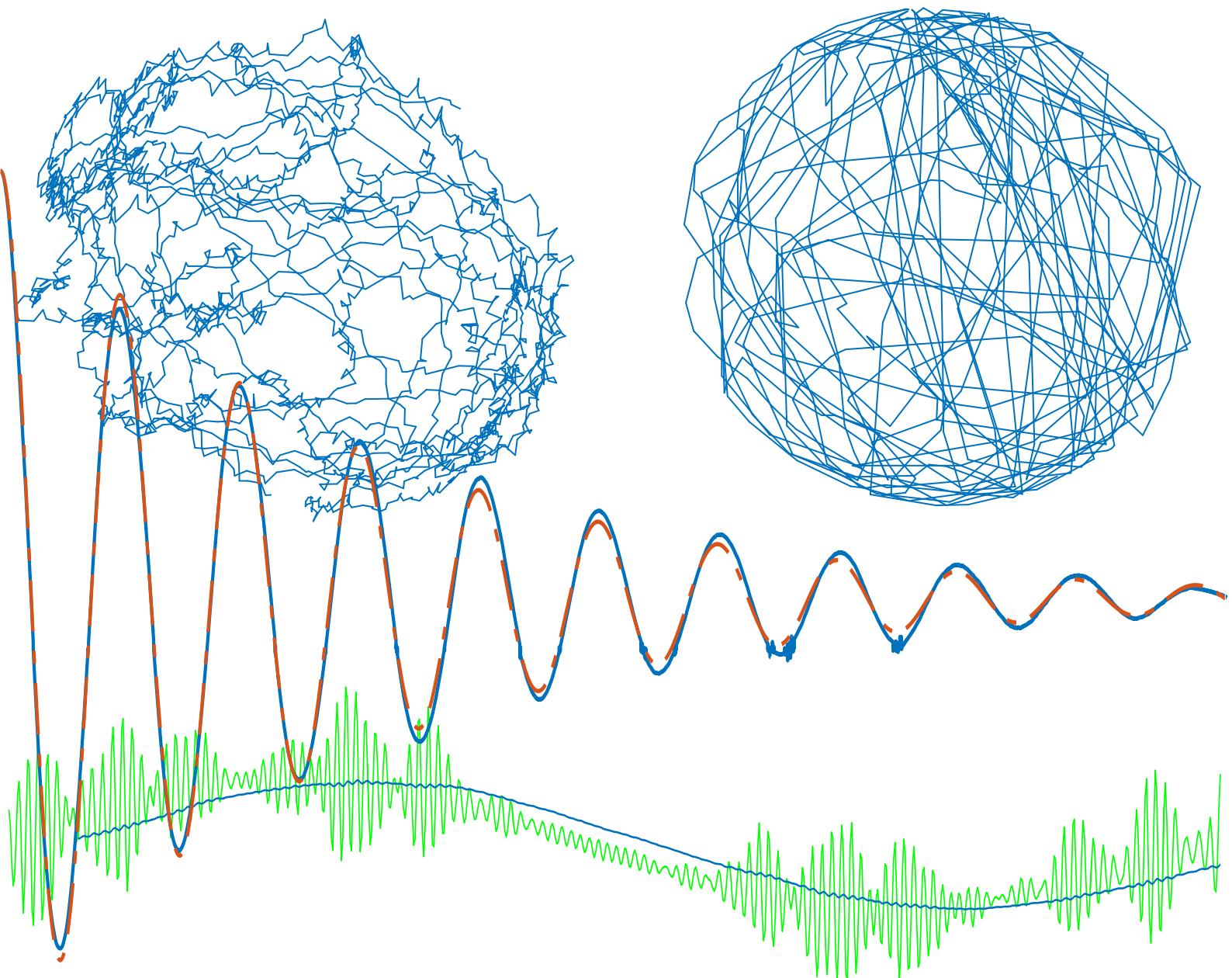


# Cost Effective Attitude Control Validation Test Methods for CubeSats Applied to PolarCube

Maxim Clarke





# Cost Effective Attitude Control Validation Test Methods for CubeSats Applied to PolarCube

by

## Maxim Clarke

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended on Thursday March 31, 2016 at 14:00.

Student number: 4047117  
Project duration: January 12, 2015 – March 31, 2016  
Thesis committee: Dr. J. Guo, TU Delft, supervisor  
                      Ir. B. T. C. Zandbergen, TU Delft  
                      Dr. P. Chu, TU Delft

An electronic version of this thesis, along with access to the source code and data  
are available upon personal request to: [maxim.clarke.thesis@gmail.com](mailto:maxim.clarke.thesis@gmail.com).





# Preface

This thesis presents a design (Chapter 3), implementation (Chapter 4) and evaluation (Chapter 5) of methods for validation tests of the attitude control system of the PolarCube satellite being developed by students at the Colorado Space Grant Consortium (COSGC). The result was a minimalist design that made efficient use of resources by focusing on meeting functional requirements with a minimum of effort and complexity. This document is intended to serve as reference to engineers who wish to perform validation tests of their own satellite's attitude control system. To this end, efforts have been made to explain all of the concepts involved to a fundamental level and make the reasoning behind every decision transparent (Chapter 2), as well as provide step-by-step instructions on how to implement the tests (Procedures at the end of the document).

I would like to thank the student engineers on the PolarCube mission who worked to help me accomplish this project: Adam St. Amand, Alex Bertman, Andrew Jones, Chris Rouw, Franklin Hinckley, Glenda Alvarenga, Jannine Vela, Kent Lee, Kristen Hanslik, Nathan Voth, Russell Gleason, Ryan Cutter and Umang Patel. Thanks as well to a number of individuals who gave valuable advice: Carlos Pulido, Gabrielle Massone, Jesse Ellison, Lee Jasper, Logan Smith and Patrick Kenneally. A debt is owed to Will Sear for a crucial favor at the end of my stay in Boulder. Next to my supervising professor: Dr. Jian Guo and Prof. Eberhart Gill for approving and supporting this project as well as Ir. Barry Zandbergen and Dr. Ping Chu for participating in the defense and evaluation of the thesis work. An most of all I would like to thank Brian Sanders and Dr. Al Gasiewski for inviting me to come work on their satellite.

*Maxim Clarke  
Delft, March 2016*



# Abstract

The problem of testing the performance of an attitude control system presents new challenges and opportunities when conducted on a CubeSat scale. This is the result of drastic reduction in physical properties (size, mass, torques) as well as project resources (funds, manpower, time) when compared to traditional satellite programs. This thesis presents an analysis of the problem of validating active attitude control of a CubeSat before launch and a proposed methodology that is demonstrated on the Colorado Space Grant Consortium's PolarCube satellite. In order for an attitude control system's performance to be measured, it must be provided with a physical environment that allows the system to act similarly to how it would in orbit and its behavior must be recorded in such a way that metrics of performance can be derived. To date, published tests of this nature on CubeSats have been limited in their precision due to uncertainty in external torques on the attitude control system and have mostly been conducted on commercially available attitude control system modules.

A string suspension testbed was chosen to provide a simulation of microgravity that allows the system to rotate free of friction. This thesis builds on the practices for string suspension testing developed for the MicroMAS CubeSat mission in which a "fit-predict-fit" method of producing metrics of attitude control system performance was first implemented for CubeSats. The project set out to identify and solve points of failure that were limiting measurement performance of the tests conducted on the MicroMAS system and ultimately produce more accurate measurements and predictions of testbed and attitude control system dynamic response.

An engineering model of the satellite bus was designed and built to provide independent power, wireless communication and data handling to the attitude determination and control subsystem. An attitude determination method was developed using MEMS magnetometers, accelerometers and rate gyroscopes to operate within a laboratory environment. A model of the dynamics of the test model's behavior in the testbed was created to generate predictions of the test model's response to test conditions, act as a platform to compare measured and expected test results, and verify the attitude determination method.

Attitude determination performance was determined through a combination of direct testing and dynamics modeling in software. The methods found a maximum (worst case scenario) heading determination error of  $4.6^\circ$  after feed-forward correction based on characterization tests. Oscillation tests were used to determine the external torque properties of the string suspension testbed to within two significant figures, a drastic improvement in performance compared to the MicroMAS test results. Less than \$300 were spent on hardware dedicated to testing.

The overall system is marked by its simplicity and cost-effectiveness. The results will render attitude control validation testing and consequently the use of active attitude control more accessible to future CubeSat missions. Improvements in performance when compared to the MicroMAS test results were identified as the result of more robust and flexible software modelling of string suspension testbed dynamics, improved methods of characterizing testbed external torque properties as well as improved attitude determination performance.



# Contents

|  |              |
|--|--------------|
| <b>List of Figures</b>   | <b>xii</b>   |
| <b>List of Tables</b>  | <b>xv</b>    |
| <b>List of Acronyms</b>  | <b>xvii</b>  |
| <b>List of mathematical symbols</b>  | <b>xviii</b> |
| <b>1 Introduction</b>  | <b>1</b>     |
| 1.1 Scope . . . . .  | 1            |
| 1.2 Research background . . . . .  | 1            |
| 1.3 Research objectives . . . . .  | 2            |
| 1.4 Tasks . . . . .  | 2            |
| 1.5 Document structure . . . . .   | 3            |
| <b>2 Project context and theory</b>  | <b>5</b>     |
| 2.1 Introduction to the PolarCube mission . . . . .                          | 5            |
| 2.2 Attitude determination theory . . . . .                                  | 6            |
| 2.2.1 Coordinate frames . . . . .  | 6            |
| 2.2.2 Attitude notation . . . . .  | 8            |
| 2.2.3 Determination . . . . .  | 11           |
| 2.3 Attitude control theory . . . . .  | 11           |
| 2.3.1 Control law basics . . . . .   | 11           |
| 2.3.2 Three dimensional rigid body dynamics . . . . .                        | 13           |
| 2.4 Polarcube's Attitude determination and Control Subsystem (ACS) . . . . . | 13           |
| 2.4.1 Hardware architecture . . . . .  | 14           |
| 2.4.2 Sensors . . . . .  | 14           |
| 2.4.3 Actuators . . . . .  | 15           |
| 2.4.4 PolarCube control law . . . . .  | 16           |
| 2.4.5 ADCS control modes . . . . .   | 17           |
| 2.4.6 Functional block diagram of ACS in flight . . . . .                    | 18           |
| 2.4.7 PolarCube mission requirements to verify . . . . .                     | 18           |
| 2.5 Chapter summary . . . . .  | 19           |
| <b>3 Design of the test system</b>   | <b>21</b>    |
| 3.1 Top level system engineering . . . . .                                   | 21           |
| 3.1.1 Stakeholder needs . . . . .  | 21           |
| 3.1.2 Research objectives . . . . .  | 22           |
| 3.1.3 Top level requirements . . . . .                                       | 22           |
| 3.2 Microgravity simulation . . . . .  | 23           |
| 3.2.1 String suspension . . . . .  | 23           |
| 3.2.2 Spherical air bearing . . . . .  | 27           |
| 3.2.3 Rotary air bearing . . . . .   | 30           |
| 3.2.4 Parabolic flight . . . . .   | 33           |
| 3.2.5 Computer simulation . . . . .  | 33           |
| 3.2.6 Trade study . . . . .  | 34           |
| 3.3 Testbed dynamics model . . . . .   | 36           |
| 3.4 Satellite engineering model for testing . . . . .                        | 36           |
| 3.4.1 Structure . . . . .  | 37           |
| 3.4.2 Power . . . . .  | 37           |
| 3.4.3 Data handling . . . . .  | 38           |
| 3.4.4 Attitude determination . . . . .                                       | 38           |

|          |  |           |
|----------|--|-----------|
| 3.4.5    | Test execution . . . . .   | 44        |
| 3.4.6    | Actuator verification . . . . .  | 45        |
| 3.5      | Test system design overview . . . . .                                  | 46        |
| 3.5.1    | Requirements breakdown . . . . .                                       | 46        |
| 3.5.2    | System architecture diagram. . . . .                                   | 50        |
| 3.5.3    | Functional block diagram of ACS on the testbed . . . . .               | 51        |
| 3.5.4    | Software architecture diagram . . . . .                                | 52        |
| 3.6      | Chapter summary. . . . .   | 52        |
| <b>4</b> | <b>Materials and methods of test system implementation</b>             | <b>55</b> |
| 4.1      | String testbed dynamics model . . . . .                                | 55        |
| 4.1.1    | Rigid body dynamics. . . . .   | 55        |
| 4.1.2    | Simulating attitude determination. . . . .                             | 56        |
| 4.2      | Attitude determination . . . . .                                       | 57        |
| 4.2.1    | Accelerometer . . . . .  | 57        |
| 4.2.2    | Arduino test model. . . . .  | 58        |
| 4.2.3    | Magnetic bias determination . . . . .                                  | 58        |
| 4.2.4    | Algorithm . . . . .  | 59        |
| 4.2.5    | Filtering . . . . .  | 60        |
| 4.2.6    | Attitude determination test . . . . .                                  | 61        |
| 4.3      | Considerations for implementation of attitude control . . . . .        | 63        |
| 4.3.1    | Attitude error . . . . .   | 64        |
| 4.3.2    | Maneuver selection . . . . .   | 64        |
| 4.3.3    | New control law . . . . .  | 65        |
| 4.3.4    | Reaction wheel speed control law . . . . .                             | 65        |
| 4.4      | Communication and data logging. . . . .                                | 65        |
| 4.4.1    | Wireless communication. . . . .  | 65        |
| 4.4.2    | User interface . . . . .   | 66        |
| 4.4.3    | Commanding the test model. . . . .                                     | 66        |
| 4.4.4    | Data stream down . . . . .   | 67        |
| 4.5      | Testbed . . . . .  | 68        |
| 4.5.1    | Test location . . . . .  | 68        |
| 4.5.2    | Material selection . . . . .   | 68        |
| 4.5.3    | Suspending the test model. . . . .                                     | 69        |
| 4.6      | String torque characterization . . . . .                               | 70        |
| 4.6.1    | Attempts to find better fits to the MicroMAS test results . . . . .    | 70        |
| 4.6.2    | Oscillation tests with the Arduino test model . . . . .                | 71        |
| 4.7      | Test procedures . . . . .  | 72        |
| 4.7.1    | Integration. . . . .   | 73        |
| 4.7.2    | Software tools . . . . .   | 73        |
| 4.7.3    | Reporting results. . . . .   | 73        |
| 4.8      | Chapter summary. . . . .   | 74        |
| <b>5</b> | <b>Verification of test system performance</b>                         | <b>77</b> |
| 5.1      | Gyro biasing correction . . . . .                                      | 77        |
| 5.2      | Influence of the reaction wheels on the magnetometer . . . . .         | 78        |
| 5.3      | Sphere center calculation . . . . .                                    | 79        |
| 5.3.1    | Filtering results . . . . .  | 79        |
| 5.3.2    | Vector magnitude plot . . . . .  | 80        |
| 5.3.3    | Sphere plots . . . . .   | 81        |
| 5.4      | Attitude determination test . . . . .                                  | 81        |
| 5.5      | String torque characterization . . . . .                               | 83        |
| 5.5.1    | Discussion of oscillation test results . . . . .                       | 87        |
| 5.5.2    | Testbed performance . . . . .  | 87        |
| 5.6      | Verification of attitude determination with a software model . . . . . | 89        |
| 5.7      | Chapter summary. . . . .   | 91        |

|   |            |
|---|------------|
| <b>6 Conclusions and future recommendations</b> | <b>93</b>  |
| 6.1 Conclusion . . . . .                        | 93         |
| 6.2 Requirements verification. . . . .          | 94         |
| 6.3 Recommendations for future work . . . . .   | 96         |
| <b>Bibliography</b>                             | <b>99</b>  |
| <b>A Reaction Wheel performance analysis</b>    | <b>101</b> |
| A.1 Output torque. . . . .                      | 101        |
| A.2 Friction torque . . . . .                   | 102        |
| A.3 Net torque . . . . .                        | 105        |
| A.4 Comparison to motor data sheet . . . . .    | 105        |
| A.4.1 Static torque . . . . .                   | 105        |
| A.4.2 Torque slope . . . . .                    | 105        |
| A.5 Discussion . . . . .                        | 106        |
| <b>B Source Code</b>                            | <b>107</b> |
| B.1 Testbed Software Model. . . . .             | 107        |
| B.2 Sphere center calculation . . . . .         | 117        |
| B.3 Attitude determination test . . . . .       | 121        |



# List of Figures

|  |    |
|--|----|
| 1.1 Diagram showing how information established in each chapter is used later on in the document.  | 3  |
| 2.1 Renderings of PolarCube in stowed (left) and deployed (right) configurations.  | 6  |
| 2.2 CAD rendering of PolarCube showing the alignment $\beta$ frame axes. The origin in this image is displaced to make it easier to display the axes. The actual $\beta$ frame is centered at the CoG of the satellite | 7  |
| 2.3 Diagram describing how the E frame is defined.   | 7  |
| 2.4 Visualization of the frame transformation described in Equation 2.13.  | 9  |
| 2.5 Diagram showing how $\beta$ , $\eta$ and $\theta$ are related in this example.   | 12 |
| 2.6 Responses of P and PD control laws to an attitude error of 1 rad.  | 12 |
| 2.7 Responses of P and PD control laws to an attitude error of 1 rad and an external torque of 0.1 Nm.   | 13 |
| 2.8 Image of a later revision of the ACS main board on its ground support equipment. The attitude determination sensors are indicated in orange. Photo credit to Adam St. Amand.                                       | 14 |
| 2.9 CAD renderings of the ACS actuators.   | 14 |
| 2.10 Image of the A3G4250D MEMS gyro.  | 15 |
| 2.11 Image of the HMC5883L magnetometer.   | 15 |
| 2.12 Concept drawing of MiniRad showing the mirror's scanning pattern [4].   | 18 |
| 2.13 Concept drawing showing PolarCube's alignment with respect to earth when performing science operations.   | 18 |
| 2.14 A functional block diagram showing the interplay of elements of ACS when performing attitude control.   | 18 |
| 3.1 A measured oscillation test deflection angle response plotted against output from damped spring models fitted to match different portions of the response. [22]  | 25 |
| 3.2 Measured response of oscillation of an attitude control system under de-tumble control compared to a prediction from a software model of testbed dynamics.   | 26 |
| 3.3 Measured and predicted results of a slew test published in [22].   | 27 |
| 3.4 A magnification of the data in Figure 3.3 showing how the strings restoring torques are compromising stability at the target attitude.   | 27 |
| 3.5 Image of a spherical air bearing [27].   | 28 |
| 3.6 An example of an "umbrella" architecture implemented on CubeSat hardware [22].   | 29 |
| 3.7 An example of using counterweights to lower the center of gravity of the test subject [14].  | 29 |
| 3.8 An example of a rotary air bearing [1].  | 30 |
| 3.9 Gravitational torques about the rotary bearing's rotation axis as the test subject rotates.  | 31 |
| 3.10 Free body diagram depicting slew motion about +x on the suspended rotary bearing testbed with reaction wheels rotating about +y and +z.   | 32 |
| 3.11 Torques to the supporting load cells about the rotary air bearing X and Y axes defined in Figure 3.10 as the test subject rotates.  | 33 |
| 3.12 Magnitude of gyroscopic torque produced by a reaction wheel as a function of precession rate.   | 33 |
| 3.13 CAD rendering of ACS mounted within the PEZ bus structure.  | 37 |
| 3.14 Diagram explaining the relativity of attitude inaccuracy when testing attitude control performance in a single dimension.   | 39 |
| 3.15 True response of the one dimensional system described in Section 2.3.1 (Page 11) performing PID control.  | 40 |
| 3.16 The same responses as in Figure 3.15 as they were perceived by the test model.  | 40 |
| 3.17 A diagram describing the interfaces between the electronic hardware elements in the testbed design.   | 50 |
| 3.18 A diagram describing the interactions between functional elements of the test subject and testbed.  | 51 |
| 3.19 Functional block diagram of the ACS flight software written for validation testing.   | 52 |

|      |  |    |
|------|--|----|
| 4.1  | Image of the ADXL345 MEMG accelerometer.   | 57 |
| 4.2  | Image of the Arduino test model. The accelerometer is not mounted. Photo credit to Adam St. Amand.   | 58 |
| 4.3  | Image of the XBee explorer dongle.   | 58 |
| 4.4  | Magnetometer measurements taken while rotating the Arduino test model. Units are presented in counts.  | 59 |
| 4.5  | Vector magnitude of the values plotted in Figure 4.4. The signal has a standard deviation of 115.6 counts.   | 59 |
| 4.6  | Data from Figure 4.5 after subtracting the determined center value from every data point. Units are presented in counts.   | 59 |
| 4.7  | Vector magnitude of the values plotted in Figure 4.4. The signal has a standard deviation of 12 counts.  | 59 |
| 4.8  | Image of the test model on the attitude determination test rig.  | 61 |
| 4.9  | Measured heading values from repeated attitude determination tests plotted with the true heading values from the first test.   | 62 |
| 4.10 | Heading determination error from the two attitude determination tests presented in Figure 4.9.   | 62 |
| 4.11 | Diagram depicting how biasing error distorts the mapping of true heading to heading determined through magnetometer measurements.  | 62 |
| 4.12 | Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.   | 63 |
| 4.13 | Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.   | 63 |
| 4.14 | Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.   | 63 |
| 4.15 | All of the data from Figures 4.12 through 4.14 plotted together. Both tests from the same figure share the same icon.  | 63 |
| 4.16 | Image of the XBee on its breakout board connected to CDH.  | 66 |
| 4.17 | Uplink GUI window for sending commands to the test model.  | 66 |
| 4.18 | Downlink GUI window for displaying the test model's state vector.  | 67 |
| 4.19 | Location in the mission operations command chosen to conduct string suspension tests due to the size of open space.  | 68 |
| 4.20 | Fire extinguisher feed pipe in the ceiling used for attachment of the suspension lines of the test model and attitude determination test rig.  | 69 |
| 4.21 | Demonstration of how the gap in the ceiling was filled to stop airflow.  | 69 |
| 4.22 | The attitude determination test rig suspended by two lines to constrain rotation.  | 69 |
| 4.23 | String that was used to suspend the test models: 60 lb test South Bend monofilament fishing line purchased from Amazon.com.  | 70 |
| 4.24 | Suspension system using fishing line that allows the alignment of the test model to be adjusted.   | 70 |
| 4.25 | Test model suspended about the -X axis.  | 70 |
| 4.26 | Comparison of the data in Figure 3.1a to the output of a simulation with external torques described by Equation 4.29. [22].  | 71 |
| 4.27 | Comparison of the data in Figure 3.1b to the output of a simulation with external torques described by Equation 3.1 that has a closer fit to initial conditions of the measured data. [22]                       | 71 |
| 4.28 | Raw heading from oscillation tests conducted with the Arduino test model plotted against responses from a one dimensional model of damped rotational spring motion.  | 72 |
| 4.29 | Image of the MicroMAS attitude control test model suspended using an S-hook and eyelet connection. [22].   | 73 |
| 5.1  | A set of raw gyro data collected while the test model was stationary.  | 78 |
| 5.2  | Raw magnetometer data that was collected on the integrated test model while two wheels were constrained and a single wheel was spinning. Each column of plots represents data that was collected simultaneously. | 79 |
| 5.3  | Raw and filtered magnetometer data used in the sphere center calculation that was implemented in the final tests.  | 80 |

|      |   |     |
|------|---|-----|
| 5.4  | Vector magnitude of the filtered and bias-corrected magnetometer data from Figure 5.3. The data has a standard deviation of 74 counts. . . . .  | 80  |
| 5.5  | Filtered magnetometer data from Figure 5.3 that has been corrected for bias and plotted in 3D. . . . .  | 81  |
| 5.6  | Heading error from the two attitude determination tests performed about the +X axis as a function of true heading. The total range of heading determination error is 0.33 rad. . . . .  | 82  |
| 5.7  | Heading error from the two attitude determination tests about the +Y axis as a function of true heading. The total range of heading determination error is 0.59 rad. . . . .  | 82  |
| 5.8  | Heading error from the two attitude determination tests about the -Z axis as a function of true heading. The total range of heading determination error is 0.2 rad. . . . .   | 82  |
| 5.9  | Raw measured heading from the oscillation tests about the +X axis plotted against the response from the best fitting damped spring model. . . . .   | 84  |
| 5.10 | Heading from the oscillation tests about the +X axis that has been retroactively corrected for attitude error using data from Figure 5.6 plotted against the response from the best fitting damped spring model. . . . .              | 84  |
| 5.11 | Raw measured heading from the oscillation tests tests about the +Y axis plotted against the response from the best fitting damped spring model. . . . .   | 85  |
| 5.12 | Heading from the oscillation tests about the +Y axis that has been retroactively corrected for attitude error using data from Figure 5.7 plotted against the response from the best fitting damped spring model. . . . .              | 85  |
| 5.13 | Raw measured heading from the oscillation tests about the -Z axis plotted against the response from the best fitting damped spring model. . . . .   | 86  |
| 5.14 | Heading from the oscillation tests about the -Z axis that has been retroactively corrected for attitude error using data from Figure 5.8 plotted against the response from the best fitting damped spring model. . . . .              | 86  |
| 5.15 | Output of the string testbed software model under the conditions specified in Table 5.3. . . . .  | 90  |
| 5.16 | Output of the string testbed software model under the conditions specified in Table 5.3 with correction for the delay inherent in magnetometer filtering. . . . .   | 90  |
| A.1  | Measured speed of the reaction wheels in response to a variety of torque commands from rest. The reference voltages received by the driver chip were equal to $3V_3 \cdot n/8$ where $n = 8, 7, \dots, 1$ . . . . .                   | 103 |
| A.2  | Plot of motor output torque experienced by the reaction wheels as a function of reaction wheel speed. . . . .   | 103 |
| A.3  | Measured speed of the reaction wheels in response to the introduction of a zero reference voltage at their maximum speed. This data was collected in between the measurements taken for Figure A.1. . . . .                           | 104 |
| A.4  | Plot of friction torque experienced by the reaction wheels as a function of reaction wheel speed. . . . .   | 104 |
| A.5  | Linear fit of the motor output torque experienced by the reaction wheels (Figure A.2) added to a linear fit of the of the friction torque experienced while motor is turned off and the wheels are decelerating (Figure A.4). . . . . | 105 |



# List of Tables

|     |  |     |
|-----|--|-----|
| 2.1 | PolarCube mission requirements that could be verified with validation testing. . . . .   | 19  |
| 3.1 | Grade criterion for the microgravity simulation testbed trade study. . . . .   | 34  |
| 3.2 | Rough evaluation of the proposed Microgravity simulation testbed concepts (rows) for a selection of performance metrics (columns). . . . .   | 35  |
| 3.3 | Indication of whether a given performance metric (columns) can be produced using a given microgravity simulation testbed (rows). An O indicates YES and an X indicates NO. . . . . | 35  |
| 3.4 | Requirements verification matrix for the attitude control validation testing thesis project. . . . .   | 46  |
| 4.1 | Op-code definition for the validation testing interface. . . . .   | 67  |
| 5.1 | Amplitude of respective reaction wheel influence on each magnetometer axis. . . . .  | 78  |
| 5.2 | Torque performance of the string suspension testbed derived from string characterization test results. . . . .   | 88  |
| 5.3 | Paramaters defining the simulation that produced the outputs displayed in Figures 5.15 and 5.16. . . . .   | 89  |
| 6.1 | Requirements verification table for the test system design. . . . .  | 94  |
| A.1 | Parameters defining the best linear fit of reaction wheel output torque as a function of reaction wheel spin rate from the data in Figure A.2. . . . .                             | 102 |
| A.2 | Parameters defining the best linear fit of reaction wheel friction torque as a function of reaction wheel spin rate from the data in Figure A.4. . . . .                           | 102 |



# Lists of acronyms and symbols

## List of Acronyms

| Acronym          | Definition   |
|------------------|--|
| ACS              | PolarCube's attitude determination and control subsystem             |
| ADCS             | Attitude Determination and Control Subsystem                         |
| ALL-STAR         | Agile Low-cost Laboratory for Space Technology Acceleration Research |
| CAD              | Computer Aided Design  |
| CDH              | PolarCube's Command and Data Handling subsystem                      |
| CMOS             | Complementary Metal-Oxide Semiconductor                              |
| COM              | PolarCube's COMmunications subsystem                                 |
| CONOPS           | CONcept of OPerations  |
| COSGC            | COLORado Space Grant Consortium                                      |
| COTS             | Commercial Off-The-Shelf   |
| CU Boulder       | University of Colorado at Boulder                                    |
| CoG              | Center of Gravity  |
| CoR              | Center of Rotation   |
| DC               | Direct Current   |
| DCM              | Direction Cosine Matrix  |
| EPS              | PolarCube's electrical power subsystem                               |
| GPS              | Global Positioning System  |
| GUI              | Graphical User Interface   |
| HIL              | Hardware-In-the-Loop   |
| I <sup>2</sup> C | Inter-Integrated Circuit   |
| MEMS             | Microelectromechanical System  |
| MMOI             | Mass Moment Of Inertia   |
| MRP              | Modified Rodriguez Parameter   |
| MicroMAS         | Micro-sized Microwave Atmospheric Satellite                          |
| MinXSS           | Miniature X-ray Solar Spectrometer                                   |
| MiniRad          | Miniature Microwave Radiometer                                       |
| PCB              | Printed Circuit board  |
| PEZ              | Payload Extension Zone   |

| Acronym | Definition   |
|---------|--|
| PID     | Potential-Integral-Derivative                            |
| P-Pod   | Poly-Picosat orbital deployer                            |
| RMS     | Root Mean Square   |
| SPI     | Serial Peripheral Interface                              |
| THEIA   | Telescopic High-definition Earth Imaging Apparatus       |
| UART    | Universal Asynchronous Receiver/Transmitter              |
| UNP-8   | The eighth cycle of the University Nanosatellite Program |

## List of mathematical symbols

### Roman

| Symbol              | Meaning  |
|---------------------|--|
| $A$                 | Amplitude  |
| $B$ or $\mathbf{B}$ | Magnetic field strength  |
| $C$                 | Celestial reference frame  |
| $C_{ij}$            | Component of a direction cosine matrix                               |
| $E$                 | Body-centered Earth-reference frame                                  |
| $H$                 | Angular momentum   |
| $I$ or $[I]$        | Mass moment of inertia   |
| $I_{Ws}$            | MMOI of a reaction wheel about its spin axis                         |
| $I_{Wt}$            | MMOI of a reaction wheel about a transverse axis                     |
| $K$                 | Potential gain   |
| $K_I$ or $[K_I]$    | Integral gain  |
| $K_{PC}$            | Predictor-corrector gain   |
| $L$                 | Known external torque  |
| $P$ or $[P]$        | Derivative gain  |
| $[R]$               | Frame rotation expressed in DCM                                      |
| $S_e$               | Normalized gyro sensitivity error                                    |
| $\hat{\mathbf{a}}$  | Vector defining an axis of the body-centered earth-reference frame   |
| $\hat{\mathbf{b}}$  | Vector defining an axis of the body frame                            |
| $c$                 | Damping constant   |
| $\hat{\mathbf{e}}$  | Principle rotation axis  |
| $\mathbf{g}$        | Gravitational field strength   |
| $\mathbf{h}_s$      | Angular momentum of the reaction wheels expressed in the body frame. |
| $k$                 | Spring constant  |

| Symbol    | Meaning                                      |
|-----------|--|
| $\hat{n}$ | Vector defining an axis of an inertial frame |
| $\hat{q}$ | Rotation axis expressed in the body frame    |
| $t$       | Time   |
| $u_s$     | Control law output torques                   |
| $z$       | Integrated attitude error                    |

## Greek

| Symbol               | Meaning  |
|----------------------|--|
| $\Theta$             | Angular position of a reaction wheel                         |
| $\Phi$               | Principle rotation angle                                     |
| $\Omega$             | Reaction wheel rotation rate                                 |
| $\Omega$             | Reaction wheel rotation rates expressed in the $\beta$ frame |
| $\beta$              | Body reference frame   |
| $\eta$               | Inertial reference frame                                     |
| $\theta$             | One-dimensional attitude or pitch                            |
| $\sigma$             | Frame rotation expressed as a modified Rodriguez parameter   |
| $\tau$ or $\tau$     | Torque   |
| $\phi$               | Roll   |
| $\psi$               | Heading  |
| $\omega$ or $\omega$ | Body rotation rate expressed in the body frame               |

## Indeces

| Index | Meaning           |
|-------|-------------------|
| 0     | Initial condition |
| RW    | Reaction wheel    |
| amb   | Ambient           |
| bs    | Bias              |
| d     | Deflection        |
| e     | Error             |
| eq    | Equilibrium       |
| est   | Estimated         |
| ext   | External          |
| fr    | Friction          |
| gyr   | Gyroscopic        |

| <b>Index</b> | <b>Meaning</b>                        |
|--------------|---------------------------------------|
| int          | Interval                              |
| mag          | Magnetorquer                          |
| mes          | Implies the measured/determined value |
| net          | Net                                   |
| no           | Noise                                 |
| out          | Output                                |
| p            | Precession                            |
| prd          | Predicted                             |
| q            | Quadratic                             |
| t            | Target                                |
| tru          | True                                  |

# Introduction

## 1.1. Scope

This report documents the thesis project on test methods for pre-launch validation of CubeSat attitude control performance. The study was performed from 12/1/2015 to 31/03/2016 at the Colorado Space Grant Consortium (COSGC) at The University of Colorado Boulder (CU Boulder) by an MSc student of aerospace engineering at the Delft University of Technology (TU Delft).

## 1.2. Research background

Advances in technology are allowing spacecraft to be built on smaller scales at a lower cost. In the past few decades, this has lead to the development of small satellites. Small satellites are defined as satellites weighing less than 500 kg and distinguish themselves from traditional satellites by making use of Commercial Off-The-Shelf (COTS) technologies as opposed to developing satellite components on a one-off basis for each mission, resulting in less expensive missions with shorter development schedules. This effectively compounds to accelerate research in space technology by providing more frequent opportunities to validate systems in space. When satellites get small and cheap enough, several can be sent into space, providing wider coverage and mitigating the risk of satellite system failure. Using this approach allows for lower standards for testing and qualification, further reducing cost and development time.

CubeSats are currently the most popular embodiment of this approach. In 1999, California Polytechnic State University, in collaboration with the Space Systems Development Laboratory at Stanford University began developing the CubeSat standard for Picosatellites and Nanosatellites [18]. A CubeSat is composed of one or more units of dimensions 10 x 10 x 10 cm and weighing a maximum of 1.33 kg. This standard allows standardized deployment systems to be built for CubeSats. The CubeSat community has created a centralized flight integration and launch service based around the Poly Picosat Orbital Deployer (P-Pod) [29]. The P-Pod is a CubeSat Launch adapter and deployment system that was developed at Cal Poly to provide this service. It can deploy any combination of 3 CubeSat units [21]. The same principle of "one size fits all" can be applied to CubeSat components and systems. Any system designed to be optimal for the CubeSat form factor can be replicated for use in another CubeSat, giving birth to a market of "plug and play" COTS systems for CubeSats. The simplicity, flexibility and low cost of CubeSat missions permits universities to use them as educational projects, providing students the opportunity to gain experience conducting a real space mission as part of their education.

TU Delft has already sent two CubeSats into space, Delfi C<sup>3</sup> and Delfi n3Xt. Both missions were developed by MSc students at TU Delft. TU Delft's next CubeSat mission, DelFFi, will be part of the QB50 mission to perform atmospheric measurements in the lower thermosphere with a constellation of 50 CubeSats built by universities across the world. DelFFi will launch two 3U CubeSats and attempt formation flying between them [5]. All of the Delfi satellites are part of TU Delft's roadmap to realize the Orbiting Low Frequency Antennas for Radio Astronomy (OLFAR) project. OLFAR will consist of a swarm of at least 50 CubeSats that act as a telescope measuring in the 0.1 to 30 MHz spectrum. [7]

The COSGC is developing the Agile Low-cost Laboratory for Space Technology Acceleration Research (ALL-STAR) bus in collaboration with CU Boulder and Lockheed Martin. ALL-STAR is envisioned to be a reproducible 3 unit (3U) CubeSat bus that could serve a variety of payloads, providing high performance at a low

cost and short development schedule. Lockheed Martin expects to eventually use this platform to conduct or facilitate large scale CubeSat missions. ALL-STAR's first mission flew in April 2014 With the Telescopic High-definition Earth Imaging Apparatus (THEIA) payload. I had been invited to participate in the testing regime of ALL-STAR's second mission, PolarCube. PolarCube's payload, the Miniature Microwave Radiometer (Mini-Rad), is a passive microwave radiometer used to determine a three dimensional temperature profile of the thermosphere to be used in climate and weather models and monitoring of sea ice retreat [4].

Prof. Al Gasiewski of CU Boulder and Brian Sanders of the COSGC had proposed that testing equipment and methods be developed to perform on-ground validation of the performance of PolarCube's 3-axis attitude control system. The testing envisioned for this thesis involved exposing a model of the satellite attitude control system to a simulated space environment to see if its control modes behave as expected.

This thesis is part of ongoing efforts at TU Delft and the COSGC to develop high performance Attitude determination and Control Subsystems (ADCS) built by students on the CubeSat platform. To date, very few student CubeSat missions have relied on pointing systems built in house to operate. Most designs either do not rely on pointing to perform, use a form of passive attitude control, or rely on COTS ADCS subsystems from CubeSat manufacturing companies. The COTS systems available are very effective but expensive and the potential for research to be conducted on them is limited. Having attitude control systems like those on PolarCube and the Delfi satellites that are open to study and publication will be invaluable in enabling the advancement of CubeSat programs across the globe.

### **1.3. Research objectives**

Analysis of project stakeholder needs revealed the following research objectives for the thesis project:

- Develop a test system (testbed, support systems, and methodology) that allows the performance of PolarCube's attitude control system to be measured in terms of its ability to meet ADCS functional requirements.
- The methods developed should be accessible to future test engineers at the COSGC and other educational CubeSat programs.

### **1.4. Tasks**

The thesis project entailed:

- Conducting a requirements investigation to determine the best means of meeting the objectives with the available resources.
- Providing a physical environment that allows actuator torques to dominate motion of the attitude control system.
- Providing independent power, data handling and wireless communication to the attitude control system.
- Developing an attitude determination method that functions within a laboratory environment.
- Producing a software model of the dynamics of the system and its interaction with the environment that can be used to verify the attitude determination method and predict the response of the system under test conditions.
- Creating a computer interface to allow users to send commands to the attitude control system and see a live stream of system data and configuration through wireless communication.
- Conducting tests that verify the test system by measuring attitude determination performance and characterizing the dynamic effects that the testbed has on the attitude control system's motion.

## 1.5. Document structure

The order of topics in this document was chosen such that sufficient context is provided when discussing each topic. After the Introduction, the second chapter provides a detailed explanation of the context of the thesis in terms of the goals of PolarCube mission, its status, and background theory on attitude determination and control subsystems that will be used as reference in further argumentation. Chapter 3 describes the system design process for the thesis project, revealing the requirements of the project as they flow from stakeholder needs and explaining the system level design choices for the test system. Chapter 4 details the systems built and techniques used to realize the test system that is defined in the previous chapter. Chapter 5 reports on tests that were conducted on the test system, presenting results, interpretation and analysis of performance. Chapter 6 provides a conclusion of the thesis, declaring what the thesis achieved and provides avenues that the author has identified as potential improvements to the methods and systems developed.

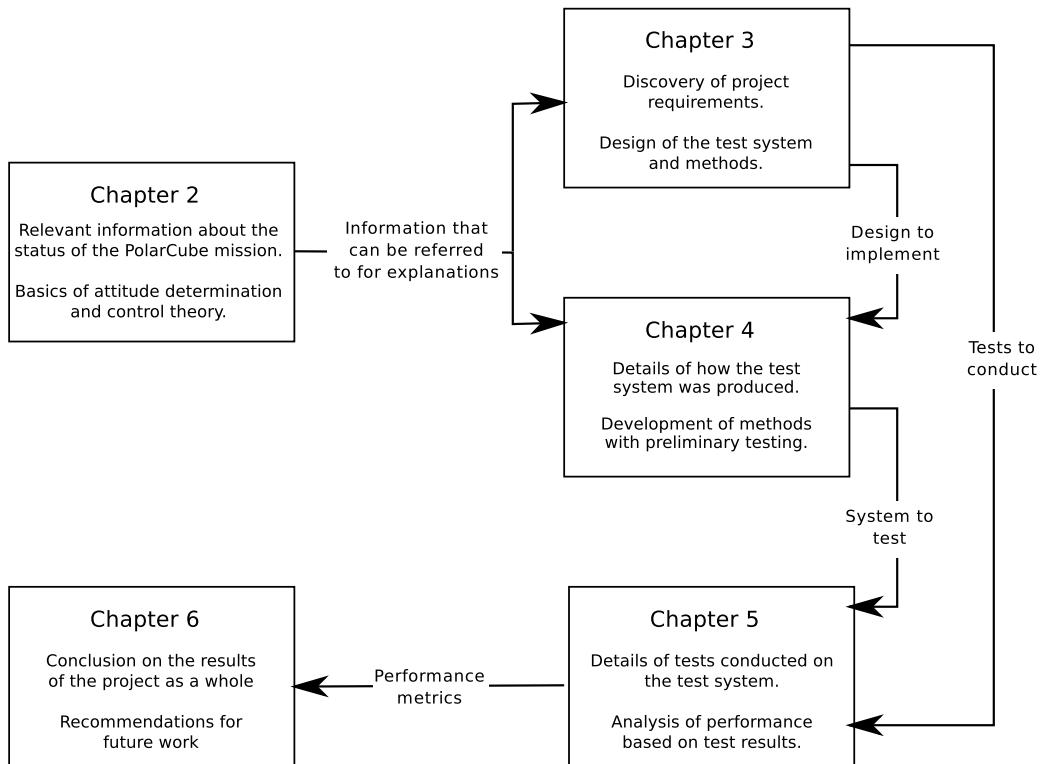


Figure 1.1: Diagram showing how information established in each chapter is used later on in the document.



# 2

## Project context and theory

This chapter serves as a reference of contextual information about the project. The chapter starts with an introduction of the PolarCube mission and how it fits into the goals of the COSGC. This is followed by an explanation of the basics of ADCS mathematical theory. The chapter ends with details about PolarCube's ADCS, including its design, how it functions and its development status.

### 2.1. Introduction to the PolarCube mission

PolarCube is the latest CubeSat mission being developed at the COSGC. The PolarCube satellite consists of the MiniRad passive microwave sensing instrument mounted on the ALL-STAR 3U CubeSat bus. MiniRad was designed in collaboration with the Center for Environmental Technology at CU Boulder to sense in the 118 GHz range in order to produce a 3D temperature profile of Earth's troposphere. Science data from PolarCube will be valuable in monitoring sea ice retreat and global climate change as well as improving weather prediction models, particularly concerning large storm formation. MiniRad images with a deployed spinning parabolic mirror. The mirror is focused at the radiometer's feedhorn antenna and rotates about the feedhorn's longitudinal axis, performing cross-track scanning and providing calibration reference points when the instrument is pointed at cold space. (see Figure 2.1)

The ALL-STAR bus is an ongoing program at the COSGC to develop a modular 3U CubeSat bus providing high power, attitude control performance and downlink capability within a 1.5U form factor, leaving 1.5U for the payload. ALL-STAR started as a joint project between the COSGC and Lockheed Martin Corporation: providing funding and mentoring to students in exchange for potential use of the bus in future missions as well as early recruitment opportunities. ALL-STAR's first flight was with the THEIA visible spectrum Earth imaging payload in the ALL-STAR/THEIA mission. ALL-STAR/THEIA flew on the SpaceX CRS-3 launch in April 2014. Communication was never established with All-STAR, but a carrier signal was detected on one of the passes, validating ALL-STAR's PEZ (Payload Extension Zone) deployment mechanism. ALL-STAR has undergone re-design of all of the subsystems for the PolarCube mission. At the time of the thesis, PolarCube was going through the process of bringing up and verifying the "silver" (full flight functionality) revisions of the subsystems in preparation for a set of subsystem verification tests required to pass "integration readiness". After which, the integration of the flight model will begin.

PolarCube is part of the eighth cycle of the University Nanosat Program (UNP-8) run by the Air Force Research Laboratory (AFRL), Air Force Office of Scientific Research and the American Institute of Aeronautics and Astronautics . The submission of their engineering design unit to the "flight competition review" won them second place among ten competing missions by a very close margin, securing significant funding and a launch opportunity.

PolarCube has received letters of support from the Air Force Weather Agency , AFRL Kirkland Air Force Base and the National Ice Center.



Figure 2.1: Renderings of PolarCube in stowed (left) and deployed (right) configurations.

## 2.2. Attitude determination theory

The purpose of an ADCS is to control the rotational orientation of the satellite with respect to other objects in the vicinity that it will interact with. Rotational orientation in this case is defined as the coordinate reference frame of the body of the satellite expressed in terms of a known external reference frame. This definition of rotational orientation is generally referred to as "attitude" in the field of aerospace engineering. Attitude determination is a process of detecting phenomena caused by other objects in space and using them to determine vectors that relate the body reference frame to the known external object. For example, a Sun sensor will detect radiation from the Sun and determine the direction that it is coming from with respect to the satellite. The satellite could then use this information to keep the solar panels perpendicular to the incident sunlight vector. This section explains the mathematics behind the process of calculating attitude to a basic level.

### 2.2.1. Coordinate frames

When relating the position, rotation and motion of two objects in space, it is convenient to think in multiple coordinate reference frames. Specific phenomena can be expressed very simply in specific reference frames (eg. gravity). So mathematics that can relate reference frames to each other and determine how phenomena occurring in one frame appear in a different frame are very effective. This section explains the reference frames that are relevant to discussion in this thesis.

This thesis work uses three dimensional Cartesian frames. These frames are defined by an origin and three orthogonal unit vectors that define the coordinate axes. Let the expressions in Equation 2.1 be true for a reference frame  $F$  defined by unit vectors  $\hat{\mathbf{u}}_1$ ,  $\hat{\mathbf{u}}_2$  and  $\hat{\mathbf{u}}_3$ :

$$\hat{\mathbf{u}}_1 := \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}^F, \hat{\mathbf{u}}_2 := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}^F, \hat{\mathbf{u}}_3 := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^F \quad (2.1)$$

The  $F$  superscript denotes that the vector is expressed in the  $F$  frame coordinate system and the  $:=$  symbol denotes a definition. Equation 2.2 describes how an arbitrary vector  $r$  is expressed in the  $F$  frame:

$$\mathbf{r}^F := \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}^F = r_x \hat{\mathbf{u}}_1 + r_y \hat{\mathbf{u}}_2 + r_z \hat{\mathbf{u}}_3 \quad (2.2)$$

### Body frame ( $\beta$ )

The body co-ordinate reference frame has its origin at the center of mass of the body in question and its axes are fixed with respect to the physical body. Let the unit vectors defining the body frame axes be  $\hat{b}_1$ ,  $\hat{b}_2$  and  $\hat{b}_3$ . In the case of PolarCube,  $\hat{b}_1$  is aligned with the S-band antenna,  $\hat{b}_2$  is aligned with the star tracking camera and  $\hat{b}_3$  is parallel to the payload mirror boom (see Figure 2.2). Attitude of a satellite is defined as an expression of the satellite's body reference frame with respect to another frame with the same origin or vice-versa.



Figure 2.2: CAD rendering of PolarCube showing the alignment  $\beta$  frame axes. The origin in this image is displaced to make it easier to display the axes. The actual  $\beta$  frame is centered at the CoG of the satellite

### Body centered Earth reference ( $E$ )

The Earth reference body frame has its origin at the center of the body in question and it's axes are defined by the body's relative position to the Earth. Let the unit vectors defining the body centered Earth reference frame axes be  $\hat{a}_1$ ,  $\hat{a}_2$  and  $\hat{a}_3$ . In this thesis project,  $\hat{a}_3$  is defined as away from the center of the Earth (up), although it is typically defined as down in aerospace engineering.  $\hat{a}_1$  points North and, in this case,  $\hat{a}_2$  points West (see Figure 2.3). The  $E$  frame is useful because idealized models of Earth's gravity and magnetic field can be expressed simply in it, as shown in Equations 2.3 and 2.4.

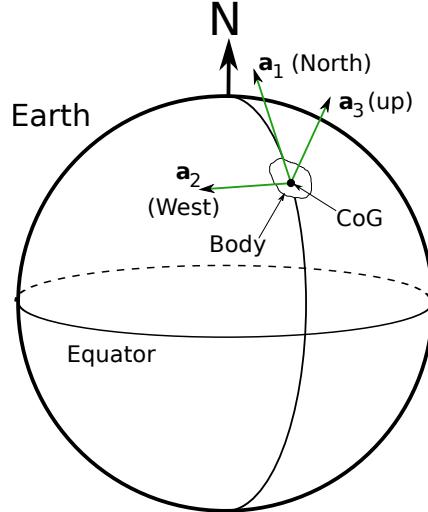


Figure 2.3: Diagram describing how the  $E$  frame is defined.

$$\mathbf{g}^E = -|\mathbf{g}|\hat{\mathbf{a}}_3 \quad (2.3)$$

$$\mathbf{B}^E = B_x \hat{\mathbf{a}}_1 + B_z \hat{\mathbf{a}}_3, \quad B_x > 0 \quad (2.4)$$

where  $\mathbf{g}$  and  $\mathbf{B}$  are the gravitational and magnetic field vectors respectively at the location of the body.

### Celestial frame ( $C$ )

The celestial frame has its axes defined with respect to visible stars. Its origin can be placed in different locations for various purposes. Its reference points, the surrounding star field, are so far away that their motion and the frame's motion within the solar system do not produce significant rotations of the frame. Attitude with respect to this frame can be termed "absolute attitude" because it is essentially invariant with respect to the relative motion and positions of solar system bodies.

### Inertial frames ( $\eta$ )

An inertial frame is defined as a frame in which Newtons laws hold (i.e. A frame in which an object that is not subject to any forces will not accelerate). Inertial frames can have an origin anywhere and their axes defined in any direction as long as motion of mass within that frame follows Newtons laws. The inertial frame is of course useful when calculating kinematics and dynamics. When defining an inertial frame, the unit vectors  $\hat{\mathbf{n}}_1$ ,  $\hat{\mathbf{n}}_2$  and  $\hat{\mathbf{n}}_3$  are used.

### External frames

External frames (i.e. Frames whose origin are not coincident with the origin of the body frame) are used when the ADCS wants to align, or determine the attitude of, the body with respect to the position of a distant object. The alignment vector in the external frame is calculated by subtracting the position vector of the alignment target by the position vector of the body. The ADCS must then convert the alignment vector into the  $\beta$  frame. External frames can only be used if the ADCS has knowledge of the position of the body and the position of the target with respect to the external frame.

## 2.2.2. Attitude notation

A variety of mathematical methods are used to express attitude, that is to say express the relative orientation of two cartesian reference frames that share an origin. Different methods are used because they have different properties, lending them application-specific advantages and disadvantages. This section explains the attitude notation methods relevant to discussing this project.

### Direction cosine matrices

A Direction Cosine Matrix (DCM) is a literal expression of the coordinate axis unit vectors of one reference frame in the coordinate system of the other reference frame. Equations 2.5 and 2.6 show how the relative orientation of two reference frames  $\beta$  and  $\eta$  that share an origin can be expressed in DCM. [25]

$$[R_{\eta\beta}] := \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}_1^\eta & \hat{\mathbf{b}}_2^\eta & \hat{\mathbf{b}}_3^\eta \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{n}}_1^{\beta T} \\ \hat{\mathbf{n}}_2^{\beta T} \\ \hat{\mathbf{n}}_3^{\beta T} \end{bmatrix} \quad (2.5)$$

$$C_{ij} := \hat{\mathbf{n}}_i \cdot \hat{\mathbf{b}}_j \quad (2.6)$$

The  $T$  subscript implies transposition and the square brackets imply that the symbol represents a matrix.  $[R_{\eta\beta}]$  is called the DCM or rotation matrix that maps the  $\beta$  frame to the  $\eta$  frame. The columns of  $[R_{\eta\beta}]$  are the unit axis vectors of  $\beta$  in terms of  $\eta$  and its rows are the unit axis vectors of  $\eta$  in terms of the  $\beta$ . The values in  $[R_{\eta\beta}]$  are the cosines of the angles between the axis vectors the two frames. DCMs can be used to take a vector expressed in one frame and express it in the other through matrix multiplication [25]:

$$\mathbf{r}^\eta = [R_{\eta\beta}] \mathbf{r}^\beta \quad (2.7)$$

DCMs are orthonormal, which means that they can be inverted by transposition [25]:

$$[R_{\eta\beta}]^T = [R_{\eta\beta}]^{-1} = [R_{\beta\eta}] \quad (2.8)$$

Sequential frame transformations can be performed through matrix multiplication of DCMs. For example, if attitude in terms of  $E$  ( $[R_{E\beta}]$ ) had been determined and the frame transformation from  $E$  to  $C$  ( $[R_{CE}]$ ) was known, attitude in terms of  $C$  ( $[R_{C\beta}]$ ) could be determined as follows:

$$[R_{C\beta}] = [R_{CE}] [R_{E\beta}] \quad (2.9)$$

DCMs are the most straightforward means of expressing attitude and they are very versatile. They remain the most effective means of converting vectors through frame transformations and most conversion between other attitude notations involves first converting to DCM. However, DCMs require far more variables than necessary to define a rotational state, making them cumbersome for attitude determination software, and it is difficult to visualize what they represent. The other attitude notation methods were designed to solve these issues.

### Euler Angles

Euler angles define a series of body rotations about axes in the  $\beta$  frame that, when performed in sequence, constitute the transformation from one frame to another. Euler angles are best known for their use in aviation. Yaw ( $\psi$ ), pitch ( $\theta$ ) and roll ( $\phi$ ) describe a transformation from the  $E$  frame to the  $\beta$  frame through successive rotations of  $\beta$  about  $\hat{b}_3$ , then  $\hat{b}_2$ , then  $\hat{b}_1$  with respect to the  $E$  frame. Figure 2.4 provides a visualization of a frame transformation through Euler angles.

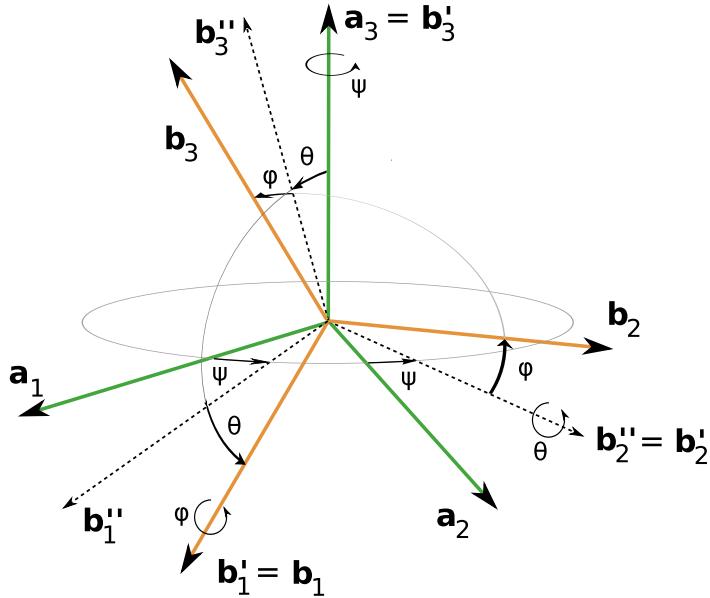


Figure 2.4: Visualization of the frame transformation described in Equation 2.13.

Euler rotations can be expressed relatively simply in DCMs [25]:

$$\text{Rotation of angle } \psi \text{ about } \hat{b}_3: [R_\psi] = [R_{E\beta''}] = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$\text{Rotation of angle } \theta \text{ about } \hat{b}_2: [R_\theta] = [R_{\beta''\beta'}] = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.11)$$

$$\text{Rotation of angle } \phi \text{ about } \hat{b}_1: [R_\phi] = [R_{\beta'\beta}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.12)$$

The full transformation would then be:

$$[R_E\beta] = [R_{E\beta''}][R_{\beta''\beta'}][R_{\beta'\beta}] = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.13)$$

$\theta$  and  $\phi$  are always defined with respect to the local horizontal. When  $\theta = 0$ ,  $\hat{\mathbf{b}}_1$  is perpendicular to  $\mathbf{g}$  and when  $\phi = 0$ ,  $\hat{\mathbf{b}}_2$  is perpendicular to  $\mathbf{g}$ . Definitions for the reference point of  $\psi$  vary. In aviation,  $\psi$  is often defined with respect to the velocity vector, which is convenient for calculations involving aerodynamic forces. In this thesis,  $\psi$  is defined with respect to local magnetic north. So when  $\psi = 0$ ,  $\hat{\mathbf{b}}_1$  is perpendicular to  $\mathbf{B} \times \mathbf{g}$ . The most unambiguous name for this definition of  $\psi$  is "heading" rather than "yaw". Further mention of  $\psi$  will refer to heading.

