

Copyright
by
Margaret Hoi Ting Tam
2015

The Thesis committee for Margaret Hoi Ting Tam
Certifies that this is the approved version of the following thesis:

An Attitude Determination and Control System for Small Satellites

APPROVED BY

SUPERVISING COMMITTEE:

Wallace Fowler, Supervisor

E. Glenn Lightsey, Co-Supervisor

An Attitude Determination and Control System for Small Satellites

by

Margaret Hoi Ting Tam, B.S.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2015

To my parents and my sister, for getting me here and keeping me going.

Acknowledgments

I would like to acknowledge and thank the following people for their contributions to this work:

My advisor, Dr. E. Glenn Lightsey, for introducing me to satellite control systems and giving me the opportunity to fly one in space. Dr. Wallace Fowler, for his insightful comments and feedback on this thesis.

Dr. Behcet Acikmese, for introducing me to convex optimization and advising me on the constrained attitude guidance problem. Dr. Benito Fernandez, for instructing me in the principles of sliding mode control.

Dr. Henri Kjellberg, for being a wonderful mentor, paving the way before me, and answering my endless stream of questions. Thank you for the knowledge and experience you imparted, the thought-provoking conversations, and the solid advice.

Parker Francis, Cody Colley, Sean Horton, David Kessler, and Terry Stevenson for all the hours you poured into these satellites. They would be in pieces without you. Thank you for being there to answer my questions and fix my problems.

All the members of the Texas Spacecraft Laboratory, for making the lab a wonderful work environment, all of the work that you do, and everything you taught me.

An Attitude Determination and Control System for Small Satellites

Margaret Hoi Ting Tam, M.S.E.

The University of Texas at Austin, 2015

Supervisors: Wallace Fowler
E. Glenn Lightsey

A flexible, robust attitude determination and control (ADC) system is presented for small satellite platforms. Using commercial-off-the-shelf sensors, reaction wheels, and magnetorquers which fit within the 3U CubeSat form factor, the system delivers arc-minute pointing precision. The ADC system includes a multiplicative extended Kalman filter for attitude determination and a slew rate controller that acquires a view of the Sun for navigation purposes. A pointing system is developed that includes a choice of two pointing controllers – a proportional derivative controller and a nonlinear sliding mode controller. This system can reorient the spacecraft to satisfy a variety of mission objectives, but it does not enforce attitude constraints. A constrained attitude guidance system that can enforce an arbitrary set of attitude constraints is then proposed as an improvement upon the unconstrained pointing system. The momentum stored by the reaction wheels is managed using magnetorquers to prevent wheel saturation. The system was thoroughly tested in realistic software- and hardware-in-the-loop simulations that included environmental disturbances, parameter uncertainty, actuator dynamics, and sensor bias and noise.

Table of Contents

Acknowledgments	v
Abstract	vi
List of Tables	x
List of Figures	xi
Chapter 1. Introduction	1
1.1 Motivation	2
1.1.1 CubeSat Form Factor	3
1.1.2 RACE	3
1.1.3 Bevo-2	5
1.1.4 ARMADILLO	7
1.2 Overview	9
Chapter 2. ADC Module Hardware	11
2.1 Sensors	11
2.1.1 Magnetometers	12
2.1.2 Gyroscopes	13
2.1.3 Sun Sensors	14
2.1.4 Star Tracker	16
2.2 Actuators	17
2.2.1 Reaction Wheels	17
2.2.2 Magnetorquers	22
2.3 Component Summary	23
Chapter 3. Simulation Environment	25
3.1 State Propagation	25
3.1.1 Orbit Propagation	27
3.1.2 Attitude Dynamics	28

3.2	Environmental Models	31
3.2.1	Atmospheric Drag	31
3.2.2	Solar Radiation Pressure	33
3.2.3	Residual Magnetic Dipole	34
3.2.4	Gravity Gradient	36
3.3	Satellite Parameters	36
Chapter 4. Controllability and Observability		38
4.1	Slew Rate	38
4.2	Quaternion	42
4.3	Sun Acquisition Controller	45
Chapter 5. Navigation Filter		48
5.1	A Priori Estimate	48
5.2	Measurement Update Step	50
5.3	Flying the Gyros	53
5.3.1	Attitude Error Accumulation	55
Chapter 6. Unconstrained Pointing		58
6.1	Pointing Manager	58
6.1.1	Pointing Modes	58
6.1.1.1	LVLH Target	60
6.1.1.2	LLA Target	61
6.1.2	Desired Attitude	62
6.1.3	STK Verification	64
6.2	Pointing Controllers	65
6.2.1	Proportional Derivative Controller	66
6.2.2	Sliding Mode Controller	68
6.2.2.1	Quaternion Control	69
6.2.2.2	Slew Rate Control	71
6.2.3	Performance Analysis	73
6.2.4	Robustness Analysis	76

Chapter 7. Constrained Attitude Guidance	79
7.1 Convex Optimization Approach	80
7.2 Problem Statement	81
7.3 Convex Parametrization of Pointing Constraints	85
7.4 Discretization of Nonlinear Dynamics	88
7.5 Convex Objective Function	89
7.6 Simulation Results	94
Chapter 8. Momentum Management	98
8.1 Reaction Wheel Saturation	98
8.2 Momentum Controller	100
8.3 Simulation Results	102
8.4 Lifetime Momentum Management	103
Chapter 9. Conclusions	106
9.1 Summary of Work	106
9.2 Current and Future Work	107
Bibliography	110

List of Tables

2.1	ADIS16251 Gyroscope Parameters	14
2.2	Reaction Wheel Imbalances	21
2.3	ADC Components	23
3.1	Physical Parameters	36
3.2	RACE Mass Properties	37
6.1	Target Vectors in Each Pointing Mode	59
6.2	Controller Performance	76

List of Figures

1.1	Poly Picosatellite Orbital Deployer (P-POD) and Cross-section [1]	3
1.2	RACE inside a P-POD	4
1.3	LONESTAR-2 Satellites	6
1.4	Bevo-2 Concept of Operations	6
1.5	ARMADILLO CAD Model	7
1.6	ARMADILLO Concept of Operations	9
2.1	Gyroscope Bias Behavior [2]	14
2.2	Reaction Wheel Mass Imbalances	20
2.3	Bevo-2 CAD Models	24
3.1	Block Diagram of the Simulation Environment [3]	26
3.2	RACE Testing inside a Helmholtz Cage	35
4.1	Sun Acquisition Simulation Results	47
5.1	Attitude Estimation Error During Loss of Observability	57
6.1	Pointing Manager Overview	59
6.2	STK Pointing Verification	65
6.3	Pointing Controller Simulation Results	75
6.4	Controller Robustness to Disturbance and Sensor Noise	77
6.5	PD Controller Performance with MOI Uncertainty	78
7.1	Sun Keep-Out Attitude Constraint	83
7.2	Bounded Control and Slew Rates	96
7.3	Pointing Error and Pointing Constraints	97
8.1	Pointing Accuracy with Varying Wheel Momentum	99
8.2	Momentum Management Deadband Logic	102
8.3	Momentum Management Simulation Results	104

8.4	Momentum Accumulation over an Orbit at Varying Altitudes	105
9.1	ADC Hardware-in-the-Loop Testing	109

Chapter 1

Introduction

Small satellites (of under 100 kg) are becoming increasingly popular due to the significant reductions in cost and delivery time that they offer. They are also becoming increasingly capable as advancements in satellite technology have scaled instruments down in size and mass, allowing them to fly on these smaller platforms. These vehicles can now provide easy access to space for a variety of purposes, such as science experiments, remote communications, space exploration, and reconnaissance. For example, the United States Army Space and Missile Defense Command is beginning to employ the small satellites SNaP and Kestrel Eye for beyond-line-of-sight communications and situational awareness [4, 5]. The Space Systems Laboratory at the Massachusetts Institute of Technology is working on the nanosatellite ExoplanetSat, which will detect exoplanet transits [6]. Small satellites also enable the use of satellite constellations (coordinated networks of multiple satellites) which would otherwise be infeasible due to the cost of the numerous satellites. These constellations can provide near-constant coverage of portions of the Earth. Planet Labs launched one such fleet of 28 small satellites from the International Space Station to image the Earth with 3 to 5 meter resolution [7].

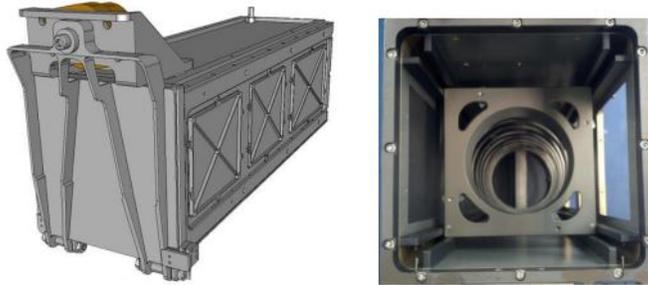
These missions of growing complexity often require the satellite to image a celestial body, direct a high-gain antenna at a ground station, or point a science instrument to collect data. The ability to precisely and reliably reorient in space is essential to meeting these mission objectives. This capability is delivered by the spacecraft's attitude determination

and control (ADC) system, which is responsible for 1) determining the current attitude (or orientation) through a combination of sensor measurements, 2) computing an appropriate control signal to send to the actuators, and 3) using the actuator devices to change the spacecraft's orientation as desired. The goal of this thesis is to present the design, development, and simulation of a complete ADC system for small satellite platforms. This chapter discusses the motivating spacecraft missions behind this research and an outline of the thesis.

1.1 Motivation

This research was primarily conducted to develop the ADC system that will fly on two 3U CubeSats currently in development at the Texas Spacecraft Laboratory (TSL) at the University of Texas at Austin. This section provides an overview of the CubeSat form factor followed by a description of these two satellites, Bevo-2 (an autonomous rendezvous technology demonstration) and ARMADILLO (Atmosphere Related Measurements And Detection of submILLimeter Objects). A brief description of a previous satellite delivered by the TSL, RACE (Radiometer Atmospheric CubeSat Experiment), is also provided. Since the TSL builds multiple satellites concurrently, the ADC system needs to be modular and reusable across different missions. Thus, the ADC system design was driven by the mission requirements of Bevo-2 and ARMADILLO, but the system could easily be used for any small satellite with similar pointing requirements and an equivalent suite of sensors and actuators.

Figure 1.1: Poly Picosatellite Orbital Deployer (P-POD) and Cross-section [1]



1.1.1 CubeSat Form Factor

Many small satellites follow the CubeSat standard, which was developed by California Polytechnic State University (Cal Poly), San Luis Obispo, and Stanford University's Space Systems Development Lab. Every CubeSat must adhere to the specifications outlined in the CubeSat Design Specification document maintained by The CubeSat Program at Cal Poly. A CubeSat is composed of 10x10x10 cm units of less than 1.33 kg. These units can be stacked together along one axis to form 2-unit (2U) or 3-unit (3U) CubeSats of 20x10x10 cm and 30x10x10 cm, respectively. This standardization allows CubeSats to share a common deployment system called the Poly Picosatellite Orbital Deployer, or P-POD system. A P-POD is a spring-loaded case that can accommodate up to three 1U CubeSats, or a single 3U CubeSat. Since a P-POD takes up minimal space, CubeSats typically piggyback on launches as secondary payloads. This expedites the process of securing a launch vehicle provider and greatly reduces the cost of launch [1].

1.1.2 RACE

RACE is a 3U CubeSat that the TSL designed, built, and tested for the Jet Propulsion Laboratory (JPL) to house their radiometer instrument. The radiometer measures

Figure 1.2: RACE inside a P-POD



electromagnetic radiation at the water vapor absorption line at 183 GHz, providing valuable data on water vapor levels in the upper atmosphere for improving weather forecasting and climate models. The radiometer instrument fits within 1.5U of the CubeSat, with the remaining volume containing the power, communications, data handling, and attitude control systems. To collect the science data, the satellite must maintain a constant spin of 6 rpm about the long axis of the satellite along which the radiometer is oriented. RACE was to be deployed in October 2014 from the NanoRacks CubeSat Deployer on the International Space Station (ISS) after arriving at the station with a Cygnus re-supply mission. The unfortunate loss of the Antares launch vehicle prevented this, but the skills and experience gained from building RACE were invaluable and will be applied to future missions.

Although RACE, Bevo-2, and ARMADILLO have different payloads and mission objectives, their design architecture is nearly identical. Each satellite's suite of sensors and actuators are composed of the same commercial-off-the-shelf (COTS) components, although the exact combination of components varies for each individual satellite. The satellite structure is also very similar across the three satellites aside from differing payload modules. As the only fully tested and delivered 3U CubeSat, RACE is thus used as a baseline for many of the physical parameters in simulation testing of the ADC system (see Section 3.3).

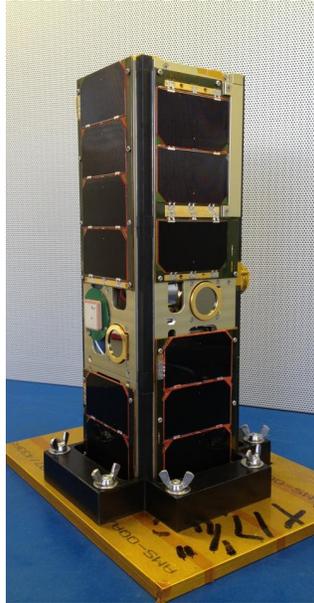
1.1.3 Bevo-2

Bevo-2 will demonstrate the technological capabilities of small satellites as part of NASA Johnson Space Center's LONESTAR-2 (Low-Earth Orbiting Navigation Experiment for Spacecraft Testing Autonomous Rendezvous and Docking) project. This 3U CubeSat has a six degree-of-freedom guidance, navigation, and control (GN&C) system consisting of the ADC system outlined in Section 2.3 and a one degree-of-freedom 3D-printed cold-gas thruster. It will deploy from within AggieSat-4, a <50 kg nanosatellite to be delivered by Texas A&M University, using a CubeSat launcher developed by Innovative Solutions in Space (ISIS). The blue ISIPOD (ISIS Payload Orbital Dispenser) can be seen in Figure 1.3b with its front door open, through which the spring-loaded mechanism that will propel Bevo-2 out into space is visible (refer to [8] for ISIPOD specifications). The two satellites will be delivered to the ISS via a SpaceX CRS-7 cargo resupply mission that is tentatively scheduled for June 2015. The satellites will then deploy from the ISS via the Space Station Integrated Kinetic Launcher for Orbital Payload Systems (SSIKLOPS).

