integral Gain	36
K_d	Derivative Gain	37
$\mathbf{C}$	Scalar MOI in z-axis	87
m_x	ADCS Testbed x-axis Magnetic Moment	87
m_y	ADCS Testbed y-axis Magnetic Moment	87

List of Abbreviations

Abbreviation		Page
ADCS	Attitude Determination and Control System	1
P-POD	Poly-PicoSatellite Orbital Deployer	2
DoD	Department of Defense	2
USAFA	United States Air Force Academy	2
SSRC	Space Systems Research Center	2
AFIT	Air Force Institute of Technology	2
CSRA	Center for Space Research and Assurance	2
COTS	Commercial Off-The-Shelf	3
RWA	Reaction Wheel Array	4
EPS	Electrical Power System	4
CDH	Command Data Handling	4
IMU	Inertial Measurement Unit	6
EOM	Equations Of Motion	9
QUEST	Quaternion Estimator	9
ECI	Earth-Centered Inertial	10
CAD	Computer-Aided Design	11
COM	Center of Mass	11
DCM	Direction Cosine Matrix	12
LEO	low earth orbit	24
PID	Proportional-Integral-Derivative	31
MIMO	Multiple-Input-Multiple-Ouput	33
HIL	Hardware in the Loop	38

NPS	Naval Postgraduate School	38
AFRL	Air Force Research Laboratory	38
CubeTAS	CubeSat Three-Axis Simulator	39
MIT	Massachusetts Institute of Technology	40
ITU	Istanbul Technical University	41
SIL	Software in the Loop	41
UA	University of Arkansas	41
CPSU	California Polytechnic State University	42
STK	Satellite Took Kit	43
WMM	World Magnetic Model	43
SEET	Space Environment and Effects Tool	49
COR	Center of Rotation	51
DAC	Digital to Analog Converter	52
RAM	Random Access Memory	57

DESIGN AND TEST OF AN ATTITUDE DETERMINATION AND CONTROL
SYSTEM FOR A 6U CUBESAT USING AFIT'S CUBESAT TESTBED

I. Introduction

1.1 Background

The term "CubeSat" is defined as a satellite that is 10 x 10 x 10 cm^3 with a mass at or below 1.33 kg [53]. The CubeSat was developed to provide hands-on practice with the design, test, and mission planning of satellites. University programs generally accepted the CubeSat concept due to the expense of designing and building larger satellites. Since 1999, the number of CubeSats being developed and launched into space has rapidly increased [36]. Most CubeSats are from universities, but larger companies such as Boeing and smaller companies such as Clyde Space have also constructed them. Satellite companies are working toward generic "nanosatellite" operations, a category CubeSats fit in.

The CubeSat has many advantages over larger satellites for groups with lower budgets and tighter time constraints. Some of these advantages include the following: lower design, test, and launch costs; reduced design time; inexpensive test facilities; transportation ease; and cost effectiveness [53]. Disadvantages include the following: higher mission risks, typically due to lack of redundancy and testing; lower performance components resulting in poor attitude estimation and pointing capabilities; reduced payload capability due to poor pointing accuracy and limited volume; and the underdevelopment of CubeSat attitude determination and control system (ADCS) testing facilities.

Advances in small satellite technology and other innovative ideas have reduced the gap between size and performance [48]. One of these major ideas was to increase the size of the CubeSat by adding more "U's." The widely used CubeSat orbital deployer, the Poly-PicoSatellite Orbital Deployer (P-POD), can accommodate CubeSats up to 3U in size. A larger CubeSat keeps the original idea of cost effectiveness while increasing payload size, power output, and control authority [53].

Due to the rising popularity and quick advancement of CubeSats, the CubeSat concept's expansion meets the needs of Department of Defense (DoD) related projects that would be more cost effective than using larger satellites. The United States Air Force Academy's (USAFA) Space Systems Research Center (SSRC) and the Air Force Institute of Technology's (AFIT) Center for Space Research and Assurance (CSRA) are currently in development of CubeSats. The SSRC is developing a 3U CubeSat called FalconSat-7 while the CSRA is designing and testing 6 and 12U CubeSats. Figure 1 shows AFIT's and Pumpkin's collaborative 6U design, which demonstrates the large payload volume inherent alongside the small bus and ADCS space taken within the 6U chassis.

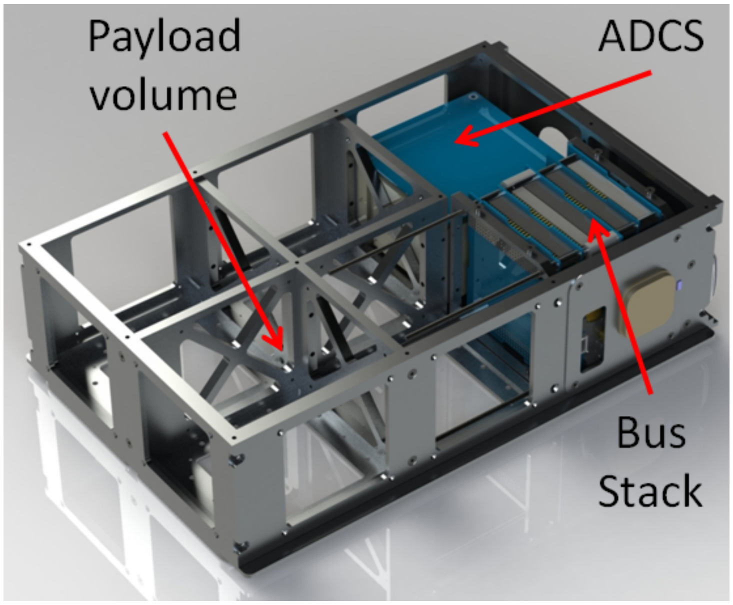


Figure 1. CAD Image of AFIT's and Pumpkin's Collaborative 6U CubeSat Design [37]

Very little research about the concept of expanding CubeSats out to sizes of 6 and 12U has occurred, but the future is promising due to the increased payload capacity inherent in the expanded design [37]. Instead of buying commercial off-the-shelf (COTS) components, AFIT personnel design and build most of their 6U CubeSat components in-house. This has given rise to the need to test and validate these designs in a space-simulated environment.

1.1.1 AFIT's CubeSat Testbed

As mentioned earlier, CubeSat test facilities tend to lack the maturity and technical capabilities typically found in larger space programs. Since AFIT's 6U components are custom built, they lack flight heritage and therefore lack the resume' to be sent into space without increased risk. As will be discussed in Section 2.6, the concept of a satellite simulator is starting to infiltrate the CubeSat domain enabling CubeSat software, hardware, and ADCS algorithms on-the-ground testing. AFIT's CubeSat testbed, which currently consists of a Helmholtz cage and hemi-spherical air bearing, is shown in Fig. 2.

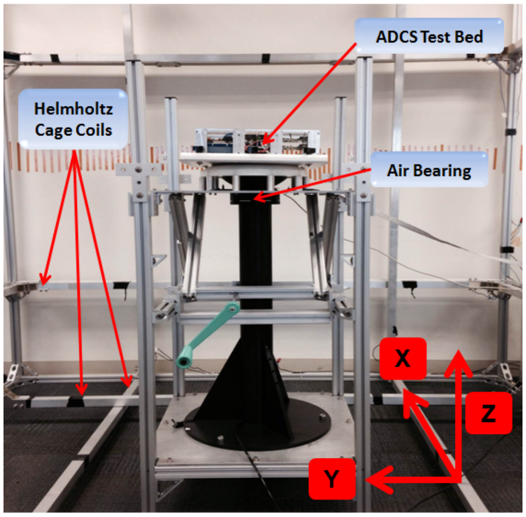


Figure 2. AFIT's CubeSat Testbed

Note the coordinate system at the bottom right of Fig. 2. The z-axis points up to the ceiling, the x-axis points at the wall, and the y-axis completes the orthogonal set. The air bearing provides a near frictionless surface for the ADCS testbed to rotate; the air bearing allows 360^o of rotation about the z-axis and $\pm 60^o$ of rotation about the x-and y-axes for ADCS testing. The Helmholtz cage produces a nearly uniform magnetic field inside the cage and is primarily used for calibrating and filtering magnetometer measurements inside the field.

1.1.2 ADCS Testbed

The 6U ADCS testbed as shown in Fig. 3 includes an ADCS board along with a motor controller board for a reaction wheel array (RWA), an Electrical Power System (EPS) board, battery pack, a Command Data Handling (CDH) board with WiFly capability, and a laser pointer to track the truth angle on the wall.

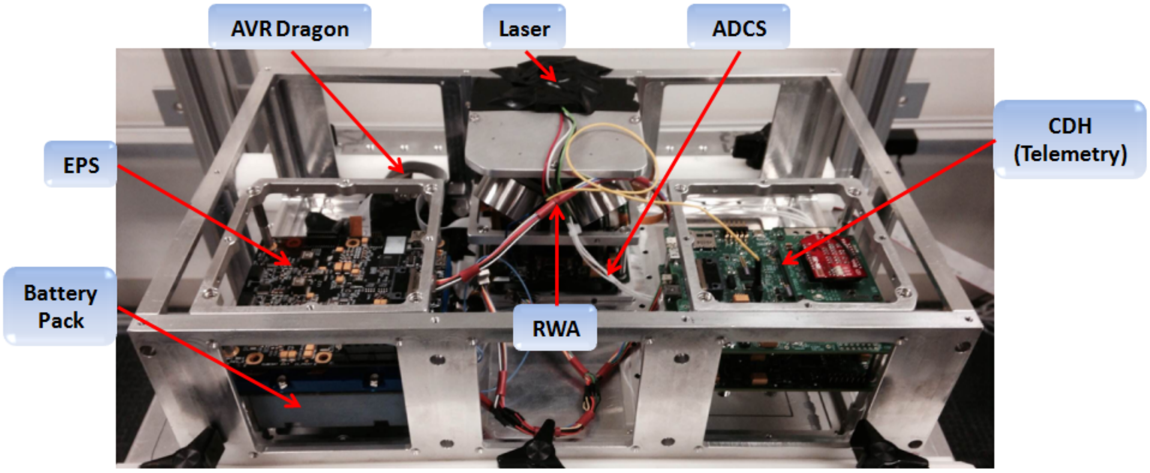


Figure 3. 6U ADCS Testbed

AFIT's CubeSat testbed is intended for use in testing the performance capabilities of the ADCS testbed. Although there is no payload, one could be included. The ADCS testbed is designed to be able to test any CubeSat ADCS and corresponding payload 3U or larger by sliding an ADCS testbed into the CubeSat testbed to validate simulation and enhance flight readiness. A full description of the hardware, software, and algorithm setup can be found in Chapter III.

1.2 Problem Statement

The inspiration for AFIT's CubeSat testbed is to test a CubeSat ADCS to validate hardware, software, and algorithm performance. By creating an ADCS and bus package for a 6 or 12U CubeSat that has high pointing accuracy capabilities and robust ADCS algorithms, payload designs that require accurate pointing can be studied and validated. AFIT's CubeSat testbed will be used to validate on the ground AFIT's goal of the design and test of an ADCS testbed that can handle a wide variety of payload requirements. The achievement of this goal will not only help spark the future development of 6 and 12U sized payloads for DoD related experiments and increase cost effectiveness of DoD related missions, but it will also support the testing methodologies of using CubeSat testbeds for validation of flight hardware, software, and algorithms.

1.3 Research Focus

The realization of the goals outlined in Section 1.2 would likely require years of design and testing. The following list details the focus of this research effort and further illustrates the tasks that should be accomplished to advance the design and testing of the ADCS testbed with the use of AFIT's CubeSat testbed at this stage in development:

1. Establish wireless communication with the ADCS testbed so that real time data can be streamed

2. Achieve a high level of attitude determination using an IMU

3. Establish control about the z-axis shown in Fig. 2

4. Characterize external torques in AFIT's CubeSat testbed

5. Create a model to accurately simulate ADCS behavior

1.4 Methodology

The goal of this research effort is to investigate the level of the ADCS's z-axis control authority using only a RWA for actuation and an inertial measurement unit (IMU) for determination by using AFIT's CubeSat testbed. Furthermore, utilizing data gathered by these tests will help determine an analytical model's simulation accuracy. In order to achieve the above stated goals, the following research and development efforts are required to improve the testbed capabilities:

- Calibrate the on board IMU and install filters to smooth noisy sensor data (Sections 4.1-4.3)
- Characterize the reaction wheel motor's behavior (Sections 4.3-4.4)
- Design and test the controller for the reaction wheel motors (Section 4.5)
- Characterize and properly model sensor noise (Section 4.6)
- Design and test the controller for the ADCS (Section 4.7)
- Demonstrate the external torques inherent in AFIT's CubeSat testbed (Section 4.8)
- Implement models of the ADCS and RWA supported by sensor data for simulation comparison (Section 4.9)

1.5 Research Effort, Scope, and Contribution

To successfully perform all activities stated in Section 1.4, knowledge of astronautical engineering, computer science, physics, electrical engineering, mechanical engineering, controls engineering, signal processing, existing hardware, existing software, and existing algorithms are required. The outcomes of this research includes CubeSat testbed characterization and significantly improved CubeSat ADCS testbed hardware, control software, and algorithms.

This contribution documents the first time, to the best of this author's knowledge, a 6U ADCS testbed's validation and achievement of single-axis control inside a Helmholtz cage atop an air bearing. Furthermore, this is the first time a CubeSat testbed's external torques, such as air current and magnetic, have been tested and characterized. It is this author's hope that this contribution to ADCS testing inside CubeSat testbeds will lead to the achievement of the goals described in Section 1.2.

1.6 Preview

Chapter I provided the background, motivation, research focus, and methodology along with an outline of the research effort's scope and contribution. Chapter II details the key background knowledge needed to achieve the goal of this research effort. This includes mathematical derivations of dynamics, establishment of notation used throughout the thesis, overviews of sensors and actuators used in spacecraft, signal processing, control design, and a detailed literature review outlining related research. Chapter III describes the methodology and testing procedures for this research effort. This includes descriptions of AFIT's CubeSat testbed, configuration of the ADCS testbed, and details the various testing procedures and setups applied during this research effort. Chapter IV details the results and analysis of the testing procedures described in Chapter III. This includes all experimental tests as well as ADCS testbed data compared to model simulation. Finally, Chapter V summarizes the results of this research effort and offers recommendations for future work in order to take the next step in realizing the goals set forth in Section 1.2.

II. Background

Chapter II provides the background concerning attitude determination and control. This discussion covers coordinate frames, kinematic and kinetic equations of motion (EOM) and how actuators are modeled. This chapter will also discuss the quaternion estimator (QUEST) algorithm's purpose and usage, varying sensors and filters used for attitude determination, and control techniques that will be implemented in typical CubeSat attitude control subsystems. Finally, related research on the topic of ADCS with CubeSat testbeds will be discussed.

2.1 Attitude Dynamics

The purpose of any ADCS is to determine and control the orientation of a spacecraft with respect to a desired orientation relative to another object. Typical examples include the Hubble Space Telescope ADCS which points to a region of the universe to take high resolution photographs [12]. Other examples include a reconnaissance satellite ADCS that tracks a specific target on Earth, or for orienting a spacecraft to correctly point solar panels to receive optimal sunlight. To describe an ADCS, coordinate frames and the derivation of the kinematic and kinetic EOM are needed.

2.1.1 Coordinate Frames

When dealing with dynamic systems, such as spacecraft, an intuitive way to describe their motion is with the use of coordinate frames otherwise known as reference frames. The kinematic and kinetic EOM must be derived with respect to an inertial coordinate frame in terms of the spacecraft's body reference coordinate frame. All calculations and analyses are performed in the satellite's body frame since the satellite's moment of inertia (

2.1.1.1 Earth-Centered Inertial

The Earth-Centered Inertial (ECI) reference frame has its origin at the center of the Earth and is assumed inertial. The x-axis is fixed in the direction of the vernal equinox while the z-axis is in the direction of Earth's spin axis. The y-axis completes the right-handed orthogonal set [1]. Although the ECI frame is used to derive the EOM, this research effort will use a varied version of the ECI frame that is defined based on the ADCS testbed as shown in Fig. 4.

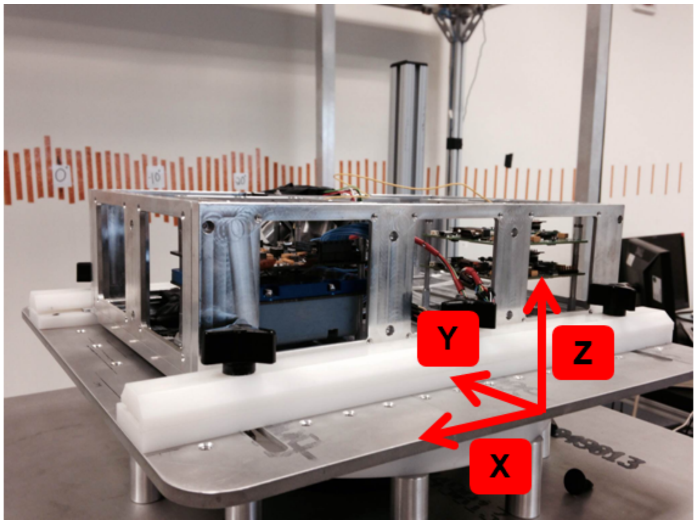


Figure 4. ADCS Testbed Inertial Reference Frame

One of the main focus points of this research is to control the ADCS testbed about the z-axis as shown in Fig. 4 inside AFIT's CubeSat testbed. The z-axis points towards the ceiling, the y-axis points towards the wall, and the x-axis completes the orthogonal set. Note that the x-axis and y-axis between the CubeSat testbed (Fig. 2) and the ADCS testbed (Fig. 4) are switched. This is due to the difference in how the IMU and Helmholtz cage coordinate frames are defined. A detailed explanation on AFIT's CubeSat testbed and the ADCS testbed are are found in sections 3.1-3.5.

2.1.1.2 Spacecraft-Fixed Frame

The spacecraft-fixed frame, or body frame, of any object is defined by the designer. In satellite applications, the computer-aided design (CAD) model is normally used in defining the body frame. Typically, the body-frame's origin is located at the center of mass (COM) of the object with the axes pointing in the direction of the principal axes [45]. However, this is only a recommended set; many satellite body axes will not be aligned with the principle axes. It may be easier computationally to align the body axes with existing sensor hardware such as an IMU, which has its own previously defined coordinate frame. As long as the body frame has mutually right-handed orthogonal unit vectors, the frame may be placed to the designer's needs. When the ADCS testbed body frame is coincident with the inertial frame defined in Fig. 4, the angular positions in the x-, y-, and z-axes are zero.

2.1.1.3 Sensor and Actuator Frames

Important to attitude determination and control are the coordinate frames from sensors and actuators. Depending on hardware specifications and placement within the satellite, a sensor or actuator frame must be defined. Once the frame is defined, it may be converted to the body frame via a rotation matrix (Section 2.1.2). Clearly defining the sensor and actuator frames is important to the accuracy of the spacecraft's attitude knowledge. If performed incorrectly, attitude knowledge may be inaccurate or control torques sent to actuators will be incorrect. Many spacecraft designers align actuators and sensors with the spacecraft body frame. Examples include a RWA, torque coils, or an IMU aligned with the designated body frame.

2.1.2 Kinematics

To begin the process of deriving the EOM, the relationship between satellite orientation and angular velocity, also known as kinematics, must be mathematically defined [12]. The first step is being able to define the orientation of objects in each of the many reference frames and be able to convert between them. The relationship used to convert vectors between reference frames is known as a rotation matrix. The rotation matrix R^{bi} will take a vector in the inertial reference frame $\vec{v}_i$ and convert it to the body reference frame $\vec{v}_b$ as shown in

$$\vec{v}_b = R^{bi}\vec{v}_i \tag{1}$$

where $\vec{v}_i$ and $\vec{v}_b$ are 3x1 vectors making R^{bi} a 3x3 matrix. The rotation matrix, also known as a direction cosine matrix (DCM), has special properties such as being orthonormal. As such, the primary property is that the DCM's inverse is equal to its transpose. Furthermore, rotation matrices can be multiplied together to relate more than two reference frames. For example,

$$R^{bi} = R^{bs}R^{si} \tag{2}$$

where R^{bs} is a rotation from the sensor frame to the body frame and R^{si} is a rotation matrix from the inertial frame to the sensor frame. Rotation matrices are fundamental to the EOM and are a primary means of how many interpretations of the kinematic equations are developed. The following sections will describe various ways to represent the kinematic EOM as well as their limitations.

2.1.2.1 Euler Angles

Leonhard Euler, an 18th century mathematician, showed that the minimum number of independent parameters to fully describe any rotation is three [41]. These three necessary rotations are an intuitive way to describe attitude. Normally, the rotations are performed based upon designer needs due to the limitations of Euler angles. Figure 5 illustrates a 3-2-1 rotation sequence. This sequence first rotates an angle θ_2 about the $\hat{i}_3$ axis. In a new intermediate frame $\Phi_{i'}$, the next rotation angle of θ_3 occurs about the $\Phi_{i2'}$ axis. In a second intermediate frame $\Phi_{i''}$, a 3rd rotation angle of θ_1 is performed about the $\Phi_{i1''}$ axis. This results in the $\hat{i}$ coordinate frame being transformed into the Φ_b coordinate frame.

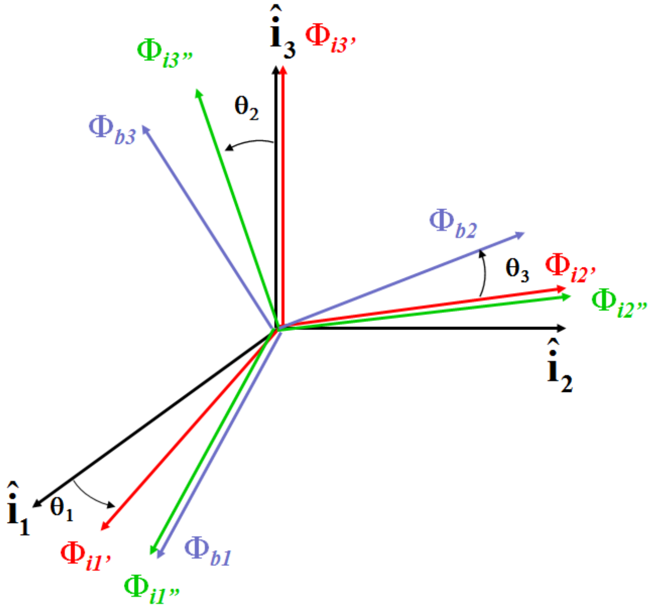


Figure 5. 3-2-1 Euler Angle Rotation Sequence [41]

Every rotation matrix will have an inherent singularity depending on the rotation performed. For symmetric rotations, such as a 3-1-3 or 1-2-1, the singularity will occur when θ_2 (defined in Fig. 5) is either 0^o or 180^o. Asymmetric rotations, such as 3-2-1 or 2-1-3, have singularities that occur when θ_2 is either 90^o or 270^o. In many attitude control cases, linearizations of the EOM are performed about 0^o so robust linear controllers can be used. This means that recovering Euler angles from a DCM is not viable for symmetric rotations. Therefore, asymmetric rotations are used to avoid singularities when recovering Euler angles in this linearized case.

To derive the kinematic EOM which relates attitude and angular velocities, we must relate the angular velocity measured in the body frame to the inertial frame. The angular velocity, however, cannot be directly integrated to compute the attitude due to its path dependent nature. Furthermore, rotations cannot be expressed as vectors, but the rotation rate can [38]. Using a 3-2-1 rotation sequence, rotations are made on each axis along a different reference frame in accordance to that particular angular velocity. Starting in the inertial frame and adding up angular velocities that sum to a vector in the body frame results in

$$\vec{\omega}^{bi} = \vec{\omega}^{bi''} + \vec{\omega}^{i''i'} + \vec{\omega}^{i'i} \tag{3}$$

where $\vec{\omega}^{bi''}$, $\vec{\omega}^{i''i'}$ and $\vec{\omega}^{i'i}$ are the angular velocity components expressed in intermediate frames [38]. These angular velocities must be in the same reference frame to be summed. By using rotation matrices to find the angular velocity of the b-frame with respect to the i-frame $\vec{\omega}^{bi}$ for each intermediate frame, we now have the angular velocity in terms of the Euler angle rates. The kinematic equations expressed as

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sin(\theta_3)}{\cos(\theta_2)} & \frac{\cos(\theta_3)}{\cos(\theta_2)} \\ 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 1 & \frac{\sin(\theta_3)\sin(\theta_2)}{\cos(\theta_2)} & \frac{\cos(\theta_3)\sin(\theta_2)}{\cos(\theta_2)} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{4}$$

relate angular velocity in the body frame to the Euler angle rates $\dot{\vec{\theta}}$. Notice that when θ_2 is 90^o or 270^o a singularity occurs. This is the major drawback to implementing Euler angles θ into satellite hardware. To achieve a robust control algorithm, a kinematic representation without the existence of a singularity is desired.

2.1.2.2 Eigenaxis

The motion of a rigid body can be described as a rotation about a fixed axis. Any two reference frames can be related by a specific rotation axis defined by a unit vector $\hat{a}$ known as the Euler axis and an angle ϕ known as the Euler angle [41]. Figure 6 shows the 3-2-1 frame rotating to the 3'-2'-1' frame using the Eigenaxis rotation

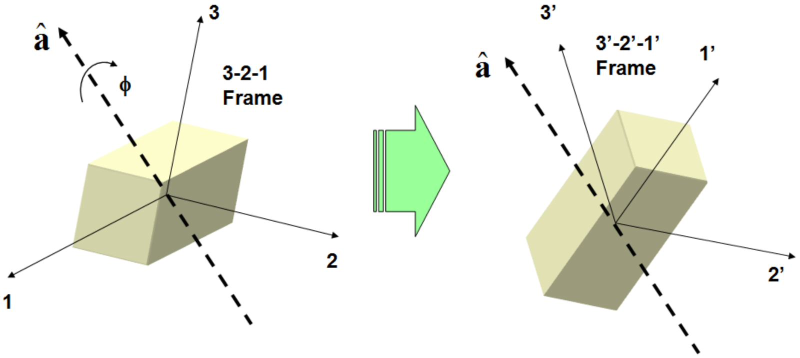


Figure 6. Eigenaxis Rotation [41]

Once $\hat{a}$ is calculated, it is rotated an angle ϕ to rotate the 3-2-1 frame into the 3'-2'-1' frame. ϕ and $\hat{a}$ can be computed from a rotation matrix that results in

$$\phi = \cos(\frac{1}{2}(\text{trace}\, R - 1))^{-1} \tag{5}$$

$$a^x = \frac{1}{2\sin\phi}(R^T - R) \tag{6}$$

where a singularity occurs when ϕ is 0^o or 180^o [38]. Note that a^x denotes a skew symmetric matrix. Although Eigenaxis rotations are another intuitive means of representing attitude, the equations still hold a singularity. Eigenaxis concepts are important, however, because describing the next attitude representation using them will lead to kinematic EOM without singularities.

2.1.2.3 Quaternions

Quaternions, as a form of attitude representation, are used extensively in spacecraft control because the non-existence of a singularity in its representation relieves computational problems. The quaternions can be defined from the Eigenaxis rotation as

$$q_1 = \hat{a}_1 \sin(\frac{\phi}{2}) \tag{7}$$

$$q_2 = \hat{a}_2 \sin(\frac{\phi}{2}) \tag{8}$$

$$q_3 = \hat{a}_3 \sin(\frac{\phi}{2}) \tag{9}$$

$$q_4 = \cos(\frac{\phi}{2}) \tag{10}$$

where $\underline{q}$ contains q_1, q_2, and q_3 while q_4 is a scalar [32]. The complete quaternion, which includes all four values, is denoted as $\bar{q}$. One disadvantage to using quaternions is their lack of intuitiveness. To counteract this, the quaternion output can be converted to a rotation matrix where the Euler angles can be recovered when not at a singularity. Furthermore, computers cannot calculate zero very well, so it is unlikely to be at a singularity in the computed state. A common spacecraft rotation sequence would be the 3-2-1 rotation sequence. A given quaternion can then be converted to a rotation matrix using

$$R = (q_4 - \underline{q}^T \underline{q})I + 2\underline{q}\ \underline{q}^T - 2q_4\underline{q}^x \tag{11}$$

where the Euler angles can be recovered using

$$\theta_1 = \sin^{-1}(R_{12}/\cos\theta_2) \tag{12}$$

$$\theta_2 = \sin^{-1}(-R_{13}) \tag{13}$$

$$\theta_3 = \sin^{-1}(R_{23}/\cos\theta_2) \tag{14}$$

where R is the rotation matrix, and R_{ij} is the corresponding element in R [16]. Note that $\underline{q}^x$ is a skew symmetric matrix. Recovering the Euler angles provides an intuitive means of understanding the actual rotation of the spacecraft. Computer code can be easily implemented to take quaternion outputs from attitude determination algorithms and convert them to Euler angles. This can then be compared with a truth measurement to discover the sensor accuracy on board a spacecraft. Quaternions, like other attitude representations change with time which are related to the angular velocity of the vehicle. The kinematic relationship shown in

$$\dot{\bar{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{15}$$

relates the rate of change of the quaternion $\dot{\bar{q}}$ to the spacecraft instantaneous angular velocity $\vec{\omega}$ [54]. Quaternions have no inherent singularity like other representations and due to their lack of trigonometric functions in the kinematic equations, quaternions are advantageous to real-time computer operations [54]. Although spacecraft attitude may be described continuously without singularities, the primary means of controlling spacecraft are with the manipulation of the angular rates. This requires a discussion about how the application of torque affects the angular rates.

2.1.3 Kinetics

In order to manipulate the spacecraft's angular rates, the relationship between torque, angular velocity, and acceleration must first be explored. The rotational analog to linear momentum, angular momentum $\vec{H}$, is formulated in

$$\vec{H} = \mathbf{I}\,\vec{\omega} \tag{16}$$

where $\mathbf{I}$ is the vehicle's mass moment of inertia and $\vec{\omega}$ is the vehicle's angular velocity [43]. The MOI is a 3x3 second-order tensor that is presented here as a matrix because the body frame has been determined. One major assumption of many satellite attitude dynamics problems is that the spacecraft is a rigid body [47]. Due to the rigid body assumption, the MOI possesses special properties. The MOI of a rigid body will not change unless it is converted to a different reference frame. Most importantly, the components of the MOI matrix are constant when written in the body frame. This is a major reason why the kinetic EOM are normally always derived in the body frame. Typically, the MOI matrix is written in Cartesian coordinates as shown in

$$\vec{H} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{17}$$

where I_{xx}, I_{yy}, and I_{zz} are known as the scalar moments of inertia and the other terms are known as the products of inertia [23]. Refer to Kunz's text[16] for derivations of these terms.

Newton's second law requires that the time rate of change of momentum be computed with respect to an inertial reference frame. The rotational form of this law shown in

$$\vec{M} = \dot{\vec{H}} = \frac{d^{(i)}}{dt}(\vec{H}) \tag{18}$$

defines the relationship between external moments $\vec{M}$ and the rate of change of angular momentum $\dot{\vec{H}}$ in an inertial frame i [12]. Due to the constant nature of the MOI in the body frame, it is preferred that the derivative be taken in the body frame with respect to the inertial frame using the transport theorem which results in Euler's equations as shown in

$$\vec{M} = \mathbf{I}\,\dot{\vec{\omega}}^{bi} + \vec{\omega}^{bi} \times \mathbf{I}\,\vec{\omega}^{bi} \tag{19}$$

where $\dot{\vec{\omega}}^{bi}$ is the angular acceleration of the body frame with respect to the inertial frame [42]. Euler's equations alongside Eq. (15) are the governing EOM for a spacecraft. External moments can be actuator torques and disturbance torques. The next section will present how internal actuators manipulate the spacecraft angular rates and can ultimately control the orientation of a spacecraft.

2.2 Attitude Actuators

The primary duty of the onboard actuators is to slew the satellite at desired rates or hold predetermined orientations. Common spacecraft actuators that apply an external moment to the spacecraft are thrusters and torque coils. Disturbance torques can be attributed to Earth's magnetic field, atmospheric drag, solar radiation pressure, and the gravitational gradient force to name a few [43]. Although actuators that apply an external torque are useful in dispelling external disturbance torques, actuators that apply an internal torque to the spacecraft can be more useful in certain spacecraft applications.

2.2.1 Angular Momentum Exchange

When deriving Euler's equations, it is assumed that the spacecraft is a rigid body. However, when moving actuators such as reaction wheels are placed inside the spacecraft, it clearly becomes non-rigid. In order for Euler's equations to be valid, the angular momentum must be broken up into individual components as shown in

$$\vec{H}_{net} = \vec{h}_{body} + \vec{h}_{act} \tag{20}$$

where $\vec{H}_{net}$ refers to the total sum of angular momentum in the spacecraft while $\vec{h}_{body}$ and $\vec{h}_{act}$ represents the angular momentum of the satellite and actuator, respectively [14]. Note that $\vec{h}_{act}$ is defined at the center of mass of the actuators [20]. When placed inside the spacecraft, the acceleration of a symmetric, rotating body will produce angular torque about its axis of rotation. This torque, however, does not change the net angular momentum of the system because the actuator momentum is internal to the spacecraft [32].

2.2.2 Reaction Wheels

As shown in Fig. 7, reaction wheels typically consist of a flywheel accelerated by an electric motor. A reaction wheel can be mounted anywhere in the spacecraft and its rotational axis aligned in any direction relative to the spacecraft body reference frame [32].

The angular momentum of the wheel $\vec{h}_{rw}$ is given by

$$\vec{h}_{act} = \vec{h}_{rw} = \mathbf{D}\vec{\psi} \tag{21}$$

where the reaction wheel's MOI $\mathbf{D}$ is multiplied by the reaction wheel's angular velocity $\vec{\psi}$.

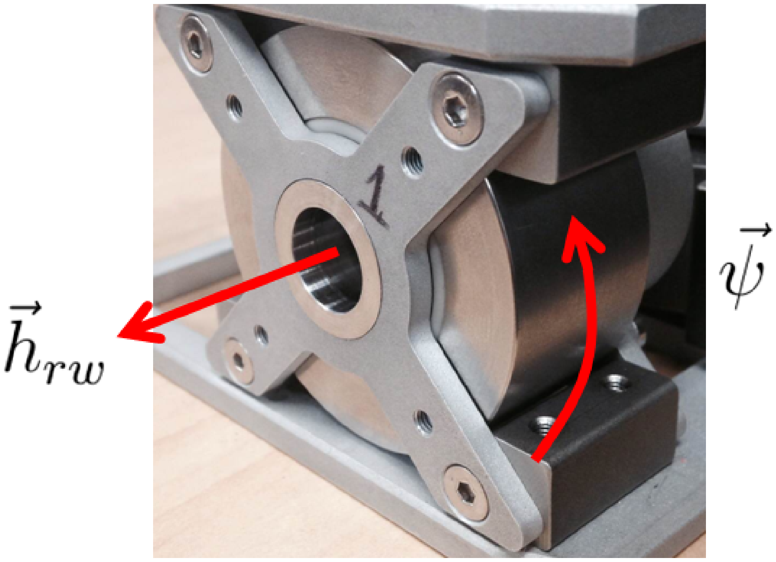


Figure 7. AFIT's 3rd Generation Reaction Wheel

The electric motor provides torque which causes the reaction wheels to accelerate at a rate of $\dot{\vec{\psi}}$. This will change the angular momentum of the reaction wheels which causes an equal and opposite change in the angular momentum of the spacecraft due to conservation of angular momentum. Note that by using the parallel axis theorem, the spacecraft MOI contains both the spacecraft's static MOI and the MOI of every actuator [14]. Also, reaction wheels are symmetric about their axis of rotation. Taking an inertial derivative of Eq. (21) and plugging into Eq. (19) yields Euler's equations

$$\vec{M} = \mathbf{I}\dot{\vec{\omega}}^{bi} + \vec{\omega}^{bi} \times \mathbf{I}\vec{\omega}^{bi} + \mathbf{D}\dot{\vec{\psi}} + \vec{\omega}^{bi} \times \mathbf{D}\vec{\psi} \tag{22}$$

with the addition of reaction wheels [46]. Note that Eq. (22) is three equations. Full three-axis control of a spacecraft will require at least three non-coplanar reaction wheels. If only three wheels were used, it would be optimal to align each wheel with each body axis. If aligned with the body axis, each wheel can devote all of its torque to each respective body axis. However, if one wheel fails on orbit, full 3-axis control will not be available using just the remaining two reaction wheels. This gives rise to the idea of redundancy by using more than 3 wheels to mitigate risk. The next section will discuss different types of sensors that will be used on the ADCS testbed.

2.3 Attitude Sensors

Accurate attitude knowledge is required for attitude control. Attitude sensors such as sun sensors, magnetometers, and IMUs are used to determine the spacecraft's orientation to a fixed reference frame. Sensors are by no means perfect in their operation, and without filtering and calibration, they can be highly inaccurate. One sensor is normally insufficient for determining attitude; multiple sensors are required within attitude determination algorithms. Each sensor must be tested independently so that filtering and bias gains can be implemented before sensor outputs can be used by attitude determination algorithms. The following subsections will give a detailed description of each sensor used in the ADCS testbed as well as typical sensor accuracy and sensor faults.

2.3.1 IMU Gyroscope and Accelerometer

Typically, IMUs consist of a rate gyroscope and accelerometer but may also include a magnetometer [8]. Gyroscopes are inertial sensors that integrate an angular acceleration measurement to provide an angular velocity about sensor axes. The gyroscope is inherently noisy when outputting angular rates and typically needs filtering. The gyroscope can be used in attitude determination algorithms for a short time period. By propagating the kinematic EOM, a quaternion may be estimated. Due to the integration of noise from the angular velocity outputs, however, there can be drift in attitude knowledge over time. This drift in the current quaternion is difficult to filter but may be useful for longer periods of time by periodically resetting the reference vector for the gyroscope [12]. This can be accomplished by resetting the gyroscope integration based on the characterization of the attitude drift rate; the initial condition needed for integration can be obtained from the quaternion output from the external sensor determination algorithm. Accuracy for gyroscopes is typically 0.001^o/hr but can be much higher for MEMS gyroscopes [43].

The linear accelerometer, as its name implies, measures the translational acceleration along a given axis. A simple means to test proper functionality of the accelerometer is to check the output when stationary on Earth. It should output Earth's gravity vector along the respective axis gravity is acting upon.

2.3.2 Magnetometers

Magnetometers are lightweight, relatively inexpensive sensors whose purpose is to measure the magnitude and direction of Earth's magnetic field in three-axes [53]. This magnetic field vector is provided in a magnetometer-fixed reference frame which can easily be converted to the body frame. This research effort assumes that the IMU is aligned with the ADCS testbed body frame so no additional rotation matrices are required.