Many series of body axis rotations can be used to define attitude in terms of Euler angles. The definition mentioned above is called a (3-2-1) convention (rotation about  $\hat{\mathbf{b}}_3$ , then  $\hat{\mathbf{b}}_2$ , then  $\hat{\mathbf{b}}_1$ ). Any convention could be used so long as two successive rotations are not about the same axis (e.g. (1-2-3) or (3-1-3)).

Euler angles are particularly useful because they are the intuitive to the user. They make interpretation and analysis of attitude data, be that from measured data or software simulations, more transparent. They are also convenient for calculation of external forces due to natural phenomena such as gravity, magnetic fields and aerodynamics. Euler angles have a singularity, implying that the system does not behave properly, when the second rotation causes the first and third rotation axes to be parallel with respect to the initial frame. When in this orientation, there are infinite possible solutions to attitude, which would cause an ADCS algorithm to behave strangely and require an inconvenient singularity escape implementation. The following attitude notation methods provide solutions to this issue.

### Principle rotation vector

The principal rotation vector comes from Euler's principal rotation theorem: *A rigid body or coordinate reference frame can be brought from an arbitrary initial orientation to an arbitrary final orientation by a single rigid rotation through a principle rotation  $\Phi$  about the principal axis  $\hat{\mathbf{e}}$ ; the principle axis is a judicious axis fixed in both the initial and final orientation* [25].  $\hat{\mathbf{e}}$  and  $\Phi$  can be derived from a DCM with the following:

$$\cos \Phi = \frac{1}{2}(C_{11} + C_{22} + C_{33} - 1) \quad (2.14)$$

$$\hat{\mathbf{e}} = \frac{1}{2\sin\Phi} \begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \quad (2.15)$$

Where  $C_{ij}$  follows the same convention stated in Equation 2.6.

Principal rotation vectors use few variables to describe attitude, they remove the singularity issue encountered with Euler angles and they indicate a "shortest path" rotation between two frames, which is desirable when addressing attitude control. Having attitude solely expressed as  $\hat{\mathbf{e}}$  and  $\Phi$  is inconvenient when attempting to sequence frame transformations and address kinematics. Quaternions, Rodriguez parameters, and Modified Rodriguez Parameters (MRP) are alternative expressions of  $\hat{\mathbf{e}}$  and  $\Phi$  that have elegant and efficient mathematical solutions to those two issues. ALL-STAR's ADCS is based in MRPs so they will be discussed in further detail.

### Modified Rodriguez parameters

MRPs have the following definition in terms of  $\hat{\mathbf{e}}$  and  $\Phi$  [25]:

$$\boldsymbol{\sigma} := \tan\left(\frac{\Phi}{4}\right) \hat{\mathbf{e}} \quad (2.16)$$

The MRP  $\boldsymbol{\sigma}$  defining a sequential frame rotation of  $\boldsymbol{\sigma}'$  followed by  $\boldsymbol{\sigma}''$  is determined as follows:

$$\boldsymbol{\sigma} = \frac{(1 - |\boldsymbol{\sigma}'|^2)\boldsymbol{\sigma}'' + (1 - |\boldsymbol{\sigma}''|^2)\boldsymbol{\sigma}' - 2\boldsymbol{\sigma}'' \times \boldsymbol{\sigma}'}{1 + |\boldsymbol{\sigma}'|^2|\boldsymbol{\sigma}''|^2 - 2\boldsymbol{\sigma}' \cdot \boldsymbol{\sigma}''} \quad (2.17)$$

The MRP  $\boldsymbol{\sigma}''$  describing the direct frame rotation from  $\boldsymbol{\sigma}'$  to  $\boldsymbol{\sigma}$  is determined similarly:

$$\boldsymbol{\sigma}'' = \frac{(1 - |\boldsymbol{\sigma}'|^2)\boldsymbol{\sigma} - (1 - |\boldsymbol{\sigma}|^2)\boldsymbol{\sigma}' + 2\boldsymbol{\sigma} \times \boldsymbol{\sigma}'}{1 + |\boldsymbol{\sigma}'|^2|\boldsymbol{\sigma}|^2 + 2\boldsymbol{\sigma}' \cdot \boldsymbol{\sigma}} \quad (2.18)$$

### 2.2.3. Determination

Attitude determination is a process of using sensors to detect the angular position of exterior objects with respect to the satellite body that define a reference frame that the attitude of the satellite can be expressed in terms of. For the purpose of explaining the concept, let us use a hypothetical attitude determination system that has nothing to do with PolarCube. (*Disclaimer: the following description serves only to explain the concepts behind the mathematics of attitude determination. It has nothing to do with other attitude determination systems mentioned in the document*)

We can start by giving the system a sun sensor. A sun sensor determines the body axis vector that points from the satellite to the sun. This already provides the ability to point any fixed satellite body vector towards or away from the sun. If the satellite knows what the sun pointing vector is in  $C$ , then it can also determine the direction of the sun pointing vector in  $\beta$  with respect to  $C$ . This would require the time and an ephemeris model of the satellite's orbit with respect to the sun. Only having a sun sensor is insufficient to determine attitude because it can only determine a single reference vector; thus rotation about the sun pointing vector is not constrained. To determine attitude, a second measurement must be taken that is not parallel to the first one. This could, for an Earth orbiting satellite, be achieved through measurement of the ambient magnetic field combined with knowledge of the ambient magnetic field's direction in the  $C$  frame. This would require knowledge of the satellite's position with respect to Earth, a model of Earth's magnetic field, the time and a model of Earth's rotation with respect to  $C$ .

The attitude measurements from the sun sensor and magnetometer would have some error and uncertainty. It is likely that one would have more than another. So, algorithms that use those measurements to define attitude can favor information from one sensor over another, provided that the measurements are not perpendicular. Error can also be reduced by conducting redundant measurements, such as measuring a third reference vector.

A certain amount of uncertainty will always remain in attitude determination. This is troublesome for two reasons, it limits pointing accuracy and makes it difficult to determine the angular rate of the satellite. Knowledge of angular rate is particularly important for attitude control. The best way to solve this is to use rate measurement gyroscopes. These directly measure the angular rate of the satellite. Rate gyroscope measurements can also be integrated to provide an additional, noise free source of attitude determination. Integration of rate gyro measurements will inevitably result in drift, but can be very accurate over the short term. This would be valuable when filtering measurements made with external sensors and filling in periods when external measurements are not available (e.g. When the Sun sensor is eclipsed by the Earth).

## 2.3. Attitude control theory

Attitude control is based on a negative feedback loop between actuator torques and determined attitude error. Attitude error is defined as a frame transformation that rotates the desired attitude onto the current attitude. Attitude error can be expressed in any of the attitude notations mentioned above. An attitude control law is an algorithm that determines the appropriate torques to produce in the actuators given an attitude error and information about the state of the system. Figure 2.14 illustrates the process of attitude control in a functional block diagram. The rest of this section first uses a one-dimensional example to explain the concepts that control laws are based upon, then explains the control law that was selected for the PolarCube mission.

### 2.3.1. Control law basics

For demonstration purposes, we can define a rigid body with body frame  $\beta$  that is constrained to rotate about the  $\hat{\mathbf{n}}_3$  axis of an inertial frame  $\eta$  that shares an origin with  $\beta$ .  $\hat{\mathbf{b}}_3$  and  $\hat{\mathbf{n}}_3$  are parallel. The body has a Mass Moment Of Inertia [MMOI] of  $1 \text{ kgm}^2$  about  $\hat{\mathbf{b}}_3$ . The resulting dynamics of the body are expressed in Equation 2.19:

$$\tau = I\dot{\omega} \quad (2.19)$$

Where  $I$ ,  $\omega$  and  $\tau$  are the MMOI, angular rate and external torque applied to the body about the rotation axis respectively. Attitude of the body can be expressed by  $\theta$ , the angle between  $\hat{\mathbf{n}}_1$  and  $\hat{\mathbf{b}}_1$  (see Figure 2.5). Attitude error is defined in Equation 2.20:

$$\theta_e := \theta - \theta_t \quad (2.20)$$

Where  $\theta_t$  is the target attitude. This section will describe how attitude of this one-dimensional system can be controlled with a Proportional-Integral-Differential(PID) control law.

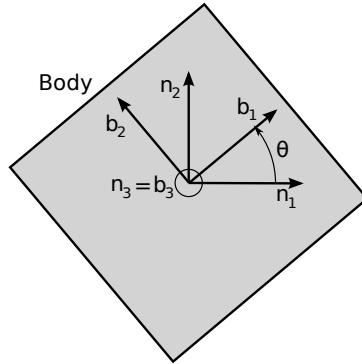


Figure 2.5: Diagram showing how  $\beta$ ,  $\eta$  and  $\theta$  are related in this example.

**Proportional control** Torque imparted to the body through proportional control is defined in Equation 2.21.

$$\tau_P := -K\theta_e \quad (2.21)$$

Where  $K$  is the proportional control gain. The result of only using proportional control is simple harmonic motion. The blue line in Figure 2.6 shows the  $\theta_e$  response to an initial  $\theta_e$  of 1 rad under proportional control with a  $K$  value of 1 with no other external torques.  $\theta_e$  does not converge.

### PD control

PD control produces convergence of  $\theta_e$  by adding damping, hence the D for derivative. PD control is defined in Equation 2.22

$$\tau_{PD} = -K\theta_e - P\omega \quad (2.22)$$

Where  $P$  is the derivative gain. The dashed red line in Figure 2.6 shows the  $\theta_e$  response to an initial  $\theta_e$  of 1 rad under PD control with a  $K$  value of 1 and a  $P$  value of 1 with no other external torques. This results in damped harmonic motion,  $\theta_e$  converges to 0 under these conditions.

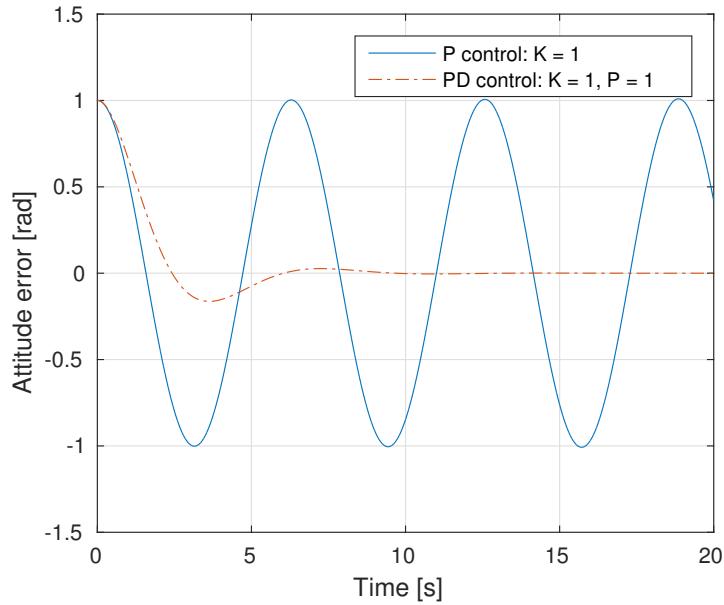


Figure 2.6: Responses of P and PD control laws to an attitude error of 1 rad.

### PID control

When a persistent external torque  $\tau_{ext}$  is introduced, PD control is no longer able to converge  $\theta_e$  to 0.  $\theta_e$  will instead converge to a value of  $\tau_{ext}/K$ . The blue line in Figure 2.7 shows the  $\theta_e$  response to an initial  $\theta_e$  of 1 rad

under PD control with a  $K$  value of 1, a  $P$  value of 1 and a persistent external torque about  $\hat{b}_3$  of 0.1 Nm. PID control addresses this by adding a third term to the control law:

$$\tau_{\text{PID}} = -K\theta_e - P\omega - K_I \int_0^t \theta_e dt \quad (2.23)$$

Where  $K_I$  is the integral control gain. The red line in Figure 2.7 shows the  $\theta_e$  response under the same conditions as the blue line except that integral control is added with a  $K_I$  value of 0.1.

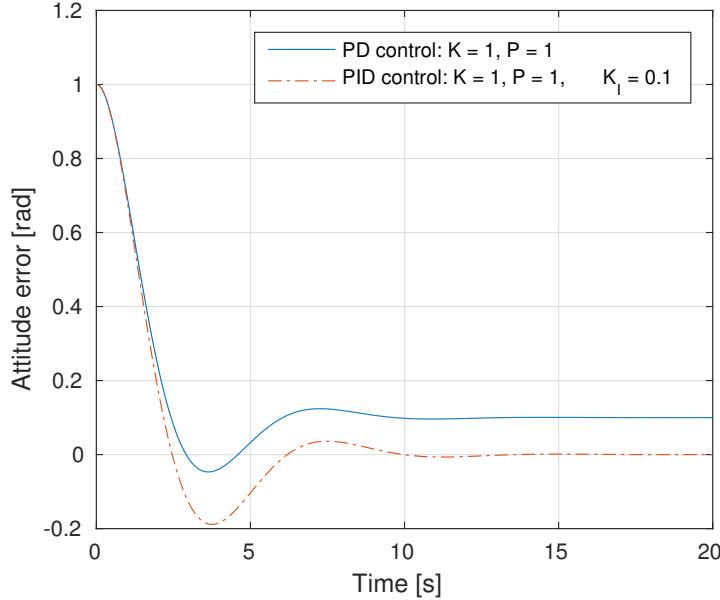


Figure 2.7: Responses of P and PD control laws to an attitude error of 1 rad and an external torque of 0.1 Nm.

### 2.3.2. Three dimensional rigid body dynamics

Rotation of a rigid body in three dimensions is defined by Equation 2.24 expressed in the  $\beta$  frame [25]:

$$\boldsymbol{\tau} = [I]\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times [I]\boldsymbol{\omega} \quad (2.24)$$

Where  $\boldsymbol{\tau}$  is the sum of torques acting on the body,  $\boldsymbol{\omega}$  is the angular velocity vector and  $[I]$  is the MMOI tensor of the body. Adding reaction wheel gyroscopic (Page 16) results in Equation 2.25 as an equation of motion for a satellite with 3 reaction wheels that spin parallel the axes of  $\beta$ :

$$\boldsymbol{\tau} = [I]\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times [I]\boldsymbol{\omega} + \boldsymbol{\omega} \times \mathbf{h}_s \quad (2.25)$$

$$\mathbf{h}_s := I_{W_s} * \boldsymbol{\Omega} \quad (2.26)$$

Where  $\mathbf{h}_s$  is the net angular momentum of the wheels and  $\boldsymbol{\Omega}_i$  is the spin rate of a reaction wheel about  $\hat{b}_i$ . Angular acceleration ( $\dot{\boldsymbol{\omega}}$ ) can be determined by inverting Equation 2.25 [3]:

$$\dot{\boldsymbol{\omega}} = [I]^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times [I]\boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{h}_s) \quad (2.27)$$

## 2.4. Polarcube's Attitude determination and Control Subsystem (ACS)

ADCS is necessary because many satellites depend on their orientation with respect to other objects in order to perform. This is the case for systems that interact with electromagnetic radiation, such as solar panels, radios and imaging instruments, as well as for propulsion and shielding systems. This section describes how PolarCube's ADCS, dubbed ACS, has been designed to be able to point PolarCube.

### 2.4.1. Hardware architecture

ACS consists of a main board, a star camera board and three actuator assemblies. The main board houses (see Figure 2.8):

- The ACS processor
- MEMS sensors
- Local radiation immune memory
- Connectivity to the Command and Data Handling subsystem (CDH)
- Connectivity to the actuator driver boards and the star tracking camera

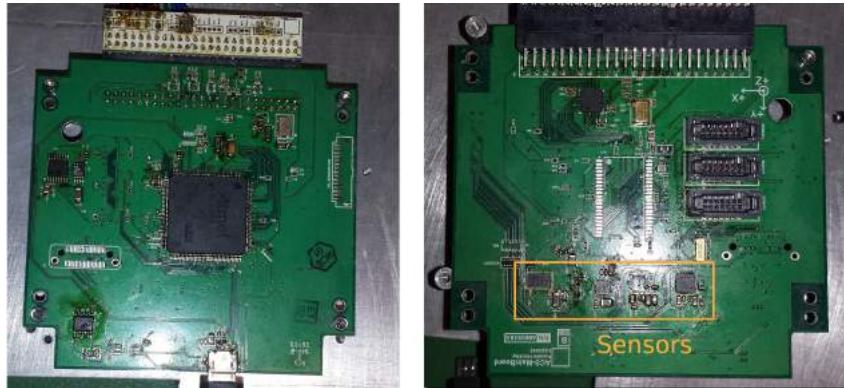


Figure 2.8: Image of a later revision of the ACS main board on its ground support equipment. The attitude determination sensors are indicated in orange. Photo credit to Adam St. Amand.

Each actuator assembly consists of a structural metal card upon which a reaction wheel, magnetorquer and a driver PCB have been mounted. The actuator assemblies are aligned such that one reaction wheel and one magnetorquer are aligned parallel to each of the satellite's primary axes (see Figure 2.9). The star camera board consists of a CMOS sensor with focusing optics and an independent processor.

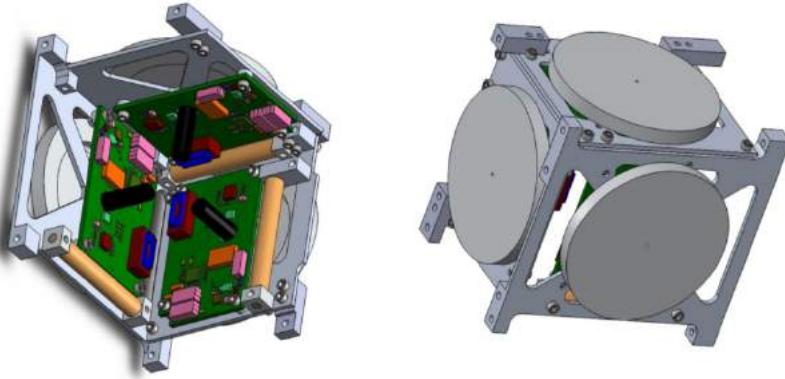


Figure 2.9: CAD renderings of the ACS actuators.

At the start of the thesis project, the decision was made to re-design ACS from a previous configuration to the one detailed above for reasons independent of the thesis project. This re-design included using new processors, a new software architecture and creating new revisions of all of the PCBs. In the previous design, communication had been established with the magnetometer and the board was shown to be able to rotate a reaction wheel and control the current through a magnetorquer. Franklin Hinckley was responsible for ACS PCB design.

### 2.4.2. Sensors

ACS uses a suite of sensors that are used to determine the satellite's attitude. This section describes what the sensors do and how they are used.

### Star tracking camera

The star tracking camera on ALL-STAR serves as ALL-STAR's only source of absolute attitude determination. It takes images of space and identifies the four brightest stars in the image. The identity of the stars is determined by measuring the area of the four triangles that the stars describe in the image and comparing to an on-board star catalog. This provides four reference vectors from which ADCS can determine attitude. The star camera will provide a lock every few seconds, attitude between star camera locks will be determined by integrating gyro measurements. The star camera has a maximum rotation rate above which it cannot produce a lock. When rotating too fast, attitude determination relies entirely on the gyros.

### Rate gyroscopes

The current revision of ALL-STAR uses two ST Microelectronics A3G4250D MEMS 3-axis rate gyroscopes (see Figure 2.10). They measure the angular rate of the body with respect to inertial space. In flight, the measured rates will be used by the attitude control algorithm and they will be propagated to provide attitude determination between star camera locks. The A3G4250D has a sensitivity of 8.75 mdeg/s per count [28].

### Magnetometer

The revision of ALL-STAR that was used in this project used one Honeywell HMC5883L 3-axis MEMS magnetometer (see Figure 2.11). It has since been swapped out for two ST Microelectronics LIS3MDL magnetometers for a higher sampling rate and redundancy. In flight, the magnetometers will primarily be used to aid the magnetorquers. They will measure the ambient magnetic field so that the appropriate current to send through the torque rods can be determined. They will also serve to verify that torque rods are firing. Magnetometers will also be used to provide an estimate of the angular rate of the spacecraft during de-tumble operations. In this thesis, the HMC5883L was configured to operate at its highest sensitivity: 0.73 mG/count [13].



Figure 2.10: Image of the A3G4250D MEMS gyro.

Figure 2.11: Image of the HMC5883L magnetometer.

### GPS

Polarcube uses a NovAtel GPS unit. The GPS unit uses signals from GPS and GNSS satellites to determine the position of the satellite with respect to the surface of the Earth and feeds coordinates to the ACS processor. This data will then be used when the satellite needs to point at something on the surface of the Earth, such as the ground station or when imaging with the payload. The GPS unit also provides GPS time to the satellite, which is useful for a number of functions.

#### 2.4.3. Actuators

ADCS actuators produce torque against the satellite body in order to induce the desired rotational motion. ALL-STAR uses two actuator systems to perform attitude control: reaction wheels and magnetorquers. This section describes what they are and how they function.

##### Reaction wheels

Reaction wheels consist of a free spinning wheel whose rotation axis is fixed in the body frame of the satellite. By applying torque to the wheel, thus accelerating it, the satellite produces a torque on its own body in the opposite direction. These torques are what is used to control the motion of the satellite. In performing any operation with a reaction wheel, the angular momentum of the total satellite system is conserved. A (simplified) slew maneuver (rotating from one fixed attitude position to another) would consist of spinning a reaction wheel up until the satellite reaches the desired angular velocity, maintaining that reaction wheel speed until the satellite has reached the desired attitude, then de-spinning the reaction wheel to stop the satellite's angular motion at the desired attitude. Precession of a reaction wheel by rotating the satellite body out of the axis of a

spinning wheel will result in gyroscopic torques being exerted on the test model body by the reaction wheel. Equation 2.28 defines gyroscopic torques according to rigid body dynamics:

$$\boldsymbol{\tau}_{\text{gyr}} = \boldsymbol{\Omega}_p \times \mathbf{H} \quad (2.28)$$

Where  $\boldsymbol{\Omega}_p$  is the angular velocity of precession and  $\mathbf{H}$  is the angular momentum of the wheel. Equation 2.29 is derived from Equation 2.24 (Page 13) by assuming  $\dot{\boldsymbol{\omega}} = 0$  and that  $I_{W_s} >> I_{W_t}$  where  $I_{W_s}$  is the MMOI of the wheel about the spin axis and  $I_{W_t}$  is the MMOI of the wheel about an axis perpendicular to the spin axis.

The fact that reaction wheels are conservative of angular momentum means that they can not compensate for external torques on the satellite that would result from tidal forces, magnetic dipoles, aerodynamic forces and radiation. To compensate for secular (persistent in a single direction) external torques, reaction wheels must continue to accelerate. This poses a number of issues:

- There is a saturation limit on reaction wheel speed after which the wheels can no longer accelerate
- Keeping reaction wheels at a high spin rate consumes more power.
- Available torque produced by the reaction wheel motor reduces as reaction wheel spin rate increases.
- Reaction wheels that are spinning quickly produce greater gyroscopic torques when rotated out of the reaction wheel's spin axis.

### Magnetorquers

Magnetorquers solve the reaction wheel spin-up problem by being able to provide an external torque to the satellite. Magnetorquers consist of a solenoid wrapped around a piece of ferrous metal mounted within the satellite. Running current through the solenoid creates a magnetic dipole that torques the satellite by interacting with the ambient magnetic field of Earth. Equation 2.29 (Page 16) defines magnetic dipole torque.

$$\boldsymbol{\tau}_{\text{mag}} = \boldsymbol{\mu} \times \mathbf{B} \quad (2.29)$$

where  $\boldsymbol{\mu}$  is the magnetic dipole vector and  $\mathbf{B}$  is the ambient magnetic field vector. Magnetorquers serve to unload the excess angular momentum stored in the reaction wheels. The torque that magnetorquers generate is dependent on the direction and magnitude of the ambient magnetic field. For this reason, magnetorquers are always used in combination with magnetometers. Care must be taken such that the ambient magnetic field can be determined despite perturbations generated by the magnetorquers and other satellite equipment.

#### 2.4.4. PolarCube control law

PolarCube uses an MRP based nonlinear Lyapunov control function developed by Hogan and Schaub [12]. The following is an adaptation of the control law presented in the paper that only works for three reaction wheels that spin parallel to the body axes:

$$\mathbf{u}_s = -[I](\dot{\boldsymbol{\omega}}_r - \boldsymbol{\omega} \times \boldsymbol{\omega}_r) + K\boldsymbol{\sigma}_e + [P]\delta\boldsymbol{\omega} + [P][K_l]\mathbf{z} - ([\tilde{\boldsymbol{\omega}}_r] - [\tilde{K_l}\mathbf{z}])([I]\boldsymbol{\omega} + \mathbf{h}_s) + \mathbf{L} \quad (2.30)$$

Where  $\mathbf{u}_s$  is the desired output torque to apply to the wheels,  $\boldsymbol{\sigma}_e$  is the attitude error MRP,  $\boldsymbol{\omega}_r$  is the rate of the attitude target with respect to inertia, and  $\mathbf{L}$  denotes any known external torques.  $[P]$  and  $[K_l]$  are diagonal matrices. The tilde denotes the skew symmetric cross product matrix of a vector:

$$\tilde{\mathbf{r}} := \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \quad (2.31)$$

$\delta\boldsymbol{\omega}$  is the angular rate difference between the body and the target:

$$\delta\boldsymbol{\omega} := \boldsymbol{\omega} - \boldsymbol{\omega}_r \quad (2.32)$$

And  $\mathbf{z}$  is the MRP error integrator term:

$$\mathbf{z} := \int_0^t (K\boldsymbol{\sigma} + [I]\delta\dot{\boldsymbol{\omega}}) dt \quad (2.33)$$

Reaction wheels and magnetorquers serve two independent functions in ACS. All attitude control is performed by the reaction wheels and the magnetorquers serve exclusively to dump angular momentum stored

in the reaction wheels. The actuator sets each have a single control law that is always controlling them. The gains and targets of the reaction wheel control law (Equation 2.30) define the control modes that are described below. This design architecture stems from a general design principle of minimizing complexity. There is a serious concern of making a satellite that is "too smart". By minimizing the number of elements (hardware or software) in the overall system, one minimizes the number of elements that can fail.

#### 2.4.5. ADCS control modes

The PolarCube mission has a Concept of Operations (CONOPS) that describes the actions that the satellite will perform to achieve the mission objectives such as collecting useful radiometric data and sending it back to the ground station. The CONOPS defines the ADCS control modes required for operations. The requirements that ACS is designed to meet flow down from these control modes. This section describes the control modes, how they flow down from the CONOPS and the requirements they place on the ADCS.

##### De-tumble

The de-tumble control mode reduces the angular rate of the satellite body produced when it is ejected from the P-Pod upon orbit insertion. A dedicated control mode is necessary under high angular rates for two reasons:

- The star tracking camera will not be able to produce an attitude estimate after angular rate passes a threshold.
- The reaction wheels can only absorb a limited amount of angular momentum.

The de-tumble control mode uses the same control law as the rest of the control modes, the difference being that the potential gain and integral gain are set to 0 and the differential gain is set low so that the magnetorquers are able to keep the reaction wheels from saturating (see Sections 2.3.1 on Page 11 and 2.4.4 on Page 16 describing control law implementations). The de-tumble control mode uses magnetometer readings to determine the angular rate of the satellite rather than the gyros because the vibrations of launch will likely introduce bias into gyro measurements.

##### Sun Pointing

"Sun Soak" is PolarCube's default attitude control mode. It should point the satellite such that power collected through the solar panels is maximized while making sure that the star camera can function (i.e. pointing away from Earth). The position of the Sun relative to the satellite is determined by combining a locally stored ephemeris model of Earth's orbit with the time known from the GPS system and the absolute attitude determined from the star tracker.

##### Science

The "Science" attitude control mode keeps the payload optics pointing within  $2.3^\circ$  of nadir to maintain a minimum spatial resolution at the satellite ground track. The payload mirror is designed such that the payload is "looking" at an angle of  $45^\circ$  from the bus structure when collecting science data (see Figure 2.12). The satellite must be tilted upward by an angle of 45 degrees from the velocity direction to keep the payload pointing nadir during payload data collection periods (see Figure 2.13). This involves a control loop that keeps a vector that represents the center of the payload's scan pattern pointing nadir and otherwise aligns the body of the satellite axially with the velocity vector to produce an optimal spatial sampling pattern. Payload data collection occurs over a 60 degree segment of the Earth orbit. PolarCube uses GPS to determine when to enter and exit science operation as well as determine the nadir direction.

##### Uplink/Downlink (COM)

The "COM" control mode should point the S-band antenna to the ground station in Boulder to within  $15^\circ$  of accuracy. This is achieved by combining position knowledge of the satellite through GPS with a target determination function that maintains the satellite's attitude with respect to a specified point on Earth's surface. The satellite will also rotate about the pointing vector to ensure that the star tracker is not obscured by the Earth and maximum power is being generated through the solar panels.

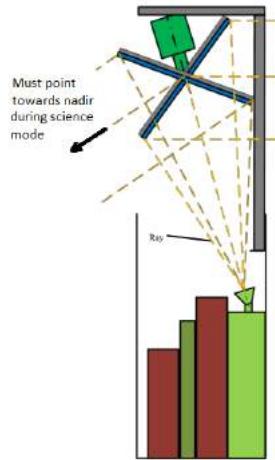


Figure 2.12: Concept drawing of Mini-Rad showing the mirror's scanning pattern [4].

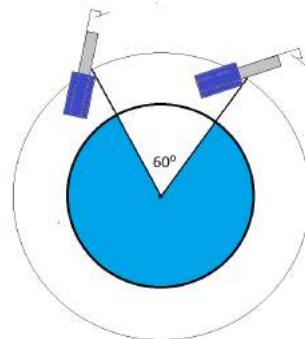


Figure 2.13: Concept drawing showing PolarCube's alignment with respect to earth when performing science operations.

#### 2.4.6. Functional block diagram of ACS in flight

Figure 2.14 serves as a visual representation of how the systems in ACS interact with each other and the environment to perform attitude control in orbit. Note how it is based on a control loop that relies on negative feedback of satellite motion and attitude.

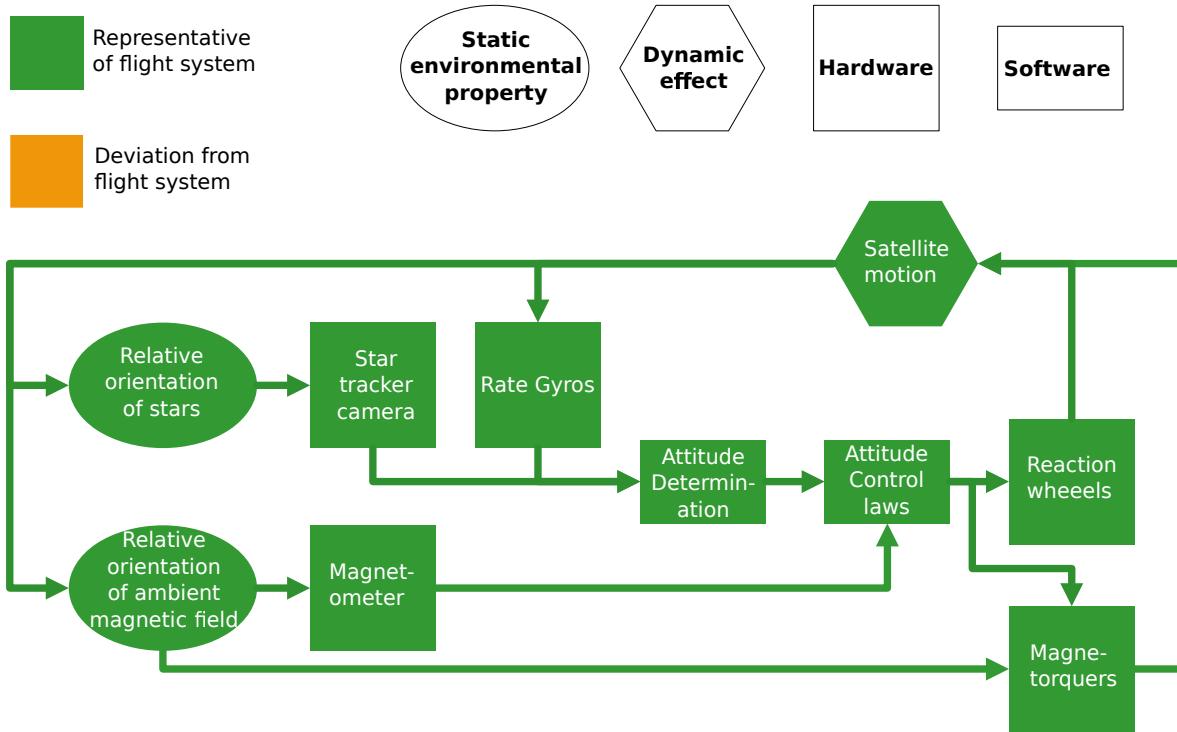


Figure 2.14: A functional block diagram showing the interplay of elements of ACS when performing attitude control.

#### 2.4.7. PolarCube mission requirements to verify

Table 2.1 lists requirements for the PolarCube satellite that were identified as ones that could potentially be verified through (or at least with the aid of) attitude control validation testing. These requirements are driven directly from PolarCube mission goals and exist independently of the validation testing project. They were taken from the official PolarCube Requirements Verification Matrix; hence the references in X.ADCS.Y format.

TBR indicates that the requirement is "To Be Reviewed/Revised"

Table 2.1: PolarCube mission requirements that could be verified with validation testing.

| REF       | Requirement  | Rationale   |
|-----------|--|---|
| 2.ADCS.5  | The subsystem shall achieve 3-axis attitude control of the PolarCube spacecraft.   | Necessary in order to gather sufficient power, payload data and downlink data using the S-band antenna.                                     |
| 2.ADCS.8  | The subsystem shall achieve angular rate below TBD deg/s within 5 minutes (TBR) of being powered on.                               | Primarily due to rate threshold for star camera to function. eventually necessary for pointing requirements. Time limit somewhat arbitrary. |
| 2.ADCS.10 | The subsystem shall be capable of tolerating rotation of the mirror lens of at least 1 Hz, up to a nominal maximum of 3 Hz.        | Nyquist frequency of the scan pattern.  |
| 2.ADCS.13 | The subsystem shall be able to point the payload's optics nadir during science operations to within 2.3 deg (TBR).                 | Nadir has best spatial resolution. 2.3 deg is the angular resolution of the payload.  |
| 2.ADCS.15 | The subsystem shall be able to maintain an angular rate about the nadir vector of below 0.1 deg/s during science operations (TBR). | Image quality metric. Threshold somewhat arbitrary.   |
| 2.ADCS.16 | The subsystem shall be able to point the solar panels towards the sun during sun soak operations to within $\pm 15$ deg .          | Driven from power budget.   |
| 3.ADCS.5  | The subsystem will control the attitude of PolarCube with the use of three reaction wheels.  | Chosen as method to meet pointing accuracy and agility requirements.  |
| 3.ADCS.10 | The subsystem will use magnetic torque rods to prevent reaction wheel saturation.  | Reaction wheels will saturate without ability to apply secular torques.   |
| 3.ADCS.13 | The subsystem shall achieve science mode pointing requirements within 10 min from beginning of science operations.                 | Driven from orbit schedule.   |

## 2.5. Chapter summary

PolarCube is a 3U CubeSat being developed by the COSGC at CU Boulder to perform microwave imaging of the troposphere. PolarCube consists of the ALL-STAR 3U satellite bus and the MiniRad microwave radiometer. ALL-STAR's first flight was with the THEIA payload, it has since been majorly re-designed and is currently in the process of integrating and verifying "silver" revisions of its hardware.

Attitude is expressed as a rotation from a reference coordinate frame to a reference frame that is fixed with respect to the satellite's body. This can be done using a number of mathematical constructs. This thesis discusses the use of DCMs, Euler angles, principal rotation vectors and MRPs. Attitude is determined using sensors that can relate the orientation of the body frame to the reference frame by measuring phenomena that are constant in the reference frame. Attitude is usually measured using a number of instruments simultaneously each with some uncertainty. Methods exist to produce the best estimate of attitude given a set of measurements and knowledge of their error properties.

Control laws use the determined attitude and body rotation rate to determine the torques that must be produced in order to orient the satellite where desired. Control laws must be able to make attitude converge at the target given the rigid body dynamics of the satellite and any external torques present. PolarCube uses a Lyapunov Control law based on MRPs.

ACS controls the orientation of the satellite by producing torques on the satellite body with reaction wheels and magnetorquers. The appropriate torques are decided using information from the on-board star tracking

camera and MEMS gyros and magnetometer. The orientation of the satellite must be controlled to orient the satellite for best energy collection through the solar panels, communication with the S-band antenna or data collection with the payload.

# 3

## Design of the test system

This chapter presents the design of a test system (testbed, support systems and methods) for validation testing of PolarCube's attitude control performance. It includes:

- A discovery the top level requirements of the project based on stakeholder needs.
- A trade study of microgravity simulation concepts and how they can be applied to test attitude control performance.
- A discussion of means to improve upon established software modeling techniques of the attitude control system's dynamic response for the purpose of producing metrics of attitude control performance.
- The design of an engineering model of the satellite that provides ACS with the necessary support systems and functionality to perform the tests envisaged.
- An overview of the test system design including a breakdown of project requirements and visualizations of how the test system functions.

Throughout the chapter, the reader will find references to formalized project requirements in the following format: (X.ACSVT.Y). This indicates that the preceding sentence(s) present the argument for that particular requirement. Table 3.4 on Page 49 contains the full list of requirements.

### 3.1. Top level system engineering

This section explains the process of deriving the most fundamental requirements on the design of the test methods through assessment of project stakeholder needs.

#### 3.1.1. Stakeholder needs

The research objectives stated in Section 1.3 flow down from the project stakeholders and their needs stated below:

**The COSGC** The COSGC is developing PolarCube, a mission to perform passive microwave sensing of the thermosphere. PolarCube's attitude control system has performance requirements that must be met in order to meet mission success criteria. Validating the attitude control system's performance before launch will increase the probability of mission success.

**TU Delft** TU Delft has a CubeSat program of its own. Research on attitude control validation methods will provide an opportunity to conduct validation tests as part of their own program. The academic impact of the research conducted will also reflect well on the university.

**Future test engineer** The test methods developed are envisaged eventually be conducted by engineers at the COSGC and engineers at TU Delft and other CubeSat programs. The testing methods should be accessible to an engineer with limited experience and resources.

### **3.1.2. Research objectives**

The stakeholder needs led to the research objectives re-stated here for reference:

- Develop a test system (testbed, support systems, and methodology) that allows the performance of PolarCube's attitude control system to be measured in terms of its ability to meet ADCS functional requirements. (0.ACSVT.1)
- The methods developed should be accessible to future test engineers at the COSGC and other educational CubeSat programs. (0.ACSVT.2)

The research objectives can be re-stated as a set of research questions that the project will attempt to answer:

- What is the best method to conduct pre-launch validation tests of PolarCube's attitude control system given the resources available and the needs of the stakeholders?
- What would the results of the validation tests reveal about PolarCube's attitude control performance?

### **3.1.3. Top level requirements**

This section details the top level requirements that need to be met in order to meet the stated research objectives. The rest of the chapter will reveal lower level requirements that can all be derived directly from those listed in this section. The full list of requirements is supplied in Section 3.5.1.

#### **Performing attitude control**

The system must be able to perform attitude control in order for its performance to be tested. (Requirement 1.ACSVT.1) In order for an attitude control system to function, it requires the following:

- **Power:** Most satellite attitude control systems (PolarCube's included) make use of electronics and do not have independent power supplies, so power shall to be provided to the attitude control system. (1.ACSVT.2)
- **Attitude determination:** Attitude control systems rely on knowledge of the system's attitude to determine output torques, so attitude determination shall be provided. (1.ACSVT.3)
- **Freedom of motion:** When an attitude control system is on the surface of the Earth, gravity forces the system to make contact with external objects, introducing friction. Frictional forces are too great for an attitude control system designed for use in microgravity to induce motion in the body frame. The system shall be subjected to a physical environment that induces low enough external torques that torques produced by the attitude control system dominate its motion. (1.ACSVT.4)

#### **Measuring attitude control performance**

Measuring attitude control performance is a matter of generating, recording, retrieving and analyzing test data. These shall all be done in such a way that data analysis produces metrics of attitude control performance that can be used to verify ADCS functional requirements, such as those in Table 2.1 (Page 19). (1.ACSVT.6) The following have been identified as metrics that would be valuable to pursue:

- **De-tumble:** The ability of the system to reduce angular rates below a threshold. This is the most fundamental function of an attitude control system. This function is usually tested first because it is the easiest to test. Testing this would help verify PolarCube mission requirement 2.ADCS.8.
- **Slew:** The ability of the system to control where it is pointing. Testing this would help verify PolarCube mission requirement 3.ADCS.5.
- **Pointing stability:** The accuracy to which the system can control attitude. Attitude will never converge absolutely, so the degree to which it converges should be measured. Testing this would help verify PolarCube mission requirements 2.ADCS.13, 2.ADCS.15 and 2.ADCS.16
- **Momentum dumping:** The ability of the system to dump angular momentum stored in reaction wheels. Testing this would help verify PolarCube mission requirement 3.ADCS.10.
- **Mirror influence:** The influence on performance produced by other spacecraft elements. In the case of PolarCube, the primary concern would be the influence that the payload mirror has on attitude control performance. Testing this would help verify PolarCube mission requirement 2.ADCS.10.

## **Accessibility**

The methods should be easy to adopt by a future test engineer at the COSGC and other educational CubeSat programs. Thus the methods developed should be inexpensive, minimize complexity, and use widely available resources. (1.ACSVT.7) The methods should also be transparently documented. (1.ACSVT.8)

## **3.2. Microgravity simulation**

Microgravity is the state of not feeling gravitational forces as a result of free-falling motion. Satellite systems are designed to operate under the freedom of motion that is afforded in microgravity and will typically not function on the surface of Earth due to friction with whatever is supporting it. Microgravity simulation is an effort to reduce the external torques on the system such that the attitude control system is able to control its motion in a way that is similar to what would occur in orbit. For attitude control systems, it is important to distinguish that only freedom of rotation is necessary and freedom of translation is not relevant and typically adds unnecessary complexity, especially considering that PolarCube does not use thrusters for attitude control. It is also important to distinguish that static friction is unacceptable for microgravity simulation because it trivializes any attempt at measuring pointing stability. (1.ACSVT.5) Even the slightest changes in torque to the body should affect its motion.

A literature study was conducted in preparation for the thesis project that outlined the state of the art of CubeSat attitude control testing research and helped define the topic of this thesis project. The literature study included an investigation into the known methods of microgravity simulation. It can be achieved through use of air bearings [26], suspension by a string [22][20][24] and feeding hardware input and output through a computer simulation [19]. This section outlines the testbed concepts that were considered based on implementation of these techniques including how the testbeds would be used to produce attitude control performance metrics. A trade study is then laid out based on the information provided to explain the eventual decision to implement a string suspension testbed.

### **3.2.1. String suspension**

The simplest method to allow an object to rotate free of friction is to suspend it by a string. Three phenomena dominate the external torques produced by a string suspension testbed: pendulum stability, elastic deformation and aerodynamic drag. Pendulum stability removes the gravitational torques on the system and elastic deformation removes friction because no surfaces will slide against each other. The result is motion that is significantly different from motion that would occur in orbit, but still meets requirements 1.ACSVT.4 and 1.ACSVT.5.

#### **External torque Properties**

When discussing motion of an object that is suspended by a string, the first concept to address is pendulum stability. Pendulum stability describes what happens when an object's translation is constrained at a point on its body within a gravity field under motion damping (air drag). The orientation of the object will align such that the Center of Gravity (CoG) of the object will be along the line of action of the gravity vector through the attachment point. This is due to torques produced by gravity acting on the CoG when it is not aligned with the attachment point. A string suspension testbed can be represented as a system of two pendulum stable systems. The first is the string that is fixed on one end with a massive object (test subject) attached to the other. Both attachments are technically rigid (knots), but they can be considered to allow free rotation due to the string's flexibility. This keeps the test subject fixed in a specific location and keeps the string straight. The second occurrence of pendulum stability is the test subject attached at some point on the outside of its body to the end of the string. This will result in the test subject aligning itself such that its center of gravity is aligned with the attachment point. The test subject's motion is thus constrained to rotation about the local vertical and the axis about which the test subject rotates in the body frame is determined by where the string attaches to the test subject's body.

Elastic deformation comes into play when investigating the nature of the attachment point of the test subject to the string. Given that the freedom of rotation is afforded by the flexibility of the string, rotation of the test subject about the local vertical will twist the string about its long axis and its deformation will incur restoring torques. If a monofilament string is used, Hooke's law of elastic deformation would suggest that the restoring torques would be proportional to the angle by which the string is twisted. (1.ACSVT.12) This property of elastic deformation takes the place of friction in allowing the test subject to move and is superior because it is more predictable (in theory) and it does not have the "static friction property" of a minimum value.

The test subject would also be subject to aerodynamic drag as it rotates. The drag torque would be expected to be directly proportional to angular velocity provided that air flow over the test subject is laminar [11]. The resulting sum of external torques on the test subject would be analogous to a damped rotational spring about the local vertical. This results in the Equation 3.1 for torques about the local vertical:

$$\tau_{\text{ext}} = -k\theta_d - c\omega \quad (3.1)$$

Where  $\theta_d$  is the angular deflection about the local vertical from the string's equilibrium position. Previous implementations of the string suspension testbed have operated on this "damped spring" assumption but have not conclusively proven it to be true. [22] [20]

### External torque characterization (Oscillation test)

The oscillation test is an adoption from tests conducted for the MicroMAS mission [22] and it is used to determine the external torque properties of the string suspension testbed. The test subject is manually rotated away from the equilibrium position of the string and released without performing any attitude control. The test subject will then oscillate as a result of the spring torques produced by the string. The attitude response is measured and a model of string testbed dynamics is fit to the measured response. The fitted model is now a predictor of string testbed external torques as a function of attitude and rotation rate. The external torque parameters determined in this test can then be used to predict the responses of future attitude control tests on the same testbed.

Figure 3.1 shows the results of an oscillation test published in [22] as part of the MicroMAS attitude control test campaign. Both lines in each plot show angular deflection of the test subject from the string's equilibrium angle ( $\theta_d$ ). The red line is output from a Simulink model of damped harmonics (Equation 3.1) that has been fit to the measured response in blue. There was a distinct limit to how well the simulation output could be made to fit the response, primarily manifested as an apparent reduction in damping as the system approaches rest. This could be the result of:

- The test subject's motion not corresponding to damped harmonic motion
- Error in measurement of the test subject's motion.
- Improper definition of initial conditions of the dynamics model.
- Any combination of the above.

The time that the oscillation test takes is a function of  $k$ . The data in Figure 3.1 was produced with a 50 lb test fishing line. Considering that a 3U CubeSat weighs a maximum of 4 kg, there is clearly room to reduce spring torques (and thus allow motion to be more realistic when conducting attitude control) without risking that the string snaps by making the string longer and/or thinner, but the time it would take to conduct the oscillation test would increase.

### Potential validation tests

The following is an explanation of how the string suspension testbed can be used to determine the performance metrics outlined in Section 3.1.3 (Page 22). This is done for all of the testbed proposals and used to help determine the viability of the testbed.

**Damped oscillation (de-tumble) test** The de-tumble test is also an adoption from [22]. It uses the oscillatory motion of the test model on the string to test the performance of a de-tumble control mode. De-tumble control modes are designed to reduce the angular rate of the satellite after orbit insertion. They typically have some sort of D control law (see Section 2.3.1, Page 11) where torque output is proportional to angular rate. The test model is again released at a deflection angle from the equilibrium point, this time with de-tumble control active. The resulting oscillation will see a significant increase in damping when compared to the uncontrolled oscillation test as a result of attitude control. De-tumble control would in theory produce a static increase in the value of  $c$  in Equation 3.1 (see Equation 3.2). The damped response can be predicted using string torque parameters determined in the oscillation test and the expected behavior of the attitude control system.

$$\tau = -k\theta_d - (c_{\text{ext}} + c_{\text{out}})\omega \quad (3.2)$$

Figure 3.2 shows the results of a de-tumble test published in [22]. The red line is a best fit of a simulation assuming torques on the body are defined by Equation 3.1 to the measured data in blue. De-tumble performance was measured as the difference between the resultant (testbed + attitude control) damping factor ( $\zeta$ )

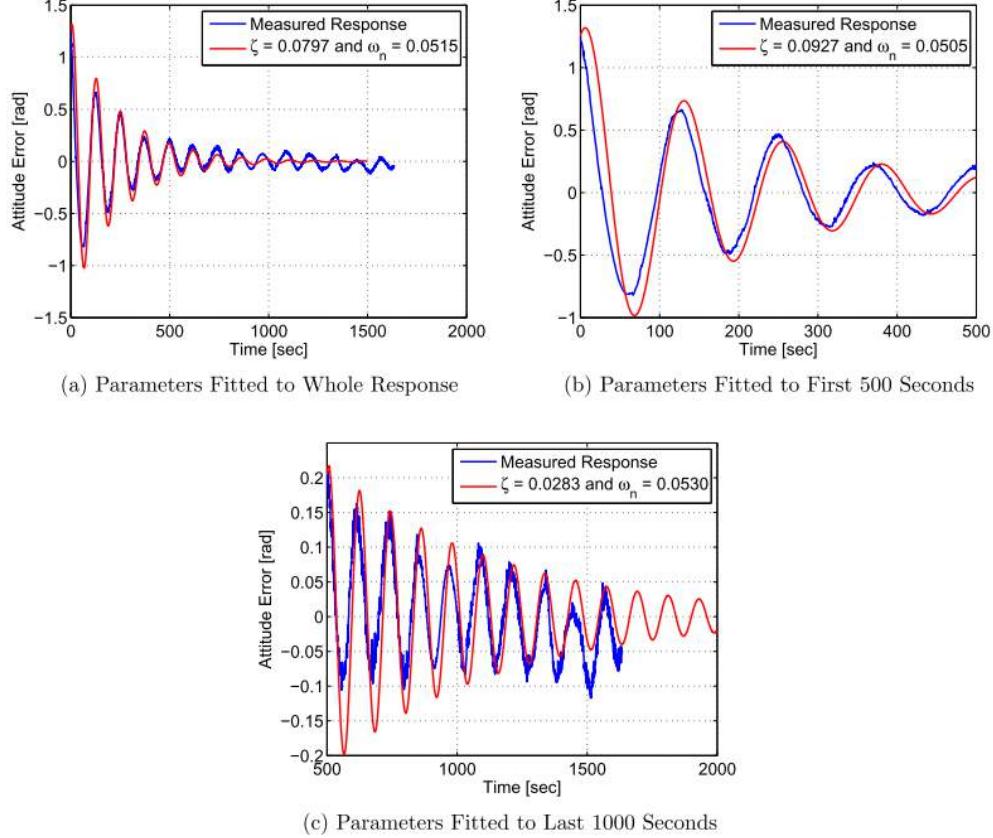


Figure 3.1: A measured oscillation test deflection angle response plotted against output from damped spring models fitted to match different portions of the response. [22]

that was predicted through analysis and the one that corresponded to the best fit to the measured data. The predicted change in damping factor was within 4% of the change in damping factor corresponding to the best fit to the measured response.

The test first and foremost served to verify the mission requirement that the system will reduce angular velocity below a threshold. However, it more importantly provided a quantitative measure of the correlation between the measured performance of the system to its designed performance. This is valuable for a number of reasons:

- It provides a higher degree of qualification to the customers.
- It provides greater insight into points of failure if the system does not pass the test.
- Provided that the test shows that translation of commanded torques to actual output torques is sufficiently transparent, the control law itself can be verified independently through computer simulation.
- It gives a measure of confidence in future simulation of satellite motion responses that have not been directly tested.

The last property is particularly important for string suspension testbeds because the measured motion is generally very different from what will eventually occur in orbit thanks to the rotation constraint and inherent external torques. The information gained in this test can allow engineers to produce a software simulation of what the satellite's motion would be in orbit given a set of initial conditions and claim that it is accurate to within known parameters. The test results could also be used to make adjustments in the simulation to better conform to the measured properties.

In summary, the concept of producing performance metrics is as follows:

1. Fit a software model of testbed dynamics to an uncontrolled response of the test subject's motion to known initial conditions.
2. Use the fitted software model to produce a prediction of the test subject's motion response when performing attitude control under a set of defined initial conditions.
3. Fit a software model to a measured motion response of the test subject to the same initial conditions used to make the prediction.
4. Compare the parameters of the fitted model to the one used to make the prediction.

The author has dubbed this a "fit-predict-fit" method of producing metrics of attitude control performance. It eventually became the focus of the thesis' research. Having knowledge of the external torque properties of the testbed is critical to producing motion prediction. (1.ACSV.T.11) This becomes an issue with testbed concepts that are discussed further.

Regardless of whether a dedicated de-tumble control mode will be implemented in flight. This is a valuable first test because it minimizes complexity by using the most basic control law. Results of a de-tumble test will be the easiest to analyze and passing a de-tumble test would suffice to verify many of the attitude control system's functions.

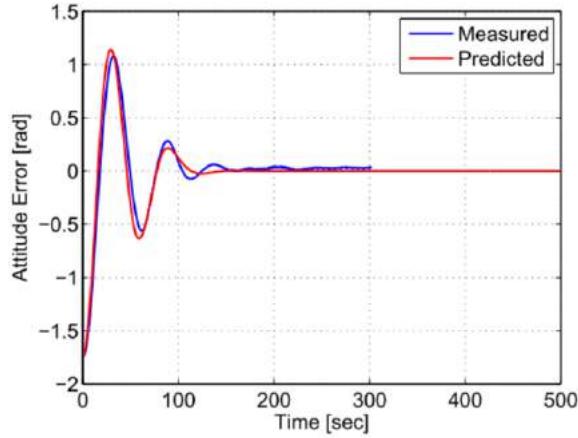
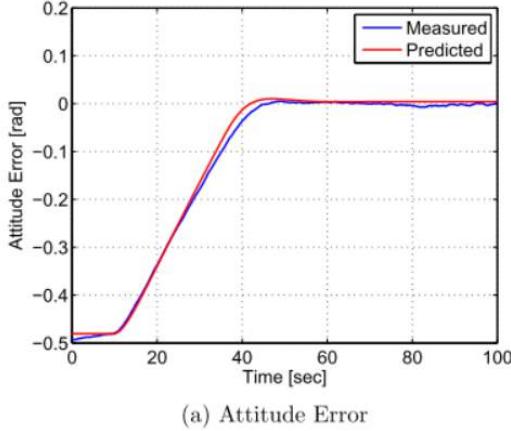


Figure 3.2: Measured response of oscillation of an attitude control system under de-tumble control compared to a prediction from a software model of testbed dynamics.

**Slew** A slew maneuver consists of a rotation of the satellite from being at rest at one attitude to being at rest at another attitude. This can be tested on the string suspension testbed by a slew maneuver about the rotation axis. The simplest slew test involves having the test subject maintain attitude at the equilibrium position of the string, then introducing an attitude target that consists of a rotation of the test model's body frame about the rotation axis. The external torques produced by the testbed will result in a different response to what would be experienced in space, but the response on the testbed can be predicted using the string torque characterization results from the oscillation test. In this case, analysis could consist of comparison of predicted and observed P, I and D gains. Figure 3.3 shows the results of a slew test on a string suspension testbed published in [22].

Figure 3.4 is a magnified view of the data in Figure 3.3. Figure 3.4 shows that pointing stability of the attitude control system was significantly impacted by the string's restoring torques at the target attitude, pulling the test model back towards the equilibrium position once it reached the target. The test verified that the system could perform a predictable slew maneuver, but it could not verify pointing stability. The behavior of the PolarCube test model under the same conditions may be different because it uses a different control law.

**Pointing stability** The issue in pointing stability measurement displayed in Figure 3.4 could be circumvented by measuring the attitude control system's ability to maintain attitude at the equilibrium position where no external torques are imparted by the testbed. The results of this test are mostly independent of string torque characterization. It might be valuable to minimize the spring coefficient by using a long thin string and foregoing time consuming string torque characterization to get the most realistic data. This test is also largely



(a) Attitude Error

Figure 3.3: Measured and predicted results of a slew test published in [22].