After deploying from the ISS and separating from AggieSat-4, Bevo-2 will establish cross-link communications with AggieSat-4 and send Global Positioning System (GPS) data across the link. The GPS data will be used to perform relative navigation, allowing Bevo-2 to image AggieSat-4 with its star tracker camera. Bevo-2 will then perform a variety of reorientation maneuvers and demonstrate momentum management to test the sensors and actuators. This will be followed by a propulsion test of the cold-gas thruster [9]. After Bevo-2 demonstrates the capabilities of the six degree-of-freedom GN&C system, the satellite will perform a rendezvous maneuver with AggieSat-4. A graphic summary of the concept of operations is presented in Figure 1.4.

Figure 1.3: LONESTAR-2 Satellites

(a) Bevo-2



(b) AggieSat-4 in Integration

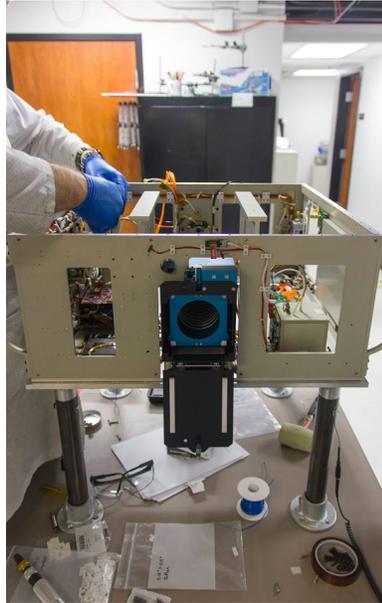
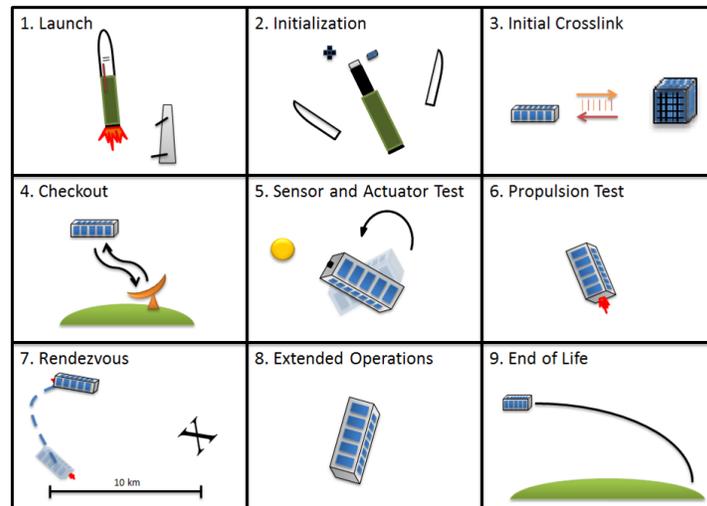


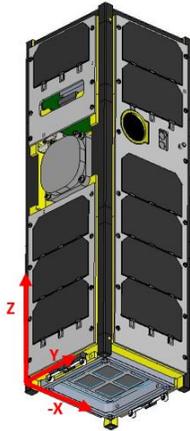
Figure 1.4: Bevo-2 Concept of Operations



1.1.4 ARMADILLO

ARMADILLO was the winning entry submitted by the TSL in the University Nanosat Program-7 (UNP-7) spacecraft competition sponsored by the Air Force Research Laboratory. This 3U CubeSat has an ADC module that is nearly identical to the one on-board Bevo-2 (refer to Section 2.3 for details). ARMADILLO will characterize space debris in Low Earth Orbit by measuring impacts from sub-millimeter particles using a piezo-electric dust detector (PDD). The PDD is being developed by the Center for Astrophysics, Space Physics, and Engineering Research (CASPER) at Baylor University. The instrument collects data regarding the number, mass, and velocity of particles colliding with the instrument by recording the piezo signals that they generate upon impact [10]. To intercept particles for debris detection, the satellite must point the PDD within 5 degrees of the velocity vector. The PDD can be seen as a grid on the -Z face of the spacecraft in Figure 1.5.

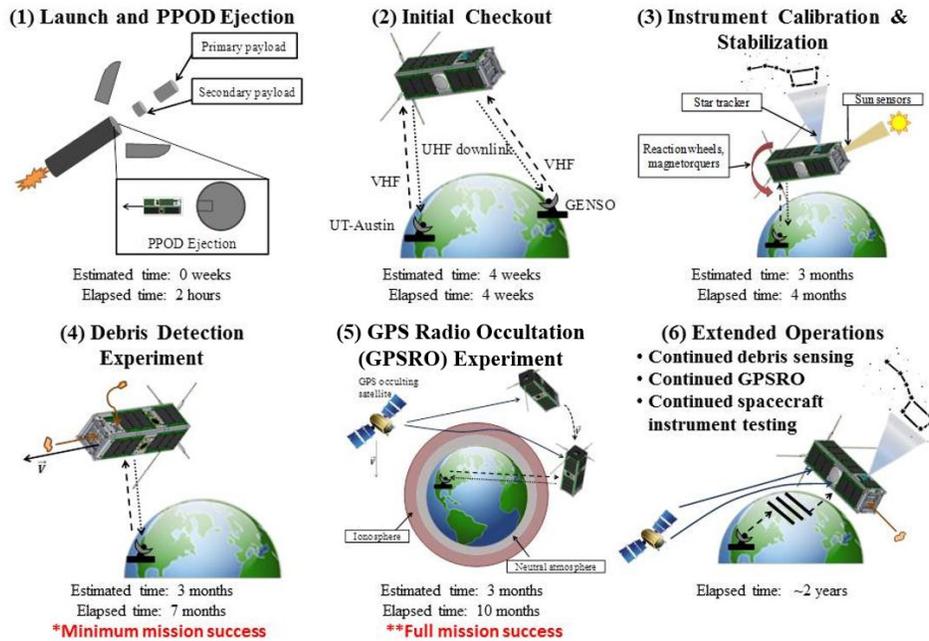
Figure 1.5: ARMADILLO CAD Model



A secondary payload science instrument on-board the satellite is the Fast, Orbital, TEC, Observables, and Navigation (FOTON) GPS receiver developed by the Radionaviga-

tion Laboratory at the University of Texas at Austin. This dual-frequency, software-defined receiver provides radio occultation observations which measure the total electron content (TEC) of the ionosphere. The GPS satellite transmitting the signal appears to set behind ARMADILLO, since the GPS satellite is in a much higher orbit than the CubeSat in low Earth orbit (LEO). During such an event (termed an occultation), the signal travels horizontally through the layers of the ionosphere and neutral atmosphere. As it passes through these layers, the signal is refracted by the free electrons in the ionosphere. This effect can be measured by the receiver to provide an estimate of the TEC along the path traveled by the signal, thus capturing the instantaneous state of the ionosphere. As the relative orientation of occulting GPS satellites and the receiving LEO satellite changes, successive layers of the atmosphere are scanned. These measurements can be used to monitor the state of the ionosphere and improve models of space weather [11]. A diagram of ARMADILLO's concept of operations is presented in Figure 1.6. The satellite is scheduled for launch in February 2016 on a SpaceX Falcon Heavy as part of the Space Test Program 2 (STP-2).

Figure 1.6: ARMADILLO Concept of Operations



1.2 Overview

This thesis is organized in the following fashion:

First, the hardware components that make up the suite of sensors and actuators in the ADC module are introduced along with the limitations that they place on the system. The configuration of components that is used on Bevo-2 and ARMADILLO is presented as well as models for simulating the output from each component. The following chapter describes the simulation environment that is used in conjunction with the sensor and actuator models to assess the performance of the system.

The controllability and observability of the attitude system is then presented, along with a controller that is used to acquire a view of the Sun for navigation purposes. This is followed by a description of the navigation filter that uses sensor measurements to provide an

estimate of the attitude states of the spacecraft. The next chapter outlines the algorithms behind the unconstrained pointing system used to satisfy the pointing requirements of a given mission objective. These algorithms include the pointing manager, which determines the desired orientation of the spacecraft, and the pointing controllers, which reorient the spacecraft into that commanded attitude.

A more sophisticated alternative to the unconstrained pointing system is the constrained attitude guidance (CAG) system presented in the following chapter. This system reorients the satellite while satisfying an arbitrary number of attitude constraints, which are not taken into account by the unconstrained pointing system. Finally, the reaction wheels used by the Sun acquisition, pointing, and CAG control laws must have a method of dissipating momentum, motivating the development of a momentum management system.

Chapter 2

ADC Module Hardware

The ADC system's capabilities and limitations are driven by the hardware components that are available on the satellite. This chapter includes a description of each component within the ADC module, its function, and the constraints it places on the system. Models of each component for simulating its output are provided, which will be used to simulate and test the system. The physical parameters that are specific to the components chosen for the TSL satellites are also presented here. These are the parameters that will be used in simulation testing, however, the parameters for any equivalent component can be substituted into this framework to test an ADC system with a different configuration.

2.1 Sensors

The sensor suite on-board the TSL satellites is composed of commercial-off-the-shelf magnetometers, gyroscopes, and sun sensors, as well as a miniaturized star tracker that was developed in-house by the TSL. Each of these sensors is rigidly attached to the frame of the satellite, so the measurements the sensors provide are in the body frame of the satellite. These measurements can be compared against model-generated estimates of their true values in the Earth-centered inertial (ECI) frame. The measurements and their inertial counterparts are consumed by the navigation filter in order to generate an estimate of the satellite's current attitude and slew rate (see Section 5.2).

2.1.1 Magnetometers

Magnetometers measure the strength and direction of the local magnetic field. Satellites in LEO are still within the magnetic field around Earth, so the field that the magnetometer experiences is mainly the terrestrial one. Note that the magnetic field generated by the electronics on-board the spacecraft must be accounted for when using these magnetic field measurements.

A true inertial magnetic field vector can be generated by using a model of Earth's magnetic field if the position of the satellite is known. The magnetic field measurement from the magnetometer is $\tilde{\mathbf{B}}_{\mathbf{B}} = [\tilde{B}_x \quad \tilde{B}_y \quad \tilde{B}_z]_B^T$, representing each magnetic field component in the spacecraft body frame. The notation $\tilde{}$ signifies a measured quantity and a subscripted B or I denotes the body or inertial frame, respectively. The magnetometer output can be modeled by:

$$\tilde{\mathbf{B}}_{\mathbf{B}} = \mathbf{B}_{\mathbf{B}} + \beta_{\mathbf{mag}} + \eta_{\mathbf{mag}} \quad (2.1)$$

where $\mathbf{B}_{\mathbf{B}}$ is the true magnetic field in the body frame, $\beta_{\mathbf{mag}}$ is the magnetometer bias, and $\eta_{\mathbf{noise}} \sim N(\mathbf{0}_{3 \times 1}, \sigma_{\mathbf{mag}})$ represents zero-mean Gaussian noise in the measurements with a standard deviation of $\sigma_{\mathbf{mag}}$. The magnetometer does not impose any constraints on the system, as it can sample the local magnetic field in any orientation.

TSL-specific parameters

The magnetometer used by the TSL satellites is the HMR2300 Smart Digital Magnetometer by Honeywell. From the data sheet, the HMR2300 magnetometer has typical offset errors and noise values of 0.01%FS, where the full scale (FS) field applied was $\pm 2Gauss$.

Thus, $\beta_{\mathbf{mag}} = [40 \ 40 \ 40]^T nT$ and $\sigma_{\mathbf{mag}} = [40 \ 40 \ 40]^T nT$ [12].

2.1.2 Gyroscopes

A gyroscope measures angular rate about a single axis, so satellites generally use a suite of 3 gyroscopes oriented along each body axis. In such a configuration, the gyroscope measurements form a vector of measured body spin rates, $\tilde{\omega}_{\mathbf{B}} = [\tilde{\omega}_x \ \tilde{\omega}_y \ \tilde{\omega}_z]_B^T$. The gyroscope measurements are simulated using:

$$\tilde{\omega}_{\mathbf{B}} = \omega_{\mathbf{B}} + \beta_{\omega} + \eta_{\omega} \quad (2.2)$$

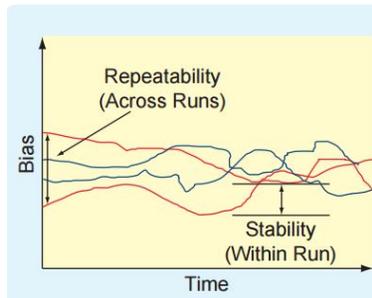
where β_{ω} is the gyro bias and $\eta_{\omega} \sim N(\mathbf{0}_{3 \times 1}, \sigma_{\omega})$ represents zero-mean Gaussian white noise. As shown in Figure 2.1, the gyro bias changes over time as well as every time the gyroscopes are turned on. The variation in bias within one run can be modeled by:

$$\beta_{\omega} = \beta_{\omega}^- + \eta_{\beta} dt \quad (2.3)$$

where $-$ denotes the prior value, dt is the simulation time step, and η_{β} is the random variation in bias over time. η_{β} is modeled by a zero-mean Gaussian distribution with standard deviation σ_{β} given by the in run bias stability. The initial bias $\beta_{\mathbf{0}}$ at the time when the gyroscopes are turned on is a random number sampled from a zero-mean normal distribution with a standard deviation of σ_{β_0} describing the variation in these initial biases across different runs.

$$\beta(t = 0) = \beta_{\mathbf{0}} \sim N(\mathbf{0}_{3 \times 1}, \sigma_{\beta_0}) \quad (2.4)$$

Figure 2.1: Gyroscope Bias Behavior [2]



TSL-specific parameters

The TSL satellites use three ADIS16251 Programmable Low Power Gyroscopes by Analog Devices. Since the same gyroscope is used on each axis, the parameters in Table 2.1 are identical for each component of the associated vectors. The bias parameters were taken from the ADIS16251 data sheet. The output noise was experimentally determined by calculating the standard deviation in gyroscope data that was collected from three ADIS16251 units over a period of 25 minutes. The standard deviations from the three units were then averaged to produce the value tabulated below.