Typically, the magnetic field vector is normalized due to direction being the primary need, not magnitude. Another requirement for determination is a mathematical model of Earth's magnetic field so that the magnetic field vector in the inertial frame can be used alongside the body frame vector to estimate the attitude [12]. Note that at least one more sensor such as an accelerometer will also be needed for statistical attitude determination algorithms such as QUEST. Earth's magnetic field, however, is subject to change and, therefore, impossible to model perfectly. Even if Earth's magnetic field was perfectly modeled, electronics and moving actuators such as reaction wheels distort the magnetic field around them. This typically encourages satellite designers to place magnetometers on the far reaches of the satellite so distortion can be minimized which can be challenging for CubeSats due to their small size and limited space for placement. Some actuators such as torque coils or rods render magnetometer readings useless when the actuator is being used [8]. Other sensors are then required when performing maneuvers with magnetic torquers. Proper testing of magnetometers is important to ensure that filters may be implemented based on the satellite's inherent magnetic field distortions. Typical magnetometer accuracy in low earth orbit (LEO) results in attitude estimation errors on the order of $\pm 5^o$ [43]. This estimation error is mostly due to inaccuracies in the magnetic field model and imperfect knowledge of the satellite's actual location. Using calibrated sensor readings from the accelerometer and magnetometer, the QUEST algorithm can produce accurate quaternion estimates. The next section will give a detailed description of the attitude determination process.

2.4 Attitude Determination

Attitude determination of any spacecraft is performed by calculating a quaternion estimate from multiple sensor inputs. Modern satellite attitude determination typically uses statistical determination algorithms instead of deterministic algorithms because deterministic algorithms can only implement measurements from two sensors and it typically assumes one sensor is completely accurate. Furthermore, standard deterministic algorithms, such as the Triad algorithm typically results in lower accuracy ratings than statistical methods [12]. These reasons make statistical methods desirable, and will be the focus of this section. While there are many statistical attitude determination algorithms, the QUEST algorithm will be the specific focus due to its ability to account for multiple sensor inputs [12]. Calibration methods and smoothing filters of sensors are generally needed before inputting a sensor output into QUEST to reduce noise and inaccuracy. These methods will also be discussed in detail.

2.4.1 Calibration Methods

As mentioned in the previous section, sensor outputs tend to be biased and noisy. Figure 8 shows three sinusoidal outputs, one representing the actual sinusoid, one representing a noisy, biased signal, and the other representing a noisy, bias corrected sensor output.

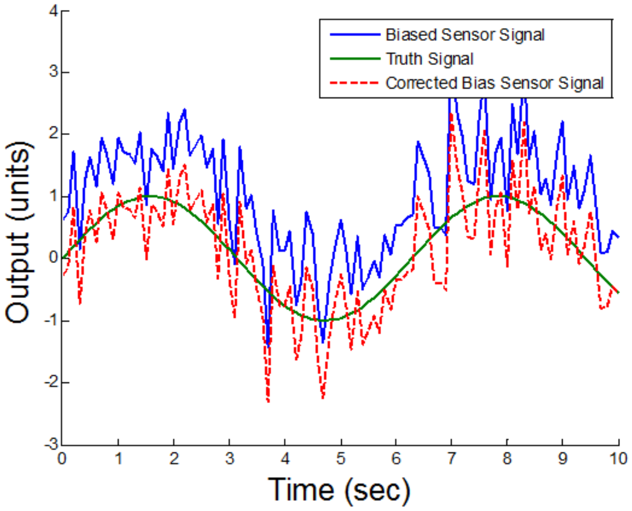


Figure 8. Biased Sensor Correction

Bias in sensors is normally very easy to determine experimentally [51]. To determine the bias, the average sensor reading must be subtracted from the truth reading. Figure 8 shows that this technique shifts the biased signal sensor to a signal that has a closer mean to the truth signal. Many different algorithms exist for filtering noise. One type of filter uses a "moving average" mathematical technique that will smooth the curve by ridding the data of outliers. This simple filter averages surrounding data points so that outliers are deleted; this essentially rids the signal of noise. The mathematical formula for a moving average filter is shown in

$$y(i) = \frac{1}{M} \sum_{i=1}^{M} x(i) \tag{23}$$

where $y(i)$ is the output signal, $x(i)$ is the input signal, and M is the number of points used to take the average [34]. Figure 9 demonstrates the usefulness of the "moving average" filter.

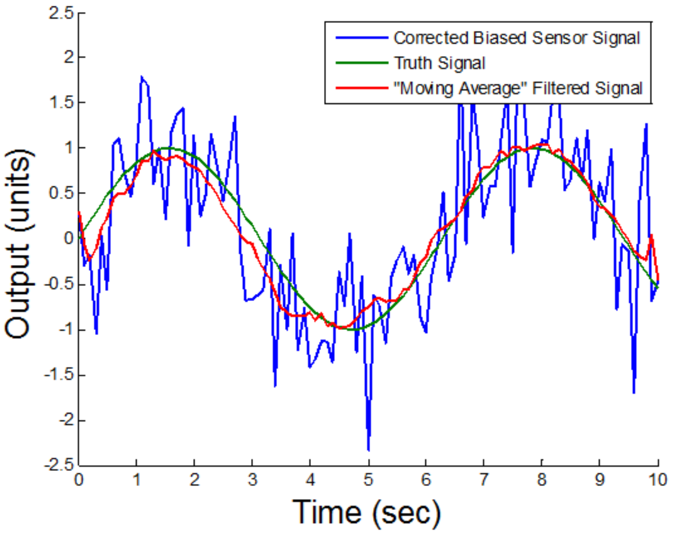


Figure 9. Moving Average Filter with Bias Calibration

Notice how well the filtered signal reduces the preceding noise. The more M is increased in Eq. (23), the smoother the curve. The trade off, however, is that the filtered signal will lose accuracy, so there is some iteration when choosing the number of points to take with the average. This can be accomplished by calculating metric values of error from corresponding values of M. Other types of filters that can be compared to the "moving average" filter are polynomial fits using a least squares estimator. Many proven algorithms exist to solve a polynomial curve given noisy data [19]. Typically, these filters can be smoother than a"moving average" filter because the end result is a continuous polynomial. Filtering may seem straight-forward, but calibration can be frustrating unless a robust algorithm is written to handle noisy data. For example, Earth's magnetic field is not constant in a satellite's orbit. A Helmholtz cage control system is a method to simulate different orbits by changing the magnetic field inside the cage [8]. Many iterations must be performed of simulating the satellite through an entire orbit with the predicted magnetic field. Filtering the measured magnetometer data may be more difficult because the magnetic field is not constant. Furthermore, actuators on board such as reaction wheels and torque coils will change the magnetic field around the spacecraft, which could cause inaccuracies within the filter. These magnetic field changes could change the bias term that was added to the magnetometer, which would in turn cause the smoothing filter to yield inaccurate results. Once sensor data is properly calibrated, it must be converted to a quaternion for attitude determination which is done via the QUEST algorithm that will be discussed in detail in the next section.

2.4.2 QUEST Algorithm [12]

The various sensors we have discussed output measurements in their respective sensor frames which must be related to estimates in the inertial frame. Conversion from the sensor frame $\vec{v}_s$ to the body frame $\vec{v}_b$ is mathematically described as

$$\vec{v}_b = R^{bs}\vec{v}_s \tag{24}$$

while the relationship between the body frame vector and the inertial frame vector is

$$\vec{v}_b = R^{bi}\vec{v}_i \tag{25}$$

where $\vec{v}_i$ is the sensor vector represented in the inertial frame. The sensor's output vector can be quickly converted to the body frame based on the fixed geometry of the spacecraft using R^{bs}. The inertial frame vector is typically calculated based on a mathematical model. For example, the Earth's magnetic field is mathematically modeled for a specific orbit and can be calculated based on position within that orbit [12]. This research effort will use a near constant magnetic field so that the mathematical model of the inertial reference frame is not needed. To estimate the attitude, R^{bi} is needed so that it may recover the Euler angles or be converted to a quaternion. Estimating the attitude with this formula, however, proves to be difficult with multiple sensor inputs. This is due to the fact that the problem becomes over determined when there are two or more sensors, and R^{bi} cannot satisfy every sensor measurement. One way to alleviate this problem is to introduce Wahba's loss function J shown in

$$J = \sum_{k=1}^{N} w_k(1 - \vec{v}_{kb}^T R^{bi} \vec{v}_{ki}) \tag{26}$$

where w_k is the measurement weight, and $\vec{v}_{kb}$ and $\vec{v}_{ki}$ are the k^{th} sensor vector in their respective frames [8]. Note that w_k is equal for all sensors used in this research effort and is equal to one. More robust uses of w_k could include a look-up table based on the accuracy of sensors for certain maneuvers. An example would be to set the w_k corresponding to the magnetometer to zero when using torque coils or rods. Minimizing Eq. (26) is equivalent to maximizing the negative of its derivative demonstrated by

$$g = \sum_{k=1}^{N} w_k \vec{v}_{kb}^T R^{bi} \vec{v}_{ki} \tag{27}$$

where g is the scalar gain function to be maximized. The problem must be converted to quaternions, which means a constraint will be added to the system. The algorithm now becomes a static optimization problem with one equality constraint as shown in

$$g = \underline{q}^T K \underline{q} \tag{28}$$

$$\bar{q}^T \bar{q} = I \tag{29}$$

where K is a matrix defining the gain function in terms of quaternions [31]. Note that I is the identity matrix.

With the addition of an equality constraint, a way to solve the static optimization problem will be to include a Lagrange multiplier. This constitutes a new gain function

$$g = \underline{q}^T K \underline{q} - \lambda \underline{q}^T \underline{q} \tag{30}$$

where λ is a Lagrange multiplier. To maximize this function, the derivative must be taken so that K can be calculated. The derivative of Eq. (30) becomes an Eigenvalue problem as shown in

$$Kq = \lambda q \tag{31}$$

where replacing λ for K and simplifying yields

$$g = \lambda_{max} \tag{32}$$

where the largest Eigenvalue of K maximizes the gain function. Since λ_{max} maximizes Eq. (27), the corresponding Eigenvector of this value is the optimal estimate of the current quaternion [12]. Although this gives an optimal estimate of the quaternion, solving the Eigenvalue problem is computationally intensive for an on-board computer; modifications to the algorithm must be made to create a more efficient way to solve the problem with little loss to accuracy. Since the solution to the optimization problem is known, Eq. (27) can be rearranged as

$$\lambda_{max} = \sum_{k=1}^{N} w_k - J \tag{33}$$

so that λ_{max} can be calculated without solving the Eigenvalue problem. Since the optimal Eigenvalue is being solved, the value of J should be very small. To ease computation, J is eliminated so that the equation becomes

$$\lambda_{max} \approx \sum_{k=1}^{N} w_k \tag{34}$$

The corresponding Eigenvector must now be calculated without solving the Eigenvalue/Eigenvector problem. An efficient way to perform this is to convert the quaternion using

$$p = \frac{q}{q_4} = \mathbf{a} * \tan\frac{\phi}{2} \tag{35}$$

where p is the Rodriguez parameters. Eq. (35) can be rearranged to result in

$$p = [(\lambda_{max} + \sigma)I - S]^{-1}Z \tag{36}$$

where S, and Z are QUEST sub-matrices. Note that I is the identity matrix. Instead of computing the inverse operation, simply rearrange Eq. (36) into the familiar $Ax = b$ and solve using Gaussian elimination from

$$[(\lambda_{max} + \sigma)I - S]p = Z \tag{37}$$

so the Rodriguez parameters can be calculated. The Rodriguez parameters shown in

$$q_{4\times 1} = \frac{1}{\sqrt{1 + p^T p}} \begin{bmatrix} p \\ 1 \end{bmatrix} \tag{38}$$

can then be converted into a quaternion.

The QUEST algorithm is one of the most efficient ways to estimate attitude using multiple sensors at once. It is the primary estimation algorithm on board the ADCS testbed. With proper sensor calibration and the use of QUEST, quaternion outputs should be accurate and relatively smooth. The next section will discuss satellite control theory and control laws.

2.5 Attitude Control

Attitude control is mainly concerned with minimizing the error between the spacecraft's desired and current orientations. Closed-loop control can be achieved by computing accurate attitude command torques from error, error rate, and the integral of the error. The torques can only be computed from a control law that inputs the quaternion error. The non-linear quaternion error must be used inside a linear operating range so that linear controllers such as proportional-integral-derivative (PID) may be used effectively. The spacecraft EOM must also be linearized. This can be performed by manipulating the control input using feedback linearization.

2.5.1 Quaternion Error

Quaternion error can be thought of as the difference between the desired and present quaternion. Quaternions cannot be simply subtracted from each other. Matrix multiplication is required to accomplish this. The quaternion error vector $\bar{q}_e$ and the commanded quaternion $\bar{q}_c$ are related with the present quaternion ($\bar{q}_p$) as shown in

$$\bar{q}_p = \begin{bmatrix} q_{4c} & -q_{3c} & q_{2c} & q_{1c} \\ q_{3c} & q_{1c} & -q_{4c} & q_{2c} \\ -q_{2c} & q_{1c} & q_{4c} & q_{3c} \\ -q_{1c} & -q_{2c} & -q_{3c} & q_{4c} \end{bmatrix} \bar{q}_e = \tilde{\mathbf{M}}(\bar{q}_c)\bar{q}_e \tag{39}$$

where $\tilde{\mathbf{M}}(\bar{q}_c)$ is orthonmormal and normally called the "transmuted quaternion" matrix [39]. Since the quaternion error is what will be inputted into the controller, Eq. (39) must be rearranged as

$$\bar{q}_e = (\tilde{\mathbf{M}}(\bar{q}_c))^{-1}\bar{q}_p = (\tilde{\mathbf{M}}(\bar{q}_c))^T\bar{q}_p \tag{40}$$

to solve for $\bar{q}_e$.

2.5.2 Linearized Quaternion Error

Quaternion error is inherently non-linear with respect to the Eigenangle ϕ from Section 2.1.2.2. Eqs. (7)-(10) present the non-linear relationship between the quaternion and ϕ due to the sine function. Due to the non-linear relationship, linear control of the spacecraft may provide erroneous results outside of a linear operating range [14]. To discover this range, the quaternions are linearized about $\phi = 0$ as shown in

$$q_1 = \hat{a}_1 \sin(\frac{\phi}{2}) \approx \hat{a}_1 \frac{\phi}{2} \tag{41}$$

$$q_2 = \hat{a}_2 \sin(\frac{\phi}{2}) \approx \hat{a}_2 \frac{\phi}{2} \tag{42}$$

$$q_3 = \hat{a}_3 \sin(\frac{\phi}{2}) \approx \hat{a}_3 \frac{\phi}{2} \tag{43}$$

$$q_4 = \cos(\frac{\phi}{2}) \approx 1 \tag{44}$$

Figure 10 graphically illustrates the non-linear portions, $\sin(\frac{\phi}{2})$ and $\cos(\frac{\phi}{2})$, vs. the linearized portions, $\frac{\phi}{2}$ and 1. Figure 10 also demonstrates that the linearized range is $\pm 0.6^o$ accurate for $q_{1,2,3}$ for ± 0.6 radians, which is about $\pm 35^o$. The accuracy range of q_4, however, is ± 0.2 radians or $\pm\ 12^o$. Using linearized quaternions assumes linear control. Every system, especially spacecraft, has non-linear qualities. The next section will discuss a proper way to linearize the EOM.

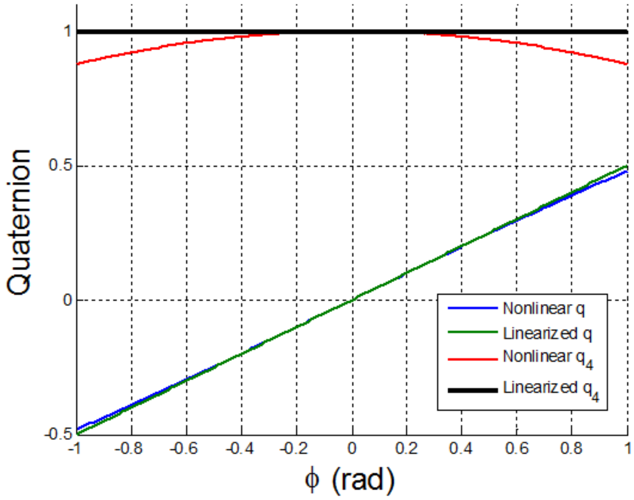


Figure 10. Linearized Quaternion Range

2.5.3 Feedback Linearization

The non-linear, kinematic EOM were discussed in Sections 2.1.3 and 2.2.2. Starting with Eq. (22) restated here:

$$\vec{M} = \mathbf{I}\dot{\vec{\omega}}^{bi} + \vec{\omega}^{bi} \times \mathbf{I}\vec{\omega}^{bi} + \mathbf{D}\dot{\vec{\psi}} + \vec{\omega}^{bi} \times \mathbf{D}\vec{\psi} \tag{45}$$

If $\vec{M}$ is set to 0 (i.e. no external torques), and Eq. (45) is solved for $\dot{\vec{\omega}}^{bi}$, then the new kinematic EOM become

$$\dot{\vec{\omega}}^{bi} = -\mathbf{I}^{-1}(\vec{\omega}^{x}\mathbf{I}\vec{\omega} + \vec{\omega}^{x}\mathbf{D}\vec{\psi} + \mathbf{D}\dot{\vec{\psi}}) \tag{46}$$

where Eq. (46) can be linearized effectively using feedback linearization. Feedback linearization for multiple-input-multiple-output (MIMO) systems is typically performed by dynamic inversion. Dynamic inversion uses feedback to linearize the system to be controlled so that desired dynamic properties such as asymptotic stability can be achieved [27]. Performing this operation requires the system dynamics to be in the 'controllability canonical form'. This form completely separates dynamics that are not attached to the input; resulting in two separate functions as shown in

$$x^n = f(x) + g(x)u \tag{47}$$

where x^n is the state derivative of order n, u is the input or control for the system, and $f(x)$ and $g(x)$ are non-linear functions of the states [33]. By simply choosing u to cancel out the non-linear functions of the dynamics, the general solution for any u is

$$u = \frac{1}{g(x)}[v - f(x)] \tag{48}$$

where v is now the linear control law demonstrated by

$$v = x_d - k_o x - k_1 \dot{x} - ... - k_{n-1} x^{n-1} \tag{49}$$

for tracking a desired state x_d [33]. Plugging u back into Eq. (47) will yield

$$x^n = v \tag{50}$$

which results in linear dynamics. Applying dynamic inversion to Euler's equations follows this format easily since it is already in controllability canonical form. Remembering that $\dot{\vec{\psi}}$ is the reaction wheel control, Eq. (45) is already in the proper form to use dynamic inversion. Rearranging Eq. (47), the controllability canonical form becomes

$$\dot{\vec{\omega}}^{bi} = \mathbf{I}^{-1}(\vec{\omega}^x \mathbf{I} \vec{\omega} - \vec{\omega}^x \vec{h}_{act} - \dot{\vec{h}}_{act}) \tag{51}$$

where solving for $\dot{\vec{h}}_{act}$ will result in the following control law:

$$\dot{\vec{h}}_{act} = \vec{v} - \vec{\omega}^x \mathbf{I}\vec{\omega} + \vec{\omega}^x \vec{h}_{act} \tag{52}$$

Plugging Eq. (52) into Eq. (51) results in the linearized dynamics that follow [40]:

$$\dot{\vec{\omega}}^{bi} = \mathbf{I}^{-1}(\vec{\omega}^x \mathbf{I}\vec{\omega} + \vec{\omega}^x \vec{h}_{act} + \vec{v} - \vec{\omega}^x \mathbf{I}\vec{\omega} - \vec{\omega}^x \vec{h}_{act}) = \mathbf{I}^{-1}\vec{v} \tag{53}$$

The ability to choose the new control v is very powerful considering that linear control theory is well established and understood. The next section will detail the PID control technique, which is the primary control law used in this research effort.

2.5.4 Proportional-Integral-Derivative Control

PID controllers are one of the most widely used controllers; more than half of all industrial controllers today use PID or modified PID controllers [24]. PID controllers are useful for systems with poorly understood dynamics and where analytical control approaches fail due to a lack of an accurate model. This approach is typically good for asymptotic stability and tuning response parameters, but not useful for optimal control problems such as minimizing time or power [24]. The typical PID controller operates on the angle error θ_e and its derivative to calculate the commanded torque τ_c that modifies the input into the "System". Figure 11 outlines a PID controller block diagram.

The controller begins by calculating the difference between the current angle θ and commanded angle θ_c which results in the angle error θ_e. Next, θ_e is split three ways and is manipulated by each segment of the controller; the separate paths are then summed to produce θ_c which is fed back to recalculate θ_e. The proportional gain K_p simply multiplies θ_e by a positive constant. Increasing K_p results in reducing response times, but consequently, results in larger overshoots and longer settling

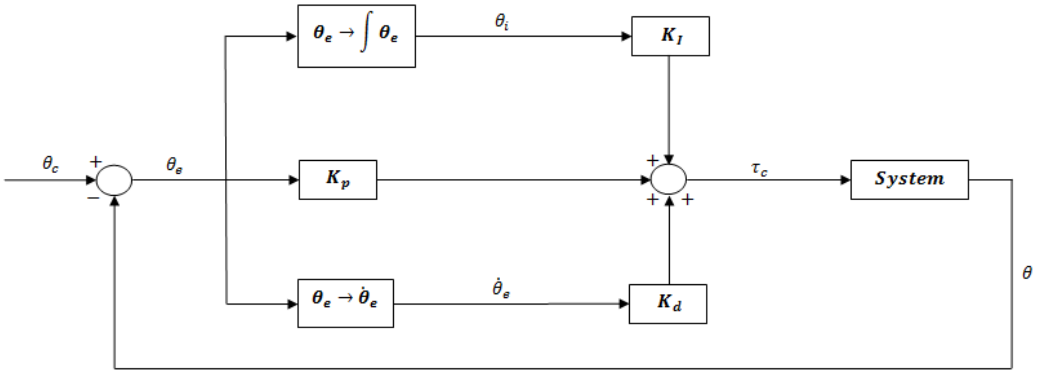


Figure 11. Block Diagram of a PID Controller

times. Depending on the system, high gain values of K_p can cause the system to become unstable.

The integral error is calculated by integrating θ_e. The integral error is scaled by the integral gain K_i. This control will act similarly to a proportional controller during the transient response, but more importantly, it is used to correct steady-state error within the system. If the actual output is steady but not at the desired value, the integral error will become very large over a long period of time. This causes a torque that should correct the steady-state error. The trade-off with using integral control is the increased response that is seen from the proportional control due to integral wind-up. Integral wind-up is "caused by large changes in error over short time periods, with finite control or limited control due to saturation, as produced when changing the desired state" [14]. While the error is corrected, the integrator winds up by accumulating error. This will cause the integral error to unwind, inducing oscillations that can increase transient response speed and settling time. Using integral control has problems for systems with limited control capability. Large error may cause control saturation, which would harm system response.

The final part of the PID controller is derivative control. This control acts differently than the other two due to its purpose to impede system response. The gain value for derivative control K_d can be thought of as a damping term that reduces overshoot caused by large gain values of K_p and K_i. Derivative control, like the other gains, has drawbacks. Since the error rate is being used to generate θ_c, outputs tend to be noisier. The ADCS testbed digitally computes the error rate $\dot{\theta}_e$ which results in a noisy output. Taking a discrete derivative of θ_e will result in

$$\dot{\theta}_{e_i} = \frac{\theta_{e_i} - \theta_{e_{i-1}}}{\Delta T} \tag{54}$$

where ΔT is the time step. Taking a discrete derivative will typically result in a large amount of noise for the estimated $\dot{\theta}_e$ because the derivative is only an approximation. By using a direct angular velocity measurement from a gyroscope, angular velocity error noise can be reduced by avoiding taking a discrete derivative. If error rates produce too much noise, it could cause system instability with higher K_d.

The addition of these terms shown in

$$\vec{v} = \tau_c = K_p\theta_e + K_d\dot{\theta}_e + K_i\int_0^t \theta_e \tag{55}$$

are used to calculate τ_c. The control law discovered from dynamic inversion $\vec{v}$ can be equated to τ_c to form the linear PID control law. Thus the closed loop dynamics of the spacecraft become

$$\dot{\vec{\omega}}^{bi} = \mathbf{I}^{-1}(K_p\theta_e + K_d\dot{\theta}_e + K_i\int_0^t \theta_e) \tag{56}$$

where $\mathbf{I}^{-1}$ remains in the control law. To counteract this, $\mathbf{I}^{-1}$ will be inverted to the left side of the equation which results in

$$\mathbf{I}\,\dot{\vec{\omega}}^{bi} = K_p\theta_e + K_d\dot{\theta}_e + K_i\int_0^t \theta_e \tag{57}$$

which is the final closed loop dynamics using the PID control law [40].

Many common spacecraft PID control laws are found in literature such as Sidi [32] and Wie [54]. Once the PID law has been established, PID tuning must take place to produce the desired step response. The gain tuning process for the PID controller on-board the ADCS testbed will be detailed in Section 3.5.7.

2.6 Literature Review

The following subsections detail previous work related to this research effort. The section will start with a discussion about satellite simulators. The section continues by surveying other universities and their research with CubeSat testbeds that include a Helmholtz cage and air bearing. The second part of this section will include other tests performed using an air bearing or Helmholtz cage that are of interest to this research effort. Finally, the latter part of the section will discuss the starting point of AFIT's CubeSat testbed and ADCS testbed.

2.6.1 AFIT's Satellite Simulator

Before satellite simulators, the method to test satellite capability was to either run simulations with a model or test the satellite while in orbit. Satellite simulators represent a cheaper way of testing hardware in the loop (HIL) systems to model uncertainties and characterize performance that computational models may not be able to do. AFIT's satellite simulator (Fig. 12) has been used since 1999 to develop satellite hardware, software, and attitude control algorithms [20].

Others have also successfully implemented satellite simulators such as the Naval Postgraduate School (NPS) [21] and the Air Force Research Lab (AFRL) [25]. Satellite simulators have successfully given designers the opportunity to test new hardware, software, and algorithms that otherwise would only had been performed with the use of models or tested in orbit.

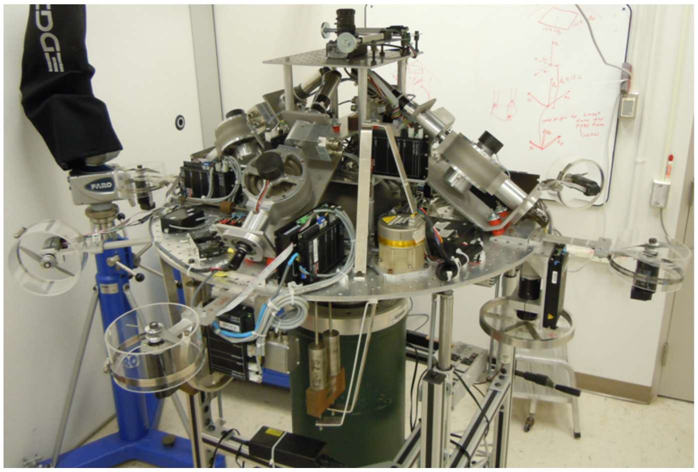


Figure 12. AFIT's Satellite Simulator

2.6.2 CubeSat Testbeds

CubeSat testbeds take existing ideas of testing hardware, software, and algorithms on satellite simulators and apply them to CubeSats. The main idea behind a CubeSat testbed is to be able to test CubeSat ADCS to gain accurate HIL data that validates performance. Three universities who have developed CubeSat testbeds are presented in the following subsections.

2.6.2.1 Naval Postgraduate School

The Naval Postgraduate School claims to have designed the first ever CubeSat three-axis simulator (CubeTAS) [21]. The CubeTAS testbed consists of typical ADCS components such as reaction wheels, a sun sensor, an IMU, and torque coils [5]. The air bearing has a custom made automatic mass balancing system that is used to align the center of mass and center of rotation, which would eliminate gravitational torque so three-axis control can be tested. The testbed also includes a Helmholtz cage for three-axis magnetic field control inside the cage [5]. The CubeTAS, air bearing, and Helmholtz cage are shown in Fig. 13. With gravitational torques removed, a three-axis stabilization maneuver was successfully performed where convergence on the desired angle occurred at around 120 seconds. NPS plans to use CubeTAS to give engineering students hands on experience of nanosatellite ADCS design and test. Future work for NPS includes the validation of attitude control techniques with the use of magnetic control torques such as momentum dumping of the reaction wheels and detumbling maneuvers [6].

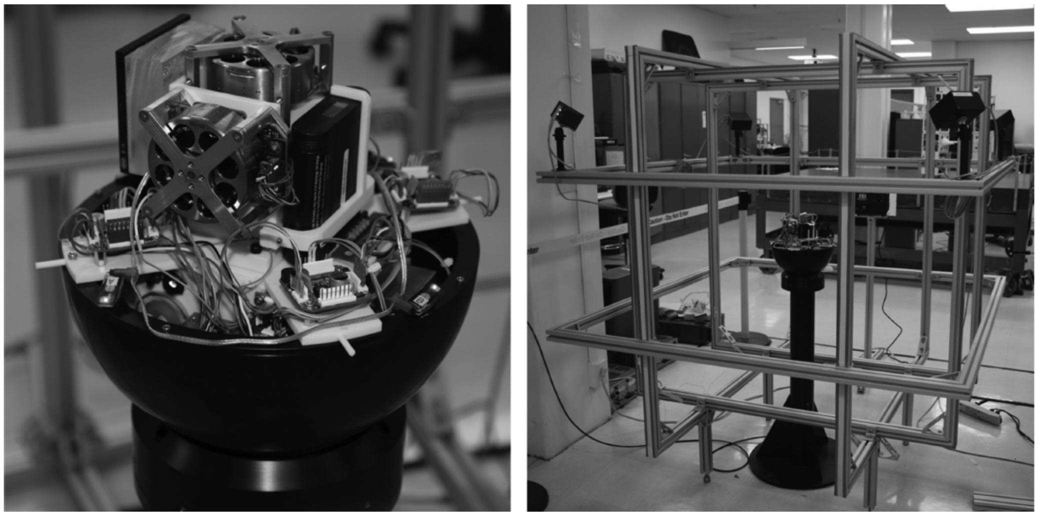


Figure 13. CubeTAS Simulator at the Spacecraft Robotics Laboratory of the NPS [6]

2.6.2.2 Massachusetts Institute of Technology

The Massachusetts Institute of Technology (MIT) has successfully built a CubeSat testbed consisting of a Helmholtz cage and spherical air bearing. Initial calibration of the cage produced good results that led MIT to think that their CubeSat testbed will be used to test the ADCS of small satellites in the future [26]. MicroMAS is a CubeSat currently being developed, and ADCS testing is planned to be performed inside MIT's CubeSat testbed. MicroMAS will be equipped with reaction wheels and torque coils which makes it similar to actuator choices on AFIT's 6U CubeSat [2]. MIT is also planning other CubeSats to be tested using this testbed. To the author's knowledge, no testing of MicroMAS or other CubeSats using this CubeSat testbed has occurred.

2.6.2.3 Istanbul Technical University

Istanbul Technical University (ITU) currently has an integrated air bearing table inside a Helmholtz cage. A student group at ITU was tasked with the development of the ITU PSAT II, a nano-satellite that requires three-axis control [50]. The students designed software in the loop (SIL) simulations that showed that the hardware was capable of achieving three-axis control and developed the simulation steps necessary to begin HIL testing [50]. Future work for ITU will be the ADCS testing of PSAT II using their CubeSAT testbed. To the author's knowledge, testing of PSAT II has not yet been performed using ITU's CubeSat testbed.

2.6.3 CubeSat Testbed Experiments

CubeSat testbeds are relatively new, and testing actual CubeSat hardware with CubeSat testbeds is difficult to find due to limited available literature. Many people have written papers on CubeSat ADCS algorithms for estimation and control and simulated results using models while very few have performed actual HIL testing inside the CubeSat testbed to validate simulation. The lack of literature on this subject becomes even more limited with reaction wheel testing in a CubeSat testbed. The following subsections discuss tests that have been performed inside and outside a CubeSat testbed.

2.6.3.1 University of Arkansas Attitude Control Testing

An older method of performing ADCS testing on the ground is to hang the CubeSat by a wire. This suspension in mid-air rids the CubeSat of all disturbance torques from friction but still leaves a small torque because of wind up from the wire. The University of Arkansas (UA) created a low cost air bearing to test a 3U CubeSat's reaction wheels. The ADCS consisted of a PD controller where the desired torque is sent to the reaction wheels. The controller was first designed in MATLAB Simulink along with attitude knowledge using an IMU [28]. Problems with the air bearing forced UA to test the ADCS by using a wire. Reaction wheel testing showed that the ADCS could point to close to one degree of accuracy even with a disturbance torque caused by the test set-up. The tests concluded that under one degree of pointing could be achieved with COTS components [28].

2.6.3.2 California Polytechnic State University Magnetometer Calibration

At California Polytechnic State University (CPSU), a graduate student worked on calibrating and testing the magnetometers that were to be installed on CPSU CubeSats with the use of a Helmholtz cage. The magnetometer was placed inside the Helmholtz cage, and the Helmholtz cage simulated a rotation about the z-axis to test the x- and y-axes sensors and a rotation about the x-axis to test the y- and z-axes sensors [10]. The maximum magnetic field strength tested was 50,000 nT which equates to 500 mG. After using a linear filter and accounting for magnetic field distortions caused by the CubeSat bus, estimation accuracy from the magnetometer was on the order of 0.2^o. It is noted that the student filtered the magnetometer in two different field strengths, and the estimation accuracy was better with the stronger field [10].

2.6.3.3 York University Attitude Control Testing

York University created a CubeSat testbed meant for 1U CubeSats, which consists of a spherical air bearing. The air bearing provides full three degrees of freedom, and the platform includes a mass balancing system [17]. Note that York does not have a Helmholtz cage as part of their CubeSat testbed. The 1U CubeSat was successfully implemented onto the air bearing, and single-axis control was tested about the z-axis. Two different controllers were tested: a PID controller and a non-linear controller. The PID controller gave pointing accuracy about the z-axis on the order of 5^o. Note that the x- and y-axes were not stabilized and had angle error as well. The non-linear controller produced better pointing accuracy on the order of 1^o. Note that simulation from a model provided York with pointing accuracy predictions of less than 0.09^o; this predicted value from simulation did not match actual data from the air bearing. Future work includes the addition of a Helmholtz cage for magnetic control tests [17].

2.6.4 AFIT's CubeSat Testbed and ADCS Testbed Starting Point

At the start of this research effort, AFIT's CubeSat testbed was composed of a fully functional Helmholtz cage and air bearing. Work had been performed by previous students to ensure AFIT's CubeSat testbed worked properly, namely Brewer [3] for the Helmholtz cage and Dannemyer [8] for validation of the air bearings operational capabilities. The four wheel RWA was used by Dannemeyer and is discussed further in Section 3.2.2.

Dannemeyer continued her research effort by creating a MATLAB Simulink diagram of the ADCS testbed. The diagram shown in Fig. 14 is not complete but has the capability to implement actuators such as torque coils and reaction wheels with a de-tumbling algorithm. The algorithm is also capable of implementing orbital information using the Satellite Tool Kit (STK). This orbit information couples with the World Magnetic Model (WMM) Simulink block to calculate the expected magnetic field while in orbit. The model can also implement scenarios with the loss of a reaction wheel, and a momentum dumping algorithm exists. The model's downside is that it does not incorporate sensor noise or external disturbance torques.

The MATLAB Simulink model was converted to C code and utilized by the ADCS

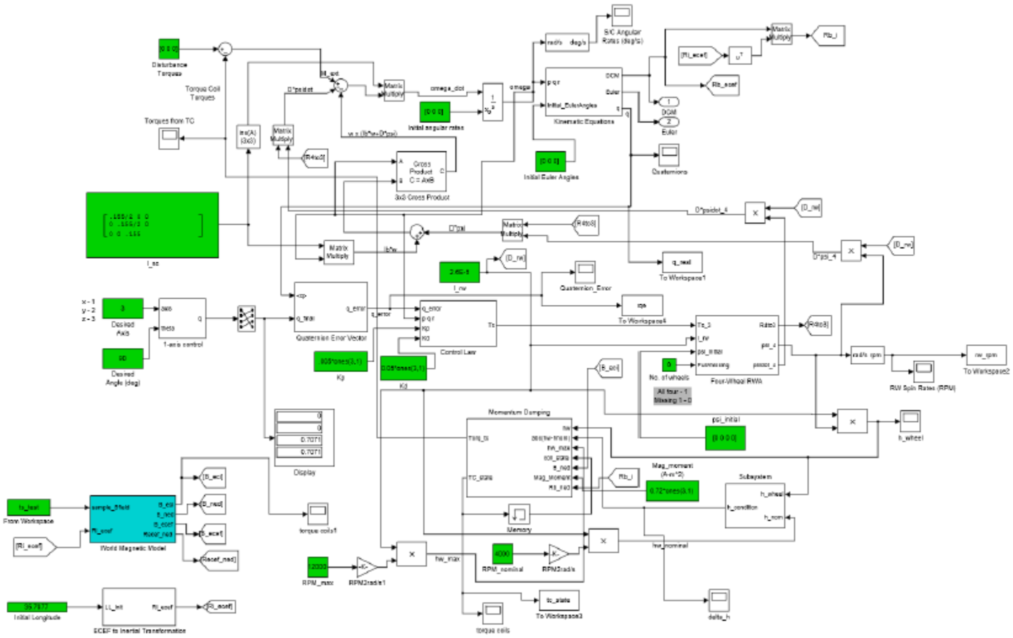


Figure 14. Dannemeyer 6U CubeSat Simulink Model [8]

testbed. The conversion was not perfect and software issues arose. Time was spent redesigning and debugging to facilitate operational code. The ADCS testbed was tested atop the air bearing at the end of her research effort as shown in Fig. 15. Pointing accuracy of the ADCS testbed when inside AFIT's CubeSat testbed resulted in pointing errors around $\pm 20^o$ while estimation accuracy was $\pm 3^o$. At the time, the ADCS algorithm could only update at 1 Hz due to limitations with the ADCS task configuration; the ADCS algorithm update speed and the Simulink diagram conversion to C code was one of the causes of the pointing performance of the ADCS testbed. Another problem that existed was the lack of magnetometer and gyroscope filtering.