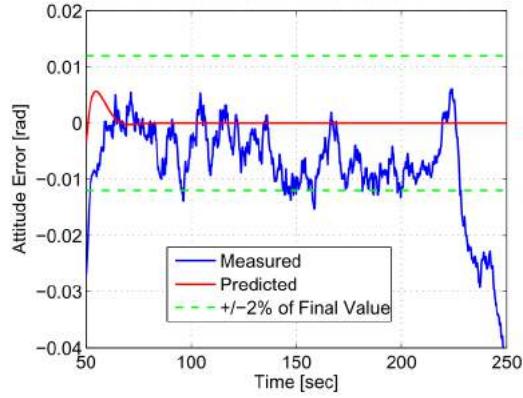


Figure 3.4: A magnification of the data in Figure 3.3 showing how the strings restoring torques are compromising stability at the target attitude.

independent of attitude determination accuracy. Overall, this test, which has yet to be conducted, provides a very important metric of attitude control performance with remarkably low requirements on the testbed and attitude determination.

**Momentum dumping** The ability to dump angular momentum stored in the reaction wheels could be measured at the equilibrium position of the string because at that point the string won't be imparting any torques in the direction of motion. The test would start with the attitude control system maintaining attitude at the equilibrium position with the parallel reaction wheel spinning at a high rate. Momentum dumping would then be initiated and the response of the system attempting to maintain attitude while slowing the wheel down could be analyzed and compared to a prediction. The fact that this test also consists of attempting to maintain attitude at the equilibrium position means that it shares the same benefits as the pointing stability test in terms of string torque characterization and attitude determination performance.

**Testing influence of the payload mirror** Options were considered for testing the influence that the spinning payload mirror would have on attitude control performance. The payload mirror spins at a rate of 1 Hz when performing science and is stationary otherwise. It is also not symmetrical about its axis of rotation. Performing tests of payload mirror influence will require the payload mirror to be suspended with the attitude control system. Three sources of torque on the satellite body produced by the mirror were identified:

- Gyroscopic torque produced by precession of the spinning mirror during the 60° science pass. (see Figure 2.13, Page 18)
- Torque produced when the mirror is spun up and down when entering and exiting science mode.
- Vibrations produced by the centrifugal force of the asymmetrical spinning mirror.

The single degree of freedom constraint of the string suspension testbed makes it impossible to test gyroscopic effects of any kind. The option of suspending the test model along the mirror spin axis to observe the effect of spinning it up and down was considered, but the resulting torques were estimated and considered too small to warrant concern about attitude control performance and the test was considered not worth the effort. The torques produced by asymmetry of the mirror have yet to be modeled and their magnitude is unknown, but they could potentially be partially observed by suspending the test model about any axis other than the mirror spin axis.

### 3.2.2. Spherical air bearing

Spherical air bearings are designed to constrain all translation while providing freedom of rotation about 3 axes. Air bearings work by forcing air between two parallel surfaces. The air keeps the two surfaces from touching each other while external forces push them together, essentially acting as a lubricant with extremely low viscosity. In the case of a spherical air bearing, the parallel surfaces are a solid sphere and a cup that it rests in.

Unless the satellite model can be fit entirely within the sphere, it must be attached to the surface somewhere. This restricts the motion of the bearing. Typically, spherical air bearings consist of equipment mounted atop a moving hemisphere, allowing free rotation about the vertical and limited motion about horizontal axes. (see Figure 3.5)



Figure 3.5: Image of a spherical air bearing [27].

### **External torque properties**

In order to allow freedom of motion without the influence of gravity, the Center of Gravity (CoG) of the moving section (sphere + satellite model) must be located at the Center of Rotation (CoR) of the bearing (center of the sphere). Misalignments give a moment arm to gravity, producing gravitational torques and corrupting the microgravity simulation. Achieving alignment that is sufficient for 3-axis microgravity simulation is especially difficult for systems on a nanosatellite scale. [22] [14] [16]

All air bearings produce some torque due to asymmetry of air flow between the two surfaces. The properties of these torques have not been well documented and they probably vary greatly depending on bearing design. Conversation with engineers from Astro- und Feinwerktechnik who worked on their attitude control testbed [23] revealed that years of development and funds on the scale of an entire student CubeSat mission went into developing an air bearing that produces external torques on the order of  $10^{-5}$  Nm and can support 150 kg of test subject mass. Torque performance certainly improves as scale decreases, but considering the resources available for student CubeSat missions and the available torque that some of them have ( $5.5 \cdot 10^{-6}$  Nm for Delfi n3Xt reaction wheels [6]), this effect may be a limiting factor depending on the quality of the bearing and the performance of the attitude control system.

A spherical air bearing system can be turned into a single axis of freedom microgravity simulation by making it pendulum stable. The further the distance between the CoG and the CoR, the more stable the system is. Many CubeSat attitude control test programs that use spherical air bearings have used them in pendulum stable configurations because the CoG alignment requirements are much more forgiving [22], [10]. Using a pendulum stable configuration removes any gravitational torques on the test subject about the axis of motion. The constraint of motion to a single axis also makes analysis of test data more straightforward. Shperical air bearing testbeds that test motion in three axes often make their system very slightly pendulum stable (with a very small distance between the CoG and CoR) to make it easier to start the test with an upright test subject. It is often a challenge to bring the CoG far below (or even near) the CoR because the test subject typically rests above the bearing. Solving this either involves an "umbrella" architecture where the equipment to be tested is suspended from the ends of a plate sitting on top of the bearing (see Figure 3.6) or suspending counterweights from the test subject (see Figure 3.7). Both of these methods significantly increase the MMOI of the test subject, reducing the realism of its motion [22]. The "umbrella" design also makes it difficult to test attitude control while rotating about different body axes because equipment would need to be re-aligned [22]. Another potential solution that has never been attempted would be to use a vacuum pre-loaded spherical air bearing to be able to suspend the model upside-down. Air bearings can be designed to incorporate a vacuum pump that pulls the two surfaces together while the forced air keeps them very slightly apart, alleviating the need for gravity to push them together [17].

### **External torque characterization**

As previously mentioned: the existence of torques produced by airflow in the bearings has been largely confirmed through word of mouth, but little if nothing has been published on how those torques behave or how



Figure 3.6: An example of an "umbrella" architecture implemented on CubeSat hardware [22].

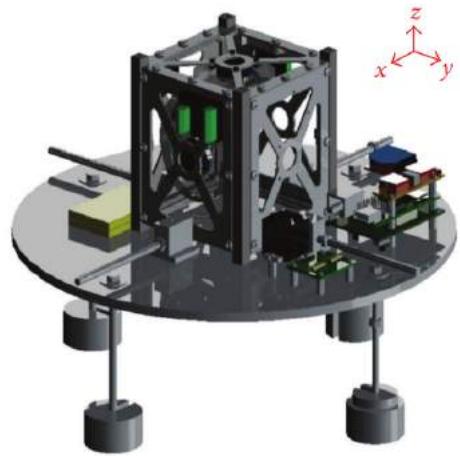


Figure 3.7: An example of using counterweights to lower the center of gravity of the test subject [14].

they could be measured. One could conceivably design a test that measures an uncontrolled attitude response of the test subject and reverse-engineers the data to determine the torques present as a function of attitude, similarly to how external torques on a string suspension testbed can be determined with an oscillation test. This would also greatly be simplified if the system is made pendulum stable.

Overall, it can not be assumed at this point that external torques on a spherical air bearing can be known and used to predict the attitude response of a test subject. Developing methods to determine them would be an interesting field of study, but it does not fall into the scope of this project because it would be expected to consume too much time and funds.

#### Potential validation tests

The following explains how a spherical air bearing testbed could be used to provide the performance metrics outlined in section 3.1.3. The options of centering the CoG (3 DOF) and using a pendulum stable configuration (1 DOF) are both considered.

**Testing de-tumble control** One could envision a test where the test subject starts at a given speed and the attitude and rates are recorded as the attitude control system attempts to reduce rates. This would verify the de-tumble requirement, but without knowledge of external torque properties, it would not provide much more information than that.

**Testing pointing control** An engineer at the Laboratory for Atmospheric and Space Physics at CU Boulder built a spherical air bearing in his garage out of plastic to test attitude control of the Miniature X-ray Solar Spectrometer (MinXSS) 3U CubeSat being developed there [15]. They were successfully able to test 3-axis attitude control of the satellite using the XACT attitude determination and control system purchased from Blue Canyon Technologies. The XACT has an output torque capacity of  $4 \cdot 10^{-3}$  Nm [2], 730 times that of the Delfi n3Xt attitude control system. The MinXSS test has shown that, provided sufficient torque output capacity, attitude control of a CubeSat in three axes can be validated in a general sense (in that attitude would converge on the target) using a spherical air bearing. However, they were unable to provide detailed metrics of the system's dynamic performance.

Using a pendulum stable configuration has the advantage over the centered configuration in that gravitational torques are removed from the equation. This puts lower requirements on torque output capacity and saves time in setting up the testbed [22]. The question then remains whether a pendulum stable spherical air bearing is superior to a string suspension testbed. In terms of performance, the only difference between the two is the magnitude and quality of torques produced about the rotation axis. Comparing the magnitude of torques produced by the two is difficult because the torque produced by the spherical bearing is unknown and the magnitude of torques produced by the string can be changed by altering the thickness and length of the string. In any case, it can at least be said that string suspension has the potential for extremely low external

torques. The string suspension testbed wins in terms of quality of torques because their properties can be determined with the oscillation test.

**Testing pointing stability** Figure 3.4 demonstrates how having a persistent external torque compromises determination of pointing accuracy. The constant presence of external torques in an air bearing testbed considerably limits the potential of conducting such a test.

**Testing momentum dumping** A momentum dumping test like that described for the string suspension testbed on an air bearing suffers from the same issue that a pointing stability test would. Persistent external torques would continue to add angular momentum to the system, compromising efforts to maintain reaction wheel speeds within a threshold. If the magnetorquers are able to output significantly higher torques than the testbed is introducing, then the test could be used to verify the requirement, but this is unlikely to be the case.

**Testing influence of the payload mirror** A spherical air bearing set up for three degrees of freedom provides opportunities for testing the influence of the spinning mirror that are prevented by constraining rotation to a single dimension. However, this would first require that the torques produced by the mirror are significant with respect to the external torques produced by the testbed, which is not guaranteed and unlikely.

### 3.2.3. Rotary air bearing

A rotary air bearing, also known as an air bearing spindle, uses air bearing principles to allow nearly frictionless motion about a single axis and constrain all other motion. (see Figure 3.8).

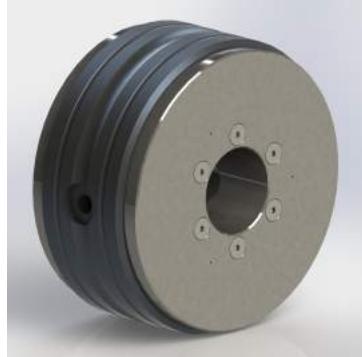


Figure 3.8: An example of a rotary air bearing [1].

#### External torque properties

By contrast with a spherical air bearing that is pendulum stable, where pitch torques (torques on the test subject that are perpendicular to the rotation axis) cause the model to rotate slightly, resulting in gravitational torques that counter them, the fact that a rotary bearing is mechanically constrained means that out of axis torques get transferred to the spindle housing. The first effect that this produces is that it solves the counterweight problem displayed in Figures 3.6 and 3.7. Mechanical constraint also provides an interesting measurement opportunity that is discussed later, but it also introduces an issue that is conveniently avoided in pendulum stable systems. If the CoG of the test subject is not located directly on the axis of rotation and the axis of rotation is not perfectly aligned with the gravity vector, gravitational torques will be produced about the rotation axis. This is a result of the bearing's normal force vector not being aligned with the gravity force vector, analogous to a frictionless ramp. The result is an external torque on the test subject about the rotation axis that is a sinusoidal function of rotation angle.

A software model of rotary air bearing testbed dynamics was produced to determine the necessary alignment tolerances. It simulated the motion of a 1.4 kg test subject with a 3U form factor and uniform density rotating about a constant body axis and the gravitational pitch torques that it would produce as a result of specified CoG and rotation axis alignment error. Figure 3.9 is output of the dynamic software model of the rotary air bearing testbed that shows the gravitational torque about the rotation axis as a result of 1mm CoG alignment error and 0.5° rotation axis alignment error. These conditions were considered reasonable alignment expectations considering the time and resource limitations on the project.

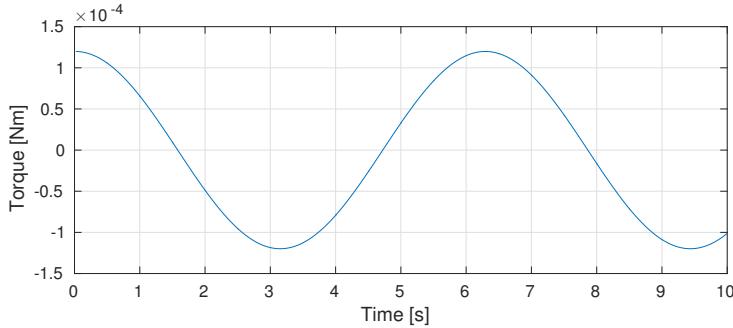


Figure 3.9: Gravitational torques about the rotary bearing's rotation axis as the test subject rotates.

Similarly to a spherical air bearing, a rotary air bearing would be expected to produce torques due to airflow, the properties of which would be entirely unknown before unit testing.

### External torque characterization

Similarly to the spherical air bearing, the uncertainty in torques due to air flow prevent the author from being able to assume that properties of testbed external torques about the rotation axis can be determined. Performing a simulation retro-fit to determine torques from uncontrolled motion for a rotary bearing would have an added level of difficulty when compared to a pendulum stable spherical air bearing system because about-axis gravitational torques will also come into play.

**Testing pointing control** The advantage of mechanical constraint over pendulum stability is that pitch torques produced by the attitude control system could potentially be measured by externally measuring the torques imparted to the spindle housing. A design was conceived to implement pitch torque measurement where the spindle was aligned to rotate about the local vertical and the housing was suspended on load cells. Three load cells would be used to make the system statically determinate and variations in the supporting force measured would be used to derive pitch torque variations on the bearing housing.

Measuring pitch torques provides an opportunity to conduct a test that is constrained about a single axis and measures the torque output performance of all three wheels. For example, consider the case of an attitude control system that is aligned such that it is free to rotate about one of its primary axes; implying that one of its reaction wheels is parallel to the rotation axis and the other two are perpendicular to it, and assume that all three of the wheels have an initial spin rate. If the attitude control system attempted to perform a slew about the rotation axis as if it were in space and unconstrained, then the parallel wheel would output torque to rotate the body and the two perpendicular wheels would have to output torque to compensate for each others gyroscopic torques. If the maneuver was true to what would be expected in space, then the testbed would measure zero pitch torques. Figure 3.10 is a free body diagram depicting this scenario. The perfect scenario, in which no torques are measured, would require a control law that could perform a maneuver about a fixed body axis. Most control laws do not do this, especially if the body is not symmetric about the rotation axis.

Gravity will produce pitch torques on the bearing unless the rotation axis is parallel to the gravity vector and the CoG of the test subject is located on the rotation axis. These torques will present as sinusoidal functions of the rotation angle about the X and Y axes of the bearing frame specified in Figure 3.10. The pattern of gravitational torques could, in theory, be characterized and removed from the net pitch torque measurement in post-processing. This could only be done if the amplitude of the gravitational torque signal was low enough that the torques produced by the test model could be distinguished from it given a certain amount of attitude uncertainty and quantization error. It was estimated that the amplitude of the gravitational pitch torque signal could be up to 10 times the pitch torques produced by reaction wheels for the concept to be worth pursuing.

Figures 3.11 and 3.12 demonstrate the results of a feasibility analysis of the proposed configuration. Figure 3.11 shows the gravitational pitch torque derived from a software model of the rotary bearing testbed as a result of 1mm CoG alignment error and 0.5° rotation axis alignment error. The resulting pitch torques have an amplitude of ~0.028 Nm. Figure 3.12 shows gyroscopic torque of a reaction wheel as a function of precession rate for different wheel speeds, as derived from Equation 2.28 (Page 16). The maximum feasible wheel speed was considered to be 1000 rad/s. The vertical black line indicates PolarCube's maximum satellite body angular rate expected on orbit (taken from a satellite dynamics model). It shows a maximum expected gyroscopic

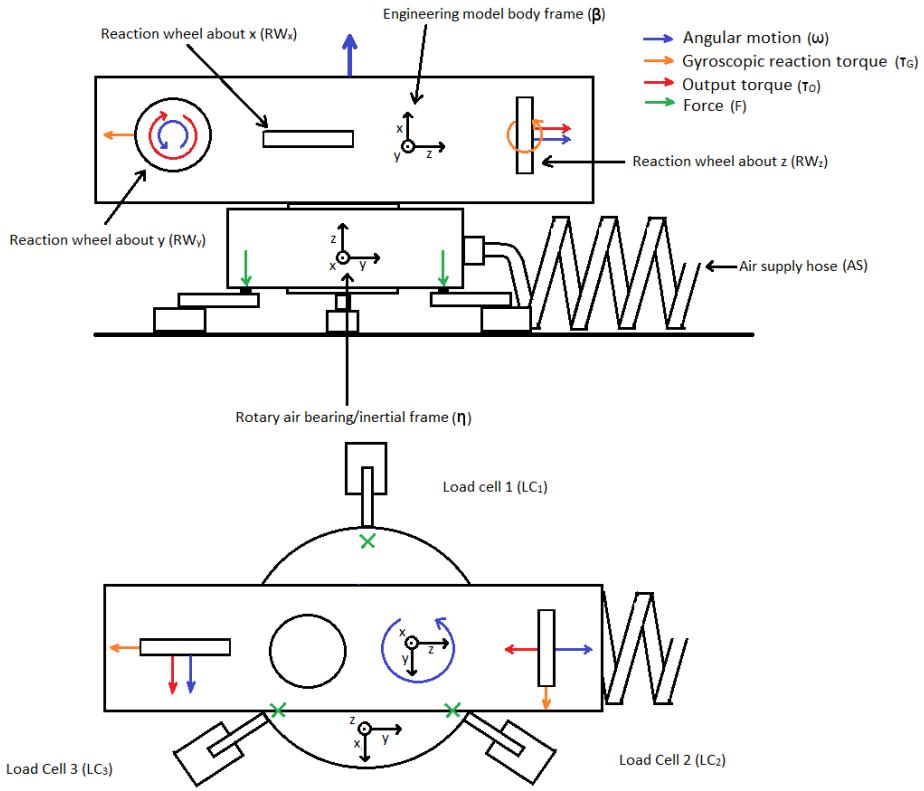


Figure 3.10: Free body diagram depicting slew motion about +x on the suspended rotary bearing testbed with reaction wheels rotating about +y and +z.

torque of  $2 \times 10^{-4}$  Nm, one hundredth of the expected gravitational pitch torques. The conclusion was that measuring pitch torques applied to a rotary air bearing would not be fruitful in this project but could be an interesting field of research for a project with the time and resources to align the system well.

This design still fares better than a perfectly aligned spherical air bearing in terms of alignment tolerances. If it is assumed that gravitational torques must be an order of magnitude lower than typical attitude control output torques for three axis testing on a spherical air bearing to be useful, then a rotary air bearing can tolerate two orders of magnitude higher gravitational torques than a spherical air bearing can.

Rotary air bearings provide the opportunity for very accurate attitude determination by implementing an optical encoder on the spindle. A pattern could be etched on the face of the spindle that the test subject is not attached to and a sensor could optically observe its motion without producing an external torque. It would be difficult for the test subject to access this information, but it could at least provide an absolute attitude reference.

Fancy load cell implementations aside, a rotary air bearing would encounter the same inability as a spherical air bearing to predict the motion response of the test subject until its external torque properties can be determined.

### Other potential validation tests

The rotary air bearing's ability to perform the other validation tests is otherwise the same as a pendulum stable air bearing aside from the following:

- The added complication of gravitational torques about the rotation axis.
- The benefit of not requiring a counterweight architecture.
- The potential for extra data by suspending the air bearing on load cells.

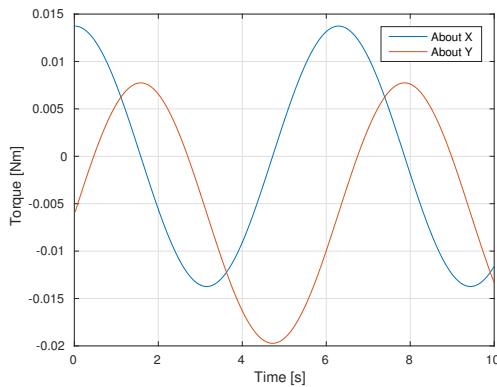


Figure 3.11: Torques to the supporting load cells about the rotary air bearing X and Y axes defined in Figure 3.10 as the test subject rotates.

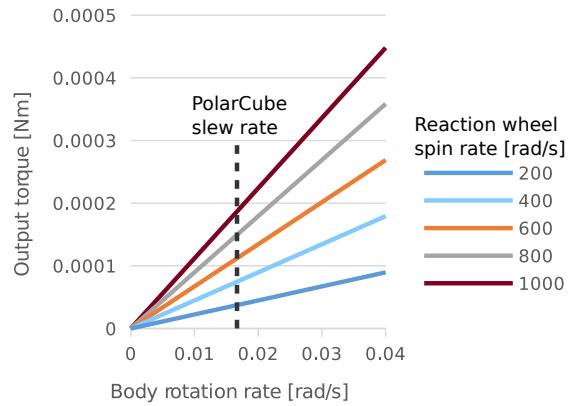


Figure 3.12: Magnitude of gyroscopic torque produced by a reaction wheel as a function of precession rate.

Given that using the load cells has been deemed unfeasible for this project, the question of which of the two to use would be whether the alignment requirements outweigh the inconvenience and performance impact of using a counterweight or umbrella architecture.

### 3.2.4. Parabolic flight

PolarCube had a rare opportunity to test attitude control in a parabolic microgravity flight provided by the Space Test Program as part of UNP-8. Microgravity flights provide true microgravity to the test subject by flying parabolic trajectories, effectively simulating free-fall. The goal was to test attitude control maneuvers that only used the propagation of rate gyros for attitude determination. The tests used most of the hardware architecture developed for this thesis project. They encountered a major issue of time constraints. Each parabola is approximately 20 seconds long. The stable release of the test subject, the attitude control maneuver and re-securing the test subject must all occur within this window. This is further complicated by the fact that the test engineers are university students who are in their first microgravity flight and thus are strapped to the floor for their own comfort and safety. Additionally, the plane's trajectory will not describe a perfect parabola. Although the test subject floating within the plane will indeed be experiencing microgravity, it will generally have an apparent acceleration with respect to the plane's walls. The test subject will have a tendency to float away from the test engineer and into something else. The PolarCube test engineers were lucky if they could provide the test subject with 5 seconds of testable microgravity. The flight did serve its primary function of verifying PEZ and mirror boom deployment with a frangibolt.

#### Potential validation tests

Ideally, a parabolic flight would permit all of the metrics listed in Section 3.1.3 (Page 22) to be tested by simply re-enacting flight conditions. However, the time constraint and inconvenience of the test environment essentially prevented any test from being executed successfully. The probability of producing useful test data could certainly be improved by a thorough upgrade of test systems and procedures, but it would be difficult to justify the cost.

### 3.2.5. Computer simulation

This concept involves using an external computer to simulate the outside environment of the attitude control system. The measured rates and accelerations of the reaction wheels and/or magnetorquer currents are fed to a computer that uses them to simulate motion of the satellite and feeds back simulated sensor information. This is typically called Hardware-in-the-Loop (HIL) testing. The feedback could involve creating sensor emulators that the attitude control system would read and use to determine attitude or simply sending the attitude through serial communication. This method avoids many of the complications of using a physical system, but the degree to which its results can be trusted is highly dependent on the degree of fidelity built into the system.

#### Potential validation tests

De-tumble, pointing control and momentum dumping could be verified by simulating those scenarios in the simulation. However, HIL simulation should probably not be considered sufficient evidence of pointing sta-

bility performance for the following reasons:

- Propagation of dynamics will rely on measurements of reaction wheel speed and acceleration whose resolution will be limited by the implementation of the tachometer or encoder.
- HIL testing will not take into account flexibility of the satellite structural elements which could play a part at that level of attitude resolution.
- It is likely that delays will be introduced into the attitude control feedback loop that create a disparity between simulated results and what would occur in microgravity,
- There is the potential for unforeseen phenomena that are not taken into account in simulation that would affect attitude control at a fine scale.

The ability to observe the influence of the spinning mirror should also be put under question. In principle, it should be possible to simulate the mirror influences listed on Page 27 and include them in the HIL testbed. This should provide a good indication of the general impact, but it might not be accurate enough to be able to determine the resulting impact on pointing stability performance.

### 3.2.6. Trade study

Tables 3.2 and 3.3 provide a shorthand visualization of the conclusions made for the microgravity simulation testbed concepts elaborated above. The concepts are evaluated for their inherent performance and their potential to be used to produce metrics of attitude control performance. Table 3.2 provides an evaluation of each concept's performance for a number of metrics. The metrics and the significance of their grades are described below:

#### Metrics

The metrics by which the concepts were evaluated are detailed below:

**Money** The predicted financial cost of purchasing hardware for the testbed. PolarCube has a tight budget, as will most educational CubeSat programs, so this is important not only for feasibility but also for realizing the goal of making the methods accessible. Labor cost was not included as PolarCube is an educational program.

**Time** The predicted number of man-hours it would take to realize and use the testbed. Reducing the time cost will always be important in every engineering endeavor and the context of a thesis project made this especially critical. Delivery times are not considered, but paperwork is.

**Torque magnitude** The predicted magnitude of external torques imparted on the test subject by the testbed. Having torques that are too strong will compromise the ability to test performance.

**Torque uncertainty** The predicted ability to determine the external torque properties of the testbed in order to produce predictions of the test subject's attitude response to known initial conditions.

**Freedom of motion** The realism of motion of the test subject in terms of how much it is constrained. The rotary air bearing was given a grade of two rather than one because pitch torques might be measurable and HIL simulation was given a grade of four rather than five because the motion is simulated.

Table 3.1: Grade criterion for the microgravity simulation testbed trade study.

| Grade | Money (\$)     | Time<br>(man-weeks) | Torque<br>magnitude (Nm) | Torque uncertainty     | Freedom of<br>motion (# of DoF) |
|-------|----------------|---------------------|--------------------------|------------------------|---------------------------------|
| 1     | >10,000        | >8                  | $>10^{-3}$               | Impossible             | 1                               |
| 2     | 1,000 to 9,999 | 8                   | $10^{-4}$ to $10^{-3}$   | Theoretically possible | 1+                              |
| 3     | 100 to 999     | 6                   | $10^{-5}$ to $10^{-4}$   | Un-proven method       | NA                              |
| 4     | 10 to 99       | 4                   | $10^{-6}$ to $10^{-5}$   | Proven method          | 3-                              |
| 5     | 0 to 9         | 2                   | $0$ to $10^{-6}$         | No torques             | 3                               |

## Grades

The systems are given a grade from one to five for each of the performance metrics. The grades have not been weighted and the total value displayed is only intended to aid in visualization. Some of the metrics have a higher bearing on feasibility for the thesis project than others (such as time and money). Some systems are given a range of possibilities for their grade on a certain metric. This is due to uncertainty in performance thanks to a lack of available information on the topic. The significance of the grades are detailed in Table 3.1:

Table 3.2: Rough evaluation of the proposed Microgravity simulation testbed concepts (rows) for a selection of performance metrics (columns).

|                                       | Money  | Time   | Torque magnitude | Torque uncertainty | Freedom of motion | Total    |
|---------------------------------------|--------|--------|------------------|--------------------|-------------------|----------|
| String suspension                     | 5      | 5      | 3 to 5           | 3                  | 1                 | 17 to 19 |
| Centered spherical air bearing        | 2      | 1 to 3 | 1 to 3           | 2                  | 5                 | 11 to 15 |
| Pendulum stable spherical air bearing | 2      | 1 to 3 | 2 to 4           | 2                  | 1                 | 8 to 12  |
| Rotary air bearing                    | 2      | 1 to 3 | 2 to 4           | 2                  | 2                 | 9 to 13  |
| Parabolic flight                      | 1      | 3      | 5                | 5                  | 5                 | 19       |
| HIL simulation                        | 3 or 4 | 2 to 4 | 5                | 5                  | 4                 | 19 to 22 |

Table 3.3 displays which attitude control performance metrics can be evaluated using each testbed concept for the PolarCube mission according to the author's estimation. Some concepts will be able to provide higher degrees of evaluation of a certain metric than others, which is not included in the table. The rationale for these evaluations is presented above for each of the testbed concepts in the sections entitled "Potential Validation Tests".

Table 3.3: Indication of whether a given performance metric (columns) can be produced using a given microgravity simulation testbed (rows). An O indicates YES and an X indicates NO.

|                                       | De-tumble | Slew | Pointing stability | Momentum dumping | Mirror influence |
|---------------------------------------|-----------|------|--------------------|------------------|------------------|
| String suspension                     | O         | O    | O                  | O                | O                |
| Centered spherical air bearing        | O         | O    | X                  | X                | O                |
| Pendulum stable spherical air bearing | O         | O    | X                  | X                | O                |
| Rotary air bearing                    | O         | O    | X                  | X                | O                |
| Parabolic flight                      | X         | X    | X                  | X                | X                |
| HIL simulation                        | O         | O    | X                  | O                | O                |

## Conclusion

String suspension and HIL testing were selected as viable methods to be used to qualify PolarCube's attitude control system. They were selected because they were the least strenuous to the mission budget, they have a history of success in previous missions [22] [19] and they have the simple and well understood properties; reducing risk of complications further down the line. String suspension testing was selected as the topic of this thesis because the fact that the system would actually be moving was considered to inherently provide a higher degree of confidence in the results and the development necessary for it was more in line with the development goals of the overall mission at the time. (1.ACSV.T.9) String suspension was also considered to be a more interesting and unexplored topic of academic study and the results of the research would be of more use to other CubeSat programs. **The goal of the research in terms of contributing to knowledge on**

**the implementation of string suspension testbeds would thus be to improve and expand the "fit-predict-fit" methods used for the MicroMAS mission by producing improved characterization of external torque properties and improving the methods by which predictions and fits are made.** The above proposals of using string suspension to test pointing stability and momentum dumping at the string's equilibrium position could also be considered contributions as they were not found in current literature and were conceived by the author, but the thesis work does not take them further than that.

### **3.3. Testbed dynamics model**

A software model that accurately describes the dynamics of the test subject's motion on the testbed is a key element of using "fit-predict-fit" methods. The dynamics model will be used to determine the external torque properties of the string suspension testbed by adjusting the model's parameters such that it produces the same response as measured uncontrolled motion of the test subject (see oscillation test on Page 24). The fitted dynamics model will then be used to predict the attitude response of the test subject under attitude control. This prediction will serve as a control to be compared to a measured attitude response of the test subject under the conditions used to make the prediction. (1.ACSV1.10) Direct comparison of error between the measured and predicted responses is not very indicative of the difference in inherent dynamic properties of the real and expected system behavior. Properties can be compared by fitting a testbed dynamics model to the measured response under attitude control. The difference between the parameters defining the model used to make the prediction and those defining the model that best fits the measured response serves as the metric of performance. More specifically, it serves as a metric of how closely the system behaves to how it is expected to.

The MicroMAS test campaign found some success in the use of one-dimensional Simulink models to produce the fits and predictions [22]. This is logical in principle because motion is constrained to rotation about a single axis. However, it limits tests to being conducted exclusively about primary body axes where one of the actuators is parallel to the axis of rotation and the other two are perpendicular. It also generally limits the complexity of the system that can be simulated. The author considered it would be wiser to use a script to simulate the attitude control system's behavior in all three dimensions, and then simulate a constraint of motion to rotation about a defined rotation axis in the body frame. A number of advantages were identified:

- The attitude control system could be tested in rotation about oblique axes. That is axes that are not parallel to a reaction wheel.
- The control response of all three reaction wheels working simultaneously could be simulated.
- The software control architecture in the software model would be truer to the embedded software in the attitude control system.
- It would be easier to implement and modify complicated functions.
- Rather than changing the test subject's properties (MMOI, asymmetrical behaviors) to meet the axis of rotation being tested, the axis of rotation would be changed and the simulation would generate the properties automatically.

Among these, the most interesting from a point of view of increasing potential for measuring attitude control performance are the first two. Together, they provide opportunities to measure how well synchronized the reaction wheels are in actuation. The simplest test would be to suspend the test subject such that its rotation is parallel to one of the reaction wheels and perform a slew about the rotation axis with all three reaction wheels actuating as if rotation was completely unconstrained. Acceleration of the perpendicular reaction wheels would not have an effect on motion, but their response could be measured and compared to a simulation. A slew test could also be performed where the test subject is suspended about an oblique axis. In which case, all reaction wheels would have an influence on motion.

The author also considered that it would be valuable to simulate expected sources of error that would be produced by the system in testing. The most important of these would be errors in attitude determination because it is the only source of measurement of the test subject's motion.

### **3.4. Satellite engineering model for testing**

In order to perform and test attitude control, an attitude control system must be supported by a number of other hardware systems. Most of these systems have to be rigidly attached to the attitude control system and small and light enough to allow realistic motion of the system when under microgravity simulation.

An engineering design model of the ALL-STAR bus, dubbed the "test model" would need to be designed and built to provide the minimum functionality required to permit validation testing. The test model would also need to be self-contained, without any wires leading out to external objects so as not to corrupt microgravity. (1.ACSV.13) This in itself proved to be a major undertaking that involved much of the PolarCube team. This section explains the requirements that the test model had to meet in order for the attitude control system to be able to perform validation tests and the methods that were chosen to meet them.

### 3.4.1. Structure

ACS is not structurally self-contained. It was designed to integrate into the the ALL-STAR bus structure (see Figure 3.13). Structure would also be necessary for any supporting subsystem hardware. The obvious decision in this case was to use the existing bus structural panels. They are convenient and they provide higher fidelity to flight conditions. Using hardware that will eventually end up on the satellite is highly desirable. Not only does it provide higher credibility to the test, overall effort is saved by not working on a deviant project and it helps overall mission development by producing shorter term deadlines for the systems involved. This also applies to the rest of the flight-like systems on the test model. Jannine Vela and Nathaniel Voth were responsible for making sure that the PolarCube structure would be ready for validation testing.

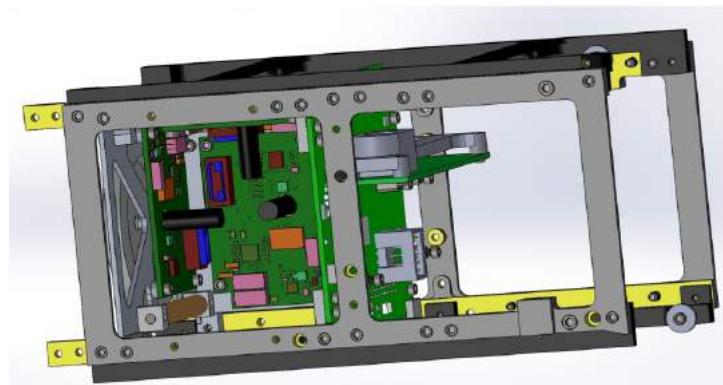


Figure 3.13: CAD rendering of ACS mounted within the PEZ bus structure.

### 3.4.2. Power

ACS requires a power supply. The supply must not require any lines running to external equipment not suspended by the string. The power supply should also be small, and provide the correct quality of power. Two options existed to provide this capability:

- Use PolarCube's Electrical Power Subsystem (EPS).
- Create an analog to EPS (batteries + regulation + power switching) that provides the minimum functionality for attitude control validation testing.

EPS would require a significant amount of "bring-up" before being capable of supplying ACS off of batteries. Creating an EPS substitute would demand significantly less effort because the system would be less complex and easier to de-bug and modify. The choice was made to use EPS because it benefits the overall mission by not diverting effort. Andrew Jones and Will Sear were responsible for making EPS able to power the test model, with help from Adam St Amand, Franklin Hinckley, Connor Kelleher, Umang Patel, Russell Gleason and the author in final integration.

This still left an issue of connectivity of the power lines to ACS. Again, two options were identified to solve this:

- PolarCube's Command and Data Handling subsystem (CDH) could be used to connect EPS and ACS as it would in flight.
- A "dummy" backplane PCB could be produced with the same form factor as CDH that had direct power routing from EPS to ACS.

The choice was made to use CDH because the risk of it failing to provide power connectivity was very low and it provided the potential for much more functionality with some development. The dummy backplane was considered a backup option in case CDH failed for some reason. Use of the dummy backplane is discussed in meeting other test model requirements to fully elaborate on what its use would entail.

### **3.4.3. Data handling**

ACS must have a means of storing data that it generates. In order to compare the response of the satellite to a predicted response from a software dynamics model, the test engineer requires a complete record of the test model's state vector for the entire period being tested. The minimum data requirement for validation of attitude control would be a record of the attitude response of the test model under known conditions. Having the reaction wheel responses would provide additional validation and deeper insight into the behavior of the testbed. Of course, having data on as many things as possible is desirable because it is invaluable for debugging, produces higher confidence in results and is often necessary for system bring-up and development regardless.

The attitude data and reaction wheel speed data are generated on the ACS processor. Architecture would need to be developed to record that data locally and make it possible to retrieve it after the test. (1.ACSV1.14) Three options of providing this functionality were considered:

- Have a live wireless data stream to an external computer where the data is stored.
- Store the data locally and wirelessly stream the data file to external computer after the test is complete
- Store the data locally on a removable SD card and remove the card to retrieve the data.

Among these, streaming the data wirelessly was deemed too much of a liability because the data stream rate would throttle the amount of data that could be collected and it would be easy for data to be corrupted by failures in the stream. The choice was made to use a removable SD card. The downside of this choice is that the test model will have to be handled between tests, which could have an effect on the rigging of the test model, compromising string torque characterization. There were two options as to how the SD card architecture could be implemented:

- Send the data to CDH and have CDH write the data to the SD cards that it will use for mass data storage in flight.
- If the dummy backplane is used, an SD card slot could be mounted on it and ACS could write to the card directly.

For the first option, communication between ACS and CDH would need to be developed as well as software to take the data from ACS and write it to the SD card. The second option would have required software for file management and SPI communication on ACS. Both significantly deviant development efforts.

The data logs would include the timestamp, sensor measurements, attitude, reaction wheel speed, torque commands and all of the configuration parameters that might be subject to change during testing. Having the configuration logged with everything else is an easy way to keep proof and reference of the test conditions when eventually conducting analysis. Adam St. Amand (CDH) and Russel Gleason (ACS) were responsible for making sure that ACS' state vector was available for download.

### **3.4.4. Attitude determination**

This project's goal is to test PolarCube's attitude control performance. Not its attitude determination performance. This distinction is important because they fundamentally exist as two separate functions. Overall attitude control performance is a compounded result of the performance of the attitude determination and attitude control systems. The performance of the attitude control system can only be truly measured if it is provided perfect attitude knowledge. This is impossible to achieve. However, it does make a case to provide the highest attitude determination performance possible regardless of how far it deviates from the flight attitude determination system. Attitude determination error can always be added after the fact to make the performance more true to what it would be in flight. Another issue to consider is that attitude determination systems that are designed to operate in orbit may not work well in a laboratory test environment. The star camera, PolarCube's nominal attitude determination instrument, would have difficulty observing stars during the day inside an engineering laboratory. Providing accurate attitude determination is arguably the greatest

challenge in developing attitude control validation testing methods and entailed most of the work that the thesis author put into the project. This section details the systems approach to attitude determination including implementation choices and the challenges faced.

### Requirements

This section details the requirements that the attitude determination used in validation testing must meet in order for attitude control performance to be measured under sufficiently close to ideal conditions.

**Scope** Attitude control validation tests will take place within a laboratory during the day, so attitude determination must be able to operate in that environment. (1.ACSVT.15) Assuming that the test model does not translate, every object or phenomenon that the test model is able to use as an attitude reference is fixed in the body-fixed Earth-reference ( $E$ ) frame (For frame definitions, see page 6). It can thus be stated that attitude shall be determined with respect to the  $E$  frame. (1.ACSVT.16) The  $E$  frame rotates approximately once per day with respect to inertia ( $7.26e^{-5}$  rad/s). For the purposes of attitude control validation tests, rotation of the  $E$  frame can be deemed negligible and it can be treated as an inertial frame.

Since the test model will only be rotating about the local gravity vector, performance of measured rotation about the rotation axis will have the greatest impact on the results of the test. (1.ACSVT.17) Measuring rotation about the rotation axis will require the orientation of the rotation axis in the  $\beta$  frame to be determined.

**Access** The attitude control system must have access to the output of the attitude determination system. This is obvious, but it becomes a major challenge for some of the concepts discussed. (1.ACSVT.19)

**Accuracy** Attitude determination must be accurate. Accuracy in validation testing is more relative than it would be in flight. In flight, attitude determination accuracy would be determined by the difference between true and measured attitude with respect to the celestial frame ( $C$ ). This is necessary because it affects how well the system would point to an external object with a known relative position with respect to the satellite (e.g. the Sun). In an attitude control validation slew test, provided that there is a continuous mapping of true attitude to measured attitude, the test engineer can simply select an attitude measurement to close on. It does not matter whether or not the target measurement reflects anything in external space. All that matters is that the attitude control system performs a maneuver specified by a commanded frame rotation. For this reason, attitude error bias is irrelevant for testing purposes.

What is important is that the measured frame rotation when performing the maneuver is the same as the rotation that the test model in fact performed in the inertial ( $\eta$ ) frame. Figure 3.14 demonstrates what this would look like if the test model measured greater changes in attitude than true motion. Note that the starting attitude is arbitrary. If the test model started at  $\theta_0$  and was commanded to perform a slew of  $\Delta\theta$ , it would rotate until  $\Delta\theta_{mes}$  was equal to  $\Delta\theta$ , but it will actually have rotated by  $\Delta\theta_{tru}$ . Attitude determination accuracy can thus be measured by the difference between true and measured rotations of the test model.

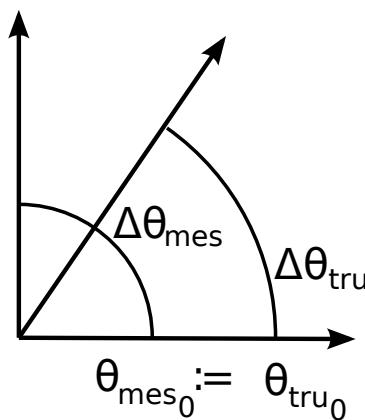


Figure 3.14: Diagram explaining the relativity of attitude inaccuracy when testing attitude control performance in a single dimension.

Figures 3.15 and 3.16 show the PID response from Figure 2.7 (Page 13) with true attitude determination and with distorted attitude determination (and no external torques). These responses demonstrate how attitude

inaccuracy affects results when attempting to test a slew maneuver. In both cases, attitude determination will show that the test model converges at the target attitude, but the response curves will look different. In Figure 3.16, the blue line can be seen as a prediction from a software model and the red line as a measured response that does not match it due to inaccuracy in attitude determination.

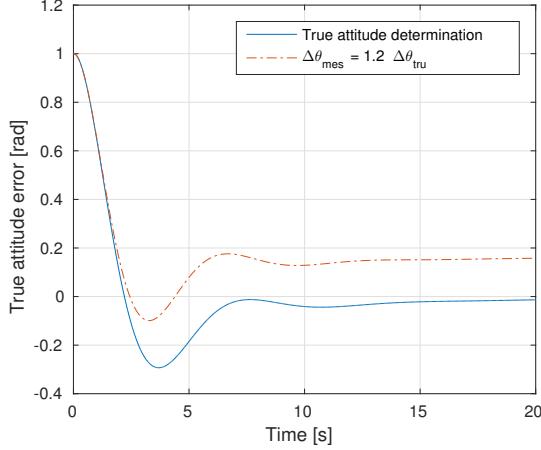


Figure 3.15: True response of the one dimensional system described in Section 2.3.1 (Page 11) performing PID control.

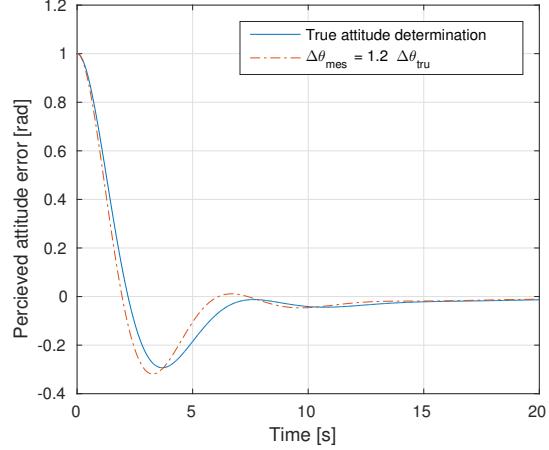


Figure 3.16: The same responses as in Figure 3.15 as they were perceived by the test model.

The difference between true and measured rotations should be below an acceptance threshold determined through analysis of mission requirements and how they flow down to validation test performance. (1.ACSV.T.18) Different thresholds would result for different degrees of validation/verification. This project set out to make attitude determination as accurate as possible given the constrained circumstances.

**Noise** Attitude determination noise can have two sources:

- A disturbance of the sensor reading by some other system on the test model.
- Some inherent property(s) of the sensor information chain.

Attitude determination noise results in jitter: small motions of the test model/satellite when attempting to maintain attitude. There are a number of methods to provide metrics for noise error, including Root-Mean-Squared (RMS) of the disturbing signal or the maximum disturbance produced. In order to determine attitude determination performance with a known degree of confidence, the metric used will have to be below an acceptance threshold.(1.ACSV.T.18) This project set out to make it as low as possible without overly compromising other performance metrics.

**Delay** A minimum of delay will always result between attitude determination and attitude control. Large enough delays will cause control instability where the attitude control system is always overcompensating. Delay will have to be below a threshold in order to produce a defined degree of confidence in attitude control performance results. (1.ACSV.T.18) This consequences of delay were not studied in detail, but it was taken into consideration when weighing concepts.

### Sensor options and potential implementations

The following lays out sensor implementations that were identified as potential attitude determination solutions within a laboratory environment along with an assessment of their feasibility.

**Rate gyroscopes** ALL-STAR's design includes MEMS rate gyroscopes. Attitude can be predicted by integrating the angular rates measured by the gyros. Gyro predictions alone are insufficient because they are subject to drift, so they must be used in conjunction with a system that directly measures the *E* frame. They can be used in a predictor-corrector architecture to greatly reduce noise in attitude determination. When used in combination with absolute attitude measurement systems, they are very effective and should be used whenever possible. As a MEMS device mounted on the same PCB as the ACS processor, they have the best possible performance in terms of access and delay. Use of the rate gyroscopes would require development of I<sup>2</sup>C communication on the ACS main board and some configuration to be able to read from the gyros.

**Accelerometer** A 3-axis MEMS accelerometer mounted on the test model would provide measurement of the local gravity field of Earth. When used in a string suspension testbed, it serves as a very accurate and reliable attitude reference along the rotation axis of the test model. Accelerometers provide no indication of the test model's motion about the rotation axis so measurement of test model's motion would rely entirely on the second attitude reference.

The accelerometer would not be at the CoG of the test model, so rotation and angular acceleration of the test model would produce some error in attitude determination. Typically, the error produced is considered negligible due to the low speeds and accelerations involved.

The original ALL-STAR design did not include an accelerometer, but there was an opportunity to include one in the next PCB revision. Having an accelerometer on board for flight could also provide verification of deployment of the ALL-STAR PEZ mechanism and mirror boom. Using it would also require working I<sup>2</sup>C and some configuration.

**Star Camera** The case for the star camera is essentially the same as the case for any other imaging instrument mounted on the test model. A number of issues arise when considering its use for attitude control validation testing:

- Some attitude reference picture would have to be created that provides easy reference for the rudimentary pattern recognition software. If the camera is pointed parallel to the rotation axis, the picture would not need to be very big, because the camera would remain pointed at the same location as the test model rotates. However, if the test model is going to be tested about a rotation axis that is not parallel to the camera, the size of the image would have to increase to cover the range of motion of the camera's imaging vector. In a situation where the camera was perpendicular to the rotation axis and full rotations would occur, the image would need to surround the test model.
- The test engineer would either need to develop new pattern recognition software for the camera or the images would need to be meticulously crafted to meet the needs of the existing software. For a star camera, this would mean reproducing a night sky.
- The camera will not be perfectly aligned with the center of mass of the test model, which means that attitude inaccuracies will be produced by the translation of the camera as the test model rotates.
- Star cameras are focused at infinity, new lenses would be necessary to perform imaging in a laboratory [8].
- Assuming that the imaging problem was solved, there is still a significant amount of delay in using a the ALL-STAR star camera. Testing for the ALL-STAR/THEIA mission showed that it took 10 seconds for the star camera to find a lock. The new star camera is expected to be faster, but is still expected have a delay in the order of seconds.

Significant effort in configuring the hardware re-design would be necessary for the star camera to be ready for use. The software for the star camera would also require major revisions before being usable.

**Imaging of the test model** A system could be devised to use pictures taken of the test model from an external camera to determine attitude [20]. This greatly relieves most of the imaging problems inherent in using an on-board camera. An external camera, if used intelligently, would provide perfect attitude determination accuracy. In a string suspension testbed, a camera would be best placed directly under or over the test model. Software would track the motion of two points that are rigid in the  $\beta$  frame and determine the rotation angle by their relative position. Another approach would be to attach a laser pointer to the test model. The camera could then track the laser's projection on an external surface, providing very high sensitivity.

Accessibility is an inherent issue with an external camera because the attitude information will have to be sent to ACS without touching the test model. This could be achieved through wireless communication, which would inevitably result in some delay. The wireless communication would require particularly good reception to make sure that the data stream is not interrupted while a maneuver is being executed. Without communication, the system could still serve as an absolute attitude reference to compare to the performance of the on-board attitude determination system. This would require some method of synchronizing data from the on-board and external camera systems.

No systems with the capabilities mentioned were directly available to the COSGC and developing one could probably stand alone as a thesis topic.

**Sun sensor** Sun sensors use photovoltaics to determine the direction of incident light. The most effective way to make use of this in a laboratory setting would be to provide an artificial light source that a sun sensor, or similar device, mounted on the test model could detect. The light source would ideally, but not necessarily be collimated. This approach was used for spherical air bearing testing of the MinXSS Sun imaging CubeSat [15].

A Sun sensor system would likely have high accuracy and low delay. Sun sensors have a limited incidence angle. The widest can detect near  $90^\circ$  from the normal axis of the sensor. This means that the test engineer would have to get creative with placement and number of sensors and light sources if the test model is going to have robust attitude determination about multiple rotation axes.

A significant amount of deviant design and implementation effort would be required to implement a sun sensor attitude determination system on the ALL-STAR test model.

**Magnetometer** The magnetometer on ALL-STAR Will not be used to directly measure attitude in flight, but it can be re-purposed to provide attitude determination on the string suspension testbed. Measuring the ambient magnetic field while not translating with respect to the Earth provides an attitude reference vector that is constant with respect to  $E$  and not parallel to the rotation axis, allowing the rotation angle about the rotation axis to be measured. A raw magnetometer reading will not only be measuring the ambient field. Electronics and ferromagnetic material in the test model will be producing magnetic fields and their influence on the reading would have to be removed somehow. Using the magnetometer on ALL-STAR would require working I<sup>2</sup>C on the ACS main board and some configuration.

### Sensor selection

The decision was made to develop an attitude determination system based on MEMS rate gyros, magnetometers and accelerometers. (1.ACSV.T.20) The choice was made primarily because it was the only feasible implementation to produce in the time available for the project. Using the gyro and acclerometer are obvious choices regardless of whatever other system is used. They are the simplest to implement and they provide the highest performance of any of the systems. Using the magnetometer pushes development in a direction that would have eventually have been necessary for full attitude control functionality and most of the development needed to be able to use the magnetometer would need to be done for the gyro and accelerometer anyways. Russel Gleason was responsible for ensuring that ACS could retrieve data from the sensors for validation testing with help from the author.

The magnetometer is robust to any orientation and independent of external equipment. The major challenges in using the magnetometer are in discerning the ambient magnetic field signal from the magnetic fields produced by other bus components. Methods to do this effectively are a major topic of this thesis.

### Magnetic influences

The following discusses the major sources of magnetic disturbance on the test model and how they affect magnetometer readings.

**Ferromagnetic material** Any metals that can be magnetized (i.e. most alloys of iron) will produce a local disturbance in the magnetic field. The disturbance that they produce is usually characterized by two patterns: hard iron and soft iron. Hard iron describes the behavior of an ideal permanent magnet: the magnet produces a field that is constant with respect to the body frame of the magnet. This means that a magnetometer that is rigid with respect to the permanent magnet will experience a bias in measurement.

Soft iron describes the behavior of annealed iron. The magnetic domains in the iron align with the ambient magnetic field and produce a magnetic field of their own depending on the magnitude and direction of the field they are subject to. The result is that soft iron produces a magnetic disturbance that is a function of attitude with respect to  $E$ .

**Electronics** Current flowing through the bus electronics produces a magnetic field that is a function of path and magnitude of the current. In the nominal case, the current in bus electronics is effectively constant and the electronics are rigid with respect to the magnetometer. This means that the magnetometer will measure a constant bias as a result of bus electronics.

**Reaction wheels** The reaction wheels use DC brushless motors. Their rotors are permanent magnets and as they rotate, the magnetic field that they produce rotates with them. From the perspective of a magnetometer that is rigid with respect to the reaction wheel motor, the magnetic disturbance appears a sinusoidal disturbance in the measured field. Each wheel produces its own sinusoidal disturbance on each of the magnetometer axis measurements, resulting in a total of 9 disturbance signals in the system.

**External sources** The magnetometer will inevitably move some while rotating on the testbed because it will not be mounted in the CoG of the test model. Variation of the ambient magnetic field throughout the space in which the magnetometer is moving will result in attitude determination errors. An idealized model of Earth's magnetic field would have the ambient field be constant in magnitude and direction throughout the space of testing. However, any nearby electronics and ferromagnetic material will produce local distortions of the field with high gradients with respect to space.

### Mitigation

The following discusses the theory behind the magnetic disturbance mitigation methods approached in this thesis.

**Parts selection** The test engineer can chose to use parts that are going to produce less magnetic disturbance. This mostly consists of using metals that do not magnetize in construction of the test model and testbed. (1.ACSV.T.24)

**Bias correction** Magnetic bias can be determined by analyzing a dataset of magnetometer data that was being collected while the test model was rotated. The dataset would represent the superposition of a constant magnetic vector in the body frame (bias), and a vector that rotates inversely to the body (ambient field). The determined bias can then be feed-forward corrected by subtracting the determined bias vector from each magnetometer measurement. (1.ACSV.T.21) If bias correction is performed successfully, any magnetic source that only produces bias (electronics and hard iron) is a minor concern for attitude determination.

**Distance** Magnetic field strength decreases with a cubed factor of the distance to a point source. This means that any additional distance that can be made between a source of disturbance and the magnetometer will significantly reduce the impact that it will have on measurements. This is true for sources within the bus structure as well as external magnetic sources.

Designing a test model that places the magnetometer far from magnetic sources, particularly the reaction wheel motors, significantly reduces the effort necessary in other mitigation techniques. In the case of this project, there was no opportunity to influence the form factor, but it should certainly be considered in any project using a magnetometer. The magnetometer on ALL-STAR is mounted on the ACS main board, which is mounted adjacent to the reaction wheel assembly with the motors very near to the ACS main board, producing an interesting worst case scenario for the problem.

The influence of external magnetic disturbance sources is best dealt with by keeping the test model as far away from them as possible. The test model should be suspended in the center of the biggest empty space available. (1.ACSV.T.23)

**Filtering attitude determination** The sinusoidal disturbance on the magnetometer readings produced by the reaction wheels can be removed with a real-time low-pass filter implementation on the magnetometer signals. (1.ACSV.T.22) This will inevitably result in some delay in attitude determination. This also produces requirements on the spin rate of the reaction wheels. A wheel speed that is too slow or does not interact well with the filter implementation will compromise attitude determination.

The gyros could then be used to further filter the determined attitude with a predictor-corrector implementation. Predictor corrector filters are a method of combining direct attitude measurement methods with attitude predictions from rate gyro data. Equation 3.3 lays out the basic predictor corrector algorithm for determination in a single dimension.

$$\theta_{\text{est}} = K_{PC}\theta_{\text{prd}} + (1 - K_{PC})\theta_{\text{mes}} \quad (3.3)$$

Where  $K_{PC}$  is the predictor-corrector gain ( $0 < K_{PC} < 1$ ),  $\theta_{\text{mes}}$  is the measured attitude (from the magnetometer and accelerometer) and  $\theta_{\text{prd}}$  is the predicted attitude derived by integrating gyro data. Predictor-corrector

filters solve the problem of noise in direct attitude measurements and drift in rate gyro propagation by combining the results of both.