Table 2.1: ADIS16251 Gyroscope Parameters

Parameter	Notation	Value
In Run Bias Stability	σ_{β}	$0.0016 \frac{deg}{sec}$
Turn-On-to-Turn-On Bias Stability	σ_{β_0}	$0.018 \frac{deg}{sec}$
Output Noise	σ_{ω}	$0.0555 \frac{deg}{sec}$

2.1.3 Sun Sensors

A sun sensor is an instrument that measures the azimuth and elevation angles of the Sun to generate a relative vector from the sun sensor to the Sun. If the ECI position of the satellite is known, a true inertial Sun vector can be generated from the positions of the

Earth and the Sun. The measured Sun vector can then be modeled by:

$$\tilde{\mathbf{r}}_{\text{sun},\mathbf{B}} = \begin{bmatrix} \cos(\tilde{el}) \cos(\tilde{az}) \\ \cos(\tilde{el}) \sin(\tilde{az}) \\ \sin(\tilde{el}) \end{bmatrix} \quad (2.5)$$

where \tilde{az} and \tilde{el} are the measured azimuth and elevation, respectively. These measured values are subject to noise based on how accurate the sensor is:

$$\tilde{el} = el + \eta_{el} \quad (2.6)$$

$$\tilde{az} = az + \eta_{az} \quad (2.7)$$

where η_{el} and η_{az} are zero-mean Gaussian noise in the measured angles, and az and el are the true azimuth and elevation angles computed from the true Sun vector in the body frame, $\mathbf{r}_{\text{sun},\mathbf{B}} = [\text{sun}X \quad \text{sun}Y \quad \text{sun}Z]_B^T$.

$$el = \arcsin(\text{sun}Z_B) \quad (2.8)$$

$$az = \arctan\left(\frac{\text{sun}Y_B}{\text{sun}X_B}\right) \quad (2.9)$$

The sun sensor can only generate a Sun vector if the Sun is in the field-of-view (FOV) of the sensor. Thus, the sun sensor places a keep-in attitude constraint on the satellite by requiring that the satellite be oriented such that the Sun vector is kept within its FOV. Also note that sun sensors cannot provide a measurement if the satellite is in eclipse.

TSL-specific parameters

The sun sensor used by the TSL satellites is the SS-411 Two-Axis Digital Sun Sensor by Sinclair Interplanetary. From the data sheet, the SS-411 sun sensor has an accuracy of 0.1 degrees. The simulated Sun vector measurements were generated using $\eta_{el} = \eta_{az} \sim N(0, 0.1deg)$ which results in a more conservative approximation of the total Sun vector angle error of 0.1414deg. The SS-411 sensors have a FOV of 140°, so the angle between the sun sensor boresight and the Sun vector must be less than 70° to obtain a Sun vector measurement [13].

2.1.4 Star Tracker

A star tracker computes relative vectors from the star tracker camera to stars within the camera's FOV. It accomplishes this by taking an image, identifying points in the image above a threshold brightness as stars, computing the vectors to these stars, and using a star catalog to identify the inertial star vectors. The star tracker delivers star vector measurements with arc-minute level accuracy, and it is used for navigation when the satellite is in eclipse and cannot obtain a Sun vector.

Since the star tracker is light-sensitive, it must avoid looking at bright objects such as the Sun and the Moon. This places a keep-out attitude constraint on the satellite by requiring that bright objects be kept out of the FOV of the star tracker. The star tracker also places a maximum slew rate constraint ω_{max} on the satellite, as rapid slewing will cause the star field to smear and result in poor attitude estimates. Given that the star tracker is more accurate than the sun sensor, navigation using a Sun vector when the satellite is not in eclipse is the more strenuous test case for the navigation filter. Thus, the sun sensor model is used for simulation and testing purposes, and a model for simulated star tracker

measurements was not developed for this thesis.

TSL-specific parameters

The star tracker used by the TSL has a FOV of 40° and imposes a maximum slew rate of $2deg/s$ on the satellite. For more details on the star tracker developed in the TSL, please refer to [14].

2.2 Actuators

The actuator suite on-board the TSL satellites is composed of commercial-off-the-shelf reaction wheels and magnetorquers which are used to generate torques on the spacecraft. The controllers in the ADC system compute the torques necessary to reorient the satellite as desired and command the actuators accordingly.

2.2.1 Reaction Wheels

Reaction wheels are spinning cylinders with variable, commandable spin rates. Since the total angular momentum of the system must be conserved, changing the momentum in a reaction wheel results in an equal and opposite momentum change in the satellite.

In the absence of external torques, conservation of total angular momentum can be expressed as:

$$\mathbf{h}_{\text{tot}} = \mathbf{h}_{\text{SC}} + \mathbf{h}_{\text{wheel}} = \mathbf{J}\boldsymbol{\omega} + \mathbf{J}_{\text{wheel}}\boldsymbol{\omega}_{\text{wheel}} = \mathbf{constant} \quad (2.10)$$

where \mathbf{h}_{SC} and $\mathbf{h}_{\text{wheel}}$ are the angular momentum of the spacecraft and reaction wheel assembly, respectively. \mathbf{J} denotes the moment of inertia, and $\boldsymbol{\omega}$ is spin rate. Since the derivative of angular momentum is torque,

$$\dot{\mathbf{h}}_{\text{tot}} = \mathbf{0} = \frac{\Delta \mathbf{h}_{\text{SC}}}{\Delta t} + \frac{\Delta \mathbf{h}_{\text{wheel}}}{\Delta t} = \mathbf{u}_{\text{SC}} - \mathbf{u}_{\text{wheel}} \quad (2.11)$$

where Δt is the differentiation time interval, \mathbf{u}_{SC} is the torque on the spacecraft, and $\mathbf{u}_{\text{wheel}}$ is equal and opposite to the torques on the wheel. The negative sign convention is used so that rearranging this equation gives:

$$\mathbf{u}_{\text{SC}} = \mathbf{u}_{\text{wheel}} \quad (2.12)$$

and $\mathbf{u}_{\text{wheel}}$ represents the torques acting on the satellite due to the reaction wheels. A more complete discussion of the torques acting on the spacecraft is presented in Section 3.1.2.

The reaction wheels place several physical constraints on the spacecraft. The wheels have a maximum spin rate, which limits the amount of momentum they can store to \mathbf{h}_{max} . Commanding additional torque will fail to make the wheels increase their speed beyond this point. This is known as wheel saturation and results in loss of controllability.

There is also a maximum torque that each of the reaction wheels can provide, u_{max} , which limits the amount of control torque available to the ADC system. The torques acting on the satellite from the reaction wheels can be modeled as:

$$\mathbf{u}_{\text{wheel}} = \text{sat}(\mathbf{u}_{\text{commanded}} + \eta_{\text{wheel}}, u_{\text{max}}) + \mathbf{u}_{\text{jitter}} \quad (2.13)$$

where the saturation function is defined as a component-wise saturation of each element in the n -dimensional vector \mathbf{u} by the limit u_{max} .

$$\text{sat}(\mathbf{u}, u_{max}) = \begin{bmatrix} \text{sat}(u_1, u_{max}) \\ \text{sat}(u_2, u_{max}) \\ \vdots \\ \text{sat}(u_n, u_{max}) \end{bmatrix} \quad (2.14)$$

where

$$\text{sat}(u_i, u_{max}) = \begin{cases} -u_{max} & u_i < -u_{max} \\ u_i & |u_i| < u_{max} \\ u_{max} & u_i > u_{max} \end{cases} \quad (2.15)$$

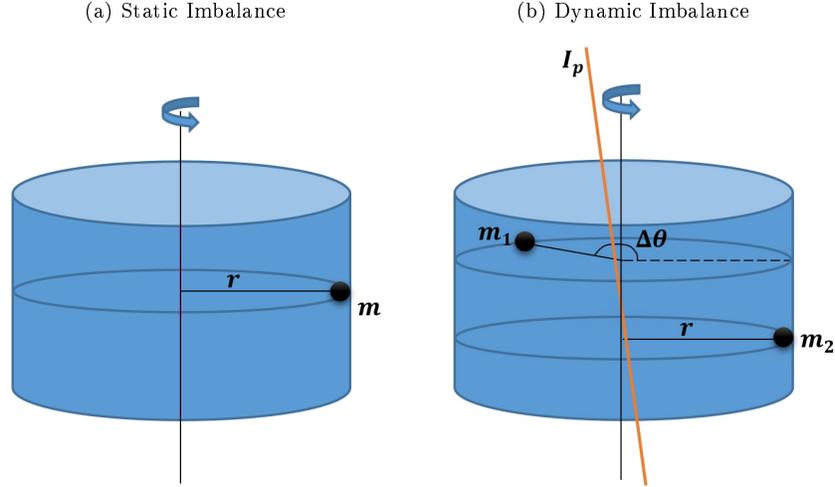
where the subscript i denotes the i th component of the vector. Note that there is an implicit assumption that each reaction wheel on the satellite is identical and thus limited by the same maximum torque.

The jitter torques, $\mathbf{u}_{\text{jitter}}$, are the result of static and dynamic imbalances in the reaction wheels. The static imbalance S of a wheel is due to an uneven distribution of mass about its spin axis, which can be modeled as an equivalent point mass on the edge of a perfect cylinder (see Figure 2.2a). The static imbalance can then be calculated by multiplying this point mass by its distance from the center of mass of the wheel, giving it dimensions of $[mass] \cdot [length]$. As the wheel spins, the mass imbalance exerts a periodic force on the spacecraft. For a reaction wheel oriented along the X body axis, this force can be modeled as:

$$\mathbf{F}_S = \begin{bmatrix} 0 \\ S\omega_{wheel}^2 \cos(\omega_{wheel}t + \phi_S) \\ S\omega_{wheel}^2 \sin(\omega_{wheel}t + \phi_S) \end{bmatrix} \quad (2.16)$$

Similar expressions can be derived for reaction wheels that are oriented along the other two axes. The phase angle ϕ_S can be modeled as a uniform random distribution from $[0, 2\pi]$, since the mass imbalance could be at any angular position about the spin axis at the

Figure 2.2: Reaction Wheel Mass Imbalances



initial time. The resulting torque $\mathbf{u}_{S,i}$ on the satellite due to the static imbalance of the i th wheel is given by:

$$\mathbf{u}_{S,i} = \mathbf{r}_{\text{wheel},i} \times \mathbf{F}_{S,i} \quad (2.17)$$

where $\mathbf{r}_{\text{wheel}}$ is the position of the wheel relative to the spacecraft center of mass.

The dynamic imbalance D of a wheel is due to a misalignment of the inertia tensor of the wheel with the spin axis. It can be modeled as two point masses that have some angular offset $\Delta\theta$ on the edge of a perfect cylinder, where each point mass is on a balancing plane normal to the spin axis. The dynamic imbalance can then be calculated as the magnitude of the resulting products of inertia, giving it dimensions of $[mass] \cdot [length]^2$. The products of inertia of the spinning wheel generate a restoring torque that attempts to realign the nearest principal axis \mathbf{I}_p with the spin axis (see Figure 2.2b). The torque on the satellite due to the dynamic imbalance of a reaction wheel along the X body axis is given by:

$$\mathbf{u}_D = \begin{bmatrix} 0 \\ D\omega_{wheel}^2 \sin(\omega_{wheel}t + \phi_D) \\ -D\omega_{wheel}^2 \cos(\omega_{wheel}t + \phi_D) \end{bmatrix} \quad (2.18)$$

Similar expressions can be derived for reaction wheels that are oriented along the other two axes. The phase angle ϕ_D can once again be modeled as a uniform random distribution from $[0, 2\pi]$. The wheel jitter torques acting on the satellite are given by the summation:

$$\mathbf{u}_{jitter} = \sum_{i=1}^3 (\mathbf{u}_{S,i} + \mathbf{u}_{D,i}) \quad (2.19)$$

TSL-specific parameters

Both Bevo-2 and ARMADILLO use a reaction wheel assembly composed of three orthogonal RW-0.01-4 Picosatellite Reaction Wheels by Sinclair Interplanetary. Each wheel provides a maximum torque of $1mNm$ and can store up to a maximum of $10mNm \cdot s$ of angular momentum. The moment of inertia of each wheel is $2.80e - 5kg \cdot m^2$, so each one has a maximum spin rate of $3410rpm$.

Prior to its delivery to the TSL, the mass imbalance of each reaction wheel unit was measured using two balancing planes. Using the recorded mass imbalances, the static and dynamic imbalances for three of the RW-0.01-4 reaction wheel flight units were calculated using the method described above.

Table 2.2: Reaction Wheel Imbalances

	Wheel 1	Wheel 2	Wheel 3
Static Imbalance ($kg \cdot m$)	9.675e-08	8.675e-08	7.025e-08
Dynamic Imbalance ($kg \cdot m^2$)	3.7141e-10	4.7936e-10	3.1356e-10

From these values, the static and dynamic imbalances in any given Sinclair wheel used on a TSL satellite are expected to be less than $1e - 7 \text{ kg} \cdot \text{m}$ and $5e - 10 \text{ kg} \cdot \text{m}^2$, respectively. These expected bounds are the static and dynamic imbalance values used to simulate the system.

2.2.2 Magnetorquers

Magnetorquers, or torque rods, are essentially electromagnetic coils. Running current through a rod creates a magnetic dipole in the direction along which the rod is oriented. A suite of magnetorquers generates a torque on the spacecraft according to:

$$\mathbf{u}_{\text{mag}} = \text{sat}(\mu_{\text{mag}} \times \mathbf{B}_{\text{B}}, \mu_{\text{max}}) \quad (2.20)$$

where μ_{mag} is the commanded magnetic dipole from the torque rods and μ_{max} is the maximum dipole they can generate. As seen in the equation above, the amount of control torque they can provide to the ADC system is limited by this maximum dipole, as well as the strength and direction of the local magnetic field. Since these devices are simple and provide much less torque than the reaction wheels, the bias and noise in the torque rods are negligible. Magnetorquers are sometimes limited to being fully on or off, making them bang-bang devices that limit the commandable dipole to the following values:

$$\mu_{\text{mag},i} = \begin{cases} -\mu_{\text{max}} \\ 0 \\ +\mu_{\text{max}} \end{cases} \quad (2.21)$$

The magnetic dipole can be set to $-\mu_{\text{max}}$ by switching the direction of the current through the rods.

TSL-specific parameters

The magnetorquers in the TSL ADC module are the ADC916 CubeSat Magnetorquer Rods from SSBV Space & Ground Systems. These torque rods have a maximum magnetic dipole of $\mu_{max} = 0.2Am^2$, and they can only be full on positive, full on negative, or off.