After Dannmeyer graduated, Lippert and Dicks, an AFIT contractor and undergraduate student at Cedarville University working as an intern, were successfully able to run the ADCS algorithms at 10 Hz. This faster update speed meant that more precise control could be achieved. Brewer, Dannemeyer, Lippert, and Dicks laid the

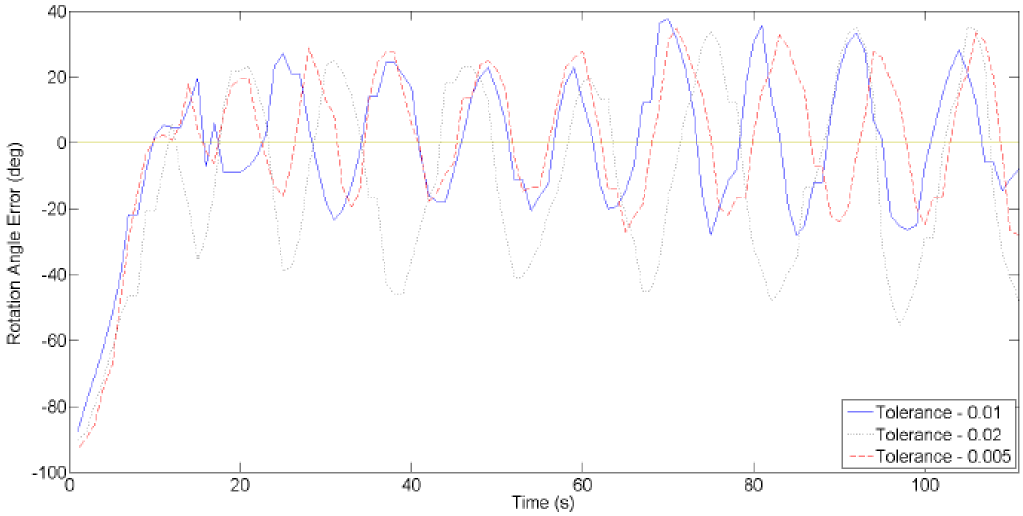


Figure 15. z-axis Rotation Error for 90^o Slew Using Different Tolerance Bands [8]

groundwork in creating a functional testbed. Without their dedication and work to the ADCS and CubeSat testbeds, I could not have accomplished the work described in this thesis.

2.7 Chapter Summary

Chapter II covered the background information and related research necessary to perform this research effort. The chapter began by defining many different coordinate frames used in this research effort. The EOM that describe rotation matrices, Euler angles, Eigenaxis rotations, and quaternions were explained in detail. Attitude dynamics were then separated and formulated into the kinematic and kinetic EOM. The chapter continued with an explanation of angular momentum exchange and how attitude actuators can be used to manipulate the kinetic EOM. The different attitude sensors used in this research effort were then discussed in detail. The next part of the chapter sought to demonstrate the process of attitude determination. This started with an explanation of bias calibration and noise filtering and ended with a detailed description of the QUEST algorithm. Attitude control was discussed next in the chapter. The quaternion error was calculated, and the effects of linearizing this error were discussed. Secondly, feedback linearization, specifically the dynamic inversion technique, was used to rid the kinetic EOM of any non-linearities so that linear control could be accomplished effectively. PID controllers were derived and implemented in the closed loop dynamics of the kinetic EOM. Finally, a literature review detailing CubeSat testbeds from other universities, specific tests that have been performed on CubeSats through the use of Helmholtz cages and air bearings, and a description of the starting point of this research effort was discussed. The overview and testing of the ADCS testbed, including actuator selection, sensor selection, AFIT's CubeSat testbed, and test plans will be discussed in Chapter III.

III. Development and Test Methodology

Chapter III will describe in detail the hardware and testing methodologies applied to the 6U ADCS. The first part of the chapter will discuss the AFIT CubeSat test bed which includes AFIT's Helmholtz cage and air bearing. Next, specific sensor and actuator choices for the ADCS test bed will be presented as well as their quoted specifications. The software configuration of the ADCS test bed circuit board and algorithms will also be illustrated. Finally, the chapter presents the experiments for testing the ADCS test bed functionality and performance inside the AFIT CubeSat test bed.

3.1 AFIT's CubeSat Test Bed

The AFIT CubeSat test bed shown in Fig. 16 includes multiple hardware testing devices that helps mimic a few aspects of an on orbit environment. First, a Helmholtz cage is used to simulate Earth's magnetic field in any orbit while the air bearing provides a near frictionless rotational environment for satellite attitude maneuvers. A description of the Helmholtz cage and air bearing are detailed in the following subsections.

3.1.1 Helmholtz Cage

A Helmholtz cage, named after German scientist Hermann von Helmholtz (1821-1894), is a device capable of controlling the magnetic field inside the cage. To ensure 3-axis control of the Helmholtz cage, three orthogonal pairs of Helmholtz coils should be utilized. Helmholtz coils consist of two coils parallel to each other with each coil consisting of N wrappings. A magnetic field B will be produced as current I passes through the coil pair [3]. Helmholtz cages are available commercially, and many

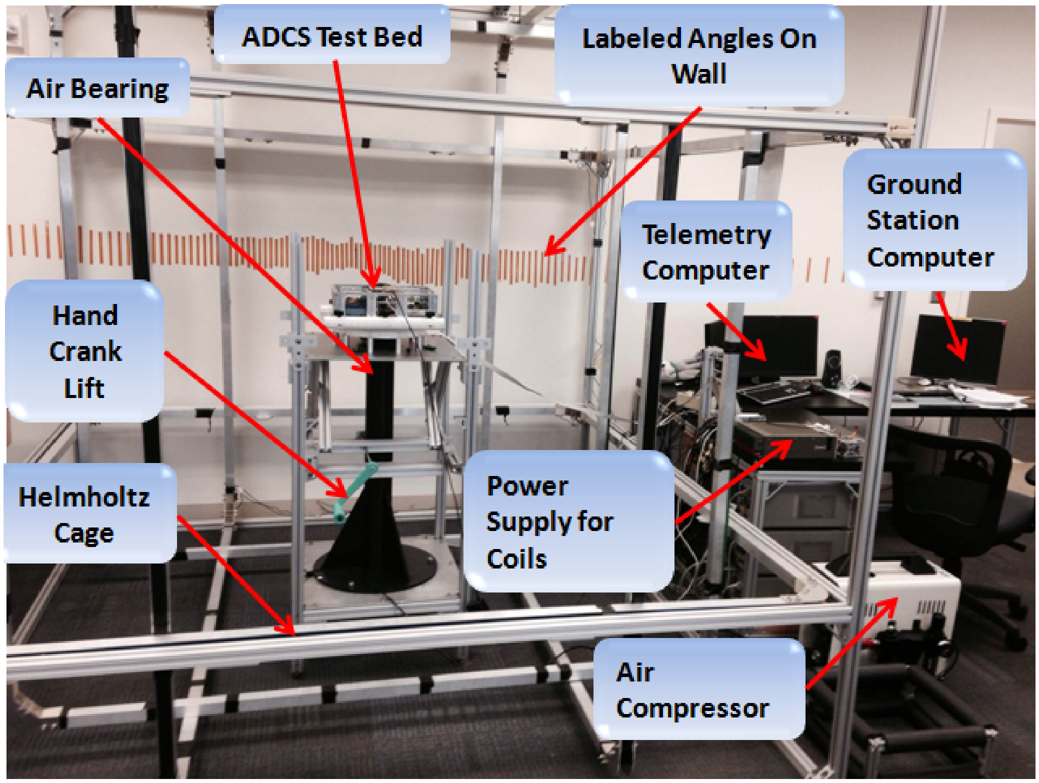


Figure 16. ADCS Test Bed Inside AFIT's CubeSat Test Bed

universities have built their own for ADCS development. For further references on AFIT's and other universities' Helmholtz cages, refer to Dannemeyer [8] and Brewer [3]. The AFIT Helmholtz cage (Fig. 17) is capable of producing ±2 Gauss in the x-, y- and z-axes .

The Helmholtz cage, while very useful, can still not produce a perfectly uniform magnetic field inside the cage. Prior research at AFIT, however, determined that magnetic field variations inside the cage are negligible [3]; for this reason, any minor variations inside the cage will be considered negligible when analyzing magnetometer data for this research effort. As discussed in Section 2.3.2, magnetometer testing using a constant magnetic field does not represent a changing magnetic field seen while orbiting Earth. The Helmholtz cage has been successfully controlled by the STK which is a software package used to simulate and model satellites and their

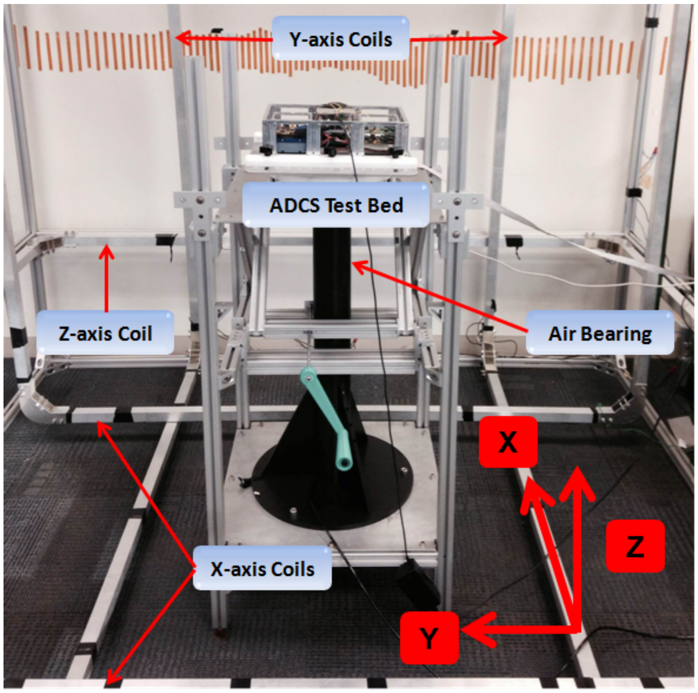


Figure 17. AFIT's Helmholtz Cage

orbits and other various platforms. A specific program inside the STK is the Space Environment and Effects Tool (SEET); this tool can simulate the magnetic field at any given point in a satellite's orbit. The IMU can then be tested in a real time orbit simulation of the magnetic field they will experience. This can further validate the magnetometers for space readiness by testing them in a constantly changing magnetic field.

3.1.2 Air Bearing

To properly test the ADCS in a terrestrial environment, minimizing friction is a necessity. Mechanically, friction can be minimized through the use of an air bearing. The air bearing separates two surfaces by a thin layer of compressed air supplied by one of the surfaces. This provides a near frictionless surface that is the preferred testing platform for ADCS development [29]. While there are many types of air bearings, this research effort will use a hemi-spherical air bearing (illustrated in Fig. 18) which grants 360^o range of motion to the z-axis while the other two axes are only able to achieve less than 90^o range of motion.

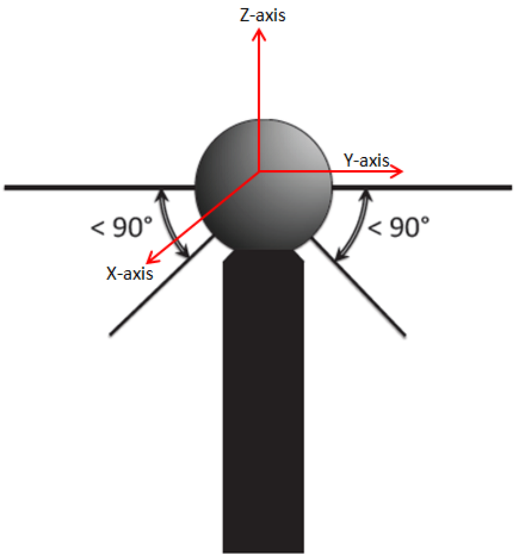


Figure 18. Spherical Air-Bearing [20]

While the spherical design grants a sufficient range of motion for three-axis testing, there is not an easily implemented platform for the ADCS test bed placement. To counteract this problem, the sphere is essentially cut in half, leaving a flat surface upon which the ADCS test bed may be placed. Illustrated in Fig. 19, the tabletop design has limited range of motion in the x- and y-axes to roughly 60^o.

AFIT's tabletop design extends the flat part of the sphere outward so that it may hold larger objects such as 6 and 12U chassis as shown in Fig. 20. The air bearing is created between the hemi-sphere and the ports on the opposing surface that jettison air to create the near-frictionless effect. The delrin ring contacts the air bearing when the air bearing is not being used (see Fig. 20). This contact also centers the bearing when lowered.

Balancing the test stand, which includes the hemi-sphere, loading surface, and ADCS test bed, is the most difficult challenge when preparing the air bearing for testing ADCS. While near frictionless, the air bearing is still subject to disturbance

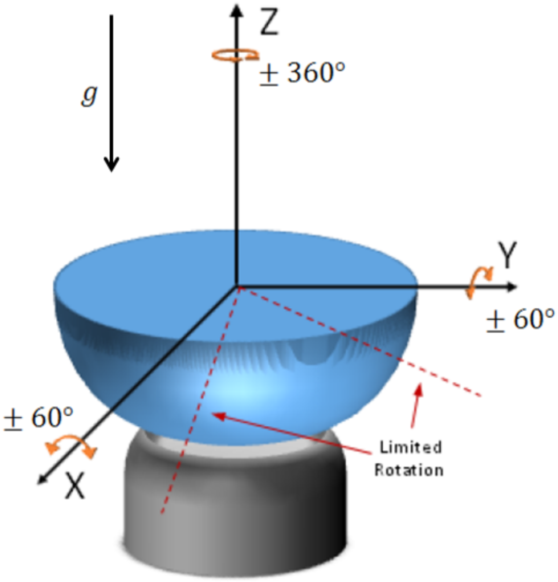


Figure 19. Tabletop Air-Bearing [8]

torques caused by air movement, magnetic fields, and gravity [8]. Other disturbance torques may be caused by wires attached to the ADCS test stand so wireless telemetry, tracking, and command system is required. Although these torques may be impossible to eliminate completely, minimizing gravitational disturbance torques can be achieved. The tabletop air bearing presents an interesting problem due to the alignment of the load surface's COM and the air bearing's center of rotation (COR). An unstable system occurs when the COM is above the COR whereas the system is stable when the reverse is true. When the COR is coincident with the COM, the loading surface can stay at any desired orientation. This precise placement, however, is not entirely necessary for control of the z-axis because gravitational torques do not affect movement about the z-axis. This research effort will ensure that the COM is closely aligned, not coincident, with the COR. This will ensure stability about the x- and y-axes so that z-axis control can be properly tested.

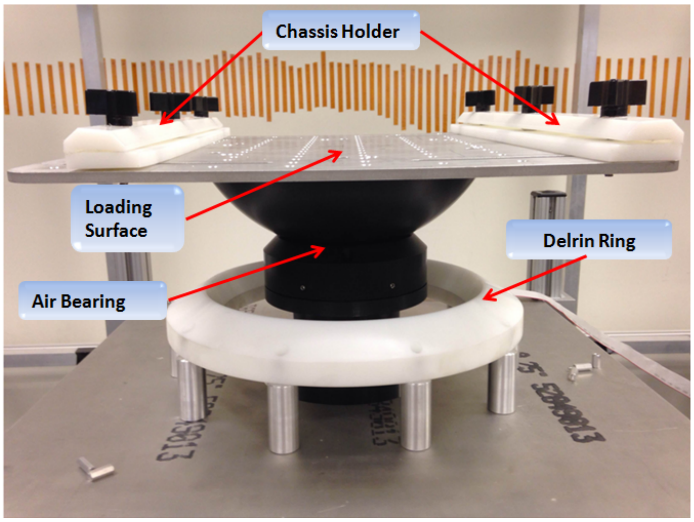


Figure 20. AFIT's CubeSat Air Bearing

3.2 ADCS Test Bed Sensor and Actuator

This section will cover the ADCS test bed sensor and actuator choices. The primary sensor used during this research effort was an IMU containing a gyroscope, accelerometer, and magnetometer. The primary actuator used was a RWA that consists of four reaction wheels.

3.2.1 IMU Selection

The IMU that was chosen for the 6U ADCS is the Analog Devices High Precision Tri-Axis IMU ADIS 16405 model. This provides a simple to use device that is compact and cost effective compared to other IMUs. The maximum rate range is ± 75 ^{o}/sec for the gyroscope, ± 18 g's for the accelerometer, and ± 2.5 Gauss for the magnetometer. The gyroscope has 0.05^o/sec of precision while the magnetometer has 0.5 mG of precision. Note that the gyroscopic drift rate from integrating the angular velocity output is not needed because the gyroscope was not used in attitude estimation. A temperature sensor is also on board as well as a digital to analog converter (DAC) [9]. The IMU measures in at 23 mm x 23 mm x 23 mm as shown in Fig. 21. Detailed descriptions of the functionality of the gyroscope, accelerometer, and magnetometer are included in Section 2.3.

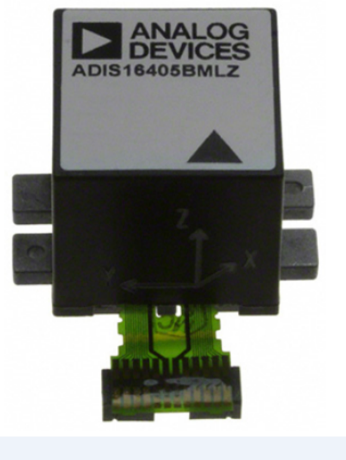


Figure 21. ADIS Model 16405 IMU

3.2.2 AFIT's Reaction Wheel Array Design

The ADCS test bed houses a 4-wheel pyramid RWA which includes by design redundancy in case of the loss of power to one wheel as shown in Fig. 22. The most significant flaw in the design was that the RWA could not survive spacecraft qualification testing which means that it would may not function properly once deployed into space from the launch vehicle [13]. The design, however, has low friction and redundancy, making this RWA very useful in a lab setting for attitude control.

Since the RWA design is not aligned with the body frame, a 4x3 torque mapping matrix will be required to account for each of the four reaction wheels. This matrix will map a torque vector from the body frame to each reaction wheel. Figure 23 illustrates the RWA coordinate system and the angles, α and β, between the RWA coordinate frame and the body frame. Note that α is 45, 135, 225, and 315^o for the x-axis for wheels 2, 3, 4, and 1, respectively while β is 45^o for the y-axis. Also note that the principle body axes of the ADCS test bed and reaction wheels are coincident.

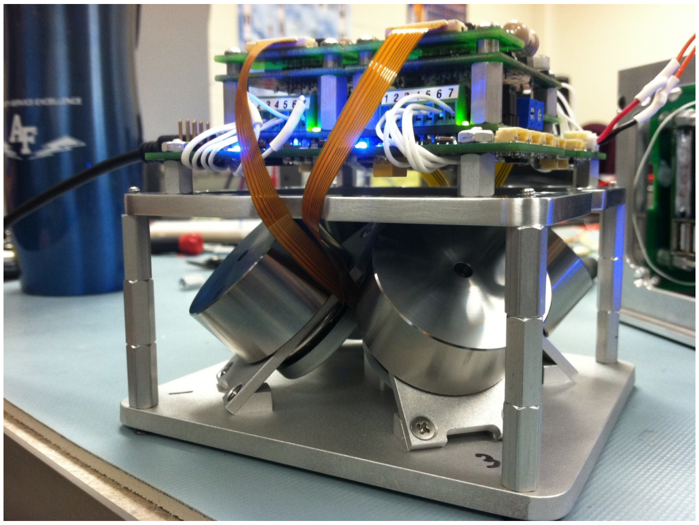


Figure 22. AFIT's Four Wheel Pyramid RWA [8]

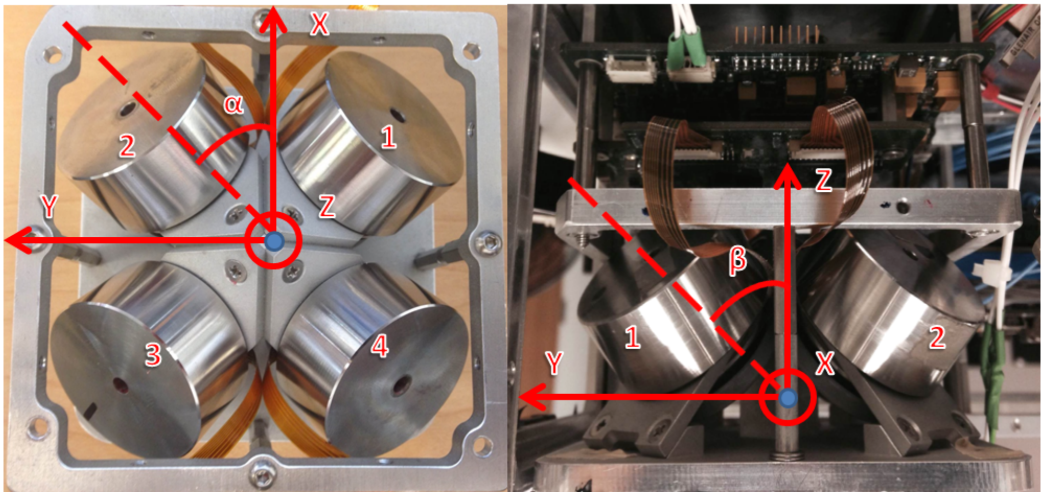


Figure 23. Four Wheel Pyramid Array Coordinate System and Torque Mapping [8]

The mapping matrix discussed is shown in Dannemeyer [8]. Since control of just the z-axis is desired, then the torque output to each reaction wheel is exactly the same as shown in

$$\tau_w = \tau \cos \beta \tag{58}$$

where τ_w is the scalar torque of the respective reaction wheel and τ is the torque in the z-axis from the ADCS PID controller.

Because the motors are not encased, the magnetic fields from the motors can cause distortions to the magnetometer in the IMU which is only centimeters away. This requires extra filtering which is detailed in Section 4.2.3.

Although there are four reaction wheels available, only three will be used during this research effort because the motor controller board currently being used has the ability to only handle three reaction wheels. This leaves one reaction wheel unused which will limit total angular momentum storage, torque output, and control authority. This research effort desires to create a single-axis control system by giving the same control input shown in Eq. (58) to each reaction wheel motor. This means that the x- and y-axes will not be controlled; this control usage along with only using the three reaction wheels will result in a non-linear coupling that causes a nutation as the ADCS test bed slews about the z-axis. The nutations will be small, and it will not effect the stability of the ADCS test bed in any of the body axes due to the alignment of the ADCS test bed's COM with the air bearing's COR. To have a three-axis controlled system, the control algorithm will need to be updated to implement torques that handle the nutations about the x- and y-axes. The hardware configuration of the ADCS test bed will be discussed next.

3.3 Hardware Configuration

The following sections detail an explanation of the required circuit boards onboard the ADCS test bed (Fig. 24): this includes the ADCS, CDH, and EPS circuit boards.

The ADCS has a motor controller daughter board, the CDH uses WiFly capability, and the EPS board contains the battery pack. The stacks that were created and implemented into the 6U chassis will also be discussed. The ADCS implmentation inside AFIT's CubeSat test bed will be analyzed as a whole. Finally, the MOI in the

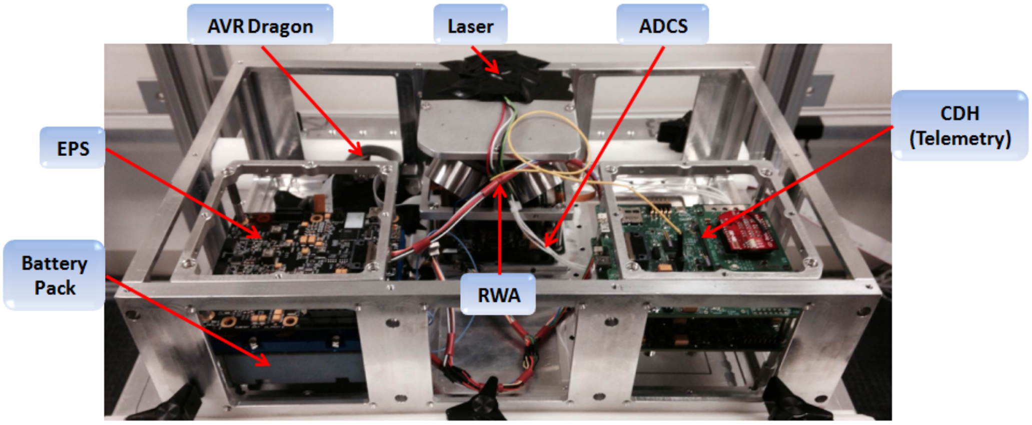


Figure 24. ADCS Test Bed

z-axis of the ADCS test bed and loading surface will be properly measured.

3.3.1 ADCS Board

The ADCS board used for the ADCS test bed customized at AFIT (Fig. 25) can interface with all required ADCS test bed sensors, actuators, and algorithms. The ADCS controller hosts a motor controller daughter board (Fig. 25) for the reaction wheels.

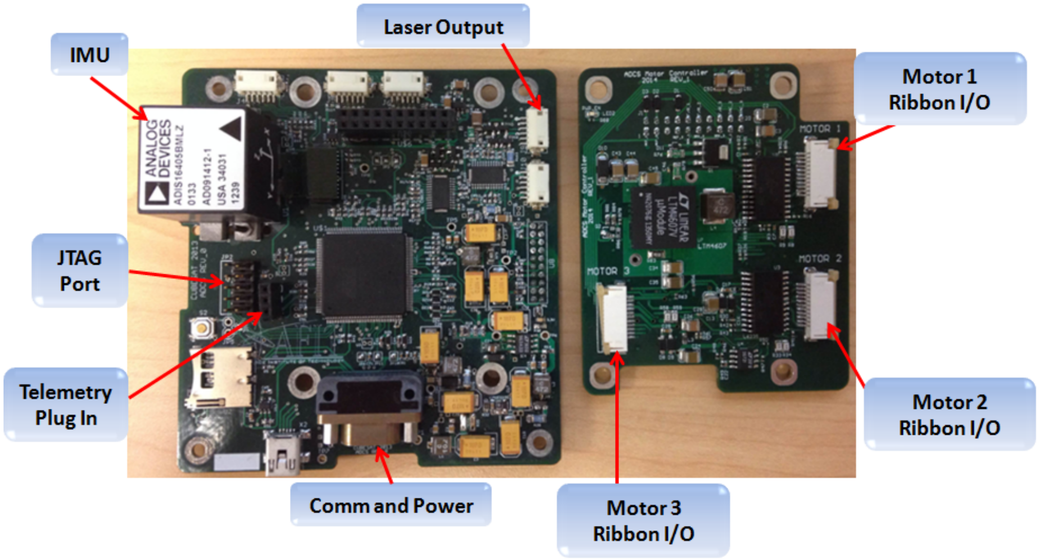


Figure 25. ADCS Circuit Board (left) Motor Controller Daughter Board (right)

The ADCS controller board has a 32-bit microcontroller (AT32UC3C 0512C) which runs in C-code in a Free RTOS real-time operating environment. The microcontroller runs at 16 MHz, contains 512 kB of boot random access memory (RAM), has 64 kB of data memory, and uses its floating-point unit for controller algorithms [8]. An AVR Dragon is used for programming and debugging the ADCS. A telemetry port is also available which is invaluable for testing.

3.3.2 CDH Board

The CDH circuit board shown in Fig. 26 provides this research effort with wireless telemetry to the user. Note that the board uses the telemetry port by connecting a wire to the ADCS telemetry port. This data is sent from the ADCS to the CDH where the WiFly sends the information to the connected computer. A more detailed description of how telemetry is sent wirelessly will be discussed in Section 3.4.5.

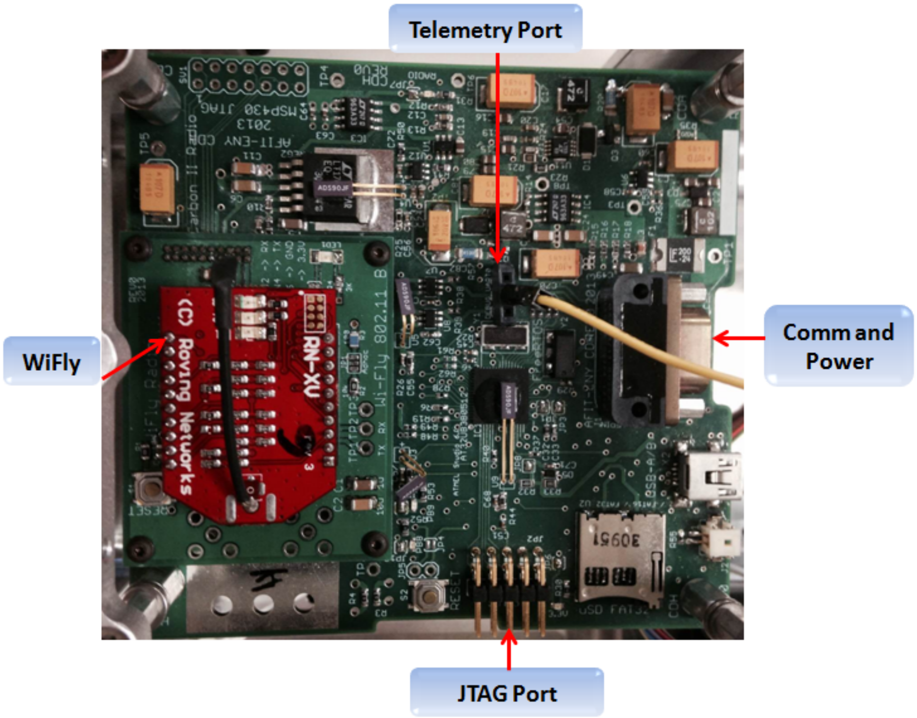


Figure 26. AFIT's CDH Board

3.3.3 EPS Board

The EPS board shown in Fig. 27 powers the ADCS, CDH, and reaction wheels. The board and battery pack output 8.4 V to the CDH and ADCS. The battery port connects the battery pack to the EPS; the charging port is used to connect the charger to the EPS. If the jumper covers are removed, the output port will not send voltage to the CDH and ADCS, thus halting operation of the board. In order to avoid damaging board components, jumper covers should be removed first before disconnecting the output port. Secondly, the jumper covers must also be resident to charge the batteries through the charging port.

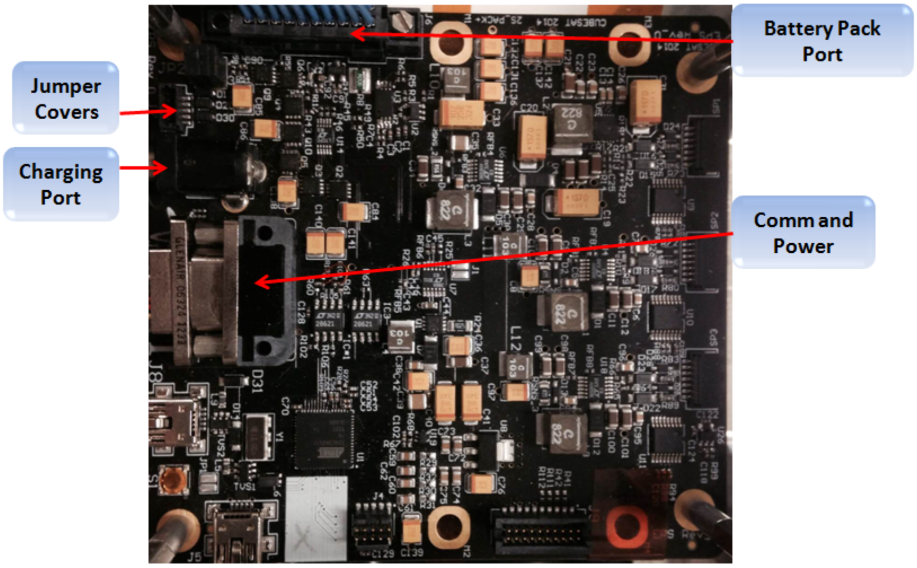


Figure 27. AFIT's EPS Board

3.3.4 Implementation of ADCS Test Bed

Before placing the 6U chassis on the air bearing, circuit board stacks and supporting hardware must be assembled and integrated. The combination of the required stacks and 6U chassis make up the ADCS test bed. As seen in Fig. 24, the first stack consists of the EPS board and battery pack. The second stack consists of the CDH board used for telemetry. The third and final stack consists of the ADCS board, motor controller board, and RWA. Mounted on top of the ADCS stack is a laser that plugs into the ADCS. This laser was used extensively to point at desired angles displayed on the wall (Fig. 16). Once the stacks were configured inside the chassis, the ADCS test bed needed to be properly mounted on top of the air bearing. Figure 20 shows clamps that hold the chassis stable. The chassis slides into the holders where the clamps secured the chassis tightly. Before securing the clamps, the chassis had to be positioned on the loading surface with the air bearing on so that it could be balanced. Imbalances can cause unwanted movement of the loading surface. Figure 28 demonstrates the balancing process.

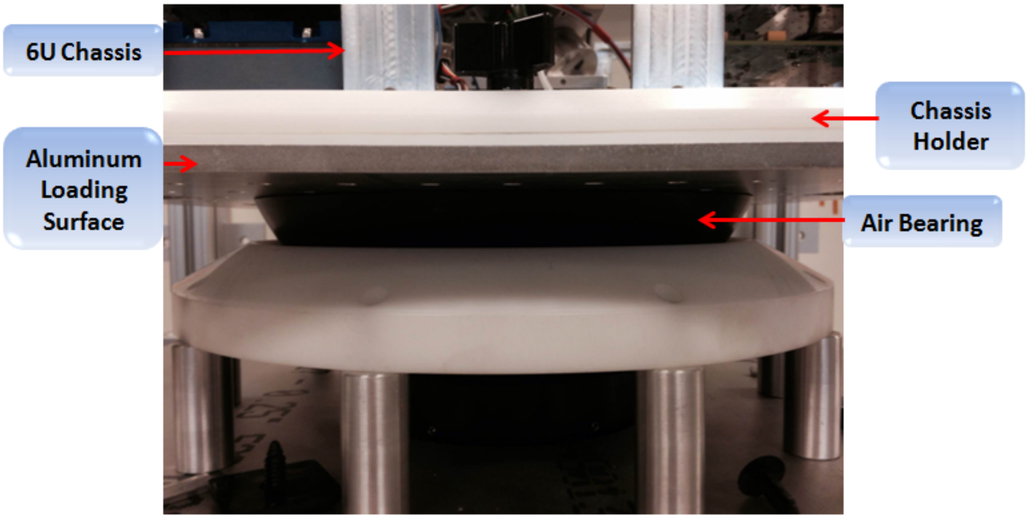


Figure 28. Image Showing Close-up View of the ADCS Test Bed, Loading Surface, and Hemi-Sphere On Air Bearing

With the air bearing on, the user wants to see that the very top of the hemi-sphere attached to the loading surface is level. Once level, the clamps are tightened, securing the ADCS tesbed onto the loading surface. This balancing technique will suffice for this research effort. Another technique not discussed here is the use of weight scales to calculate COM, as seen in Dannemeyer [8]. The ADCS test bed implementation into AFIT's CubeSat test bed will now be discussed.

3.3.5 ADCS Test Bed Implementation Inside AFIT's CubeSat Test Bed

After the chassis is balanced on the loading surface, the ADCS test bed is almost ready for testing. Figure 16 shows the ADCS test bed inside AFIT's CubeSat test bed. The Helmholtz cage is first activated and the desired magnetic field value is established. The ADCS test bed is then programmed as desired. The air compressor is brought to 60 psi, and then the hand crank is used to lower the chassis and loading platform onto the air bearing. Both computers must be connected to the respective CDH boards to receive telemetry and command the CubeSat. The laser, shown in Fig. 24, will be pointing at the angles displayed on the wall which indicate a z-axis angle.

3.3.6 MOI of the Chassis and Loading Surface

In order to properly model the behavior of the ADCS test bed in MATLAB, an accurate measurement of the ADCS test bed MOI is required. Since only z-axis control is of interest for this research, only the z-axis MOI will be measured. Figure 29 shows the device used to calculate the MOI of the ADCS test bed and loading surface. The oscillator rotates the object of interest while the period display outputs the period of the oscillation. The shortest oscillation period corresponds to the object's COM aligned with the oscillator's axis of rotation [8]. Testing becomes an iterative process where the user slightly moves the object until the period is the lowest. Once the lowest period is found, the measurement software calculates the MOI.

The loading platform cannot be sitting on the oscillator because of the hemisphere attached to the bottom. A tare piece was created so that the ADCS test

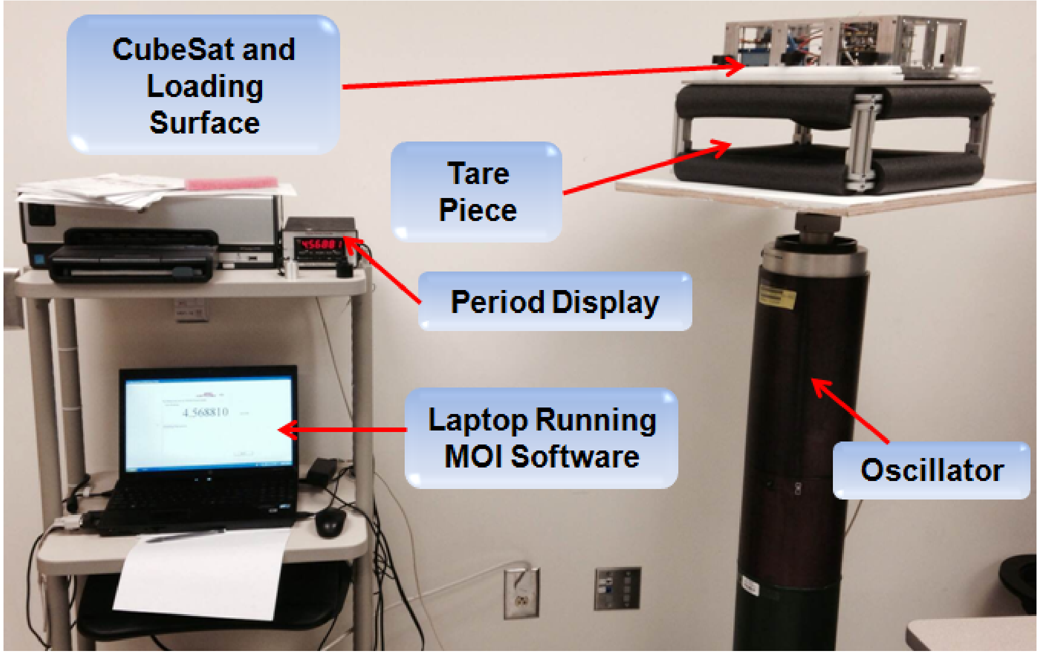


Figure 29. Moment of Inertia Test Bed

bed and loading surface could properly be placed on the oscillator. First the tare piece MOI must be measured. After the tare piece MOI is measured, the ADCS test bed and loading surface are placed on top and the MOI is re-measured. Taking the difference between the two values results in the MOI of the ADCS test bed and loading surface. The calculated MOI of the ADCS test bed and loading surface found by the measurement software is 0.1593157 kg-m^2. This MOI is used in the MATLAB model to produce realistic simulations. Note that the measured MOI will typically be much larger than an actual 6U CubeSat due to the inclusion of the loading surface in the measurement. This increase in MOI from the loading surface, however, can be assumed to offset the lack of a payload in the ADCS test bed. Once the MATLAB simulation matches the ADCS test bed behavior, the MOI of the 6U CubeSat including the payload can be implemented to predict 6U CubeSat behavior in orbit. The software configuration of the ADCS test bed will now be discussed.

3.4 Software Configuration

Before this research started, update rates on the ADCS circuit board could not exceed 1 Hz without overflowing the board's memory. Presently, the estimation and control algorithm run at 10 Hz with an eventual goal of 40 Hz. The following subsections will detail the flow of the ADCS algorithm and how telemetry is streamed to the computer for analysis.

3.4.1 ADCS Algorithm

The ADCS algorithm originally ran as two separate tasks in the FreeRTOS operating system and at two separate rates. This was changed so that the estimation and control algorithms are completed in one task. The overall flow of the ADCS algorithm is outlined in Fig. 30.