### 3.4.5. Test execution

The initial conditions of every attitude control test will need to be controlled to a degree in order to ensure that the test will provide usable results and that the results are repeatable. This means that the test model must be able to control the speed of the reaction wheels at the start of the test (1.ACSVT.25) and must be able to wait for its angular velocity and/or attitude to reach the desired condition before initiating the control mode (Requirement 1.ACSVT.27). There are two options to achieving this:

- The test model autonomously tries to reach the desired initial conditions, then initiates the control mode itself when it detects that initial conditions are met.
- The test model receives a command to initiate the control mode from a test engineer that has confirmed that initial conditions are met.

The first option would involve implementing a state machine architecture in the software on ACS and would require the processor to be re-programmed every time test conditions were changed. In the state that the hardware was in, the processor could only be re-programmed by disassembling the test model and accessing the ACS main board directly. Disassembly compromises any magnetic and gyro bias correction efforts by introducing minor changes in the form of the test model, so programming capability through the main satellite ground support port on EPS would need to be developed.

The second option would require wireless communication to be established between the test model and an external computer. (1.ACSVT.33) The state of the test model would be streamed to the computer and displayed for the test engineer and the test engineer would be able to send commands to the test model through an interface on the computer that would change configuration parameters that would determine the behavior of the test model. (1.ACSVT.28) (1.ACSVT.32)

The second option was chosen because it was deemed more reliable, achievable, easy to use and because the development, especially that of the computer interface, would be more in line with the ultimate goals of the mission. (1.ACSVT.27) This section will explain the architecture of the system implemented to meet test execution requirements.

#### Reaction wheel speed control

In flight, the speed of PolarCube's reaction wheels when not performing a maneuver will be controlled through momentum dumping. This functionality cannot be replicated outside microgravity, especially if the magnetotorquers are not yet functional. Controlling reaction wheel speed while suspended would require a dedicated control law that closes the loop on the measured speed of the reaction wheels. This control law would need to be written into the software on ACS.

#### Commands

The following is the list of functions that the commands sent to the test model must serve in setting up an attitude control validation test.

- Configure attitude determination.
- Configure the control mode to be tested.
- Set the initial reaction wheel speed.
- Command the start of the test.

This section describes how the commands chosen are able to perform these functions.

**Sensor biases** The appropriate bias correction values for the sensors are determined through characterization tests of the test model. The only way to upload the determined values without disassembling the test model would be to send them up with a command. (1.ACSVT.31) Commands should exist to upload the bias values that should be subtracted from each sensor reading.

**Attitude control law gains and targets** Every control mode on PolarCube uses the same control law (Page 16), the only elements that change are the gains in the control law and the target rate and attitude. Commands must exist to upload the desired gains and targets for the coming test. (1.ACSVT.30)

**Reaction wheel speed target** Setting the initial speeds of the reaction wheel consists of sending a target to the reaction wheel speed control law. The control law will use this value as a reference to command the torque applied to the reaction wheel. (1.ACSVT.30)

**Reaction wheel control modes** In order to get the reaction wheels at the correct initial speed for a validation test, the wheels must all first be driven by the reaction wheel speed control law. A test would be initiated when the wheels switch from being commanded by the reaction wheel speed control law to being commanded by the attitude control law. A command should exist to select which control law determines the torques sent to the wheels. (1.ACSVT.29) If the test is going to be conducted about one of the test model's primary axes, torques produced by the two reaction wheels perpendicular to the test model will have no influence on the test model's motion. The test engineer might opt to keep the perpendicular reaction wheels on reaction wheel speed control to reduce the risk of attitude determination being compromised by a reaction wheel slowing down too much. This would require the ability to select the control law commanding each wheel.

#### Data stream

A stream of ACS's state vector to a display would allow the test engineer to verify that initial conditions for a test have been met. The state vector should include the following:

- Attitude
- Sensor outputs
- Reaction wheel speeds
- Commanded torque to the reaction wheels
- Every configuration variable that can be changed with a command (gains, biases, modes, targets)

The attitude, reaction wheel speeds and configuration are required to verify test initial conditions. The sensor outputs and commanded torques would be used in component level verification tests.

#### Automation of data processing

In order for testing to not waste time (which is particularly important when occupying a significant proportion of the satellite hardware), data processing should be handled through parsing scripts that are easy to implement. (1.ACSVT.34) This would also make it easier for the tests to be conducted/repeated by an engineer that is less familiar with the methodology.

### 3.4.6. Actuator verification

Below are the component level tests that the attitude actuator systems would have to pass before attitude control validation testing becomes relevant.

#### Reaction wheel torque output

Before being used to control attitude, it must first be confirmed that the reaction wheel motors can output the commanded net torque to the reaction wheels. Available motor torque and motor friction are both functions of the reaction wheel spin rate. So the system must be able to show that it can output the commanded torque at all reaction wheel speeds. This can be tested by commanding a constant torque to the reaction wheel and logging reaction wheel speed as the wheel accelerates. The speed response can then be processed to return the torque experienced by the reaction wheel as a function of reaction wheel speed. The system would be verified if this was done for several values spanning the range of the motors' available torque and the resulting torque curves were flat with respect to reaction wheel speed. Kristen Hanslik, Franklin Hinckley and Ryan Cutter were responsible for populating and de-bugging the reaction wheel driver boards. Russel Gleason and Umang Patel were responsible for establishing communication between the ACS main board and the driver boards.

#### Magnetorquer output

The first test that the magnetorquers would need to pass is to prove that the commanded current is being sent through the coils. This could be done relatively simply by connecting a laboratory ammeter in series with the magnetorquer. The second test that it would need to pass would be to show that the magnetorquers can be used in conjunction with the magnetometer. The magnetometer should be able to measure the ambient magnetic field despite the local magnetic fields produced by the (very nearby) magnetorquers. In validation testing, measuring the ambient field is not only necessary for correct implementation of the magnetorquer

but also for attitude determination. Ambient field measurement is typically done by firing the magnetorquers in pulses and taking field measurements when the magnetorquers are not firing. The magnetorquers will also have a hysteresis effect where their ferrous cores will retain some magnetization after they turn off, so a method of "de-Gaussing" must be implemented to remove this by running a small amount of current through the magnetorquer in the opposite direction after the pulse is over. Ambient field measurement could be verified by logging magnetometer readings of a stationary test model before, during and after the magnetometers are firing. Time did not allow for the magnetorquers to be integrated or tested for the thesis project.

### 3.5. Test system design overview

This section uses a number of visual aides to describe the overall design of the attitude control test system. These tools served to properly define the test system design and aided in communication when organizing the development effort of the test model, computer interface and testbed. The section contains:

- A requirements breakdown of the test system.
- A system architecture diagram of the test system hardware.
- A functional block diagram of ACS operating in the test system.
- A functional block diagram of the ACS on-board software.

#### 3.5.1. Requirements breakdown

Table 3.4 lists the test system requirements as they flow down from the thesis research objectives. They are all referred to in the previous sections of this chapter with an identifier in the following format: (X.ACSV.T.Y). Three verbs are used in requirements definition:

- **Shall** implies a requirement.
- **Will** implies a course of action that was decided upon among a number of options. They are more a statement of how a requirement will be met than a requirement. "Will" statements are only included when the course of action drives further requirements.
- **Should** implies that the requirement can be ignored under carefully weighed circumstances.

Performance metric tolerances of the testbed were left as To-Be-Determined (TBD) because they were not quantified in the project. Their values will vary depending on mission functional requirements.

Table 3.4: Requirements verification matrix for the attitude control validation testing thesis project.

| REF                      | Requirement   | Parent requirement | Reasoning  |
|--------------------------|---|--------------------|--|
| <b>Thesis objectives</b> |   |                    |  |
| 0.ACSV.T.1               | The test System shall allow the attitude control performance of ACS to be measured.                       |                    | Thesis objective.  |
| 0.ACSV.T.2               | The methods used shall be accessible to other university CubeSat programs.                                |                    | Thesis objective.  |
| <b>Requirements</b>      |   |                    |  |
| 1.ACSV.T.1               | The test system shall provide the functionality necessary for ACS to be able to perform attitude control. | 0.ACSV.T.1         | Necessary in order for attitude control to be validated. |
| 1.ACSV.T.2               | ACS shall be provided with a power supply.  | 1.ACSV.T.1         | ACS does not have an independent power supply.           |
| 1.ACSV.T.3               | ACS shall have knowledge of its attitude.   | 1.ACSV.T.1         | Necessary in order to perform attitude control.          |

Table 3.4: Requirements verification matrix for the attitude control validation testing thesis project.

| REF        | Requirement   | Parent requirement     | Reasoning  |
|------------|---|------------------------|--|
| 1.ACSVT.4  | Disturbing torques on the test model shall not exceed TBD Nm.   | 1.ACSVT.1              | Necessary for the attitude control output torques to dominate the attitude response.   |
| 1.ACSVT.5  | The test model shall experience no static friction in the direction of motion when on the testbed.  | 1.ACSVT.1              | A torque threshold for motion trivializes pointing control.  |
| 1.ACSVT.6  | ACS data shall be generated and analyzed in such a way as that metrics of ACS' ability to meet attitude control functional requirements can be derived.     | 0.ACSVT.1              | Basic concept of testing.  |
| 1.ACSVT.7  | The test procedures and test method theory shall be transparently documented.   | 0.ACSVT.2              | Lowers test engineer experience threshold.   |
| 1.ACSVT.8  | Testing equipment shall be inexpensive and consist of easily available resources.   | 0.ACSVT.2              | Lowers project budget and schedule load.   |
| 1.ACSVT.9  | ACS will be suspended by a string.  | 1.ACSVT.4<br>1.ACSVT.5 | <ul style="list-style-type: none"> <li>• Meets basic microgravity requirements</li> <li>• Inexpensive</li> <li>• Easy to build</li> <li>• Has the best potential to measure pointing stability performance</li> <li>• Academically interesting</li> <li>• Provides opportunity to implement fit-predict-fit method.</li> </ul> |
| 1.ACSVT.10 | Motion of the test model and reaction wheel rates will be measured and compared to predictions from a computer simulation of the test model on the testbed. | 1.ACSVT.6              | Fit-predict-fit method of determining attitude control performance.  |
| 1.ACSVT.11 | Disturbing torques on the test model shall be known to within TBD Nm.   | 1.ACSVT.10             | Determines the confidence in predictions.  |
| 1.ACSVT.12 | The string shall be monofilament.   | 1.ACSVT.11             | Most consistent elastic properties.  |
| 1.ACSVT.13 | ACS shall not be mechanically attached (wires) to anything that is not suspended by the string.   | 1.ACSVT.11             | Attachment makes torques unpredictable.  |
| 1.ACSVT.14 | Attitude and reaction wheel speeds from each test shall be recorded and available for analysis.   | 1.ACSVT.10             | Attitude and reaction wheel speed are the output of attitude control. The responses shall be compared to predictions.  |

Table 3.4: Requirements verification matrix for the attitude control validation testing thesis project.

| REF        | Requirement   | Parent requirement  | Reasoning   |
|------------|---|---|---|
| 1.ACSVT.15 | Attitude determination shall function indoors on Earth's surface.   | 1.ACSVT.3   | Tests will be conducted indoors on Earth's surface.   |
| 1.ACSVT.16 | Attitude shall be measured with respect to the <i>E</i> frame.  | 1.ACSVT.3   | <i>E</i> frame is the only detectable frame and it is inertial for all intents and purposes of testing.   |
| 1.ACSVT.17 | Attitude determination shall measure rotation about the local vertical.   | 1.ACSVT.3<br>1.ACSVT.9  | ACS will only move about the local vertical.  |
| 1.ACSVT.18 | Attitude determination noise, delay and inaccuracy (as defined on page 39) shall be below TBD thresholds.                                       | 1.ACSVT.3   | Determines the confidence in performance determination.   |
| 1.ACSVT.19 | ACS shall have access to attitude determination.  | 1.ACSVT.1   | Attitude control relies on attitude determination.  |
| 1.ACSVT.20 | Attitude will be determined by ACS using the MEMS magnetometer, accelerometer and rate gyroscope mounted on the ACS main board.                 | 1.ACSVT.3<br>1.ACSVT.15<br>1.ACSVT.16<br>1.ACSVT.17<br>1.ACSVT.19 | <ul style="list-style-type: none"> <li>• Meets attitude determination requirements</li> <li>• Inexpensive</li> <li>• Easy to implement</li> <li>• Minor deviation from overall satellite development</li> </ul> |
| 1.ACSVT.21 | Magnetic bias produced by test model components shall be determined and removed from magnetometer measurements.                                 | 1.ACSVT.3<br>1.ACSVT.18   | Necessary in order to meet performance requirements.  |
| 1.ACSVT.22 | Low-pass filtering of the magnetometer and attitude signals shall be implemented to remove the magnetic influence of the reaction wheel motors. | 1.ACSVT.18  | Necessary in order to meet performance requirements.  |
| 1.ACSVT.23 | ACS shall be suspended far away from external magnetic field sources.   | 1.ACSVT.18  | Maximizes consistency of ambient magnetic field.  |
| 1.ACSVT.24 | Materials that magnetize should be avoided in construction of the test model.   | 1.ACSVT.18  | Soft iron effects compromise attitude determination performance.  |
| 1.ACSVT.25 | ACS shall be able to control the spin rate of the reaction wheels.  | 1.ACSVT.10  | Necessary in order to control initial test conditions. Requires a separate control law.   |
| 1.ACSVT.26 | ACS shall initiate the test control mode when initial conditions are met.   | 1.ACSVT.25  | This is how a test starts.  |

Table 3.4: Requirements verification matrix for the attitude control validation testing thesis project.

| REF        | Requirement   | Parent requirement                     | Reasoning  |
|------------|---|--|--|
| 1.ACSVT.27 | The test engineer will confirm that the initial conditions are met before commanding the start of a test. | 1.ACSVT.26                             | Better than ACS doing it autonomously because it would be extremely cumbersome.  |
| 1.ACSVT.28 | The test engineer shall be able to observe ACS' state vector.   | 1.ACSVT.27                             | Necessary in order to confirm initial conditions.  |
| 1.ACSVT.29 | The test engineer shall have the ability command the start of a test.                                     | 1.ACSVT.27                             | Necessary in order for the test to start when the test engineer confirms initial conditions.                           |
| 1.ACSVT.30 | The test engineer shall be able to set the control law gains and targets of the test model.               | 0.ACSVT.1<br>1.ACSVT.10<br>1.ACSVT.27  | This allows the test engineer to configure test conditions.  |
| 1.ACSVT.31 | The test engineer shall be able to upload sensor bias correction values to an integrated test model.      | 1.ACSVT.21                             | Can not be uploaded before integration because bias changes after integration.   |
| 1.ACSVT.32 | ACS shall be able to receive commands from the test engineer.   | 1.ACSVT.29<br>1.ACSVT.30<br>1.ACSVT.31 | Basic functionality to configure and start a test.   |
| 1.ACSVT.33 | The test model shall be able to communicate wirelessly with an interface computer.                        | 1.ACSVT.13<br>1.ACSVT.28<br>1.ACSVT.32 | The test engineer needs observation of ACS' state vector and ability to command ACS without any mechanical attachment. |
| 1.ACSVT.34 | The test engineer shall be able to analyze test data quickly and easily.                                  | 0.ACSVT.2                              | Saves valuable time when the satellite hardware is occupied, also lowers test engineer experience threshold.           |

### 3.5.2. System architecture diagram

Figure 3.17 illustrates the connectivity of all of the electronic hardware in the testbed design. Each of the block could be colored (red-yellow-green) to indicate how far along they were in development. This diagram shows that the magnetorquers were included in the test model, which is not the case for the test model that was actually realized because they were not ready for use by the time testing was conducted.

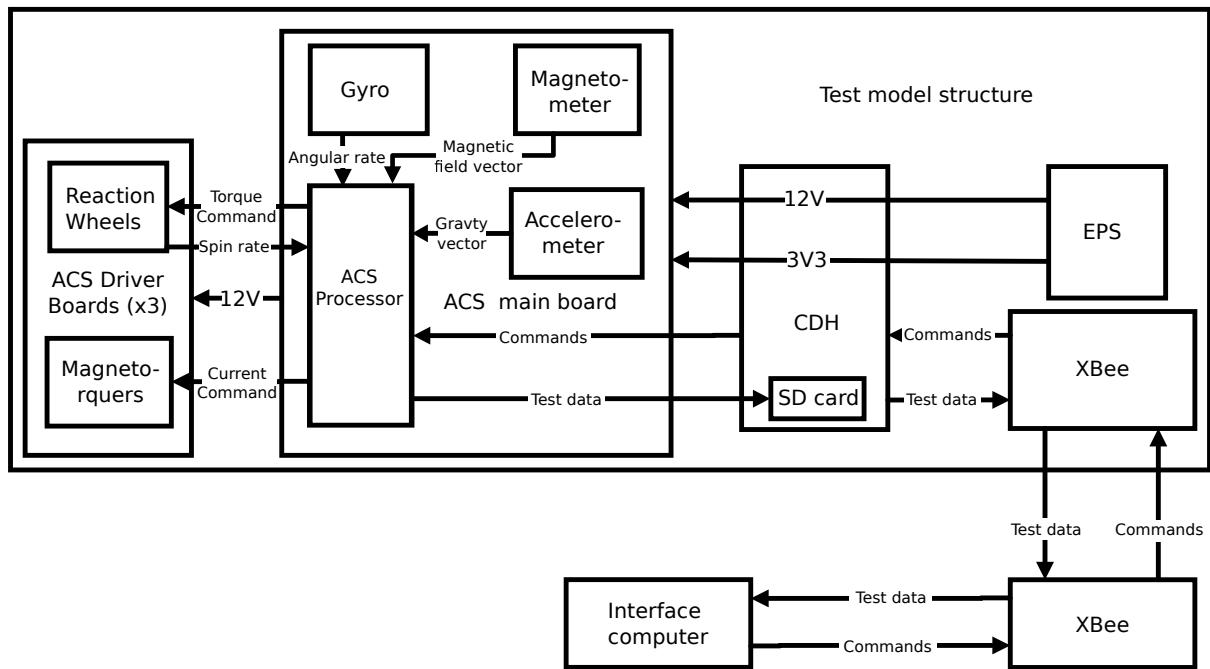


Figure 3.17: A diagram describing the interfaces between the electronic hardware elements in the testbed design.

### 3.5.3. Functional block diagram of ACS on the testbed

Figure 3.18 shows the control flow of the attitude control system and its interaction with the exterior physical environment and the test engineer. Every element in orange is a deviation from the how the system will operate in flight. This could either be the direct result of a change in environment or implemented deliberately in order for the system to be testable in the new environment.

Figure 3.18 can be compared to Figure 2.14 on Page 18 to observe the degree to which the system had to be altered. The dotted line indicates the elements of the overall system that the tests intend to measure. Note how that part mostly remains green while almost everything else had to change. The fact that the "Test model motion" block is orange reflects the fundamental limitations of attempting to validate attitude control performance before the satellite is launched.

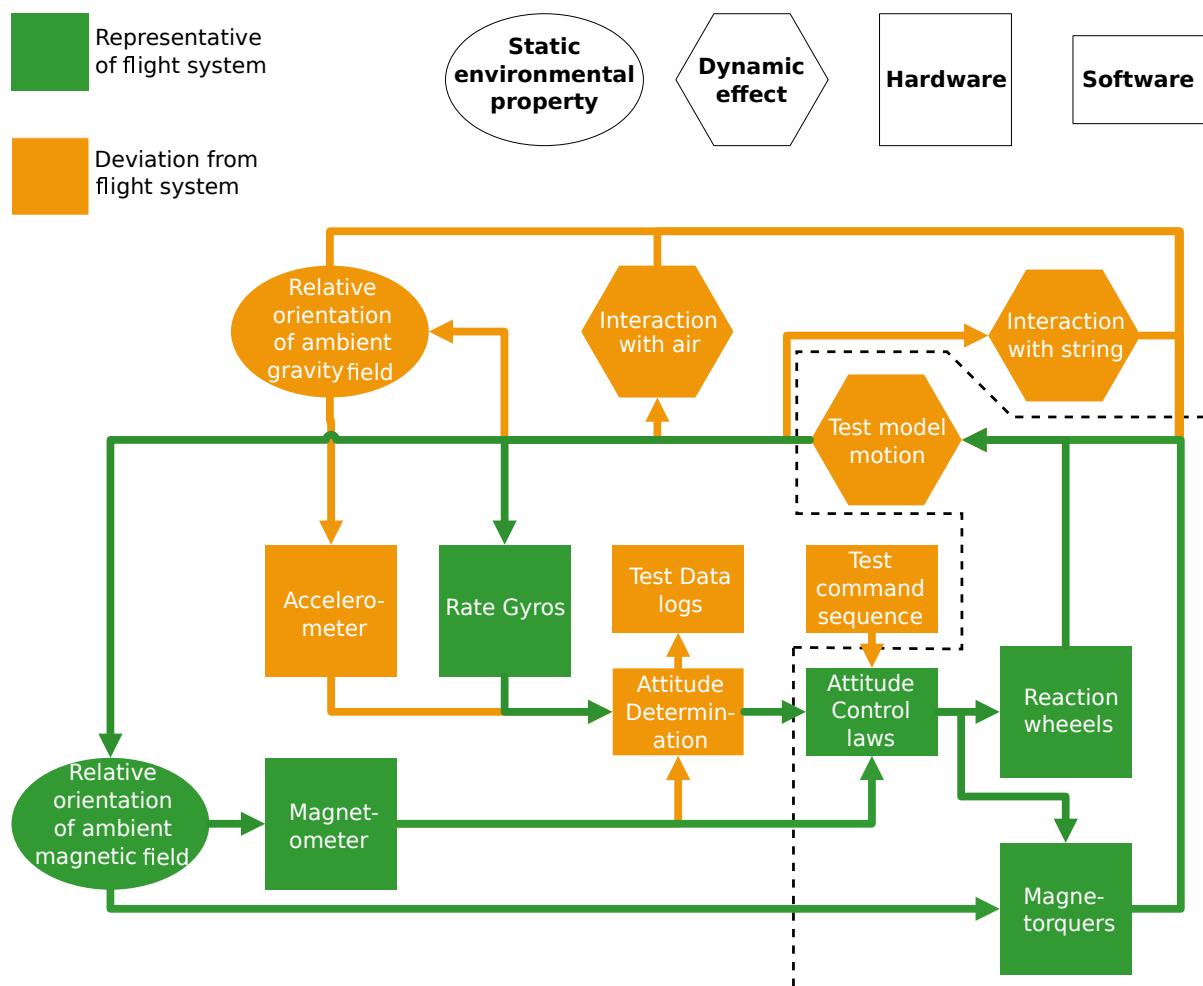


Figure 3.18: A diagram describing the interactions between functional elements of the test subject and testbed.

### 3.5.4. Software architecture diagram

Figure 3.19 is a functional block diagram of the on-board software written for ACS to be used in attitude control validation testing. It served as a blueprint for design and reference when writing the code. The on-board software for testing was the first iteration of embedded software written on ACS to close an attitude control loop between sensors and actuators. The author was responsible for designing the software architecture and writing the ADCS functions with help from Adam St. Amand, Russell Gleason and Kristen Hanslik.

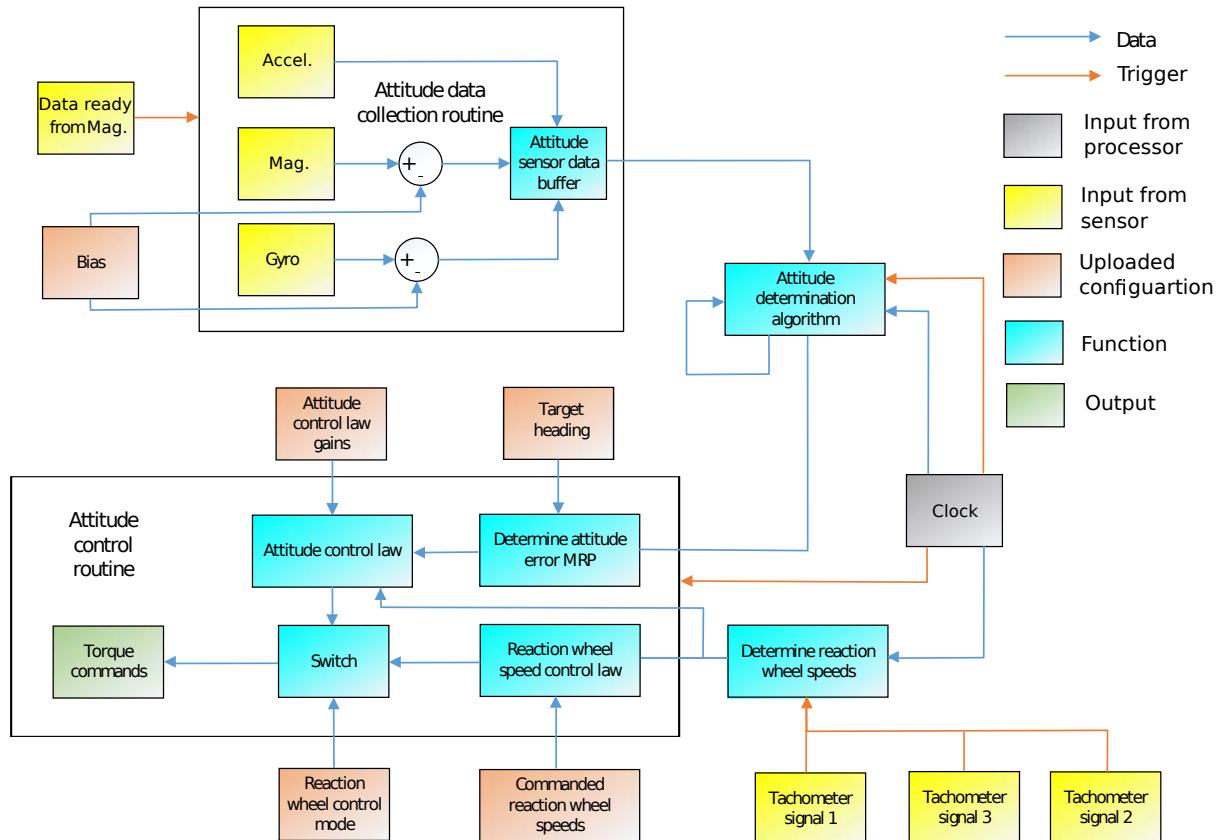


Figure 3.19: Functional block diagram of the ACS flight software written for validation testing.

## 3.6. Chapter summary

Design of the overall attitude control validation testing system flowed down from the research objectives derived from project stakeholder needs. Three top level functional requirements flow from the research objectives

1. The attitude control system shall be able to perform attitude control.
2. The performance of the system shall be measured
3. The methods developed shall be accessible to university CubeSat programs

The first issue to address when attempting to meet the first requirement is that when the attitude control system is on the ground, gravity forces it to touch external objects, introducing friction far greater than the torques that it is able to produce. Several methods exist to reduce the external torques on the attitude control system such that the torques that it produces dominate its motion. A literature study prior to the thesis project identified four methods that should be considered in detail: spherical air bearings, rotary air bearings, HIL simulation and suspension by a string. A trade study was conducted to asses the feasibility and potential of the microgravity simulation methods and resulted in the selection of a string suspension testbed. String suspension was selected because it is inexpensive, simple and is expected to have highly consistent external

torque properties that can be determined through an oscillation test. The decision was made to adopt the "fit-predict-fit" methods of determining attitude control performance metrics pioneered for the MicroMAS satellite [22] and improve determination of external torque properties in order to provide more accurate metrics of system performance.

The first step in meeting the second requirement is producing data that is representative of the attitude control system's performance. This is present in the form of the attitude and actuator output responses of the system given a set of initial conditions. The response of the system must have an expected response to be compared to. The expected response would be produced with a software model of the dynamics of the attitude control system and its interaction with the environment. The software model would need to take into account the interaction of the attitude control system with the string. Some of the properties of that interaction would initially be unknown and must be determined.

The first requirement also requires a functioning attitude control system. This would be solved by building an engineering model of the satellite for testing (test model) that provides ACS with structure, independent power and attitude determination. Meeting the second requirement requires that data produced by the test model (especially the attitude and reaction wheel speeds) be recorded and retrievable by the test engineer for eventual analysis. The test model would meet this by providing means of storing data on a micro SD card.

Several methods were considered as means of providing attitude determination to ACS that would work in a laboratory setting. The decision was made to use a combination of a MEMS gyro, a MEMS accelerometer and a MEMS magnetometer. The accelerometer was not part of ACS' design initially, but one would be included for the purposes of testing. The accelerometer and gyro are very convenient to use for testing, but the magnetometer signal would experience disturbances from the electronics in the test model, the reaction wheels, ferromagnetic material in the test model and any magnetic field sources in the vicinity of testing. Potential mitigation methods were identified for all of these effects.

The test engineer must be able to control the initial conditions of a test in order for the test to be repeatable and so that a desirable response can be measured. A desirable response would be one that can be compared to simulation output to provide metrics of the system's performance. The initial conditions are defined by the attitude of the test model, the speed of the reaction wheels, the configuration of attitude determination and the configuration of the control law. The test engineer must be able to set these without disassembling the test model. The test engineer must also be able to start the test once he has observed that the desired initial conditions are met. These functions would be met by establishing wireless communication between the test model and an interface computer. The test engineer would be able to send commands to the test model and observe its state vector through the interface computer.

A set of tests would be conducted to provide different metrics of the attitude control system's performance. Each of the tests would have procedures written for it as well as software automating the process of analysis. Producing the expected response of the system with a software model would require knowledge of the properties of the string suspension testbed in terms of external torques imparted on the test model. This can be achieved by recording the attitude of the test model as it oscillates freely on the testbed and producing a model that fits the measured attitude response. The test model would be tested under de-tumble control as well as pointing control. Tests were also devised to test pointing stability, ability of the system to dump angular momentum stored in the reaction wheels with magnetorquers and the influence of the payload mirror on performance.

A number of documents were produced to define the test system design. These included a formal requirements definition, a hardware system architecture, a functional block diagram of ACS performing attitude control on the testbed and a system architecture of the ACS on-board software for testing.



# 4

## Materials and methods of test system implementation

This chapter explains the materials and methods used to produce a test system (testbed, support systems and methods) for validation testing of PolarCube's attitude control performance. The test system was based on the design outlined in Chapter 3. It was designed to be used for fit-predict-fit testing where a dynamics model is used to make predictions of the test model's motion when executing its control modes on the testbed. The goal when producing the test system was to maximize the potential efficacy of fit-predict-fit methods by producing the best possible performance in terms of attitude determination and characterization of external torques produced by the string suspension testbed. The chapter includes:

- An explanation of the testbed dynamics model in MATLAB for producing fits and predictions of test model motion.
- The methods used to configure, perform and test attitude determination using the MEMS sensors, including attitude determination configuration and testing conducted on a preliminary Arduino-based test model.
- The methods used to ensure that attitude control would function on the string suspension testbed.
- The methods used to provide the communication capabilities necessary for attitude control validation testing
- The methods used to produce a string suspension testbed that met attitude control validation requirements.
- Results of an oscillation test conducted with the Arduino test model.
- The techniques used to formalize the testing process

### 4.1. String testbed dynamics model

This section explains the methods used in creating a software model of the dynamics of the test model's motion on the string suspension testbed in MATLAB. The model exists as a script that propagates a dynamic response of the system based on a specified configuration and initial conditions for the amount of time specified. Much of the rigid body dynamics in the simulation (everything aside from any interaction with the string) were adopted from PolarCube's existing satellite dynamics software model written by Carlos Pulido. The source code for the model is attached in Appendix B.

#### 4.1.1. Rigid body dynamics

The model uses MRPs to define the attitude and propagate motion of the test model. The rate change of an MRP can be determined from the angular velocity with Equation 4.1 [25].

$$\dot{\sigma} = \frac{1}{4}[(1 - |\sigma|^2)[I_{3x3}] + 2[\tilde{\sigma}] + 2\sigma\sigma^T]^\beta\omega \quad (4.1)$$

$\omega$  is determined by numerically integrating  $\dot{\omega}$  determined from a version of Equation 2.24 (Page 13) that has been adjusted to constrain motion about the rotation axis. The dynamics model essentially assumes that all torques and rotation that are not parallel to the rotation axis are negated:

$$\dot{\omega} = \hat{q}(\hat{q} \cdot ([I]^{-1}(\hat{q}(\hat{q} \cdot \tau) - \omega \times [I]\omega - \omega \times h_s))) \quad (4.2)$$

Where  $\hat{q}$  is the rotation axis defined in the body frame as an initial condition in the script. The dynamics model uses the orientation of the test model when it is at the equilibrium position of the string as the reference attitude. Equation 4.3 relates attitude in MRP to deflection angle ( $\psi_d$ ). (See Equation 2.16 on Page 10 for reference.)

$$\sigma = \tan\left(\frac{\psi_d}{4}\right)\hat{q} \quad (4.3)$$

$$\psi_d = \psi - \psi_{eq} \quad (4.4)$$

$\psi_{eq}$  is the heading of the test model when it is at the equilibrium position of the string.  $\psi_{eq}$  and  $\psi_{d0}$  are defined as initial conditions in the script.  $\psi_{d0}$  is used to define  $\sigma_0$  using equation 4.3. From that point onward, Equations 4.1 and 4.2 are used to propagate  $\sigma$  and  $\psi_d$  is determined by inverting equation 4.3.

$\tau$ , the sum of torques on the test model body, is defined in Equation 4.5.

$$\tau = \tau_{ext}(\psi_d, \omega) - u_s(\sigma_e, \omega_{mes}, \omega_r, K, [P], [K_l], z, [I], \Omega, I_{ws}) \quad (4.5)$$

Attitude error ( $\sigma_e$ ) is determined from the determined attitude ( $\sigma_{mes}$ ) and the attitude target ( $\sigma_r$ ) using Equation 2.18.  $\omega_{mes}$  is the angular rate determined with the gyros.  $\Omega$  is determined by integrating  $u_s$ :

$$\Omega = \Omega_0 + \int_0^t u_s dt \quad (4.6)$$

$\Omega_0$  is an initial condition of the simulation.

#### 4.1.2. Simulating attitude determination

The software model includes a simulation of the test model performing attitude determination. This is a process of simulating sensor input signals based on the true attitude and signal disturbance properties of the test model, then passing the sensor inputs through the same attitude determination algorithm that will be used on the test model. This served as a platform to design and verify the attitude determination algorithm. Most importantly, it served to tune the filtering methods that were used to remove the magnetic disturbance from the reaction wheels. This section explains how this was all achieved.

##### Accelerometer signal

The direction of the ambient gravity field is defined in the initial conditions of the simulation:

$$g \equiv -g\hat{q} \quad (4.7)$$

When the accelerometer is stationary, it will measure the ambient gravity field as well as any errors produced by the sensor (bias, noise, misalignment etc.). Error in measuring the gravity field will also be produced by the rotation and angular acceleration of the test model because the accelerometer will not be located on the rotation axis. Measurement of the gravity field was assumed to be ideal in the simulation because other sources of attitude error (magnetometer) would have a far more significant impact on attitude determination performance. No accelerometer reading was simulated and  $\hat{q}$  was used as the gravity reference instead.

##### Magnetometer signal

Producing an ambient magnetic field vector in the  $\beta$  frame requires  $[R_{\beta E}]$ : the full frame transformation from the  $E$  frame to the  $\beta$  frame expressed in DCM ( $E$  and  $\beta$  frame definitions on Page 6). This transformation is defined in the simulation input by  $\hat{q}$  and  $\psi$ .  $[R_{\beta E}]$  was determined by reverse engineering the attitude determination method described in Section 4.2.4 (Page 59).  $\theta$  and  $\phi$  were determined from  $\hat{q}^\beta$  and using a given  $\psi$ ,  $[R_{E\beta}]$  was determined with Equation 2.13 (Page 9).  $[R_{E\beta}]$  was then inverted to be able to project the ambient magnetic field vector onto the body frame.

A true magnetometer signal would then be disturbed by all of the sources listed in Section 3.4.4 (Page 42); namely bias, distortion, changes in the ambient field and reaction wheel disturbance. The simulation assumes that the first three sources of disturbance have been removed by effective bias correction, material selection and selection of the testbed location. The reaction wheel disturbance on a magnetometer signal is simulated

as a sinusoidal function of the reaction wheel's angular position. The following formulas describe the total signal simulated:

$$\mathbf{B}_{\text{mes}}^{\beta} = (R_{\beta E} \mathbf{B}_{\text{amb}}^E) + \mathbf{B}_{\text{RW}} \quad (4.8)$$

$$B_{\text{RW}i} = A_i^X \sin(\Theta_X) + A_i^Y \sin(\Theta_Y) + A_i^Z \sin(\Theta_Z) \quad (4.9)$$

$$\Theta_i = \Theta_{i0} + \int_0^t \Omega_i dt \quad (4.10)$$

Where  $\Theta_i$  is the angular position of the reaction wheel rotating about  $\hat{\mathbf{b}}_i$  and  $A_i^j$  is the amplitude of the sinusoidal magnetic disturbance of reaction wheel  $j$  on magnetometer measurement axis  $i$ .  $A_i^j$  can be measured for all  $i$  by logging magnetometer data while only reaction wheel  $j$  is spinning.  $\Theta_i$  is numerically integrated from the previous value at every simulation loop. If  $\Theta_i$  is simulated ideally,  $\mathbf{B}_{\text{RW}}$  will have overly consistent signal resonance patterns, so  $\Theta_{i0}$  is selected randomly and a small amount of uncertainty is added to  $dt$  when performing integration, producing a signal that has more realistic variation in resonance over time.

### Gyro signal

The simulated gyro signal is generated from the angular velocity  $\omega$  of the test model. The gyro readings will have noise, bias and sensitivity error produced by the sensor. Adding these properties, especially bias, to the signal is important because drift in gyro measurement integration has to be represented in the simulation to demonstrate the effectiveness of the predictor-corrector filter. The simulation generates gyro sensor readings with the following formula:

$$\omega_{\text{mes}} = \omega(1 + S_e) + \omega_{\text{no}} + \omega_{\text{bs}} \quad (4.11)$$

Where  $S_e$  is the gyro's normalized sensitivity error,  $\omega_{\text{no}}$  is the measured angular velocity contributed by signal noise and  $\omega_{\text{bs}}$  is the angular velocity contributed by signal bias.  $\omega_{\text{no}}$  is randomly sampled from a normal distribution.  $\omega_{\text{bs}}$  and the standard deviation of  $\omega_{\text{no}}$  of an actual sensor can be measured by logging gyro output data while the test model is stationary. Measurement of  $S_e$  would be far more difficult.

### Timing

The simulation performs attitude determination and attitude control at realistic time intervals so that attitude control performance can be traded off against demands on the ACS processor.

## 4.2. Attitude determination

This section describes the design process and final implementation of the attitude determination used for attitude control validation testing.

### 4.2.1. Accelerometer

The ADXL345 3-axis accelerometer was added to the first revision of the ACS main board that followed the start of the thesis project. Its integration into the system was fairly straightforward as it only required power, connection to the I<sup>2</sup>C bus lines and space on the PCB. Franklin Hinckley was responsible for ACS PCB design. (see Figure 4.1)

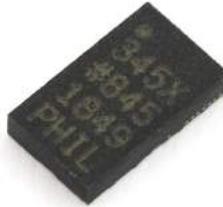


Figure 4.1: Image of the ADXL345 MEMG accelerometer.

#### 4.2.2. Arduino test model

ACS was in the process of integration for most of the period of the thesis project, so the attitude determination method used in validation testing was first implemented and tested on an Arduino-based test model. The model uses an Arduino Fio v3 from Sparkfun electronics. The Fio has a minimalist form factor, is designed for wireless communication with an XBee miniature radio transceiver and is powered by a lithium polymer battery, making it perfect for this implementation. The Fio was mounted on a prototyping board and connected to breakout boards of the three MEMS sensors (accelerometer, magnetometer, gyro) that ACS uses and a micro SD card slot. The sensors were being read via I<sup>2</sup>C and the Fio was made to write data logs to the SD card through SPI. The Arduino test model was designed, built and used by the author. (see Figure 4.2)

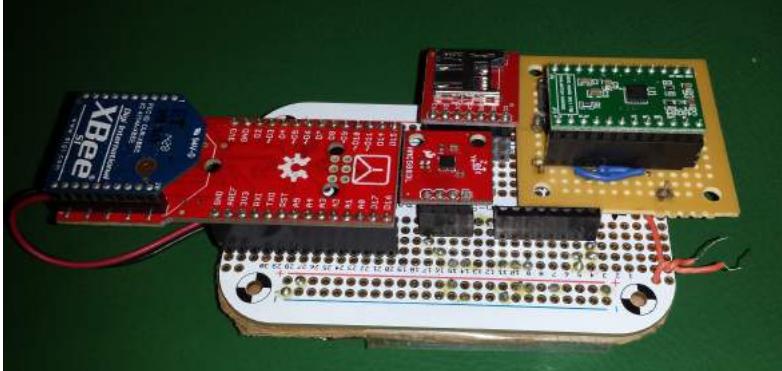


Figure 4.2: Image of the Arduino test model. The accelerometer is not mounted. Photo credit to Adam St. Amand.



Figure 4.3: Image of the XBee explorer dongle.

XBees were chosen for wireless communication in the Arduino test model and the final test model because of their convenience. They can be configured to work analogously to a UART serial port. They have limited range, but that is not a concern when testing in a laboratory. The XBee on the test model communicates with an XBee that is connected to a USB port on the interface computer using an XBee Explorer adapter made by Sparkfun Electronics (see Figure 4.3).

Having the Fio powered by a lithium polymer battery is convenient because it will recharge automatically when the Fio is plugged in to be programmed. It's also important for the attitude determination because AA and 9V batteries are typically built out of steel that is easily magnetized, which will impact magnetic measurements. The Arduino test model's simplicity gave it excellent performance in terms of magnetic disturbance. The only influence on the magnetometer measurement would come from currents in the electronics, thus a constant bias. The Arduino test model eventually served as a control sample from which the magnetometer performance of the final test model could be measured.

#### 4.2.3. Magnetic bias determination

Magnetic bias from the test model is determined by analyzing magnetometer readings taken while the test model is rotating in all directions. If bias is the only magnetic disturbance, the resulting magnetometer readings will describe a sphere that is not centered about the origin. This is the result of adding the bias (center of the sphere) to the ambient magnetic field that is rotating with respect to the body frame (spherical pattern). The magnetic bias can be determined by finding the center of the sphere described by the data points.

An algorithm was developed to find the best center from the dataset. The algorithm determines the center of the sphere described by four randomly sampled data points several hundred times and selects the result that produces the lowest standard deviation of vector magnitude after the data is corrected for bias by subtracting the center vector from all of the data points. This process has been named "sphere center calculation". The source code is available in Appendix B. Figures 4.4 and 4.6 show magnetometer data collected with the Arduino test model in an open space within the COSGC before and after bias correction using data from a sphere center calculation. Figures 4.5 and 4.7 show the vector magnitudes of the data before and after correction.

Conducting the test with the final test model will be complicated primarily by the influence of the reaction wheels. The best approach determined for this was to take the test while the wheels were spinning and to pass the three magnetometer signals through low pass filters before attempting sphere center calculation.

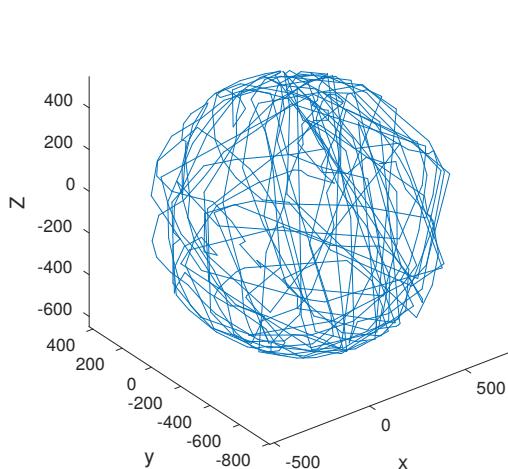


Figure 4.4: Magnetometer measurements taken while rotating the Arduino test model. Units are presented in counts.

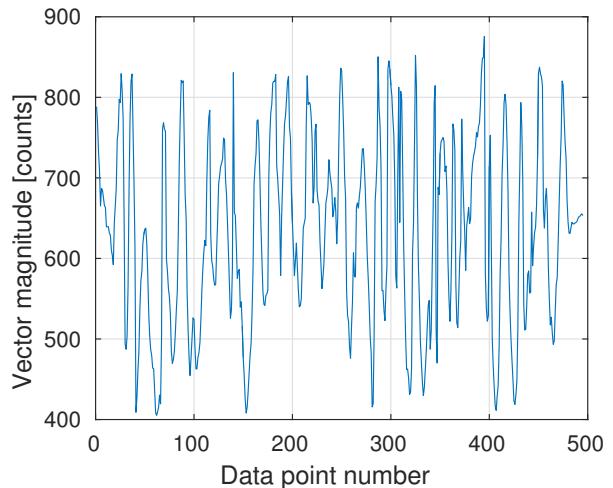


Figure 4.5: Vector magnitude of the values plotted in Figure 4.4. The signal has a standard deviation of 115.6 counts.

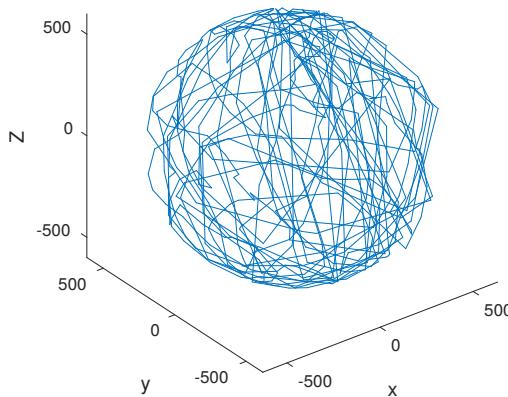


Figure 4.6: Data from Figure 4.5 after subtracting the determined center value from every data point. Units are presented in counts.

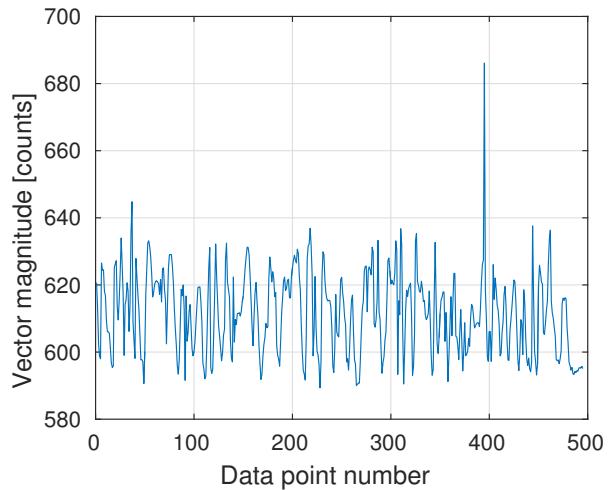


Figure 4.7: Vector magnitude of the values plotted in Figure 4.4. The signal has a standard deviation of 12 counts.

#### 4.2.4. Algorithm

An adaptation of the TRIAD method was chosen for attitude determination using the accelerometer and magnetometer. The TRIAD method is probably the simplest attitude determination algorithm for three dimensions. It accepts two reference vectors in the body frame that are not parallel, places full confidence in one of the measurements and uses the other to determine the body's rotation about the first reference vector. In this case, full confidence is placed in the accelerometer measurement of the local gravity vector because it is subject to negligible disturbances. Equations 4.12, 4.13 and 4.14 show how the TRIAD method is used to determine the  $E$  frame in terms of the  $\beta$  frame (Frame definitions on Page 6) given knowledge of the ambient magnetic and gravity field vectors.

$$\hat{\mathbf{a}}_3^\beta := -\hat{\mathbf{g}}^\beta \quad (4.12)$$

$$\hat{\mathbf{a}}_2^\beta := \hat{\mathbf{B}}^\beta \times \hat{\mathbf{g}}^\beta \quad (4.13)$$

$$\hat{\mathbf{a}}_1^\beta := \hat{\mathbf{a}}_2^\beta \times \hat{\mathbf{a}}_3^\beta \quad (4.14)$$

The attitude determination system determines  $\psi$ ,  $\theta$  and  $\phi$  as they were defined in Section 2.2.2 (Page 9) with Equations 4.15 through 4.18.

$$\psi = \cos^{-1}(\mathbf{r} \cdot \hat{\mathbf{a}}_2) \cdot \text{sgn}(a_{2x}^\beta) \quad (4.15)$$

$$\theta = -\cos^{-1}(|\mathbf{r}|) \cdot \text{sgn}(a_{3x}^\beta) \quad (4.16)$$

$$\phi = \cos^{-1}\left(\frac{r_y^\beta}{|\mathbf{r}|}\right) \cdot \text{sgn}(r_z^\beta) \quad (4.17)$$

$$\mathbf{r} = \hat{\mathbf{b}}_1 \times (-\hat{\mathbf{a}}_3) \quad (4.18)$$

$\psi$  is particularly useful in string suspension testbeds because it directly measures rotation of the test model body about the rotation axis. The software model can thus predict the  $\psi$  response of the attitude control system to be compared to the measured  $\psi$  response. Note that  $\psi$  is the only Euler angle measurement that is dependent on the magnetometer.

It is convenient to re-define the  $\beta$  frame depending on the orientation of the test model with respect to the  $E$  frame. If the standard PolarCube  $\beta$  frame (Page 7) is always used, then  $\psi$  determination becomes compromised when the test model's X axis is parallel to the rotation axis (Euler angle singularities on Page 10). This was solved by creating an intermediate body frame  $\beta^*$  and unit axes  $\hat{\mathbf{b}}_1^*$ ,  $\hat{\mathbf{b}}_2^*$  and  $\hat{\mathbf{b}}_3^*$ . All axes of  $\beta^*$  are parallel to an axis of  $\beta$ . The rules below describe how  $\beta^*$  is defined.

- $\hat{\mathbf{b}}_3^*$  is whatever axis (positive or negative) of  $\beta$  that is pointing closest to vertical.
- $\hat{\mathbf{b}}_1^*$  is  $\hat{\mathbf{b}}_1$ , unless  $\hat{\mathbf{b}}_3^*$  is defined as  $\hat{\mathbf{b}}_1$  or  $-\hat{\mathbf{b}}_1$ , in which case  $\hat{\mathbf{b}}_1^*$  is  $\hat{\mathbf{b}}_2$ .
- $\hat{\mathbf{b}}_2^* = \hat{\mathbf{b}}_3^* \times \hat{\mathbf{b}}_1^*$

$\psi$ ,  $\theta$  and  $\phi$  are then determined for  $\beta^*$  instead of for  $\beta$ . This also keeps  $\theta$  and  $\phi$  below  $\pi/4$  rad which is more intuitive for the test engineer. The frame transformation from  $\beta$  to  $\beta^*$  in DCM is defined in Equation 4.19

$$[R_{\beta^*\beta}] = \begin{bmatrix} \hat{\mathbf{b}}_1^{*\beta T} \\ \hat{\mathbf{b}}_2^{*\beta T} \\ \hat{\mathbf{b}}_3^{*\beta T} \end{bmatrix} \quad (4.19)$$

The full transformation from the  $\beta$  frame to the  $E$  frame is thus:

$$[R_{E\beta}] = [R_\psi][R_\theta][R_\phi][R_{\beta^*\beta}] \quad (4.20)$$

#### 4.2.5. Filtering

Attitude determination data goes through two stages of filtering. The first stage is a moving average filter that averages the last  $N$  number of magnetometer readings. This removes most of the magnetic disturbance produced by the reaction wheels with the consequence of introducing a delay of  $N/(2f_s)$  seconds where  $f_s$  is the sampling frequency. Simulation of implementing this filter on a reaction wheel disturbance showed that filtering will be compromised if the rotation rate of the reaction wheel becomes too low or if it approaches a harmonic of  $f_s$ . It also showed that filtering generally improves with increasing  $f_s$ , which led to efforts to maximize the  $f_s$  of the magnetometer. A higher  $f_s$  was achieved by implementing interrupt-driven sampling of the MEMS sensors using the magnetometer's "Data Ready" pin, which improved the magnetometer sampling rate from 75 Hz to 170 Hz [13].

Attitude is then determined with the TRIAD method explained in Section 4.2.4. The resulting attitude will still have significant noise in  $\psi$  due remaining magnetometer noise. The determined  $\psi$  then passes through a one-dimensional predictor-corrector filter where it is weighed against predictions made from gyro measurements. The predictor-corrector filter is implemented using Equations 4.21 through 4.23.

$$\psi_{\text{est}}(t) = K_{\text{PC}}\psi_{\text{prd}} + (1 - K_{\text{PC}}) \cdot \psi_{\text{mes}} \quad (4.21)$$

$$\psi_{\text{prd}} = \psi_{\text{est}}(t - dt) + dt \cdot \dot{\psi} \quad (4.22)$$

$$\dot{\psi} = \omega_{\text{mes}} \cdot (-\hat{\mathbf{g}}_{\text{mes}}) \quad (4.23)$$

Where  $\omega$  and  $\hat{\mathbf{g}}$  are measured with the gyro and accelerometer respectively.  $dt$  in this case is the time elapsed between attitude determination events.

The most well known predictor-corrector filter is the Kalman filter. Kalman filters use probabilistic mathematics to optimize  $K_{\text{PC}}$  for best performance. Implementing a Kalman filter was deemed unnecessarily complex for this project and  $K_{\text{PC}}$  was instead selected manually based on observed performance in the testbed dynamics model. A value of 0.98 for  $K_{\text{PC}}$  was used in testing.

#### 4.2.6. Attitude determination test

The attitude determination test measures  $\psi$  determination accuracy by recording the measured  $\psi$  at a set of known angles about the vertical. Figure 4.8 is a picture of the test rig designed to allow this. It consists of a turntable with a protractor for angular reference. The procedure is summarized as follows:

1. Set up the attitude determination test rig such that it is horizontal.
2. Mount the test model on the attitude determination test rig.
3. Align the test rig turntable with the zero point on the protractor.
4. Record heading.
5. Rotate the turntable by  $\psi_{\text{int}}$  degrees.
6. Repeat from step 4 until a full rotation has been conducted.



Figure 4.8: Image of the test model on the attitude determination test rig.

The test does not indicate true magnetic North. Instead, the first value measured is assumed to be true. True heading is extrapolated from that point. The source code for parsing attitude determination test data is available in Appendix B. Figures 4.9 and 4.10 show the results of two early attitude determination tests with the Arduino test model. Figure 4.9 plots the measured heading values ( $\psi_{\text{mes}}$ ) against true heading ( $\psi_{\text{tru}}$ ), where true heading is defined by Equation 4.24.

$$\psi_{\text{tru}} = \psi_{\text{mes}_0} + (N - 1)\psi_{\text{int}} \quad (4.24)$$

Where  $\psi_{\text{mes}}$  is the heading measured at measurement number  $N$  and  $\psi_{\text{mes}_0}$  is the  $\psi_{\text{mes}}$  from the first measurement ( $N = 1$ ). Figure 4.10 plots heading determination error ( $\psi_e$ ) as a function of true heading ( $\psi_{\text{tru}}$ ) where heading determination error is defined by Equation 4.25.

$$\psi_e = \psi_{\text{mes}} - \psi_{\text{tru}} \quad (4.25)$$

It is important to note that the heading measurement error in these tests is not absolute but relative. One of the data points has an error of 0 and that is because it was the first measurement taken.

Both tests in Figures 4.9 and 4.10 were taken at the same location within an hour of each other with no change in configuration. They show that results can be repeated under those conditions. However, results would change significantly if the test location was moved between tests, even if both tests were taken relatively

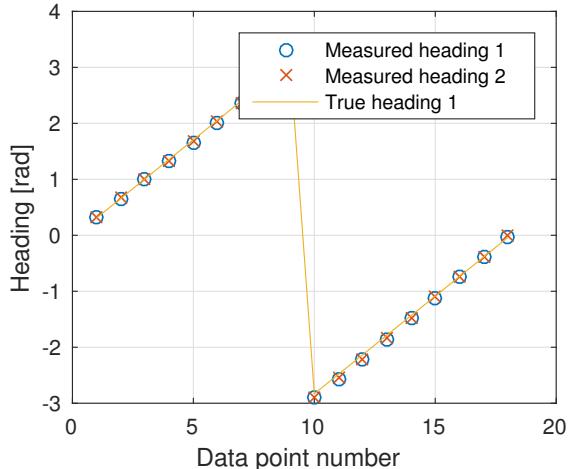


Figure 4.9: Measured heading values from repeated attitude determination tests plotted with the true heading values from the first test.