2.3 Component Summary

The combination of components that is specific to the ADC modules on Bevo-2 and ARMADILLO is presented in the Table 2.3. The constraints that each component imposes on the ADC system are also stated, with asterisks * demarcating the attitude constraints. The orientations of the components are given in the body frame of the satellites, where the body axes are defined in Figures 1.5 and 2.3b for ARMADILLO and Bevo-2, respectively.

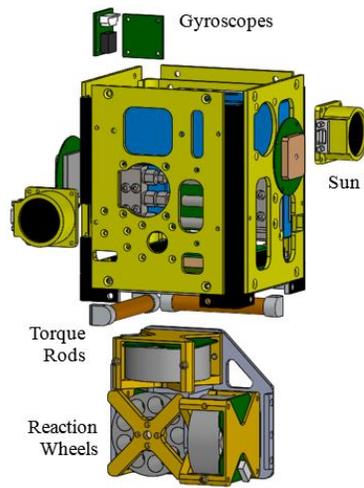
Table 2.3: ADC Components

Component	Bevo-2	ARMADILLO	Constraint
Magnetometer	1	1	N/A
Gyroscope	3 (each axis)	3 (each axis)	N/A
Sun Sensor	2 (+X, +Y)	2 (+Y, -Y)	*Keep Sun within 140° FOV
Star Tracker	1 (-Z)	1 (-X)	Maximum $2deg/s$ slew rate
			*Keep bright objects out of 40° FOV
Reaction Wheel	3 (each axis)	3 (each axis)	Maximum $1mNm$ of torque
			Maximum $10mNm \cdot s$ momentum
Magnetorquer	2 (+X, +Y)	3 (each axis)	Maximum $0.2Am^2$ magnetic dipole

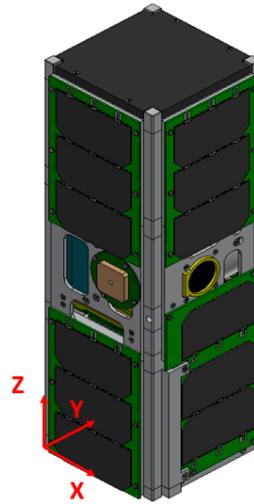
The component layout for the Bevo-2 ADC module is presented in Figure 2.3a. ARMADILLO has a very similar layout with the differences tabulated in Table 2.3.

Figure 2.3: Bevo-2 CAD Models

(a) ADC Component Layout



(b) Body Frame Axes



Chapter 3

Simulation Environment

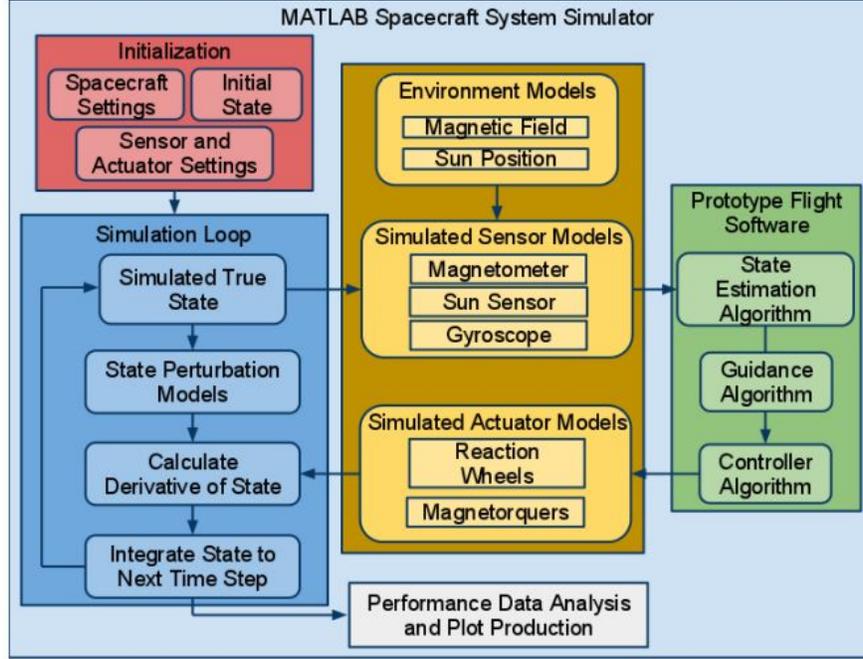
This chapter describes the MATLAB simulation environment that was used to evaluate the on-orbit behavior and performance of the ADC system. The simulation is heavily based on the one developed by Kjellberg [3] and has the architecture that he developed (refer to Figure 3.1).

As outlined in Figure 3.1, the simulator propagates the state of the spacecraft, generates true inertial vectors for the sensor models presented in Section 2.1, and uses those models to provide simulated sensor output to the flight software. The algorithms which make up the flight software are presented in Chapters 5 through 8 and are used to generate the control inputs to the actuator devices. The simulator then uses the actuator models in Section 2.2 to compute the resulting control torques on the spacecraft as well as environmental models for external disturbance torques. These are fed back into the simulation loop as the states are propagated forward in time. The nominal physical parameters of the spacecraft and initial conditions that were used to simulate the ADC system are based on those of the motivating spacecraft missions, ARMADILLO and Bevo-2. The resulting simulation is a robust test environment for the ADC system that includes actuator and sensor dynamics, noise, and biases; environmental perturbations; and realistic physical parameters.

3.1 State Propagation

The spacecraft state is represented by the following vector,

Figure 3.1: Block Diagram of the Simulation Environment [3]



$$\mathbf{x} = [\mathbf{r} \quad \mathbf{v} \quad \mathbf{q} \quad \omega \quad \mathbf{h}_{\text{wheel}}]^T \quad (3.1)$$

where \mathbf{r} and \mathbf{v} are the position and velocity of the spacecraft in the ECI frame, \mathbf{q} is the quaternion attitude of the spacecraft, ω is the spacecraft slew rate, and $\mathbf{h}_{\text{wheel}}$ is the total angular momentum in the assembly of reaction wheels. The spacecraft states are propagated forward in time by numerically integrating the differential equations for each state over each simulation time step:

$$\dot{\mathbf{x}} = \frac{d}{dt} [\mathbf{r} \quad \mathbf{v} \quad \mathbf{q} \quad \omega \quad \mathbf{h}_{\text{wheel}}]^T \quad (3.2)$$

Simulation Parameters

The simulations performed for this thesis used a time step of 0.25s. Given that the maximum slew rate of the spacecraft is limited to 2deg/s, the largest change in attitude over 0.25s would be approximately half a degree. Since the attitude state has the fastest dynamics, this time step was deemed sufficient for capturing the state of the spacecraft with reasonable accuracy.

3.1.1 Orbit Propagation

The equations of motion for the spacecraft position and velocity are governed by simple two-body orbital mechanics:

$$\dot{\mathbf{r}} = \mathbf{v} \quad (3.3)$$

$$\dot{\mathbf{v}} = -\frac{\mu_{Earth}}{\|\mathbf{r}\|^3} \mathbf{r} + \frac{\mathbf{F}_{drag}}{m_{SC}} \quad (3.4)$$

where μ_{Earth} is the gravitational parameter of Earth, \mathbf{F}_{drag} is the drag force acting on the satellite (refer to Section 3.2.1), and m_{SC} is the mass of the spacecraft (refer to Section 3.3). Higher order orbital perturbations such as the gravitational effects of other celestial bodies and the oblateness of Earth are neglected here, since the primary objective of the simulation software is to evaluate attitude determination and control capabilities. Thus, characterizing the attitude of the spacecraft in the space environment is more important than precisely predicting its orbit.

Simulation Parameters

A 400 km, circular orbit with an inclination of 51.65 degrees was used to simulate the system. These nominal orbit parameters were based on the ISS orbit, since CubeSats are often deployed from the ISS.

3.1.2 Attitude Dynamics

The attitude of the spacecraft is represented using quaternions, since they consume less memory than rotation matrices and avoid issues with singularities. For any orientation, there is a corresponding rotation between the base reference frame to which the orientation is relative, and the body frame attached to and rotating with the object. In this case, the orientation of the spacecraft is defined with respect to the ECI frame. The rotation can be broken down into a rotation of angle θ_r about the unit eigenaxis, \mathbf{e}_r . The quaternion is defined in terms of the eigenaxis and angle of rotation as follows:

$$\mathbf{q} = \begin{bmatrix} \mathbf{e}_r \cdot \sin(\theta_r/2) \\ \cos(\theta_r/2) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_v \\ q_4 \end{bmatrix} \quad (3.5)$$

where \mathbf{q}_v and q_4 are the vector and scalar part of the quaternion, respectively.

The kinematic equations for rigid body dynamics as expressed using quaternions are given below:

$$\dot{\mathbf{q}} = \frac{1}{2} \Xi(\mathbf{q})\boldsymbol{\omega} \quad (3.6)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}(-\boldsymbol{\omega}^\times(\mathbf{J}\boldsymbol{\omega} + \mathbf{h}_{\text{wheel}}) + \mathbf{u}) \quad (3.7)$$

$$\dot{\mathbf{h}}_{\text{wheel}} = -\mathbf{u}_{\text{wheel}} \quad (3.8)$$

where \mathbf{u} represents the sum of the torques acting on the satellite.

$$\mathbf{u} = \mathbf{u}_{\text{mag}} + \mathbf{u}_{\text{wheel}} + \mathbf{u}_{\text{ext}} \quad (3.9)$$

The external disturbance torques, \mathbf{u}_{ext} , are computed according to the environmental models described in Section 3.2. The operator Ξ used in Equation 3.6 is defined as:

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 \mathbf{I}_{3 \times 3} + \mathbf{q}^\times \\ -\mathbf{q}^T \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (3.10)$$

The operator

$$[\mathbf{A}^\times] = \begin{bmatrix} 0 & -A_3 & A_2 \\ A_3 & 0 & -A_1 \\ -A_2 & A_1 & 0 \end{bmatrix} \quad (3.11)$$

is the standard skew-symmetric matrix for vector cross product, and $\mathbf{I}_{n \times n}$ is the $n \times n$ identity matrix. The quaternion must satisfy the unity norm constraint in order to preserve length in a rotation.

$$\|\mathbf{q}\| = 1 \quad (3.12)$$

The multiplication of two quaternions is defined as:

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} \Xi(\mathbf{q}) & \mathbf{q} \end{bmatrix} [\mathbf{p}] \quad (3.13)$$

Any quaternion can be converted into a direction cosine matrix (DCM), which is a rotation matrix from the body frame to the ECI frame. Multiplying a vector in the inertial frame by the direction cosine matrix rotates it into the body frame, as shown below.

$$\mathbf{x}_B = \mathbf{I}_B \mathbf{T} \cdot \mathbf{x}_I \quad (3.14)$$

The subscripts I and B represent the vector \mathbf{x} being in the inertial and body frames, respectively, and $\mathbf{I}_B \mathbf{T}$ is the transformation matrix from inertial to body frame. The transformation matrix can be computed from the quaternion as follows:

$$\mathbf{I}_B \mathbf{T} = \mathbf{DCM}(\mathbf{q}) = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (3.15)$$

The quaternion can also be converted into Euler angles, which are the angles of rotation in a series of three single-axis rotations about each body axis. This series of body rotations produces the same rotation matrix as the DCM, since both are representations of the same orientation. The rotation matrix can be computed from the Euler angles by multiplying the rotation matrices for a rotation about the X, Y, and Z body axes together. This gives the following expression relating the DCM to Euler angles:

$$\begin{aligned} \mathbf{I}_B \mathbf{T} &= \mathbf{R}_1(\phi)\mathbf{R}_2(\theta)\mathbf{R}_3(\psi) = \mathbf{DCM}(\mathbf{q}) \\ &= \begin{bmatrix} \cos\psi \cos\theta & \cos\theta \sin\psi & -\sin\theta \\ \cos\psi \sin\phi \sin\theta - \cos\phi \sin\psi & \cos\phi \cos\psi + \sin\phi \sin\psi \sin\theta & \cos\theta \sin\phi \\ \sin\phi \sin\psi + \cos\phi \cos\psi \sin\theta & \cos\phi \sin\psi \sin\theta - \cos\psi \sin\phi & \cos\phi \cos\theta \end{bmatrix} \end{aligned} \quad (3.16)$$

$$\theta = \arcsin(-DCM_{1,3}) \quad (3.17)$$

$$\psi = \arctan\left(\frac{DCM_{1,2}}{DCM_{1,1}}\right) \quad (3.18)$$

$$\phi = \arctan\left(\frac{DCM_{2,3}}{DCM_{3,3}}\right) \quad (3.19)$$

where $\mathbf{R}_i(\theta)$ denotes the rotation matrix for a rotation of angle θ about the i th axis, and ϕ , θ , and ψ are the Euler angles about the X, Y, and Z body axes, respectively. Note that quadrant checks must be performed in order to unambiguously solve for the Euler angles. The angles ψ and ϕ are easily resolved by using the four-quadrant inverse tangent for these computations. For θ , a quadrant check can be manually performed by storing the previous value of θ for comparison.

3.2 Environmental Models

The perturbations acting on the satellite include atmospheric drag, solar radiation pressure, residual magnetic dipole, and gravity gradient torques. Models for each of these external environmental effects are presented in this section.

3.2.1 Atmospheric Drag

Satellites in LEO experience drag from traveling through the Earth's atmosphere. The drag force acting on the satellite in the ECI frame can be calculated according to:

$$\mathbf{F}_{\text{drag}} = -\frac{1}{2}\rho\|\mathbf{v}_{\text{rel}}\|C_D A_{\text{proj},V}\mathbf{v}_{\text{rel}} \quad (3.20)$$

where ρ is the density of the atmosphere at the altitude at which the satellite is traveling, C_D is the satellite's coefficient of drag, and $A_{\text{proj},V}$ is the projected area of the

satellite into the velocity vector. The vector \mathbf{v}_{rel} is the instantaneous velocity of the satellite relative to the rotating atmosphere attached to a spinning Earth in the ECI frame.

$$\mathbf{v}_{\text{rel}} = \mathbf{v} + \begin{bmatrix} \omega_E r_y \\ -\omega_E r_x \\ 0 \end{bmatrix} \quad (3.21)$$

where ω_E is Earth's rotation rate, and r_x and r_y are the X and Y components of the satellite's ECI position, respectively. The drag torque can be calculated by summing over each surface of the satellite:

$$\mathbf{u}_{\text{drag}} = \sum_{i=1}^{N_{\text{faces}}} \mathbf{r}_i \times \mathbf{F}_{\text{drag},i} \quad (3.22)$$

where \mathbf{r}_i is the position of the i th face of the satellite relative to the center of mass, and $\mathbf{F}_{\text{drag},i}$ is the drag force acting on that face in the body frame. $\mathbf{F}_{\text{drag},i}$ is calculated according to Equation 3.20 using the relative velocity vector in the body frame and the projected area of the i th face. \mathbf{v}_{rel} can be rotated into the body frame by using Equation 3.14.