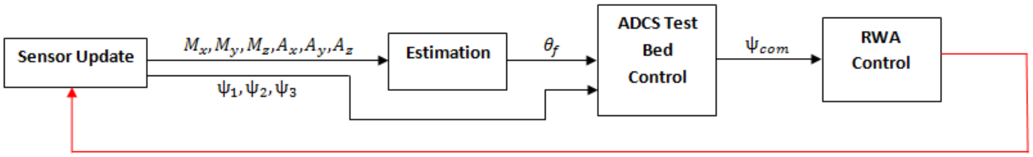


Figure 30. ADCS Main Flow Chart

The sensor update subroutine first collects the magnetometer (M_x, M_y, M_z) and accelerometer (A_x, A_y, A_z) output from the IMU. The reaction wheel speeds (ψ_1, ψ_2, ψ_3) are also measured from the reaction wheel's Hall sensors. Notice that the gyroscope velocity estimations are not shown in the flow chart because they are not used in either the estimation or control algorithms. The gyroscope measurements were not used in estimation because of gyroscopic drift when integrating the angular velocity measurement. For control, the angular velocity error is calculated via a discrete derivative from the attitude error.

The estimation algorithm discussed in Section 3.4.2 uses the magnetometer and accelerometer data to output an estimated z-axis angle θ_f of the ADCS test bed with respect to the inertial reference frame (Fig. 4). The ADCS test bed control algorithm discussed in Section 3.4.3 uses θ_f to compute a control torque based on attitude error which results in a commanded wheel speed ψ_{com} for the reaction wheels. The RWA algorithm discussed in Section 3.4.4 converts the commanded RPM to a specific duty cycle. The red line in Fig. 30 indicates that the entire process is an autonomous feedback loop. As soon as the process is complete, the sensor re-updates the sensor outputs and runs through the algorithm again. The estimation algorithm will be discussed in detail next.

3.4.2 Estimation Algorithm

The estimation algorithm detailed in Fig. 31 shows how the magnetometer is calibrated and filtered.

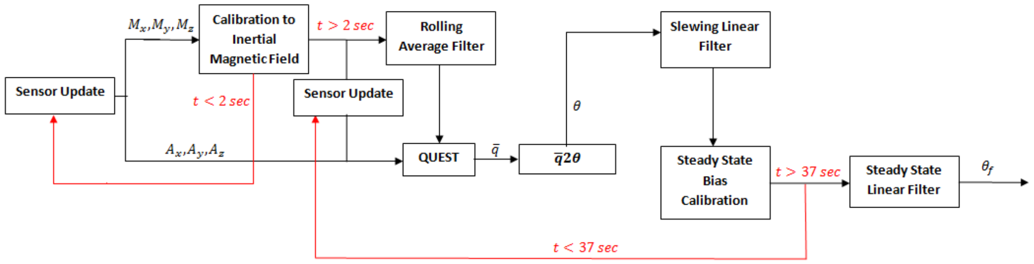


Figure 31. ADCS Estimation Flow Chart

The magnetometer values from the sensor update command initiate with a calibration algorithm as shown in the top left corner of Fig. 31. The calibration algorithm takes in magnetometer data for two seconds. During this time it is taking the average values of the magnetic field outputs. After two seconds, the measured magnetic field values in the three directions are subtracted from the known magnetic field set inside the Helmholtz cage; this becomes the calibration bias for the magnetic field. That bias is then subtracted off of the magnetic field values from the IMU so that the inertial and body frame magnetic field vectors match. After calibration, the magnetic field outputs are inputted into a moving (or rolling) average filter to smooth out noise. Refer to Section 2.4.1 for an explanation on smoothing data.

After the magnetometer output from the rolling average, the QUEST algorithm takes the magnetometer and accelerometer outputs and produces a current quaternion $\bar{q}$. For a description on the QUEST algorithm, refer to Section 2.4.2. Converting the current quaternion to an angle in the z-axis is

$$\theta = -\sin^{-1}(2\ q_1\ q_2 - 2\ q_3\ q_4) \tag{59}$$

Two linear filters are then used on θ: a slewing linear filter and a steady-state linear filter. The slewing linear filter uses a slope correction and no bias correction. The steady-state bias calibration takes into account the distortion of the magnetic field when the reaction wheels run at different speeds. This steady-state bias calibration algorithm will run for 35 seconds so that it can determine the average bias of the z-axis angle with the reaction wheels on. At 37 seconds, the estimation algorithm ceases to use the steady-state bias calibration algorithm, and it becomes a constant for the steady-state linear filter, which applies a slope term to the calibrated bias and results in the filtered angle θ_f. These filters will be explained in Section 4.2.

3.4.3 ADCS Test Bed Control Algorithm

The ADCS test bed control algorithm uses a PID controller to send commanded speeds to the RWA controller. For an overview on PID controllers, see Section 2.5.4. As seen from Fig. 32, θ_f is subtracted from the desired angle θ_D to determine the angle error θ_e.

The angle error is then split three different ways. One way multiplies it by the proportional gain K_p. The bottom line subtracts the old angle error from the current

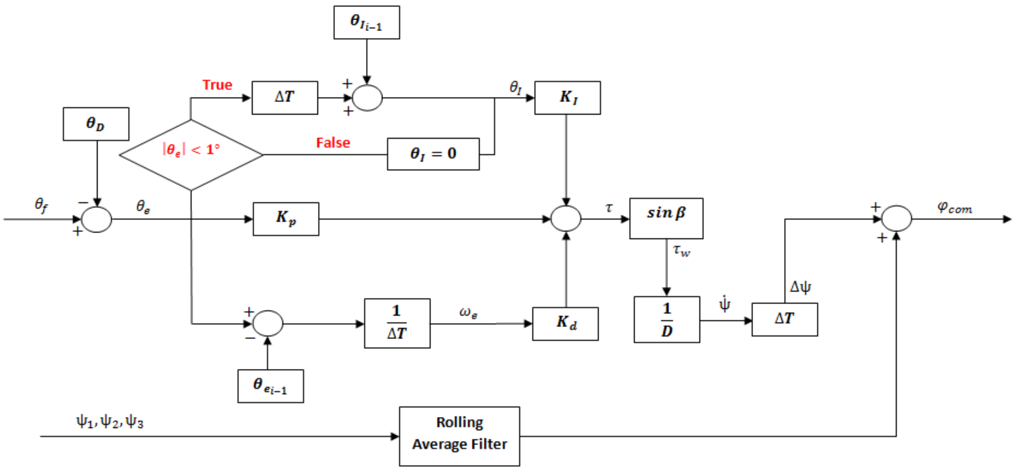


Figure 32. ADCS CubeSat Control Flow Chart

angle error and divides by the time step. This results in the angular velocity error ω_e which is then multiplied by the derivative gain K_d. The upper line uses an if statement to check if the absolute value of the angle error is within one degree. If it is not, then the integral error θ_I is zero. If it is within one degree, θ_e is multiplied by the time step and then added to the previous integral error. This creates a band around the desired angle so that integral windup is minimized. Note that this band makes the PID controller for the ADCS test bed non-linear. This integral error is then multiplied by the integral gain K_I. Summing these three terms together produces the desired torque output from the PID controller. Analysis of chosen gains on this PID controller are discussed in Section 4.7.

As discussed in Section 3.2.2, the torque output from the PID must be multiplied by $\cos\beta$ to produce the commanded reaction wheel torque. Although the torque is known, the reaction wheel controller needs a commanded wheel speed in RPM. To do this the torque of the wheel is multiplied by the reciprocal of the wheel MOI D to obtain $\dot{\psi}$. Discretizing $\dot{\psi}$ requires multiplication by the time step which will result in $\Delta\psi$ which is added to the current wheel speed in RPM of the first reaction wheel. This becomes the commanded speed ψ_{com} that is sent to the reaction wheel controller.

Note that typically $\Delta\psi$ is added to each of the individual current wheel speeds. Since the wheels do not all act the same as discussed in Section 4.5, the decision was made to give the exact same command to all the reaction wheels which is dictated only by the speed of reaction wheel one. The RWA control algorithm will now be discussed.

3.4.4 RWA Control Algorithm

The reaction wheel control algorithm only has a proportional gain. The wheel speed is first converted to a duty cycle count and then passed through the proportional controller. Discussions on duty cycle calibration are found in Section 3.5.3. Figure 33 shows the reaction wheel control algorithm for an individual reaction wheel.

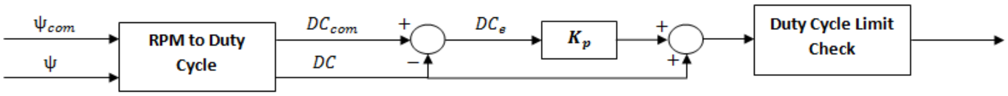


Figure 33. ADCS RWA Control Flow Chart

As soon as the current wheel speed ψ and commanded wheel speed ψ_{com} enter the algorithm, both are converted to a corresponding duty cycle (DC and DC_{com}) in the "RPM to Duty Cycle" block shown in Fig. 33. Once the commanded and current duty cycles are known, duty cycle error DC_e is calculated from their subtraction. Duty cycle error is then multiplied by the proportional gain and added to the current duty cycle. This value then goes through a duty cycle limit check to make sure it does not go outside certain bounds. These bounds and how they were calculated are discussed in Section 4.5. This command will go to the reaction wheels, causing movement of the reaction wheels and ADCS test bed. Telemetry and algorithm speed is discussed in the next section.

3.4.5 Algorithm Speed and Telemetry

Although the ADCS algorithm is set to run at 10 Hz, the algorithm speed needs to be measured so that the total computation time of the algorithm may be used in a MATLAB model. Time is estimated in the output from the ADCS test bed by adding a counter at the end of the algorithm. This will increase in number by one each revolution of the algorithm. Dividing this number by 10 ensures that the time output is in 0.1 second increments which corresponds to 10 Hz. Measuring the actual speed of the algorithm was performed by using the USBeeZX logic analyzer shown in Fig. 34. Each duty cycle takes 99.5913 ms where the determination and control algorithms take 29.1312 ms leaving a 70.4601 ms wait state as shown in Fig. 35.

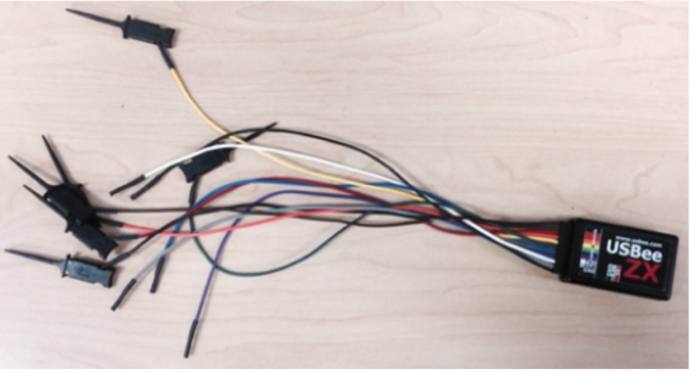


Figure 34. USBeeZX Measurement Device

The algorithm, running at 10.041 Hz, is faster than the commanded update rate of 10 Hz. With this value being known, MATLAB modeling can be corrected to update at 10.041 Hz instead of 10 Hz. Furthermore, the time output from data collection can be corrected by using the actual update rate to obtain more accurate analysis. The process of gathering data from the ADCS test bed is important for analysis of behavior and matching data with simulation. For these reasons, an explanation on how telemetry is gathered and handled will be needed. When the ADCS algorithm completes a revolution, desired data is sent to the CDH board via a wired connection.

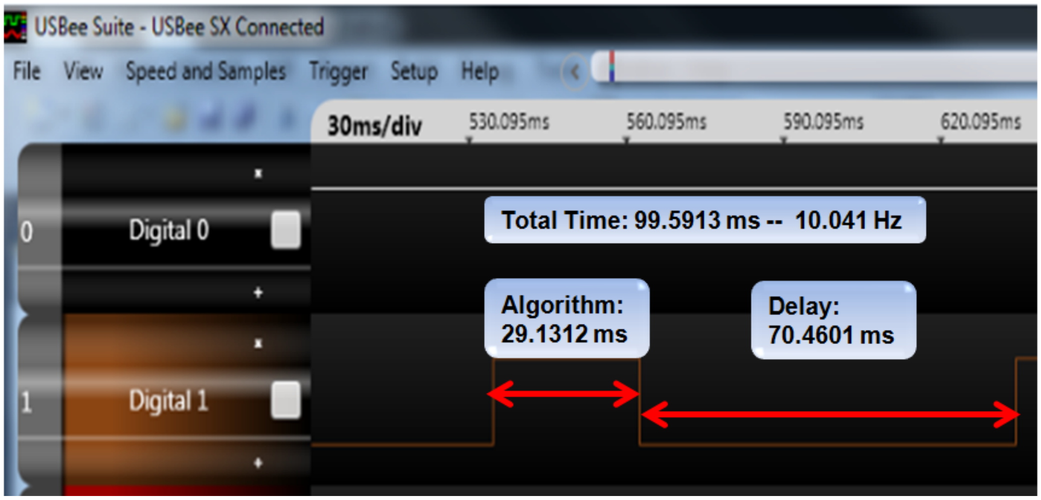


Figure 35. USBeeZX Software Output of ADCS Algorithm Speed

The on-board WiFly from the CDH then wirelessly sends data into a software program called Tera Term on the computer as shown in Fig. 36.

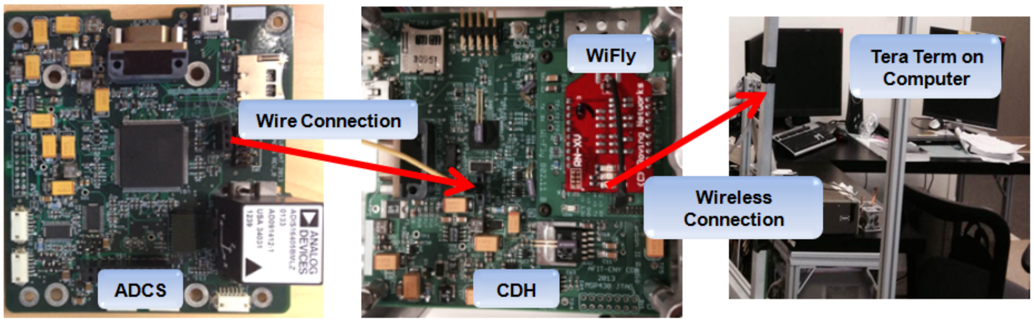
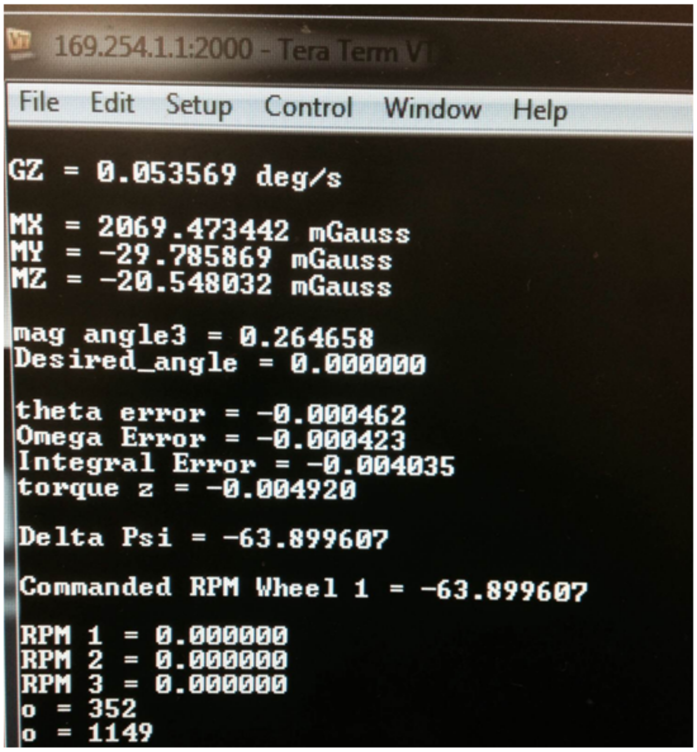


Figure 36. Telemetry Gathering From ADCS to Computer

The software inside the ADCS is set up so Tera Term displays the data in two separate ways. The first way, shown in Fig. 37, demonstrates visual aid to assess data quickly, but not necessarily record data. Figure 37 shows that the description of the number is on the left while the data value is on the right.

This output is generally used to validate a desired test before taking data. It allows easy visual comparisons of estimated values to truth values. Once the desired test has been validated in this format, the actual test will be run with another Tera Term setup used to acquire data shown in Figure 38.



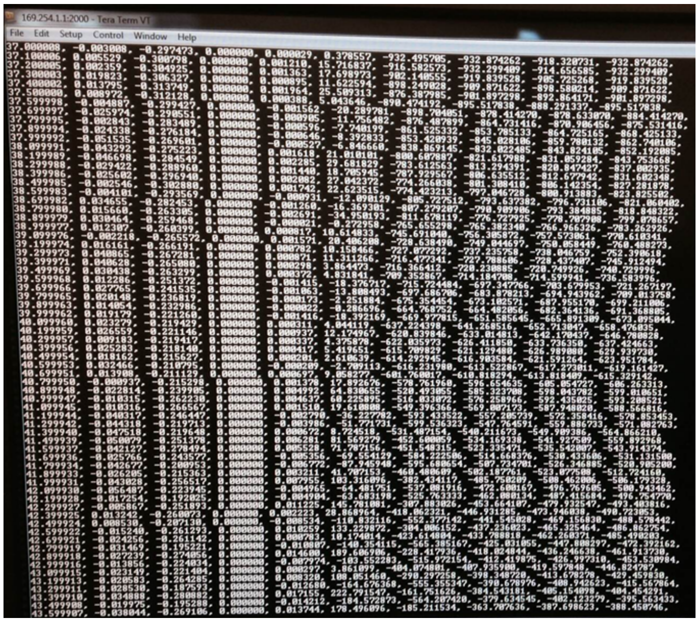
```
GZ = 0.053569 deg/s

MX = 2069.473442 mGauss
MY = -29.785869 mGauss
MZ = -20.548032 mGauss

mag angle3 = 0.264658
Desired_angle = 0.000000

theta error = -0.000462
Omega Error = -0.000423
Integral Error = -0.004035
torque z = -0.004920

Delta Psi = -63.899607

Commanded RPM Wheel 1 = -63.899607

RPM 1 = 0.000000
RPM 2 = 0.000000
RPM 3 = 0.000000
o = 352
o = 1149
```

Figure 37. Tera Term Data Output: No Data Recording

This Tera Term output is not recommended for demonstration to others since the numbers shown on the screen are not clearly defined. Once the test is complete, the Tera Term file is converted to a .csv file. The data in this file is then imported into MATLAB for analysis and graphical display. The procedures for the tests to be performed on the ADCS test bed inside AFIT's CubeSat test bed will be discussed next.

3.5 Testing Procedures

The following section will detail each specific test to be performed on the ADCS test bed in this research effort. The first two subsections outline the testing procedures for the inertial and external sensors. The second set of tests outline the RWA duty cycle calibration, modeling, and control test procedures. The sensor noise calculation procedures will be outlined in the following subsection. The final subsections outline



```
37.000008, -0.003008, -0.297473, 0.000000, 0.000029, 0.378557, -932.495705, -932.874262, -918.320731, -932.874262,
37.100006, 0.005529, -0.300798, 0.000000, 0.001210, 15.716837, -916.582573, -932.299409, -914.656585, -932.299409,
37.200005, 0.002359, -0.304925, 0.000000, 0.001363, 17.698973, -902.140555, -919.839528, -905.725268, -919.839528,
37.300003, 0.019823, -0.306327, 0.000000, 0.000895, 11.622594, -898.249028, -909.871622, -898.580214, -909.871622,
37.400002, 0.013795, -0.313749, 0.000000, 0.001964, 25.509313, -876.387985, -901.897298, -892.864171, -901.897298,
37.500000, 0.008972, -0.312128, 0.000000, 0.000388, 5.043646, -890.474192, -895.517838, -888.291337, -895.517838,
37.599998, -0.004887, -0.299427, 0.000000, -0.001562, -20.289781, -898.704051, -878.414270, -878.633070, -884.414270,
37.699997, -0.025974, -0.290551, 0.000000, -0.000905, -11.756482, -876.487899, -864.731416, -870.906456, -875.531416,
37.799995, -0.002843, -0.283489, 0.000000, -0.000596, -7.740199, -861.525332, -853.785133, -864.725165, -868.425133,
37.899994, -0.024338, -0.276184, 0.000000, -0.000646, -8.392833, -853.420939, -845.028106, -859.780132, -862.740106,
37.999992, -0.051535, -0.269601, 0.000000, -0.000527, -6.846660, -838.869145, -832.022485, -843.824105, -852.192085,
38.099991, -0.043292, -0.275223, 0.000000, 0.001618, 21.010101, -800.607887, -821.617988, -831.059284, -843.753668,
38.199989, -0.046698, -0.284549, 0.000000, 0.002285, 29.681829, -783.612563, -813.294391, -820.847427, -837.002935,
38.299988, -0.029422, -0.288960, 0.000000, 0.001440, 18.705945, -787.929567, -806.635512, -812.677942, -831.602348,
38.399986, -0.025602, -0.296940, 0.000000, 0.002082, 27.042372, -774.266038, -801.308410, -806.142354, -827.281878,
38.499985, -0.002546, -0.302880, 0.000000, 0.001742, 22.623515, -774.423213, -797.046728, -800.913883, -823.825502,
38.599983, -0.004101, -0.293569, 0.000000, -0.000931, -12.090129, -805.727512, -793.637382, -796.731106, -821.060402,
38.699982, 0.034655, -0.282453, 0.000000, -0.001260, -16.369301, -807.279207, -790.909906, -793.384885, -818.848322,
38.799980, 0.015660, -0.263305, 0.000000, -0.002691, -34.950193, -811.678119, -776.727925, -778.707908, -799.078657,
38.899979, 0.030464, -0.259468, 0.000000, -0.000021, -0.272958, -765.655298, -765.382340, -766.966326, -783.262926,
38.999977, 0.012307, -0.260399, 0.000000, 0.000817, 10.614833, -745.691040, -756.305872, -757.573061, -770.610341,
39.099976, -0.002218, -0.265572, 0.000000, 0.001571, 20.406208, -728.638490, -749.044697, -750.058449, -760.488273,
39.199974, 0.016161, -0.266191, 0.000000, 0.000782, 10.158466, -727.077292, -737.235758, -738.046759, -752.390618,
39.299973, 0.040865, -0.267281, 0.000000, 0.000871, 11.311266, -716.477341, -727.788606, -728.437407, -745.912494,
39.399971, 0.040628, -0.265080, 0.000000, 0.000298, 3.864473, -716.366412, -720.230885, -720.749926, -740.729996,
39.499969, 0.030438, -0.263299, 0.000000, 0.000372, 4.831841, -709.352867, -714.184708, -714.599941, -736.583996,
39.599968, 0.012286, -0.251372, 0.000000, -0.001415, -18.376717, -715.724484, -697.347766, -703.679953, -721.267197,
39.699966, 0.027765, -0.241535, 0.000000, -0.001063, -13.808493, -697.686706, -683.878213, -694.943962, -709.013758,
39.799965, 0.020148, -0.236819, 0.000000, -0.000173, -2.251884, -675.354454, -673.102571, -687.955170, -699.211006,
39.899963, 0.014054, -0.232986, 0.000000, -0.000022, -0.286696, -664.768752, -664.482056, -682.364136, -691.368805,
39.999962, 0.019179, -0.221260, 0.000000, -0.001416, -18.392169, -669.977815, -651.585645, -665.891309, -673.095044,
40.099960, 0.023279, -0.219429, 0.000000, 0.000311, 4.044119, -637.224398, -641.268516, -652.713047, -658.476035,
40.199959, 0.026559, -0.221146, 0.000000, 0.000937, 12.174967, -620.839846, -633.014813, -642.170437, -646.780828,
40.299957, 0.009183, -0.219417, 0.000000, 0.000337, 4.375878, -622.035973, -626.411850, -633.736350, -637.424663,
40.399956, 0.025283, -0.218140, 0.000000, 0.000417, 5.419651, -615.709829, -621.129480, -626.989080, -629.939730,
40.499954, 0.018162, -0.215627, 0.000000, 0.000201, 2.610639, -614.292945, -616.903584, -621.591264, -623.951784,
40.599953, 0.032466, -0.210795, 0.000000, -0.000209, -2.709113, -616.231980, -613.522867, -617.273011, -619.161427,
40.699951, 0.013908, -0.211129, 0.000000, 0.000697, 9.058277, -601.760017, -610.818294, -613.818409, -615.329142,
40.799950, -0.000937, -0.215298, 0.000000, 0.001378, 17.892676, -578.761960, -596.654635, -605.054727, -606.263313,
40.899948, -0.022814, -0.239905, 0.000000, 0.004992, 64.830655, -520.493055, -585.323708, -598.043702, -599.010651,
40.999947, -0.010315, -0.262230, 0.000000, 0.004637, 60.225895, -516.033072, -576.258966, -592.435025, -593.208521,
41.099945, -0.010316, -0.266567, 0.000000, 0.001510, 19.610808, -549.396366, -569.007173, -587.948020, -588.566816,
41.199944, -0.040317, -0.246447, 0.000000, -0.002790, -36.227782, -593.433522, -557.205739, -572.358416, -578.853453,
41.299942, -0.044318, -0.219713, 0.000000, -0.003986, -51.771731, -599.536323, -547.764591, -559.886733, -571.082763,
41.399940, -0.047518, -0.219648, 0.000000, 0.000672, 8.724518, -531.487154, -540.211673, -549.909386, -564.866210,
41.499939, -0.050079, -0.251378, 0.000000, 0.006281, 81.569279, -452.600059, -534.169338, -541.927509, -559.892968,
41.599937, -0.042127, -0.278494, 0.000000, 0.005528, 71.787156, -451.548315, -523.335471, -535.542007, -543.914374,
41.699936, -0.025766, -0.243577, 0.000000, -0.005356, -69.554129, -584.222506, -514.668376, -530.433606, -531.131500,
41.799934, -0.042677, -0.200953, 0.000000, -0.006772, -87.945948, -595.680654, -507.734701, -526.346885, -520.905200,
41.899933, -0.026205, -0.215363, 0.000000, 0.003211, 41.707153, -460.480609, -502.187761, -523.077508, -512.724160,
41.999931, -0.043028, -0.256517, 0.000000, 0.007955, 103.316096, -382.434117, -485.750209, -508.462006, -506.179328,
42.099930, -0.056487, -0.223945, 0.000000, -0.004965, -64.481751, -537.081916, -472.600167, -496.769605, -500.943462,
42.199928, -0.017254, -0.192025, 0.000000, -0.004904, -63.683198, -525.763333, -462.080134, -487.415684, -496.754770,
42.299927, -0.005867, -0.251893, 0.000000, 0.011225, 145.781081, -307.883026, -453.664107, -479.932547, -493.403816,
42.399925, 0.013242, -0.260073, 0.000000, 0.002223, 28.868964, -418.062321, -446.931285, -473.946038, -490.723053,
42.499924, 0.008530, -0.207130, 0.000000, -0.008534, -110.832116, -552.377142, -441.545028, -469.156830, -488.578442,
42.599922, -0.015240, -0.261293, 0.000000, 0.010259, 133.229872, -304.006152, -437.236023, -465.325464, -486.862754,
42.699921, -0.024256, -0.261142, 0.000000, 0.000783, 10.174014, -423.614804, -433.788818, -462.260371, -485.490203,
42.799919, -0.031469, -0.195221, 0.000000, -0.010807, -140.354115, -565.385167, -425.031055, -447.808297, -472.392162,
42.899918, -0.027239, -0.274056, 0.000000, 0.014600, 189.606906, -228.417936, -418.024844, -436.246638, -461.913730,
42.999916, -0.013856, -0.224034, 0.000000, -0.007974, -103.552283, -515.972154, -412.419875, -426.997310, -453.530984,
43.099915, -0.023149, -0.221404, 0.000000, 0.000297, 3.861099, -404.074801, -407.935900, -419.597848, -446.824787,
43.199913, -0.020583, -0.264285, 0.000000, 0.008320, 108.051460, -290.297258, -398.348720, -413.678278, -429.459830,
43.299911, -0.028530, -0.187593, 0.000000, -0.012680, -164.676368, -555.355347, -390.678976, -408.942623, -415.567864,
43.399910, -0.034888, -0.280882, 0.000000, 0.017155, 222.791547, -161.751626, -384.543181, -405.154098, -404.454291,
43.499908, -0.019975, -0.195288, 0.000000, -0.014212, -184.572873, -564.207420, -379.634545, -402.123279, -395.563433,
43.599907, -0.038044, -0.269106, 0.000000, 0.013744, 178.496096, -185.211534, -363.707636, -387.698623, -388.450746,
```

Figure 38. Tera Term Data Output: Data Recording

testing procedures for tuning the ADCS PID controller, single-axis slewing maneuvers, disturbances, calculating the angular momentum and external torque applied to the ADCS test bed, and MATLAB modeling.

3.5.1 IMU Gyroscope Calibration Testing

The gyroscope is usually susceptible to non-zero bias. This is due to the gyroscope's noisy angular velocity output. For this reason, the gyroscope angular velocity is desired, not the integrated *z*-axis angle. Although the angular velocity from the gyroscope is not used in estimation or control of the ADCS test bed, the gyroscope angular velocity data can be used to compare with simulated angular velocity outputs from a MATLAB model. The calculation of the average bias can be performed by pointing the ADCS testbed at zero with the reaction wheels on and air bearing off. Having the reaction wheels on is needed to simulate vibrational noise that will be inherent in the ADCS test bed. This task can be fully automated to be performed before the control and estimation tasks start. The gyroscope output will be measured for 35 seconds, and each point will be summed. After 35 seconds, the sum total of all the angular velocities will be divided by the number of data points recorded. This results in the angular velocity bias at steady state.

The angular velocity tracking accuracy must also be tested to ensure the accuracy of the angular velocity when the ADCS test bed is moving. In the absence of a truth measurement device inside AFIT's CubeSat test bed, testing the gyroscope angular velocity accuracy while moving the ADCS test bed is performed using the PASCO scientific ME-8951 Rotating Platform Base shown in Fig. 39. With the ability to track angular velocity, the PASCO rotation table will be used to validate the gyroscope output. The ADCS test bed will be reduced to the small stack shown in Fig. 40. The rotation table will be pushed and the gyroscope and truth data will be gathered.

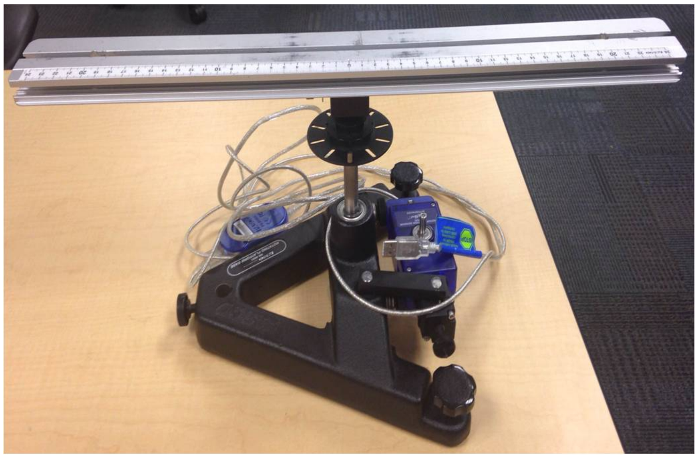


Figure 39. PASCO Rotation Table

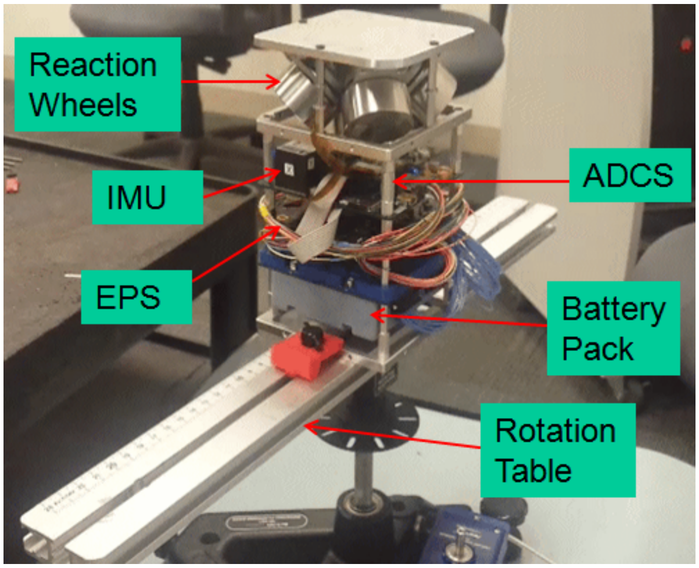


Figure 40. Gyroscope Test Experimental Setup

3.5.2 IMU Magnetometer Calibration and Filtering

Although the Helmholtz cage is capable of mimicking a LEO orbit's magnetic field, a constant magnetic field was chosen for this research effort. Calibrating and filtering the magnetometer to reach performance specifications using a weak, changing magnetic field could potentially be a thesis in and of itself. Putting effort into calibrating the magnetometer in this way was a lower priority than testing the full control capabilities of the reaction wheels. Using a strong, nearly constant magnetic field will cause the magnetometer to outperform its accuracy specifications; allowing for greater control accuracy when commanding the reaction wheels.

3.5.2.1 Choosing the Magnetic Field

Choosing the magnetic field came down to two parameters: strength and complexity. As mentioned in Section 3.1.1, the Helmholtz cage's strongest field in any direction is 2000 mG. If we assume that the magnetometer sensor frame is coincident with the ADCS test bed body frame, then Eq. (25) can be rewritten as

$$\vec{M}_b = R^{bi}\vec{M}_i \tag{60}$$

where $\vec{M}_b$ is the magnetic field vector in the body frame, R^{bi} is a 3-2-1 rotation matrix, and $\vec{M}_i$ is the magnetic field vector in the inertial frame. Although there are minor variations of the magnetic field inside the Helmholtz cage, this research effort assumes that it is constant. If the inertial magnetic field is constant, then the body frame magnetic field vector is easily calculated if the angle is known. For a review on rotation matrices, see Section 2.1.2.

The first thought for the the desired magnetic field was to maximize all magnetic field directions to 2000 mG. Figure 41 demonstrates the simulated change in the magnetic field when moving from -50^o to 50^o. Note that the magnetic field value in the z-axis does not change for a z-axis maneuver. This field change is undesirable due to the fact that readings would exceed 2000 mG in the x- and y-directions. Furthermore, this output over a short period causes the magnetic field to become less constant due to overuse of the Helmholtz cage. Eventually, the current cannot be raised any further to reach the same magnetic field value. For this reason, it was decided to reduce the z-axis magnetic field to zero. Setting either the x- or y-axes magnetometer does not allow for the body axis readings to go above 2000 mG as shown in Figure 42.

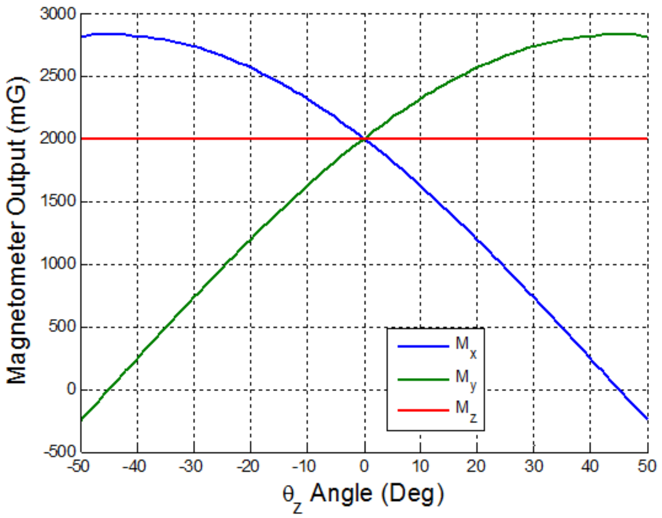


Figure 41. Magnetometer Change in Body Frame with Max Inertial Magnetic Field

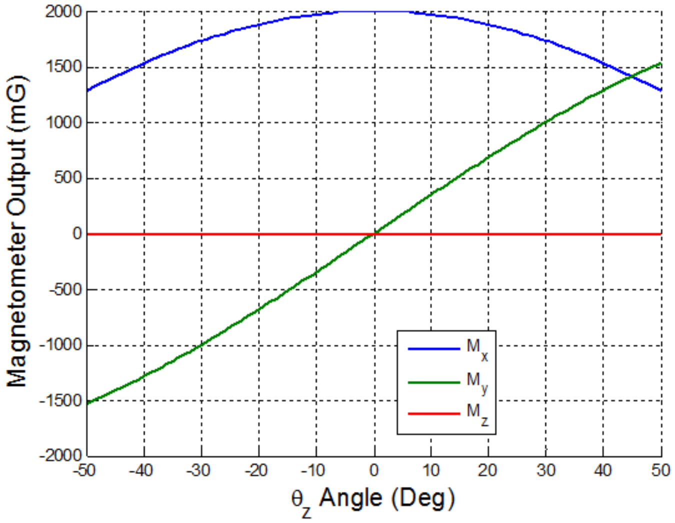


Figure 42. Chosen Magnetic Field for Testing

Note that the truth magnetometer inside the Helmholtz cage is 90^o rotated from the IMU inside the ADCS test bed. This means that the Helmholtz cage y-axis is the ADCS test bed x-axis and vice versa. So when setting the Helmholtz cage magnetic field, the sensor should report 2000 mG in the y-axis and 0 mG in the x- and z-axes. Although a stronger magnetic field will make the magnetometer more accurate in its estimation, it increases undesirable external torques. The amount of external torque applied to the ADCS test bed will be explored in Section 4.8.

3.5.2.2 Initial Calibration of the Magnetometer

The first step in obtaining accurate information from the magnetometer is to calibrate the IMU magnetic field output before starting any estimation/control algorithms. With the air bearing off and the ADCS test bed pointing at 0^o, the magnetic field will be calculated for a certain period of time and averaged. Once averaging is complete, the magnetic field calculated by the IMU will be subtracted by the known inertial magnetic field setup inside the Helmholtz cage as

$$\vec{M}_{bias} = \vec{M}_b - \vec{M}_i \tag{61}$$

where $\vec{M}_{bias}$ is the magnetic field bias correction. This term will then be subtracted by every magnetic field output from the IMU magnetometer as shown in

$$\vec{M}_{cal} = \vec{M}_b - \vec{M}_{bias} \tag{62}$$

where $\vec{M}_{cal}$ is the calibrated magnetic field from the IMU.

3.5.2.3 Filtering

A filter between $\pm 50^o$ will be performed on the attitude estimate. Using the truth measurement angles displayed the wall (Fig. 16), the sensor angle and magnetometer measurements will be recorded at each angle. Note that truth measurement for this test is performed by visually checking that the laser is pointed to the correct angle. It does not matter how long the ADCS test bed is at each angle since the test is time independent. The sensor angle and magnetometer values will be compared to simulation to test for their error. A linear filter will then be created based on the sensor angle measurement as shown in

$$\theta_{filtered} = \theta_{sensor} + m\theta_{sensor} \tag{63}$$

where $\theta_{filtered}$ is the filtered angle, θ_{sensor} is the unfiltered sensor angle, and m is the calculated slope m. Note that Eq. (63) has no bias term. This is due to the fact that the magnetometer angle has already been calibrated to zero.