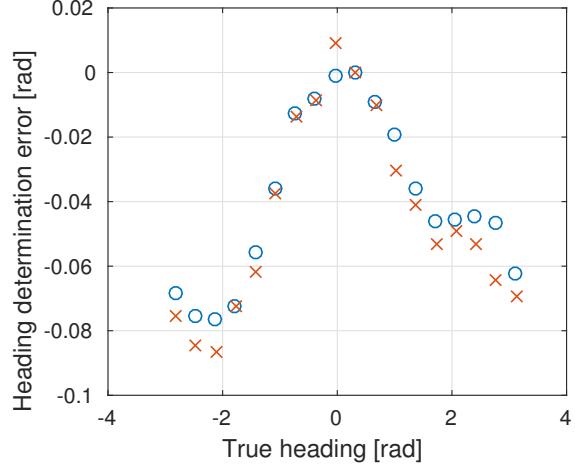


Figure 4.10: Heading determination error from the two attitude determination tests presented in Figure 4.9.

far from any external magnetic field sources. The pattern of heading measurement error also indicates that it is the result of biasing error. Figure 4.11 shows how bias error influences attitude determination accuracy. It will generally have a pattern of overestimating attitude for half of the arc and underestimating it for the other half. It was thus concluded that magnetic bias was also a function of the ambient magnetic field and that sphere center calculation must be conducted at the location that attitude determination will be performed in order to get best performance in terms of accuracy.

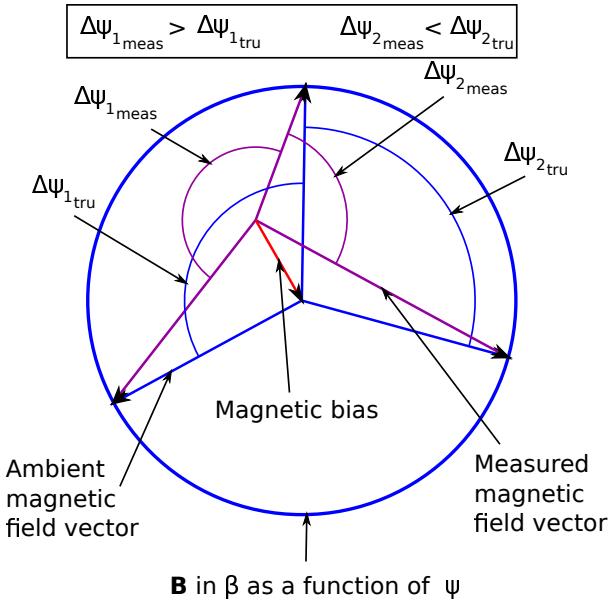


Figure 4.11: Diagram depicting how biasing error distorts the mapping of true heading to heading determined through magnetometer measurements.

Figures 4.12 through 4.14 show the result of 6 attitude determination tests that were conducted at the same location of and directly after magnetic bias correction. Each one of the figures shows the results of two tests that were taken 30 minutes apart without power cycling the test model. The test model was then power cycled before conducting the next two tests. These results reveal a number of properties:

- They show that sphere center correction at the test location greatly improves attitude determination accuracy.

- They show a heading determination error pattern that is not indicative of bias correction error.
- They indicate that at this level of accuracy, the mapping of true heading to heading determination error is not a function of time (changes in the environment), but rather changes as the test model is power cycled.

The changes from power cycling could be a property of the magnetometer or due to variations in the magnetic field produced by the electronics. In any case, the changes are not very great. Figure 4.15 plots the data from Figures 4.12 through 4.14 together. The largest change in heading determination error through the entire process was approximately 0.025 rad.

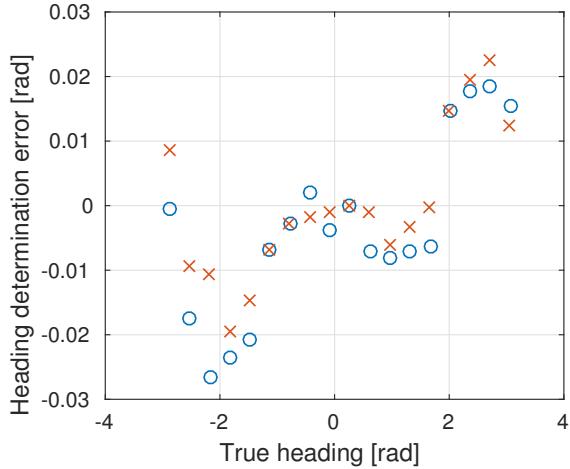


Figure 4.12: Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.

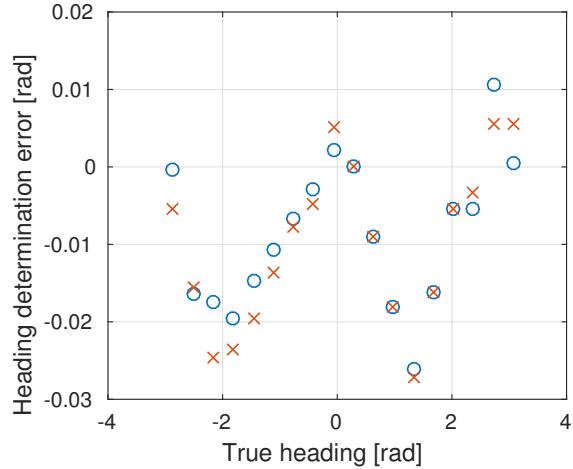


Figure 4.13: Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.

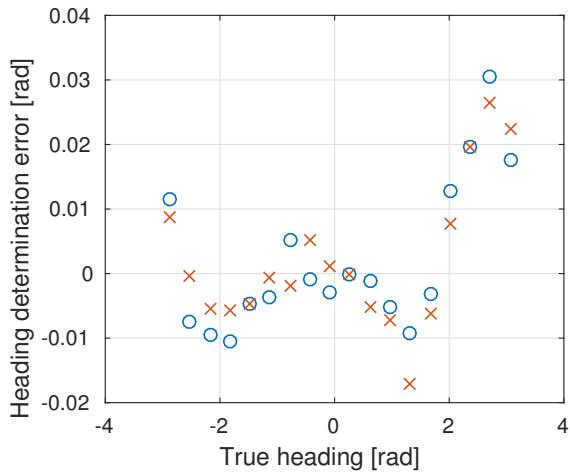


Figure 4.14: Heading determination error as a function of true heading for two attitude determination tests conducted 30 minutes apart.

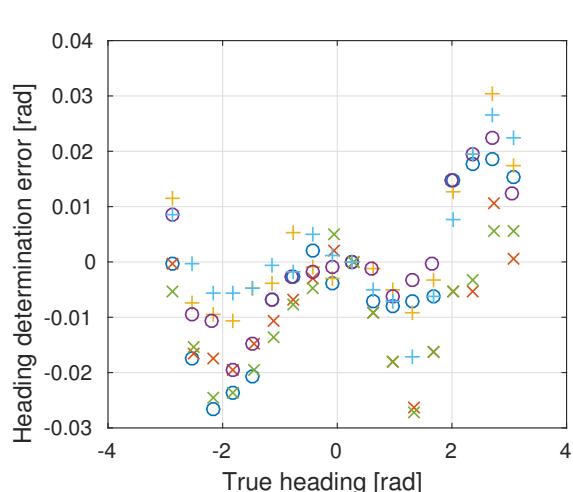


Figure 4.15: All of the data from Figures 4.12 through 4.14 plotted together. Both tests from the same figure share the same icon.

### 4.3. Considerations for implementation of attitude control

This section describes the methods concerning the implementation of attitude control that were developed to make sure that, in the event that test system is used to test attitude control performance, attitude control would function and provide usable test data given the constraints presented by the testbed.

### 4.3.1. Attitude error

ACS calculates the error MRP term using Equation 4.3 (Page 56). The target attitude is sent to the test model as a heading value and the attitude control system converts it to an MRP:

$$\boldsymbol{\sigma}_t = -\tan\left(\frac{\psi_t}{4}\right) \frac{\mathbf{g}}{|\mathbf{g}|} \quad (4.26)$$

In this case,  $\mathbf{g}$  is determined with the accelerometer. The system finds its own attitude using the same equation and determines the attitude error MRP ( $\boldsymbol{\sigma}_e$ ) with Equation 2.18 (Page 10):

$$\boldsymbol{\sigma}_e = \frac{(1 - |\boldsymbol{\sigma}_t|^2)\boldsymbol{\sigma}_{mes} - (1 - |\boldsymbol{\sigma}_{mes}|^2)\boldsymbol{\sigma}_t + 2\boldsymbol{\sigma}_{mes} \times \boldsymbol{\sigma}_t}{1 + |\boldsymbol{\sigma}_t|^2|\boldsymbol{\sigma}_{mes}|^2 + 2\boldsymbol{\sigma}_t \cdot \boldsymbol{\sigma}_{mes}} \quad (4.27)$$

Attitude error could alternatively be determined by finding heading error and then converting that to an MRP using Equation 4.26. Both methods are mathematically equivalent, but the former method was chosen because it is truer to flight conditions.

### 4.3.2. Maneuver selection

Proper selection of test initial conditions, particularly the initial speed of the reaction wheels, the control law gains and the target heading is important to avoid test conditions that would compromise results. The testbed software model can be used to verify the maneuver selection by performing a prediction of the test results beforehand. This section explains the requirements for selection of test conditions and how they can be met.

#### Magnetometer filtering

The moving average filter of the magnetometer will fail if the speed of any of the reaction wheels gets too close to 0. When performing a slew maneuver, if all of the reaction wheels are being commanded by the attitude control law, the perpendicular reaction wheels will be attempting to cancel out each other's gyroscopic torques as the test model rotates, which will eventually result in one of the wheels crossing 0 if the rotation angle is large enough. Two methods were identified as means to avoid this:

- Having the perpendicular reaction wheels commanded by speed control, meaning that they will retain whatever speed they start at.
- Carefully selecting the target deflection angle of the maneuver and initial speeds of the reaction wheels such that zero crossing does not occur.

If the test model is suspended along a primary axis, having the perpendicular reaction wheels on speed control would not change the expected attitude response.

#### Attitude filtering

The chosen attitude determination method is subject to error that results from motion of the test model. The biggest contribution comes from the inherent delay in using a moving average filter. The delay produces heading determination error when the test model has angular velocity. Delay is a concern when performing attitude control because it requires a live attitude feed, but it is irrelevant in tests like the oscillation test where the attitude of the system is only relevant in data analysis. Barring delay, there is also attitude error that gets produced when the test model is experiencing angular acceleration by interaction between a delayed attitude measurement and a prediction that is based on gyro measurements that do not experience delay. These effects can be reduced by reducing the angular velocity and acceleration that the test model is experiencing.

#### Reaction wheel torque saturation

A situation in which the attitude control system is applying maximum torque to a reaction wheel is not very indicative of its ability to control attitude and thus should typically be avoided. Reaction wheel saturation torque is a function of the reaction wheel spin rate and the expected output torque commands during a maneuver can be predicted in the testbed software model. Torque saturation can be avoided by selecting gains, the target heading and the initial reaction wheel spin rates such that a prediction of the maneuver in the software model does not indicate torque saturation.

### Max external torque

There may be a limit to the external torque that the test engineer wants the test model to experience. Provided that the spring properties of the testbed are fixed, this can be solved by limiting the deflection angle of the maneuver.

### Test time

There may be a limit to how long the test should take so the maneuver and testbed should be planned accordingly.

#### 4.3.3. New control law

Simulation of the control law that PolarCube will use on orbit (Equation 2.30 on Page 16) in the testbed software model revealed that integral control behaves inconveniently when rotation is constrained to one axis. Something about the  $(\widetilde{[K_i]z})([I]\omega)$  term makes the perpendicular reaction wheels continue torqueing after completing a slew maneuver. Little investigation was made into understanding why this happens. The main PolarCube orbit and attitude software model that simulates three dimensional rigid body dynamics free of constraints shows that the control law will converge on orbit. This issue was resolved by implementing the control law in Equation 4.28 [12]. Equation 4.28 is a more primitive form of the PolarCube control law. The  $(\widetilde{[K_i]z})([I]\omega)$  term was created to avoid reaction wheel saturation torque during high rotation rates due to the quadratic  $\omega \times [I]\omega$  term. The testbed dynamics model showed that using Equation 4.28 resulted in the perpendicular reaction wheels maintaining their speed once the target attitude is reached.

$$\mathbf{u}_s = -[I](\dot{\omega}_r - \omega \times \omega_r) + K\sigma_e + [P]\delta\omega + [P][K_L]z - \omega \times ([I]\omega + \mathbf{h}_s) + \mathbf{L} \quad (4.28)$$

This choice does somewhat compromise the testing because the system will be behaving differently to how it would in space. However, the control law used in testing is proven to be globally stable and thus the tests would at least validate a space viable attitude control system. It's also important to note that the validity of the control law itself can essentially be proven in a software model. The most important part of validation testing in terms of mission risk reduction is to prove that the hardware is able to execute the control law. There is always the option to perform tests with the original control law anyways. The testbed software model would be able to simulate the response, it would just complicate test maneuver selection and results analysis.

#### 4.3.4. Reaction wheel speed control law

The reaction wheel speed control law uses saturation control. The controller applies maximum torque in the direction of the desired reaction wheel speed until it is reached within a given deadband. This method will attain the desired rate as fast as possible and there is no risk that the controller will overshoot the target because angular velocity is a linear function of torque. This method proved effective. However, the available torque at the time of testing was far lower than it will be in the final design, so a more delicate control law might need to be implemented in the future.

### 4.4. Communication and data logging

This section discusses the implementation of the systems created to establish communication with the test model and provide the required uplink and downlink functionality. The communication architecture developed for attitude control validation testing is expected to be re-purposed in the future for various full system tests and mission operations.

#### 4.4.1. Wireless communication

XBee RF modules were used to establish wireless communication between CDH and an external interface computer. One XBee module connects to the auxiliary communications UART port on CDH and the other connects to the interface computer via USB with the XBee Explorer dongle (see Figure 4.3). A breakout board was produced with headers to mount the XBee and a connector to allow a cable to connect it to the port on CDH. The breakout board has the same form factor as PolarCube's GPS unit, making it easy to mount it into the test model. Kent Lee was responsible for the creation of the XBee breakout board with help from Franklin Hinckley. (see Figure 4.16)

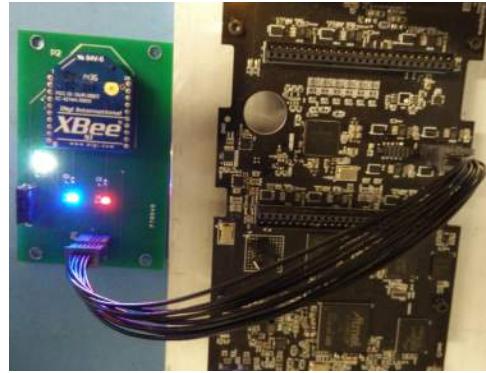


Figure 4.16: Image of the XBee on its breakout board connected to CDH.

#### 4.4.2. User interface

Figures 4.17 and 4.18 show the Graphical User Interface (GUI) developed in Python to provide a visual display of the incoming data stream of the test model's measurements and configuration as well as provide an interface to send commands to the test model. The uplink window keeps a history of commands sent so that they can be repeated easily without having to type the inputs back in. It also gives the test engineer the ability to turn ACS on and off as well as start and stop a test. Starting a test initiates the data stream down to the interface computer, allows op-codes to be sent up, opens a new data file on the SD card, and starts writing ACS data to it. This GUI architecture is expected to eventually serve as the ground station interface for the satellite when it is in orbit. Adam St. Amand was responsible for creation of the GUI and establishing communication between the interface computer and CDH with help from Russell Gleason and Umang Patel.

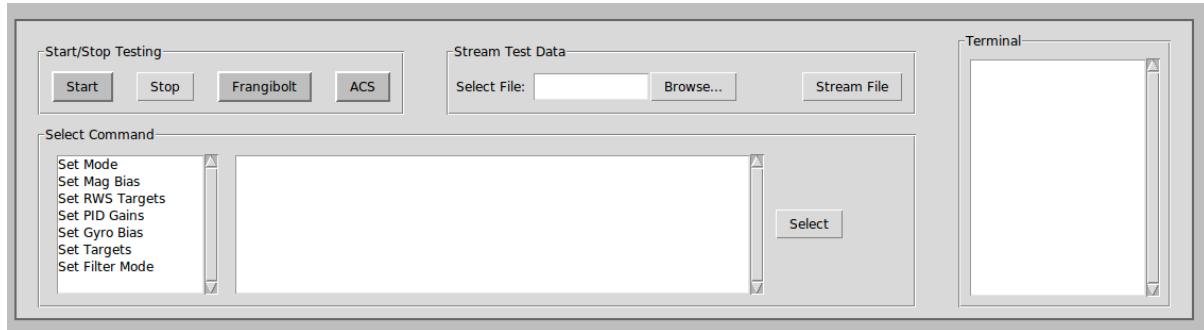


Figure 4.17: Uplink GUI window for sending commands to the test model.

#### 4.4.3. Commanding the test model

Commands are sent to the test model in the form of operation codes (op-codes). An op-code is a message with a destination address, an identifier byte and a list of input variables. ACS receives the op-code and writes the inputs to flight software variables based on the value of the identifier byte. Table 4.1 lists all of the op codes created for attitude control validation testing.

- **Mag bias** uploads the values that should be subtracted from every magnetometer measurement.
- **Gyro bias** uploads the values that should be subtracted from every gyro measurement. A set must be sent up for each Gyro.
- **Attitude control gains** uploads the  $K$ ,  $[P]$  and  $[K_I]$  values that go into the control law.  $[P]$  and  $[K_I]$  are  $3 \times 3$  matrices but they only contain values along the diagonal for testing.
- **Reaction wheel control mode** uploads variables that determine whether the torque sent to each reaction wheel gets retrieved from the attitude control law, the reaction wheel speed control law or is turned off. A negative value implies attitude control, a positive value implies reaction wheel speed control and 0 implies no torque sent.

**ACS Test Log**

|  |  |   |   |   |
|--|--|---|---|---|
| <b>Sensors</b><br>Mag X<br>Mag Y<br>Mag Z<br>Acc X<br>Acc Y<br>Acc Z<br>Gyro1 X<br>Gyro1 Y<br>Gyro1 Z<br>Gyro2 X<br>Gyro2 Y<br>Gyro2 Z | <b>Control Law Gains</b><br>K<br>Kix<br>Kiy<br>Kiz<br>Px<br>Py<br>Pz     | <b>Bias Values</b><br>Mag X<br>Mag Y<br>Mag Z<br>Gyro1 X<br>Gyro1 Y<br>Gyro1 Z<br>Gyro2 X<br>Gyro2 Y<br>Gyro2 Z | <b>Attitude</b><br>MRP X<br>MRP Y<br>MRP Z<br>Error MRP X<br>Error MRP Y<br>Error MRP Z | <b>Attitude</b><br>Heading<br>Pitch<br>Roll |
| <b>RW Speeds</b><br>X<br>Y<br>Z  | <b>Control Law Targets</b><br>Rate<br>Heading<br>RWS X<br>RWS Y<br>RWS Z | <b>Motor Override Mode</b><br>CTL Mode X<br>CTL Mode Y<br>CTL Mode Z  | <b>Mag Filter Mode</b><br>Filter Mode   | <b>Commanded Torques</b><br>X<br>Y<br>Z     |

Figure 4.18: Downlink GUI window for displaying the test model's state vector.

Table 4.1: Op-code definition for the validation testing interface.

| Op-code name                 | Variables  |
|------------------------------|--|
| Mag. bias                    | Mag. bias X, Mag bias Y, Mag bias Z  |
| Gyro bias                    | Gyro1 bias X, Gyro1 bias Y, Gyro1 bias Z,<br>Gyro2 bias X, Gyro2 bias Y, Gyro2 bias Z, |
| Control law gains            | K, Px, Py, Pz, Kix, Kiy, Kiz   |
| Reaction wheel control mode  | RWx mode, RWy mode, RWz mode   |
| Attitude targets             | Target heading, Target rate  |
| Reaction wheel speed targets | RWx target, RWy target, RWz target   |
| Filter mode                  | Filter mode  |

- **Attitude targets** uploads the target heading and rate about the rotation axis that are used in attitude error determination and in the control law.
- **Reaction wheel speed targets** uploads the target speeds for reaction wheel speed control of each of the wheels.
- **Filter mode** uploads a variable that determines whether or not the moving average filter is used on magnetometer data. Not using the moving average filter is good for sphere center calculation where the data will be filtered in data analysis. If "Filter mode" is set to 0, filtering is turned off. Turning filter mode off also makes ACS write data to the SD card at every sensor update (170 Hz) rather than at every attitude determination update (20 Hz).

The GUI does not include any commands pertaining to the magnetorquers, because by the time that the GUI was being created, it was already clear that magnetorquers would not be ready for testing before the end of the thesis project, so efforts were focused on testing with only the reaction wheels.

#### 4.4.4. Data stream down

ACS sends the state vector of the system, including sensor measurements, determined attitude in terms of Euler angles, reaction wheel speeds, torque commands to the wheels and all of the variables listed in Table 4.1 to CDH every attitude determination update, which was set at 20 Hz for testing. CDH writes that data to a text file on the one or both micro SD cards. Every half of a second, CDH sends the state variable to the

interface computer via XBee and it gets displayed on the GUI window in Figure 4.18. The slower data rate of live streaming was necessary for XBee communication to work stably.

## 4.5. Testbed

This section discusses the materials and methods used in implementing a string suspension testbed.

### 4.5.1. Test location

The string suspension testbed was set up in the mission operations command room of the COSGC. This location was selected as a best option for three primary considerations:

**Ventilation** The test location should have as little air flow as possible.

**Magnetic interference** The test location should be sufficiently distanced from local magnetic field sources. This does not have to be extremely far because magnetic field intensity decreases by a third power of distance from a point source. Figure 4.19 demonstrates how the test model was suspended in a large empty space.



Figure 4.19: Location in the mission operations command chosen to conduct string suspension tests due to the size of open space.

**The attachment point** Before the test model is suspended, magnetic bias correction and an attitude determination test must be performed at the test location. This means that the attitude determination test rig must be suspended from the same attachment point. The attitude determination test rig requires two suspension lines in order to constrain rotation about the vertical (see Figure 4.22). Thus the attachment point must be able to support three linearly aligned strings. The two outer strings are for the test rig and the central one will suspend the test model. Figure 4.20 shows the fire extinguisher pipe that the strings are tied to. The hole in the ceiling was closed up to stop airflow coming from the gap above the ceiling panes (see Figure 4.21).

### 4.5.2. Material selection

Ferromagnetic material should be avoided in building the test model and the attitude determination test rig because it has the potential to impact attitude determination performance. Ferrous material is present in the test model in the form of screws, reaction wheel rotors, magnetorquer cores and battery casings:

- Fortunately, the batteries and the screws used are made of a form of stainless steel alloy that does not magnetize. This was tested by placing them near a very powerful magnet. The attitude determination test rig has some steel screws and wire that are used to hold down the test model. They have also been tested with the magnet and do not magnetize.
- The magnetorquers would require more development before validation testing with them would be relevant so they were not included in the test model. When they are eventually implemented, they could pose a major concern to attitude determination performance in validation testing.



Figure 4.20: Fire extinguisher feed pipe in the ceiling used for attachment of the suspension lines of the test model and attitude determination test rig.

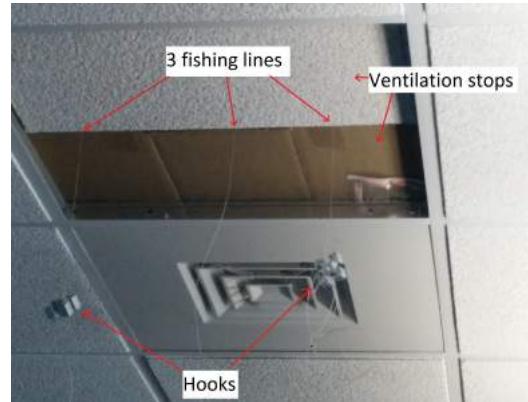


Figure 4.21: Demonstration of how the gap in the ceiling was filled to stop airflow.

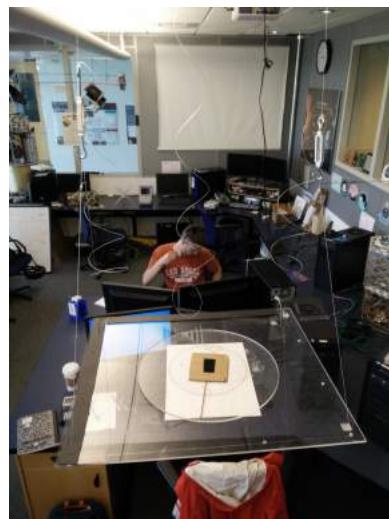


Figure 4.22: The attitude determination test rig suspended by two lines to constrain rotation.

- The hard iron effect of the reaction wheel rotors can be removed through appropriate filtering, but sphere center calculation of the test model with fixed wheels revealed that they have some soft iron properties and will distort the ambient field "sphere". One would expect this effect to be more or less negated when the reaction wheels are spinning and the signals are filtered.

#### 4.5.3. Suspending the test model

The string used to suspend the test model needs to have very consistent elastic deformation properties in order for it to act predictably as a spring. This calls for the use of monofilament fishing line. Figure 4.23 shows the brand of fishing line used. Many strengths of line were experimented with, but they were all from the same manufacturer. Rotational stiffness of the testbed can be adjusted by changing the length and thickness of the suspension line. The string must be rigidly attached to the test model's body as it rotates, any play will compromise the spring torque response.

The rotation axis of the testbed in the body frame is determined by where the string attaches to the test model body. The test model will rotate about the axis that is described by the point where the string attaches to the test model and the center of gravity of the test model. Alignment of the rotation axis requires that the attachment point be adjustable. Figure 4.24 shows a system of knots that allows the alignment of the test model to be adjusted and will act as a more or less rigid attachment due to friction when suspended. This system requires four holes at corners of the test subject that the attachment loop can be strung through. The test model conveniently has holes at its corners that would be used to attach the payload and external structures during flight. Figure 4.25 shows the test model suspended about the -X axis with this rigging. The pink tape is



Figure 4.23: String that was used to suspend the test models: 60 lb test South Bend monofilament fishing line purchased from Amazon.com.

intended to constrain alignment.

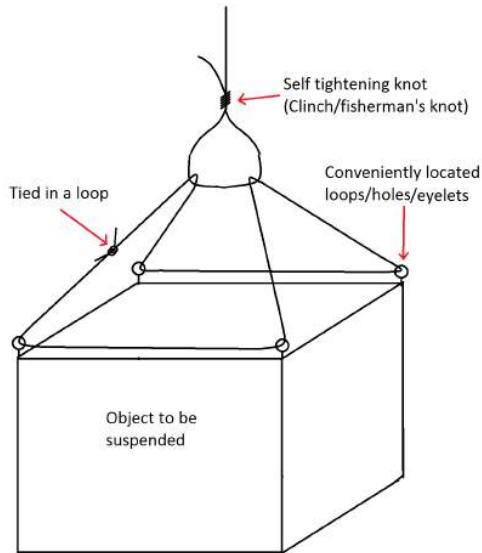


Figure 4.24: Suspension system using fishing line that allows the alignment of the test model to be adjusted.

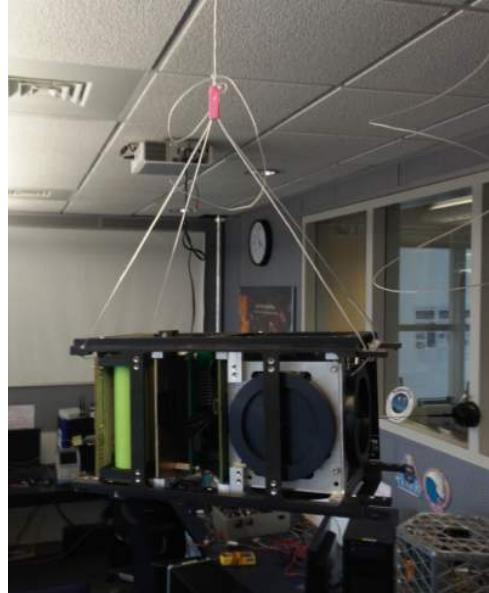


Figure 4.25: Test model suspended about the -X axis.

## 4.6. String torque characterization

This section details the experiments that were conducted to better understand the sources of fitting error in the oscillation test results conducted for the MicroMAS mission published in [22]. (Figure 3.1 on page 25). The efforts began with attempts to alter the dynamics model to better fit the measured data that was reported. The oscillation test was then repeated with the Arduino test model.

### 4.6.1. Attempts to find better fits to the MicroMAS test results

Efforts to improve upon the string torque characterization results published in [22] began with attempts to improve the fit to the MicroMAS test data by altering the dynamics model. Figures 4.26 and 4.27 show the results of the two most successful attempts. Figure 4.26 shows the output of a software model simulating motion of the one-dimensional testbed described on Page 11 under torques described by Equation 4.29 that has been fitted to the measured response in Figure 3.1a. It produces better correlation than the damped spring model, suggesting that drag torque is in fact proportional to the square of angular velocity rather than the damped spring assumption.

$$\tau_{\text{ext}} = -k\theta_d - c_q\omega^2 \text{sgn}(\omega) \quad (4.29)$$

The simulation output in Figure 3.1b shows a different initial angular velocity than the measured data,

which the author suspected to be impairing the fit. Figure 4.27 shows the output of a software model simulating motion of the one-dimensional testbed described in Section 2.3.1 (Page 11) under damped spring torques (Equation 3.1 on Page 24) that has been fitted to the measured response in Figure 3.1b where care was taken to match the initial  $\theta_d$  and  $\omega$  to the measured data. The new simulation fits the output better for the first three oscillations, but the amplitude drops faster than the measured data or the original simulation output. It is also interesting to observe that the first minimum (70s) and third maximum (250s) of the measured response show contradictory indications of  $c$ .

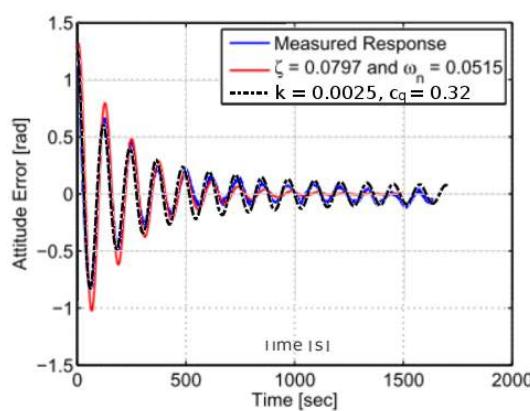


Figure 4.26: Comparison of the data in Figure 3.1a to the output of a simulation with external torques described by Equation 4.29. [22].

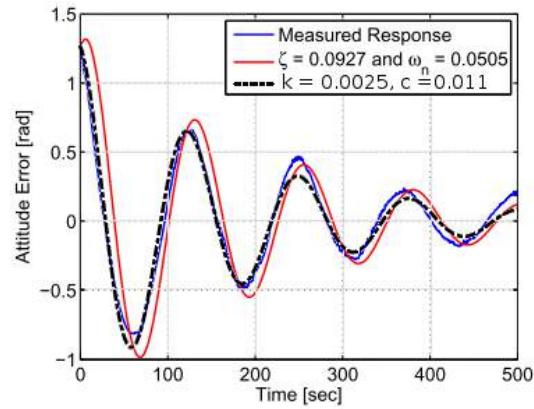


Figure 4.27: Comparison of the data in Figure 3.1b to the output of a simulation with external torques described by Equation 3.1 that has a closer fit to initial conditions of the measured data. [22]

#### 4.6.2. Oscillation tests with the Arduino test model

String torque characterization was first performed with the Arduino test model because it was ready for use far earlier than the ACS test model and it showed very good attitude determination performance. The Arduino test model first underwent magnetic bias correction, then was placed within a cardboard box that was suspended at the same location with the mechanism described in Figure 4.24. The box was deflected from equilibrium, then released and allowed to oscillate until motion was mostly damped out. This was repeated three times to show repeatability of results.

The one dimensional model of rigid body dynamics presented in Section 2.3.1 was then used to attempt to match the measured heading response by defining  $\tau_{ext}$ , and  $\psi_{eq}$  from Equations 4.4 and 4.5. The best fits were found with the external torque formulation defined in Equation 3.1, which would imply the following formulation for external torques produced by the testbed in three dimensions:

$$\tau_{ext} = -k\psi_d \hat{\boldsymbol{q}} - c\boldsymbol{\omega} \quad (4.30)$$

Figure 4.28 shows the measured heading responses of the Arduino test model plotted against the one dimensional simulation output that was best a fit to all of them. The same model of external torques is used for all four of the plots. The only element that varies in the production of the four output curves is the initial deflection angle. Some (second) of the responses fit better than others (first). This was discovered to be due to a difference in optimal  $\psi_{eq}$ , which is probably a result of the suspension rig shifting slightly when the test model was manually deflected. The measured heading responses were truncated to start at a local maximum so that initial angular velocity could be assumed to be 0 in the simulation.

These results more or less confirm the damped spring hypothesis, raising questions as to how the results in Figure 3.1 were produced. A number of possibilities were identified:

- Different string properties: This is unlikely because [22] explicitly mentions the use of a monofilament string and correspondence with the author confirmed this.
- Rigging: [22] used a hook and eyelet system to hang their test model (see Figure 4.29). This could potentially add play into the attachment of the test model to the string.

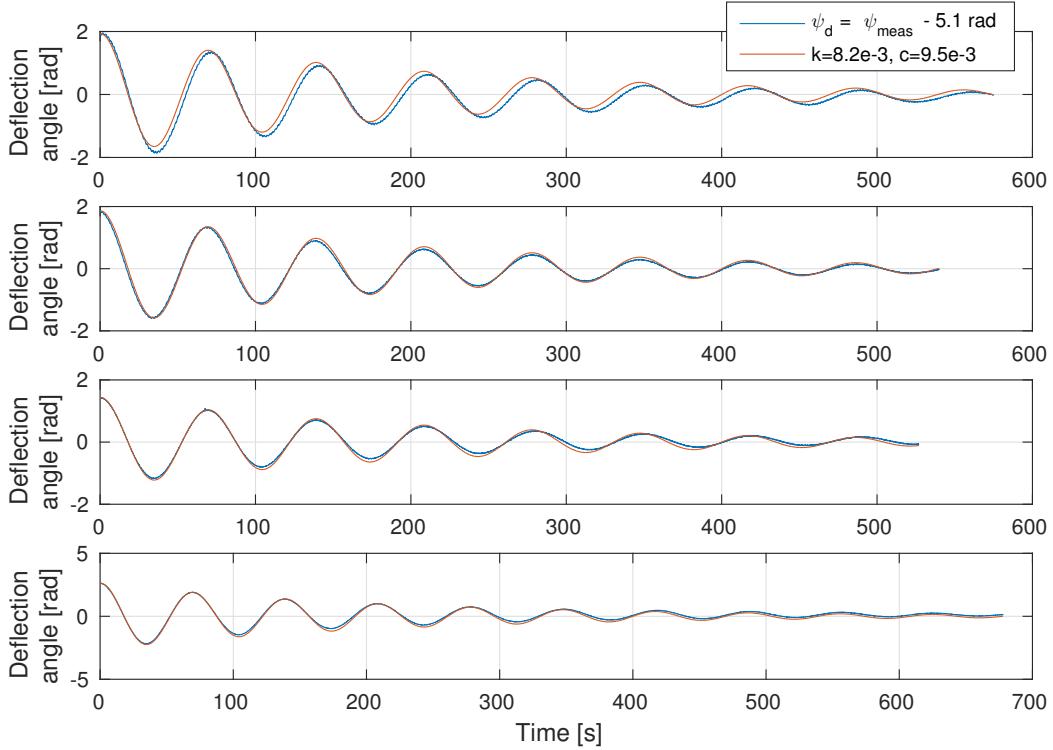


Figure 4.28: Raw heading from oscillation tests conducted with the Arduino test model plotted against responses from a one dimensional model of damped rotational spring motion.

- Ventilation: Correspondence with the author of [22] revealed that the tests were performed in a ventilated room with several other activities going on simultaneously, so this could be a contributing factor. However, the results in Figure 3.1 are claimed to be very repeatable, so if ventilation was contributing, it was doing so consistently.
- Aerodynamics: the fact that a quadratic response fits the results from [22] so well (see Figure 4.26) could be because air flow was turbulent around their test model [11]. This would not be due to speed because the responses in Figure 4.28 have a higher frequency and amplitude than in the response in Figure 3.1. It is unlikely to be due to form factor, because they both had the shape of a box. It might be the result of a ventilated room. Ventilation could, in theory, produce 0 net torque because the test model is essentially symmetric about the rotation axis, but render the air surrounding the test model turbulent as it collides with the test model.
- Attitude determination inaccuracy: [22] makes no mention of magnetic bias correction or of verification of attitude determination. The results in Figure 3.1 could be the result of an amplified version of the effects observed in Figure 4.10 (Page 62). The author suspects this to be one of the primary contributors.

## 4.7. Test procedures

This section discusses the approach to developing the validation test procedures. The primary goal of the procedures is to enable test engineers at the COSGC to repeat and further the tests performed in the thesis to validate PolarCube's attitude control system when it is ready. They will hopefully also be valuable to any other Cubesat project that is attempting attitude control validation, especially at TU Delft. The procedures themselves were written by the author and can be found at the end of the document. The procedures cover integration of the test model, verification of electronics and software, sensor biasing, setting up the testbed, testing attitude determination and string torque characterization. The software developed for testing is available for distribution upon personal request. The test procedures are an official document for the PolarCube mission, and so they conform to the PolarCube test procedure format.

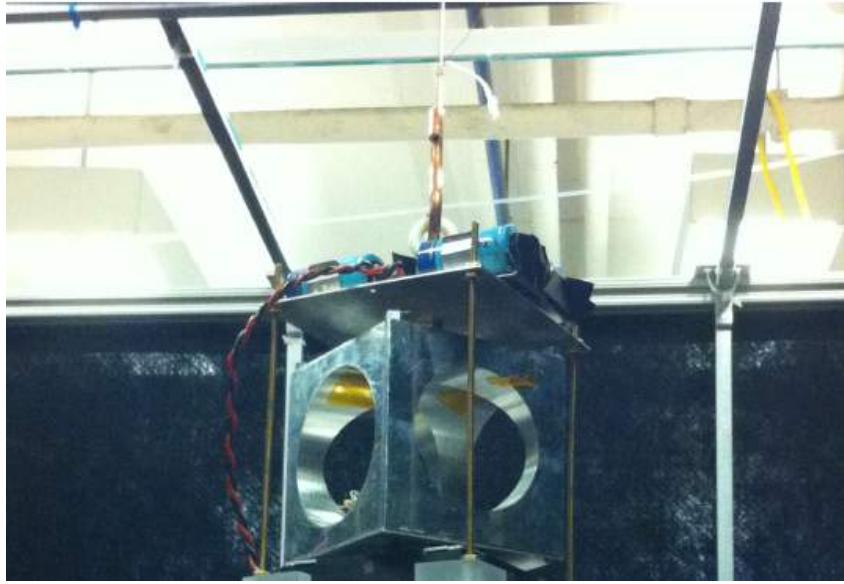


Figure 4.29: Image of the MicroMAS attitude control test model suspended using an S-hook and eyelet connection. [22].

#### 4.7.1. Integration

Integration procedures for the test model were written to ensure that it had the minimum functionality required to perform attitude control validation testing. The integration procedures themselves consist mostly of a process of incrementally verifying functionality of the electronics and software of the test model so as to understand the root cause of any errors that might occur. They also served as a set of incremental functional goals when developing the first test model. By passing the integration acceptance checklist, the test model should be ready to perform any of the functions asked of it in the following procedures. Integration and debugging of the test model for test system verification was conducted by Adam St. Amand, the author, Franklin Hinckley, Kristen Hanslik, Russel Gleason, Connor Kelleher, Umang Patel and Christopher Rouw.

#### 4.7.2. Software tools

The author wrote software to automate the process of collection, parsing and analysis of validation testing data. Each test has a set of scripts in a file structure that can take a raw data file in the target directory and process it into test results.

The scripts include a universal parsing algorithm that accepts a delimited text file and organizes it into a struct of data sets defined by the user. The script can use system configuration parameters in the data logs as cues to whether data should be retained or removed. The test procedures take the test engineer through the process of using the parsing algorithm effectively for each test.

There are also scripts to take the output of the parsing algorithm and determine the required test metrics, such as sphere center calculation, attitude determination error or fitting the output of a testbed software model to measured data. Instructions on using the analysis scripts are provided in the test procedures.

Every test campaign should have its own version of the parsing and analysis software directory. This keeps a full record of the test results and how they were obtained down to the raw data and it allows the test engineer to modify the scripts to cater to each set of test results without being concerned about compromising the original.

#### 4.7.3. Reporting results

Any full set of test procedures should be useable as a formal record of the tests taking place with space for notes and fields where test results should be reported, such as determined bias values or the location of the analysis scripts. Each test procedure ends with a list of acceptance criteria that must be met for the test to be considered successful.

## 4.8. Chapter summary

This chapter lays out the methods and materials used to produce a test system (Testbed, support systems and methodology) for measurement of CubeSat attitude control performance based on a string suspension microgravity simulation testbed. The test system was intended to produce the best performance possible in terms of attitude determination, determination of external torques on the test model and ability to simulate the test model's motion with a software dynamics model.

A software model of the dynamics of the test model and its interaction with the testbed was produced in order to make predictions of the test model's measured attitude response to test initial conditions. The model exists as a MATLAB script that integrates a set of differential equations with shared inputs and outputs. The model propagates rigid body dynamics of the test model using MRPs and assumes that motion is constrained to rotate about a user defined rotation axis in the  $\beta$  frame. The software model assumes that the test model is subject to external torques produced by the testbed and torques produced by the reaction wheels. The model also simulates attitude determination by first simulating sensor readings, then performing the attitude determination algorithm on them.

Attitude determination and wireless communication were first developed using an Arduino-based test model. The Arduino test model uses the same attitude sensors and radio that would eventually be used on the satellite test model. The first step in achieving attitude determination is to correct for magnetic bias produced by the electronics. This is achieved by collecting magnetometer readings while rotating the test model in all directions. The resulting data will represent a sphere that is not centered about the origin. Magnetic bias is determined by calculating the center of the sphere and can be removed by subtracting the determined bias from every magnetometer reading. A bias corrected magnetometer signal is still subject to sinusoidal disturbances produced by rotating reaction wheels. This was mostly removed by implementing a live moving average filter on magnetometer readings.

A TRIAD algorithm was implemented to determine the attitude of the test model with respect to the local gravitational and magnetic field in terms of Euler angles. Singularities were avoided by creating a transitional body frame depending on the direction of the measured gravity vector. Attitude measurements produced by the TRIAD algorithm go through a predictor-corrector filter where they get weighed against attitude predictions made with readings from the gyros.

Heading determination performance is tested by using a protractor to align the test model along several known heading angles and comparing measured heading values to the known heading angles. Conducting this test with the Arduino test model revealed that magnetic bias must be corrected for at the same location that attitude determination will be performed in order to get best performance. Testing with optimal bias correction revealed that heading determination performance will change when the test model is power cycled.

ACS' attitude control law accepts attitude error in the form of an MRP. The target attitude is provided as a heading value and then converted to an MRP by ACS. When selecting the initial conditions for a slew test, several eventualities were identified that would compromise results. These can be avoided by using the testbed software model to first simulate the maneuver and make sure that none of them are expected to occur. The control law that PolarCube will use in orbit is ill-suited to conducting tests with a constrained rotation axis. A variation on the control law was identified that would be easier to test. A separate control law was developed to produce the desired initial reaction wheel speeds when conducting a test.

A breakout board was produced to allow the XBee radio module to communicate with CDH through its auxiliary communications port. ACS receives commands from the test engineer in the form of op-codes that are generated with a GUI on the interface computer, sent wirelessly to CDH, then sent via Ethernet to ACS. A set of op-codes was developed to provide all of the functionality necessary to conduct the tests envisioned. ACS' state vector, including sensor measurements, determined attitude and configuration is stored on a micro SD card on CDH at every attitude determination update and sent to the interface computer to be displayed on a GUI twice a second.

When suspending the test model, care must be taken to minimize ventilation and the use of ferromagnetic material. The location must also have points from which to suspend the test model and the attitude determination test rig far from any nearby magnetic field sources. The test model should be suspended using a monofilament line. A rigging was devised to allow tuning of the alignment of the test model with respect to the rotation axis.

String torque characterization tests were conducted with the Arduino test model. The tests confirmed the hypothesis that external torques on the test model are analogous to a damped rotational spring. This raised the question as to why this was not observed in tests of the MicroMAS attitude control system. A number of theories were proposed.

Test procedures were produced to allow attitude control validation tests to be conducted by a future test engineer at the COSGC. The goal was to make conducting the tests as straightforward as possible. The procedures currently only cover the integration and tests that were conducted during the thesis project. Software tools for automatic parsing and analysis of data were produced and their use is included in the procedures.



# 5

## Verification of test system performance

This chapter presents the results of tests conducted to verify the performance of a test system (testbed, support systems and methods) to measure CubeSat attitude control performance. An attitude control test system was produced for the PolarCube satellite based on a string suspension microgravity simulation testbed and attitude determination based on MEMS magnetometer, rate gyroscope and accelerometer measurements. The tests presented in this chapter measured the performance of attitude determination and the ability of the methods to determine the external torque properties of the testbed. Both of which are important in order to conduct high fidelity fit-predict-fit (Page 26) analysis of attitude control performance. The chapter also includes the results from tests necessary to configure attitude determination as well as results from an analysis of attitude determination performance using a computer simulation of testbed dynamics. The tests presented constitute all of the steps necessary to take a freshly integrated test model and conduct a successful oscillation test. The procedures for these tests and integration of the test model are available at the end of the thesis document.

The execution of each test is explained in detail, the results are presented and followed by some discussion when relevant. Testing was limited to verification of the test system because component-level testing revealed that ACS could not control the magnitude of torque sent to the reaction wheels. The results of the component-level reaction wheel torque tests are presented in Appendix A. ACS was however able to control the speed of the reaction wheels through the control law explained on Page 65, which was essential in order for attitude determination to function. Schedule did not allow most of these tests to be repeated, so some of the data is incomplete or has errors that could be removed by repeating the test more carefully.

### 5.1. Gyro biasing correction

Error in MEMS rate gyroscope measurements is the result of combined noise, bias and sensitivity uncertainty. The effect that these properties have on attitude determination was included in the dynamic model of the string testbed. The results showed that the levels of noise measured and the sensitivity error cited in the gyro's datasheet [28] would not have a significant impact on attitude determination performance. Bias error had the greatest potential to corrupt attitude determination as it was implemented for the validation tests.

Figure 5.1 shows recorded output of the gyros while the test model was at rest. The data collected allows measurement of noise and bias. The measured bias is then sent to the test model using the uplink GUI and bias is automatically corrected for in the flight software. The following measured bias values were determined from the data displayed in Figure 5.1 and used in the attitude determination and string torque characterization tests that are reported on further in the document:

$$X \text{ bias} = 7.5 \text{ counts} \quad Y \text{ bias} = -66 \text{ counts} \quad Z \text{ bias} = 62.5 \text{ counts} \quad (5.1)$$

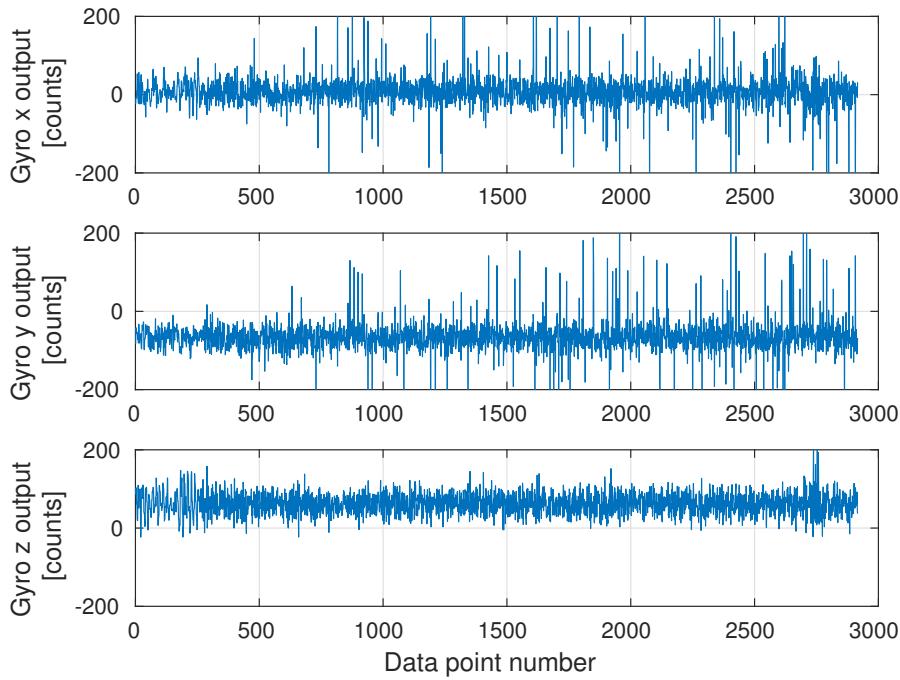


Figure 5.1: A set of raw gyro data collected while the test model was stationary.

## 5.2. Influence of the reaction wheels on the magnetometer

This test measures the influence of each reaction wheel on each magnetometer axis by sampling magnetometer data while a single wheel is spinning. Doing this for each wheel provides the amplitudes of all the sinusoids that result from the spinning wheels affecting each magnetometer. These amplitudes go into the testbed dynamic software model and are used to find appropriate gains for the moving average and predictor-corrector filters.

The data from this test was conducted while the test model was connected to the ACS electrical ground support equipment, which due to some improvised electrical fixes, does not allow torque on the Z reaction wheel. Time constraints and absent mindedness prevented this test from being repeated in the complete test model before the author's departure from Boulder. In further analysis, the testbed dynamics model assumes that the amplitude of the Z wheel's signal is 700 counts for all three magnetometer axes.

Table 5.1 gives the amplitudes that were determined by taking half of the range of the magnetometer reading dataset. Figure 5.2 displays a portion of the raw test output data.

Table 5.1: Amplitude of respective reaction wheel influence on each magnetometer axis.

|                     | X Wheel | Y wheel | Z wheel |
|---------------------|---------|---------|---------|
| X magnetometer axis | 244     | 500     | -       |
| Y magnetometer axis | 267     | 770     | -       |
| Z magnetometer axis | 213     | 349     | -       |

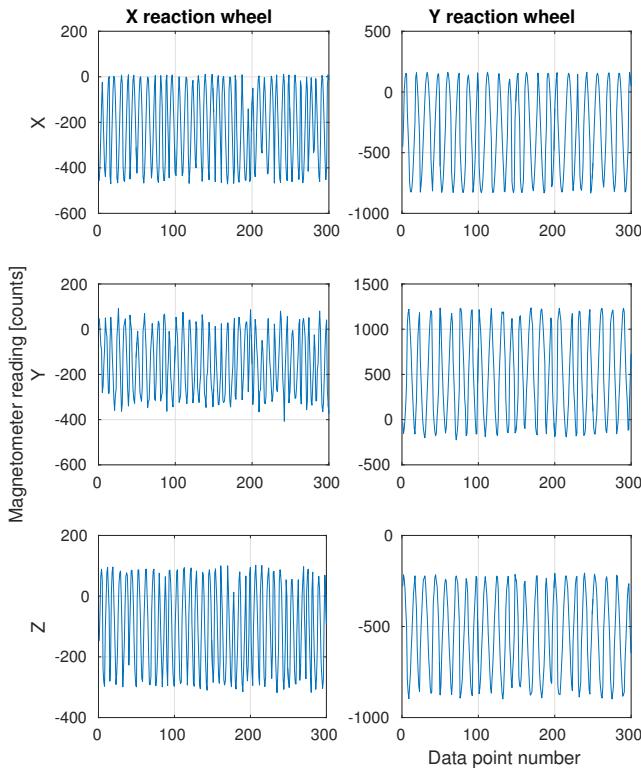


Figure 5.2: Raw magnetometer data that was collected on the integrated test model while two wheels were constrained and a single wheel was spinning. Each column of plots represents data that was collected simultaneously.

### 5.3. Sphere center calculation

Sphere center calculation (Page 58) with the satellite test model is far more problematic than doing so with the Arduino test model because the magnetic influence of the reaction wheels means that the data gets significantly corrupted and requires filtering. Data filtering for sphere center calculation was performed in post-processing to allow the filter to be adjusted for best performance. Two filtering methods were considered, a moving average filter and a Butterworth filter. The moving average filter showed better performance.

The performance of this method is very difficult to quantify, comparison was made using the vector magnitude standard deviation, vector magnitude plots and visual inspection of the sphere plots (looking for something that most resembles a sphere). A bin size of 16 was selected, vector magnitude standard deviation did not significantly decrease with larger sizes. A bin size of 16 roughly represents two full oscillations of the disturbance signal.

Sphere center correction with the test model is a highly sensitive endeavor. If the test engineer touches a wheel, overly favors any position, moves the model too fast or too far away from the test location, the result will be significantly far from optimal. Eight datasets for sphere center correction were collected. One of those datasets was selected based on the performance criteria mentioned above

#### 5.3.1. Filtering results

Figure 5.3 shows the magnetometer data from the sphere center calculation that was used for the final string characterization tests. The raw data is displayed in blue and the green line is the filtered data that was used for the sphere center calculation. The red arrows indicate rapid motion events that correlate with the sharp drops in vector magnitude displayed in Figure 5.4. Using the sphere center calculation algorithm on the filtered data produced the following:

$$\text{X bias} = 234 \text{ counts} \quad \text{Y bias} = -313 \text{ counts} \quad \text{Z bias} = -363 \text{ counts} \quad (5.2)$$

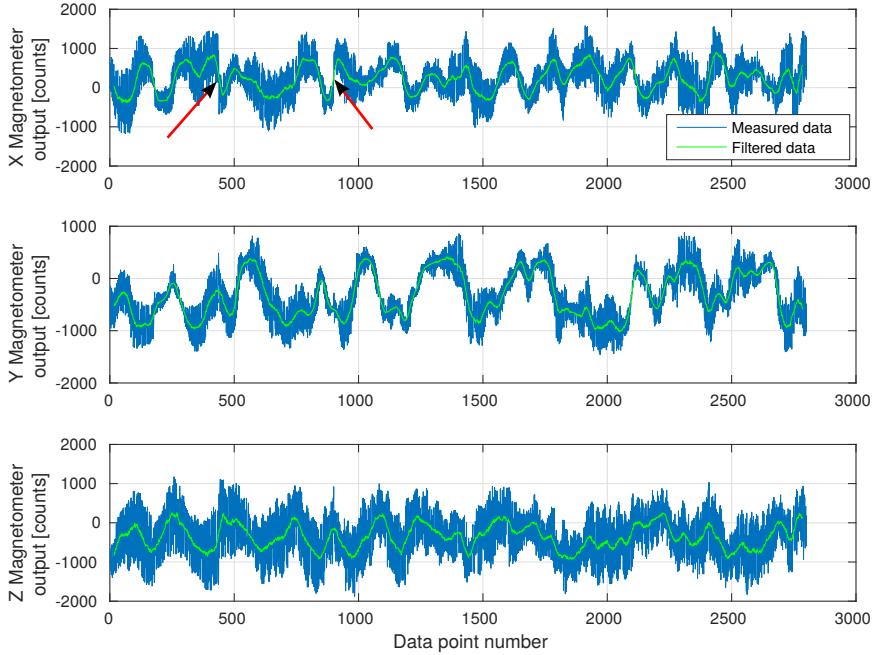


Figure 5.3: Raw and filtered magnetometer data used in the sphere center calculation that was implemented in the final tests.

### 5.3.2. Vector magnitude plot

Figure 5.4 plots the vector magnitude of the filtered data in Figure 5.3. Note two large drops in vector magnitude near the 500 and 1000 data point marks. These kinds of drops were present in all of the sphere center correction datasets. This dataset was chosen primarily because it had the fewest (only two). Their cause is uncertain, but there appears to be a temporal correlation with moments when the magnetometer data curve on one of the axes is particularly steep. The red arrows on Figure 5.3 show the events that correlate to the sharp drops in Figure 5.4. This could mean that the filter is unable to handle these kinds of transitions and they could be avoided by carefully handling the model such that it does not move too fast during data collection.

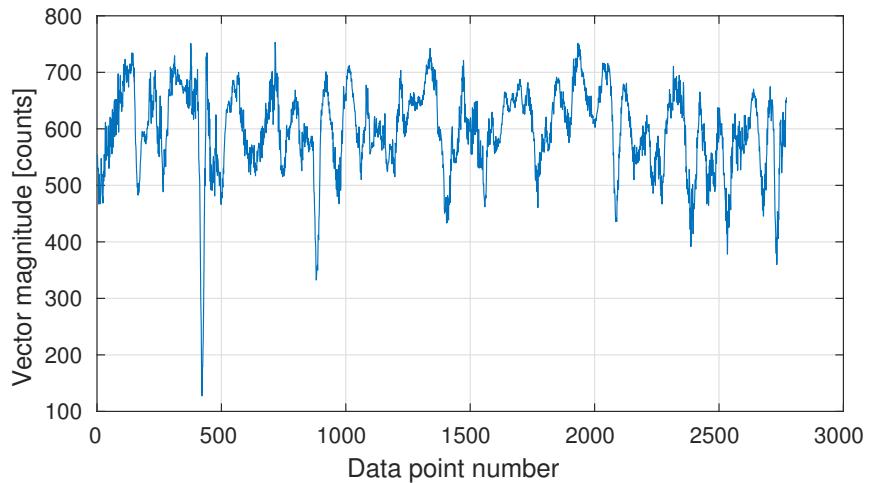


Figure 5.4: Vector magnitude of the filtered and bias-corrected magnetometer data from Figure 5.3. The data has a standard deviation of 74 counts.