Simulation Parameters

To represent the worst-case drag scenario, the atmospheric density was calculated using the Mass Spectrometer and Incoherent Scatter Radar (MSIS) atmospheric model during maximum solar activity at the nominal orbit altitude of 400 km. This gives a density of $\rho = 7.55e - 12 \text{ kg/m}^3$. A conservative estimate for the coefficient of drag of a cube is 2.5 [15]. The position and area of each face is determined by the 3U CubeSat geometry of the spacecraft.

3.2.2 Solar Radiation Pressure

Incident radiation from the Sun is absorbed and/or reflected by the satellite, transferring momentum from the incoming photons to the surface they hit. The resulting force in the ECI frame due to this solar radiation pressure (SRP) is calculated according to:

$$\mathbf{F}_{\text{SRP}} = -P_{\text{SR}}C_R A_{\text{proj},S} \|\mathbf{r}_{\text{sun}}\| \quad (3.23)$$

The solar radiation pressure P_{SR} is a function of the satellite's distance from the Sun. C_R is the satellite's radiation coefficient which captures the optical properties of the satellite, and $A_{\text{proj},S}$ is the projected area of the satellite into the Sun vector. The SRP torque can be calculated by summing over the SRP forces on each face of the satellite:

$$\mathbf{u}_{\text{SRP}} = \sum_{i=1}^{N_{\text{faces}}} \mathbf{r}_i \times \mathbf{F}_{\text{SRP},i} \quad (3.24)$$

where \mathbf{r}_i is the position of the i th face of the satellite relative to the center of mass, and $\mathbf{F}_{\text{SRP},i}$ is the SRP force acting on that face in the body frame. $\mathbf{F}_{\text{SRP},i}$ is calculated according to Equation 3.23 using the relative Sun vector in the body frame, $\mathbf{r}_{\text{sun},\mathbf{B}}$, and the projected area of the i th face. The solar radiation pressure forces and torques are zero when the satellite is in eclipse.

The SPICE toolkit developed by NASA's Navigation and Ancillary Information Facility (NAIF) is used to compute the relative position of the Sun from the spacecraft in the ECI frame, \mathbf{r}_{sun} [16]. The true inertial vector given by SPICE is rotated into the body frame by using Equation 3.14. The body-frame Sun vector is also used by the sun sensor models in Section 2.1.3.

Simulation Parameters

Earth is at a distance of 1 Astronomical Unit (AU) from the Sun, where the solar radiation pressure is $P_{SR} = 4.59e - 6Pa$. A conservative radiation coefficient based on typical absorptivity and reflectivity values for the satellite is $C_R = 1.5$ [17]. The position and area of each face is determined by the 3U CubeSat geometry of the spacecraft.

3.2.3 Residual Magnetic Dipole

The spacecraft itself has a small magnetic dipole due to current running through the electronics on-board. Additionally, magnetic materials in the construction of the spacecraft often contribute a small dipole bias. This residual magnetic dipole interacts with the magnetic field of the Earth according to:

$$\mathbf{u}_m = \mu_{SC} \times \mathbf{B}_B \quad (3.25)$$

where μ_{SC} is the residual magnetic dipole of the spacecraft, and \mathbf{u}_m is the resulting magnetic dipole torque on the satellite.

The International Geomagnetic Reference Field 11 (IGRF-11) mathematical model of the Earth's magnetic field and its secular variation is used to compute the local magnetic field vector in the ECI frame at the position of the satellite [18]. The true inertial vector given by the IGRF-11 model is rotated into the body frame by using Equation 3.14. The body-frame magnetic field vector is also used in the magnetorquer and magnetometer models in Sections 2.1.1 and 2.2.2.

Figure 3.2: RACE Testing inside a Helmholtz Cage



Simulation Parameters

Measurements of the residual magnetic dipole of the RACE satellite were taken inside a Helmholtz cage which minimized the effects of Earth's magnetic field (see Figure 3.2). The maximum dipole measured had a magnitude of $0.025Am^2$ [19]. The test procedure did not measure the direction of this dipole vector, so an arbitrary unit vector scaling with the dimensions of the spacecraft was assigned.

$$\mu_{SC} = 0.025Am^2 \cdot \frac{\begin{bmatrix} 0.1 & 0.1 & 0.34 \end{bmatrix}^T}{\left\| \begin{bmatrix} 0.1 & 0.1 & 0.34 \end{bmatrix} \right\|} \quad (3.26)$$

Note that only the relative orientation of the dipole vector to the magnetic field matters for torque calculations. Since this relative orientation is constantly changing with the spacecraft's attitude and position, the direction assigned to the residual magnetic dipole vector is not important for assessing the ability of the ADC system to handle this source of disturbance.

Since ARMADILLO and Bevo-2 are structurally similar to RACE, the residual

magnetic dipole of RACE was taken as a reasonable approximation of the expected dipole. Thus for simulation purposes, RACE’s residual magnetic dipole was used with a factor of safety of 2.

3.2.4 Gravity Gradient

The differential gravity force acting over the body of the spacecraft results in a gravity gradient torque \mathbf{u}_{GG} about its center of mass:

$$\mathbf{u}_{GG} = 3 \frac{\mu_{\text{Earth}}}{\|\mathbf{r}\|^3} \mathbf{r}_{B,u} \times \mathbf{J} \mathbf{r}_{B,u} \quad (3.27)$$

where $\mathbf{r}_{B,u}$ is the unit vector position of the satellite in the body frame.

$$\mathbf{r}_{B,u} = \frac{\mathbf{r}_{B,u}}{\|\mathbf{r}_{B,u}\|} \quad (3.28)$$

3.3 Satellite Parameters

The physical spacecraft parameters used in the simulations are based on AR-MADILLO and Bevo-2. These 3U CubeSats have the following dimensions and mass:

Table 3.1: Physical Parameters

Parameter	Value
Width	10cm
Length	10cm
Height	34cm
Mass	4kg

RACE’s center of mass and moments of inertia were measured using a Space Electronics KGR-500 spin table. These measurements are tabulated below.

Table 3.2: RACE Mass Properties

(a) Center of Mass Offset from Geometric Center		(b) Moments of Inertia	
CM	Offset (mm)	Moment of Inertia	Value ($kg \cdot m^2$)
X	1.46	I_{xx}	0.041094
Y	-2.68	I_{yy}	0.040940
Z	3.39	I_{zz}	0.008128

Given that RACE's center of mass is within 4 mm of its geometric center in each axis, the center of mass for the simulated spacecraft is conservatively set at 5mm from the geometric center in each axis.

$$\mathbf{r}_{\text{CM}} = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix} mm \quad (3.29)$$

To account for possible products of inertia, the simulated moment of inertia tensor is based on RACE, but has 5 degrees of misalignment about each axis.

$$\mathbf{J} = \mathbf{R}_1(5deg)\mathbf{R}_2(5deg)\mathbf{R}_3(5deg)\mathbf{J}_{\text{RACE}} \quad (3.30)$$

To account for uncertainty in the moment of inertia, the estimated moment of inertia has an error of 25% and assumes that the principal axes are aligned with the body axes of the spacecraft.

$$\hat{\mathbf{J}} = 0.75\mathbf{J}_{\text{RACE}} \quad (3.31)$$

The notation $\hat{\cdot}$ denotes an estimated quantity.

Chapter 4

Controllability and Observability

This chapter presents the results of the nonlinear controllability and observability analysis performed for the attitude system of the spacecraft. The observability analysis demonstrates that the slew rates of the spacecraft are always observable with the presence of gyroscopes on each axis, whereas attitude observability requires at least two linearly independent sensor measurement vectors.

The controllability analysis demonstrates that the cascade nature of the system dynamics can be exploited to perform nonlinear control on the system. Equation 3.7 shows that the slew rate is directly dependent on the torque, \mathbf{u} , and is thus controllable. The quaternion is dependent upon the slew rate but not the torque, and the slew rate is not dependent on the quaternion. Therefore, the slew rate can be used as the effective control input for quaternion control. This is the control approach that is taken for the nonlinear pointing controller described in Section 6.2.2. The controllability and observability of the slew rates and attitude were analyzed independently by dividing the states of the system into two separate subspaces.

4.1 Slew Rate

The output for the state subspace containing the slew rates is a vector of measured spin rates from the gyroscopes, $\tilde{\omega}$. The gyroscope measurements are modeled using Equation 2.2, which is restated here for convenience:

$$\tilde{\omega}_{\mathbf{B}} = \omega_{\mathbf{B}} + \beta_{\omega} + \eta_{\omega}$$

Farrenkopf's gyroscope model is used to model the bias drift that was described in Section 2.1.2, since this is the model that the navigation filter will use to estimate the slew rates (refer to Chapter 5).

$$\dot{\beta}_{\omega} = \frac{1}{\tau}\beta_{\omega} + \eta_{\beta} \quad (4.1)$$

where τ is the bias time correlation. As $\tau \rightarrow \infty$, the bias becomes a constant. In the equations above, ω is the true (unknown) slew rate. The estimate of the slew rate $\hat{\omega}$ is the expected value of the true slew rate if the estimator is unbiased. Similarly, the estimate of the gyro bias $\hat{\beta}_{\omega}$ is the expected value of the true gyro bias. Also recall that η_{ω} is zero-mean Gaussian noise. Using these relations and rearranging the equation above gives the following expression for $\hat{\omega}$:

$$\hat{\omega} = \mathbf{E}[\omega] = \mathbf{E}[\tilde{\omega}] - \mathbf{E}[\beta_{\omega}] - \mathbf{E}[\eta_{\omega}] = \tilde{\omega} - \hat{\beta}_{\omega} \quad (4.2)$$

Thus, the states of interest are the slew rate and gyroscope bias:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\omega} \\ \hat{\beta}_{\omega} \end{bmatrix} \quad (4.3)$$

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{g}(\hat{\mathbf{x}})\mathbf{u} = \begin{bmatrix} \mathbf{J}^{-1}(-\hat{\omega}^{\times}\mathbf{J}\hat{\omega}) \\ -\frac{1}{\tau}\hat{\beta}_{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{J}^{-1} \\ 0_{3 \times 1} \end{bmatrix} \mathbf{u} \quad (4.4)$$

The output can be expressed in terms of the states:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \tilde{\omega} = \hat{\omega} + \hat{\beta}_{\omega} \quad (4.5)$$

The observability and controllability of a nonlinear system can be determined by computing its Lie brackets and derivatives. The Lie bracket is defined as:

$$[\mathbf{f}, \mathbf{g}] = \frac{\delta \mathbf{g}}{\delta \mathbf{x}} \mathbf{f} - \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \mathbf{g} \quad (4.6)$$

Higher order Lie brackets can be computed according to:

$$\begin{aligned} (\mathbf{ad}_{\mathbf{f}}^1, \mathbf{g}) &= [\mathbf{f}, \mathbf{g}] \\ (\mathbf{ad}_{\mathbf{f}}^2, \mathbf{g}) &= [\mathbf{f}, [\mathbf{f}, \mathbf{g}]] \\ &\vdots \\ (\mathbf{ad}_{\mathbf{f}}^k, \mathbf{g}) &= [\mathbf{f}, (\mathbf{ad}_{\mathbf{f}}^{k-1}, \mathbf{g})] \end{aligned} \quad (4.7)$$

Lie derivatives are defined as:

$$\mathbf{L}_{\mathbf{f}} \mathbf{h} = \frac{\delta \mathbf{h}}{\delta \mathbf{x}} \cdot \mathbf{f} \quad (4.8)$$

where by definition,

$$\mathbf{L}_{\mathbf{f}}^0(\mathbf{h}) = \mathbf{h} \quad (4.9)$$

The controllability matrix is composed of Lie brackets as follows:

$$\mathbf{C} = [\mathbf{g}_1, \dots, \mathbf{g}_n, [\mathbf{g}_i, \mathbf{g}_j], \dots, [\mathbf{ad}_{\mathbf{g}_i}^k, \mathbf{g}_j], \dots, [\mathbf{f}, \mathbf{g}_i], \dots, [\mathbf{ad}_{\mathbf{f}}^k, \mathbf{g}_i], \dots] \quad (4.10)$$

where \mathbf{g}_i denotes the i th column of the function $\mathbf{g}(\mathbf{x})$. For this system, the function $\mathbf{g}(\mathbf{x})$ is composed of the column vectors $\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3$, giving the following controllability matrix:

$$\mathbf{C} = [\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, [\mathbf{ad}_{\mathbf{f}}^1, \mathbf{g}_1], [\mathbf{ad}_{\mathbf{f}}^1, \mathbf{g}_1], [\mathbf{ad}_{\mathbf{f}}^1, \mathbf{g}_1]] \quad (4.11)$$

Computing the controllability matrix by calculating the Lie brackets using the functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ from Equation 4.4 results in a matrix that always has a rank of 3. For example, performing this calculation for the CubeSat moment of inertia given in Section 3.3 gives the following controllability matrix:

$$\mathbf{C} = \begin{pmatrix} 24.2 & -1.93 & 2.3 & 10.2\omega_2 - 0.93\omega_3 & 2.3\omega_1 - 12.8\omega_2 - 19.6\omega_3 & 1.9\omega_1 - 96.2\omega_2 - 2.3\omega_3 \\ 2.1 & 24.3 & -1.9 & 1.94\omega_2 - 12.6\omega_1 + 19.7\omega_3 & 11\omega_1 - 1.6\omega_3 & 97\omega_1 + 2.1\omega_2 - 2.76\omega_3 \\ -10.7 & 10.7 & 122 & 10.7\omega_3 - 7.31\omega_2 - 25.5\omega_1 & 5.4\omega_1 - 14.3\omega_2 + 10.7\omega_3 & 46.9\omega_1 + 59.36\omega_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.12)$$

Thus, the slew rates are controllable and the gyroscope biases are not. Since the biases only need to be estimated, not controlled, this is perfectly acceptable.