Finally, calculating error angle versus magnetic field error is useful to correlate the two outputs. By knowing the respective magnetometer error, the angle error can be estimated. This can easily be calculated by subtracting the simulated values versus the filtered and unfiltered sensor outputs.

3.5.2.4 Steady-State Filtering

Steady state accuracy is very important for CubeSat operations. To see how well the control system can point at its target, good estimation is a necessity. As noted in Section 3.2.2, the RWA distorts the magnetic field when the reaction wheels are spinning. Even though the magnetic field was calibrated originally, it does not take into account distortions of the magnetic field caused by the wheels. The solution chosen for this problem, though not ideal, is that the reaction wheels will be commanded to move in a saw tooth pattern as shown in Fig. 43. Note that this test will occur with the air bearing off. The reaction wheels will start at zero and speed to 5000 RPM. The wheels will then slow to -5000 RPM before speeding back up to a stop at 0 RPM. The sawtooth command is superior to a sinusoid command because it is theoretically at each RPM for an equal amount of time. Note that the test case time for this is 2000 seconds.

A linear filter is created to reduce angle error for the range of wheel speeds tested. This bias term for the filter will be automated due to unpredictable changes in the

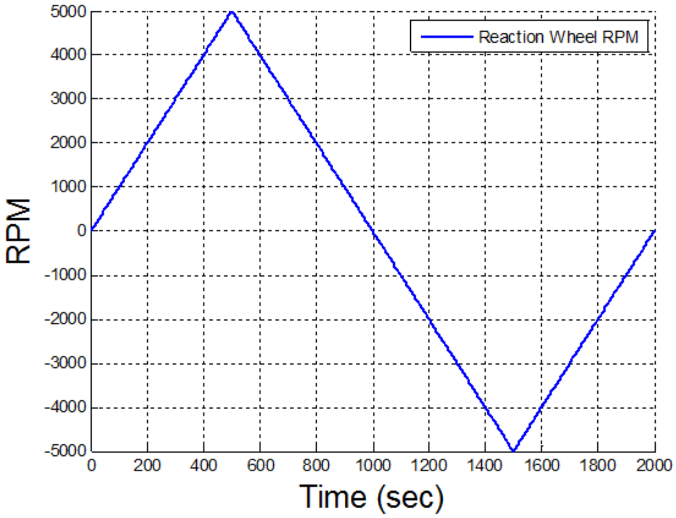


Figure 43. Sawtooth Command for Reaction Wheels

bias similar to the gyroscope bias calibration in Section 3.5.1. This filter allowed for a changing bias correction factor to the angle as shown in

$$\theta_{final} = \theta_{filtered} - \theta_{ss} \tag{64}$$

where θ_{final} is the final angle output of the estimation algorithm that is sent to the control algorithm and θ_{ss} is the changing steady-state calibration. The next subsection will discuss the RWA duty cycle calibration.

3.5.3 RWA Duty Cycle Calibration Testing

To ensure that reaction wheels will be properly commanded when handed reaction wheel speed, there must be proper conversion between reaction wheel speed and duty cycle. A duty cycle corresponds to the width of a pulse sent by the motors to the reaction wheels. Figure 44 demonstrates how the width of a pulse changes with duty cycle.

Note that the lines are at different amplitudes only for illustrative purposes. Duty cycle, as it is programmed on the ADCS, works just as shown in Fig. 44. A duty

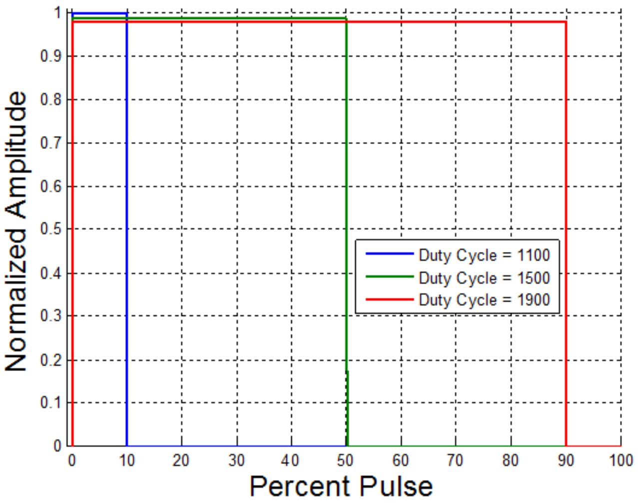


Figure 44. Duty Cycle Related to Pulse Width Modulation

cycle of 1100 corresponds to 10 percent of a pulse at every time step. It can be seen that 1500 duty cycle corresponds to a 50 percent pulse every time step. When the duty cycle reaches 2000, the response ceases to be a pulse and becomes a constant line at the respective amplitude. This means that a duty cycle of 1000 corresponds to a reaction wheel speed of 0 RPM. Duty cycles below 1000 relate the same way as previously mentioned except the wheels would be spinning in the opposite direction. This means that a duty cycle count of 0 is equal in magnitude and opposite in direction for a duty cycle count of 2000.

Typically, there is a linear relationship between reaction wheel speed and duty cycle. Although it will not perfectly match the exact duty cycle count for a specified wheel speed, it will serve as a good approximation. Testing the duty cycle will be performed by starting the respective reaction wheel at a duty cycle of 1000. Every ten seconds the duty cycle will increase or decrease in count by one until it has reached 2000 or 0. Every ten seconds to increase the count has been chosen so that the wheels have enough time to reach a steady-state speed. Note that there is no control on the wheels; the duty cycle is simply changed, and the wheel speed is recorded. Linear fits to the data will be made and implemented into the software. Testing procedures on RWA time constants will be discussed next.

3.5.4 RWA Time Constant Testing

Testing the time constants of the reaction wheels assumes that the reaction wheels will behave like a first-order system when given an input and no feedback control. A first-order system can be modeled as

$$T\dot{x} + x = u \tag{65}$$

where x is the state, $\dot{x}$ is the state derivative, u is the input, and T is the time constant. By taking the Laplace transform of this differential equation, the first-order system is

$$X(s) = \frac{U(s)}{Ts + 1} \tag{66}$$

where $X(s)$ and $U(s)$ are the Laplace domain output and input, respectively. Solving for the time response of this system leads to the general solution

$$X(t) = u(1 - e^{\frac{-t}{T}}) \tag{67}$$

where t is time and is greater than zero [7]. According to Eq. (67), as t goes to infinity, the output, $X(t)$, equals the input, u. Figure 45 shows the time response of a first-order system as well as an illustration on approximating the time constant.

Figure 45 represents a case where the system starts with zero initial conditions and goes to the input value of one. Note that $c(t)$ is equivalent to $X(t)$ from Eq. (67). The time constant is approximated by finding the time it takes for the output to reach 63.2 percent of the input. This time constant can be further validated by checking if

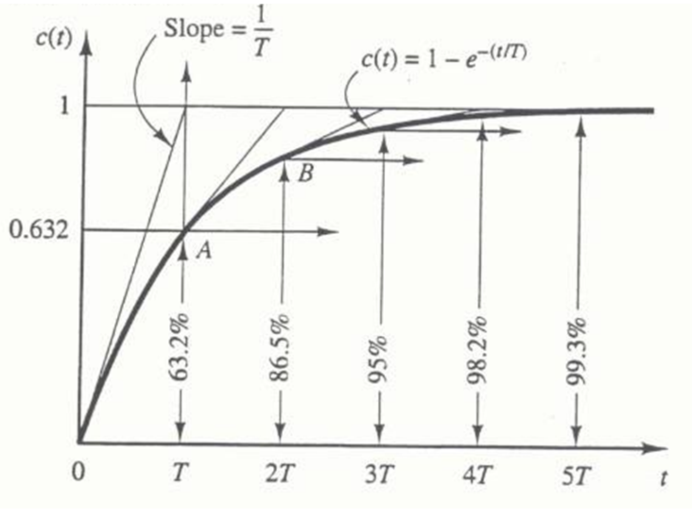


Figure 45. Time Constant Approximation of a First-Order System [7]

it has doubled when the output is 86.5 percent of the input. The faster the output reaches the input, the lower the time constant will be for a first-order system. This is illustrated by Fig. 46. The amount of time it takes to reach the desired input is decreased when the time constant is lowered. The time constants of the system are inherent in the reaction wheel system and cannot be changed. The only way to speed system response is to increase the input with a proportional gain as discussed in the next subsection.

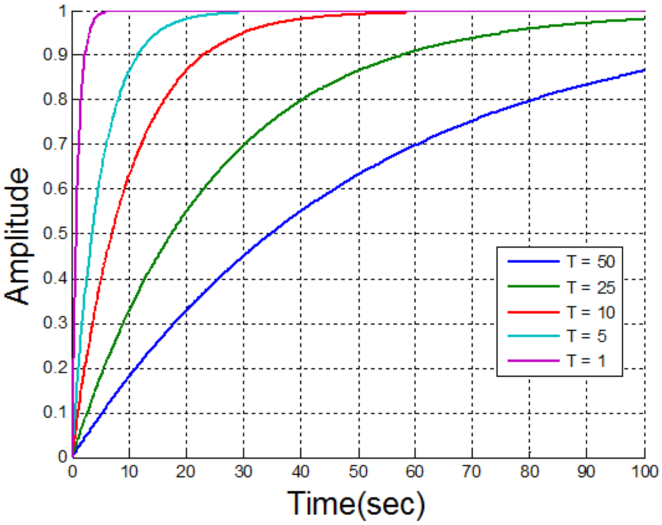


Figure 46. Simulated Effects of Time Constant on System Response

Calculating the time constant for a large number of reaction wheel speed inputs is important for two reasons. First, the time constants must be well documented for implementation into an analytical model as a look-up table. Second, consistent time constants of the reaction wheels for a wide variety of inputs is important in defining how linear the system is. If the time constants are vastly different for different inputs, then the system has non-linearities. This is reason enough to limit the maximum and minimum duty cycle counts the wheels are allowed to achieve to ensure some linearity.

The testing will be based on a starting and ending wheel speed (in RPM) without the proportional controller. Initial tests have the wheels start at 0 RPM and ramp to 500 RPM, 1000 RPM, and then by 1000 RPM increments to 6000 RPM. The same test will then be performed for the wheels spinning in the opposite direction. The wheel's starting speed will change to 1000 RPM and the tests will be performed again up until 6000 RPM. The starting wheel speed will then be 2000 RPM and incrementally climb to 6000 RPM. This pattern will continue until the starting wheel speed is 5000 RPM. These tests also apply for the negative case. The next set of tests will have the wheels start at 6000 RPM and decrease to 0 RPM in 1000 RPM increments. The starting speed will change by increments of 1000 RPM up until 1000 RPM, and then 500 RPM will be the starting speed. The tests will be repeated with the final wheel speed increasing in 1000 RPM increments goes to 6000 RPM. The negative wheel speed tests will be performed the same way. In total, 84 time constants will be calculated for each reaction wheel.

3.5.5 RWA Controller Testing

After the duty cycle is calibrated, tuning the PI RWA controller must be performed. Two different tests will be performed, one with only a proportional gain and the other with a proportional and integral gain. A simulated example of how the responses vary with and without the integral gain is illustrated in Fig. 47.

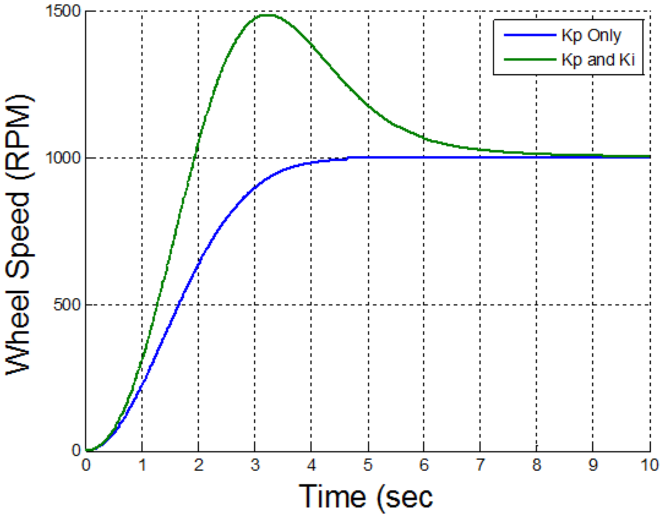


Figure 47. Simulated Effects of Integral Gain on Closed Loop System Response

Figure 47 shows a simulated wheel speeding from 0 to 1000 RPM with and without an integral term. The addition of only a proportional gain will keep the traditional first-order response whereas adding an integral term will cause overshoot. An integral term may be needed if there is excessive steady-state error with proportional gain only.

Increasing only K_p should result in faster rise times of the system with no overshoot. With the addition of just a proportional term, the open loop system in the Laplace domain becomes

$$X(s) = K_p \frac{U(s)}{Ts+1} \tag{68}$$

where implementing the proportional gain K_p simply scales the desired input on the system; this results in faster system response when closed-loop control is enacted as shown in Fig. 48.

The higher K_p is raised, the faster the system response; the system response speed increases less rapidly as K_p increases. Increasing K_i will also increase system response,

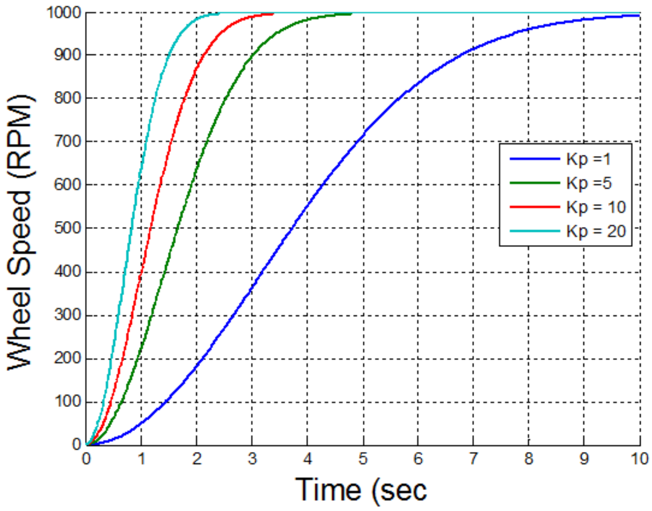


Figure 48. Simulated Effects of Increasing Proportional Gain on Closed Loop System Response

but the more it is increased, the more the system will overshoot the desired input. This means that the integral gain is typically much lower than the proportional gain for reaction wheels.

Although the integral term will be tested, the overshoot that corresponds with its use is not desired. Larger overshoot means larger settling times. If the settling time is larger for the reaction wheels, the settling time will also be larger for the ADCS testbed. If just a proportional term results in low steady-state error and acceptable rise times, then an integral term will not be needed.

3.5.6 Sensor Noise Testing

Calculating sensor noise is important for two reasons. First, characterizing sensor noise aids in calibrating and filtering for better estimation accuracy. Second, noise modeling is important in implementing accurate sensors in simulation.

Two tests were conducted on the sensor outputs from the IMU. The states consist of the z-axis angle output from the magnetometer, the z-axis angular velocity from the gyroscope, and the reaction wheel speeds from the tachometer. The first test will output the states without the use of a rolling average filter while the second test will output the states with the use of a rolling average filter. The rolling average is discussed in Section 2.4.1. The ADCS testbed will not move while pointing at zero while the reaction wheels will spin to 1000 RPM. The means and standard deviations will be computed from test data corresponding to each state before and after the implementation of the rolling average. Histograms will also be computed for each test for the purpose of illustrating the difference between the two tests.

The reaction wheel tachometer accuracy will also be tested on its accuracy by comparing its output with that of an optical tachometer. The Monarch PLT200 optical tachometer is shown in Fig. 49. By pointing the optical laser at the reflective tape on the reaction wheel, the speed will be known within ±0.01% of the reading [22]. This will be used to validate the tachometer reading. The test procedures for tuning the ADCS testbed PID controller will be discussed next.

Figure 49. The Monarch PLT200 Optical Tachometer

3.5.7 ADCS Testbed Control Tuning and Performance Testing

Tuning the PID controller will come in three steps. First, K_p will be varied with K_d and K_i equal to 0. Typically, the desired response utilizing only a proportional gain is to acquire a fast response with marginal or slightly unstable stability. Even though the proportional gain may create faster than desired angular velocities, the added K_d term will slow down the angular velocity when implemented.

Next, K_d will be varied with the chosen K_p value and with K_i equal to zero. The desired response here is to minimize overshoot of the desired angle with fast rise times. This response seems counter-intuitive, but with the right derivative gain, the angular velocity and overshoot become a trade-off. K_d will be tested in multiples of K_p to identify trends in raising K_d. The angle overshoot should decrease with increasing K_d while the rise time increases.

Finally, K_i will be varied with the chosen proportional and derivative gains. This gain will be based on how long it takes the system to reach a steady-state value and how well the pointing accuracy is. Integral-wind up must also be taken into account. For this reason, the integral portion of the controller will be tested by using its output for all time. Another way of approaching the integral term is to toggle whether it is on or off. By putting a small band around the set point, the integral term can be turned on inside the band and turned off outside the band. This is used to minimize transient response characteristics such as percent overshoot and rise time. Note that the band makes the PID controller non-linear.

Each test will consist of the ADCS testbed holding position at zero and then slewing to 10^o. The rise times, settling times, and percent overshoot of these responses will be compared. The chosen PID gains will be used for different angle commands other than 10^o. The rise time, settling time, and percent overshoot of each response can be compared to each other to determine how linear the system acts. The ADCS testbed will be commanded to 5^o, 10^o, 15^o, 20^o, and 25^o.

Finally, the ADCS testbed will experience a set of impulse disturbance torques to simulate a micro-meteoroid impact. The external torque will be applied by the user pushing the ADCS testbed. A 10^o, 20^o, and 30^o disturbance push will be tested and performance recorded. Note that the amount of external torque applied to the ADCS testbed will be difficult to control; this will have a significant impact on the angular momentum and system performance. However, the test will show that the ADCS testbed can overcome large disturbance torques and settle back to the desired orientation. Calculating the change in angular momentum and average external torque while the ADCS testbed is pointing at steady-state will be discussed next.

3.5.8 Steady-State Angular Momentum/External Torque Testing

The calculation of the change in angular momentum and average external torque on the ADCS testbed while at steady-state is important because it identifies imperfections in AFIT's CubeSat testbed. Theoretically, there should not be external torques and, therefore, no change in the angular momentum of the ADCS testbed as shown in Eq. (20). The test will consist of the CubeSat pointing at zero while atop the air bearing. The angular velocity and wheel speeds will be recorded. Since there are only three of four working wheels, each wheel will need to be accounted for as shown in

$$H_z = \mathbf{I}\,\omega_z + \mathbf{D}\,(\psi_1 + \psi_2 + \psi_3) \tag{69}$$

where H_z is the net angular momentum in the z-axis, ω_z is the z-axis angular velocity of the ADCS testbed, and ψ_1, ψ_2, ψ_3 are the respective reaction wheel speeds in rad/sec.

The average external torque calculation will use the kinetic EOM described in Eq. (22) with some assumptions. The major assumption used is that the angular velocities in the *x*- and *y*-axes are zero. This reduces the kinetic EOM in the *z*-axis to

$$M_z = \mathbf{C}\ \dot{\omega}_z + \mathbf{D}\ (\dot{\psi}_1 + \dot{\psi}_2 + \dot{\psi}_3) \tag{70}$$

where **C** is the scalar MOI in the *z*-axis (denoted I_{zz} in Section 2.1.3), $\dot{\omega}_z$ is the angular acceleration of the ADCS testbed in the *z*-axis, and $\dot{\psi}_1$, $\dot{\psi}_2$, $\dot{\psi}_3$ are the respective reaction wheel accelerations in rad/sec^2. Eq. (70) is useful in imparting information on the inherent external torques when using AFIT's CubeSat testbed.

Three different tests will be conducted. The first two tests will be a steady-state pointing test with and without the doors to the room closed. This is to see if the air current torque is affected by the open doors. Reaction wheel speeds and ADCS testbed angular velocity will be recorded. The next test will consist of covering the ADCS testbed with a trash bag as shown in Fig. 50. The idea is to limit disturbance torques caused by air currents circulating in the room. Note that the trash bag has just under two feet of clearance from the ADCS testbed.

These two tests will compare their average external torques and angular momentum change over time. A conclusion will then be made as to whether air currents cause a significant disturbance on the ADCS testbed. A second possible conclusion is the amount of magnetic torque that is inherent in the ADCS testbed. Once the average external torque is calculated, then the left side of Eq. (70) can be augmented to include a magnetic torque as shown in

$$M_{mag} = B_y\ m_x - B_x\ m_y = \mathbf{C}\ \dot{\omega}_z + \mathbf{D}\ (\dot{\psi}_1 + \dot{\psi}_2 + \dot{\psi}_3) \tag{71}$$

where M_{mag} is the external moment in N-m, and m_x and m_y are the magnetic moments in the *x*- and *y*- axes in units of A-m^2 [44]. The magnetic moments are assumed

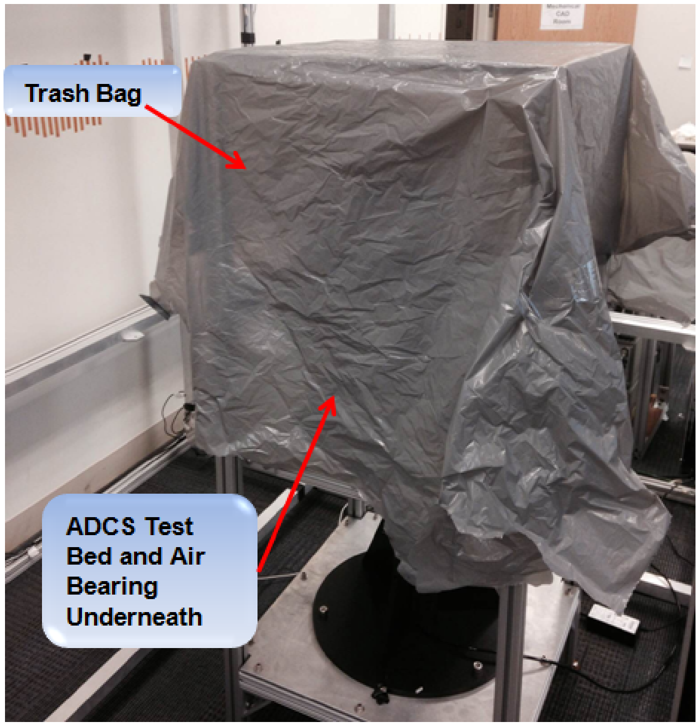


Figure 50. Trash Bag Over ADCS Testbed and Air Bearing for Air Current Testing

constant for this research effort. B_x and B_y are the magnetic field magnitudes in the x- and y-axes. The magnetic fields in the body frame are dependent on the z-axis angle. Using this relationship, the magnetic torque experienced on the body is

$$M_{mag} = |B_i| \left(m_x \; sin\theta_z - m_y \; cos\theta_z\right) \tag{72}$$

where $|B_i|$ is the magnitude of the inertial magnetic field and θ_z is the z-axis angle. Note that m_x and m_y are both unknown constants. We can first solve for m_y by keeping θ_z equal to 0. Now m_x can be solved by ensuring that θ_z is not 0. This will result in a magnetic torque equation dependent on the z-axis angle. Characterizing the magnetic torque inherent in the ADCS testbed's magnetic moment will set the stage for future work using external torque control with torque coils. The creation of the ADCS testbed analytical model will be discussed next.

3.5.9 ADCS Testbed Model Creation and Testing

The purpose of the ADCS testbed model, which was built in MATLAB script, is to predict ADCS testbed performance inside AFIT's CubeSat testbed before the actual test is performed. If the MOI's of the ADCS testbed and reaction wheels are known along with proper noise modeling and an accurate RWA model, then an accurate simulated time response can be computed. First, this model makes one major assumption. Since only z-axis control is desired, the angular velocities in the x- and y-axes are assumed to be zero. This reduces the kinetic EOM to Eq. (70).

The model follows the same estimation and control algorithm that was outlined in Figs. 31, 32, and 33 discussed in Section 3.4. There are a few differences because the MATLAB model is a simulated system. First, there are no simulated sensors in the algorithm. The state outputs, such as the z-axis angle, z-axis angular velocity, and reaction wheel speeds, are all calculated from the dynamics equations derived in Section 2.1.2 and 2.1.3. This is remedied by implementing noise into the state output before going into control. This research effort will assume that simulated noise follows a Gaussian distribution. To create Gaussian noise, the mean and standard deviation of the sensor output must be calculated. The equation for noise can then be written as

$$N = m + s \ randn(1) \tag{73}$$

where N is the noise value, m is the mean around zero, s is the standard deviation, and $randn(1)$ is a random value from a Gaussian distribution. The simulated output from the estimation algorithm is then simply the output from the dynamics plus the noise as shown in

$$x_{sensor} = x + N \tag{74}$$

where x_{sensor} is the simulated sensor state output and x is the output from the dynamics. These values are inputted into the control algorithm.

The MATLAB model control law is the same algorithm the ADCS testbed discussed in Section 3.4.3 software. The commanded wheel speed goes into a reaction wheel model with a time delay determined from gathered data. The reaction wheel model uses wheel RPM to duty cycle conversion similar to the RWA algorithm discussed in Section 3.4.4. The difference is that the dynamics must now be simulated. This is performed by creating a look-up table of time constants inherent in the reaction wheels assuming a first-order model as discussed in Section 3.5.4. The analysis of the time constants is discussed in Section 4.4.

Once the actual reaction wheel speed value is outputted from the reaction wheel model, the reaction wheel speed is converted to torque for input into the ADCS testbed dynamics. The actual torque exerted from the reaction wheels is proportional to the difference in reaction wheel speed before and after the reaction wheel model. The torque inputted into the ADCS testbed model is then calculated using

$$\tau_{sim} = \frac{\mathbf{D}\ (\psi_a - \psi_b)}{\Delta t} \tag{75}$$

where τ_{sim} is the simulated torque of one wheel, ψ_a is the wheel speed after the reaction wheel model, ψ_b is the wheel speed before the input into the reaction wheel model. and Δt is the update rate. Since the torque for the wheels is the same, then simply adding the torques together results in the internal torque input into the dynamics equations.

The model outputs attitude in quaternions and degrees, angular velocity in the z-axis, reaction wheel speeds for all three reaction wheels, commanded orientation, and commanded wheel speed. The model also outputs all simulated sensor measurements for comparison to real data.

Although a reaction wheel model and sensor noise will come from real data, more information is necessary to have an accurate MATLAB model. This information is primarily accurate MOI measurements of the wheels and ADCS testbed as well as time delays inherent in the actual system. This information can be inferred from data analysis. Modeling is important in this way because parameters such as MOI can be accurately estimated by iterating the value so that it matches the data.

3.6 Chapter Summary

Chapter III covered the design and development of the testing procedures that were performed during this research effort. The chapter started by introducing AFIT's CubeSat testbed that includes the Helmholtz cage and air bearing. The Helmholtz cage can manipulate the magnetic field inside the cage, and the air bearing can create a near frictionless surface for the ADCS testbed and loading surface to mimic a space environment. The chosen IMU and RWA as well as specific characteristics were discussed afterwards. The chapter continued by discussing the hardware configuration for the ADCS testbed. This included the ADCS, CDH, and EPS boards. The implementation of these boards inside the ADCS testbed, the ADCS testbed's implementation into AFIT's CubeSat, and the calculation of the ADCS testbed MOI was also discussed. The software configuration of the ADCS testbed section included an explanation of the ADCS algorithm. This algorithm included the estimation, ADCS testbed control, and RWA control algorithms. The section concluded with a discussion on actual versus commanded algorithm speed and how telemetry is wirelessly sent to a computer for data analysis. The last section discussed the various testing procedures for this research effort. The first set of tests focused on the calibration and filtering of the IMU. The second set of tests focused on characterizing and the control design of the RWA. The last set of tests focused on the ADCS testbed itself.

This included sensor noise testing procedures, control design of the ADCS testbed, performance characteristics, and the procedures for calculating the change in angular momentum and external torque applied to the ADCS testbed while atop the air bearing. The creation of the ADCS testbed model in MATLAB was also discussed in detail. The results and analysis of these tests will be discussed in the next chapter.

IV. Results and Analysis

Chapter IV presents the results and analysis of the tests described in Chapter III. First, the results from calibrating the IMU and filtering the magnetometer will be discussed. The following sections detail the calibration, control design, and characterization of the RWA while the following sections feature results from sensor noise analysis, control design and performance of the ADCS testbed, and the results from CubeSat testbed inherent external torques. The chapter concludes with a comparison of data from the ADCS testbed and a analytical model simulation.

4.1 Gyroscope Calibration

As discussed in Section 3.5.1, the gyroscope calibration test is fully automated; the bias constant that is subtracted off of the actual measurements is re-calculated every time the ADCS testbed is re-programmed or reset. Figure 51 demonstrates the histogram of the non-calibrated gyroscope.

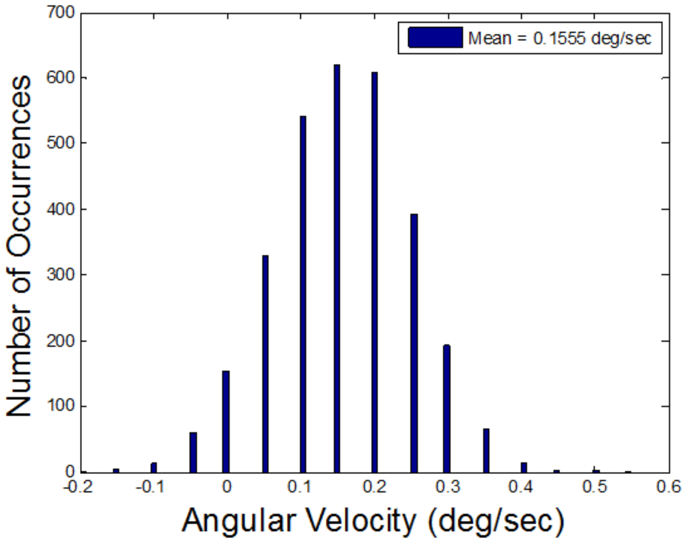


Figure 51. Histogram for Static Response of Non-Calibrated Gyroscope

Notice that the gyroscope's resolution is 0.05^o/sec. Also notice that the mean of Fig. 51 is 0.15^o/sec. As will be discussed in Section 4.7, the slew rate of the ADCS testbed rarely exceeds 1^o/sec. This means that the static gyroscope output is over 15 percent of the maximum speed of the ADCS testbed. The static histogram of the gyroscope output after calibration is shown in Fig. 52.

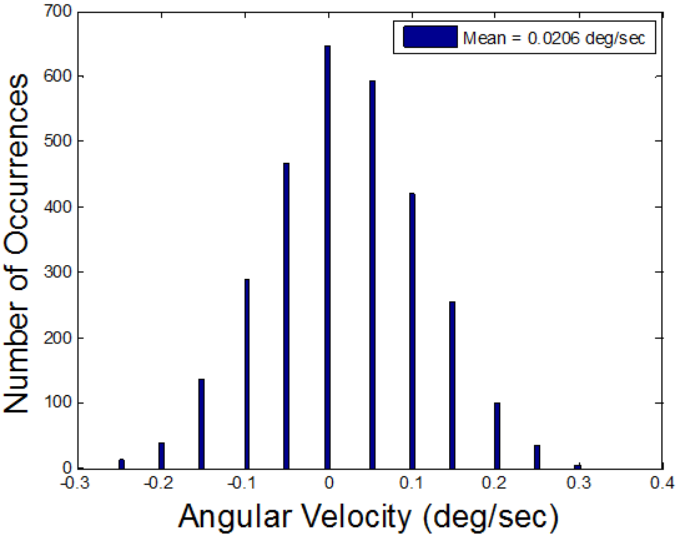


Figure 52. Histogram for Static Response of Calibrated Gyroscope

The new mean of 0.02^o/sec decreases static angular velocity output to just over 2 percent of the maximum angular velocity of the ADCS testbed. The results from filtering the calibrated angular velocity to reduce noise (i.e lower standard deviation) will be discussed in Section 4.6.

The PASCO rotation table (Fig. 39) outputs the measured angular velocity so as to validate the moving accuracy of the gyroscope. Figure 53 illustrates the histograms of the angular velocity output from the rotation table and the gyroscope.

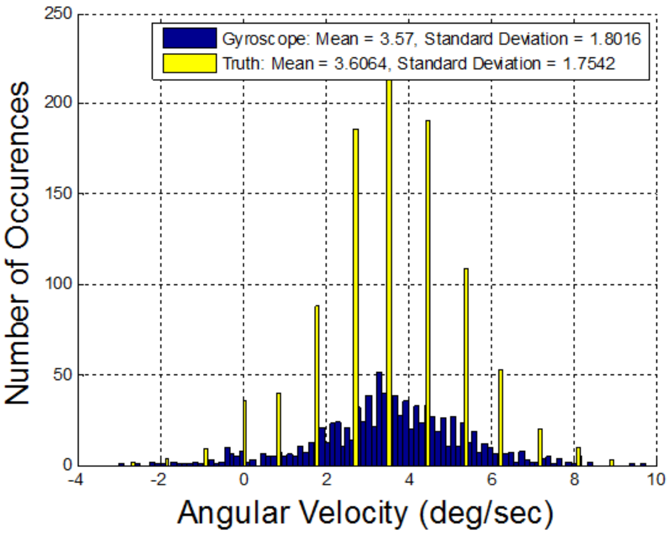


Figure 53. Histogram of Gyroscope vs. Truth Output

The difference in the means of the rotation table output and the gyroscope output is 0.03^o/sec while the difference in standard deviation is 0.04^o/sec. The gyroscope output was only 1% different in the average mean of the rotation table. Discussed next are the results and analysis from calibrating and filtering the magnetometer.

4.2 Magnetometer Analysis

Calibrating and filtering the magnetometer helped quantify the determination capability of the ADCS testbed. The following subsections discuss the results of the testing procedures detailed in Section 3.5.2.

4.2.1 Initial Calibration of the Magnetometer

The initial angle output from the estimation algorithm described in Section 3.4.2 before calibration ranged between 0.5-1^o of error. The automated calibration, which executes for two seconds, subtracts the bias off of the magnetic field while the ADCS testbed is static which equates the body frame magnetometer and inertial magnetic field while the ADCS testbed is coincident with the inertial frame. Figure 54 demonstrates a histogram of angle outputs after the magnetometer has been calibrated.

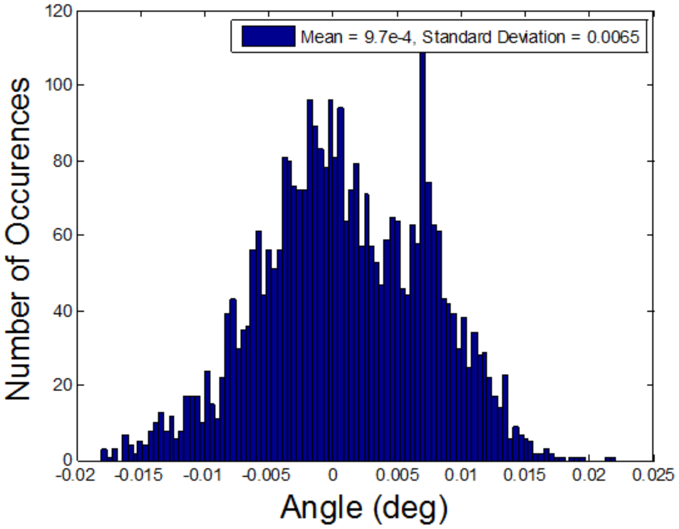


Figure 54. Histogram of Static Estimation with Reaction Wheels Off

Note that Fig. 54 is the estimation of the angle with the body and inertial frames aligned and the reaction wheels off. Figure 54 shows the mean is essentially zero at $9.7E-4$ and the standard deviation is 0.0065^o. The static pointing accuracy with the reaction wheels off is 0.02^o (3σ).

These statistical values are far above the typical magnetometer performance of $\pm 5^o$ as discussed in Section 2.3.2 for reasons that we now theorize. First, the magnetic field generated in these tests by the Helmholtz cage is on average 4 times stronger than Earth's magnetic field. The Helmholtz cage uses a closed-loop controller to keep the magnetic field lines fixed. Magnetometers are more accurate the stronger the magnetic field is. Secondly, the magnetometer bias calibration is very effective with the reaction wheels not spinning which is typically not the case during operation. The distortion of the magnetic field from the reaction wheels spinning will be addressed in Section 4.2.3.

4.2.2 Filtering

As previously discussed in Section 3.5.2.3, the angle output from the estimation algorithm is compared to the truth measurement. Figure 55 shows the static sensor error when the ADCS testbed points at different angles. Note that the reaction wheels were off during the filtering test.Figure 55 shows that the angle error increases the farther away the angle is from zero. To create a filter, the error vs sensor angle is plotted along with its corresponding linear fit as shown in Fig. 56.

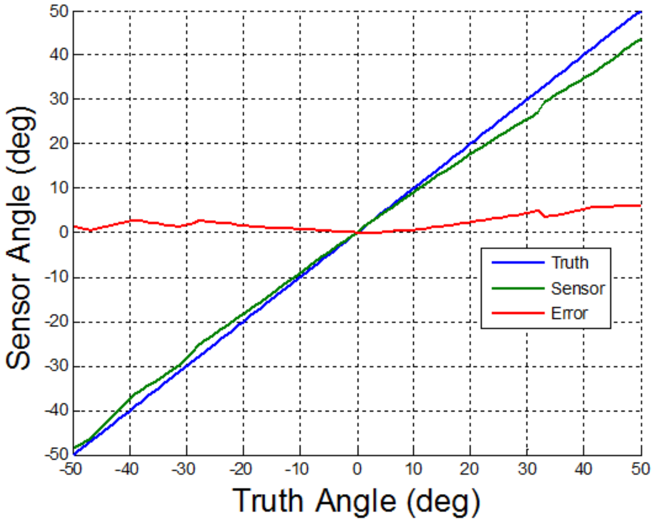


Figure 55. Static Magnetometer Sensor Error

Note that the linear fit equation is located in the legend. The linear fit shown in Fig. 56 is not accurate at angles past ±20^o. Using this linear fit and and applying it to the sensor values results in Fig. 57.