### 5.3.3. Sphere plots

Figure 5.5 shows the filtered data (green lines) from Figure 5.3 plotted in three dimensions after receiving the bias correction detailed in Equation 5.2. The result is by no means a perfect sphere. Error is expected to be a combined result of filtering failures and soft iron effects.

Sphere center calculation biases can occur if the data is not evenly distributed. In this case, there is high data point density in the region where Y is large, because it was a position that provided easy finger holds. This combined with some oblateness in the Y direction appears to result in a slight -Y bias in the corrected data. The test engineer decided to use a Y bias correction value from a different sphere center calculation that had looked better in that respect, resulting in a magnetometer bias of -320 rather than -313. This kind of analysis is highly subjective and not entirely recommended.

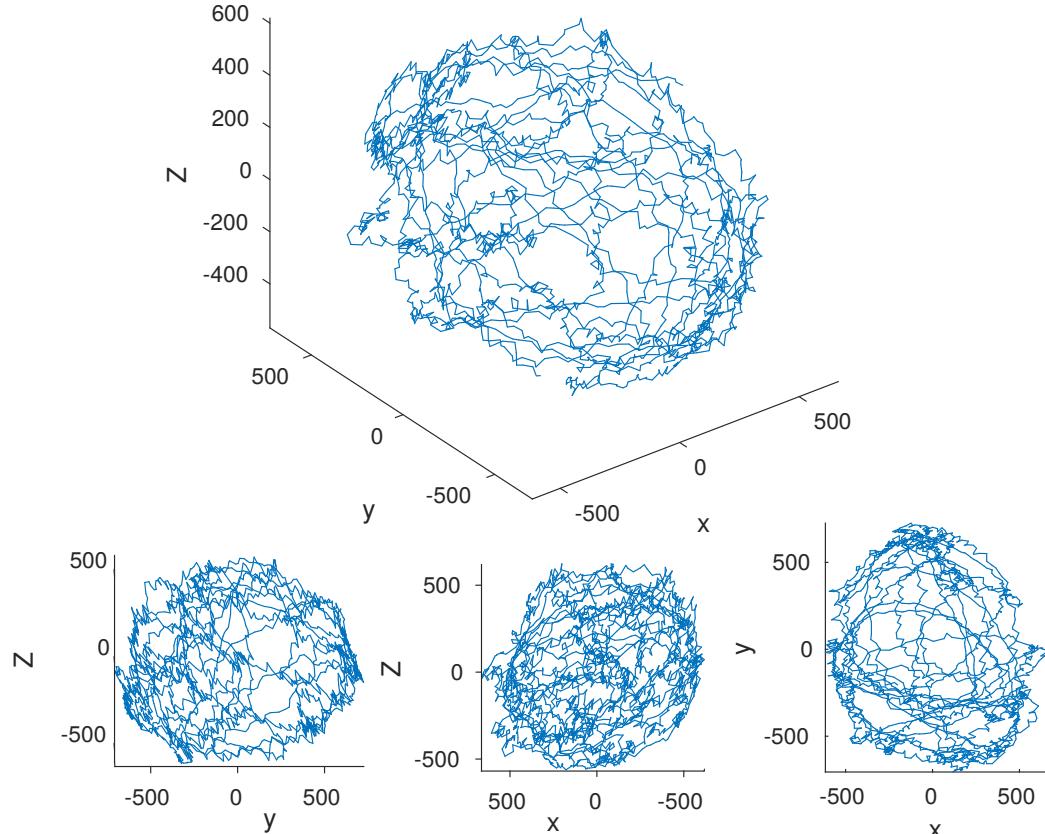


Figure 5.5: Filtered magnetometer data from Figure 5.3 that has been corrected for bias and plotted in 3D.

## 5.4. Attitude determination test

The attitude determination test verifies heading determination by recording heading measurements from the test model at a set of known heading points. The procedure is explained in Section 4.2.6 (Page 61) and is performed with a  $\psi_{int}$  of  $20^\circ$ .

The test was conducted in each orientation that the test model would be in for the oscillation tests at the same location that they would take place to give the best indication of attitude determination performance. For these tests, the +X, +Y and -Z axes were chosen for attachment purposes and because they minimized rattling of the Z reaction wheel. Each axis was tested in turn, then the process was repeated. The model was power cycled between each test. Getting the same output from repeated tests shows that the performance measured can be expected to remain during the string characterization tests.

Figures 5.6, 5.7 and 5.8 show attitude error as a function of true heading. The measure of attitude determination performance in these tests is the range of heading determination error. These tests show much lower attitude determination accuracy than the tests with the Arduino test model (Figures 4.9 through 4.15 on Page 62). However, they demonstrate a very similar degree of repeatability. Figures 5.6 through 5.8 show a maxi-

mum variation of 0.03 rad between repeated heading measurements at the same true heading, which can be compared to a maximum variation of 0.025 rad in the data in Figure 4.15 (Page 63). Major sources of heading determination error certainly include imperfect magnetometer biasing. They could also include magnetic coupling due to materials in the test model with "soft iron" properties.

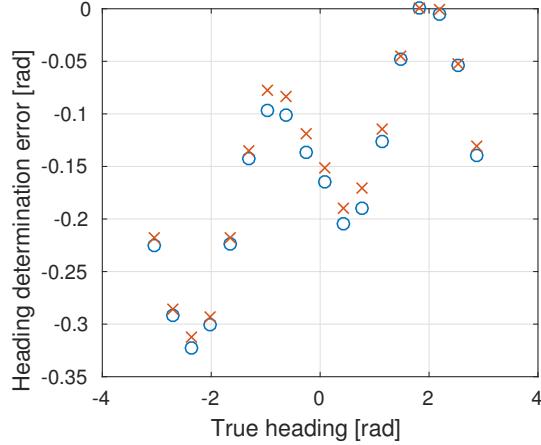


Figure 5.6: Heading error from the two attitude determination tests performed about the +X axis as a function of true heading. The total range of heading determination error is 0.33 rad.

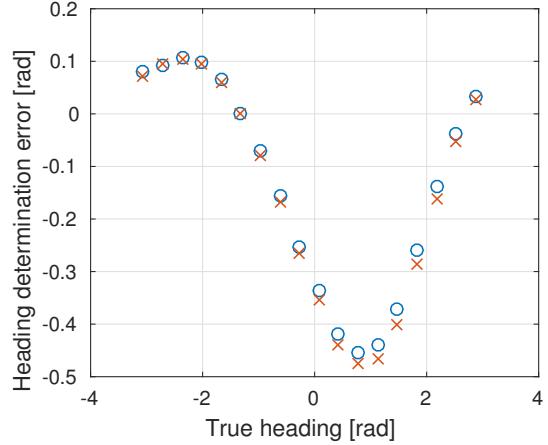


Figure 5.7: Heading error from the two attitude determination tests about the +Y axis as a function of true heading. The total range of heading determination error is 0.59 rad.

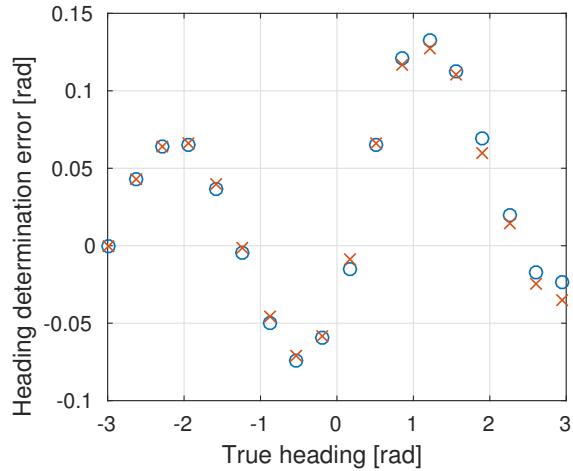


Figure 5.8: Heading error from the two attitude determination tests about the -Z axis as a function of true heading. The total range of heading determination error is 0.2 rad.

## 5.5. String torque characterization

Conducting an oscillation test with the satellite test model differs from the test that was reported on in Section 4.6.2 (Page 71) in that rather than comparing results to the one-dimensional scenario described in Section 2.3.1 (Page 11), the results are compared to the testbed dynamics model described in Section 4.1 (Page 55). When performing string torque characterization, the string testbed software model accepts an inertia tensor ( $[I]$ ), rotation axis ( $\hat{q}$ ), spring coefficient ( $k$ ), damping coefficient ( $c$ ), equilibrium heading ( $\psi_{eq}$ ) and initial heading position ( $\psi_0$ ). The software model propagates motion from the initial position assuming that the initial angular velocity is 0, the test model is only subject to spring and damping forces and motion is constrained about the rotation axis. The inertia tensor of the test model was determined using a CAD model of the test model:

$$[I] = \begin{bmatrix} 2.106 \cdot 10^{-3} & -1.1 \cdot 10^{-5} & 1.2 \cdot 10^{-4} \\ -1.1 \cdot 10^{-5} & 2.14 \cdot 10^{-3} & 1.58 \cdot 10^{-5} \\ 1.2 \cdot 10^{-4} & 1.58 \cdot 10^{-5} & 9.75 \cdot 10^{-4} \end{bmatrix} \text{Kg} \cdot \text{m}^2 \quad (5.3)$$

In this case, The CAD model did not include the XBee breakout board or any of its electrical or mechanical connections.

Figures 5.9, 5.11 and 5.13 show the measured deflection angle ( $\psi_d$ ) of the test model plotted with the response produced by a string testbed model that has been fit to the data. Measured deflection angle is determined by subtracting the equilibrium heading in the testbed software model ( $\psi_{eq}$ ) from the measured heading ( $\psi_{mes}$ ). It should be noted that the simulated responses are of true attitude and do not include simulation of attitude determination as described in Section 4.1.2 (Page 56). The measured heading responses were truncated to make sure that they began at a local maximum or minimum. A minimum of three oscillation responses are recorded for each configuration to prove repeatability of the test results. The same inputs for  $k$ ,  $c$  and  $\psi_{eq}$  are used for all of the tests conducted about a particular axis. The only variation in input is the initial position. The values of  $k$ ,  $c$  and  $\psi_{eq}$  corresponding to the best fit string testbed model for each axis are provided in the legend. The units of  $k$  and  $c$  are Nm/rad and  $\frac{\text{Nm}}{\text{rad/s}}$  respectively.

Heading determined by the test model is the only measure of the model's motion. Hence, any errors in heading determination will result in a worse fit to the dynamic model and errors in determination of string properties. Figures 5.6 through 5.8 show that significant heading determination error remained in the system when conducting the tests and the consequences for string torque characterization are very evident in Figure 5.11. Thankfully, although attitude determination error remains, uncertainty in attitude determination remains very low. In other words, the heading determination error at any measured heading is known. Feed-forward correction using data from the attitude determination tests can give roughly an order of magnitude of improvement in heading knowledge.

Figures 5.10, 5.12 and 5.14 show the measured heading data from Figures 5.9, 5.11 and 5.13 after a heading correction using the data displayed in Figures 5.6 through 5.8. The purple dotted lines show the original measured responses. Heading is corrected using a linear interpolation of the heading error values as a function of measured heading. The values of  $k$ ,  $c$  and  $\psi_{eq}$  corresponding to the best fit string testbed model for each axis are provided in the legend.

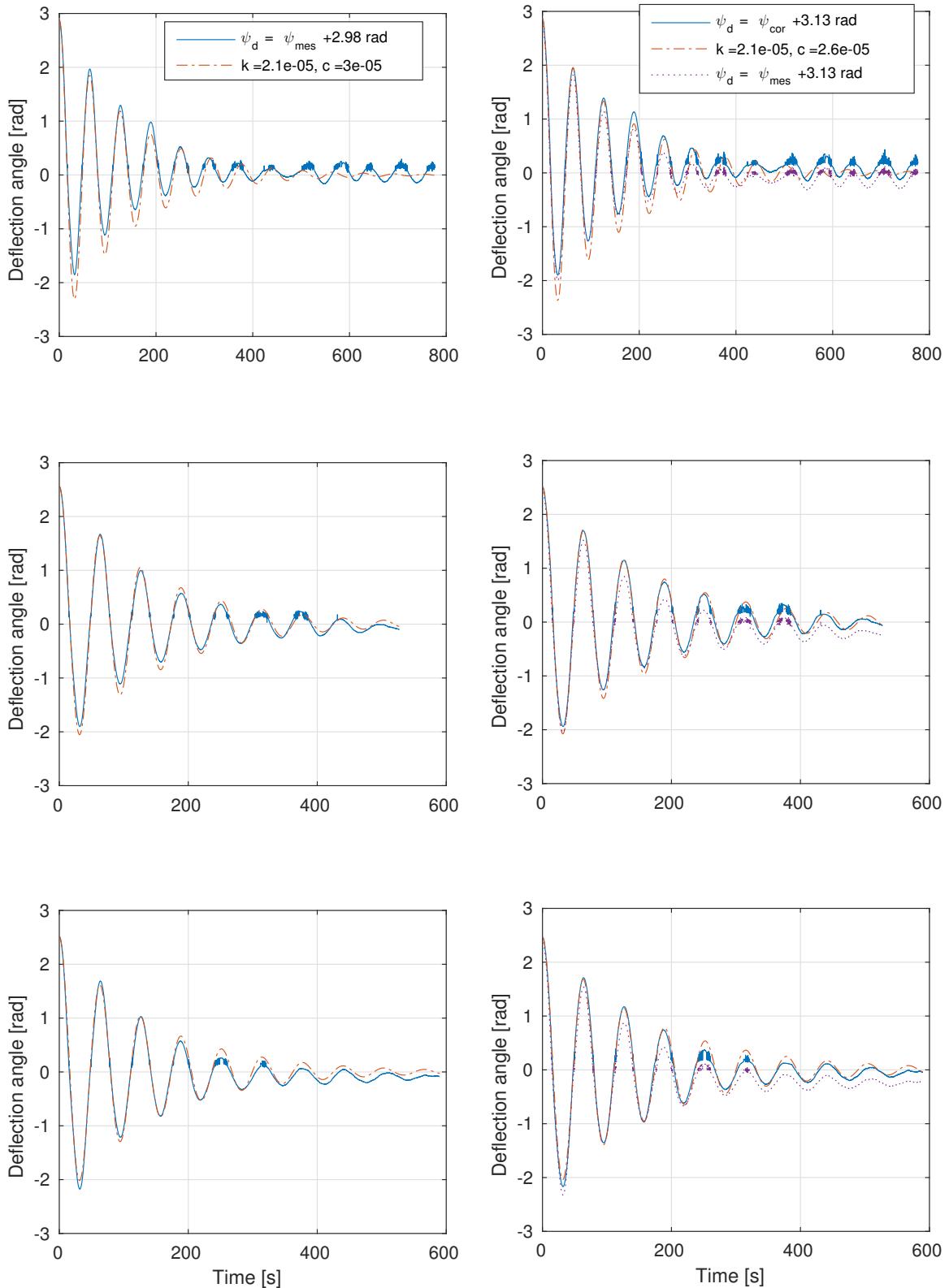


Figure 5.9: Raw measured heading from the oscillation tests about the +X axis plotted against the response from the best fitting damped spring model.

Figure 5.10: Heading from the oscillation tests about the +X axis that has been retroactively corrected for attitude error using data from Figure 5.6 plotted against the response from the best fitting damped spring model.

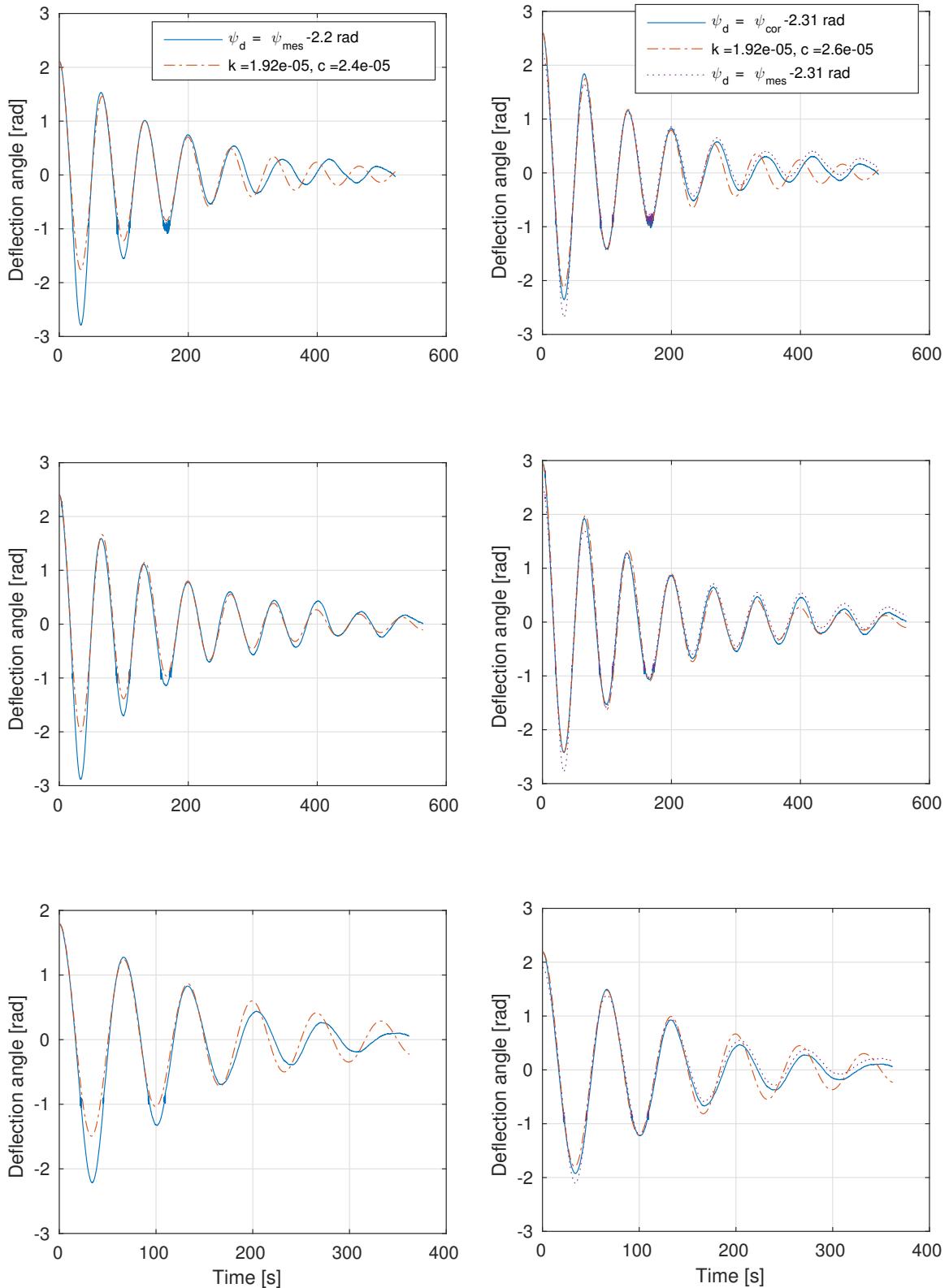


Figure 5.11: Raw measured heading from the oscillation tests tests about the +Y axis plotted against the response from the best fitting damped spring model.

Figure 5.12: Heading from the oscillation tests about the +Y axis that has been retroactively corrected for attitude error using data from Figure 5.7 plotted against the response from the best fitting damped spring model.

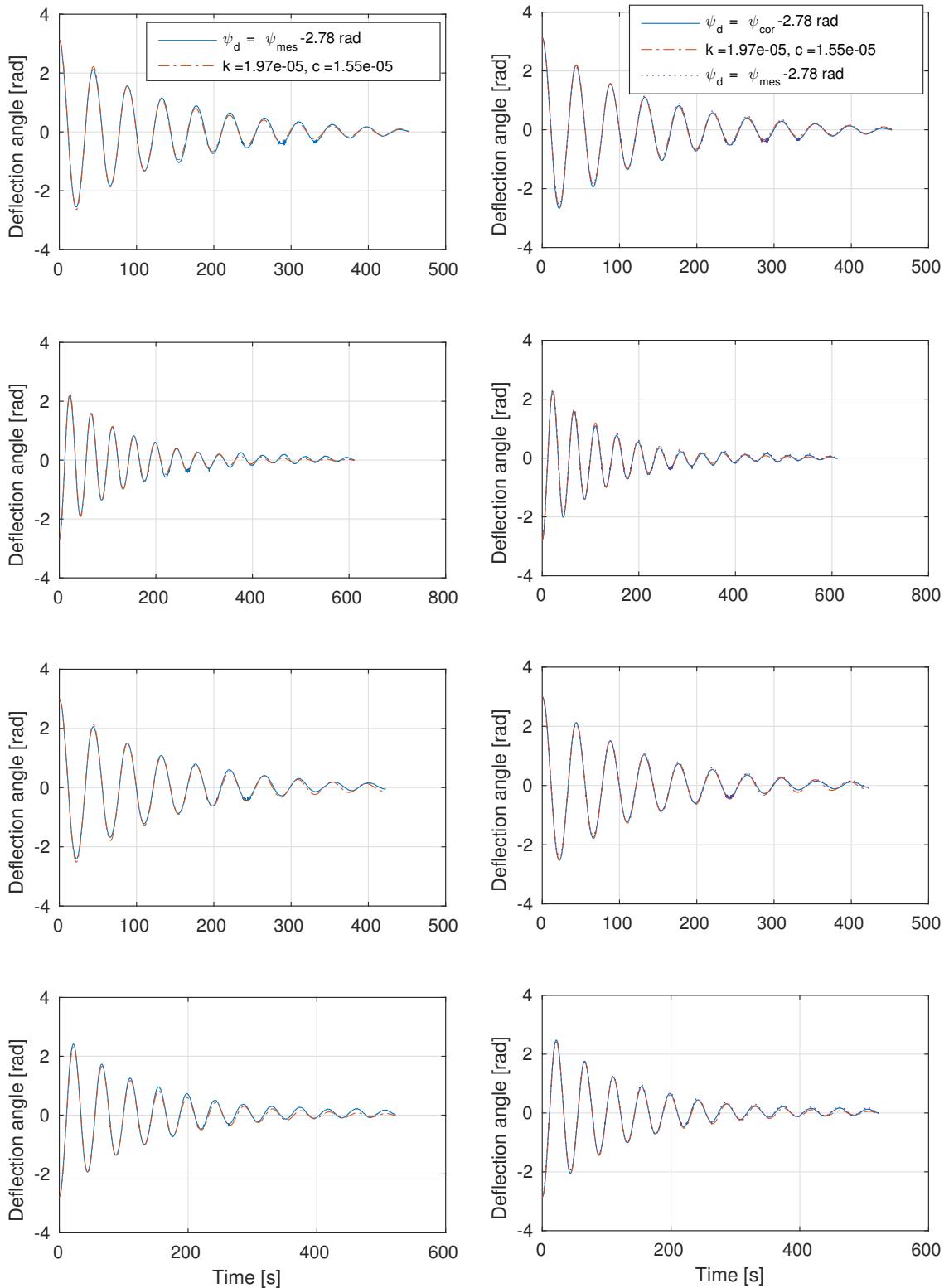


Figure 5.13: Raw measured heading from the oscillation tests about the -Z axis plotted against the response from the best fitting damped spring model.

Figure 5.14: Heading from the oscillation tests about the -Z axis that has been retroactively corrected for attitude error using data from Figure 5.8 plotted against the response from the best fitting damped spring model.

### 5.5.1. Discussion of oscillation test results

Overall, the output from the string testbed model fit very well to the measured heading responses after attitude error correction. Attitude error correction showed the greatest improvement in fit with the tests about the +Y axis. This was because a large attitude error existed between  $\psi_0$  and  $\psi_{eq}$ . Fit did not discernibly improve for the tests about the other two axes.

The tests about the +X and +Y axes did not show a perfect fit of the software model's output to all of the corrected heading curves. For the +X tests, it is primarily manifested in a change of equilibrium angle between the first and second test. This could be explained by the test rig being shifted when the test model was manually deflected. Additionally, the first test appears to demonstrate a re-excitation at around 500 s. The cause of this is unknown and it is comparable to the response reported from the MicroMAS test campaign (Figure 3.1 on Page 25). None of the other tests were conducted long enough to repeat this phenomena. The noise in the measured response in the first test conducted about the X axis is a result of  $|\psi| > \pi$ . In the tests conducted about the +Y axis, the first and third test appear to demonstrate a decrease in  $k$  at the 200 and 150 second marks respectively, the cause of this is unknown. The -Z tests produced essentially perfect results. The measured heading curves and the output curves from the software model are pretty much identical. This shows that performing the oscillation test with the satellite test model can reproduce the results produced with the Arduino test model shown in Figure 4.28 (Page 72). The test also shows that the external torques produced by the testbed can be determined very accurately.

The value of  $k$  varies slightly for each set of tests. This is most likely the result of differences in string length. The test model is suspended by a knot so variation would be unavoidable. The +X and +Y tests show identical values for  $c$  and the -Z tests showed a significantly reduced value. This strongly suggests that  $c$  is coupled to form factor, which would imply that air drag is a significant contributor.

### 5.5.2. Testbed performance

Some performance metrics of this validation testing method can be derived from the test results. Namely, we are interested in the torque levels that the test model would experience and how closely those torques would be known. Performance results are displayed in Table 5.2

#### Maximum torque

Let us first constrain the problem and assume that the test model would not rotate more than  $\pi$  rad from the equilibrium point. Any larger rotation would be unrealistic in representing mission operations. Let us also assume that the test model will not move faster than 0.4 rad/s: the highest rotation rate that the test model had during testing (first -Z test). Let's also decide on values of  $2.1 \cdot 10^{-5}$  Nm/rad and  $2.6 \cdot 10^{-5} \frac{\text{Nm}}{\text{rad/s}}$  for  $k$  and  $c$  respectively as the torque coefficients of the system, as they were the highest values measured. These values result in a maximum spring torque of  $6.6 \cdot 10^{-5}$  Nm and a maximum damping torque of  $1.04 \cdot 10^{-5}$  Nm. It is reasonable to claim that these maximum values are not additive, as the test model would be moving slowly near the rotation limit, so the maximum external torque experienced on the testbed is  $6.6 \cdot 10^{-5}$  Nm. This is roughly a fifth of the advertised maximum torque output of the reaction wheel motors ( $3 \cdot 10^{-4}$  Nm) [9]. If the limit for test model motion being dominated by the reaction wheels is defined as a factor 10 difference between the wheel's output and the external torques, then rotation would be limited to 1.41 rad from the equilibrium point. This would again change if the control law was expected to command an even lower output torque. Thinner and/or longer strings can be used to reduce  $k$  if necessary.

#### Torque uncertainty

Uncertainty in external torques on the model can be extrapolated from uncertainty in the  $k$ ,  $c$  and  $\psi_{eq}$  parameters. We can define uncertainty as an amount by which if the parameter changed, there would be a clearly worse fit of the software model output to the measured and corrected heading response. This would conservatively amount to  $1 \cdot 10^{-6}$  Nm/rad,  $1 \cdot 10^{-6} \frac{\text{Nm}}{\text{rad/s}}$  and 0.05 rad for  $k$ ,  $c$  and  $\psi_{eq}$  respectively. These values can be converted to a minimum and maximum torque uncertainty using the values for the maximum torque calculation.

The minimum torque uncertainty is defined as the torque uncertainty resulting from  $\psi_{eq}$  uncertainty. It is the only uncertainty that remains if  $\psi_d$  and  $\omega$  are 0. This is calculated by multiplying the  $\psi_{eq}$  uncertainty with  $k$ , which results in  $1.05 \cdot 10^{-6}$  Nm. Maximum torque uncertainty due to uncertainty in  $k$  is calculated by multiplying uncertainty in  $k$  with  $\pi$ , resulting in  $3.14 \cdot 10^{-6}$  Nm. Maximum Torque uncertainty due to uncertainty in  $c$  is calculated by multiplying uncertainty in  $c$  by the maximum speed of 0.4 rad/s, resulting in  $4 \cdot 10^{-7}$  Nm.

Again, maximum spring and damping torques are not additive, so the overall maximum torque uncertainty is the sum of torque uncertainty due to uncertainty in  $\psi_{eq}$  and  $k$ , resulting in  $4.19 \cdot 10^{-6}$  Nm.

Torque uncertainty will also be introduced by uncertainty in  $\psi$  and  $\omega$ . Those values have not been quantified for these tests and would require further testing to determine accurately and so were omitted from analysis. Uncertainty in  $k$  and  $c$  is also produced through uncertainty in  $[I]$ . A CAD model can be very accurate in theory, but  $[I]$  should probably be independently verified.

Table 5.2: Torque performance of the string suspension testbed derived from string characterization test results.

| Parameter                           | Value                   |
|-------------------------------------|-------------------------|
| Max. spring torque                  | $6.6 \cdot 10^{-5}$ Nm  |
| Max. damping torque                 | $1.04 \cdot 10^{-5}$ Nm |
| Max. torque                         | $6.6 \cdot 10^{-5}$ Nm  |
| Min. torque                         | 0 Nm                    |
| Max. spring torque uncertainty      | $3.14 \cdot 10^{-6}$ Nm |
| Max. damping torque uncertainty     | $4 \cdot 10^{-7}$ Nm    |
| Torque uncertainty from $\psi_{eq}$ | $1.05 \cdot 10^{-6}$ Nm |
| Max. torque uncertainty             | $4.19 \cdot 10^{-6}$ Nm |
| Min. torque uncertainty             | $1.05 \cdot 10^{-6}$ Nm |

### Torque prediction error

In an effort to understand how sensitive this validation testing method is to attitude determination performance, we can calculate the torque prediction error that would have resulted had the heading responses been used without attitude error correction. We can discuss the result of error in determining  $k$ ,  $c$  and  $\psi_{eq}$  parameters as well as the result of error in determining  $\psi_d$ :

- $k$  did not show any change after attitude error correction. This is because the resulting change of when the local maxima and minima of  $\psi_d$  occur due to attitude error correction is negligible. It can be said that determination of  $k$  is robust to the type of attitude error displayed in Figures 5.6 through 5.8.
- $c$  shows significant changes after attitude error correction, particularly in the +X tests, where  $c$  changes by  $5 \cdot 10^{-6} \frac{\text{Nm}}{\text{rad/s}}$ . Multiplying this by a maximum rotation rate of 0.4 rad/s gives a maximum damping torque prediction error of  $2 \cdot 10^{-6}$  Nm.
- A change in  $\psi_{eq}$  itself will not affect torque prediction because, so long as the model fit prioritizes the small oscillations at the end of the test (see Figure 5.11), the value determined as  $\psi_{eq}$  will still represent the equilibrium position in the current attitude mapping. Another way to see it is that the measured attitude errors are not absolute but relative, so attitude error correction of a single position out of context of any others is meaningless. If, during the attitude determination test, the zero point of the protractor was aligned with the eventual equilibrium position of the string, there would never be a change in  $\psi_{eq}$  after attitude error correction.
- The greatest source in torque prediction error is  $\psi_d$  error. The tests about the +Y axis are a very good case study for this because the heading error determination range is relatively large (0.59 rad) and  $\psi_{eq}$  is at an attitude error maximum (see  $\psi = -2.31$  rad in Figure 5.7). It represents a truly worst case scenario. Any other value of  $\psi_{eq}$  would have resulted in a lower maximum  $\psi_d$  error. In this case maximum  $\psi_d$  error is  $\sim 0.59$  rad. If  $\psi_{eq}$  were -0.5 rad, the maximum  $\psi_d$  error would be more like  $\sim 0.3$  rad. Multiplying a maximum  $\psi_d$  error of 0.59 rad to a  $k$  value of  $2.1 \cdot 10^{-5}$  Nm/rad results in a maximum torque prediction error of  $1.24 \cdot 10^{-5}$  Nm.

The other great contributor to prediction error is of course attitude determination error itself, it produces the effect of an apparent "contraction" and "dilation" of space from the point of view of the test model, which in a sense renders moot the discussion of torque prediction analysis. To observe the overall effect of attitude determination error on prediction performance, consult Figures 3.15, 3.16 (Page 40) and 5.11.

## 5.6. Verification of attitude determination with a software model

This section demonstrates how the testbed software model described in Section 4.1 (Page 55) was used to design and verify the attitude determination method. Figure 5.15 shows a simulated response of the test model under similar conditions to the first test displayed in Figure 5.13. The input parameters are listed in Table 5.3. An important distinction to make is that the simulation assumes that the reaction wheels are the only disturbance on the magnetometer (see Section 4.1.2). The type of attitude error displayed in Figures 5.6 through 5.8 is not present. The goal of this analysis is to determine attitude determination error produced by the interaction of the filtering methods with motion and acceleration. The simulation was only propagated for a single period because it is sufficient to show the full range of attitude determination performance under the conditions tested.

The results in Figure 5.15 show a maximum heading determination error of  $\sim 0.1$  rad. The largest source of heading determination error is the delay inherent in the moving average filter. The mean value of the 100 last magnetometer samples is in fact a best estimate of the ambient magnetic field 0.295 seconds before the measurement takes place. Delay is not a concern for string characterization tests because the data is analyzed after the test takes place. The situation is different when attitude control is involved because attitude determination will be required in real time. Figure 5.16 shows another set of output data from the software model with the delay of 0.295 seconds removed. Delay correction reduces the maximum heading determination error to  $\sim 0.04$  rad. The remaining attitude error is the result of effectively asynchronous gyro and magnetometer estimates, resulting in a marginally higher damping factor represented in the results than in reality. It should be noted that the situation represented in these plots is a worst case scenario, the test model is undergoing much larger rotation rates and accelerations than would be expected in a typical attitude control test.

The value of 100 as the moving average bin size was chosen somewhat arbitrarily. Time did not allow the bin size to be optimized and changed in the flight software. Optimization efforts in post-processing found that a bin size of 42 would have been optimal for the conditions listed in table 5.3, resulting in a real-time maximum heading determination error of  $\sim 0.048$  rad and  $\sim 0.022$  rad after delay correction.

Table 5.3: Parameters defining the simulation that produced the outputs displayed in Figures 5.15 and 5.16.

| Parameter                                | Value   |
|--|---|
| Sensor sample rate                       | 170 Hz  |
| Attitude determination period            | 50 ms   |
| Rotation axis in the body frame          | $[0 \ 0 \ 1]^T$                                       |
| Moving average bin size                  | 100   |
| $k$                                      | $1.97 \cdot 10^{-5}$ Nm/rad                           |
| $c$                                      | $1.55 \cdot 10^{-5}$ $\frac{\text{Nm}}{\text{rad/s}}$ |
| $\psi_{d_0}$                             | 2,8 rad.  |
| $\psi_{eq}$                              | 0 rad.  |
| $K_{PC}$                                 | 0.98  |
| Amplitude of mag. disturbance in X and Y | See table 5.1   |
| Amplitude of mag. disturbance in Z       | 700 counts  |
| Residual Z gyro bias                     | 10 counts   |
| $I$                                      | See equation 5.3                                      |
| Gyro noise RMS                           | 30 counts   |
| Gyro sensitivity error                   | 0 mdeg./count   |
| Ambient magnetic field in inertial frame | $[440 \ 0 \ -630]^T$ counts                           |
| Reaction wheel rates ( $\Omega$ )        | $[100 \ 100 \ 100]^T$ rad/s                           |

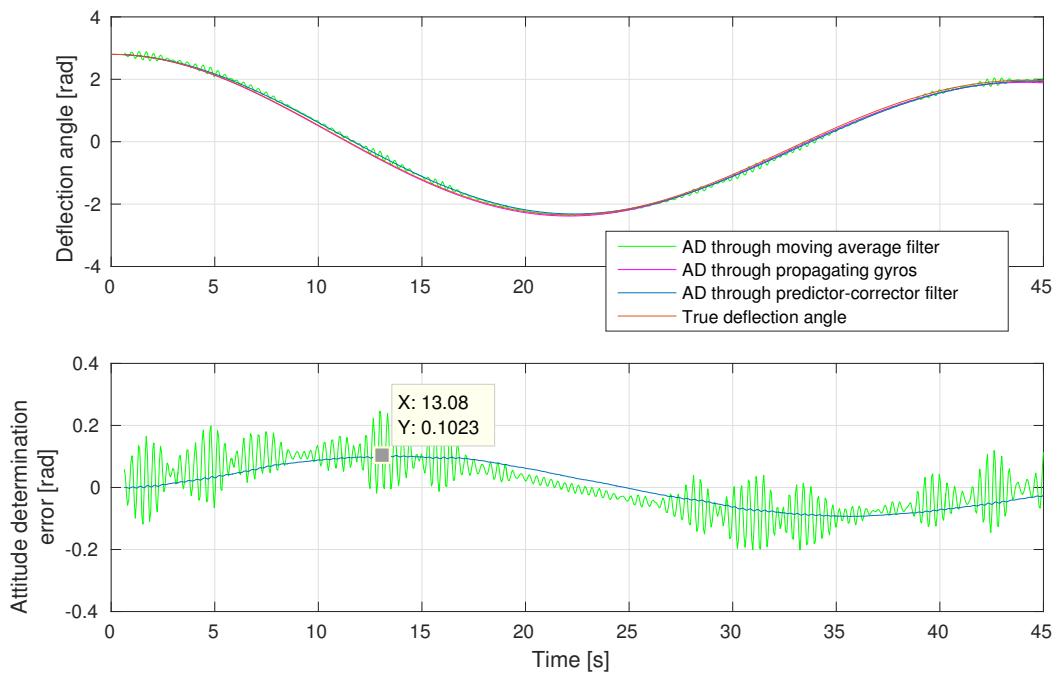


Figure 5.15: Output of the string testbed software model under the conditions specified in Table 5.3.

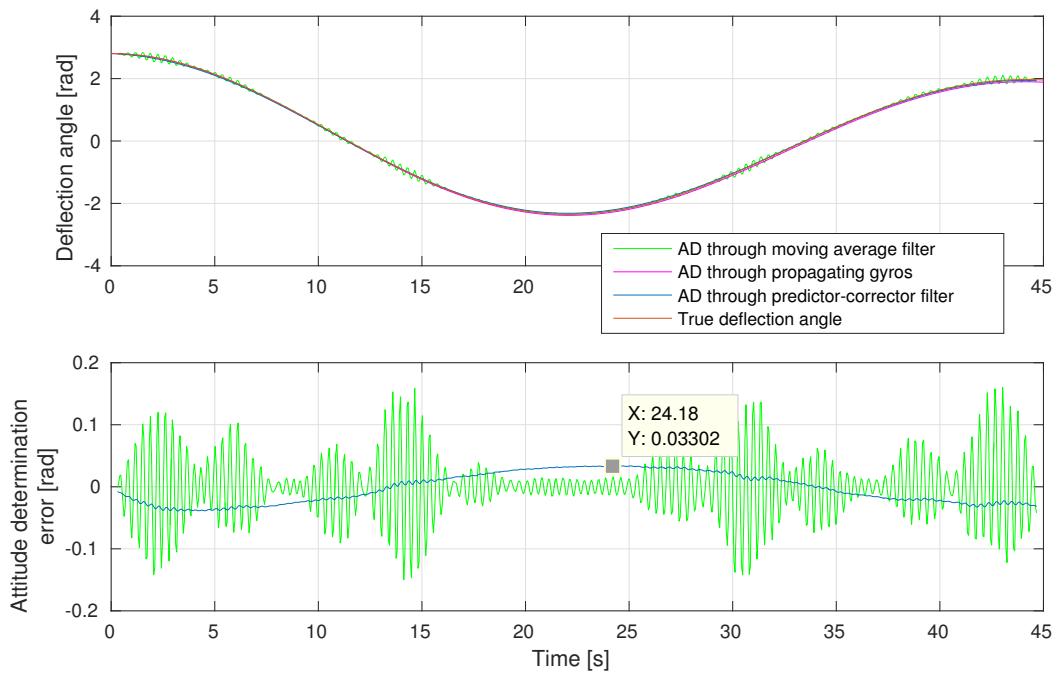


Figure 5.16: Output of the string testbed software model under the conditions specified in Table 5.3 with correction for the delay inherent in magnetometer filtering.

## 5.7. Chapter summary

This chapter presented the details of the tests conducted to prove the effectiveness of the test system developed to enable validation tests of ACS. This includes the methodology, results and discussion of each test. Due to time constraints, the tests were only conducted in their entirety one time.

The first tests determined the bias to be removed from gyro and magnetometer readings. Gyro bias was determined by recording data produced while the test model was stationary. Magnetometer bias was determined using the sphere center calculation method. Sphere center calculation required filtering of the magnetometer signals due to magnetic disturbances produced by the reaction wheels. Overall, sphere center calculation with the satellite test model was challenging to implement and had a much lower degree of certainty than with the Arduino test model.

The test model was tested for stationary heading determination performance about the +X, +Y and -Z axes. The tests showed a relatively large range of heading determination error when compared to tests conducted on the Arduino test model. The tests were repeated and showed that the mapping of true heading to heading determination error varied very little across time and repeated power cycles.

The test model then underwent oscillation tests to demonstrate that the external torque properties of the testbed could be determined using the satellite test model. The test about the -Z axis showed an essentially perfect fit of the measured heading response to output of the software model. The test about the +Y axis showed significant distortion due to attitude determination error. The heading determination error profiles determined in the attitude determination tests were used to correct for heading error through subtraction of heading error as a function of measured heading. The +Y test showed a much better fit after attitude error correction. Phenomena were observed of re-excitation of the test model as it approaches equilibrium and changes in the spring constant mid-test. The results from the oscillation tests were used to produce metrics of the overall testbed and approach in terms of knowledge of external torques on the test model.

The response from one of the tests about the -Z axis was simulated in the testbed software model to determine attitude error that would result from interaction of the filtering methods used with the angular motion and acceleration of the test model. The model output revealed that the greatest source of error was due to delay inherent in the moving average filter on magnetometer data. Removing delay showed error that results from the gyro being out of sync with the delayed magnetometer readings.



# 6

## Conclusions and future recommendations

### 6.1. Conclusion

A system engineering approach was used to develop a test system (testbed, support systems and methodology) that enabled pre-launch validation tests of PolarCube's attitude control system. This involved creating an engineering model of the satellite bus for testing (test model) that provides independent power, data handling and wireless communication to the attitude control system. The test model would be suspended by a monofilament string, providing a simulation of microgravity where torques about the local vertical would be low enough that attitude control could be performed and measured. A number of tests were proposed using the testbed that could provide metrics of the attitude control system's performance. The tests are based on a "fit-predict-fit" methodology that relies on fitting a software model of testbed dynamics to measured attitude responses. The thesis project set out to set new heights of resolution of "fit-predict-fit" methods for string suspension testbeds by improving the detail to which external torques produced by the testbed are determined. A communication interface between the test model and the test engineer was produced to provide sufficient control over the attitude control system to be able to conduct the proposed tests.

An attitude determination method was developed based on MEMS magnetometer, accelerometer and rate gyro measurements that would work within a laboratory system. Calibration was performed on magnetometer and gyro measurements to account for residual bias in the fully integrated test model. Attitude determination, communication and test methodology were first developed on an Arduino based test model that used the same sensors and radio that would eventually be used on the satellite test model. When implemented on the satellite test model, a combination of moving average and predictor-corrector filtering was implemented to remove disturbance on the magnetometers produced by the magnets in the reaction wheel motors. Heading determination accuracy when the test model is stationary was measured using a protractor. The Arduino test model showed a maximum relative heading determination error of 0.045 rad under optimal conditions and the first and only attempt at configuring attitude determination with the satellite test model showed relative errors ranging from 0.2 to 0.59 rad depending on the orientation of the test model. The mapping of measured heading to heading determination error was consistent upon repeated testing to within 0.025 rad for the Arduino test model and 0.03 rad for the satellite test model when the test model is power cycled between tests. The consistency of heading determination error allowed it to be corrected for in post-processing.

The test model was suspended by a fishing line to simulate microgravity. When suspended, the test model is constrained to rotate about the local vertical and it experiences torques about the local vertical that were hypothesized to be analogous to a rotational damped spring. External torques on the test model were determined by recording the heading response of the test model to release at a deflection angle from the equilibrium point of the string. The heading response was compared to a software model of the dynamics of the test model's behavior in the test system. This test was first conducted with the Arduino test model and comparison confirmed for the first time that string suspension testbeds do in fact behave exactly like a damped spring under the conditions tested. **The test was repeated successfully with the satellite test model and showed that the testbed torque properties can be determined to within two significant figures, a significant improvement to previously published results of similar tests.** Phenomena appeared in certain oscillation tests that have yet to be explained and will require further investigation. Improvement in string testbed external torque characterization with respect to [22] can be attributed to improved attitude determination performance, a more robust

and flexible software model of testbed dynamics, and more rigorous practices in string torque characterization. Increased attitude determination performance is particularly thanks to verification with the protractor test and heading error correction in post-processing, which is only possible thanks to the single degree of freedom constraint inherent in string suspension systems.

A software model of the test model performing attitude determination on the testbed was created as a platform to design the attitude determination algorithm used in the validation tests. Simulation of the test model's motion through very similar conditions to the physical tests was conducted to reveal attitude determination errors that would result from its motion, and thus would not appear in the protractor test. The software model revealed a maximum heading determination error of  $\sim 0.033$  rad after correcting for delay produced by the moving average filter with room to improve the filter implementation. Maximum attitude determination error occurs when angular acceleration is highest.

Overall, test results show that the test system developed for pre-launch validation of PolarCube's attitude control performance are effective to the extent to which they were tested and their use in testing would provide valuable knowledge of the attitude control system's performance. The methods and software developed in this thesis can serve as tools to enable a future test engineer to repeat these results and further fit-predict-fit test methodology. This resulting test system stands out through cost efficiency and simplicity as well as independence from COTS solutions. Through careful analysis of requirements and resources available, a path of least resistance was found to realize a complex system and achieve new heights of performance.

## 6.2. Requirements verification

Table 6.1 lists the means by which the test system design requirements (Table 3.4 on Page 49) were verified. "Will" statements are not verified. Three verification methods are listed:

- **Analysis** implies that verification was achieved through "a priori" deduction. This could be because lower-level requirements were verified, mathematical analysis was conducted or some other logical argument was made.
- **Testing** implies that a test was conducted in order to verify the requirement.
- **Design** implies that the requirement was satisfied through choice of system implementation.

Table 6.1: Requirements verification table for the test system design.

| REF                      | Verified? | Verification method | Details  |
|--------------------------|-----------|---------------------|--|
| <b>Thesis objectives</b> |           |                     |  |
| 0.ACSVT.1                | YES       | Analysis            | Requirements 1.ACSVT.1, and 1.ACSVT.6 verified.  |
| 0.ACSVT.2                | YES       | Analysis            | Requirements 1.ACSVT.7, 1.ACSVT.8 and 1.ACSVT.37 verified.   |
| <b>Requirements</b>      |           |                     |  |
| 1.ACSVT.1                | YES       | Analysis            | Requirements 1.ACSVT.2, 1.ACSVT.3, 1.ACSVT.4 and 1.ACSVT.5 verified.                                   |
| 1.ACSVT.2                | YES       | Testing             | The test model was successfully powered by EPS.  |
| 1.ACSVT.3                | YES       | Testing             | See attitude determination test results. (Page 81)   |
| 1.ACSVT.4                | YES       | Testing             | See testbed performance analysis (Page 87).  |
| 1.ACSVT.5                | YES       | Design              | The test model was suspended by a string.  |
| 1.ACSVT.6                | YES       | Testing & analysis  | Requirement 1.ACSVT.14 verified. Fit-predict-fit method will be used to determine performance metrics. |
| 1.ACSVT.7                | YES       | Analysis            | Thesis was written including procedures.   |
| 1.ACSVT.8                | YES       | Analysis            | Less than \$300 spent on dedicated test hardware. Source code is available upon request.               |
| 1.ACSVT.9                | -         | -                   | -  |
| 1.ACSVT.10               | -         | -                   | -  |
| 1.ACSVT.11               | YES       | Testing             | See testbed performance analysis. (Page 87)  |

Table 6.1: Requirements verification table for the test system design.

| REF        | Verified? | Verification method | Details  |
|------------|-----------|---------------------|--|
| 1.ACSVT.12 | YES       | Design              | 60 lb test fishing line was used to suspend the test model.  |
| 1.ACSVT.13 | YES       | Analysis            | Requirement 1.ACS.33 verified. ACS is powered by EPS.  |
| 1.ACSVT.14 | YES       | Testing             | Full ACS state vector, including attitude, reaction wheel speed, sensor measurements and configuration data was written to a removable SD card.  |
| 1.ACSVT.15 | YES       | Testing             | See attitude determination test results. (Page 81)   |
| 1.ACSVT.16 | YES       | Design              | Attitude Determination is based on measurements of local gravitational and magnetic fields.  |
| 1.ACSVT.17 | YES       | Analysis            | Accelerometer measures orientation of the local vertical and a TRIAD algorithm was used to determine rotation about the vertical.  |
| 1.ACSVT.18 | YES       | Testing & Analysis  | See attitude determination test results (Page 81) and verification of attitude determination with a software model (Page 89).  |
| 1.ACSVT.19 | YES       | Design              | ACS performs attitude determination with on-board sensors  |
| 1.ACSVT.20 | -         | -                   | -  |
| 1.ACSVT.21 | YES       | Testing             | See gyro bias determination (Page 77) and sphere center calculation results (Page 79). Functionality has been implemented in the ACS software to correct for known sensor bias values. |
| 1.ACSVT.22 | YES       | Testing             | A moving average filter is implemented on the magnetometer signal and a predictor-corrector filter is implemented on the heading signal.   |
| 1.ACSVT.23 | YES       | Design              | ACS was suspended in the center of the Mission Operations Command room.  |
| 1.ACSVT.24 | YES       | Testing             | Materials were tested for magnetization before being integrated.   |
| 1.ACSVT.25 | YES       | Testing             | A reaction wheel speed control law was written into ACS and tested.  |
| 1.ACSVT.26 | YES       | Testing             | The test engineer will send a command to switch from reaction wheel speed control to attitude control when he/she confirms the initial conditions on the downlink GUI.                 |
| 1.ACSVT.27 | -         | -                   | -  |
| 1.ACSVT.28 | YES       | Testing             | ACS' state vector is sent wirelessly to an interface computer and displayed on the downlink GUI.   |
| 1.ACSVT.29 | YES       | Testing             | An op-code can be sent to switch from reaction wheel spin rate control to attitude control.  |
| 1.ACSVT.30 | YES       | Testing             | Op-codes can be sent to set control law gains and targets.   |
| 1.ACSVT.31 | YES       | Testing             | Op-codes can be sent to set sensor bias correction values.   |
| 1.ACSVT.32 | YES       | Testing             | Op-codes can be sent to the test model wirelessly using the uplink GUI   |
| 1.ACSVT.33 | YES       | Testing             | Wireless communication was established between ACS and an interface computer via XBee and CDH.   |
| 1.ACSVT.34 | YES       | Testing             | Scripts were written to automatically parse and analyze data for the test system verification tests.   |

### **6.3. Recommendations for future work**

This section proposes areas that future engineers working on attitude control validation testing could focus on to improve performance and provide new knowledge on the topic.

#### **Perform validation tests**

No attitude control validation tests could be conducted in the test system because the attitude control system was not ready. It would certainly be interesting to attempt attitude control validation using the proposed methodology.

#### **Test with magnetorquers**

The test model used for verification of the test system did not include magnetorquers. It would be necessary to include them in order to conduct momentum dumping tests. The use of magnetorquers has a significant potential to compromise attitude determination with the magnetometer. This is not only because of the magnetic field they produce when firing, but also potentially due to the presence of the massive soft iron cores. Implementing the magnetorquers will doubtlessly provide a challenge and an interesting field of study.

#### **Accounting for attitude determination error when testing attitude control**

Retroactively correcting for heading determination error was developed after analyzing the results of the full system tests and its implementation has not been fully integrated into the overall system design. Performing the correction as it was done for the oscillation tests would not work when attitude control is active because the heading determination error would feed into the control law, further affecting the response (see Figures 3.15 and 3.16 on Page 40). Two methods were identified to implement attitude error correction on a controlled response:

- Add the determined attitude error profile into the testbed software model and simulate how the system would respond given heading determination error. This would be similar to what was done to produce Figures 3.15 and 3.16 (Page 40).
- Send the attitude error profile to the test model and have it implement feed-forward correction.

The first option would far easier to implement and should probably be attempted first.

#### **Using simulation of attitude determination in response fitting**

Figures 5.15 and 5.16 (Page 90) show that error is introduced into response fitting when interaction of the filters with motion is not accounted for. This was not done for the results in this thesis for practical reasons, but would ideally be included eventually. It becomes especially relevant when attitude control is being tested because it will no longer be possible to ignore the effect of delay in magnetometer readings and delay will feed back into the control law, changing the response.

#### **Apparent excitation at low deflection angles**

The results of the first oscillation test about the +X axis show a resemblance to the results in Figure 3.1 (Page 25). It was the only test that was conducted for longer than 600 seconds. This might suggest that the same phenomenon of excitation of the test model as it approaches rest is being observed in both results. Unfortunately, there was no opportunity to repeat the tests and record for longer periods after analysis revealed this phenomenon. The results in Figures 4.28 (Page 72) and 5.14 (Page 86) undoubtedly prove that the damped spring hypothesis is true for high amplitude motion, but the results of the thesis have not been able to prove that the test model can come to rest normally. This certainly merits closer investigation.

#### **Improve sphere center correction**

The methods developed for magnetic bias correction of the test model (Page 58) showed a great deal of room for improvement when compared to the results that could be achieved with the Arduino test model (Page 79). Attempts to improve the techniques are expected to be interesting, challenging and useful.

#### **Conduct oscillation tests in a vacuum**

Comparison of the damping constants in the oscillation tests conducted on the test model (Page 83) indicate that damping is strongly driven by aerodynamic drag. It would be interesting to conduct oscillation tests in a vacuum chamber to see how the system behaves when drag is no longer a factor.

### **Simulate torque output error**

The testbed software model currently assumes that the torques commanded by the control law would be flawlessly sent to the reaction wheels. The reaction wheel output torque tests show that they are currently far from being able to do so and they are unlikely to eventually be able to do it perfectly in any case. It would be interesting to include a model of the reaction wheel output torque error to the testbed software model to get a closer approximation of the behavior of the attitude control system. This would first require that torque imparted to the reaction wheel be determined as a function of the commanded torque and speed of the reaction wheel.

### **Improved attitude filtering**

The moving average and predictor-corrector filters implemented on attitude determination (Section 4.2.5 on Page 60) are rudimentary and a number of more sophisticated implementations are available. This could include the use of a true Kalman filter. The overall goal would be to reduce the kinds of attitude determination error presented in Figures 5.15 and 5.16.

### **Measuring string torque directly with reaction wheels**

String torque characterization with the oscillation test could be verified by recording reaction wheel speeds while the test model is holding attitude at a non-zero deflection angle. The acceleration of the parallel wheel while the test model is stationary will indicate the spring torque at that deflection angle. This should reduce external torque uncertainty due to uncertainty of  $[I]$  of the test model.

### **Improve suspension rigging**

The current suspension rigging design (see Figure 4.24 on Page 70) is elegant in its simplicity and effectiveness, but it has continued to show occasional shifting when the test model is handled. The tape solution in Figure 4.25 was not very effective. Effective improvements on this design would reduce a lot of risk, especially if multiple validation tests would rely on the results of a single oscillation test.

The rigging would also be improved by a means to keep the length of the string constant when the test model is removed and replaced, like the hook system in Figure 4.29 (Page 73). This might remove the need to repeat string torque characterization every time the test model is suspended.

### **Keep $\psi_{eq}$ near 0**

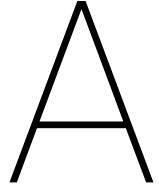
The oscillation tests about the +X axis were partially compromised by the fact that the test model kept crossing  $\psi = \pm\pi$  near the equilibrium point. This produced unappealing data and could be avoided making sure that the test model is suspended such that  $\psi_{eq}$  is near 0. The noisy attitude response could also be a result of how the transition interacts with the attitude determination algorithm and that might warrant looking into.