Similarly, the observability matrix \mathbf{O} is the gradient of a matrix composed of Lie derivatives:

$$\mathbf{O} = \begin{bmatrix} dL_f^0(h_1) \\ \vdots \\ dL_f^0(h_p) \\ \vdots \\ dL_f^{n-1}(h_1) \\ \vdots \\ dL_f^{n-1}(h_p) \end{bmatrix} \quad (4.13)$$

for a system that has p output functions [20]. For this system, there is only one output function, so the observability matrix is:

$$\mathbf{O} = \begin{bmatrix} dL_f^0(h) = dh \\ dL_f^1(h) \end{bmatrix} \quad (4.14)$$

Computing the observability matrix by calculating the Lie derivatives using the output function defined in Equation 4.5 results in a matrix that is always full rank. For example, the observability matrix for the simulated CubeSat case is:

$$\mathbf{O} = \begin{pmatrix} 1.0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 1.0 \\ -0.087 \omega_2 - 0.017 \omega_3 & 0.13 \omega_2 - 0.087 \omega_2 + 0.79 \omega_3 & 0.79 \omega_2 - 0.017 \omega_1 + 0.031 \omega_3 & -0.001 & 0 & 0 \\ 0.18 \omega_1 - 0.088 \omega_2 - 0.8 \omega_3 & -0.0872 \omega_1 - 0.0157 \omega_3 & 0.037 \omega_3 - 0.016 \omega_2 - 0.8 \omega_1 & 0 & -0.001 & 0 \\ 0.88 \omega_1 + 0.02 \omega_2 - 0.4 \omega_3 & 0.02 \omega_1 + 0.8 \omega_2 - 0.47 \omega_3 & -0.47 \omega_2 - 0.4 \omega_1 & 0 & 0 & -0.001 \end{pmatrix} \quad (4.15)$$

Thus, the slew rate and gyroscope bias are always observable.

4.2 Quaternion

In the second subspace, the quaternion attitude is the state that needs to be estimated and controlled. Using the estimate of the slew rate as the effective control input to this subspace, the state vector and differential equations are as follows:

$$\mathbf{x} = [\hat{\mathbf{q}}] \quad (4.16)$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}_{\text{eff}} = 0_{4 \times 1} + \left[\frac{1}{2} \Xi(\hat{\mathbf{q}}) \right] \hat{\omega} \quad (4.17)$$

The outputs for the subspace containing the quaternion state are the measurements from the magnetometers, sun sensors, and star tracker on-board the spacecraft. Each of these sensors produces a measurement in the body frame of the spacecraft. These should match the model-generated inertial vectors rotated into the body frame of the spacecraft. If there are two measurements available from the sensor suite, the output expressed in terms of the states is:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \tilde{\mathbf{m}}_{1,\mathbf{B}} \\ \tilde{\mathbf{m}}_{2,\mathbf{B}} \end{bmatrix} = \begin{bmatrix} \text{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{1,\mathbf{I}} \\ \text{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{2,\mathbf{I}} \end{bmatrix} \quad (4.18)$$

where $\mathbf{m}_{i,\mathbf{I}}$ represents the true inertial vector that corresponds with the body-frame measurement $\tilde{\mathbf{m}}_{i,\mathbf{B}}$ of the i th sensor.

The controllability matrix is given by:

$$\mathbf{C} = [\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3] = \frac{1}{2} \begin{pmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{pmatrix} = \frac{1}{2} \Xi(\mathbf{q}) \quad (4.19)$$

which is full rank for all unity-norm quaternions. Thus, the quaternion is always controllable. The observability matrix is given by:

$$\mathbf{O} = \mathbf{d}\mathbf{h} = \begin{pmatrix} 4a_1q_1 + 2a_2q_2 + 2a_3q_3 & 2a_3q_1 - 2a_3q_4 & 2a_3q_1 + 2a_2q_4 & 4a_1q_4 + 2a_2q_3 - 2a_3q_2 \\ 2a_1q_2 + 2a_3q_4 & 2a_1q_1 + 4a_2q_2 + 2a_3q_3 & 2a_3q_2 - 2a_1q_4 & 2a_3q_1 - 2a_1q_3 + 4a_2q_4 \\ 2a_1q_3 - 2a_2q_4 & 2a_1q_4 + 2a_2q_3 & 2a_1q_1 + 2a_2q_2 + 4a_3q_3 & 2a_1q_2 - 2a_2q_1 + 4a_3q_4 \\ 4b_1q_1 + 2b_2q_2 + 2b_3q_3 & 2b_3q_1 - 2b_3q_4 & 2b_3q_1 + 2b_2q_4 & 4b_1q_4 + 2b_2q_3 - 2b_3q_2 \\ 2b_1q_2 + 2b_3q_4 & 2b_1q_1 + 4b_2q_2 + 2b_3q_3 & 2b_1q_2 - 2b_1q_4 & 2b_3q_1 - 2b_1q_3 + 4b_2q_4 \\ 2b_1q_3 - 2b_2q_4 & 2b_1q_4 + 2b_2q_3 & 2b_1q_1 + 2b_2q_2 + 4b_3q_3 & 2b_1q_2 - 2b_2q_1 + 4b_3q_4 \end{pmatrix} \quad (4.20)$$

To avoid an overabundance of subscripts, the inertial measurements vectors $\mathbf{m}_{1,\mathbf{I}}$ and $\mathbf{m}_{2,\mathbf{I}}$ are represented by $\mathbf{a} = [a_1 \ a_2 \ a_3]^T$ and $\mathbf{b} = [b_1 \ b_2 \ b_3]^T$ in the matrix above. This matrix is full rank if the measurement vectors are linearly independent, so the quaternion is observable. However, if there is only one linearly independent measurement vector available, the output is reduced to:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \tilde{\mathbf{m}} = \text{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{\mathbf{I}} \quad (4.21)$$

The observability matrix in this case is:

$$\mathbf{O} = \begin{bmatrix} \frac{dh}{dL_f^1(h)} \\ \end{bmatrix} = \begin{pmatrix} 4 a_1 q_1 + 2 a_2 q_2 + 2 a_3 q_3 & 2 a_3 q_1 - 2 a_3 q_4 & 2 a_3 q_1 + 2 a_2 q_4 & 4 a_1 q_4 + 2 a_2 q_3 - 2 a_3 q_2 \\ 2 a_1 q_2 + 2 a_3 q_4 & 2 a_1 q_1 + 4 a_2 q_2 + 2 a_3 q_3 & 2 a_3 q_2 - 2 a_1 q_4 & 2 a_3 q_1 - 2 a_1 q_3 + 4 a_2 q_4 \\ 2 a_1 q_3 - 2 a_2 q_4 & 2 a_1 q_4 + 2 a_2 q_3 & 2 a_1 q_1 + 2 a_2 q_2 + 4 a_3 q_3 & 2 a_1 q_2 - 2 a_2 q_1 + 4 a_3 q_4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.22)$$

This matrix is only of rank 3, and thus the quaternion attitude is unobservable if there is only one sensor measurement available or if all of the available measurement vectors align. The former case is handled by the Sun acquisition controller introduced in the next section. The latter case occurs if the satellite is not using a star tracker and the Sun vector is aligned with the local magnetic field. In this scenario, the only way of reacquiring attitude observability is to change the relative orientation of the Sun vector and the local magnetic field by changing the position of the spacecraft. The small satellite using this ADC system most likely does not have a propulsion system or would consider it much too costly to expend fuel on recovering attitude observability. Thus, the satellite will simply need to drift through this period of unobservability until its orbit sufficiently alters the relative orientation of the Sun and magnetic field vectors.

4.3 Sun Acquisition Controller

When the satellite is not using a star tracker and the Sun is out of the field-of-view of the sun sensors, only measurements from the magnetometer will be available. However, at least two linearly independent sensor measurements are required in order to estimate the attitude of the spacecraft according to the observability analysis above.

In order to avoid this scenario, a controller was developed to acquire the Sun vector by rotating the spacecraft until the Sun is within the FOV of a sun sensor. Since the slew rate is always observable, this is a slew rate controller that does not require an attitude estimate. The controller uses a simple proportional control law with nonlinearity cancellation for the known gyroscopic torques in the slew rate dynamics. The resulting control torque is saturated to meet the maximum torque constraint from the reaction wheels:

$$\mathbf{u}_{\text{wheel}} = \text{sat}(-c\mathbf{J}\hat{\boldsymbol{\omega}}_{\mathbf{E}} + \hat{\boldsymbol{\omega}}_{\mathbf{E}}^{\times}(\mathbf{J}\hat{\boldsymbol{\omega}}_{\mathbf{E}} + \mathbf{h}_{\text{wheel}}), \mathbf{u}_{\text{max}}) \quad (4.23)$$

where sat is the saturation function defined in Equations 2.14 and 2.15, \mathbf{u}_{max} is the maximum torque from the reaction wheels, c is the controller gain, and $\hat{\boldsymbol{\omega}}_{\mathbf{E}}$ is the slew rate error given by:

$$\hat{\boldsymbol{\omega}}_{\mathbf{E}} = \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_{\text{desired}} \quad (4.24)$$

where $\boldsymbol{\omega}_{\text{desired}}$ is the desired slew rate of the spacecraft. The scanning algorithm will initially null the rates of the spacecraft by setting $\boldsymbol{\omega}_{\text{desired}} = \mathbf{0}_{3 \times 1}$. The controller will then spin the spacecraft up to some constant speed ω_{scan} about the Z axis to begin scanning for the Sun by setting $\boldsymbol{\omega}_{\text{desired}} = [0 \ 0 \ \omega_{\text{scan}}]^T$. If the rotation about the Z

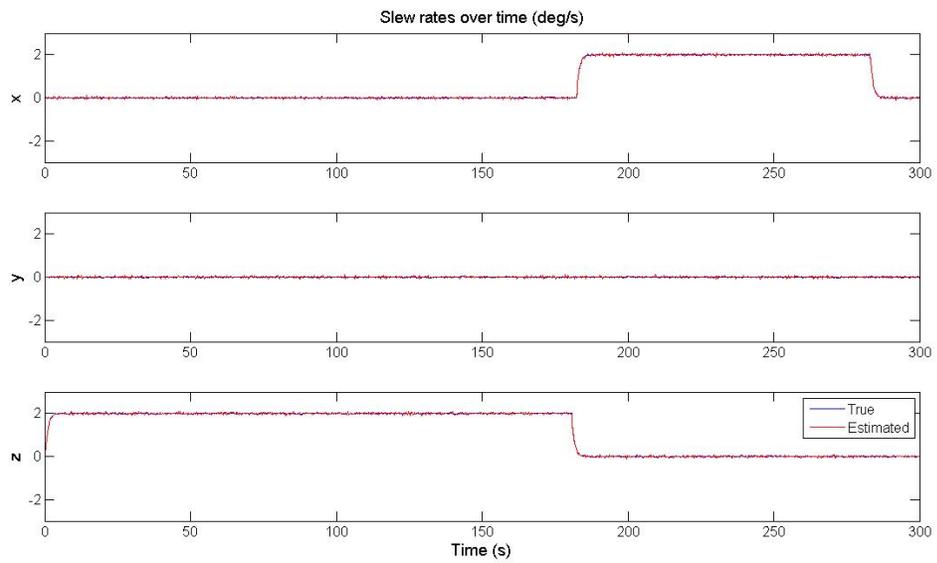
axis is completed without acquiring the Sun within the FOV of a sun sensor, the controller will then set $\omega_{\text{desired}} = [\omega_{\text{scan}} \ 0 \ 0]^T$ to begin rotating about the X axis at the same constant scanning rate. As soon as the Sun is acquired, the controller despins the satellite again to hold that orientation.

Since ARMADILLO and Bevo-2 have sun sensors on the X-Y plane of the spacecraft with a FOV of 140° , this set of rotations is guaranteed to put the Sun within view of a sun sensor if the satellite is not in eclipse. For any arbitrary satellite configuration that includes a sun sensor with a FOV $> 90^\circ$, selecting the two body axes that are orthogonal to the boresight vector of the sensor for this series of rotations will also guarantee Sun acquisition. Note that care should be taken to select a sufficiently low ω_{scan} to ensure that the reaction wheels can despin the satellite before the Sun passes back out of the FOV of the sun sensors. For the ARMADILLO and Bevo-2 satellites, ω_{scan} is the maximum slew rate of 2deg/s , which allows near-instantaneous despinning.

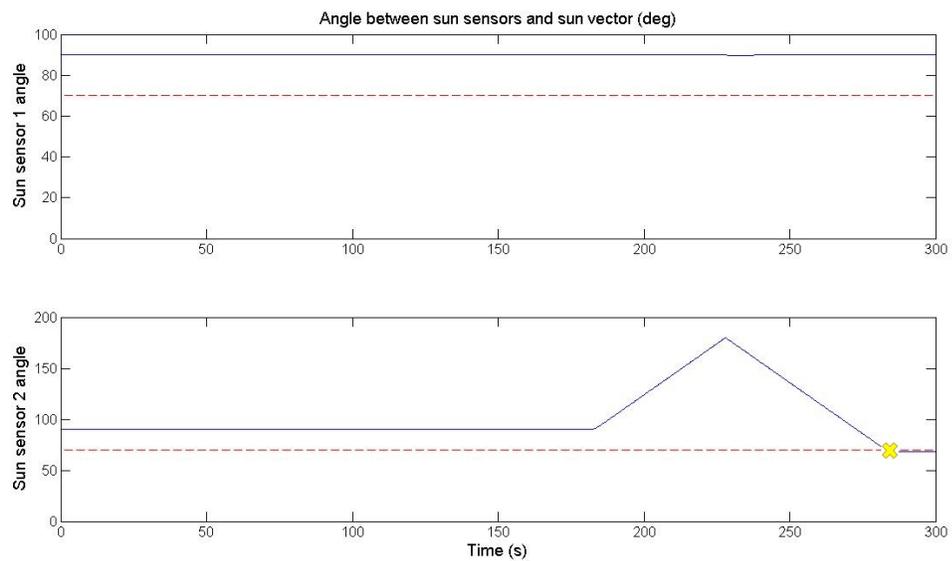
Using the environment described in Chapter 3, the Sun acquisition controller was simulated using the configuration of sun sensors on-board Bevo-2. The initial orientation of the satellite is the worst-case initial orientation for the scanning algorithm, in which the Sun vector is initially aligned with the Z axis of the spacecraft. Figure 4.1a shows the slew rates of the spacecraft as it rotates about the Z and X body axes sequentially while searching for the Sun. Figure 4.1b is a plot of the angle between each sun sensor’s boresight vector and the vector from the satellite to the Sun during this scanning procedure. Given the 140° field-of-view, the sun sensors are able to see the Sun when the angle between those vectors is less than 70° . This threshold is delineated in red on the angle plots. The “X” marks the point at which the Sun is acquired within the FOV of a sensor. As illustrated in this plot, the worst-case Sun acquisition time is less than five minutes.