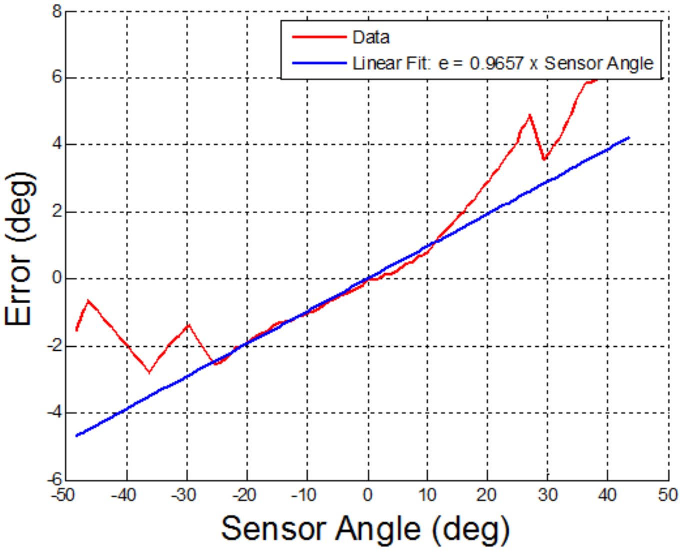


Figure 56. Angle Error vs. Sensor Angle with Linear Fit

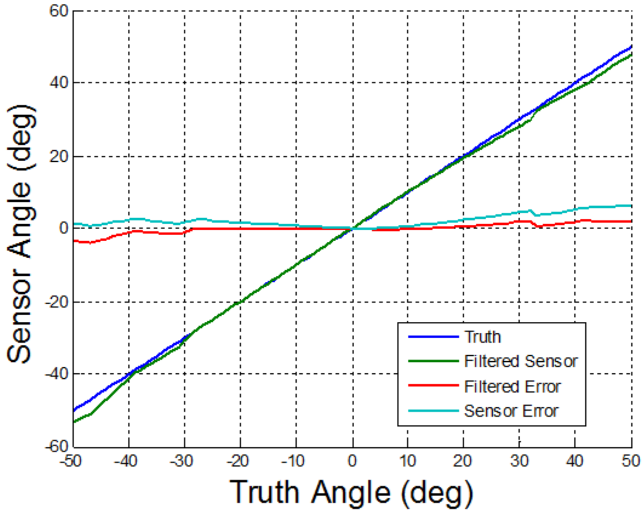


Figure 57. Filtered Angle Error

The unfiltered sensor has a mean error of 2.3^o whereas the filtered sensor's mean error is 0.97^o. Figure 57 shows that the angle error is near zero for angles below $\pm 20^o$, but angle discrepancies of over 1^o will occur past $\pm 20^o$. Figure 58 demonstrates the relationship between angle error and magnetic field error.

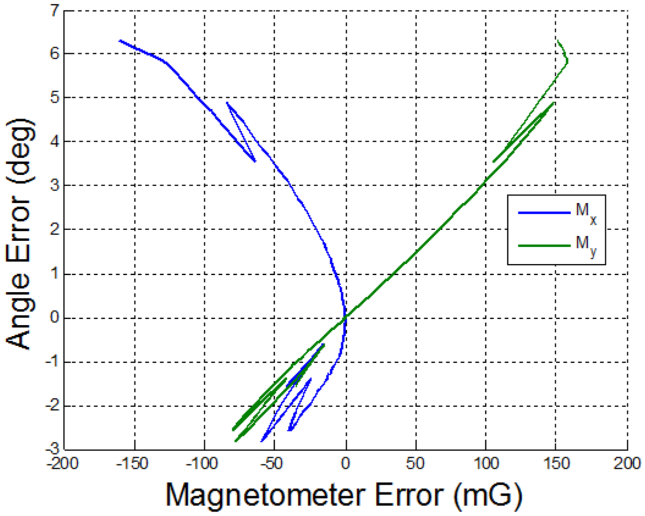


Figure 58. Angle Error vs. Magnetometer Error

Figure 58 shows that a magnetic field error of 150 mG in the y-axis and -100 mG error in the x-axis corresponds to an angle error of 5^o. Note that there will not be positive magnetometer error for M_x because of how the inertial magnetic field inside the Helmholtz cage was set (Section 3.5.2.1). Figure 58 is a useful reference for understanding the angle error associated with a corresponding magnetic field error.

4.2.3 Steady-State Filtering

Recalling Section 3.5.2.4, the reaction wheels were commanded to perform the sawtooth pattern shown in Fig. 43 while the ADCS testbed was static. Figure 59 illustrates the histogram of the angle output for the test. The test suggests that the data is within 1.2^o (3σ) of 0^o. Although that statistic is within typical magnetometer accuracy ranges of $\pm\ 5^o$ as mentioned in Section 2.3.2, the magnetometer can be improved with filtering. When comparing the histograms of Figs. 54 and 59, it is obvious that using magnetometer data for angular position estimates is very sensitive to the magnetic field disturbances caused by spinning the reaction wheels. A linear fit exists when one plots the angle error vs the reaction wheel speed. Creating a linear fit from this curve results in Fig. 60.

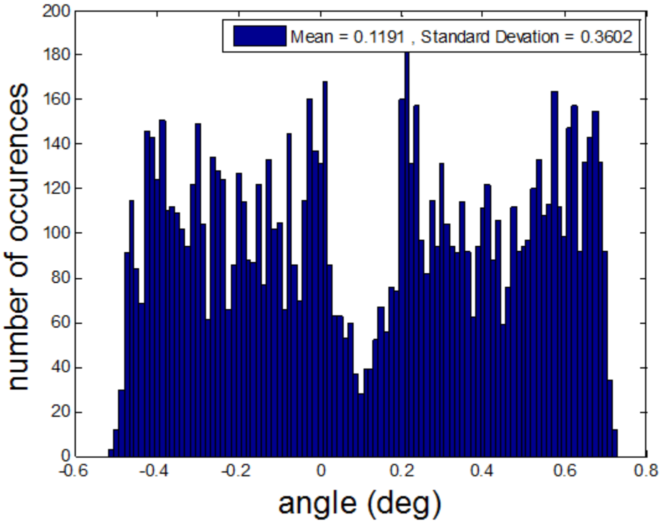


Figure 59. Histogram of Angle Error During Sawtooth Test

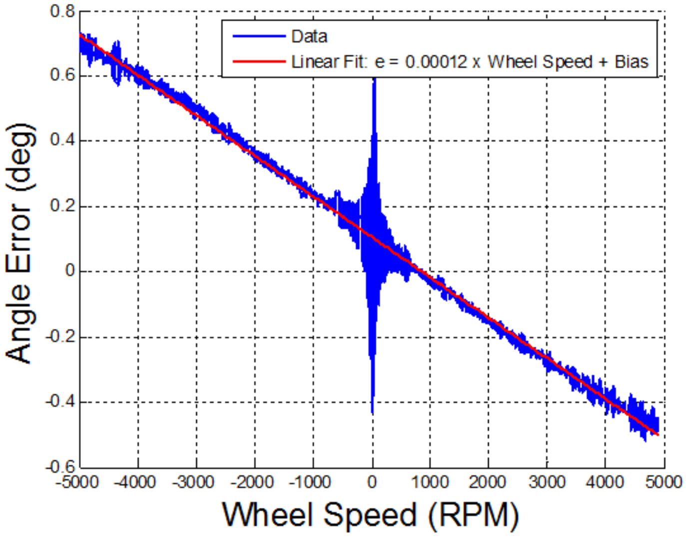


Figure 60. Linear Fit of Angle Error vs. Wheel Speed

Also keep in mind that the "Bias" term labeled in Fig. 60 is subject to change each time the ADCS is re-programmed or powered on. The bias calibration process simply averages the current angle output of the estimation algorithm for half a minute. Since the ADCS testbed is sitting still then any angle output is noise that can be subtracted to better mimic the truth measurement. Once averaged, the mean noise becomes the bias term for the linear fit.

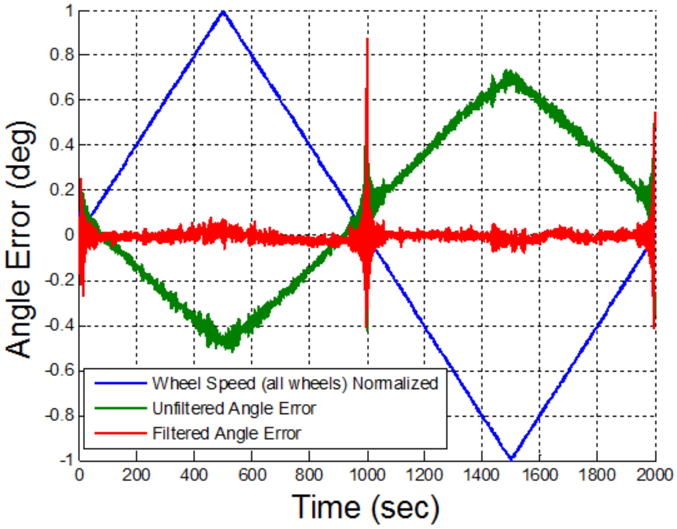


Figure 61. Filtered Angle with Changing Wheel Speed

Also notice how the error spikes when the wheel speed reaches 0 RPM. This may be caused by the wheels switching directions which causes the magnetic field to rapidly change. Outside of this spike, the most significant errors occur at the highest wheel speeds. Figure 61 shows both calibrated and uncalibrated angle estimate errors when using magnetometer data only. The wheel speed is normalized to show how the speed effects the angle error. Notice how well the filter responds to changes in wheel speed compared to the unfiltered response. Notice when the wheel speed is around zero the angle error increases. This is further supported by Fig. 60 because the angle error spikes when the wheel's speeds are near 0 RPM. This means that when the ADCS testbed is at steady-state angular position, it is undesirable for the reaction wheel speeds to be close to 0 RPM. To avoid these spikes in attitude estimation, the wheels are spun to 1000 RPM to calibrate the initial bias for the steady-state filter.

Figure 62 shows the histogram of the estimation algorithm at steady-state with the saw-tooth input on the reaction wheels.

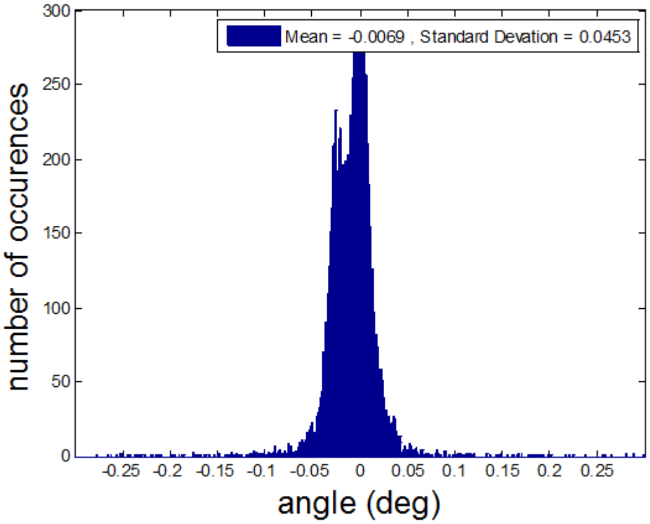


Figure 62. Histogram of Filtered Angle Error During Sawtooth Test

With the steady-state filter, the estimation algorithm will be accurate to 0.1^o (3σ) compared to the original 3^o of estimation accuracy identified at the beginning of this research effort. Comparing Figs. 62 and 59 shows the difference in standard deviation to be 0.3^o and mean to be 0.2^o. The addition of this filter makes the estimation algorithm 30 times more accurate.

The calibration and filtering of the magnetometer was strictly for a static magnetic field that is stronger than normal which results in improved attitude estimates than one would normally see from typical magnetometers. Using this logic, it is useful to manipulate the magnetometer into being more accurate than it actually is so as to see the real capability of the RWA and control algorithms. The duty cycle calibration of the reaction wheels are discussed in the next section.

4.3 RWA Duty Cycle Calibration

The duty cycle calibration testing described in Section 3.5.3 is performed to fit a linear curve between the reaction wheel speed and duty cycle. An explanation of how duty cycle count relates to reaction wheel speed is located in Section 3.5.3. Figures 63 through 68 show the duty cycle vs. reaction wheel speed curves and their respective linear fits. Note that the linear fit is located in the legend.

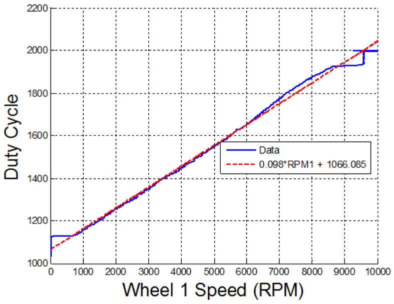


Figure 63. Forward DC vs. Wheel 1

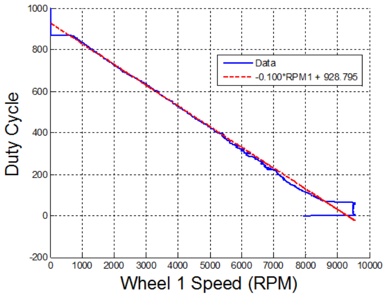


Figure 64. Reverse DC vs. Wheel 1

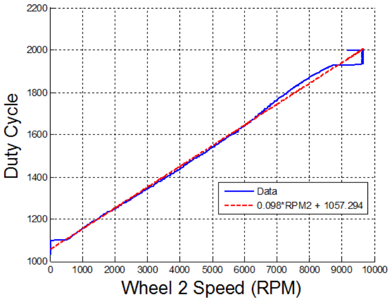


Figure 65. Forward DC vs. Wheel 2

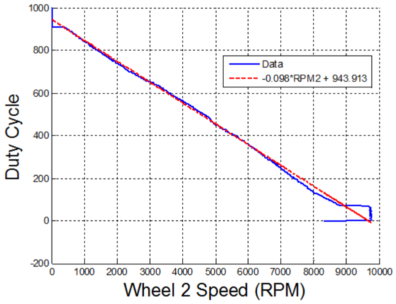


Figure 66. Reverse DC vs. Wheel 2

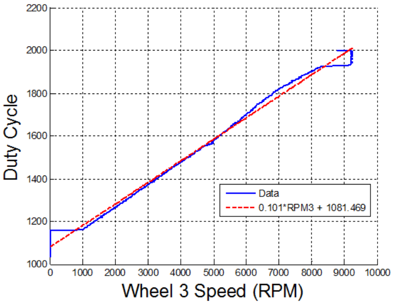


Figure 67. Forward DC vs. Wheel 3

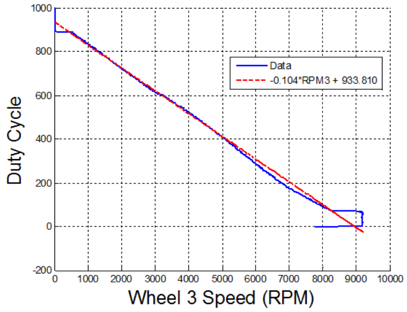


Figure 68. Reverse DC vs. Wheel 3

Notice that when the duty cycle count is roughly ± 100 from 1000, the motor is not enough to overcome the static friction inherent in the bearings when the wheels are not moving. The reaction wheels, however can rotate at duty cycles between 900 and 1100 once the wheels start moving. For this reason, the reaction wheels will be initially directed to a nominal speed of 1000 RPM before starting any control experiments. Overall, the reaction wheel speed to duty cycle conversion will produce the actual wheel speed close to the desired wheel speed. The reaction wheel controller is then used to decrease the rise time of this response and achieve low steady-state error. The next section will discuss the analysis and results of reaction wheel time constant testing.

4.4 RWA Time Constant Analysis

As previously discussed in Section 3.6.4, calculating the time constant for a wide range of wheel speeds is useful in determining the wheel speed cutoff for control and determining the best way to model the wheels. Time constants were calculated for each wheel in a variety of ways as discussed in Section 3.5.4. Figures 69 through 71 illustrate the histogram of the time constants for each of the 3 reaction wheels. Notice how each figure contains an outlier at a higher time constant. These values demonstrate the time constant when speeding up to 6000 or -6000 RPM from a lower wheel speed. 3σ of each wheel (mean and standard deviation shown on legend) puts the time constant at 23.27, 23.31, and 26.08 for wheels 1, 2, and 3 respectively. The outliers far exceed 3σ which means that when the wheels ramp up to these speeds, the wheels time constants become too far off the mean value to be considered near linear. With such a slow time constant compared to other wheel speeds, control that would require the wheels to transition to this speed will cause a larger difference in response.

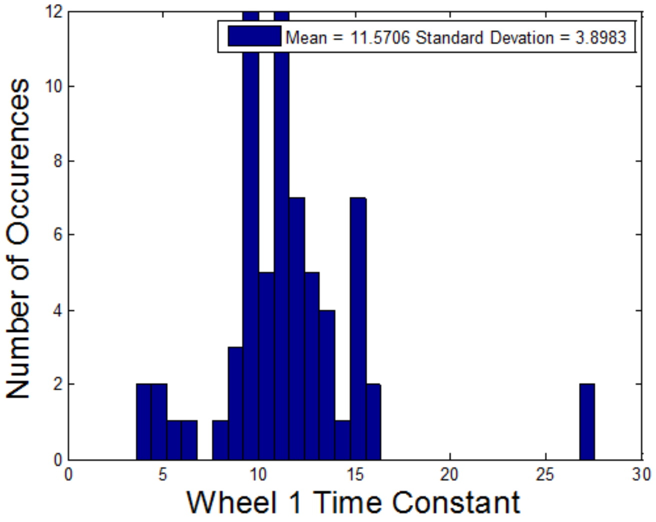


Figure 69. Histogram of Wheel 1 Time Constants

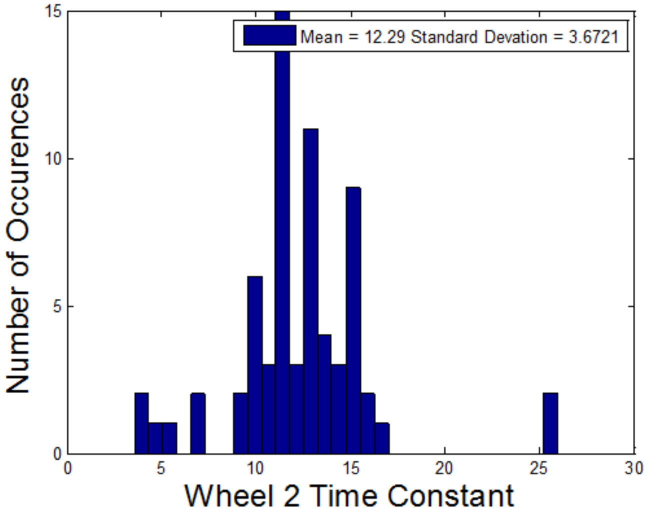


Figure 70. Histogram of Wheel 2 Time Constants

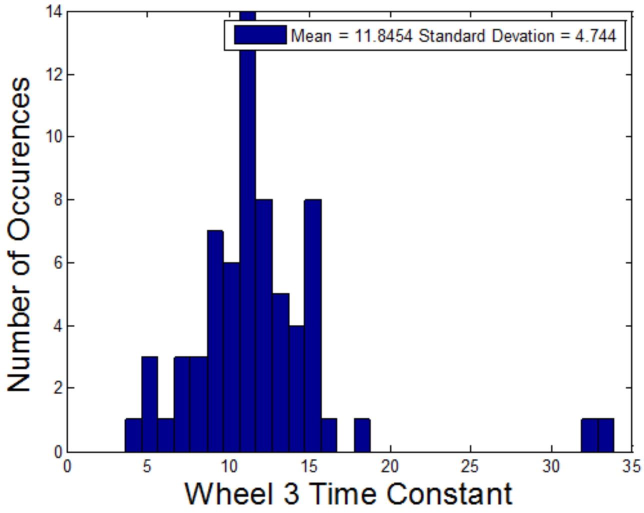


Figure 71. Histogram of Wheel 3 Time Constants

The slow time constant from high reaction wheel speeds makes the reaction wheel's performance, and therefore the ADCS testbed's performance, less predictable and more nonlinear. For this reason, the wheel speed cutoff was chosen to be 5500 and -5500 RPM. This keeps all time constants well within 3σ which creates more linear behavior. With the wheel speed cutoff implemented, the new mean and standard deviations of the wheel time constants are outlined in Table 1.

Table 1. Mean and Standard Deviation of Wheel Speed Time Constants

Wheel Number	Mean (sec)	Standard Deviation (sec)
1	11.0851	2.7659
2	11.8766	2.8443
3	11.1960	2.9722

Although the mean and standard deviation of the time constants are not the same for each reaction wheel, they are closely aligned. The mean of each reaction wheel is within 0.78 seconds of each other. The standard deviations only differ at a maximum of 0.17 seconds. Each reaction wheel, however, has extremely low time constants in the ranges of 4 to 8 seconds. Whenever the wheels are commanded to go from a positive wheel speed to a negative wheel speed, the time constant becomes much lower than the mean value. Going from positive to negative wheel speeds is necessary for improved control authority which allow these values to be within 3σ of the mean.

The fact that the time constant is better at switches between positive and negative wheel speeds suggests that starting from a low wheel speed will improve desired time response characteristics. This, however, stands in contrast with the conclusions found in Section 4.2 with filtering the magnetometer. Estimation is poor with wheel speeds around zero. Control is only as good as the estimation, so it is still preferred to control the reaction wheels at speeds above 1000 RPM. This will, however, decrease control authority because the time constants will be higher. Although the time constants will not be as low while spinning at higher wheel speeds, the time response performance characteristics can be improved with a controller. Tuning the RWA controller will be discussed in the next section.

4.5 RWA Controller Analysis

The reaction wheel control proved to be very useful in increasing performance characteristics of the wheel time response given a step input. Tuning just the proportional gain will first be discussed followed by the results from changing the integral gain. Finally, the next subsection details the results from different step inputs using the chosen controller gains.

4.5.1 Tuning the Proportional and Integral Gains

To start, only the proportional gain K_p was varied. The rise time and settling time were calculated for each wheel. The wheels started at rest and were accelerated to 3000 RPM. Figures 72 through 74 show the time response for ramping up to 3000

RPM from 0 RPM for varying the proportional gain for the three reaction wheels. Comparing these responses with the simulated response given by Fig. 48 shows that the larger K_p is the shorter the time to the commanded speed. The data differs from Fig. 48 because it appears that the speed of the response reaches its limit at gain values of 7 and above. Notice how quickly from Figs. 72 through 74 that K_p converges. Any higher rise in K_p would merit a similar response. There are mechanical limits to how fast the reaction wheels can accelerate which simple simulation cannot predict.

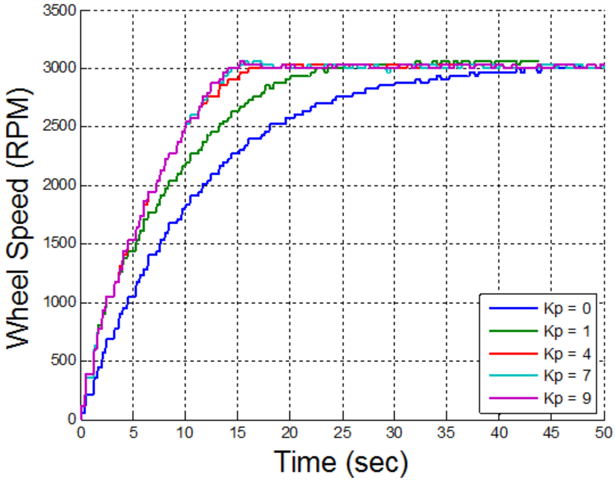


Figure 72. Wheel 1 Time Response for Varying K_p

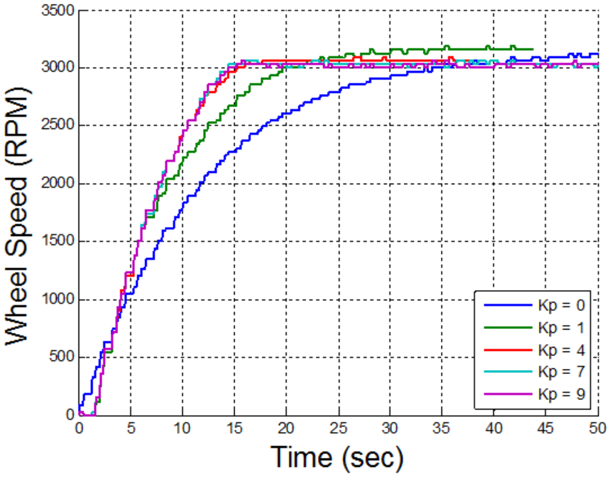


Figure 73. Wheel 2 Time Response for Varying K_p

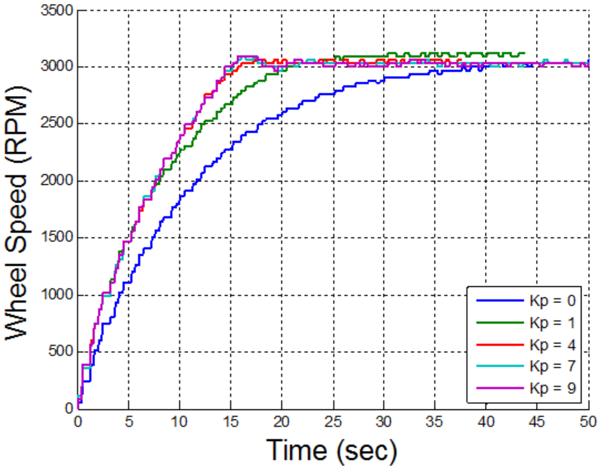


Figure 74. Wheel 3 Time Response for Varying K_p

Table 2 summarizes the rise times and settling times of reaction wheel 1. Note that rise time was calculated by subtracting the times where the response was 10% and 90% percent of its desired value. There are many rise time rules, but this rule is generally used for over damped (i.e first-order) systems [7]. The settling time is calculated as the time at which the speed is within 2% of the desired value.

Table 2. Wheel 1 Time Response Characteristics for Varying Gains

K_p	K_i	Rise Time (sec)	Percent Overshoot	Settling Time
0	0	22.8	0	28.4
1	0	15.6	0	N/A
4	0	11.6	0	13.2
7	0	11.2	0	12.4
9	0	11.2	0	12.4
9	0.1	11.2	5	N/A
9	0.5	11.2	15	25.6

Notice that there is no difference in the rise and settling times of the time response for any of the wheels when K_p is 7 or 9. This means that raising the gain any higher will not yield a faster response. Mechanically, the motor cannot apply a higher voltage to the reaction wheel, this means that any higher K_p value would not merit a larger torque. Also notice that the steady-state error in Figs. 72 through 74 decreases with higher K_p as well. This is evident by comparing the steady-state response of K_p equal to 1 versus the other chosen gains. K_p equal to 1 overshoots the desired wheel speed whereas the higher gain values do not. The chosen K_p for this research effort is 9.

Looking again at Table 2, one will notice that the integral gain K_i was varied as well. The results from tuning this gain showed much poorer performance than just using a proportional gain. Figure 75 shows the time response of varying K_I values for reaction wheel 1. Note that the commanded speed is 3000 RPM.

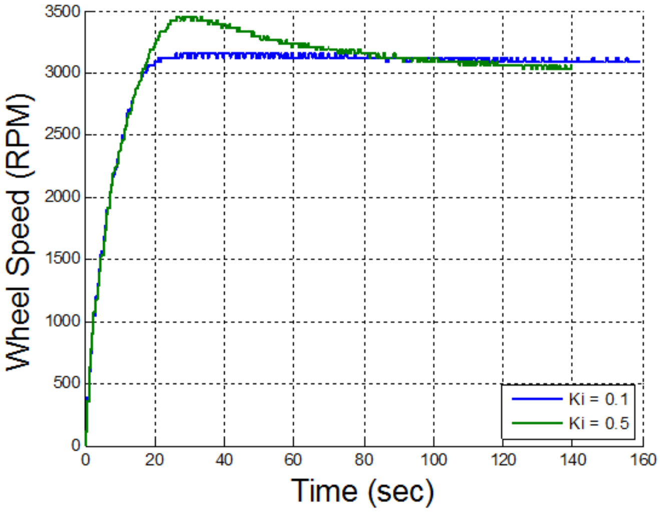


Figure 75. Wheel 1 Time Response for Varying K_i

Figure 75 illustrates that a low K_i that minimizes overshoot is not ideal for settling back to the desired speed. The test was run for more that 150 seconds and it still was not able to settle back to 3000 RPM. The gain was then increased to 0.5 which caused even more overshoot but it did settle within 2% of 3000. Although the system did settle back to 3000 RPM, the settling time was over 140 seconds. Due to the integral control's poor performance on reaction wheel response, it will not be used for control of the reaction wheel motors

4.5.2 Variable Step Inputs with Chosen Gains

Now that the gains had been chosen, the wheels were given a multitude of different inputs so as to analyze their response characteristics. Figure 76 shows reaction wheel one given a multitude of inputs ranging from 1000 RPM to 5000 RPM. Figure 77 shows reaction wheel 1 given inputs ranging from -1000 RPM to -5000 RPM. Note the rise time increases with increasing wheel speed difference. Each test produced nearly identical transient responses. Notice that the rise time for 5000 and -5000 RPM is almost double that of the rise time for 4000 and -4000 RPM.

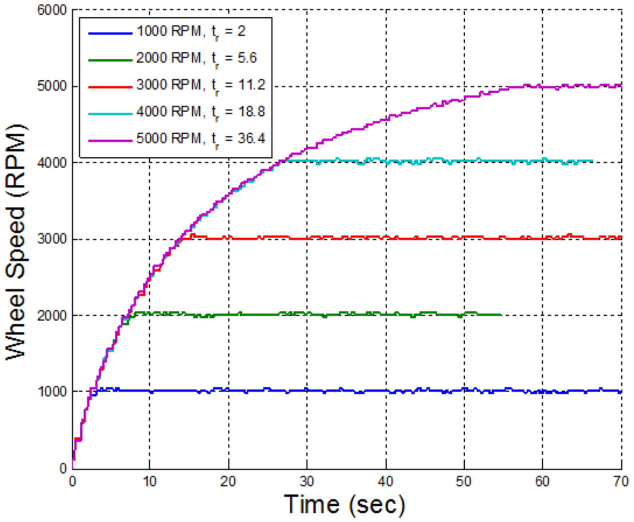


Figure 76. Wheel 1 Time Response for Varying Positive Inputs

Typically a linear response will have close to the same time response characteristics; the large differences in rise time for different inputs denotes a non-linear response. Achieving higher speeds will be more difficult for the controller because of wheel speed cutoff discussed in Section 4.4. The highest command the controller can give is that of 5500 RPM. This means that the time constant of the response will actually be lower than a response a with wheel speed cutoff. This is little cause for concern in satellite attitude control because rarely is the wheel speed commanded from the ADCS controller going to be 5000 RPM greater than the initial wheel speed. Conserving

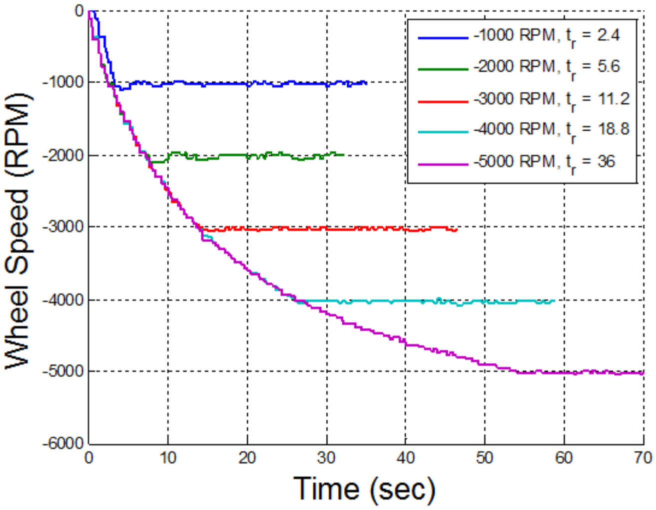


Figure 77. Wheel 1 Time Response for Varying Negative Inputs

time constant linearity by using a wheel speed cutoff would then be preferred. The results and analysis of filtering sensor noise will be discussed next.

4.6 Sensor Noise Analysis

As mentioned in Section 3.5.6, two tests were performed to determine the performance of the rolling average on the standard deviation and mean of the states. Note that the wheels were commanded to stay at a constant speed of 1020 RPM. This means that the following histograms will be misleading when it comes to non-constant wheel speeds. The data presented is meant to represent steady-state noise where the wheel speeds will be near constant. The histogram before the rolling average for the z-axis angle is illustrated in Figure 78. Notice that the resolution of the attitude estimate is roughly 0.02^o. The mean is within 0.01^o of truth while the standard deviation is 0.02^o. Figure 79 shows the attitude estimate with the implementation of the rolling average.

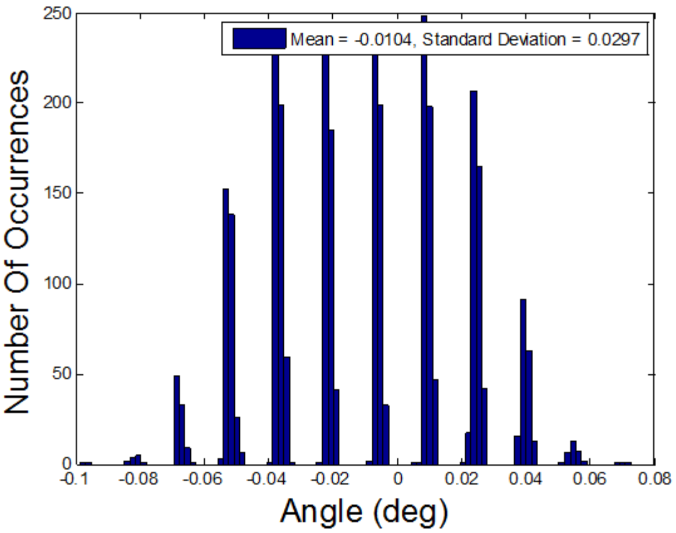


Figure 78. Histogram of Angle without Rolling Average

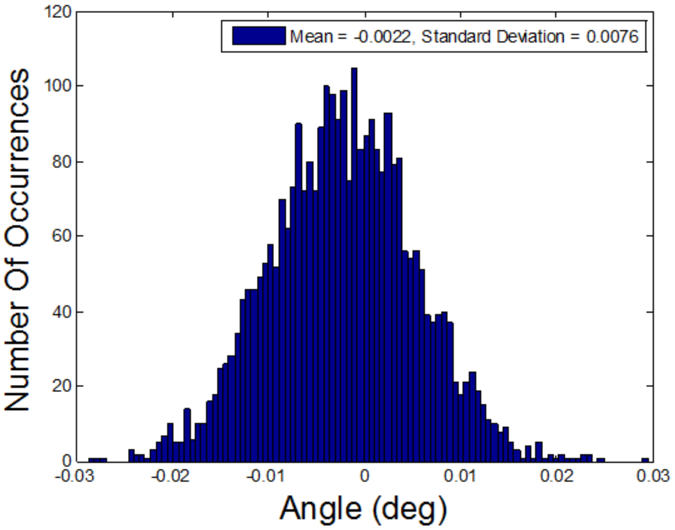


Figure 79. Histogram of Angle with Rolling Average

The mean and standard deviation using the rolling average decrease by a factor of ten: the new mean is -0.002^o and standard deviation is 0.007^o.

The rolling average also decreased sensor noise on the gyroscope as well. Figures 80 and 81 show the histograms of the angular velocity output without and with the rolling average respectively.

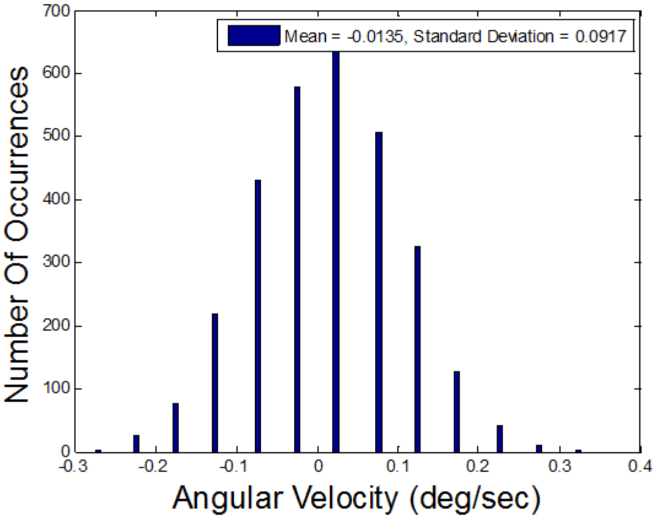


Figure 80. Histogram of Angular Velocity without Rolling Average

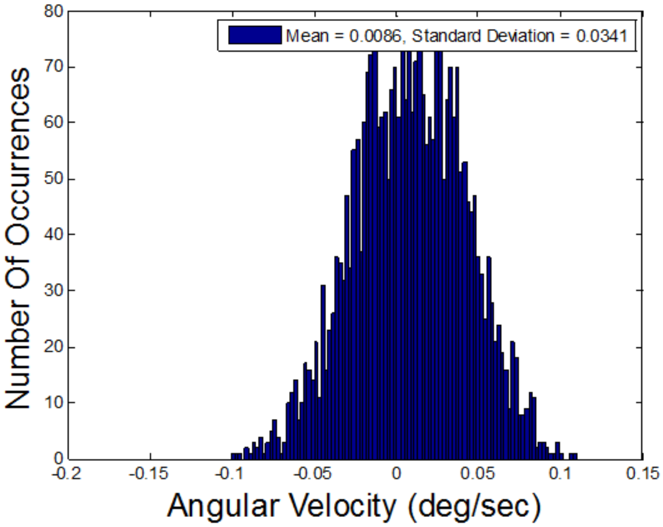


Figure 81. Histogram of Angular Velocity with Rolling Average

Figure 80 shows that the data is irregular as compared to Figure 78. The mean and standard deviation without the rolling average is -0.01^o/sec and 0.09^o/sec respectively. The implementation of the rolling averaged reduced the mean to 0.008^o/sec and the standard deviation to 0.03^o/sec. The decrease in noise for the angle and angular velocity using the rolling average filter increases steady-state estimation accuracy which will therefore increase steady-state control accuracy.

The reaction wheel speeds stay relatively constant when commanded to a desired RPM. The optical tachometer shown in Fig. 49 was used to determine sensor accuracy of the reaction wheel tachometer. The wheel speed stays within ±15 RPM of the commanded value of 1020 RPM. Unfortunately, the reaction wheel Hall sensors are only capable of producing sensor readings of ±30 RPM resolution. This means that along with 1020 RPM readings, occasionally a reading of 990 or 1050 will output as shown in Fig. 82. Note that only reaction wheel 1 is demonstrated here, but the other wheels produced similar results.

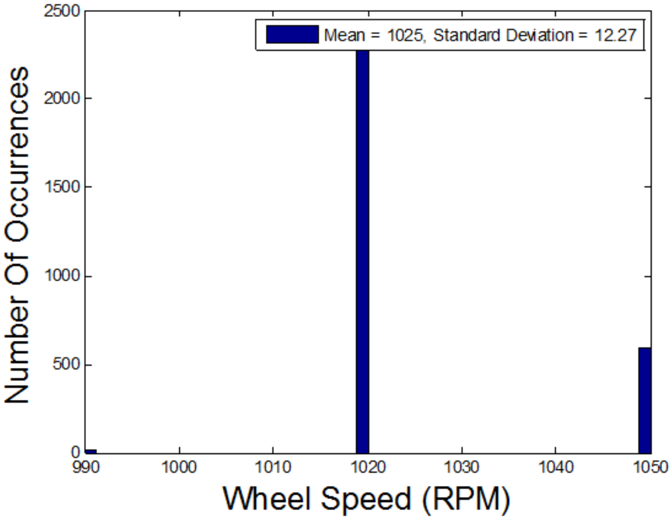


Figure 82. Histogram of Tachometer without Rolling Average

The tachometer shows that it is at 1020 RPM for the majority of the time. The mean is slightly higher at 1025 RPM and the standard deviation is 12.27 RPM. Figure 83 illustrates the implementation of the rolling average onto the tachometer.