# Bibliography

- [1] OAV Air Bearings. OAV Roller Air Bearing. URL <http://www.oavco.com/rollerairbearing.html>.
- [2] Blue Canyon Technologies. XACT Data Sheet. URL [http://bluecanyontech.com/wp-content/uploads/2015/05/XACT-Data-Sheet{\\_}1.0.pdf](http://bluecanyontech.com/wp-content/uploads/2015/05/XACT-Data-Sheet{_}1.0.pdf).
- [3] Colorado Space Grant Consortium. Guidelines for Attitude Determination and Control Subsystem Design at Space Grant. Technical Report 303, Boulder CO, 2014.
- [4] Michael Deiml, Albin J Gasiewski, David W Gallaher, and Brian T Sanders. PolarCube: A 3U CubeSat For Remote Sensing Using The 118 GHz Oxygen Emission Band. In *65th International Astronautical Congress*, pages 1–8, Toronto, 2014. IAF.
- [5] Delft University of Technology. Delfi Space Program, . URL <http://www.delfispace.nl/general/delfi-space-program>.
- [6] Delft University of Technology. Delfi-n3Xt Attitude Determination & Control Subsystem, . URL <http://www.delfispace.nl/delfi-n3xt/attitude-determination-and-control-subsystem>.
- [7] Steven Engelen. The Road to OLFAR - a Roadmap to Interferometric Long-Wavelength radio Astronomy Using Miniaturized Distributed Space Systems. In *64th International Astronautical Congress*, pages 1–7, Beijing China, 2013.
- [8] Giancarmine Fasano, Giancarlo Rufino, Domenico Accardo, and Michele Grassi. Satellite angular velocity estimation based on star images and optical flow techniques. *Sensors (Basel, Switzerland)*, 13:12771–12793, 2013. ISSN 14248220. doi: 10.3390/s131012771.
- [9] Faulhaber. Series 0620...B Edition 2012-2013. URL [http://www.nrc.com.tw/CAT/Faulhaber/0620{\\_}B.pdf](http://www.nrc.com.tw/CAT/Faulhaber/0620{_}B.pdf).
- [10] Christoffel Johannes Groenewald. Attitude Determination and Control System for EyasSAT for Hardware In the Loop Application. Master's thesis, Stellenbosch University, 2014.
- [11] Manon E Grugel. Modeling the viscous torque acting on a rotating object, 1998.
- [12] Erik Hogan and Hanspeter Schaub. Three-Axis Attitude Control Using Redundant Reaction Wheels with Continous Momentum Dumping. In *AAS/AIAA Spaceflight Mechanics Meeting*, Kauai, Hawaii, 2013. AAS Publications Offic. doi: 10.2514/1.G000812.
- [13] Honeywell. HMC5883l 3-Axis Digital Compass IC, 2013. URL [https://www.adafruit.com/datasheets/HMC5883L{\\_}3-Axis{\\_}Digital{\\_}Compass{\\_}IC.pdf](https://www.adafruit.com/datasheets/HMC5883L{_}3-Axis{_}Digital{_}Compass{_}IC.pdf).
- [14] Junquan Li, Mark Post, Thomas Wright, and Regina Lee. Design of Attitude Control Systems for. *Journal of Control Science and Engineering*, 2013, 2013. ISSN 1687-5249. doi: 10.1155/2013/657182.
- [15] James Paul Mason, Thomas N Woods, Amir Caspi, Phillip C. Chamberlin, Christopher Moore, Andrew R Jones, Rick Kohnert, Xinlin Li, Scott E Palo, and Stanley C. Solomon. Miniature X-Ray Solar Spectrometer (MinXSS) - A Science-Oriented, University 3U CubeSat. In *29th Annual AIAA/USU Conference on Small Satellites*, Logan UT, 2015. ISBN 3035466351.
- [16] David M Meissner. a Three Degrees of Freedom Test Bed for Nanosatellite and Cubesat Attitude Dynamics, Determination, and Control. Master's thesis, Naval Postgraduate School, 2009.
- [17] New Way Air Bearings. Vacuum pre-loaded spherical air bearings. URL <http://www.newwayairbearings.com/products/spherical-air-bearings/custom-vacuum>.

- [18] Ryan Nugent, Riki Munakata, Alexander Chin, Roland Coelho, and Graduate Student. The CubeSat: The Picosatellite Standard for Research and Education, 2008.
- [19] Óscar R. Polo, Segundo Esteban, Lorenzo Cercos, Pablo Parra, and Manuel Angulo. End-to-end validation process for the INTA-Nanosat-1B Attitude Control System. *Acta Astronautica*, 93:94–105, jan 2014. ISSN 004945765. doi: 10.1016/j.actaastro.2013.07.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S00494576513002336>.
- [20] F M Poppenk, R Amini, and G F Brouwer. Design And Application Of a Helmholtz Cage For Testing Nano-Satellites. In *ESA-ESTEC 6th International Symposium on Environmental Testing for Space Programmes.*, number Figure 1, Noordwijk, 2004.
- [21] J. Puig-Suari, C. Turner, and W. Ahlgren. Development of the standard CubeSat deployer and a CubeSat class\nPicoSatellite, 2001. ISSN 1095323X.
- [22] Meghan Kathleen Quadrino, David Miller, and Kerri Cahoy. Testing the Attitude Determination and Control of a CubeSat with Hardware-in-the-Loop. Master's thesis, Massachusetts Institute of Technology, 2014.
- [23] Christian Raschke, Stephan Roemer, and Karsten Grossekatthoefer. Test bed for verification of attitude control system.
- [24] Devon S Sanders, Daniel L Heater, Steven R Peeples, and James K Sykes. Pushing the Limits of Cubesat Attitude Control : A Ground Demonstration. In *AIAA/USU Conference on Small Satellites*, Salt Lake City UT, 2013.
- [25] Hanspeter Schaub and John L Junkins. *Analytical mechanics of space systems*. AIAA, second edition, 2003. ISBN 1600867219. doi: 10.2514/4.861550.
- [26] Jana L. Schwartz, 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. ISSN 0731-5090. doi: 10.2514/2.5085.
- [27] Specialty Components. Spherical air bearings. URL <http://www.specialtycomponents.com/Products/sra850-r45/>.
- [28] ST Microelectronics. A3G4250D MEMS motion sensor. URL <http://www.st.com/web/en/resource/technical/document/datasheet/DM00047823.pdf>.
- [29] The CubeSat Project. CubeSat Project Mission Statement. URL <http://www.cubesat.org/index.php/about-us/mission-statement>.



# Reaction Wheel performance analysis

The goal of this test was to understand the relationship between the reference voltage sent to the driver chip of a motor and the actual output torque as a function of reaction wheel speed. Initial tests had shown that no torque was produced on the wheel when a zero reference voltage was supplied and that a torque is produced when a non-zero reference voltage is supplied (max voltage and half of max voltage). Initial tests also confirmed that the direction of torquing could be controlled. It had also been determined that all of the wheels had a maximum speed that was independent of the value of the reference voltage supplied.

These tests were the first to use speed measurement logs and measure the acceleration of the wheels. They could not have been conducted any earlier because the support systems necessary (GUI, ACS main board, driver boards, test-driven ACS software) were not ready. The software necessary for this test would not have existed were it not for the ACS validation testing project. Although the results of the test (That there was no means of controlling reaction wheel output torque) put an unfortunate stop to attempts to validate Polar-Cube's attitude control system, they in themselves constitute the most significant result for the PolarCube mission to come out of the ACS validation project as of writing this document.

The test starts with the reaction wheels at rest and a reference voltage input of 0. The maximum reference voltage (3V3) is then introduced to all three wheels and they are allowed to accelerate to their maximum speed. This provides the data necessary to determine the motor output torque as a function of reaction wheel speed for the reference voltage supplied. Once the reaction wheels reach their maximum speed, a reference voltage of 0 is supplied to the motors. The wheels are then allowed to decelerate due to friction in the motor, this data is also collected and used to determine friction torque as a function of reaction wheel speed for the motors. When the wheels have stopped, a reference voltage of 7/8 of the maximum reference voltage is introduced and the wheels are allowed to accelerate. This is repeated until 1/8 of the maximum value. The goal was to have a single response curve for each of the reference voltages and 7 curves that can be used to determine a friction profile.

## A.1. Output torque

Figure A.1 shows the reaction wheel speed ( $\Omega$ ) responses from rest to all of the reference voltages supplied. The figure makes it very clear that the performance of the motor was identical for all of the reference voltages supplied. Commanding of the reference voltage using the GUI was verified before and after the test took place.

Figure A.2 shows motor output torque ( $\tau_{\text{out}}$ ) of the reaction wheels as a function of reaction wheel speed as derived from the raw data shown in Figure A.1. A third degree polynomial was fit to the speed response curves and the derivative of the fitted curve was plotted against the curve's output for a set of time values representing the time elapsed in Figure A.1. The result closely approximates a linear curve. The deviations from a linear curve, particularly at max speeds, are mostly a mathematical artifact of the method used. Table A.1 lists the parameters that define a linear best fit to the data from all of the curves in Figure A.2 as defined by the following formula:

$$\tau(\Omega) = \tau(0) + \Omega \cdot \frac{d}{d\Omega} \tau_{\text{out}} \quad (\text{A.1})$$

Table A.1: Parameters defining the best linear fit of reaction wheel output torque as a function of reaction wheel spin rate from the data in Figure A.2.

| Reaction<br>wheel axis | $\tau_{\text{out}}(0)$  | 95% confidence          | $\frac{d}{d\Omega}\tau_{\text{out}}$                 | 95% confidence          |
|------------------------|-------------------------|-------------------------|--|-------------------------|
| -X                     | $3.69 \cdot 10^{-5}$ Nm | $\pm 2.8 \cdot 10^{-7}$ | $-2.09 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 2.4 \cdot 10^{-9}$ |
| Y                      | $2.68 \cdot 10^{-5}$ Nm | $\pm 2 \cdot 10^{-7}$   | $-1.92 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 2.2 \cdot 10^{-9}$ |
| Z                      | $2.33 \cdot 10^{-5}$ Nm | $\pm 4.9 \cdot 10^{-7}$ | $-2.16 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 6.6 \cdot 10^{-9}$ |

## A.2. Friction torque

Figure A.3 shows the reaction wheel speed responses to the introduction of a 0 reference voltage while they were spinning at the maximum speed that their motors could sustain. The maximum speed reached by each motor was slightly different for each test, so the data was truncated to make sure that all of the curves began at the same speed. The responses are all identical, as would be expected given they were all recorded under the same conditions.

Figure A.4 shows friction torque of the motors as a function of reaction wheel speed as derived from the data presented in Figure A.3. The same methods were used as in Figure A.2. The results also show a more or less linear function of torque to speed. Table A.2 lists the parameters that define a linear best fit to the data from all of the curves in Figure A.4.

Table A.2: Parameters defining the best linear fit of reaction wheel friction torque as a function of reaction wheel spin rate from the data in Figure A.4.

| Reaction<br>wheel axis | $\tau_{\text{fr}}(0)$    | 95% confidence          | $\frac{d}{d\Omega}\tau_{\text{fr}}$                  | 95% confidence          |
|------------------------|--------------------------|-------------------------|--|-------------------------|
| -X                     | $-1.76 \cdot 10^{-5}$ Nm | $\pm 2.2 \cdot 10^{-6}$ | $-2.47 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 2.7 \cdot 10^{-9}$ |
| Y                      | $-1.45 \cdot 10^{-5}$ Nm | $\pm 2.2 \cdot 10^{-6}$ | $-2.22 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 3.5 \cdot 10^{-9}$ |
| Z                      | $-2.08 \cdot 10^{-5}$ Nm | $\pm 4.5 \cdot 10^{-6}$ | $-2.52 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ | $\pm 9.1 \cdot 10^{-9}$ |

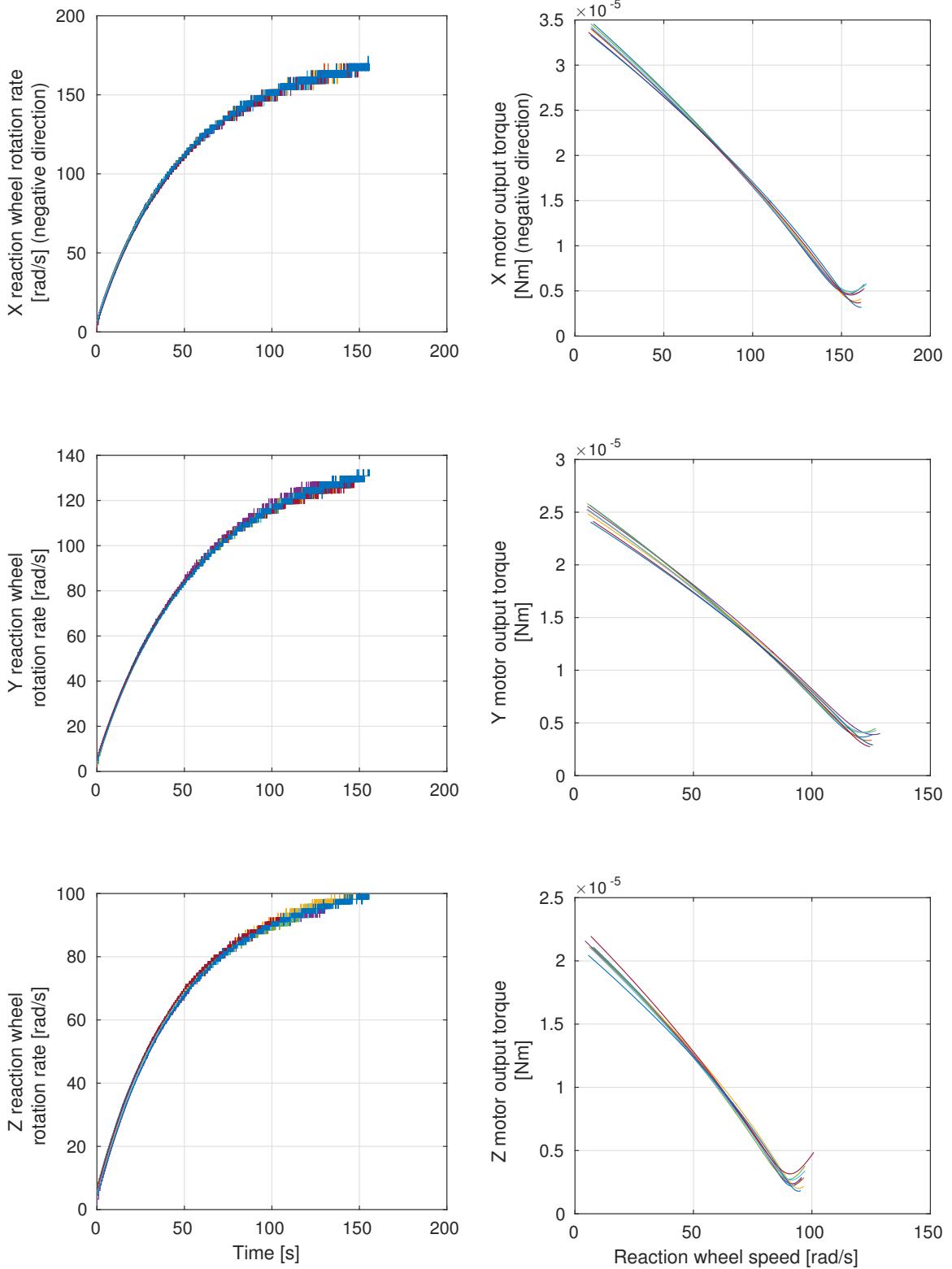


Figure A.1: Measured speed of the reaction wheels in response to a variety of torque commands from rest. The reference voltages received by the driver chip were equal to  $3V_3 \cdot n/8$  where  $n = 8, 7, \dots, 1$ .

Figure A.2: Plot of motor output torque experienced by the reaction wheels as a function of reaction wheel speed.

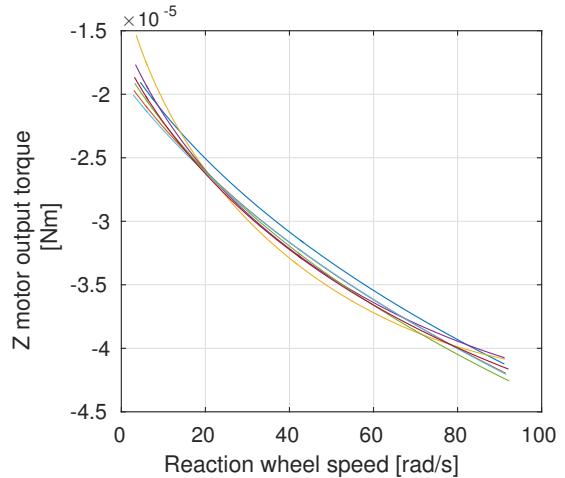
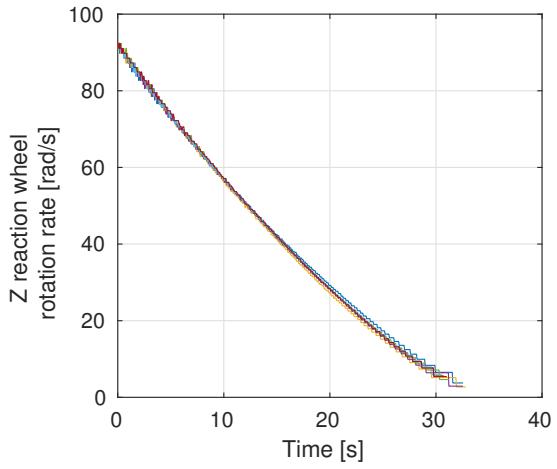
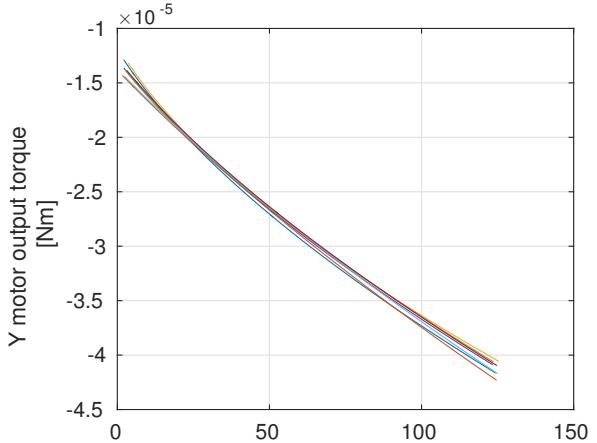
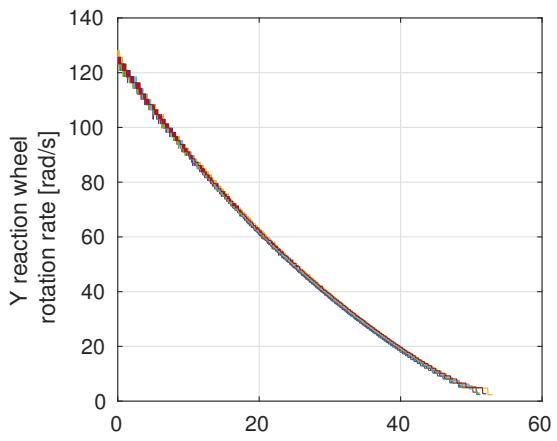
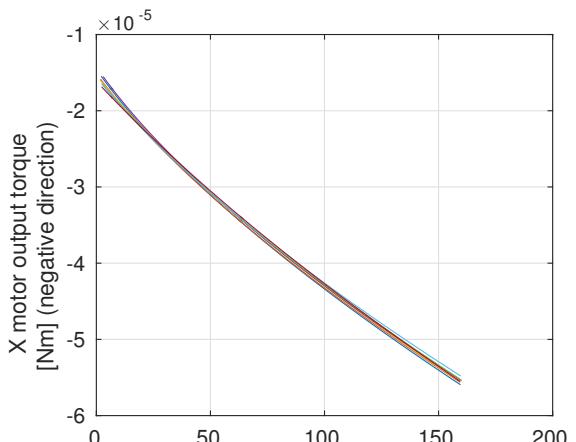
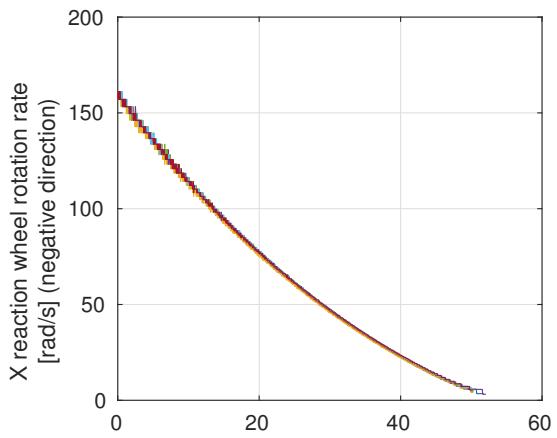


Figure A.3: Measured speed of the reaction wheels in response to the introduction of a zero reference voltage at their maximum speed. This data was collected in between the measurements taken for Figure A.1.

Figure A.4: Plot of friction torque experienced by the reaction wheels as a function of reaction wheel speed.

### A.3. Net torque

Figure A.5 is an attempt to determine the torque being generated by the reaction wheel motors. The blue line is the net torque of the system where  $\tau_{\text{net}} = \tau_{\text{out}} - \tau_{\text{fr}}$ . In this representation  $\tau_{\text{out}}$  and  $\tau_{\text{fr}}$  are defined by Equation A.1 using the data in tables A.1 and A.2. The red dashed line shows  $\tau_{\text{out}}$  for comparison.  $\tau_{\text{net}}$  is nearly constant, all of the wheels exhibit a slight upward slope of  $\tau_{\text{net}}$  as a function of reaction wheel speed. There are a few potential explanations for this effect:

- This could be an artifact of the numerical methods used to determine torque.
- Torque generated by the motor in fact increases as a function of reaction wheel speed.
- Retarding torque properties change when the motor is turned off.

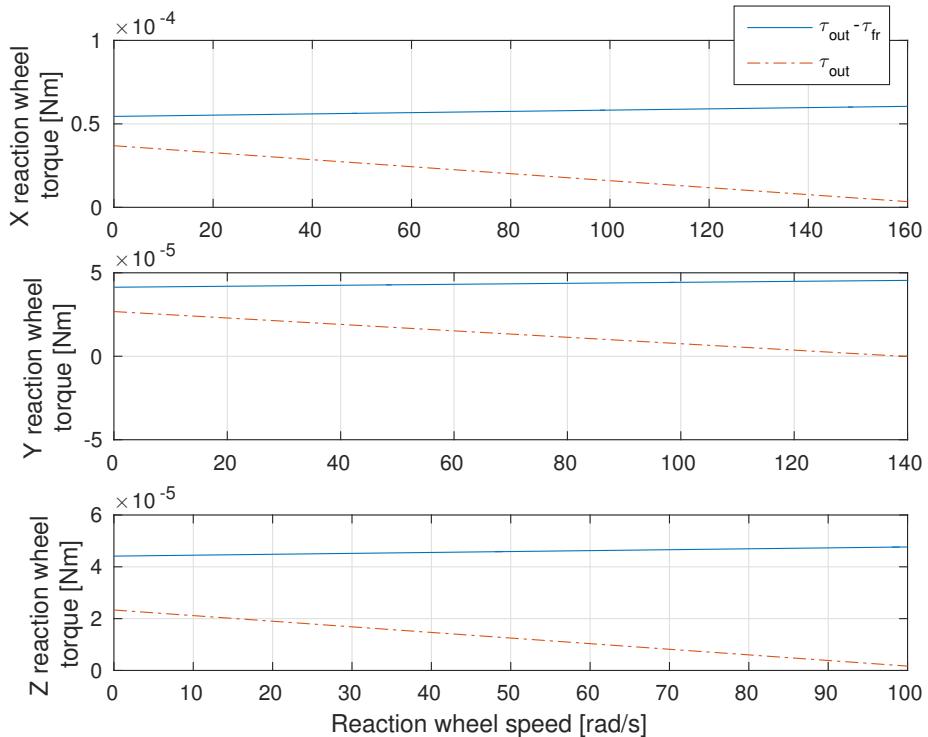


Figure A.5: Linear fit of the motor output torque experienced by the reaction wheels (Figure A.2) added to a linear fit of the friction torque experienced while motor is turned off and the wheels are decelerating (Figure A.4).

### A.4. Comparison to motor data sheet

The results of this test can be compared to some of the performance data cited in the motor's datasheet [9].

#### A.4.1. Static torque

The datasheet cites a "static torque" of  $-2.3 \cdot 10^{-5} \text{ Nm}$ . This value is close to the  $\tau_{\text{fr}}(0)$  values listed in Table A.2. This verifies that the methods used to calculate torque are likely correct and that friction is the dominant retarding torque on the wheels at low speeds.

#### A.4.2. Torque slope

The motor datasheet cites a n/M ( $\Omega/\tau$ ) curve of 68054 rpm/mNm, which converts to  $-1.4 \cdot 10^{-7} \frac{\text{Nm}}{\text{rad/s}}$ . This value is close to the measured values of  $\frac{d}{d\Omega} \tau_{\text{out}}$  listed in table A.1, providing further verification that the measurement methods are sound. The datasheet cites a dynamic friction torque of  $-1.02 \cdot 10^{-6} \text{ mNm/rpm}$ . This converts to  $-9.6 \cdot 10^{-9} \frac{\text{Nm}}{\text{rad/s}}$ , very different to the values of  $\frac{d}{d\Omega} \tau_{\text{fr}}$  listed in Table A.2.

## A.5. Discussion

This test proved effective in measuring torque experienced by the reaction wheels and verified some of the motor's cited performance characteristics. The test also exposed a critical flaw in the implementation of the reaction wheel hardware. Later testing revealed that torque could not be controlled because the wrong motors were being used. Component testing of the correct motors showed that current draw varied with the reference voltage whereas it did not with the motors used in this test. The next set of reaction wheels will be built with the correct motors and this test will serve to determine their performance and eventually verify their functionality.

# B

## Source Code

### B.1. Testbed Software Model

```
% StringTestbedModel.m
% author: Maxim Clarke
% email: maxim.clarke.thesis@gmail.com

% StringTestbedModel.m is a dynamic model of test model motion on the
% string suspension testbed. The test model is subject to spring toreqs and
% damping torques from the string as well as gyroscopic torques from the
% reaction wheels. The test model is slao constrained to motion about the
% local vertical (the rotation axis) torques that are not parallel to the
% roation axis do not affect test model motion.

% INPUT PARAMETERS

% Body axis about which rotation occurs ('x', 'y' or 'z')
BFRotationAxis = [0; 0; -1];
% Normalize the roation axis in the body frame
BFRotationAxis = BFRotationAxis/norm(BFRotationAxis);

t= 45; % (length of propagation)

SampleBinPeriod = 0.588; % seconds
SampleFrequency = 170; % Hertz

% Define how often attitude control is performed with a counter
ControlPeriod = 0.05; % seconds
ControlStepSize = ControlPeriod*SampleFrequency;
ControlCounter = round(rand*10);
% This variable holds the current reaction wheel torque command
ControlTorque = [0; 0; 0];

% Define how often attitude determination is performed with a counter
DeterminationPeriod = 0.05;
DeterminationStepSize = DeterminationPeriod*SampleFrequency;
DeterminationCounter = round(rand*10);
```

```

% Initial deflection from equilibrium angle
InitialTheta = 2.8;

% These variables hold the current determined attitude values
ThetaPC = InitialTheta;
ThetaMA = nan;
PredictedTheta = InitialTheta;

% Initial Speeds of the reaction wheels
InitialRWSpeeds = [100+(rand*5-2.5); 100+(rand*5-2.5); 100+(rand*5-2.5)];
InitialBodyRate = 0; % initial rotation rate about the rotation axis

% string properties
SpringConstant= 0.0000197; % Nm/rad
DampingConstant = 0.0000155; %Nm/(rad/s)

% Heading that corresponds to the equilibrium angle of the string(rad)
EquilibriumAngle = 0;

% Confidence in gyro propagation in predictor-corrector filter
PredictorCorrectorGain = 0.98;

% Target deflection from equilibrium angle
% (defining it as NaN turns control off)
TargetTheta = nan;

% Control law Gains
K = 0.0002; % Potential gain
P = diag([0.003; 0.003; 0.0005]); % Derivative gains
Ki = diag([20; 20; 15]);% Integral gains

% Ambient magnetic field vector in Earth-reference
AmbientMagnetic = [440 0 -630]';

% Measured influence of the reaction wheels on magnetometer measurement
AmplitudeXonX = 244;
AmplitudeXonY = 267;
AmplitudeXonZ = 213;
AmplitudeYonX = 500;
AmplitudeYonY = 770;
AmplitudeYonZ = 349;
AmplitudeZonX = 700;
AmplitudeZonY = 700;
AmplitudeZonZ = 700;

%Gyro Properties
GyroSensitivity = 8.75; %mdps/count
Gyro1SensitivityError = 0; %mdps/count
Gyro2SensitivityError = 0; %mdps/count
Gyro1ZeroRateLevel = [40; 74; 10]; %counts
Gyro2ZeroRateLevel = [-20; 35; 10]; %counts
Gyro1NoiseRMS = [24; 25; 30]; %counts
Gyro2NoiseRMS = [30; 30; 30]; %counts

% Moment of inertia of the reaction wheels
J_RW=1.12e-5;

```

```

% Moment of inertia of the Test model body (from CAD)
IB=[0.002106 -1.1e-5 -0.00012;
    -1.1e-5 0.00214 1.58e-5;
    -0.00012 1.58e-5 0.000975];

% SETUP

% Find B*
% Empty vectors defining B* in B
UpAxis = zeros(3,1);
ForwardAxis = UpAxis;

% Find b3
[~, UpAxisIndex] = max(abs(BFRotationAxis));
UpAxis(UpAxisIndex) = 1*sign(BFRotationAxis(UpAxisIndex));

% Find b1
if UpAxisIndex == 1
    ForwardAxisIndex = 2;
else
    ForwardAxisIndex = 1;
end
ForwardAxis(ForwardAxisIndex) = 1;

% Find b2
LeftAxis = cross(UpAxis, ForwardAxis);
[~, LeftAxisIndex] = max(abs(LeftAxis));

% DCBBprime = B* in terms of B
DCMBBprime = [ForwardAxis'; LeftAxis'; UpAxis']';

% Find pitch and roll with TRIAD (will not change through propagation)
ForwardCrossGrav = cross(ForwardAxis, -BFRotationAxis);

Pitch = -acos(norm(ForwardCrossGrav))*...
    sign(BFRotationAxis(ForwardAxisIndex));

ForwardCrossGrav = ForwardCrossGrav/norm(ForwardCrossGrav);

Roll = -acos(dot(ForwardCrossGrav, LeftAxis))*...
    sign(dot(ForwardCrossGrav, UpAxis));

% Pitch and roll in DCM
RollDCM = EulerDCM(1, Roll);
PitchDCM = EulerDCM(2, Pitch);

% Sensor Sampling Properties
SamplePeriod = 1/SampleFrequency; % seconds
% # of samples per measurement
SamplesPerMeasurement = round(SampleFrequency*SampleBinPeriod);
SampleCount = 0;

% timestep at the sample frequency of the magnetometer

```

```

dt = 1/SampleFrequency;
simtime = dt:dt:t; % Timestamp of the simulation
outputlength = t/dt; % Useful for defining empty vectors

% Empty vectors

% Test model body motion
omegaBB = zeros(3,outputlength); % Body rotation rates in the body frame
omegaRW = zeros(3,outputlength); % Reaction wheel speeds
MRP = zeros(3,outputlength); % Attitude in Modified Rodriguez parameters
ErrorMRP = zeros(3,outputlength); % MRP representing attitude error
theta = zeros(1,outputlength); % Deflection angle

% Torques
Tspring = zeros(3,outputlength);
Tdamping = zeros(3,outputlength);
Tcontrol = zeros(3,outputlength);
torques = zeros(3,outputlength);

% initial conditions

% Initial body rate will be about the rotation axis
omegaBB(:,1) = BFRotationAxis*InitialBodyRate;
% Initial rates of the reaction wheels
omegaRW(:,1) = InitialRWSpeeds;
% Initial MRP WRT the equilibrium position
MRP(:,1) = BFRotationAxis*tan(InitialTheta/4);
theta(:,1) = InitialTheta;
ADnumber = 0;

% Control Law parameters
targetMRP = BFRotationAxis*tan(TargetTheta/4);
sigInt = [0; 0; 0];
target_omega = [0; 0; 0];
target_omegaDot = [0; 0; 0];
d_omega0 = omegaBB(:,1) - target_omega;

% Empty vectors for magnetometer measurements
MeasuredMag = zeros(3,outputlength);
MagnetometerError = [0; 0; 0];

% Empty vectors for determined attitude
MeasuredThetaPeriodic = zeros(1,outputlength);
MeasuredThetaMovingAverage = zeros(1,outputlength);
MeasuredThetaInstantaneous= zeros(1,outputlength);
MeasuredThetaGyro= zeros(1,outputlength);
MeasuredThetaPredictorCorrector= zeros(1,outputlength);

% Initial conditions for determined attitude
MeasuredThetaPeriodic(1) = nan;
MeasuredThetaMovingAverage(1) = nan;
MeasuredThetaGyro(1)= InitialTheta;
MeasuredThetaPredictorCorrector(1)= InitialTheta;

% initial angular position of the reaction wheels

```

```

RWangles = rand(3,1)*2*pi;
% Wait bar indicates progress
progressbar;

% Magnetic field as perceived in the body frame
BodyAxisMag = zeros(3,outputlength);
BodyAxisMag(:,1) = [nan; nan; nan];

% COMPUTE

for i=2:length(simtime)
    progressbar(i/length(simtime))

% PROPAGATE DYNAMICS

    % Rigid body dynamics
    % Runge-Kutta 4 integrator is used but perhaps not necessary
    [omegaBB(:,i),MRP(:,i)] = TestbedRK4integrate(dt,omegaBB(:,i-1),...
        MRP(:,i-1),omegaRW(:,i-1),torques(:,i-1),IB,J_RW,BFRotationAxis);

    % Integrate reaction wheel speed
    omegaRW(:,i) = omegaRW(:,i-1)+dt/J_RW*Tcontrol(:,i-1);

% DETERMINE ATTITUDE

    % Update deflection angle
    theta(i) = 4*atan(norm(MRP(:,i)))*sign(dot(MRP(:,i),BFRotationAxis));
    Heading = EquilibriumAngle+theta(i);
    HeadingDCM = EulerDCM(3, Heading);
    % Convert heading to a rotation matrix
    DCMBE = DCMBBprime * RollDCM * PitchDCM' * HeadingDCM' ;

    % Find ambient field vectors in body frame
    BodyAxisMag(:,i) = DCMBE*AmbientMagnetic;

    % Propagate position of reaction wheels
    RWangles = RWangles+(dt+(rand-rand)*dt/10)*omegaRW(:,i-1);

    % Calculate magnetometer error by sampling a signal composed of 3
    % sinusoids.
    MagnetometerError(1) = AmplitudeXonX *sin(RWangles(1))+...
        AmplitudeYonX*sin(RWangles(2))+...
        AmplitudeZonX*sin(RWangles(3));
    MagnetometerError(2) = AmplitudeXonY *sin(RWangles(1))+...
        AmplitudeYonY*sin(RWangles(2))+...
        AmplitudeZonY*sin(RWangles(3));
    MagnetometerError(3) = AmplitudeXonZ *sin(RWangles(1))+...
        AmplitudeYonZ*sin(RWangles(2))+...
        AmplitudeZonZ*sin(RWangles(3));

    % Add error to the magnetometer measurement
    MeasuredMag(:,i) = BodyAxisMag(:,i)+MagnetometerError;

    % Gyro output

```

```

% Simulate output
Gyro1Out = omegaBB(:,i)/3.141592*180*1000/(GyroSensitivity+...
    Gyro1SensitivityError)+ Gyro1ZeroRateLevel + Gyro1NoiseRMS.*randn(3,1);
Gyro2Out = omegaBB(:,i)/3.141592*180*1000/(GyroSensitivity+...
    Gyro2SensitivityError)+ Gyro2ZeroRateLevel + Gyro2NoiseRMS.*randn(3,1);

MeanGyroOut = (Gyro1Out+Gyro2Out)/2;

% Calculate rate from output
MeasuredOmegaAboutAxis = dot(BFRotationAxis,MeanGyroOut)/1000/...
    180*3.141592*GyroSensitivity;

% Direct integration of Gyro output
MeasuredThetaGyro(i) = MeasuredThetaGyro(i-1)+dt*MeasuredOmegaAboutAxis;

% Calculate heading at a defined frequency
if DeterminationCounter > DeterminationStepSize
    if i > SamplesPerMeasurement
        % Calculate a moving average of the magnetometer data
        MovingAverageMagnetometerMeasurement =...
            mean(MeasuredMag(:,(i-SamplesPerMeasurement):i),2);

        % TRIAD to determine heading
        DueWest = ...
            cross(MovingAverageMagnetometerMeasurement, -BFRotationAxis);
        DueWest = DueWest/norm(DueWest);

        MeasuredHeading = acos(dot(DueWest, ForwardCrossGrav)) *...
            sign(DueWest(ForwardAxisIndex));

        % Convert to deflection angle
        ThetaMA = MeasuredHeading - EquilibriumAngle;

        % Predict heading with gyro data
        PredictedTheta = ThetaPC + MeasuredOmegaAboutAxis * DeterminationCounter*dt;

        if abs(PredictedTheta-ThetaMA) > 6
            % This is in case one measurement is grater than pi and the
            % other is not
            ThetaPC = PredictedTheta;
        else
            % Predictor-corrector filter
            ThetaPC = PredictedTheta * PredictorCorrectorGain+ThetaMA *...
                (1-PredictorCorrectorGain);
        end

        % This keeps the determined attitude in a seperate vector so that
        % it looks better when plotted
        ADnumber = ADnumber+1;

        % Without delay compensation
        % ADdata(:,ADnumber)...%
        % = [ThetaMA; ThetaMA-theta(i); ThetaPC; ThetaPC-theta(i); simtime(i)];%

        % With delay compensation
    end
end

```

```

ADdata(:,ADnumber) = [ThetaMA; ThetaMA - theta(i - round(SampleFrequency *...
    SampleBinPeriod / 2)); ThetaPC; ThetaPC - theta(i - round(SampleFrequency...
    * SampleBinPeriod / 2)); simtime(i) - (SampleBinPeriod / 2)];
```

else

% If insufficient data has been collected, do not record data

ThetaPC = MeasuredThetaGyro(i);

ThetaMA = nan;

end

DeterminationCounter = 0;

% Obtain attitude and body rates from sensor measurements

MeasuredMRP = BFRotationAxis \* tan(ThetaPC/4);

omegaBBfromGyro = BFRotationAxis \* MeasuredOmegaAboutAxis;

end

% Counter for attitude determination

DeterminationCounter = DeterminationCounter+1;

% ATTITUDE CONTROL

% Error terms

ErrorMRP(:,i) = diff\_sigma(MeasuredMRP,targetMRP); % MRP of attitude error

d\_omega = omegaBBfromGyro - target\_omega; % error in body rate

if isnan(ErrorMRP(:,i))

elseif ControlCounter >= ControlStepSize

% Error integrator

sigInt = sigInt + ErrorMRP(:,i)\*dt\*ControlStepSize;

zeta = K\*sigInt + IB\*(d\_omega - d\_omega0);

% Control law (Motor output torque)(Nm) Hogan Schaub 2013

ControlTorque = -IB \* (target\_omegaDot - CrossMatrix(omegaBBfromGyro) \*...

target\_omega) + K \* ErrorMRP(:,i) + P \* d\_omega + P \* Ki \* zeta -...

CrossMatrix(omegaBBfromGyro) \* (IB \* omegaBBfromGyro + J\_RW \*...

(omegaRW(:,i) + omegaBBfromGyro));

ControlCounter = 0;

end

% Control torque saturation

for j = 1:3

if abs(ControlTorque(j)) > 0.00025

ControlTorque(j) = 0.00025\*sign(ControlTorque(j));

end

end

ControlCounter = ControlCounter + 1;

Tcontrol(:,i) = ControlTorque;

% TESTBED DYNAMICS

```

% Torques from testbed (Nm)
Tspring(:,i) = -SpringConstant*theta(i)* BFRotationAxis;
Tdamping(:,i) = -DampingConstant*omegaBB(:,i);

% Sum of torques
torques(:,i) = Tspring(:,i)+Tdamping(:,i)-Tcontrol(:,i);

end % End of propagation

% PLOT

figure(1)

subplot(3,1,1)
plot(simtime,Tcontrol(1,:))
title('Motor output torques')
ylabel('x [Nm]')

subplot(3,1,2)
plot(simtime,Tcontrol(2,:))
ylabel('y [Nm]')

subplot(3,1,3)
plot(simtime,Tcontrol(3,:))
ylabel('z [Nm]')
xlabel('Time [s]')

figure(2)

subplot(3,1,1)
plot(simtime,omegaRW(1,:))
title('reaction wheel speeds')
ylabel('x [rad/s]')
grid on

subplot(3,1,2)
plot(simtime,omegaRW(2,:))
ylabel('y [rad/s]')
grid on

subplot(3,1,3)
plot(simtime,omegaRW(3,:))
ylabel('z [rad/s]')
xlabel('Time [s]')
grid on

targetthetas = ones(1,length(simtime))*TargetTheta;

figure(3)

subplot(2,1,1)
plot(ADdata(5,:), ADdata(1,:),'g',simtime, MeasuredThetaGyro,'m',...
ADdata(5,:), ADdata(3,:), simtime,theta, simtime,targetthetas)

```

```

ylabel('Deflection angle [rad]')
legend('AD through moving average filter','AD through propagating gyros',...
       'AD through predictor-corrector filter','True deflection angle')
grid on

subplot(2,1,2)
plot( ADdata(5,:), ADdata(2,:),'g',ADdata(5,:), ADdata(4,:))
    ylabel({'Attitude determination'; 'error [rad]'})
    xlabel('Time [s]')
    grid on


% CrossMatrix.m
% author: Maxim Clarke
% email: maxim.clarke.thesis@gmail.com@gmail.com

% CrossMatrix.m converts a vector to a cross-product matrix, allowing cross
% product operations to be conducted on matrices. The function is analogous
% to tilde.m and skew.m in other Space Grant MATLAB projects.

function [ Atilde ] = CrossMatrix( A )

    Atilde = [0 -A(3) A(2);
              A(3) 0 -A(1);
              -A(2) A(1) 0];

end


% diff_sigma.m
% author: Maxim Clarke
% Edited from a function written by Lee Jasper
% email: maxim.clarke.thesis@gmail.com@gmail.com

% diff_sigma.m calculates the modified rodriguez parameter that represents
% the rotational difference between two MRP vectors.

function d_sigma = diff_sigma(sigma1,sigma2)

    d_sigma = ((1 - (sigma2' * sigma2)) * sigma1 - (1 - (sigma1' * sigma1)) * ...
               sigma2 + 2 * CrossMatrix(sigma1) * sigma2...
               / (1 + (sigma2' * sigma2) * (sigma1' * sigma1) + 2 * sigma2' * sigma1);

    % Convert to shadow MRP if necessary
    if d_sigma' * d_sigma >= 1
        d_sigma = -d_sigma ./ (d_sigma' * d_sigma);
    end
end


% EulerDCM.m
% author: Maxim Clarke

```

```

% email: maxim.clarke.thesis@gmail.com@gmail.com
% EulerDCM.m produces a direction cosine matrix corresponding to a frame
% rotation about a specified body axis.

function [DCM] = EulerDCM(axis, angle)
    if axis == 1
        % Rotate about X (Roll)
        DCM = [1 0 0; 0 cos(angle) -sin(angle); 0 sin(angle) cos(angle)];
    elseif axis == 2
        % Rotate about Y (Pitch)
        DCM = [cos(angle) 0 sin(angle); 0 1 0; -sin(angle) 0 cos(angle)];
    else
        % Rotate about Z (Yaw/Heading)
        DCM = [cos(angle) -sin(angle) 0; sin(angle) cos(angle) 0; 0 0 1];
    end
end

% RK4integrate.m
% author: Maxim Clarke
% Edited from a function written by Lee Jasper
% email: maxim.clarke.thesis@gmail.com@gmail.com

% RK4integrate.m performs a Runge-Kutta 4 integration of rigid body dynamics
% for reference: http://en.wikipedia.org/wiki/Runge%20%93Kutta_methods

function [new_omega,new_sigma] = TestbedRK4integrate...
    (time_step,omega,sigma,Omega,torques,I,J_w,BFRotationAxis)

[omega_dot_k1, sigma_dot_k1] = ...
    TestbedEOM(omega,sigma,Omega,torques,I,J_w,BFRotationAxis);

[omega_dot_k2, sigma_dot_k2] = TestbedEOM(omega + 0.5 * time_step * omega_dot_k1, ...
    sigma + 0.5 * time_step * sigma_dot_k1,Omega,torques,I,J_w,BFRotationAxis);

[omega_dot_k3, sigma_dot_k3] = TestbedEOM(omega + 0.5 * time_step * omega_dot_k2, ...
    sigma + 0.5 * time_step * sigma_dot_k2,Omega,torques,I,J_w,BFRotationAxis);

[omega_dot_k4, sigma_dot_k4] = TestbedEOM(omega + time_step * omega_dot_k3,sigma + ...
    time_step * sigma_dot_k3,Omega,torques,I,J_w,BFRotationAxis);

new_omega = omega + time_step / 6 * ( omega_dot_k1 + 2 * omega_dot_k2 + 2 * ...
    omega_dot_k3 + omega_dot_k4);

new_sigma = sigma + time_step / 6 * ( sigma_dot_k1 + 2 * sigma_dot_k2 + 2 * ...
    sigma_dot_k3 + sigma_dot_k4);

end

% omega2sigmaD.m
% author: Maxim Clarke
% Edited from a function written by Lee Jasper

```

```

% email: maxim.clarke.thesis@gmail.com@gmail.com

% TestbedEOM contains the equations of motion of the test model on the
% string testbed according to rigid body dynamics. Any torques that are
% not parallel to the rotation axis are removed, analogously to how they
% would be removed by gravity.

function [omega_dot, sigma_dot] = ...
    TestbedEOM(omega,sigma,Omega,torques,I,J_w,BFRotationAxis)

    % Angular momentum of the reaction wheels
    h_s = J_w * eye(3) * ( Omega + omega );

    % Derive motion, remove perpendicular torques
    omega_dot = I \ (BFRotationAxis * dot((-CrossMatrix(omega) * I * omega -...
        CrossMatrix(omega) * h_s + torques),BFRotationAxis));

    omega_dot = BFRotationAxis * dot(omega_dot,BFRotationAxis);

    % rate change of MRP for integration
    sigma_dot = omega2sigmaD(omega,sigma);
end

```

## B.2. Sphere center calculation

```

% SphereCenterCalculation.m
% author: Maxim Clarke
% email: maxim.clarke.thesis@gmail.com@gmail.com

% SphereCenterCalculation.m takes a data set of magnetometer readings and
% attempts to calculate the bias produced by local electronic components by
% finding the center of the sphere described by the data points. This is
% performed by calculating the sphere described by hundreds of sets of
% four data points and finding the one that has the lowest overall standard
% deviation in vector magnitude after subtracting the center from all data
% points.

% SETUP

% collect data set
load 'SCC_data.mat'

% isolate magnetometer data
datasetnumber = 4;
OriginalSet = [TestData(datasetnumber).Mag_x', TestData(datasetnumber).Mag_y',...
    TestData(datasetnumber).Mag_z'];

% Butterworth Filter

SampleFrequency = 170; %Hz
CutoffSpeed = 50; % Nyquist speed at sample frequency of 160 is 500 rad/s
CutoffFrequency = CutoffSpeed/2*pi;
NormalizedCutoffFrequency = CutoffFrequency*2/SampleFrequency;

```

```

[BFa,BFb] = butter(2,NormalizedCutoffFrequency);

ButterFiltered = [filter(BFa,BFb,OriginalSet(:,1)) filter(BFa,BFb,OriginalSet(:,2))...
    filter(BFa,BFb,OriginalSet(:,3))];

% First few values are way off
ButterFiltered = ButterFiltered((10:length(OriginalSet(:,1))),:);

% Moving average filter

BinWidth = 15;
MADataPoints = (BinWidth+1):(length(OriginalSet(:,1))-BinWidth);
MovingAveraged = zeros(length(MADataPoints),3);

for i = MADataPoints
    MovingAveraged(i-BinWidth,:) = [mean(OriginalSet((i-BinWidth):(i+BinWidth),1))...
        mean(OriginalSet((i-BinWidth):(i+BinWidth),2))...
        mean(OriginalSet((i-BinWidth):(i+BinWidth),3))];
end

%choose which dataset will be corrected,
%UncorrectedSet = OriginalSet;
%UncorrectedSet = ButterFiltered;
UncorrectedSet = MovingAveraged;

% Empty vector
BestCenter = [0 0 0];

DataDivision = 9; % Minimum value
SampleSetNumber = (DataDivision-1)/4;
BestStd = 1000; %intentionally large

% CALCULATE
iterations = length(TestData(1).Mag_x)/16;
for j = 1:iterations
    % Define the selection of data
    DataDivision = DataDivision+4;
    SampleDistance = round(length(UncorrectedSet(:,1))/DataDivision)-1;
    SampleSetNumber = (DataDivision-1)/4;

    radii = zeros(SampleSetNumber,1);
    centers = zeros(SampleSetNumber,3);

    for i = 1:SampleSetNumber
        % Construct a matrix for input to the FindSphere function out
        % of four points deliberately selected to be far away from each
        % other
        SphereSample= [UncorrectedSet(i*SampleDistance,1)...
            UncorrectedSet(i*SampleDistance,2)...
            UncorrectedSet(i*SampleDistance,3);...
            UncorrectedSet((i+SampleSetNumber)*SampleDistance,1)...
            UncorrectedSet((i+SampleSetNumber)*SampleDistance,2)...
            UncorrectedSet((i+SampleSetNumber)*SampleDistance,3);...
            UncorrectedSet((i+2*SampleSetNumber)*SampleDistance,1)...
            UncorrectedSet((i+2*SampleSetNumber)*SampleDistance,2)...
```

```

UncorrectedSet((i+2*SampleSetNumber)*SampleDistance,3) ; ...
UncorrectedSet((i+3*SampleSetNumber)*SampleDistance,1)...
UncorrectedSet((i+3*SampleSetNumber)*SampleDistance,2)...
UncorrectedSet((i+3*SampleSetNumber)*SampleDistance,3)];

% Collect sphere calculation data
[radii(i),centers(i,:)] = FindSphere(SphereSample);
end

meanradii=mean(radii);
stdradii=std(radii);

% Remove spheres with abnormal radii
i=1;
while i <= length(radii)
    if abs(radii(i)-meanradii) >= stdradii
        radii(i) = [];
        centers(i,:) = [];
    else
        i=i+1;
    end
end

% Use the mean center value to correct for bias
AverageCenter = [mean(centers(:,1)) mean(centers(:,2)) mean(centers(:,3))];
CorrectedSet = [UncorrectedSet(:,1) - AverageCenter(1) UncorrectedSet(:,2) - ...
    AverageCenter(2) UncorrectedSet(:,3)-AverageCenter(3)];

% If the new center produces the lowest standard deviation in vector
% magnitude, then keep it.

magnitude = (CorrectedSet(:,1).^2+CorrectedSet(:,2).^2+CorrectedSet(:,3).^2).^0.5;
if std(magnitude) < BestStd
    BestCenter = AverageCenter;
    BestStd = std(magnitude);
end
end

% Displace the dataset by the best center value
CorrectedSet = [(UncorrectedSet(:,1)-BestCenter(1)) (UncorrectedSet(:,2) - ...
    BestCenter(2)) (UncorrectedSet(:,3)-BestCenter(3))];

magnitude = (CorrectedSet(:,1).^2+CorrectedSet(:,2).^2+CorrectedSet(:,3).^2).^0.5;
std(magnitude)

% PLOT

figure(1)
plot3(CorrectedSet(:,1),CorrectedSet(:,2),CorrectedSet(:,3),'.')
axis equal
axis vis3d
title('Corrected Magnetometer readings')
zlabel('Z')
ylabel('y')

```

```

xlabel('x')

% Adjust for the delay introduced by the filter by filling in the first
% values with NaN
nans = [];
for i = 1:BinWidth
    nans = [nans; nan];
end

figure(2)

subplot(3,1,1)
plot(OriginalSet(:,1))
hold on
plot([nans; UncorrectedSet(:,1)],'g')
hold off
title('Raw magnetometer readings (blue) plotted against the filtered signal (green)')
ylabel({'X Magnetometer output'; '(counts)'})

subplot(3,1,2)
plot(OriginalSet(:,2))
hold on
plot([nans; UncorrectedSet(:,2)],'g')
hold off
ylabel({'Y Magnetometer output'; '(counts)'})

subplot(3,1,3)
plot(OriginalSet(:,3))
hold on
plot([nans; UncorrectedSet(:,3)],'g')
hold off
ylabel({'Z Magnetometer output'; '(counts)'})
xlabel('Data point number')

figure(3)

plot(magnitude)
title('Magnitude of corrected magnetometer readings')
xlabel('Data point number')
ylabel('Vector magnitude')

% FindSphere.m
% author: Maxim Clarke
% email: maxim.clarke.thesis@gmail.com@gmail.com

% FindSphere.m calculates the center and radius of a sphere that is
% represented by four points (3D coordinates) on its surface.

% For explanation consult: http://www.abecedarical.com/zenosamples/zs_sphere4pts.html

function [ r,center ] = FindSphere( SphereSample )

```

```

SquareVector = SphereSample(:,1).^2+SphereSample(:,3).^2+SphereSample(:,2).^2;

a = [SphereSample ones(4,1)];
M11=det(a);

a = [SquareVector SphereSample(:,2) SphereSample(:,3) ones(4,1)];
M12=det(a);

a = [SquareVector SphereSample(:,1) SphereSample(:,3) ones(4,1)];
M13=det(a);

a = [SquareVector SphereSample(:,1) SphereSample(:,2) ones(4,1)];
M14=det(a);

a = [SquareVector SphereSample(:,1) SphereSample(:,2) SphereSample(:,3)];
M15=det(a);

xo = 0.5*M12/M11;
yo = -0.5*M13/M11;
zo = 0.5*M14/M11;
r = (xo^2+yo^2+zo^2-M15/M11)^0.5;
center = [xo yo zo];
end

```

### B.3. Attitude determination test

```

% AD_test.m
% author: Maxim Clarke
% email: maxaclarke@gmail.com

% AD_test.m calculates the error of heading measurements taken at a set of
% true heading values with a constant interval.

% SETUP
clear all
load 'AD_data.mat'
% TestData.Heading contains all measured heading values that have been
% parsed in sequence.

AngleStep = 20/180*3.1415;
Breakpoints = [1 19 37];
% CALCULATE

for i = 1:(length(Breakpoints)-1) % In case of multiple tests

    % Empty vectors
    ADtestResults(i).TrueHeading = zeros(1,(Breakpoints(i+1)-Breakpoints(i)));
    ADtestResults(i).HeadingError = zeros(1,(Breakpoints(i+1)-Breakpoints(i)));
    ADtestResults(i).MeasuredHeading = zeros(1,(Breakpoints(i+1)-Breakpoints(i)));

    % First heading measurement is considered true
    ADtestResults(i).MeasuredHeading(1) = mean(TestData(1).Heading);

```

```

ADtestResults(i).TrueHeading(1) = ADtestResults(i).MeasuredHeading(1);

Transition = 1;

for j = 2:(Breakpoints(i+1)-Breakpoints(i))
    % Integrate true heading
    ADtestResults(i).TrueHeading(j) =...
        ADtestResults(i).TrueHeading(j - 1) + AngleStep;

    % Keep true heading within +-pi
    if ADtestResults(i).TrueHeading(j) > 3.141592
        ADtestResults(i).TrueHeading(j) =...
            ADtestResults(i).TrueHeading(j) - 6.2832;

    % Keeps track of the event
    Transition = j;
end

% Retrieve heading
ADtestResults(i).MeasuredHeading(j) =...
    mean(TestData(j + Breakpoints(i) - 1).Heading);

% Determine error
ADtestResults(i).HeadingError(j) = ADtestResults(i).MeasuredHeading(j) -...
    ADtestResults(i).TrueHeading(j);

% This corrects for when measured and true heading are on
% different sides of +-pi
if abs(ADtestResults(i).HeadingError(j)) > 3.141592
    if sign(ADtestResults(i).HeadingError(j)) > 0
        ADtestResults(i).HeadingError(j) = ...
            ADtestResults(i).HeadingError(j) - 6.2832;
    else
        ADtestResults(i).HeadingError(j) =...
            ADtestResults(i).HeadingError(j) + 6.2832;
    end
end
end

if Transition == 1
    BeforeTransition = 18;
else
    BeforeTransition = Transition - 1;
end

% This interpolates across +-pi and adds data points at +pi and -pi

% Interpolate value between first and last data points
MiddleValue = ADtestResults(i).HeadingError(BeforeTransition) +...
    (ADtestResults(i).HeadingError(Transition) -...
        ADtestResults(i).HeadingError(BeforeTransition)) / 2 ;

% last two points are the middle value
ADtestResults(i).HeadingError((Breakpoints(i + 1)-Breakpoints(i)+1): ...
    (Breakpoints(i + 1)-Breakpoints(i) + 2)) = MiddleValue;

```

```

% Add heading reference values
ADtestResults(i).TrueHeading(length(ADtestResults(i).TrueHeading)+1) = 3.142;
ADtestResults(i).TrueHeading(Breakpoints(i + 1) - Breakpoints(i)+2) = - 3.142;
% This isn't mathematically perfect but meh
ADtestResults(i).MeasuredHeading(Breakpoints(i + 1) - Breakpoints(i)+1) = 3.142;
ADtestResults(i).MeasuredHeading(Breakpoints(i + 1) - Breakpoints(i)+2) = - 3.142;
end

% PLOT

figure(1)

plot(ADtestResults(1).TrueHeading(1:length(ADtestResults(1).TrueHeading)-2),...
      ADtestResults(1).HeadingError(1:length(ADtestResults(1).TrueHeading)-2),'o',...
      ADtestResults(2).TrueHeading(1:length(ADtestResults(1).TrueHeading)-2),...
      ADtestResults(2).HeadingError(1:length(ADtestResults(1).TrueHeading)-2),'x')
title('Heading error as a function of heading')
ylabel('Heading error (rad)')
xlabel('Heading as determined by the protractor')
grid on

figure(2)

plot(ADtestResults(1).TrueHeading(1:length(ADtestResults(1).TrueHeading)-2), 'r')
hold on
plot(ADtestResults(1).MeasuredHeading(1:length(ADtestResults(1).TrueHeading)-2), 'o')
plot(ADtestResults(2).MeasuredHeading(1:length(ADtestResults(1).TrueHeading)-2), 'xm')
hold off
title('Measured heading values (blue and purple) plotted against true heading (red)')
xlabel('Data point number')
ylabel('Heading (rad)')
grid on
save('AD_results.mat', 'ADtestResults')

```





# UN8-ACS717.1

## ACS Validation Testing Setup

---

### Attitude Determination and Control (ACS)

#### March 07, 2016

**Maxim Clarke**  
**ACS Testing Engineer**

#### Approvals and Dates

##### Author

---

Maxim Clarke

03/07/2016

##### Project Manager

---

Glenda Alvarenga

03/07/2016

##### Chief Systems Engineer

---

Christopher Rouw

03/07/2016

##### Avionics Engineer

---

Franklin Hinckley

03/07/2016

##### Integration & Testing Lead

---

Jannine Vela

03/07/2016

| Revision Table |  |              |            |
|----------------|--|--------------|------------|
| Revision       | Notes  | REA          | Date       |
| UN8-ACS717.0-0 | Initial Draft  | Maxim Clarke | -          |
| UN8-ACS717.1-0 | <ul style="list-style-type: none"> <li>• New template/structure</li> <li>• Included integration</li> <li>• Combined integration, magnetometer biasing, AD test and string characterization into one procedure</li> </ul> | Maxim Clarke | 21/5/2015  |
| UN8-ACS717.2-0 | <ul style="list-style-type: none"> <li>• Thesis version</li> <li>• Interface with new parsing scripts</li> </ul>   | Maxim Clarke | 17/12/2015 |
| UN8-ACS717.3-0 | <ul style="list-style-type: none"> <li>• Final thesis version</li> <li>• New requirements</li> </ul>   | Maxim Clarke | 26/02/2016 |

| Special Notes |
|---------------|
|               |

## Contents

|   |           |
|---|-----------|
| <b>I Overview</b>                           | <b>4</b>  |
| A Document Scope . . . . .                  | 4         |
| B Relevant documentation . . . . .          | 4         |
| C Mission Acronyms . . . . .                | 4         |
| D Assumptions . . . . .                     | 4         |
| <b>II Introduction</b>                      | <b>5</b>  |
| <b>III Integration</b>                      | <b>5</b>  |
| A Objective . . . . .                       | 5         |
| B Methodology . . . . .                     | 5         |
| C Hardware List . . . . .                   | 6         |
| D Software List . . . . .                   | 7         |
| E Applicable Requirements . . . . .         | 7         |
| F Procedure . . . . .                       | 8         |
| G Acceptance Checklist . . . . .            | 15        |
| <b>IV Testbed Setup</b>                     | <b>17</b> |
| A Objective . . . . .                       | 17        |
| B Methodology . . . . .                     | 17        |
| C Hardware List . . . . .                   | 17        |
| D List of Applicable Requirements . . . . . | 17        |
| E Procedure . . . . .                       | 18        |
| F Acceptance Checklist . . . . .            | 20        |
| <b>V Sphere Center calculation</b>          | <b>22</b> |
| A Objective . . . . .                       | 22        |
| B Methodology . . . . .                     | 22        |
| C Hardware List . . . . .                   | 22        |
| D Software List . . . . .                   | 22        |
| E List of Applicable Requirements . . . . . | 23        |
| F Procedure . . . . .                       | 23        |
| G Acceptance Checklist . . . . .            | 25        |
| <b>VI Attitude determination test</b>       | <b>27</b> |
| A Objective . . . . .                       | 27        |
| B Methodology . . . . .                     | 27        |
| C Hardware List . . . . .                   | 27        |
| D Software List . . . . .                   | 27        |
| E List of Applicable Requirements . . . . . | 28        |
| F Procedure . . . . .                       | 28        |
| G Acceptance Checklist . . . . .            | 30        |
| <b>VII String Torque Characterization</b>   | <b>32</b> |
| A Objective . . . . .                       | 32        |
| B Methodology . . . . .                     | 32        |
| C Hardware List . . . . .                   | 32        |
| D Software List . . . . .                   | 32        |
| E List of Applicable Requirements . . . . . | 33        |
| F Procedure . . . . .                       | 33        |
| G Acceptance Checklist . . . . .            | 37        |

## I. Overview

### A. Document Scope

The purpose of this document is to define the necessary actions to prepare the PolarCube attitude control system for testing in a string suspension microgravity simulation environment.