Figure 4.1: Sun Acquisition Simulation Results

(a) Slew Rates



(b) Sun Sensor Angle to Sun Vector



Chapter 5

Navigation Filter

This chapter discusses the multiplicative extended Kalman filter (MEKF) that is used to provide estimates of the current state of the spacecraft. The filter is initialized by using Davenport's q-method to provide an initial estimate of the quaternion attitude. The navigation filter then updates the previously computed (a priori) estimate by consuming measurements from the suite of sensors on-board the spacecraft. When the system loses observability of the attitude state, the attitude is propagated by integrating the slew rate measurements from the gyroscopes.

5.1 A Priori Estimate

The states estimated by the navigation filter are the quaternion and gyroscope bias. The slew rate is not directly estimated, since it can be calculated using the estimate of the gyroscope bias with Equation 4.2. After receiving the first set of measurements from the sensors, an a priori estimate of the states must be calculated to initialize the filter. The a priori estimate of the gyroscope bias is $\mathbf{0}_{3 \times 1}$, but Davenport's q-method is used to obtain an initial estimate of the quaternion attitude. The optimal quaternion solution minimizes the performance index,

$$J(\mathbf{q}) = \lambda_0 - \mathbf{q}^T \mathbf{K} \mathbf{q} \quad (5.1)$$

where λ_0 is the sum of the weights assigned to each of the N_{meas} measurements, w_i .

$$\lambda_0 = \sum_{i=1}^{N_{meas}} w_i$$

\mathbf{K} is the Davenport matrix, defined as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \mu \mathbf{I}_{3 \times 3} & \mathbf{z} \\ \mathbf{z}^T & \mu \end{bmatrix} \quad (5.2)$$

$$\mathbf{z} = \sum_{i=1}^{N_{meas}} w_i \tilde{\mathbf{m}}_{\mathbf{B},i} \times \mathbf{m}_{\mathbf{I},i} \quad (5.3)$$

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T \quad (5.4)$$

$$\mu = \text{trace}(\mathbf{B}) \quad (5.5)$$

$$\mathbf{B} = \sum_{i=1}^{N_{meas}} w_i \tilde{\mathbf{m}}_{\mathbf{B},i} \mathbf{m}_{\mathbf{I},i}^T \quad (5.6)$$

$$\mathbf{W} = \begin{bmatrix} w_1 \mathbf{I}_{3 \times 3} & & & [\mathbf{0}] \\ & w_2 \mathbf{I}_{3 \times 3} & & \\ & & \ddots & \\ [\mathbf{0}] & & & w_{N_{meas}} \mathbf{I}_{3 \times 3} \end{bmatrix} = \mathbf{R}^{-1} \quad (5.7)$$

The matrix \mathbf{R} is the covariance for the measurement error, which is determined by the accuracy of the sensors. To give less weight to measurements with greater uncertainty, the measurement weights are the inverse of the covariance of the measurement error.

$$w_i = 1/\sigma_{\phi,i}^2 \quad (5.8)$$

where $\sigma_{\phi,i}^2$ is the error variance in the i th measurement. For example, the magnetometer has greater error uncertainty than the sun sensors on the TSL satellites, so the magnetometer measurements are weighted less heavily in this estimate. The minimum of the performance index in Equation 5.1 occurs when

$$\mathbf{K}\mathbf{q} = \lambda\mathbf{q} \quad (5.9)$$

which is an eigenvalue/eigenvector problem. The eigenvector \mathbf{q} associated with the largest eigenvalue λ is the optimal solution for the attitude quaternion and is thus the a priori estimate for the MEKF [21].

5.2 Measurement Update Step

A sequential estimator computes estimates of the state in real-time, updating the a priori estimate as new measurements from the attitude sensors are received. Usually, extended Kalman filters use an additive update of the state:

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- + \mathbf{K}[\tilde{\mathbf{y}} - \mathbf{h}(\hat{\mathbf{x}})] \quad (5.10)$$

where \mathbf{K} is the Kalman gain and $\mathbf{h}(\hat{\mathbf{x}})$ is the predicted value of the output based on the system model. The residuals, or the difference between the actual measured output and the expected value, are given by $\tilde{\mathbf{y}} - \mathbf{h}(\hat{\mathbf{x}})$. The superscripts $+$ and $-$ denote the updated and a priori estimates, respectively. However, this additive update does not work well for quaternion states due to the unity-norm constraint, i.e.,

$$\hat{\mathbf{q}}^+ = \hat{\mathbf{q}}^- + \mathbf{K}[\tilde{\mathbf{y}} - \mathbf{h}(\hat{\mathbf{q}})] \quad (5.11)$$

will give a quaternion $\hat{\mathbf{q}}^+$ that does not have a magnitude of 1, violating the constraint in Equation 3.12. Therefore, a multiplicative update step is used for the quaternion estimate:

$$\hat{\mathbf{q}}^+ = \begin{bmatrix} \frac{1}{2}\delta\hat{\alpha}^+ \\ 1 \end{bmatrix} \otimes \hat{\mathbf{q}}^- = \hat{\mathbf{q}}^- + \frac{1}{2}\boldsymbol{\Xi}(\hat{\mathbf{q}}^-)\delta\hat{\alpha}^+ \quad (5.12)$$

where \otimes is the quaternion multiplication operator defined in Equation 3.13. $\delta\alpha$ is a vector of small angle corrections about each body axis, namely:

$$\delta\alpha = \begin{bmatrix} \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} \quad (5.13)$$

As long as $\delta\hat{\alpha}^+$ is small, the quaternion is close to unity-norm. This multiplicative update results in slower error build-up in the norm of the quaternion, however, the constraint is still violated to a lesser degree. To resolve this issue, the quaternion is brute-force normalized every time a new estimate is calculated.

$$\mathbf{q} = \frac{\mathbf{q}}{\|\mathbf{q}\|} \quad (5.14)$$

The state correction vector, $\Delta\mathbf{x}$, is defined as:

$$\Delta\mathbf{x} = \mathbf{x}^* - \hat{\mathbf{x}} \quad (5.15)$$

where \mathbf{x}^* is the nominal state. With the multiplicative attitude update, the corrections to the quaternion and gyroscope bias are given by:

$$\Delta \mathbf{x} = \begin{bmatrix} \delta \alpha \\ \delta \beta \end{bmatrix} \quad (5.16)$$

Substituting these two equations into Equation 5.10 gives the following expression for the estimated state correction vector:

$$\hat{\Delta} \mathbf{x} = \begin{bmatrix} \hat{\delta} \alpha \\ \hat{\delta} \beta \end{bmatrix} = \mathbf{K}[\tilde{\mathbf{y}} - \mathbf{h}(\hat{\mathbf{q}})] \quad (5.17)$$

The predicted value of the output is given by rotating the known, true measurement vectors into the body frame according to Equation 4.18, which is restated here for convenience:

$$\mathbf{h}(\hat{\mathbf{q}}) = \begin{bmatrix} \text{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{1,\mathbf{I}} \\ \text{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{2,\mathbf{I}} \end{bmatrix}$$

The Kalman gain is computed as follows:

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T[\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}]^{-1} \quad (5.18)$$

where \mathbf{P} is the error covariance matrix which the sequential estimator seeks to minimize.

$$\mathbf{P} = \text{E}[(\Delta \mathbf{x} - \hat{\Delta} \mathbf{x})(\Delta \mathbf{x} - \hat{\Delta} \mathbf{x})^T] \quad (5.19)$$

This covariance matrix is initialized to the identity matrix and decreases as the estimator converges upon the true state, driving the error in the estimated state towards zero. \mathbf{H} is the sensitivity matrix for the measurements and satisfies the equation

$$\mathbf{H}(\hat{\mathbf{q}})\hat{\Delta\mathbf{x}} = \begin{bmatrix} \mathbf{m}_{\mathbf{B},1} - \hat{\mathbf{m}}_{\mathbf{B},1} \\ \mathbf{m}_{\mathbf{B},2} - \hat{\mathbf{m}}_{\mathbf{B},2} \\ \vdots \\ \mathbf{m}_{\mathbf{B},N_{\text{meas}}} - \hat{\mathbf{m}}_{\mathbf{B},N_{\text{meas}}} \end{bmatrix} \quad (5.20)$$

For this system, the \mathbf{H} matrix is calculated according to:

$$\mathbf{H}(\hat{\mathbf{q}}) = \begin{bmatrix} \text{DCM}(\hat{\mathbf{q}})\tilde{\mathbf{m}}_{\mathbf{I},1}^{\times} & \mathbf{0}_{3 \times 3} \\ \text{DCM}(\hat{\mathbf{q}})\tilde{\mathbf{m}}_{\mathbf{I},2}^{\times} & \mathbf{0}_{3 \times 3} \\ \vdots & \vdots \\ \text{DCM}(\hat{\mathbf{q}})\tilde{\mathbf{m}}_{\mathbf{I},N_{\text{meas}}}^{\times} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (5.21)$$

After calculating the state correction vector $\Delta\mathbf{x}$, the estimate of the quaternion is updated according to Equation 5.12 and the estimate of the bias is updated according to

$$\beta^+ = \beta^- + \delta\beta \quad (5.22)$$

The covariance matrix is updated according to

$$\mathbf{P}^+ = [\mathbf{I} - \mathbf{KH}(\hat{\mathbf{q}}^-)]\mathbf{P}^- \quad (5.23)$$

5.3 Flying the Gyros

While the magnetometer, sun sensor, and star tracker vectors are used to provide measurement updates to the estimated states, these sensors are typically sampled at a lower rate than the gyroscopes. In between vector measurement updates from the attitude sensors, the gyroscope measurements are used to propagate the attitude solution. This integration

step is known as “flying the gyros.” The state differential equations were previously defined in Equations 4.4 and 4.17 and are restated here:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2} \boldsymbol{\Xi}(\hat{\mathbf{q}}) \hat{\omega}$$

$$\dot{\hat{\beta}} = -\frac{1}{\tau} \hat{\beta}$$

where the estimated slew rate $\hat{\omega}$ is calculated by subtracting the estimated gyroscope bias from the incoming slew rate measurements. The propagated states become the a priori estimate for the next measurement update step. The covariance matrix also evolves according to:

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{G}^T\mathbf{Q}\mathbf{G} \quad (5.24)$$

where \mathbf{F} is the derivative of the differential equations with respect to the states, \mathbf{Q} is the covariance matrix of the process noise, and \mathbf{G} is a mapping matrix from the process noises to the states. For this system,

$$\mathbf{F} = \begin{bmatrix} -\hat{\omega}^\times & -\mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\frac{1}{\tau} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.25)$$

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.26)$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_f^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_g^2 \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.27)$$

where σ_f and σ_g are the standard deviations of the process noise in the slew rate and gyroscope biases, respectively. Since the sequential estimator is based on a linear, unbiased, minimum variance estimate (LUMVE), the estimator minimizes the error covariance matrix as more measurements are processed. However, the Kalman gain will also decrease as the covariance matrix decreases, and eventually the filter will ignore incoming data if the covariance matrix becomes very small. This is known as filter divergence. Adding in process noise inflates the covariance between measurements, thus preventing the filter from diverging. Selecting the σ 's for the process noise in matrix \mathbf{Q} is a trial-and-error process known as tuning the filter. Too little process noise will cause the estimator to track poorly when the states are changing quickly, and using too much will cause the estimator to track the noise in the data too closely, resulting in a noisy estimate.

For further details on the formulation of the multiplicative extended Kalman filter, please refer to [22].

5.3.1 Attitude Error Accumulation

When the attitude is unobservable, there is an insufficient number of measurements available to update the estimated state. The attitude estimate is thus propagated by flying the gyros without angle corrections from the sensor measurement vectors. Integrating the noisy slew rate measurements from the gyroscopes results in an angle random walk, causing the estimated attitude to diverge from the true attitude of the spacecraft. The amount of error that accumulates in the attitude estimate depends on the quality of the gyroscopes and the amount of time spent integrating the noise in their measurements.

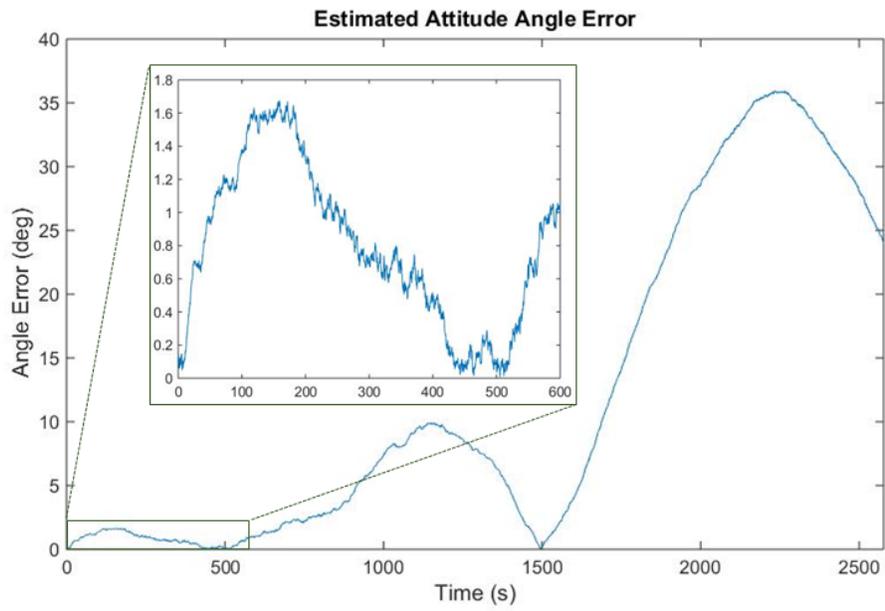
The simulation results presented in Figure 5.1 represent typical error accumulation in the attitude estimate during the worst-case attitude unobservability scenario for the TSL

satellites. This scenario assumes that:

1. The satellite requires a Sun vector for observability (i.e. it does not have a source of independent measurement vectors in addition to the magnetometer, such as a star tracker);
2. The Sun vector is in the orbital plane of the spacecraft, causing the satellite to spend 41% of its orbit in eclipse where the view of the Sun is obscured by Earth's shadow; and
3. After exiting eclipse, the satellite takes the worst-case acquisition time of five minutes to acquire a view of the Sun within a sun sensor.