The mean and standard deviation of the wheel speed output decrease to 1024 RPM and 6 RPM respectively. The mean is nearly identical, but the standard deviation was reduced to half. 3σ of this data is 18 which comes close to matching the accuracy value represented by the optical tachometer. The increased accuracy and reduction of noise for the sensors will increase steady-state estimation and control accuracy.

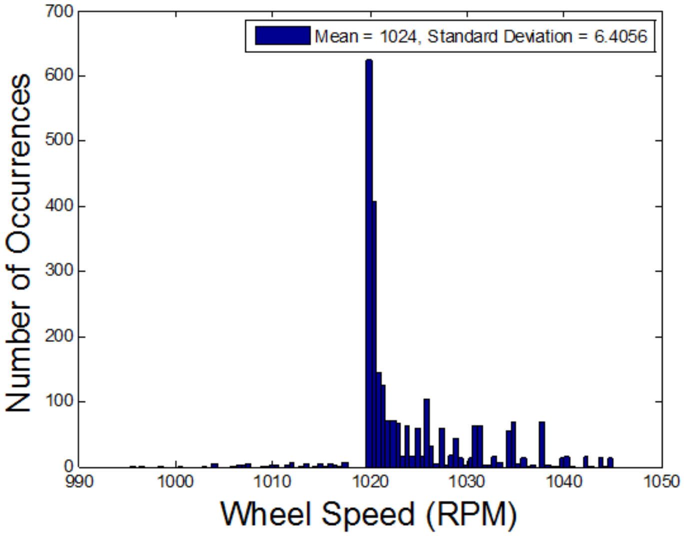


Figure 83. Histogram of Tachometer with Rolling Average

4.7 ADCS Testbed Tuning and Performance Analysis

The PID controller aboard the ADCS testbed was quantified through the testing procedures described in Section 3.5.7. Figure 84 illustrates the time responses of the ADCS testbed slewing to the desired angle of 10^o with varying K_p. Also discussed in Section 3.5.7, increasing K_p will increase the rise time of the respective time response. Notice from Fig. 84 that the rise time increases with increasing K_p but the percent overshoot does not. Table 3 displays the rise time and percent overshoot of each time response shown in Fig. 84.

Table 3. ADCS Testbed Time Response Characteristics of Varying K_p for a 10^o Slew

K_p	Rise Time (sec)	Percent Overshoot
0.01	13.1	111.2
0.02	9.8	100.4
0.05	6.0	118.5
0.1	4.6	104.2

The cause of uncharacteristic behavior in the percent overshoot category is related to how the reaction wheels respond to smaller torque commands. For the reaction wheels

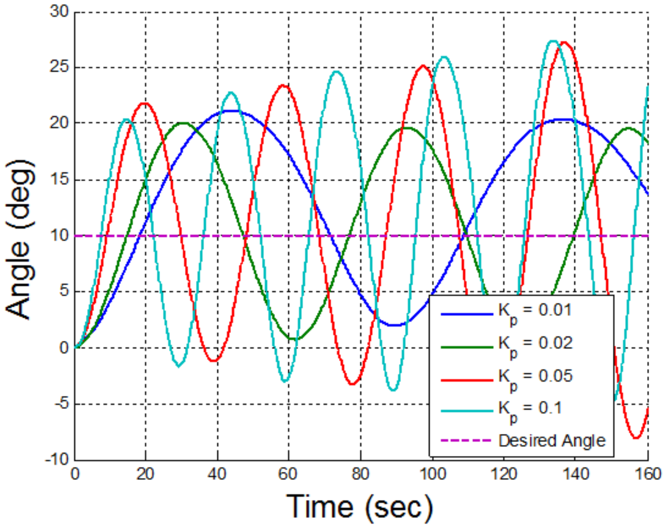


Figure 84. ADCS Testbed Time Response with Varying K_p for a 10^o Slew

to slow down, the momentum built from spinning the reaction wheels up must also be counteracted; the build-up of momentum is overcome by larger torque commands given from the ADCS testbed. Because K_p is higher, the torque commands will be larger at lower desired angle errors. This will make the reaction wheels react quickly to speed up or slow down. Since the rise time increases with increasing K_p and the overshoot does not present a noticeable trend. K_p equal to 0.1 was chosen for this research effort because of its fast response time. Although K_p equal to 0.1 is unstable, adding derivative control will stabilize the ADCS testbed.

Varying K_d while keeping K_p equal to 0.1 yielded predictable results. Figure 85 shows the time response of the ADCS testbed with varying K_d and constant K_p. Notice that increasing K_d decreases percent overshoot and increases rise time which confirms the behavior predicted in Section 3.5.7. Although the addition of a derivative term allowed the ADCS testbed to stabilize, the ADCS testbed angle starts to drift when settling at the commanded position. Table 4 presents the rise time, percent overshoot, average steady-state error, and standard deviation value of each time response of varying K_d. To rid the response of overshoot, a K_d 10 times larger than

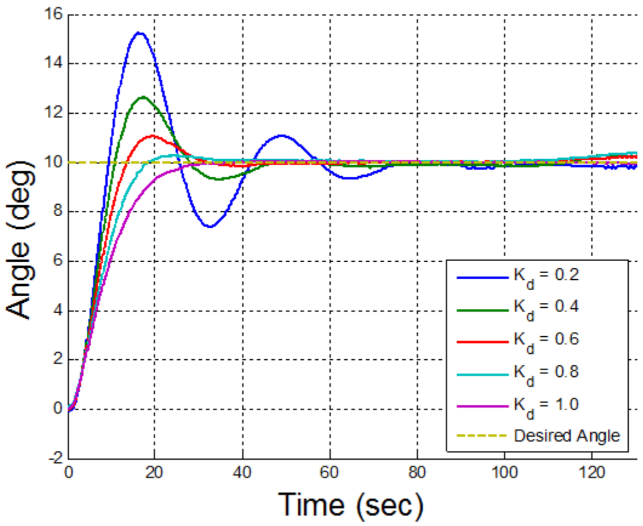


Figure 85. ADCS Testbed Time Response with Varying K_d for a 10^o Slew

K_p is used. Note that the "critically damped" K_p would be 1.16 using simulation. This more than doubles the rise time when using a K_d twice the size of K_p. Notice that settling time is not a specification in Table 4. This means that the steady-state value was not able to stay within ± 2% of the final value for any of the time responses. This will be corrected with the use of an integral term to the PID controller.

Table 4. ADCS Testbed Time Response Characteristics of Varying K_d for a 10^o Slew

K_d	Rise Time (sec)	Percent Overshoot	Average Steady State Error (deg)	Steady-State Std Dev (deg)
0.2	6.0	52.2	0.15	0.067
0.4	7.1	26.0	0.04	0.14
0.6	8.9	10.7	-0.08	0.082
0.8	11.5	2.8	-0.16	0.16
1.0	15.7	0.5	-0.10	0.14

The angular velocity of these time responses is also worth analyzing. CubeSat slew rates are typically only about 1^o/sec or slower; the ADCS testbed seeks to have a range of slew speeds that are above or below 1^o/sec. Figure 86 shows the angular velocity vs time of the ADCS testbed slewing to 10^o with varying K_d. Obviously, the angular velocity decreases with increasing K_d as noted by the increasing rise time shown in Table 4. The angular velocities of the ADCS testbed at K_d equal to 0.6 and under are above 1^o/sec. The addition of an integral term will only increase the angular velocity response. Although the ADCS testbed has the ability to reach a wide range of angular velocities and remain stable, this research effort will keep percent overshoot low by choosing a K_d value of 1.0.

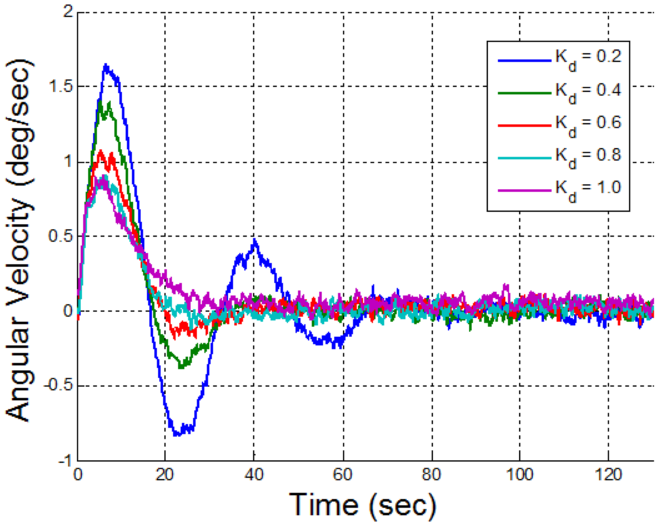


Figure 86. ADCS Testbed Angular Velocity Response with Varying K_d for a 10^o Slew

The integral gain must now be inserted into the PID controller so that settling time and steady-state error of the time response of the ADCS test be can be minimized. First, the addition of K_i was tested with and without a band around the set point that toggles K_i on and off (Section 3.5.7). Figure 87 shows the time response of the ADCS testbed slewing to 10^o without the use of K_i, with the use of K_i but without a band around the set point, and with K_i plus the band around the set point.

The transient response of the ADCS testbed without the set point band produces a larger rise time and percent overshoot as predicted in Section 3.5.7. The ADCS testbed response with the set point band produces a response similar to the no K_i response. The difference is that the addition of K_i keeps the steady-state value much

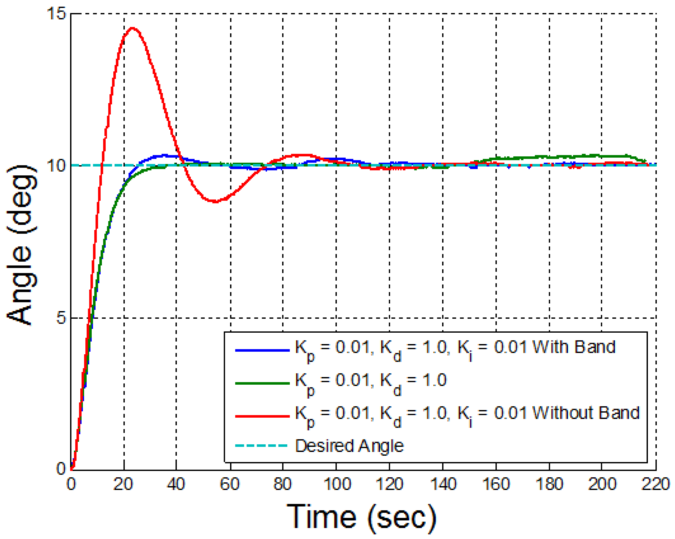


Figure 87. ADCS Testbed Angular Time Response with K_i Set Point Band for a 10^o Slew

lower than without K_d. This can be seen at the end of Fig. 87 by noticing the drift in the response without K_i versus the smaller drift with the use of K_i. Figure 88 illustrates the angular velocities of the ADCS testbed time responses with and without a set point band. Note that the no K_i response is there for reference.

The speed of the response without the set point band is approximately 1^o/sec while the speed of response with the set point band is approximately 0.8^o/sec. Although both of these responses are within the desired angular velocity limits, larger slews will result in larger angular velocities. For this reason, the set point band will be implemented while using an integral term. After the set point band was established, steady-state performance was measured by varying K_i. Figure 89 shows the steady-state response of the ADCS testbed slewing to 10^o with varying K_i.

K_i equal to 0.15 has the highest percent overshoot, but the settling time is almost equal to a K_i equal to 0.1. K_i equal to 0.005 has a longer settling time due to the response going above ±2% of the desired value at times above 100 seconds. Table 5 presents the percent overshoot, settling time, average steady-state error, and standard

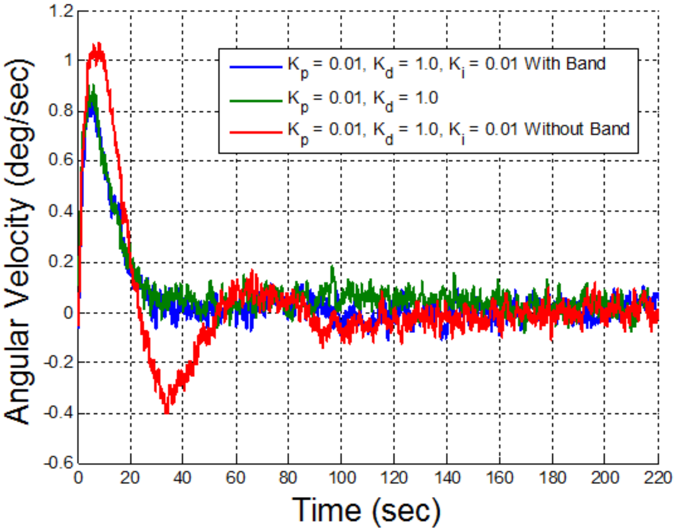


Figure 88. ADCS Testbed Angular Velocity Response with K_i Set Point Band for a 10^o slew

deviation value of each time response of varying K_i.

Table 5. ADCS Testbed Time Response Characteristics of Varying K_i

K_i	Percent Overshoot	Settling Time (sec)	Average Steady State Error (deg)	Steady-State Std Dev (deg)
0.005	2.2	139.6	0.001	0.02
0.01	3.2	43.0	0.005	0.02
0.015	3.8	43.5	0.004	0.02

A trade-off exists with the transient response and steady-state characteristics of the ADCS testbed. The K_i equal to 0.005 response goes outside the required band for settling time whereas higher K_i values drastically decrease the settling time of the ADCS testbed time response. The difference between the characteristics of K_i equal to 0.01 versus 0.015 is very small. K_i equal to 0.015 has a slightly larger percent overshoot and settling time, but the average steady-state error is slightly lower. Since the average steady-state error between the two time responses is only 0.0015^o, a K_i value of 0.01 will be used for this research effort. In summary, the chosen gains are as follows: K_p equal to 0.1, K_d equal to 1.0, and K_i equal to 0.01.

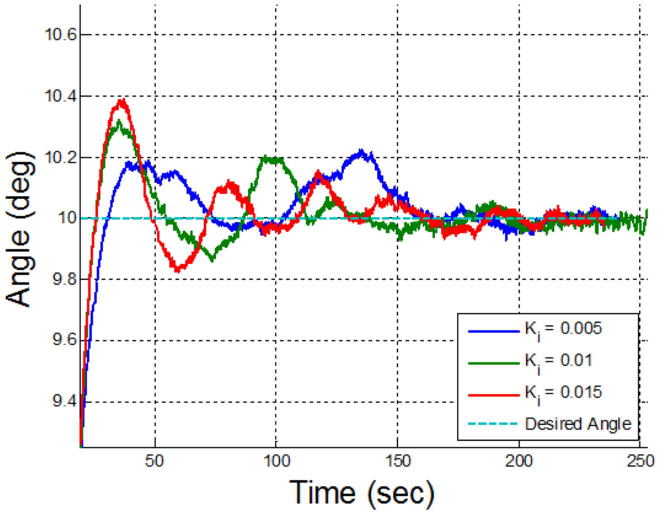


Figure 89. ADCS Testbed Angular Time Response with Varying K_i

A major aspect to compare is the performance of the ADCS testbed steady-state control vs. the estimation accuracy detailed in Section 4.2. Two different performances were given for estimation accuracy, one with constant wheel speed and one with the wheels speeding up and slowing down in a sawtooth pattern (Fig. 61). The mean and standard deviation of these estimation performances plus the control performance is detailed in Table 6.

Table 6. Estimation and Control Steady-State Performance Characteristics

Scenario	Average Steady State Error (deg)	Steady-State Std Dev (deg)
Estimation: Constant RPM	-0.002	0.007
Estimation: Sawtooth	-0.006	0.04
Pointing Accuracy	0.005	0.02

Notice that the mean and standard deviation of the control are lower than the sawtooth estimation. The sawtooth estimation sought to quantify the average accuracy of the magnetometer by using different wheel speeds. When the ADCS testbed is pointing at steady-state, the wheel speed is essentially constant for short time periods. This negligible change in wheel speed does not have to handle the inaccuracies inherent in the linear filter created for the magnetometer off of the sawtooth estimation. In this way, the control pointing accuracy should actually be compared with the constant RPM estimation accuracy. Comparing these two results demonstrate the steady-state pointing capability using the reaction wheels with very accurate estimation. The pointing accuracy of the control algorithm is 0.07^o (3σ) with an estimation accuracy of 0.02^o (3σ). AFIT's four wheel pyramid array has shown that it is possible to achieve less than 0.1^o of pointing accuracy around the z-axis inside AFIT's CubeSat testbed. The next subsection will discuss how the ADCS testbed time response varies when the desired angle is changed.

4.7.1 ADCS Testbed Slew Performance Analysis

With the PID gains set in the ADCS testbed control algorithm, the ADCS testbed time response characteristics while varying the commanded angle were tested and analyzed. Figure 90 illustrates the ADCS testbed time response for slews from 5^o to 25^o.

Each ADCS testbed time response has similar overshoot and rise times. Settling times are also similar with different slews. Table 7 provides values for rise time, percent overshoot, and settling time of each slew.

Table 7. ADCS Testbed Characteristics for Varying Slews

Slew (deg)	Rise Time (sec)	Percent Overshoot	Settling Time (sec)
5	15.50	8.45	51.32
10	15.38	4.44	53.27
15	15.77	3.16	48.85
20	15.47	3.58	47.97
25	15.52	3.06	47.07

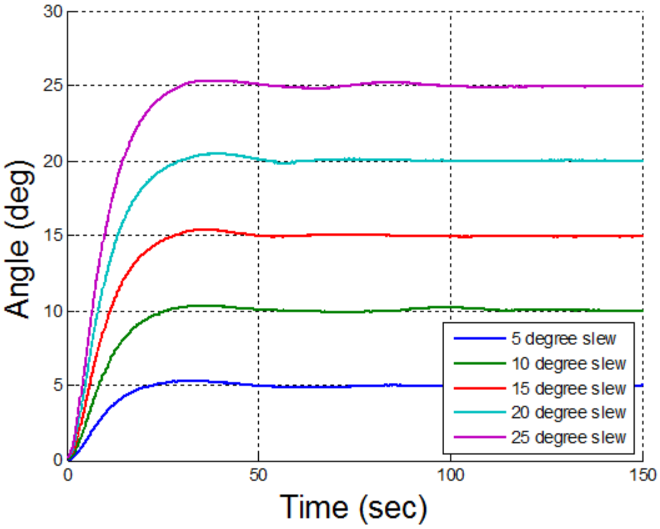


Figure 90. ADCS Testbed Time Response with Varying Commanded Angle

Notice that while the rise time is consistent between slews, the settling time and percent overshoot decrease with increasing slew. This is explained via the set point band created to decrease the amount of integral wind up caused by using K_i. The overshoot for the 5^o slew response is larger than the others because the torque applied from the non-linear PID controller to achieve the desired angle is low. The integral wind up for a 5^o slew will be larger proportionally to the low amount of torque applied to the reaction wheels; this will tend to increase overshoot and settling time for smaller slews. The integral wind-up will be of lesser effect on the transient response at larger slews; this results in lower percent overshoots. The rise time consistency shows the linear behavior of the ADCS testbed while the decreasing overshoot and settling time with increasing slew show the effects of the non-linear PID controller brought about from the set point band on K_i. Note that larger slews of up to 60^o, while not tested, would most likely continue the non-linear behavior of the ADCS testbed. Estimation accuracy was within $\pm 1^o$ for $\pm 25^o$ of slew angle so slew tests were performed only in this range.

The amount of control used to reach these angles is also of interest. It is important to know if the wheels come close to saturation when commanded to slew to large angles. Figure 91 illustrates the total change in reaction wheel 1's speed for each slew.

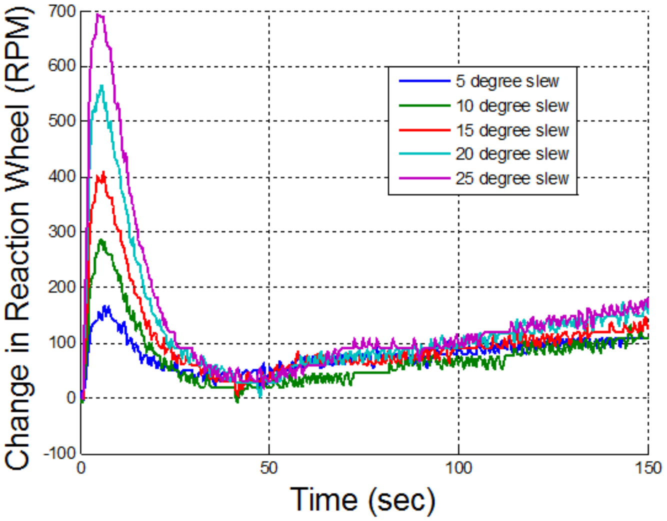


Figure 91. Reaction Wheel 1 Change in Wheel Speed with Varying Commanded Angle

For the ADCS testbed to slew to 25^o, it only takes a change in reaction wheel speed of just under 700 RPM for each reaction wheel. As discussed in Section 3.3.6, the MOI of the ADCS testbed is that of the chassis and the loading surface. This increases the MOI which means the reaction wheels will have to increase in speed to output the same response. Furthermore, only three reaction wheels are being used. If another reaction wheel was granted, control usage would decrease for each wheel. Notice that the reaction wheel keeps increasing once the ADCS testbed is settled at the desired angle; this is due to the fact that an external torque is inherent in AFIT's CubeSat testbed. The characterization of this external torque will be discussed further in Section 4.8. The next subsection will discuss the ADCS performance when subjected to a large disturbance torque.

4.7.2 ADCS Testbed Impulse Disturbance Torque Analysis

Pushing the ADCS testbed about the z-axis simulates an instantaneous (impulse) external torque for the reaction wheels to overcome. The ADCS testbed time response to repeated external torques is shown in Fig. 92.

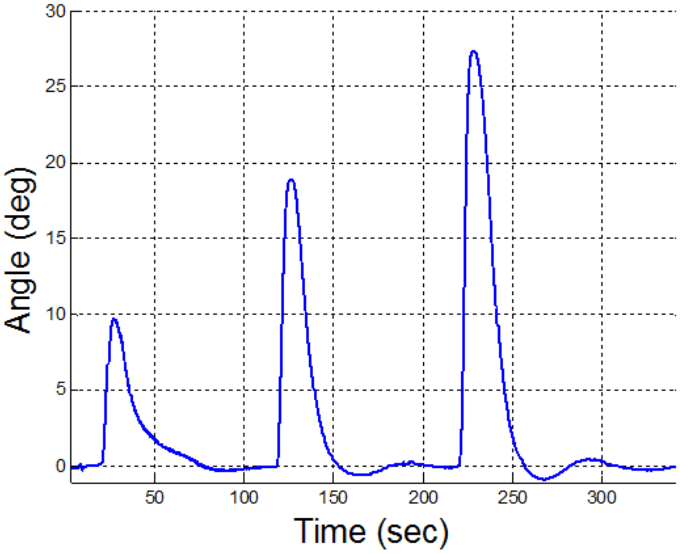


Figure 92. ADCS Testbed Time Response to Impact Disturbance Torque

The ADCS testbed was subjected to a 10^o, 20^o and 30^o impulse disturbance deflections. Figure 92 shows that the control was able to bring the ADCS testbed back to the desired angle of 0^o. The input of an external torque means that the change in angular momentum for the ADCS testbed is non-zero; This means that the reaction wheels will increase in speed to counteract the external disturbance torque. Figure 93 shows reaction wheel 1's change in speed throughout the disturbance test shown in Fig 92.

Due to the external disturbance torque, the reaction wheels never come back to the starting wheel speed of 1000 RPM. The change in speed from steady-state value before and after the disturbance is proportional to the applied external torque. Notice how the reaction wheels are at almost -4000 RPM after the third disturbance torque.

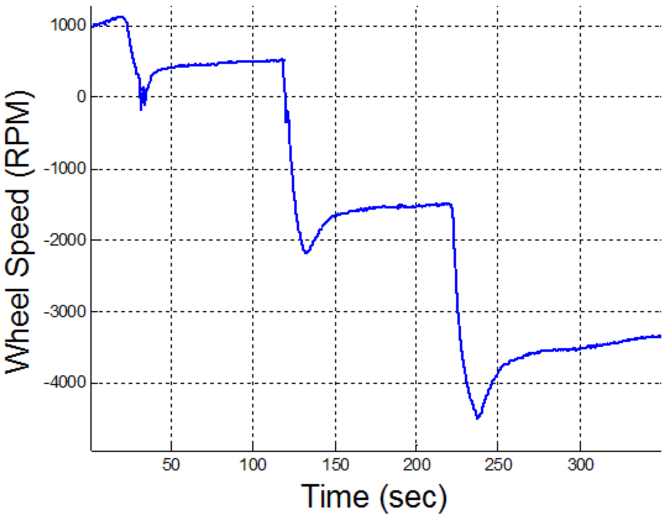


Figure 93. Reaction Wheel 1 Time Response to Impact Disturbance Torque

Eventually, the wheels would saturate; this would require a counteracting external torque such as torque coils to de-saturate the reaction wheels. Using torque coils to de-saturate the reaction wheels will be discussed in Section 5.3.

4.8 Angular Momentum/External Torque Analysis

As discussed in 3.5.8, the ADCS testbed will point at 0^o for 15 minutes with and without the trash bag covering the ADCS testbed (Fig. 50) to test the amount of external torque caused by air currents. The test without the trash bag was performed on three occasions to ensure the air current provides a consistent external torque. The third test without the trash bag over the ADCS testbed had the doors to the room closed where the CubeSat testbed is located. This was to ensure that air currents coming into the room from the hallways were not a factor. The trash bag was then pulled over the ADCS testbed and the test was run twice to ensure consistency. All three cases without the trash bag returned similar results and the two cases with the trash bag provided similar results. Figure 94 demonstrates the change in angular momentum of the following tests: the doors open, the doors closed, and the trash bag pulled over the ADCS testbed.

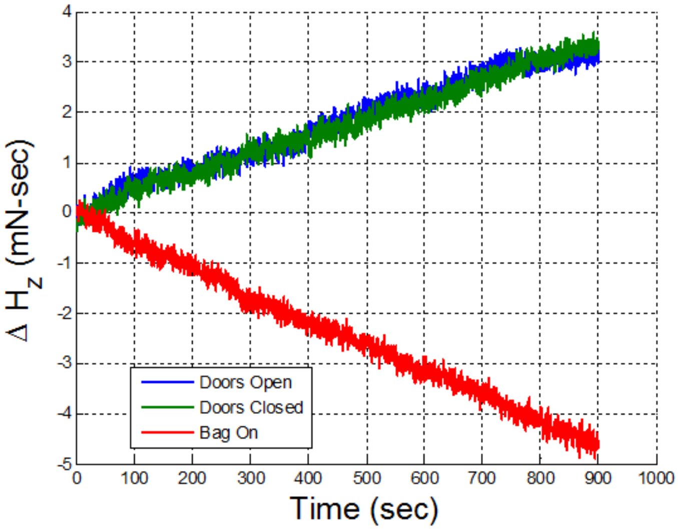


Figure 94. Change in Angular Momentum with and without Air Disturbance

Notice that the change in angular momentum $\Delta\ H_z$ with and without the doors closed is nearly identical. Therefore any air current coming from the hallway is assumed negligible towards the external torque on the ADCS testbed. Notice the striking difference in the angular momentum change with and without the trash bag. The angular momentum with and without the trash bag goes in the opposite directions. This means that the air current torque is larger in size and opposite in sign of the magnetic torque inherent in the ADCS testbed at 0^o. Table 8 outlines the external torque values calcluated from the gathered data using Eq. (70).

The air current torque is 1.39 times greater than the magnetic torque when the ADCS testbed is at 0^o. This brings the net total torque to 0.0036 mNm at 0^o which causes the reaction wheels to increase in speed. Note that the air current torque is 0.135 mNm. This torque did not affect the pointing accuracy of the ADCS testbed and is only 0.36% of the motor torque of one reaction wheel motor. Table 8 also shows the average magnetic torque at 10^o and -10^o. The magnetic moments of the ADCS testbed using Eq. (71) are 0.0485 A-m^2 in the y-axis and -0.1283 A-m^2 in the x-axis.

This makes the x-axis magnetic moment 2.65 times as large as the y-axis magnetic moment. The estimated magnetic torque varies with the z-axis angle as shown in Fig. 95.

Table 8. ADCS Testbed Characteristics for Varying Slews

Scenario	External Torque (mNm)
Doors Open	0.0036
Doors Closed	0.0038
Bag On	-0.0097
Magnetic Torque at 0^o	-0.0097
Magnetic Torque at 10^o	-0.0140
Magnetic Torque at -10^o	0.0104
Air Current Torque	0.0135

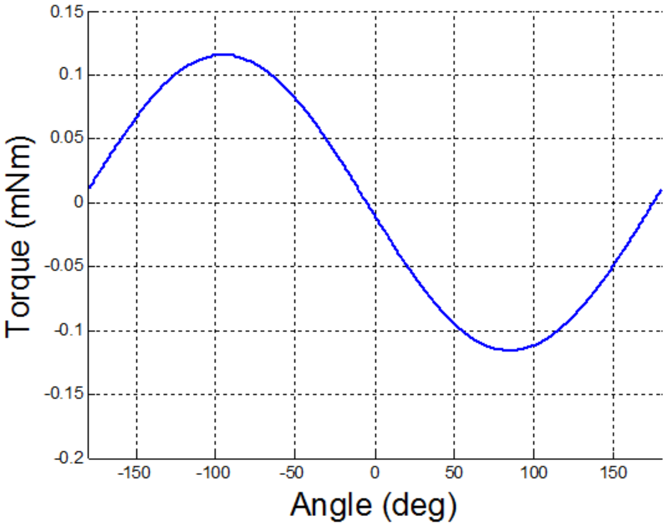


Figure 95. Estimated Magnetic Torque with Varying Angle

The estimation predicts that there will be no magnetic torque if the ADCS testbed is pointing at -5^o and the maximum magnetic torque can be 0.115 mNm. As previously stated, this assumes the magnetic moments are constant values. As will be discussed in Section 4.9.2, this is only a valid assumption for small angles around 0^o. Note that the air current torque, while not calculated as a function of z-axis angle, could be by testing the external torque at multiple locations and subtracting off the predicted magnetic torque using Fig. 95.

If the ADCS testbed was covered and stationary at 0^o with the reaction wheel's spinning at 5500 RPM, then the magnetic torque would saturate the wheels in roughly 2 hours and 8 minutes. Since the reaction wheels are typically spun to 1000 RPM, the saturation time would be roughly 1 hour and 15 minutes.

Covering the ADCS testbed eliminates most of the air current torques. A torque free environment could be achieved based on the manipulation of torque coils and the elimination of air currents. Furthermore, the magnetic torque is easier to predict and model which will help with predicting ADCS testbed behavior. The next section will compare the MATLAB simulation to the ADCS testbed sensor data.

4.9 ADCS Testbed Model Analysis

As discussed in Section 3.5.9, the following list details specific data used in an analytical model created in MATLAB script:

- ADCS testbed MOI (0.1593157 kg-m^2)
- Sensor noise
- External torque model
- Systemic time delay of 0.2 seconds
- Reaction wheel 1st order model with duty cycle conversion and a 201 time constant look-up table
- ADCS testbed z-axis model

First, the MOI of the reaction wheels and the ADCS testbed (Section 3.3.6) were implemented into the model. Next, sensor noise (Section 4.6) from the IMU and tachometer were characterized as a Gaussian distribution. The time delay and conversion from commanded reaction wheel speed to duty cycle was implemented. The reaction wheel model, which was modeled as a first-order system, includes 201 time constants between the three reaction wheels with a look-up table based on current and desired reaction wheel speed. The ADCS testbed model is for the *z*-axis only and will need to be expanded to three-axis. The following list shows the assumptions used to create the analytical model:

- Only external torque is magnetic
- Sensor noise assumed Gaussian
- Angular velocity in the *x*- and*y*-axes is negligible
- Reaction wheels behave as a first-order system with changing time constants
- The MOI of the ADCS testbed and reaction wheels are constant

The magnetic torque model calculated in Section 4.8 is used to predict the simulated external torque for the model. Section 4.6 shows that the noise of each state follows closely to a Gaussian distribution. Due to the same command for each reaction wheel, a small nutation occurs about the *x*- and *y*-axes. For short periods of time, however, this nutation is not visible. Therefore, for small slew demonstrations it is assumed that these angular velocities are zero. This means that the analytical model only simulates *z*-axis maneuvers which must be expanded to three-axis control in the future. Section 4.4 demonstrated the first-order behavior of the reaction wheels; this first-order system is modeled with a time constant look-up table. Finally, the MOI of the ADCS testbed and reaction wheels are assumed constant. Figure 96 shows a block diagram that illustrates how the model functions.

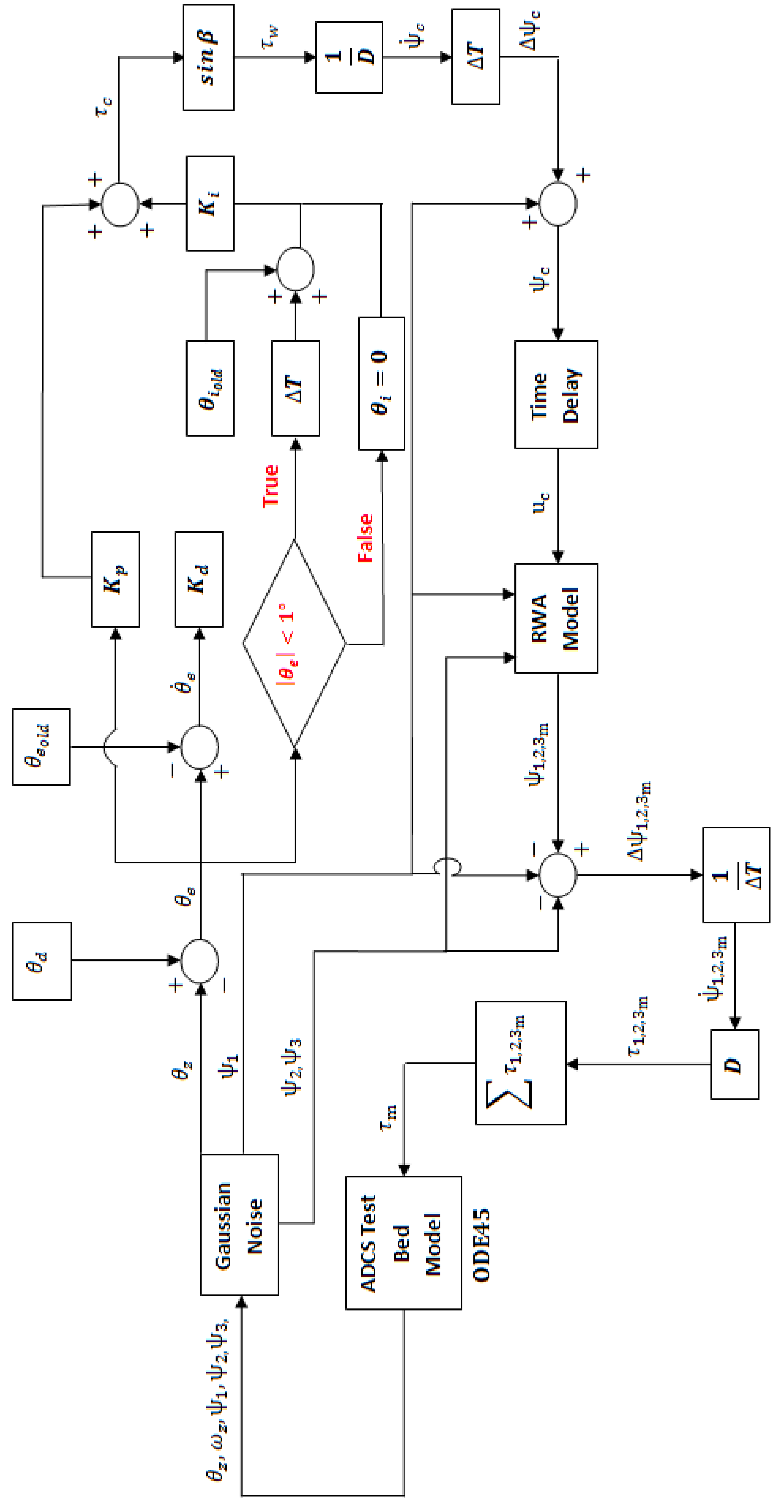


Figure 96. Analytical Model to Simulate ADCS Testbed Behavior

The model begins by adding Gaussian noise to each of the five states. The PID controller then operates on the z-axis angle error θ_e to output a desired torque τ_c. The desired torque is then converted to the change in commanded wheel speed $\Delta\psi_c$ which is added to reaction wheel 1's speed to output the desired wheel speed ψ_c. A time delay is inserted to delay the input into the RWA model shown in Fig. 97.

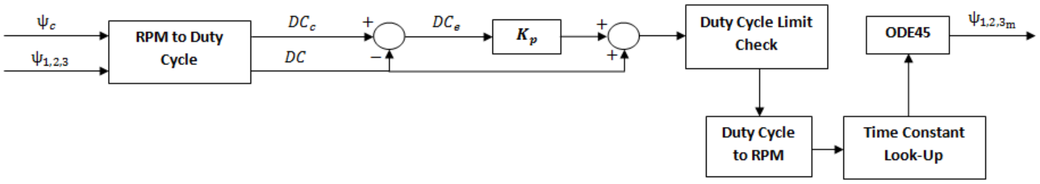


Figure 97. Analytical Model to Simulate RWA Behavior

The RWA model inputs the commanded and current wheels speeds and converts them to a duty cycle DC and commanded duty cycle DC_c. The duty cycle error DC_e is calculated and is multiplied by a proportional gain K_p. After going through the duty cycle limit check discussed in Section 4.4, the duty cycle is converted back to a wheel speed in RPM. The time constant look-up table is used to determine the best time constant based on current wheel speed and wheel speed error. The chosen time constant is used in ODE45 to propogate through Eq. (65) to output the actual wheel speed $\psi_{1,2,3_m}$. The actual wheel speed is differenced with the prior wheel speed to output the actual change in wheel speed $\Delta\psi_{1,2,3_m}$. The commanded torque to the ADCS testbed model is then calculated by converting $\Delta\psi_{1,2,3_m}$ to each wheel speed torque $\tau_{1,2,3_m}$ and summing them. The ADCS testbed model uses ODE45 to propagate through the following equations:

$$\dot{\theta}_z = \omega_z \tag{76}$$

$$\dot{\omega}_z = \frac{\tau_m}{\mathbf{C}} + \frac{M_{mag}}{\mathbf{C}} \tag{77}$$

$$\dot{\psi}_{1,2,3} = \frac{\tau_{1,2,3_m}}{\mathbf{D}} \tag{78}$$

These linear first-order differential equations calculate the state values and are fed back into the model shown in Fig. 96. The following subsections will include the tuning of the reaction wheel MOI, comparisons of time response characteristics for various slews between the simulated and actual responses, pointing accuracy predictions from simulation, and analyzing the effectiveness of external torque modeling.