### B. Relevant documentation

The following is a list of documents that are referenced or otherwise applicable to this document.

| Doc. Number   | Title  |
|---------------|--|
|               | Attitude Control Validation Testing for the PolarCube satellite      |
|               | Attitude Control Validation Testing Requirements Verification Matrix |
| UN8-STR510.4  | Full assembly procedure  |
| UN8-ACS718.1  | De-tumble procedure  |
| UN8-ACS719.1  | Slew Test Procedure  |
| UN8-ACS720.1  | Pointing stability test procedure                                    |
| UN8-ACS721.1  | Momentum dumping test procedure                                      |
| to be written | Installation procedures  |
| to be written | Interface procedures   |
| to be written | Functional checkout procedures                                       |

### C. Mission Acronyms

All acronyms utilized within this document are defined in alphabetical order.

| Acronym   | Definition   |
|-----------|--|
| ACS       | The PolarCube Attitude Determination and Control Subsystem   |
| ALL-STAR  | Agile Low-cost Laboratory for Space Technology And Research  |
| CAD       | Computer Aided Design  |
| CDH       | Command and Data Handling Subsystem  |
| EGSE      | Electrical Ground Support Equipment  |
| EPS       | Electric Power Subsystem   |
| FSW       | Flight Software  |
| GPS       | Global Positioning System  |
| GUI       | Graphical user interface   |
| MGSE      | Mechanical Ground Support Equipment  |
| PEZ       | Payload Extension zone   |
| PolarCube | The UN8 satellite system from the Colorado Space Grant Consortium at the University of Colorado at Boulder |
| RBFP      | Remove-Before-Flight Pin   |
| UN8       | The University Nanosatellite Program   |

### D. Assumptions

1. This document assumes that a CAD model of the ACS validation test model exists.
2. This document assumes that the XBee's being used have been configured to communicate with each other.
3. This document assumes that the software in the project Git repository is internally consistent. In that it will function using the actions described in the procedure.
4. This document assumes that the test engineer has all of the necessary screws and tools at hand.

## II. Introduction

This procedure should fully prepare the systems involved to perform attitude control validation tests of the PolarCube ACS on a string suspension microgravity simulation testbed. In particular, This procedure is catered to perform the tests described in the reference documents. This consists of physically integrating a test model that would allow ACS to perform on the testbed, installing the appropriate FSW, establishing wireless communication between the test model and an interface computer, assembling the test system, correcting the magnetometer for magnetic bias introduced by bus components, verifying attitude determination performance and characterizing the external torques introduced by the string suspension rig.

The test model hardware consists of the ACS main board and actuators, the CDH backplane, an XBee breakout, the EPS main board and the battery assembly all mounted to the ALL-STAR PEZ-bus structural panels. This provides the minimum functionality required to perform ACS Validation tests. Integration will also verify test model electronics and software functionality as well as determine residual gyro biases and the amplitude of influence that the reaction wheels have on the magnetometers.

Sphere center calculation will determine the body-constant magnetic influence of bus components on magnetometer measurements caused by currents in the electronics. This information will be used for magnetometer biasing when attitude determination is performed.

The attitude determination test will measure the test model's heading determination error as a function of true heading and verify that heading determination error is an invariant function of heading.

String testbed characterization will verify that the external torques the test model is experiencing on the string suspension testbed can be modeled as a damped spring and determine the modeling parameters of those forces; namely the heading of the equilibrium position, the spring constant and the damping constant.

## III. Integration

### A. Objective

This procedure describes integration of the ACS validation test model, set-up of wireless communication between the test model and the Interface computer and steps that should be taken to ensure that it is ready for the next phase of setup. If the system passes the acceptance checklist, the test model is considered to be properly integrated and programmed with an appropriate version of FSW; communication between the test model and interface computer is established and working as expected and attitude determination and control have been verified to a preliminary degree. Integration also includes characterization of the influence of the reaction wheels on the magnetometer and gyro bias correction.

### B. Methodology

Flight software must be installed before physical integration because no programming interface will be available once the system is integrated. Eventually, the full system EGSE will be developed to the point where this is possible, but for now, changes to FSW will require a direct interface to the subsystem via the subsystem EGSE. Once the software is installed, the subsystems will go through a flat-sat verification process to incrementally verify communication and functionality. Flat-sat verification will conclude with the full system, independent of EGSEs, being verified through XBee communication.

**C. Hardware List**

| Subassembly                            | Component                     | Quantity |
|--|-------------------------------|----------|
| <b>Test Model Assembly Integration</b> |                               |          |
|  | Bus panels                    | 4        |
| <b>CDH</b>                             |                               |          |
|  | CDH backplane                 | 1        |
|  | XBee breakout                 | 1        |
|  | Test model XBee               | 1        |
|  | XBee to AUX COM cable         | 1        |
|  | Dummy COM RF board with clips | 1        |
|  | CDH EGSE                      | 1        |
|  | CDH MGSE                      | 1        |
|  | Micro SD card                 | 1 or 2   |
|  | Laboratory power supply       | 2        |
| <b>EPS</b>                             |                               |          |
|  | EPS Main board                | 1        |
|  | Battery brackets              | 2        |
|  | Battery boards                | 2        |
|  | Charged batteries             | 4        |
|  | EPS Inhibit pin               | 1        |
|  | EPS EGSE                      | 1        |
|  | EPS MGSE                      | 1        |
|  | Laboratory power supply       | 1        |
| <b>ACS</b>                             |                               |          |
|  | ACS Main board                | 1        |
|  | Driver boards                 | 3        |
|  | Reaction wheel assemblies     | 3        |
|  | Main board to actuator cables | 3        |
|  | ACS EGSE                      | 1        |
|  | ACS MGSE                      | 1        |
|  | Laboratory power supply       | 2        |
| <b>Testbed</b>                         |                               |          |
|  | Interface computer            | 1        |
|  | External XBee                 | 1        |
|  | XBee Explorer                 | 1        |
|  | Micro SD adapter              | 1        |
|  | USB pen drive                 | 1        |

## D. Software List

**Git repository address:** `git@thinker.colorado.edu:polarcube/acs-validation-testing.git`

The test engineer should have a copy of the full repository on their computer. This directories listed are required in their entirety. The procedure only mentions the files that the test engineer will have to interact with and are referred to in the procedure. Many of these files require peripheral files that are included in the directories, but not mentioned in the procedure.

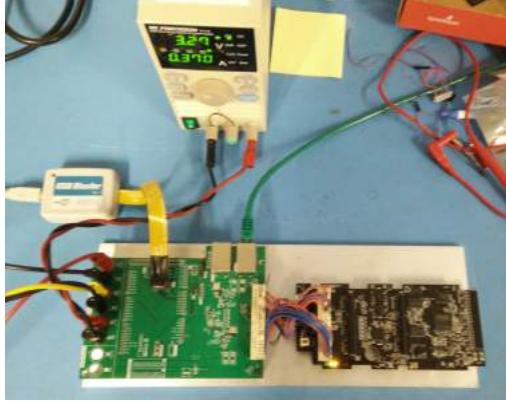
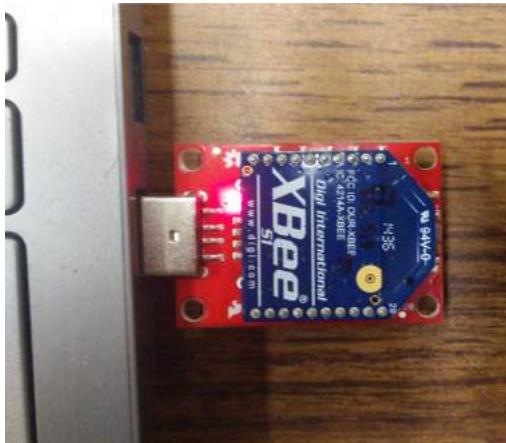
| Subsystem                 | File/Directory  | Location  |
|---------------------------|---|---|
| <b>ACS</b>                |   |   |
|                           | ACS validation testing FSW  | <code>FSW\ACS\..</code>   |
| <b>CDH</b>                |   |   |
|                           | CDH validation testing FSW  | <code>FSW\CDH\..</code>   |
| <b>EPS</b>                |   |   |
|                           | EPS validation testing FSW  | <code>FSW\EPS\..</code>   |
| <b>Interface computer</b> |   |   |
|                           | Validation testing GUI  | <code>GUI\OPGUI9_Maxim.py</code>  |
| <b>Data analysis</b>      |   |   |
|                           | Gyro bias data analysis repository                                | <code>MATLAB\validation_test_data_analysis\data_analysis\Gyro_bias_sample</code>      |
|                           | Reaction Wheel influence on magnetometer data analysis repository | <code>MATLAB\validation_test_data_analysis\data_analysis\Single_wheel_spinning</code> |

## E. Applicable Requirements

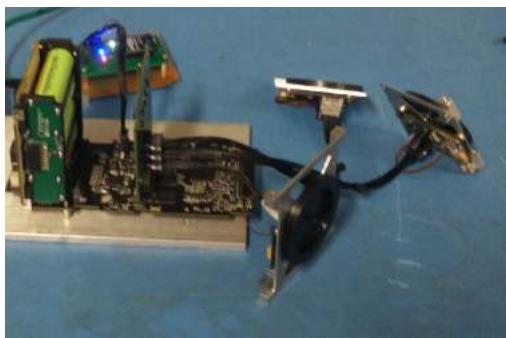
| Req. Number | Description  | Verification method                                  |
|-------------|--|--|
| 1.ACSVT.2   | ACS shall be provided with a power supply.   | EPS powers the test model.                           |
| 1.ACSVT.3   | ACS shall have knowledge of its attitude   | Attitude determination preliminary check successful. |
| 1.ACSVT.14  | Attitude and reaction wheel speeds shall be recorded and available for analysis.                     | Writing data to the SD card is successful.           |
| 1.ACSVT.25  | ACS shall be able to control the spin rate of the reaction wheels.                                   | Reaction wheel spin rate control is verified.        |
| 1.ACSVT.28  | The test engineer shall be able to observe ACS' state vector.  | ACS state vector is visible on the downlink GUI      |
| 1.ACSVT.29  | The test engineer shall have the ability to command the start of a test.                             | Sending op-codes with the GUI is verified.           |
| 1.ACSVT.30  | The test engineer shall be able to set the control law gains and targets of the test model.          | Sending op-codes with the GUI is verified.           |
| 1.ACSVT.31  | The test engineer shall be able to upload sensor bias correction values to an integrated test model. | Sending op-codes with the GUI is verified.           |
| 1.ACSVT.32  | ACS shall be able to receive commands from the test engineer.  | Sending op-codes with the GUI is verified.           |

|            |  |   |
|------------|--|---|
| 1.ACSVT.33 | ACS shalll be able to communicate wirelessly with an interface computer. | Uplink and downlink GUI operation verified. |
|------------|--|---|

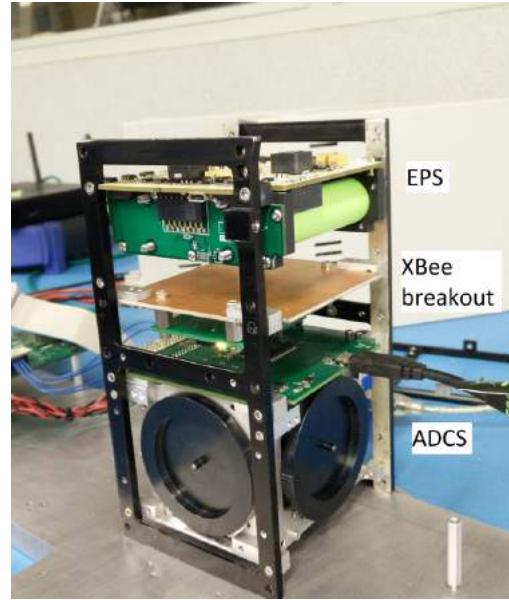
## F. Procedure

| Step | Description  | Pictures/Notes   |
|------|--|--|
| 1    | Mount CDH to its ground support equipment (MGSE and EGSE). The EGSE should have power, Ethernet and JTAG connections. Check that all connections are safe before supplying power.  |    |
| 2    | Load the FSW from the validation testing repository onto CDH. (See CDH FSW installation procedure)   | CDH Installation procedure does not exist yet.                                       |
| 3    | Conduct a CDH functional checkout (See CDH Functional checkout procedure)  | CDH functional checkout procedure does not exist yet.                                |
| 4    | Mount the test model XBee to the XBee breakout board (power switch in OFF position) and connect it to CDH. The alignment notch on the female connector that has been cut off should align with the side of the male connector (cable) that has three groves on the outside rather than five. Check for safe connections before commanding CDH to supply power to AUX COM and flipping the power switch on the breakout board to "ON" |  |
| 5    | Mount the external XBee to the XBee explorer and plug it into the interface computer.  |  |

- 
- 6 Boot Linux on the interface computer and run the GUI program: open a terminal (ctrl+ALT+T) and write: `python OPGUI9_Maxim` when prompted for a serial port, enter "0". Picture
- 
- 7 Mount the ACS main board to its ground support equipment. The EGSE should have power, Ethernet and JTAG connections. The ACS main board should be connected to the ACS lab computer via USB. Check that all connections are safe before powering on. picture
- 
- 8 Open the ACS FSW in Atmel Studio on the ACS lab computer. Go into the `AttitudeDetermination` function in `ACSmath.c` and un-comment the print statements for heading, pitch and roll. Click on "Run without debugging" to load the software onto ACS.
- 
- 9 Open a serial terminal in Atmel studio and set the port to COM6, there should be a stream of attitude data. Carefully remove the main board from the MGSE and rotate it to verify that heading, pitch and roll are being measured correctly. Pitch and roll should be fairly accurate. Heading will not because the magnetometer data won't be bias corrected, but the direction should be consistent (anticlockwise motion results in increase in heading). The Euler angle definitions should change every time the magnitude of pitch or roll surpasses 45 deg.
- 
- 10 Comment out the print statements in Atmel Studio and re-load the software onto the main board.
- 
- 11 Mount the actuator cards (reaction wheel assembly + driver boards) on their ground support equipment and connect them to the ACS main board. Make sure the connections are safe before powering on.
- 
- 12 The X and Y wheels should start spinning once the main board boots. The Z wheel won't spin on the EGSE unless the enable pin bug has been solved.
- 
- 13 Click on "Start" in the uplink window of the GUI on the interface computer. A data stream should appear in the downlink window. Check that the sensor, attitude and reaction wheel speed values make sense.
- 
- 14 Send few op-codes with the uplink GUI (biases, targets) and verify that the parameters change in the downlink window. Uplink GUI manual does not exist yet
-

|    |   |  |
|----|---|--|
| 15 | Set the control mode of all three reaction wheels to spin rate control at 100 rad/s and verify that they speed up visually and through the tachometer readings.   |  |
| 16 | Use the uplink GUI to turn off reaction wheel control for all three reaction wheels by setting all reaction wheel control modes to 0.   |  |
| 17 | Turn off the power supply to both ACS and CDH. Remove the ACS main board from its ground support equipment disconnect the reaction wheels and connect it directly to CDH. Reconnect the reaction wheels to ACS and turn on the 3V3 and 12V power supplies to CDH. |  |
| 18 | Repeat steps 13-16  | The Z wheel should work from this point forward.                                     |
| 19 | Mount the EPS main board to its ground support equipment  |  |
| 20 | Install FSW from the validation testing repository onto the EPS main board (See EPS installation procedure)   | EPS installation procedure does not exist yet  |
| 21 | Integrate the battery/bracket/boards assembly (See Assembly procedure chapter 5 )   |  |
| 22 | Perform battery assembly functional checkout and main board functional checkout. (See EPS functional checkout procedure)  | EPS functional checkout procedure does not exist yet                                 |
| 23 | Integrate the battery assembly to the EPS Main board (See integration procedure section 5.3.5)  |  |
| 24 | Perform a functional checkout of the integrated EPS system (See EPS functional checkout procedure)  |  |
| 25 | Turn off the power supply to CDH, make sure that EPS is inhibited and connect EPS directly to CDH.  |  |
| 26 | Un-inhibit EPS. CDH should turn on.   |  |
| 27 | Once the lights on the XBee breakout have turned on, click on "ACS" in the GUI uplink window to turn on ACS.  |  |
| 28 | Repeat steps 13-16  |  |
| 29 | Congratulations, you have verified the test model's electronics and software!   | Electronics and software checkout: <input type="checkbox"/>                          |

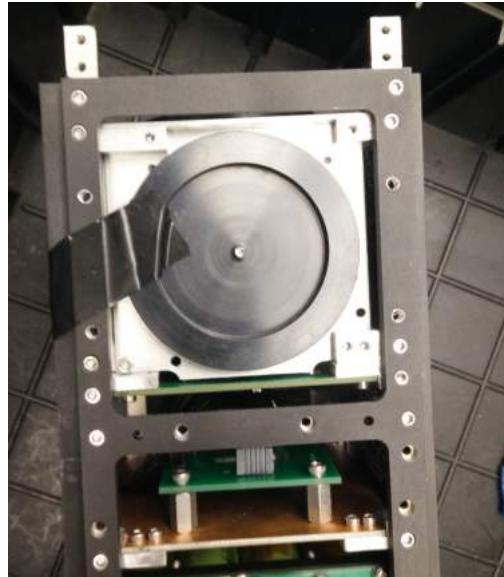
- 
- 30 Integrate EPS, ACS, CDH and the XBee into the PEZ-BUS structure (See assembly procedure chapter 10). The XBee breakout should be mounted to the dummy COM RF board and integrated as if it was COM RF with the cable going to the AUX COM port.



- 
- 31 Congratulations, you have built the ACS validation test model! Documentation that refers to the ACS validation test model refers to the assembly that you now have.
- 
- 32 Repeat steps 13-16
- 
- 33 Turn off the test model by inhibiting EPS. Remove the SD card from CDH, plug it into the interface computer using the adapter and delete all of the files on it.
- 
- 34 Replace the SD card into CDH and un-inhibit it. If necessary, re-open the GUI. When the lights on the XBee breakout turn on, command CHD to turn on ACS with the GUI. Once ACS boots click "Start" on the GUI. In further documentation, this process will be referred to as "Start a test."
- 

Test model has been built:

- 
- 35 Command the X reaction wheel to spin at a constant rate below 200 rad/s while the other two remain stationary. Electrical tape can be used to fix the other two wheels. The wheels also have "dead" positions where they won't start even if the motor is on. Once the wheel has reached the target speed.



- 
- 36 Send an op-code setting the "Filter mode" to 0. Let it record for a few seconds, then set "Filter mode" back to 1.

- 
- 37 Without stopping the test, repeat the previous two steps for spinning Y and Z wheels.

- 
- 38 Click on "Stop" in the GUI, stopping data logging. then click on "ACS", turning ACS off. Then inhibit EPS. In future documentation, this process will be referred to as "Stop the test".

- 
- 39 Access the micro SD card on the interface computer. There should be a single file on it. Give the file a unique and identifying name like `influence_RW_on_mag_(etc...).txt`. Back it up on a the computer and USB drive. Once you have named the test file on the SD card, clearing the SD card at this point is optional. There should be no files on the SD card with a default file name. In further documentation, this process will be referred to as "Retrieve the test data".

- 
- 40 Copy the `MATLAB\Validation_test_data_analysis\data_analysis` directory from the git repository and paste it in the `Validation_test_data_analysis` directory with whatever name you chose for this test project. This directory will be referred to as the data analysis repository in further documentation with the following handle: `..\data_analysis_new\`.

Directory name: \_\_\_\_\_

- 
- 41 Open `..\data_analysis_new\Single_wheel_spinning\Raw_data_files`. Empty the folder and add the test data file that you just created.

- 
- 42 Open and run ..\Single\_wheel\_spinning\Parsing\_main.m in MATLAB. A plot of the magnetometer responses should pop up and the resulting amplitudes should appear in the workspace.
- 43 Record the amplitudes:
- Amplitude X RW on mag. X axis: \_\_\_\_\_
- Amplitude X RW on mag. Y axis: \_\_\_\_\_
- Amplitude X RW on mag. Z axis: \_\_\_\_\_
- Amplitude Y RW on mag. X axis: \_\_\_\_\_
- Amplitude Y RW on mag. Y axis: \_\_\_\_\_
- Amplitude Y RW on mag. Z axis: \_\_\_\_\_
- Amplitude Z RW on mag. X axis: \_\_\_\_\_
- Amplitude Z RW on mag. Y axis: \_\_\_\_\_
- Amplitude Z RW on mag. Z axis: \_\_\_\_\_
- 
- 44 At this point the operator can chose between attempting to parse the influence\_RW\_on\_mag\_(etc).txt file to find gyro data representing a stationary test model or collecting a new dataset to determine gyro biases. Provided that the test model was stationary for a period during the test, it would be simple for a person who understands how the Parsing\_main.m script works. The influence on the reaction wheels on the magnetometers is unlikely to change much between subsequent re-integrations, so the test engineer might also opt to only do gyro bias determination because it will always have to be repeated. The following steps illustrate how to go about doing it from scratch.
- 45 Start a test, make sure that the test model is not moving. Then send an op-code setting "Filter mode" to 0. Wait for a minute (or 30 seconds or whatever), then set "Filter mode" to 1. Then stop the test and retrieve the test data.
- 46 Open ..\data\_analysis\_new\Gyro\_bias\_sample\Raw\_data\_files. Delete the contents and replace with the new data file.
-

47 Open and run ..\Gyro\_bias\_sample\Parsing\_main.m. A plot of the gyro data should pop up and the bias and standard deviation values should appear in the workspace.

---

48 Record gyro bias and noise standard deviation values:

Gyro 1 X mean value: \_\_\_\_\_

Gyro 1 X standard dev.: \_\_\_\_\_

Gyro 1 Y mean value: \_\_\_\_\_

Gyro 1 Y standard dev.: \_\_\_\_\_

Gyro 1 Z mean value: \_\_\_\_\_

Gyro 1 Z standard dev.: \_\_\_\_\_

Gyro 2 X mean value: \_\_\_\_\_

Gyro 2 X standard dev.: \_\_\_\_\_

Gyro 2 Y mean value: \_\_\_\_\_

Gyro 2 Y standard dev.: \_\_\_\_\_

Gyro 2 Z mean value: \_\_\_\_\_

Gyro 2 Z standard dev.: \_\_\_\_\_

---

**G. Acceptance Checklist****ACS Validation Test Model Integration Acceptance Checklist**

Project: PolarCube

Date:

Subsystem Revision: Silver

Passed (Y/N):

**ACS Validation test model Integration Requirements**

| Item   | Actual | Required  | Verification Type | Verification Notes |
|--|--------|-----------|-------------------|--------------------|
| ACS, CDH, EPS and an XBee have been integrated into the PEZ-BUS structure  | _____  | ( Y / N ) | Inspection        |                    |
| ACS performs attitude determination.   | _____  | ( Y / N ) | Inspection        |                    |
| ACS performs reaction wheel spin rate control.   | _____  | ( Y / N ) | Inspection        |                    |
| ACS sends status data to the downlink GUI on the interface computer via the XBee on CDH  | _____  | ( Y / N ) | Inspection        |                    |
| CDH Logs test data locally on an SD card.  | _____  | ( Y / N ) | Inspection        |                    |
| ACS Commands can be sent to ACS via XBee with the uplink GUI.  | _____  | ( Y / N ) | Inspection        |                    |
| Influence of the reaction wheels on the magnetometer has been measured and documented in step 44 of the integration procedure. | _____  | ( Y / N ) | Inspection        |                    |
| Gyro bias has been measured and documented in step 49 of the integration procedure.  | _____  | ( Y / N ) | Inspection        |                    |
| <b>All Requirements Met</b>  | _____  | ( Y / N ) |                   |                    |

**Overall Test Notes:**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## IV. Testbed Setup

### A. Objective

This procedure describes the method used to set up the string suspension testbed for ACS validation tests.

### B. Methodology

The test location is selected. The fishing lines are then suspended and hooks for the attitude determination test rig attached. The attitude determination test rig is then used to determine the height of the test location.

### C. Hardware List

| Subassembly    | Component             | Quantity |
|----------------|-----------------------|----------|
| <b>Testbed</b> |                       |          |
|                | Spool of fishing line | 1        |
|                | Aluminum hooks        | 1        |
|                | Cardboard             | 1        |

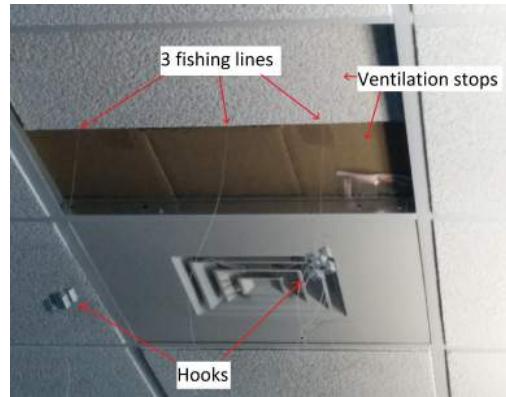
### D. List of Applicable Requirements

| Req. Number | Description   | Verification method  |
|-------------|---|--|
| 1.ACSVT.23  | ACS shall be suspended far away from external magnetic field sources. | Test location is selected to be in the center of a large open space. |

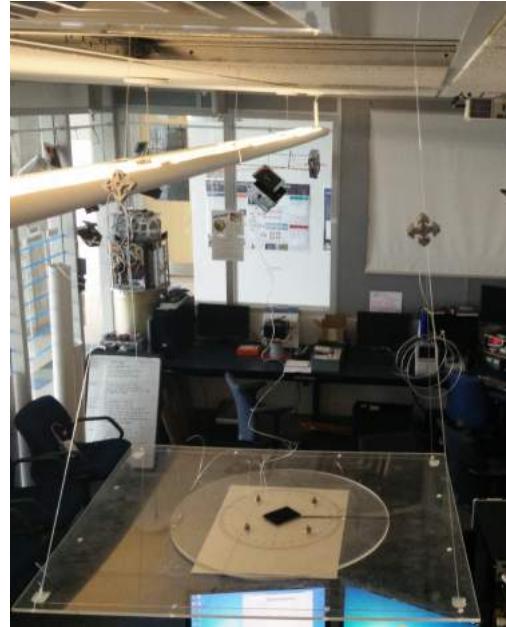
## E. Procedure

| Step | Description   | Pictures/Notes  |
|------|---|---|
| 1    | Select a location where the ACS validation tests will be performed. Far from magnetic sources, under an object that the test model and attitude determination test rig can be hung by. This means that the object must be able to support the weight of the test model on the test rig and allow multiple linearly aligned attachments (like a horizontal bar). The current favorite location is the fire extinguisher pipe in the center of the ceiling of the mission operations command. | <br> |
| 2    | Tie three fishing lines to the object that the test model will be hung by. One will serve to suspend the test model, the other two will hold hooks for the attitude determination test rig.   |   |
| 3    | Tie aluminum hooks to the two outer strings. They should be roughly the same height and the attitude determination test rig should be at the height that the tests will take place in when suspended by them.   |   |

- 
- 4 Make sure that airflow around the test location is minimized, in this case it meant sealing up the hole made in the ceiling using the displaced ceiling panel and a sheet of cardboard.



- 
- 5 Suspend the attitude determination test rig from the hooks and align it so that it is approximately level.



- 
- 6 Pull the test model suspension line (middle line) tight towards the center of the attitude determination test rig. It should describe a vertical line, if it doesn't then the attachment point should be adjusted.

- 
- 7 Use a marker to mark the point on the suspension line that approximates where the magnetometer will be during suspension tests. This point on the line will be used as reference for the test location in later procedures.



**F. Acceptance Checklist****ACS Validation Testbed Setup**

Project: PolarCube

Date:

Subsystem Revision: Silver

Passed (Y/N):

**ACS Validation Test Sphere Center Correction Acceptance Checklist**

| Item  | Actual | Required  | Verification Type | Verification Notes |
|---|--------|-----------|-------------------|--------------------|
| A test location was selected that meets the criterion stated in the procedure.  | _____  | ( Y / N ) | Inspection        |                    |
| Three fishing lines were suspended from the attachment point and hooks were tied to the two outer lines at roughly the same height about a foot and a half above the desired test location. | _____  | ( Y / N ) | Inspection        |                    |
| The attitude determination test rig was suspended from the hooks and the test location was marked on the suspension string, using the attitude determination test rig as reference.         | _____  | ( Y / N ) | Inspection        |                    |
| <b>All Requirements Met</b>   | _____  | ( Y / N ) |                   |                    |

**Overall Test Notes:**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## V. Sphere Center calculation

### A. Objective

This procedure describes the method used to determine the requisite bias correction in magnetometer measurements to correct for body-constant magnetic influences caused by hard iron and steady current flow through electrical equipment.

### B. Methodology

Magnetometer data is collected while the test model is rotated in several directions. The collected data should describe a sphere with a center that is not at the origin. The sphere is produced by varying the relative orientation of the ambient magnetic field with respect to the test model's body. The position of the center of the sphere is the magnetic bias produced by the bus components. The optimal center of the sphere is calculated with an algorithm and used to correct bias in magnetometer measurements. This procedure must be performed at the specific location of testing. Requisite bias correction is in part a function of the ambient field so it must be done at the test location for best attitude determination performance. The Arduino test model will be used as a control to see how the reaction wheels are influencing the measurements.

### C. Hardware List

| Subassembly        | Component                 | Quantity |
|--------------------|---------------------------|----------|
| <b>Test Models</b> |                           |          |
|                    | ACS validation test model | 1        |
| <b>Testbed</b>     |                           |          |
|                    | Interface computer        | 1        |
|                    | External XBee             | 1        |
|                    | XBee Explorer             | 1        |

### D. Software List

Git repository address: <git@thinker.colorado.edu:polarcube/acs-validation-testing.git>

The test engineer should have a copy of the full repository on their computer. This directories listed are required in their entirety. The procedure only mentions the files that the test engineer will have to interact with and are referred to in the procedure. Many of these files require peripheral files that are included in the directories, but not mentioned in the procedure.

| Subsystem                 | File/Directory                              | Location  |
|---------------------------|---|---|
| <b>Interface computer</b> |   |   |
|                           | Validation testing GUI                      | GUI\OPGUI9_Maxim.py                             |
| <b>Data analysis</b>      |   |   |
|                           | Sphere center calculation parsing directory | ..\data_analysis\<br>\Sphere_center_calculation |

## E. List of Applicable Requirements

| Req. Number | Description   | Verification method  |
|-------------|---|--|
| 1.ACSVT.21  | Magnetic bias produced by test model components shall be determined and removed from magnetometer measurements. | Sphere center calculation is conducted, the standard deviation of corrected vector magnitude is below a TBD threshold. |

## F. Procedure

| Step | Description  | Pictures/Notes  |
|------|--|---|
| 1    | Place the test model on the suspended attitude determination test rig and start a test.  |   |
| 2    | Send op-codes ensuring that the magnetometer biases are all 0. Make sure that the reaction wheels are on speed control. Set the reaction wheel target speed to somewhere in the 100-300 rad/s range.   |   |
| 3    | When the reaction wheels have reached the target speed, set "Filter mode" to 0.  |   |
| 4    | Rotate the test model in all directions while holding it at the test location (directly above the center of the attitude determination test rig) for a period of 20 seconds. care must be taken to not touch the reaction wheels, rotate the test model too quickly or move it from the test location. |   |
| 5    | Place the test model back on the attitude determination test rig and set "Filter mode" to 1.   |   |
| 6    | It is recommended to repeat the previous three steps a number of times to get a view if the variance in results and be able to select a good dataset.  |   |
| 7    | Stop the test and retrieve the data.   |   |
| 8    | Open the \data_analysis_new\Sphere_center_calculation\Raw_data_files directory. Empty the directory and replace it with the data file you just created.  |   |
| 9    | Open and run ..\Sphere_center_calculation\Parsing_main.m. The size of the TestData struct in the workspace should be the number of datasets you collected.   |   |
| 10   | Send the following MATLAB command:<br><code>plot(TestData(1).Mag_x)</code>   | The plot should indicate where the data should be truncated. Identify the data point numbers (X value) that indicate the starting and ending truncation points. |

---

11 Open ..\Sphere\_center\_calculation\Parsing\_main.m. Edit the TruncationMatrix variable such that TruncationMatrix(1,1) is the starting truncation point number and TruncationMatrix(1,2) is the ending truncation point number. Then run the script again.

---

12 Send plot(TestData(1).Mag\_x) again to confirm that the truncation was successful.

---

13 Repeat the previous three steps for TestData(i) and TruncationMatrix(i,:) for i = 2:length(TestData)

---

14 Open and run ..\Scsphere\_center\_calculation\SphereCenterCalculation.m. Changing the DatasetNumber variable to switch between datasets. Use your judgment to select the best dataset to take as reference. The BestCenter variable provides the optimal sphere center position.

Optimal sphere center position (**BestCenter**):  
X \_\_\_\_\_  
Y \_\_\_\_\_  
Z \_\_\_\_\_

Standard Deviation of vector magnitude  
(**std**): \_\_\_\_\_

---

**G. Acceptance Checklist****ACS Validation Test Sphere Center Correction Acceptance Checklist**

Project: PolarCube

Date:

Subsystem Revision: Silver

Passed (Y/N):

**ACS Validation Test Sphere Center Correction Acceptance Checklist**

| Item   | Actual | Required  | Verification Type | Verification Notes |
|--|--------|-----------|-------------------|--------------------|
| A dataset of unbiased magnetometer readings from the ACS Validation test model was collected as it was continuously rotated in all directions at the location of the string suspension tests | _____  | ( Y / N ) | Inspection        |                    |
| Optimal magnetometer bias values were determined based on the aforementioned data.   | _____  | ( Y / N ) | Analysis          |                    |
| All Requirements Met   | _____  | ( Y / N ) |                   |                    |

**Overall Test Notes:**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## VI. Attitude determination test

### A. Objective

This procedure describes the method used to characterize the performance of the test model's attitude determination system in determining heading as a function of true heading angle.

### B. Methodology

The test model is subjected to a set of known heading angles and the heading as determined by the test model is recorded alongside the true heading. The difference between the true heading and the measured heading will represent the heading error. True heading will be determined using the attitude determination test rig. The test rig consists of a turntable mounted on a protractor. This allows the test engineer to know the difference in true heading between two attitude measurements. This test method assumes that the first measurement that is made is accurate and then records the true heading relative to the first measurement. Data collection and processing is handled by the ADtest.m MATLAB script.

For the same reason as the sphere center correction, this test should be conducted at the location where the string suspension tests will take place. The attitude error curve will vary depending on the strength and direction of the ambient magnetic field. For best attitude determination performance, the test should be conducted in the same place as, and directly after a sphere center correction. This has been shown to provide the best heading determination performance. This test only determines heading determination performance for a single rotation axis. The test should be repeated each time string tests are to be conducted about a different body frame axis.

This procedure assumes that testbed setup has been conducted and the attitude determination test rig is mounted at the test location.

### C. Hardware List

| Subassembly        | Component                       | Quantity |
|--------------------|---------------------------------|----------|
| <b>Test Models</b> |                                 |          |
|                    | ACS Validation Test Model       | 1        |
| <b>Testbed</b>     |                                 |          |
|                    | Attitude determination test rig | 1        |
|                    | Hook with an eyelet             | 2        |
|                    | Fishing line                    | 1        |
|                    | Interface computer              | 1        |
|                    | XBee                            | 1        |
|                    | XBee Explorer                   | 1        |

### D. Software List

Git repository address: <git@thinker.colorado.edu:polarcube/acs-validation-testing.git>

The test engineer should have a copy of the full repository on their computer. This directories listed are required in their entirety. The procedure only mentions the files that the test engineer will have to interact with and are referred to in the procedure. Many of these files require peripheral files that are included in the directories, but not mentioned in the procedure.

| Subsystem                 | File/Directory                                | Location  |
|---------------------------|---|---|
| <b>Interface computer</b> |   |   |
|                           | Validation testing GUI                        | GUI\OPGUI9_Maxim.py                                   |
|                           | Attitude determination test parsing directory | ..\data_analysis_new\<br>\Attitude_determination_test |

## E. List of Applicable Requirements

| Req. Number | Description   | Verification method                        |
|-------------|---|--|
| 1.ACSTV.15  | Attitude determination shall function indoors on Earth's surface. | ACS passes the attitude determination test |
| 1.ACSTV.18  | Attitude determination inaccuracy shall be below a TBD threshold. | ACS passes the attitude determination test |

## F. Procedure

| Step | Description   | Pictures/Notes  |
|------|---|---|
| 1    | Define the orientation that the test model will be tested in. This is expressed as the body frame axis that will point upwards parallel to the string.  | Body frame rotation axis: _____   |
| 2    | Place the test model on the turntable of the test rig in the same orientation that it will be for the following string suspension tests. Use the wires on the test rig to tie the test model down. Start a test.      |  |
| 3    | Set the reaction wheel speed targets to 100 rad/s. Send the gyro bias values that were determined in integration and the magnetometer bias values that were determined in the sphere center calculation.              |   |
| 4    | Set the turntable dial to 0° on the protractor. It's important that this is done very accurately. When the wheels reach their target speed, set "Target heading" to 1. This will provide reference points in parsing. |   |
| 5    | Wait a few seconds, then set "Target heading" to 0.   |   |
| 6    | Turn the table by +20°, again very accurately, then set "Target heading" to 1.  |   |
| 7    | Wait a few seconds, then set "Target heading" to 0.   |   |
| 8    | Repeat the previous two steps 16 times. Then stop the test and retrieve the data.   |   |

- 
- 9 In order to verify attitude determination, the process up to this point must be repeated at least once to show that the mapping of measured heading to true heading remains constant (to a low degree of uncertainty). This shows that we can expect the mapping to remain the same while the string tests are happening. It would ideally be repeated after a validation test. If attitude determination has already been verified for the current hardware configuration, as it was with the original tests, the test engineer could chose not to repeat the test.
- 
- 10 Open the data analysis directory. If tests are going to be conducted about several axes, separate directories should be created for attitude determination tests analysis in each orientation. Either re-name or name a copy of ..\Attitude\_determination\_test for the body frame rotation axis that the test was conducted about (eg. ..\Attitude\_determination\_test\_minusZ). In further documentation, this directory will be referred to as ..\Attitude\_determination\_test\_new.
- 
- 11 Open ..\Attitude\_determination\_test\_new\Raw\_data\_files, delete the contents and add the data file(s) that you just created. If the test was repeated, all of the output files can be added.
- 
- 12 Open and run ..\Attitude\_determination\_test\Parsing\_main.m. The size of TestData should be equal to the total number of data points collected (eg. 18 for one test, 36 for two tests). If it isn't then an error was made, the test will have to be repeated or the ADtest.m script will have to be adapted to accommodate a missed data point somewhere.
- 
- 13 Open ..\Attitude\_determination\_test\ADtest.m. Edit the Breakpoints variable to identify where the script should start a new test. For one test with 18 points, Breakpoints should be [1 19], for two tests, it should be [1 19 37] etc...
- 
- 14 Run \ADtest.m. A plot should appear with the true and measured headings as well as one with heading error as a function of heading. Heading determination performance is determined by the range of heading error and repeatability of test results.

Range of heading error: \_\_\_\_\_

Largest difference between two heading measurements at the same location: \_\_\_\_\_

**G. Acceptance Checklist****ACS Validation Attitude Determination Test Acceptance Checklist**

Project: PolarCube

Date:

Subsystem Revision: Silver

Passed (Y/N):

**ACS Validation Attitude Determination Test Acceptance Checklist**

| <b>Item</b>  | <b>Actual</b> | <b>Required</b> | <b>Verification Type</b> | <b>Verification Notes</b> |
|--|---------------|-----------------|--------------------------|---------------------------|
| A minimum of two sets of measured heading samples were collected for a set of known angles representing a full rotation about the local vertical, separated by a minimum of one power cycle. | _____         | ( Y / N )       | Inspection               |                           |
| Heading determination error as a function of true and measured heading was determined for a stationary test model based on the aforementioned data.  | _____         | ( Y / N )       | Analysis                 |                           |
| The range of heading determination error was within reasonable limits (TBD)  | _____         | ( Y / N )       | Analysis                 |                           |
| Difference in measured heading between the two data collections was within performance requirements (TBD).   | _____         | ( Y / N )       | Analysis                 |                           |
| <b>All Requirements Met</b>  | _____         | ( Y / N )       |                          |                           |

**Overall Test Notes:**



## VII. String Torque Characterization

### A. Objective

This procedure describes the method used to characterize the external torques that the string suspension testbed will be imparting on the test model. The testbed is expected to constrain motion to rotation about the local vertical and act as a damped rotational spring. This test will produce a measured heading response of the test model to release at a deflection angle. A damped-spring model of the testbed will be adjusted to match the measured response, providing an indication of the spring constant, damping constant and equilibrium heading of the testbed. This will be critical in analyzing data from the validation tests.

### B. Methodology

The test model is suspended at the test location and allowed to oscillate until it comes near rest. The test model is then deflected significantly from the equilibrium position, data logging is initiated and the test model is released to oscillate until it again comes to rest, at which point data logging is terminated. This is repeated a few times in order to get a sufficient number of control samples. The heading data is then fit to a dynamic model of the testbed and the equilibrium angle, spring constant and damping constant of the current string suspension setup are determined. This must be repeated each time that the test model is suspended because properties will change due to slight variations in rigging. In order to have best confidence in the test results, an attitude determination test should be conducted at the test location immediately before setting up the testbed.

### C. Hardware List

| Subassembly        | Component                 | Quantity |
|--------------------|---------------------------|----------|
| <b>Test Models</b> |                           |          |
|                    | ACS validation test model | 1        |
| <b>Testbed</b>     |                           |          |
|                    | Spool of fishing line     | 1        |
|                    | Interface computer        | 1        |
|                    | XBee                      | 1        |
|                    | XBee Explorer             | 1        |

### D. Software List

Git repository address: <git@thinker.colorado.edu:polarcube/acs-validation-testing.git>

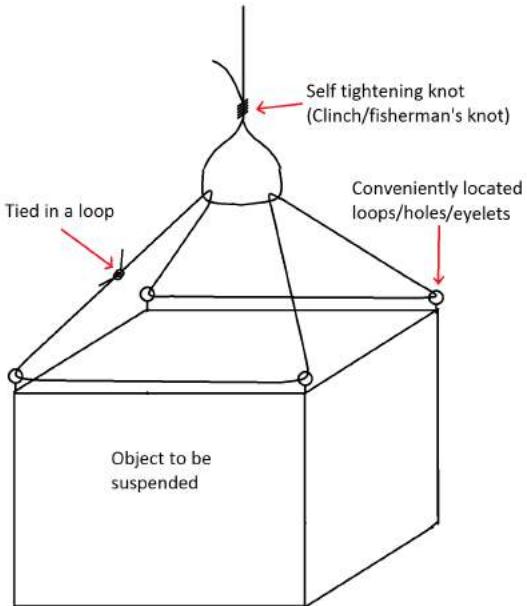
The test engineer should have a copy of the full repository on their computer. This directories listed are required in their entirety. The procedure only mentions the files that the test engineer will have to interact with and are referred to in the procedure. Many of these files require peripheral files that are included in the directories, but not mentioned in the procedure.

| Subsystem                 | File/Directory                            | Location                              |
|---------------------------|---|---------------------------------------|
| <b>Interface computer</b> |   |                                       |
|                           | Validation testing GUI                    | GUI\OPGUI9_Maxim.py                   |
| <b>Data analysis</b>      |   |                                       |
|                           | Oscillation test data analysis repository | ..\data_analysis_newOscillation\_test |
|                           | CAD Model of the Test model               |                                       |

## E. List of Applicable Requirements

| Req. Number | Description   | Verification method  |
|-------------|---|--|
| 1.ACSVT.4   | Disturbing torques on the test model shall not exceed TBD Nm.         | Determined $k$ is below the required threshold.  |
| 1.ACSVT.11  | Disturbing torques on the test model shall be known to within TBD Nm. | Oscillation test results match model simulation output to within TBD values of $k$ , $c$ and $\psi_{eq}$ . |

## F. Procedure

| Step | Description  | Pictures/Notes   |
|------|--|--|
| 1    | Attach a rigging loop to the test model such that it will rotate about the appropriate axis when suspended. The original test model had convenient holes at the corners to use. Tying knots with fishing line can be problematic. It is recommended to research fishing knots. |   |
| 2    | If the attitude determination test rig is still suspended, it's convenient to place the test model on the rig while tying it up. It frees up both hands and keeps the test model in the same position that the attitude determination test was conducted in.                   |  |

---

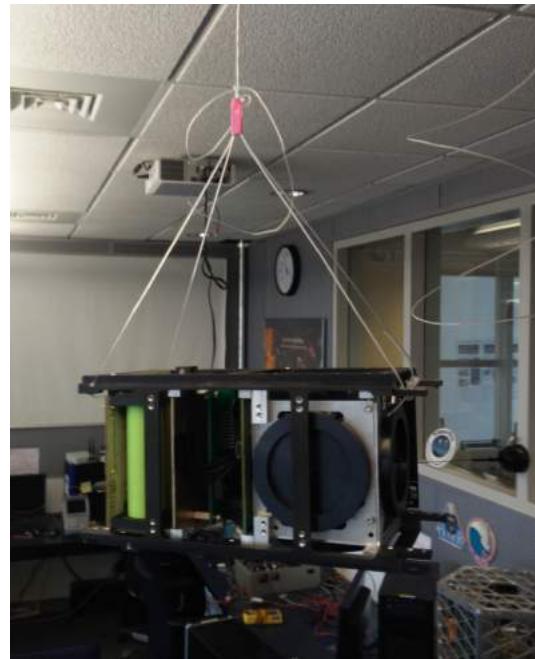
3 Hang the test model from the string by tying the string around two opposing lobes of the rigging loop. (See picture in step 1)

---

4 Start a test. Use the pitch and roll feed from the GUI downlink stream to align the model. Upload the gyro biases determined in integration and the magnetometer biases determined in sphere center calculation. Set reaction wheel target speeds to 100.

---

5 When the test model is aligned, constrain the rigging loop so that it is unlikely to shift in the future. Wrapping duct tape directly under the knot has proven somewhat successful.



---

6 Allow the test model to come to rest at the equilibrium position. Record the measured equilibrium heading somewhere, this will speed up fitting of the software model later. The equilibrium heading would ideally be between  $\pm\frac{\pi}{2}$ . If not, then consider starting over from step 2 and tying the model up a different way.

---

7 Deflect the model significantly (slightly more than a half turn) from the equilibrium. This should only consist of rotation about the string axis. If the rigging loop shifts in the knot, the test will likely be compromised.

---

8 Release the test model and immediately set "Target heading" to 1. This will provide reference points to the parsing algorithm. The test model should only be rotating and not swinging.

---

9 Allow the test model to oscillate until it has essentially stopped, then set "Target heading" to 0.

- 
- 10 Repeat from step 7 three times, then stop the test and retrieve the test data. If further attitude control validation tests are to be conducted upon the basis of results of this test, testbed properties must be preserved. This means that extreme care must be taken to not shift the rigging when inhibiting EPS and removing (and eventually replacing) the SD card. The test engineer might consider moving directly to the validation test without touching the test model and trusting that the characterization test was successful. This process would benefit greatly from improved rigging fixation methods.
- 
- 11 Open the data analysis directory. If validation tests are going to be conducted about several axes, separate directories should be created for string torque characterization in each orientation. Name a copy of ..\Oscillation\_test for the body frame rotation axis that the test was conducted about (eg. ..\Oscillation\_test\_minusZ). In further documentation, this directory will be referred to as ..\Oscillation\_test\_new.
- 
- 12 Open the ..\Oscillation\_test\_new\Raw\_data\_files directory. Empty the directory and replace it with the data file you just created.
- 
- 13 Open and run ..\Oscillation\_test\Parsing\_main.m. The size of the **TestData** struct in the workspace should be the number of datasets you collected.
- 
- 14 Send the following MATLAB command:  
`plot(TestData(1).Heading)`
- The data should be truncated to start at a local maximum or minimum of heading. There might also require truncation at the end if the test model was disturbed. Identify the data point numbers (X value) that indicate the starting and ending truncation points.
- 
- 15 Open ..\Oscillation\_test\Parsing\_main.m. Edit the **TruncationMatrix** variable such that **TruncationMatrix(1,1)** is the starting truncation point number and **TruncationMatrix(1,2)** is the ending truncation point number. Setting any value in **TruncationMatrix** to 0 implies no truncation is necessary.
-

- 
- 16 Run ..\Oscillation\_test\Parsing\_main.m again. Send plot(TestData(1).Heading) again to confirm that the truncation was successful.
- 
- 17 Repeat the previous three steps for TestData(i) and TruncationMatrix(i,:) for i = 2:length(TestData)
- 
- 18 Copy ..\data\_analysis\_new\Attitude\_determination\_test\_new\AD\_results.mat into ..\data\_analysis\_new\Oscillation\_test\_new. This will provide reference for attitude error correction.
- 
- 19 Open ..\Oscillation\_test\_new\MatchTestToSim.m and set EquilibriumHeading to the value that you recorded in step 6.
- 
- 20 The original MatchTestToSim.m script will have the MMOI of the test model used in the original test. If the test model structure has changed, then retrieve the MMOI tensor matrix from a CAD model of the test model and edit the IB variable in the script accordingly.
- 
- 21 Run the script. Plots of heading response and testbed software model output should appear for all of the tests conducted.
- 
- 22 In MatchTestToSim.m, Adjust the EquilibriumAngle, SpringConstant and DampingConstant variables to best fit the test measurements.
- 
- 23 In order for the test to be considered successful, the two curves should fit extremely closely. For reference, see the thesis results about the -Z axis or run the original ..\data\_analysis\Oscillation\_test\MatchTestToSim.m script.

Test model MMOI tensor matrix. :

$$\begin{bmatrix} \text{_____} & \text{_____} & \text{_____} \\ \text{_____} & \text{_____} & \text{_____} \\ \text{_____} & \text{_____} & \text{_____} \end{bmatrix}$$

Equilibrium angle: \_\_\_\_\_

Spring constant: \_\_\_\_\_

Damping constant: \_\_\_\_\_

Successful fit:

**G. Acceptance Checklist****ACS Validation Oscillation Test Acceptance Checklist**

Project: PolarCube

Date:

Subsystem Revision: Silver

Passed (Y/N):

**ACS Validation De-tumble Test Acceptance Checklist**

| Item  | Actual | Required  | Verification Type | Verification Notes |
|---|--------|-----------|-------------------|--------------------|
| Data logs have been recorded of four heading responses to release at a deflection angle of the test model on the string suspension testbed. | _____  | (Y/N)     | Inspection        |                    |
| The aforementioned data logs have successfully been fit to a model of string testbed dynamics.  | _____  | (Y/N)     | Analysis          |                    |
| <b>All Requirements Met</b>   | _____  | ( Y / N ) |                   |                    |

**Overall Test Notes:**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---