4.9.1 Time Delay and Reaction Wheel MOI

The time delay of the system was calculated by comparing the commanded angle from the data to when the ADCS testbed responded. The average time delay observed from the data is 0.2 seconds. The reaction wheel MOI was chosen to be a design variable due to it never being measured like the ADCS testbed was. Figure 98 shows how the simulated speed of reaction wheel one changes with varying reaction wheel MOI.

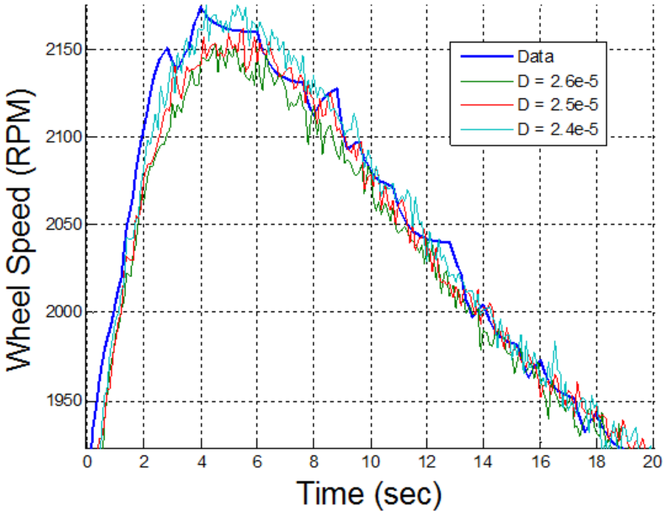


Figure 98. Simulated Reaction Wheel Response to Varying Wheel MOIs vs. Tachometer Data

The original specification given to the reaction wheel MOI was 2.6E-5 kg-m^2. Notice how using this MOI results in a simulation that does not match the peak of the data gathered from the tachometer during a 10^o slew. As the reaction wheel MOI is lowered, the reaction wheel speed must increase to retain the same response. When the reaction wheel MOI is 2.4E-5 kg-m^2, the simulation peak is very close to the actual peak. This test shows that the previous wheel MOI may be as much as 8% off. For this reason, the simulation will use a reaction wheel MOI of 2.4E-5 for the remainder of the tests.

4.9.2 Time Response Characteristics

Section 3.5.9 discussed that the slew test performed on the ADCS testbed in Section 4.7 would be simulated by the model. Note that these the tests were performed without anything covering the ADCS testbed, so air current torque was a factor. It is desirable to test the time response of the system with a cover for the ADCS testbed because the magnetic torque varies with changes to the z-axis angle whereas the air current torque is assumed constant. The air current torque is not modeled as well as the magnetic torque, so the data gathered from the slews by the ADCS testbed are all covered so that only the magnetic torque is active. Figures 99 through 101 illustrate a 10^o slew comparison between the ADCS testbed sensor data and the model.

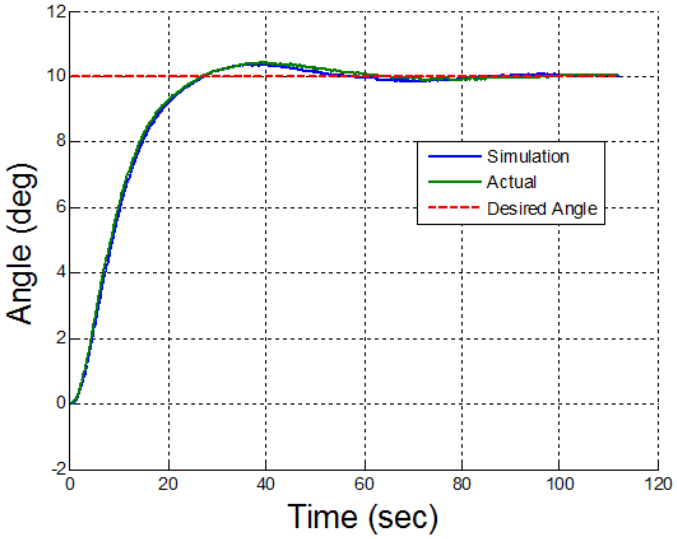


Figure 99. Attitude: Actual vs. Simulation

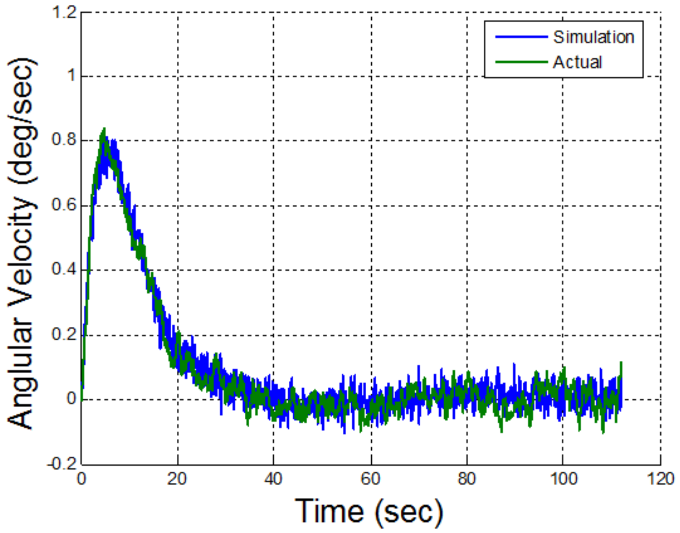


Figure 100. Angular Velocity: Actual vs. Simulation

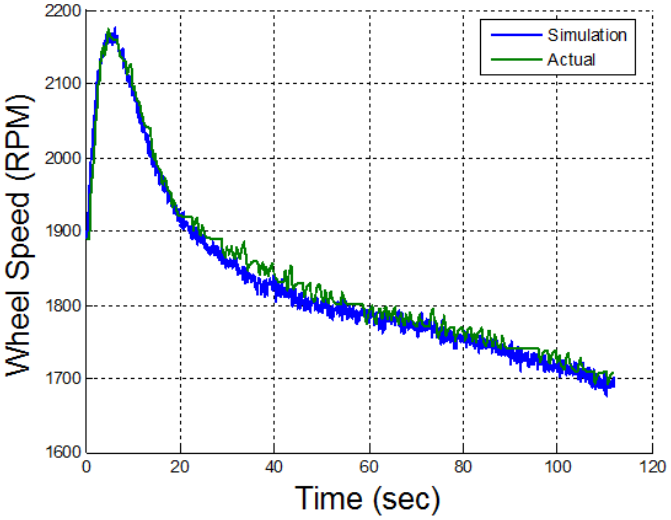


Figure 101. Reaction Wheel 1 Speed: Actual vs. Simulation

Taking the difference between the actual data and simulation leaves average errors of 0.079^o in the attitude response, 0.042^o/sec in the angular velocity response, and 13 RPM in the reaction wheel response. Notice that in Fig. 101 the reaction wheel speed decreases even though the attitude remains at around 10^o. This is caused by the magnetic external torque that was calculated in Section 4.8. The magnetic torque is counteracted by the decrease in reaction wheel speed. As mentioned in Section 3.5.9, tests were also performed for slews of 5, 15, 20 and 25^o. The rise time, percent overshoot, and settling time differences between the simulated and actual time responses for each slew are detailed in Table 9.

Table 9. Time Response Characteristics Percent Difference Between Sensor Data and Simulation

Slew (deg)	Rise Time Difference (%)	Overshoot Difference (%)	Settling Time Difference (%)
5	0.45	0.48	3.24
10	0.13	0.61	7.97
15	0.70	0.81	4.51
20	1.53	1.54	45.72
25	2.51	1.46	43.21

Notice that the time response characterisitcs difference between the sensor data and simulation increases with increasing slew. The rise time and overshoot differences are all less than 3% different whereas the settling time error at slews past 15^o shoots up to 45%. This may be due for a couple of reasons. First, the magnetic torque that was calculated in Section 4.8 does not correspond to the tachometer response when slewing to 25^o as shown in Fig. 102.

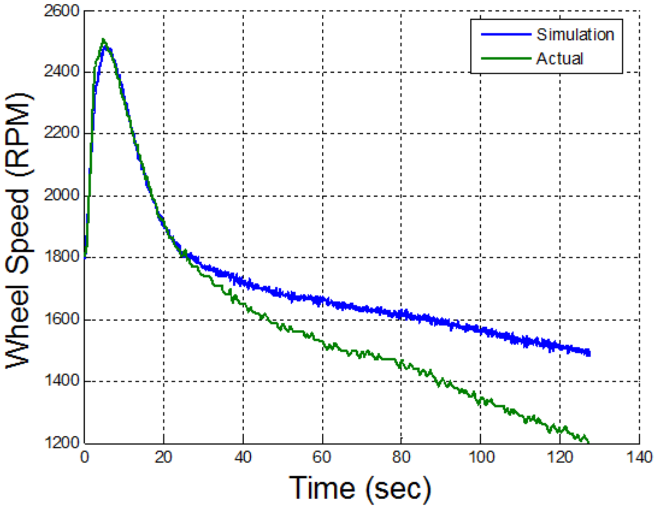


Figure 102. Reaction Wheel 1 Speed: Actual vs. Simulation for 25^o Slew

The simulated magnetic torque is not as strong as the actual magnetic torque experienced by the ADCS testbed. This may be caused by a changing magnetic moment that is related to reaction wheel speed. This larger external torque may cause slightly higher overshoot and delay the settling process.

The second and most important reason that the time response settling times and percent overshoots are different between the simulation and actual response is due to the lack of a truth measurement. Without a truth measurement, the accuracy of the simulation is a function of the accuracy of the sensor. For all that we know, the simulation could be closer to the truth measurements than the actual sensor measurements. Once a truth measurement device is implemented, the model and sensors can both be adjusted so that they match. Without a truth measurement device, the simulated and sensor responses accuracy is left up to just speculation. The only case where this is not true is with steady-state measuring because of angle markings on the wall of the CubeSat testbed; and even this method has inherent human error.

Another comparison between the simulation and ADCS testbed responses is the pointing accuracy. Recalling back to Section 4.7, the pointing accuracy of the ADCS testbed is 0.07^o. By comparison, the simulated pointing accuracy is 0.09^o. This means the error between the sensed and simulated pointing accuracy is 0.02^o.

4.10 Chapter Summary

Chapter IV presented the results and analysis for the various tests discussed in Section 3.5. First the gyroscope and magnetometer calibrations and filters were created and analyzed. The estimation accuracy using the IMU was calculated to be 0.1^o (3σ). The Chapter continued by characterizing and designing the controller for the RWA. The reaction wheel speed was found to have a linear relationship with the duty cycle count. The controller for the RWA uses a proportional gain and is modeled as first-order system with the use of 201 time constants. Next, the sensor noise of the attitude, angular velocity, and tachometer were characterized for implementation into the MATLAB model. The Chapter then presented the testing and analysis of the control design and performance of the ADCS testbed. The ADCS testbed uses a PID controller that has a pointing accuracy of 0.07^o. The derivative gain is 10 times the proportional gain while the integral gain is 10 times less than the proportional gain. Slewing the ADCS testbed to commanded angles showed asymptotic stability and linear behavior with a second order system response. The ADCS was then given impulse disturbance torques to overcome. The ADCS testbed was able to overcome a 10^o, 20^o, and 30^o disturbance deflection without reaction wheel saturation. The external torque inherent in the CubeSat testbed was then calculated. It was discovered that a strong air current torque was able to overcome the magnetic torque inherent in the ADCS testbed. The air current torque was assumed constant while the magnetic torque was characterized as a function of the z-axis angle. The Chapter concluded with a comparison of the ADCS testbed sensor data versus a MATLAB model time response. The model's time response characteristics were very close to the sensor data at low slew angles. The simulated pointing accuracy was calculated with a variant of only 0.014^o from the pointing accuracy gathered from real data. Differences in the model start to be clearer at larger slews. This was due to the lack of a truth measurement and the magnetic torque model accuracy at larger slews. Overall, the pointing accuracy of the ADCS testbed pointing accuracy was improved from $\pm 20^o$ to 0.07^o and the MATLAB model shows similar performance to the sensor data.

V. Conclusions and Recommendations

5.1 Research Summary

The goal of this research effort discussed in Chapter I was to investigate the level of the ADCS's z-axis control authority using only a RWA for actuation and an IMU for determination validated by using AFIT's CubeSat testbed and and utilizing data gathered by these tests to determine a model's simulation accuracy. The 6U CubeSat meets the needs of DoD related projects that would be more cost effective than using larger satellites. AFIT researchers are currently designing a 6U and 12U CubeSat. CubeSat testbeds that consist of a Helmholtz cage and air bearing are a concept taken from satellite simulators; the characterization and advancement of these testbeds could greatly improve on-the-ground testing of CubeSat ADCS. This contribution documents the first time, to the best of the author's knowledge, a 6U ADCS testbed's validation and achievement of single-axis control inside a Helmholtz cage atop an air bearing. Furthermore, this is the first time a CubeSat testbed's inherent external torques have been tested and characterized.

Background research was performed in Chapter II to gain an understanding of attitude determination and control. The kinematic and kinetic EOM describing satellite motion were derived with the use of coordinate frames, euler angles, and quaternions. The QUEST algorithm was used to estimate a current quaternion with inputs from the magnetometer and accelerometer. The PID controller was derived and implemented into the satellite EOM; this implementation demonstrated the closed loop dynamics of the kinetic EOM. These dynamics were used to create the MATLAB model of the ADCS testbed. Finally, a literature review helped establish past and current contributions related to this research effort.

The CubeSat testbed and ADCS testbed hardware, software, and algorithms were explained in detail in Chapter III. The hardware configuration of the ADCS testbed includes a ADCS board, motor controller board, RWA, EPS, battery pack, and a CDH with WiFly. The ADCS control algorithm uses a PID controller and z-axis angle error to output a desired reaction wheel speed for the RWA control algorithm.

Various tests were performed to find the level of 6U ADCS performance with only a RWA and IMU using AFIT's CubeSat testbed and how the MATLAB model compared to ADCS sensor data. The estimation accuracy improved from $\pm 3^o$ to $\pm 0.02^o$ (3σ) with constant reaction wheel speed. It was found that an air current torque and a magnetic torque are are being applied to the ADCS testbed. The air current torque is 1.39 times greater than the magnetic torque when the ADCS testbed is at 0^o. The ADCS testbed controller was designed and its performance was compared to MATLAB simulation. The pointing accuracy improved from $\pm 20^o$ to 0.07^o (3σ) and slewing the ADCS testbed resulted in asymptotic stability and linear behavior. The ADCS testbed was able to overcome large disturbance torques without saturation of the reaction wheels. The MATLAB model showed similar time response and steady-state characteristics to the gathered data.

5.2 Conclusions

This research effort outlined testing that needed to be performed in Section 1.3 so that the level of ADCS performance using only a RWA and IMU could be characterized with the use of AFIT's CubeSat testbed for on-the-ground testing. The following list presents the research topics discussed in Section 1.3 along with the author's conclusions about each respective topic:

1. Establish wireless communication with the ADCS testbed so that real time data can be streamed

   Before this research effort began, telemetry was obtained from the ground station *at 1 Hz. At the start of this research effort, only wired telemetry at 10 Hz was available. Although this was useful for determination testing, control testing was near impossible due to the external torque applied by the wire. Wireless telemtry was achieved by wiring an ADCS to a CDH board with WiFly capability. The WiFly is able to send all telemetry to Tera Term where it can be easily converted into MATLAB for data analysis.*

2. Achieve a high level of attitude determination using an IMU

   Along with the QUEST algorithm, the ADCS testbed uses magnetometer and acclerometer sensor measurements to estimate its attitude. The magnetometer was filtered to account for angle discrepancies and reaction wheel magnetic field distortions. The estimation accuracy improved from $\pm 3^o$ *to* $\pm 0.1^o$ *(*3σ*) with any reaction wheel speed. The estimation accuracy at constant reaction wheel speed is* $\pm 0.02^o$*. When comparing to pointing accuracy, the constant reaction wheel speed estimation accuracy specification should be used considering the small amount of external torque applied to the ADCS testbed; the small torque applied keeps the reaction wheels near constant speed. The estimation algorithm can sense accurately to within* $\pm 1^o$ *when the ADCS testbed is within* $\pm 20^o$ *of* 0^o*.*

3. Establish control of the z-axis

   Before control design, the RWA required duty cycle calibration. It was found that the reaction wheel speed and duty cycle count produced a linear relationship. Control design consisted of a PID controller for the ADCS testbed and a proportional controller for the RWA. The derivative gain for the PID controller is 10 times the proportional gain, and the integral gain is 1/10 of the proportional gain. The RWA proportional gain does not increase in transient performance past K_p *equal to 9. The inclusion of an integral gain in the RWA controller caused undesirable high overshoot and long settling times. Slewing the*

ADCS testbed demonstrated asymptotic stability with a second order response. Linear behavior was observed due to similar time response characteristics for varying slews. The ADCS testbed was able to overcome impulse disturbance torques without wheel saturation. The pointing accuracy of the ADCS testbed improved from ±20° to ±0.07° (3σ).

4. Characterize external torques in AFIT's CubeSat testbed

   The external torque applied to the ADCS testbed was calculated by observing the change in reaction wheel speed while the ADCS testbed pointed at a commanded angle. The air current torque and magnetic torque were estimated. The magnetic torque was characterized as a function of z-axis angle whereas the air current torque was assumed constant. The magnetic torque can apply estimated external torques between ±0.05 mNm within a linear attitude range. The air current torque applies 0.0135 mNm of torque at 0°. Overall, it would take the ADCS testbed roughly 1 hour and 15 minutes to saturate when pointing within a linear range.

5. Create a model to accurately simulate ADCS behavior

   The MATLAB model uses linearized kinetic and kinematic EOM to simulate ADCS testbed behavior. The model also includes a 1st order system approximation of the RWA that includes a 0.2 sec time delay and 201 time constants. The reaction wheel speed to duty cycle conversion was also implemented. Simulated sensors are not used in the model; however, noise was added to state outputs to mimic the IMU. The control algorithm for the ADCS and RWA are the same as is used on the ADCS testbed. The PID gains are also the same as is used on the ADCS testbed. The external torque estimation includes a constant air current torque and a magnetic torque that depends on the z-axis angle. The reaction wheel MOI was changed from 2.6E-5 to 2.4E-5 to better match reaction wheel *data from the tachometer. The transient response characteristics, such as rise time, percent overshoot, and settling time matched closely with actual data for small slews less than or equal to 15^o. Slews up to 25^o resulted in larger settling time and percent overshoot differences between simulation and sensor data. This may be due to the external magnetic torque modeling imperfections and the lack of a truth sensor measurement. Only a truth measurement device will detail whether the simulation or the sensors accurately portray ADCS testbed behavior better. The simulated pointing accuracy of 0.09^o is only 0.02^o different than the pointing accuracy determined by the sensors.*

5.3 Recommendations for Future Work

The following subsections present recommended work that is in the best interest of reaching the goal of obtaining full three-axis control of the 6U ADCS testbed validated using AFIT's CubeSat testbed.

5.3.1 Addition of an External Magnetometer

The estimation accuracy of just using the IMU is less than 1^o within $\pm 20^o$ of z-axis angle. The RWA, which is only centimeters away from the IMU, causes distortions in the magnetic field. This can generate undesirable jumps in angle estimation outside of 30^o. The addition of a filtered external magnetometer that is placed far away from the RWA could increase the estimation accuracy of the IMU and help better characterize the magnetic field distortions caused by the RWA. Once filtered, the magnetic field data from the external magnetometer can be inputted into QUEST, and one can compare estimation accuracy based on changing the weighting between the IMU and external magnetometers. The inclusion of an external magnetometer could smooth the estimated angle and improve pointing accuracy at larger slews. This research effort explored external magnetometers, but due to sensor accuracy issues, the project was abandoned and was not included in the thesis.

5.3.2 Implementation of a Kalman Filter

A Kalman filter is a proven way to give a more accurate and less noisy state estimate by taking into account sensor data, system dynamics, and control inputs. The author has already created a Kalman filter in MATLAB and C code. The implementation of a Kalman filter reduced state estimation noise and mimicked system behavior, but the state estimation lagged the actual sensor data which drastically decreased control authority. It is the author's recommendation that the estimation and control tasks be split so that the estimation algorithm can run much faster than the control algorithm. The slow convergence rate of the Kalman filter may be due to the estimation and control tasks running at the same speed of 10 Hz. This could be tested by changing the estimation algorithm to 50 Hz and keeping the control algorithm to 10 Hz. Due to the failure of the Kalman filter implementation, the results were not presented in this thesis.

5.3.3 Developing a New RWA

This thesis effort originally began with a three wheel RWA designed to pass vibrations testing. After the conclusion of characterizing this RWA, it was determined that there were too many problems that could not be overcome to gain asymptotic stability of the ADCS testbed. The RWA contained large amounts of friction in the bearings which in turn caused a dead-band in the reaction wheel speed to duty cycle conversion. When designing the controller, the proportional gain values could not be very high; otherwise, the tachometer would display highly inaccurate values. This means that full control of the reaction wheels was difficult to achieve which corresponded to $\pm 5^o$ of pointing accuracy for the ADCS testbed. It was found later that the wrong motors were used for the reaction wheels; at this point these wheels were replaced by the older generation four wheel RWA.

A new RWA can be constructed that iterates from the three wheel RWA design. By lowering friction and replacing the motor, the new RWA should be able to be controlled. Furthermore, the three wheel RWA encased the motors which limited magnetic field distortion. In fact, only a slewing linear filter was needed for the magnetometer, and pointing accuracy was superior with the three wheel RWA than with the four wheel pyramid RWA. If the new RWA can be controlled, then encasing the reaction wheel motors to limit magnetic field distortion will greatly increase estimation accuracy.

5.3.4 Truth Measurement Implementation

When comparing the ADCS testbed sensor output to simulation, the time response characteristics within a linear range are very close. Although the sensors were tested for static accuracy based on angle labeled on the wall, dynamic accuracy was not possible due to lack of a truth measurement device. A truth measurment device, such as a camera system, could be mounted to the Helmholtz cage to accurately give attitude estimates within 0.01^o. With full three-axis knowledge of attitude, robust filters can be implemented within the ADCS testbed estimation algorithm that would increase estimation authority and accuracy. The truth measurement device can also be compared to model simulation which in turn can make the model more robust. This will also allow robust filtering of the magnetometer using a non-constant magnetic field simulated by the Helmholtz cage.

5.3.5 ADCS Testbed Cover Implementation

Consistency of the ADCS testbed is very important so that repetitive tests will produce the same time response and pointing accuracy. With a cover over the ADCS testbed, air current torques can be neglected and only a magnetic torque can be assumed. The magnetic torque can then be fully characterized to ensure that the Kalman filter and the MATLAB model are more accurate. The testbed cover could be of a non-ferrous material, such as Plexiglas, that is box shaped and goes over the ADCS testbed. The cover could be designed with a door for easy access to make changes to the ADCS testbed. Another solution would be to instead cover the entire CubeSat testbed. This implementation could be of medium weight flexible vinyl fixed to the 80-20 tracks of the Helmholtz cage. this cover could make it easier to modify the ADCS testbed when comparing it to the cover just over the ADCS testbed.

5.3.6 Magnetometer Filter for Simulated Orbit

Filtering the magnetometer for a simulated magnetic field seen on orbit is important for achieving a space-like environment in AFIT's CubeSat testbed. Due to the constantly changing magnetic field, knowledge of orbital mechanics and orbital magnetic fields is required. The orbital dynamics could be estimated within the ADCS algorithm as well a look-up table or equations governing the respective orbital position's magnetic field. With the orbit knowledge, the ADCS testbed should be able to point at a desired angle, and the angle should not drift in the changing magnetic field if the estimated orbital magnetic field is properly modeled. Slewing the ADCS testbed can also be tested for accuracy with the help of a truth measurement device that is independent of the changing magnetic field.

5.3.7 Momentum Dumping Capability with Torque Coils

Saturation of the reaction wheels can occur due to inherent external torques in the CubeSat testbed. The implementation of torque coils can make it possible to de-saturate the reaction wheels so longer tests can be performed. AFIT's torque coils have already been characterized and tested [8]. Dumping the ADCS testbed's angular momentum using torque coils will add an extra external torque to the kinetic EOM. The use of torque coils will be a controlled external torque that will battle against the reaction wheel's de-saturation and the magnetic torque already inherent in the ADCS testbed. Note that the magnetometer will cease to be a valid source of attitude estimation, so having another means of attitude sensing such as an integrated angular velocity from the IMU gyroscope is recommended.

5.3.8 Develop and Test ADCS Control Algorithms

Although the PID controller is an effective controller to output a desired response, other control algorithms exist that may perform better or make the control more efficient. Different linear controllers such as ones developed from root-locus, Bode, or Nyquist techniques can be used to take advantage of the near linear system observed for z-axis control of the ADCS testbed. Furthermore, optimal linear controllers such as a linear quadratic regulator could be used to minimize certain specifications such as power consumption. Optimal techniques used in outer loop controllers could help establish the most efficient path for the ADCS testbed to reach a desired orientation. Non-linear controllers such as adaptive and sliding could be used to counter non-linear effects inherent in the ADCS testbed. These controllers could also be compared in MATLAB simulation to the existing model before actually testing the algorithm on hardware. These tests will help further the development of the 6U ADCS control algorithms and contribute to AFIT's CubeSat testbed as a valid proving ground for

CubeSat ADCS control algorithm development.

5.3.9 Dynamically Neutral Behavior of the ADCS Testbed/Loading Surface

The ADCS testbed is currently only shows statically neutral behavior atop the air bearing. Statically neutral behavior is described as the center of mass of the ADCS testbed aligned with the center of rotation of the hemi-sphere about the x- and y-axes. Dynamically neutral behavior means that the center of mass is coincident with the center of rotation. With statically neutral behavior, there is a pendulum motion about the x- and y-axes due to a difference in height of the center of mass and center of rotation. This gravitational torque will cause the reaction wheels to saturate very quickly if commanded to rotate about either the x- or y-axes. Gaining dynamically neutral behavior will eliminate pendulum motion and gravitational torques, making it possible to test the ADCS for three-axis control. The loading surface is designed to be able to create dynamically neutral behavior. Statically neutral behavior is achieved by translating the ADCS testbed along the loading surface in the x-y plane. The threaded holes in the loading surface are designed to allow the height of the ADCS testbed to be altered enabling dynamically neutral behavior. A good starting point would be to study the NPS's design of an automated mass balancing system for a CubeSat testbed that would allow for three degrees of freedom [21]. Once this is accomplished, three-axis attitude estimation and control can be achieved.

5.3.10 Model Updates

Although the model simulation was able to closely match sensor data for small slews, the model will need to be updated frequently to continue outputting accurate results. The addition of an external magnetometer may change the attitude estimation accuracy of the ADCS testbed sensors which must be accounted for in the model. Although a Kalman filter is already built into the model, it must be updated to account for a faster estimation algorithm that will be implemented in the C code. The implementation of a new RWA will require the re-testing of the external magnetic torque experienced by the ADCS testbed. This torque will need be changed with each iteration. The MOI will also need to be re-measured with any additions to the ADCS testbed which could include extra sensors or different actuators. The addition of a truth measurement sensor will require a re-evaluation of simulation performance. Orbital dynamics equations along with a look up table or equation of magnetic fields must also be implemented to properly simulate the magnetic field. Creating sensors by simply adding noise may not be a liable assumption in a constantly changing magnetic field. The addition of torque coils will be needed to simulate momentum dumping maneuvers. Any new control algorithms will have to also be added. If the ADCS testbed achieves dynamic stability with the air bearing, then the nonlinear equations of motion must be re-established in the model to accurately predict ADCS testbed performance.

Bibliography

[1] Bhavani, K.H. and R.P. Vancour. *Coordinate systems for space and geophysical applications.* Technical report, Air Force System Command, Hanscom Air Force Base, 1991.

[2] Blackwell, William, G Allen, C Galbraith, R Leslie, I Osaretin, M Scarito, Mike Shields, E Thompson, D Toher, D Townzen, et al. "MicroMAS: A First Step Towards a Nanosatellite Constellation for Global Storm Observation". *Conference on Small Satellites*, 2013.

[3] Brewer, Megan R. *CubeSat Attitude Determination and Helmholtz Cage Design.* M. S. thesis, Air Force Institute of Technology, Wright-Patterson AFB, March 2012.

[4] Cai, Guowei, Ben M Chen, and Tong Heng Lee. *Unmanned Rotorcraft Systems.* Springer., London, England, 2011.

[5] Chesi, Simone, Qi Gong, Veronica Pellegrini, Roberto Cristi, and Marcello Romano. "Automatic Mass Balancing of a Spacecraft Three-Axis Simulator: Analysis and Experimentation". *Journal of Guidance, Control, and Dynamics*, 37(1), January-February 2014.

[6] Chesi, Simone, Octavio Perez, and Marcello Romano. "A Dynamic Hardware-in-the-Loop Three-Axis Simulator of Nanosatellite Dimensions". *To appear, 2015. Journal of Small Satellites*, N/A(N/A):N/A, N/A.

[7] Cobb, Richard G. "Control and State Space Concepts", 2014. AFIT Course ASYS 565 Lecture Slides.

[8] Dannemeyer, Erin. *Design and Analysis of an Attitude Determination and Control Subsystem (ADCS) for AFIT's 6U Standard Bus.* M. S. thesis, Air Force Institute of Technology, Wright-Patterson AFB, March 2014.

[9] Devices, Analog. "Triaxial Inertial Sensor and Magnetometer ADIS 16405". `http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16400_16405.pdf`.

[10] Foley, Justin Dean. *Calibration and Characterization of CubeSat Magnetic Sensors using a Helmholtz Cage.* M. S. thesis, California Polytechnic State University, 2012.

[11] Grewal, Mohinder S, Lawrence R Weill, and Angus P Andrews. *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley & Sons, 2007.

[12] Hall, Christopher D. "Spacecraft Attitude Dynamics and Control", 2011. Virginia Polytechnic Institute and State University.

[13] Hartsfield, Carl R. “New 6U CubeSat Reaction Wheel Array”, 2014. Air Force Institute of Technology.

[14] Johnson, Samuel. *Design of a Control Moment Gyroscope Attitude Actuation System for the Attitude Control Subsystem Proving Ground.* M. S. thesis, Air Force Institute of Technology, Wright-Patterson AFB, March 2013.

[15] Kirk, Donald E. *Optimal Control Theory, An Introduction.* Dover., Mineola, N.Y., 1998.

[16] Kunz, Donald L. *Intermediate Dynamics for Aeronautics and Astronautics.* Air Force Institute of Technology, 2013.

[17] Li, Junquan, Mark Post, Thomas Wright, and Regina Lee. “Design of attitude control systems for cubesat-class nanosatellite”. *Journal of Control Science and Engineering*, 2013:4, 2013.

[18] Mathworks. “MATLAB Document: Smooth”, 2014. `http://www.mathworks.com/help/curvefit/smooth.html`.

[19] Mathworld. “Least Squares Fitting–Polynomial”. `http://mathworld.wolfram.comLeastSquaresFitting.html`.

[20] McChesney, Christopher. *Design of Attitude Control Actuators for a Simulated Spacecraft.* M. S. thesis, Air Force Institute of Technology, Wright-Patterson AFB, March 2012.

[21] Meissner, David M. *A Three Degrees of Freedom Test Bed for Nanosatellite and CubeSat Attitude Dynamics, Determination, and Control.* M. S. thesis, Naval Postgraduate School, Monterey,CA, 2009.

[22] Monarch. “Pocket Laser Tach 200 Digital Contact/Non-Contact Tachometer”.

[23] Nelson, Robert C. *Flight Stability and Automatic Control*, volume 2. WCB/McGraw Hill, 1998.

[24] Ogata, Katsuhiko. *Modern Control Engineering Fifth Edition.* Prentice-Hall Englewood Cliffs, NJ, 2010.

[25] Penn, Dylan. *Characterization and Modeling of a Control Moment Gyroscope.* M. S. thesis, Air Force Institute of Technology, Wright-Patterson AFB, March 2015.

[26] Prinkey, Meghan, DW Miller, P Bauer, K Cahoy, ED Wise, CM Pong, RW Kingsbury, AD Marinan, HW Lee, and E Main. “CubeSat Attitude Control Testbed Design: Merritt 4-Coil per axis Helmholtz Cage and Spherical Air Bearing”. *manuscript in preparation for AIAA GNC August, Boston, MA*, 2013.

[27] Reiner, Jacob, Gary J Balas, and William L Garrard. “Robust Dynamic Inversion for Control of Highly Maneuverable Aircraft”. *Journal of Guidance, Control, and Dynamics*, 18(1):18–24, 1995.

[28] Sanders, Devon S, Daniel L Heater, Steven R Peeples, Po-Hao Adam Huang, et al. “Pushing the Limits of Cubesat Attitude Control: A Ground Demonstration”. *Conference on Small Satellites*, 2013.

[29] Schwartz, Jana L, Mason A Peck, and Christopher D Hall. “Historical Review of Air-Bearing Spacecraft Simulators”. *Journal of Guidance, Control, and Dynamics*, 26(4):513–522, 2003.

[30] Sellers, Jerry Jon, William J Astore, Robert B Giffen, and Wiley J Larson. *Understanding Space: An Introduction to Astronautics*. Primis, 2000.

[31] Shuster, Malcolm David and SD Oh. “Three-axis Attitude Determination from Vector Observations”. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.

[32] Sidi, Marcel J. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge university press, 1997.

[33] Slotine, Jean-Jacques E, Weiping Li, et al. *Applied Nonlinear Control*. Prentice-Hall Englewood Cliffs, NJ, 1991.

[34] Smith, Steven W et al. “The Scientist and Engineer’s Guide to Digital Signal Processing”. *California Technical Pub. San Diego*, 1997.

[35] Studio, Atmel. “AVR4018: Inertial Two (ATAVRSBIN2) Hardware User’s Guide”. `http://www.atmel.com/Images/doc8369.pdf`.

[36] Swartwout, Michael. “The First One Hundred CubeSats: A Statistical Look”. *JoSS 2pp*, 213–233, 2014.

[37] Swenson, Eric D. “6U CubeSat - Design, Build, Test”, 2014. Air Force Institute of Technology.

[38] Swenson, Eric D. “Angular Velocity Kinematics”, 2014. AFIT Course MECH 632 Lecture Slides.

[39] Swenson, Eric D. “Attitude Control Part 1”, 2014. AFIT Course MECH 632 Lecture Slides.

[40] Swenson, Eric D. “Attitude Control Part 2”, 2014. AFIT Course MECH 632 Lecture Slides.

[41] Swenson, Eric D. “Euler Angles”, 2014. AFIT Course MECH 632 Lecture Slides.

[42] Swenson, Eric D. "Euler's Equations", 2014. AFIT Course MECH 632 Lecture Slides.

[43] Swenson, Eric D. "Intermediate Spacecraft Dynamics Introduction", 2014. AFIT Course MECH 632 Lecture Slides.

[44] Swenson, Eric D. "Magnetic Torques", 2014. AFIT Course MECH 632 Lecture Slides.

[45] Swenson, Eric D. "Notation, Reference Frames, and Rotation Matrices", 2014. AFIT Course MECH 632 Lecture Slides.

[46] Swenson, Eric D. "Reaction Wheels", 2014. AFIT Course MECH 632 Lecture Slides.

[47] Swenson, Eric D. "Rigid Body Dynamics", 2014. AFIT Course MECH 632 Lecture Slides.

[48] Team, NASA Mission Design. "Small spacecraft technology state of the art". *NASA*, 2014.

[49] University, Embry Riddle. "Helmholtz Coils". `http://physicsx.pr.erau.edu/HelmholtzCoils`.

[50] Ure, N Kemal, Yigit Bekir Kaya, and Gokhan Inalhan. "The development of a Software and Hardware-in-the-Loop Test System for ITU-PSAT II nano satellite ADCS". *Aerospace Conference, 2011 IEEE*, 1–15. IEEE, 2011.

[51] Voyles, Richard M, JD Merrow, and Pradeep K Khosla. "Including Sensor Bias in Shape from Motion Calibration and Sensor Fusion". *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on*, 93–99. IEEE, 1996.

[52] Wertz, James Richard. *Spacecraft Attitude Determination and Control*, volume 73. Springer Science & Business Media, 1978.

[53] Wertz, James Richard, David F Everett, and Jeffery John Puschell. *Space Mission Engineering: The New SMAD*. Microcosm Press, 2011.

[54] Wie, Bong. *Space Vehicle Dynamics and Control Second Edition.* American Institute of Aeronautics and Astronautics, 2008.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 26-03-2015	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Oct 2013 — Mar 2015

4. TITLE AND SUBTITLE

Design and Test of an Attitude Determination and Control System for a 6U CubeSat using AFIT's CubeSat Testbed

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Tibbs, Michael, L, 2nd Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-MS-15-M-240

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space Vehicles Directorate
Air Force Research Laboratory
3550 Aberdeen SE, Bldg 497
Kirtland AFB, NM 87117
Timothy J. Shuck, Capt, USAF. (505) 853-4513, timothy.shuck@kirtland.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RVES

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A. Approved for Public Release;Distribution Unlimited

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States

14. ABSTRACT

The design and test of a 6U CubeSat Attitude Determination and Control System (ADCS) are explored to establish single-axis control using AFIT's CubeSat testbed consisting of a Helmholtz cage and hemi-spherical air bearing. The Helmholtz cage produces a near-uniform magnetic field inside the cage while the air bearing provides a near-frictionless surface for ADCS testing. The ADCS testbed includes a four wheel pyramid reaction wheel array (RWA) for actuation and an inertial measurement unit (IMU) for attitude determination. Along with the ADCS hardware, the ADCS testbed also includes an Electrical Power System (EPS) and Command and Data Handling (CDH) for power and wireless telemetry, respectively. Attitude estimation is performed using the QUEST algorithm with magnetometer and accelerometer sensor data to estimate a current quaternion. A Proportional-Integral-Derivative (PID) controller is used for control of the ADCS testbed while each reaction wheel motor is controlled by a proportional gain. After calibration, the static estimation accuracy improved from $\pm 3^o$ to $\pm 0.02^o$ (3σ). The RWA is characterized as a 1st order system with a systemic 0.2 second time delay. The ADCS testbed controller demonstrates linear time response characteristics at small angle slews. Through this research effort, the pointing accuracy was improved from $\pm 20^o$ to $\pm 0.07^o$ (3σ). External torques caused by air currents and inherent magnetic moments were explored and characterized. The inherent magnetic moment of the ADCS testbed was calculated. Simulated time response characteristics of the model showed similar behavior to the ADCS testbed results.

15. SUBJECT TERMS

CubeSat, ADCS, RWA, Helmholtz Cage, Air Bearing, CubeSat Testbed, ADCS Testbed

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Eric D Swenson, AFIT/ENY
U	U	U	UU	176	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x7479; eric.swenson@afit.edu