As shown in Figure 5.1, the accumulated attitude error is too large to maintain pointing knowledge for this entire duration. Thus, the satellite requires a star tracker or better gyroscopes in order to perform pointing maneuvers while in eclipse. However, the zoomed-in portion on the first ten minutes of the plot demonstrates that the error in the attitude estimate is sufficiently small for navigation purposes if observability is only lost for a few minutes.

Figure 5.1: Attitude Estimation Error During Loss of Observability



Chapter 6

Unconstrained Pointing

This chapter presents the unconstrained pointing algorithms that were developed for the ADC system. The pointing system is composed of two parts: a pointing manager that computes the desired attitude of the spacecraft for some stated mission objective, and the pointing controllers that reorient the spacecraft into that desired orientation. This set of pointing algorithms is unconstrained in the sense that attitude constraints are not enforced on the satellite, though all other constraints are met.

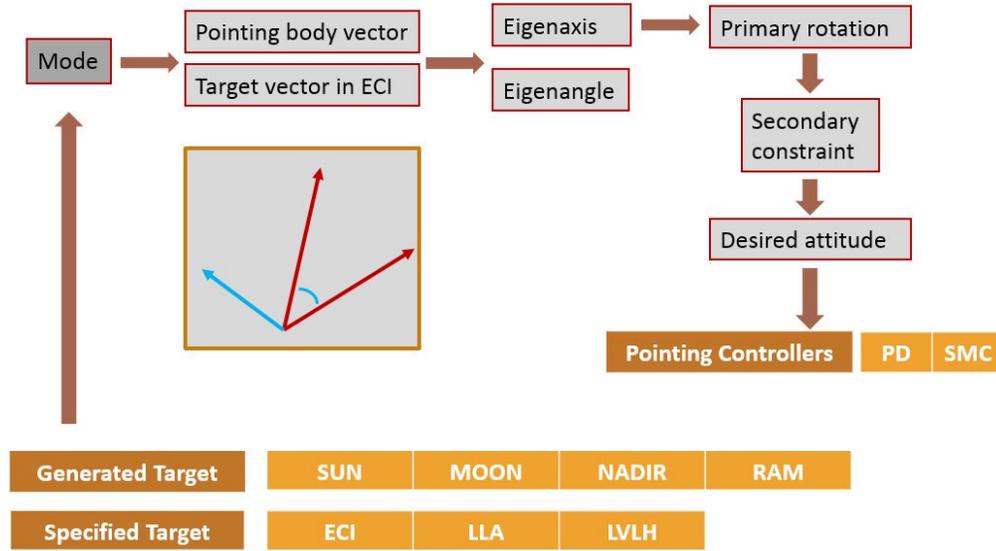
6.1 Pointing Manager

The pointing manager computes the quaternion attitude that will align a specified vector in the body frame of the spacecraft with an inertial target vector. The pointing vector in the body frame and the inertial vector to the target are determined by the mission mode of the spacecraft. The resulting quaternion is then sent to the pointing controllers which generate the actuator control signal necessary to perform the reorientation maneuver into this desired attitude. An schematic representation of the pointing manager is given in Figure 6.1.

6.1.1 Pointing Modes

This section discusses the commandable pointing modes of the spacecraft and the associated algorithms for computing the target vector \mathbf{t}_I in each mode. The target vector

Figure 6.1: Pointing Manager Overview



is a relative vector from the satellite to the target in the ECI frame. Depending on the pointing mode, the target vector is either automatically generated or specified. If a target vector \mathbf{t}_X is specified in frame X , the appropriate rotation matrix ${}^X\mathbf{T}_I$ is applied to rotate it from frame X into the ECI frame. Table 6.1 gives the target vector for each available pointing mode. Each of these target vectors must be normalized to form a unit vector.

Table 6.1: Target Vectors in Each Pointing Mode

(a) Generated Targets		(b) Specified Targets	
Pointing Mode	Target Vector	Pointing Mode	Target Vector
Sun	$\mathbf{r}_{\text{sun},I}$	ECI	\mathbf{t}_I
Moon	$\mathbf{r}_{\text{moon},I}$	LVLH	${}^I_{LVLH}\mathbf{T} \cdot \mathbf{t}_{LVLH}$
Nadir	$-\mathbf{r}$	LLA	${}^I_{LLA}\mathbf{T} \cdot \mathbf{t}_{LLA}$
Ram	\mathbf{v}		

The pointing vector \mathbf{p}_B is a unity-norm body-frame vector that is specified by the user. For example, to collect debris data on ARMADILLO, the spacecraft operator would

command the ADC system into the ram pointing mode and set \mathbf{p}_B to be the -Z body axis on which the PDD science instrument is located. As shown in Table 6.1, the corresponding target vector \mathbf{t}_I in this mode is the inertial velocity vector of the spacecraft, \mathbf{v} . The following sections provide further detail on the transformations that are applied to the target vector if it is specified in the LVLH or LLA frames.

6.1.1.1 LVLH Target

The local vertical/local horizontal (LVLH) reference frame is a spacecraft-fixed reference frame that is defined by the position of the satellite relative to Earth \mathbf{r} , and the unit normal vector to the orbital plane. The transformation matrix to rotate a target vector from the LVLH to ECI frame is given by:

$$\mathbf{I}^{\text{LVLH}}_{\text{T}} = [\mathbf{i}_{\text{LVLH}} \quad \mathbf{j}_{\text{LVLH}} \quad \mathbf{k}_{\text{LVLH}}] \quad (6.1)$$

where

$$\mathbf{i}_{\text{LVLH}} = \frac{\mathbf{r}}{\|\mathbf{r}\|} \quad (6.2)$$

$$\mathbf{k}_{\text{LVLH}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \quad (6.3)$$

and

$$\mathbf{j}_{\text{LVLH}} = \mathbf{k}_{\text{LVLH}} \times \mathbf{i}_{\text{LVLH}} \quad (6.4)$$

completes the right-handed frame [23]. Since the LVLH frame is centered at the spacecraft, the resulting target vector after applying this rotation matrix will be a relative vector from the spacecraft to the target as desired.

6.1.1.2 LLA Target

Latitude/longitude/altitude (LLA) is a set of geodetic coordinates rather than a reference frame, so it is an abuse of notation to define a $\mathbf{I}^{\text{LLA}}\mathbf{T}$. In reality, the LLA coordinates are transformed into an Earth-centered-Earth-fixed (ECEF) vector and then rotated from the ECEF frame to ECI. However, the rotation matrix is a convenient notation to represent this series of calculations that transforms an LLA vector into an ECI vector, even if the transformation is not really a simple rotation.

The geodetic latitude λ , longitude ϕ , and altitude alt of a point on Earth's surface can be transformed into a position vector in the ECEF frame according to:

$$\mathbf{r}_{\text{ECEF}} = \begin{bmatrix} (N + alt) \cos(\lambda) \cos(\phi) \\ (N + alt) \cos(\lambda) \sin(\phi) \\ ((1 - e^2)N + alt) \sin(\lambda) \end{bmatrix} \quad (6.5)$$

where

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2(\lambda)}} \quad (6.6)$$

is the ellipsoidal radius of curvature, a is the semi-major axis, and e is the eccentricity of Earth given by:

$$e = \sqrt{2f - f^2} \quad (6.7)$$

where f is the flattening factor of Earth. Using the constants defined by the World Geodetic System 1984 (WGS 84) ellipsoidal model of Earth, these parameters are $f = 1/298.257223563$ and $a = 6378.137km$ [24]. After converting the target's LLA coordinates into an ECEF position vector, the vector can be rotated into the ECI frame according to:

$$\mathbf{r}_{\text{ECI}} = \mathbf{I}_{\text{ECEF}}^{\text{I}} \mathbf{T} \cdot \mathbf{r}_{\text{ECEF}} \quad (6.8)$$

$$\mathbf{I}_{\text{I}}^{\text{ECEF}} \mathbf{T} = \mathbf{R}_3(-\theta_{GST}) \quad (6.9)$$

where θ_{GST} is the current Greenwich Sidereal Time [23]. To get the relative target vector from the satellite to the target, the position of the satellite in the ECI frame must be subtracted from the vector.

6.1.2 Desired Attitude

The desired attitude of the spacecraft is one which aligns the specified pointing vector in the body frame $\mathbf{p}_{\mathbf{B}}$ with the inertial target vector $\mathbf{t}_{\mathbf{I}}$. The eigenaxis of rotation $\mathbf{e}_{\mathbf{r}}$ is the unit normal vector to the plane containing $\mathbf{p}_{\mathbf{B}}$ and $\mathbf{t}_{\mathbf{I}}$:

$$\mathbf{e}_{\mathbf{r}} = \mathbf{p}_{\mathbf{B}} \times \mathbf{t}_{\mathbf{I}} \quad (6.10)$$

The eigenangle of rotation about this axis θ_r is the angle between the two vectors:

$$\theta_r = \arccos(\mathbf{p}_{\mathbf{B}} \cdot \mathbf{t}_{\mathbf{I}}) \quad (6.11)$$

The quaternion that corresponds to this rotation can be calculated using Equation 3.5:

$$\mathbf{q}_1 = \begin{bmatrix} \mathbf{e}_r \cdot \sin(\theta_r/2) \\ \cos(\theta_r/2) \end{bmatrix} \quad (6.12)$$

After constraining the pointing vector to be aligned with the target vector, there is another degree of freedom which can be used to satisfy a secondary pointing objective. If the user specifies a secondary pointing and target vector, $\mathbf{p}_{\mathbf{B},2}$ and $\mathbf{t}_{\mathbf{I},2}$, the angle between the two vectors can be minimized while satisfying the primary pointing constraint by performing another rotation about the constrained direction. The secondary rotation aligns the projected vectors $\mathbf{p}'_{\mathbf{B},2}$ and $\mathbf{t}'_{\mathbf{B}_1,2}$ into the plane normal to the constrained direction by using the procedure outlined above. The vector $\mathbf{t}'_{\mathbf{B}_1,2}$ is in the body frame of the satellite after the first rotation, denoted by \mathbf{B}_1 . Thus, the secondary rotation captures the \mathbf{B}_1 to body frame rotation that will align the projected pointing and target vectors. This is analogous to the ECI to \mathbf{B}_1 rotation represented by \mathbf{q}_1 . The projected vectors can be computed according to:

$$\mathbf{t}'_{\mathbf{B}_1,2} = \mathbf{p}_{\mathbf{B}} \times (\text{DCM}(\mathbf{q}_1)\mathbf{t}_{\mathbf{I},2} \times \mathbf{p}_{\mathbf{B}}) \quad (6.13)$$

$$\mathbf{p}'_{\mathbf{B},2} = \mathbf{p}_{\mathbf{B}} \times (\mathbf{p}_{\mathbf{B},2} \times \mathbf{p}_{\mathbf{B}}) \quad (6.14)$$

The secondary rotation that will align these projected vectors is given by:

$$\mathbf{q}_2 = \begin{bmatrix} \mathbf{e}_{r,2} \cdot \sin(\theta_{r,2}/2) \\ \cos(\theta_{r,2}/2) \end{bmatrix} \quad (6.15)$$

$$\mathbf{e}_{r,2} = \mathbf{p}'_{B,2} \times \mathbf{t}'_{B_1,2} \quad (6.16)$$

$$\theta_{r,2} = \arccos(\mathbf{p}'_{B,2} \cdot \mathbf{t}'_{B_1,2}) \quad (6.17)$$

These two successive rotations can be represented in a single rotation using quaternion notation:

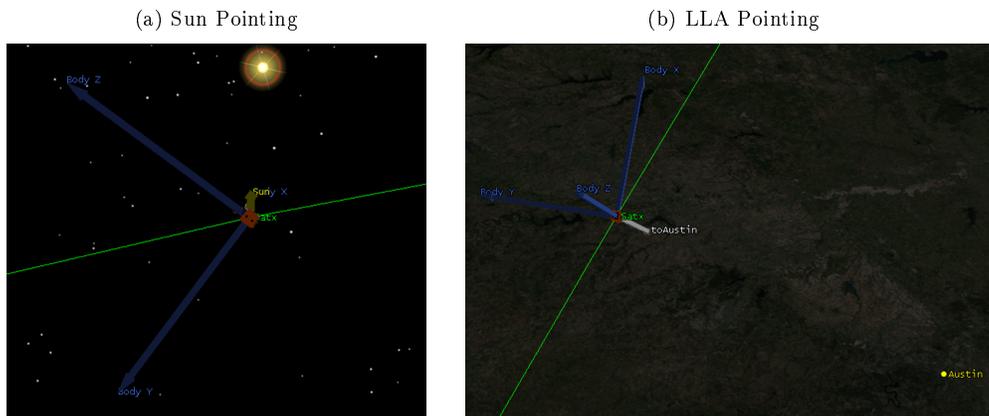
$$\mathbf{q}_{\text{cmd}} = \mathbf{q}_1 \otimes \mathbf{q}_2 \quad (6.18)$$

where \otimes is the quaternion multiplication operator defined in 3.13. Thus, the commanded attitude \mathbf{q}_{cmd} represents an eigenaxis rotation that aligns the primary pointing vector with the primary target and simultaneously minimizes the angle between the secondary pointing vector and target. For example, the satellite can point a camera at a target while minimizing the angle between the boresight vector of the sun sensor and the Sun for navigation purposes. On Bevo-2 and ARMADILLO, the pointing manager's default $\mathbf{p}_{B,2}$ is the sun sensor boresight vector that will have a smaller angle to the Sun since there are two sun sensors on-board these satellites.

6.1.3 STK Verification

The commanded quaternions from the pointing manager were verified using Analytical Graphics' Systems Tool Kit (STK) [25]. The MATLAB interface was used to automatically load the satellite and orbit parameters into STK, propagate the orbit, compute the desired quaternions for a given mode, and apply the computed quaternions to the STK

Figure 6.2: STK Pointing Verification



satellite object. The quaternion output from the pointing manager was successfully verified to produce the desired orientation for each mode. Examples of the simulated pointing verification in STK are provided in Figure 6.2.

6.2 Pointing Controllers

The pointing controllers accept a commanded quaternion from the pointing manager and compute the torques required to reorient the satellite into that attitude. There are two pointing controllers in this ADC system – a proportional derivative (PD) controller and a sliding mode controller (SMC). The controllers differ in their approaches to handling the nonlinear dynamics of the system. The PD controller is a linear controller that approximates the derivative of the attitude state as the slew rate and performs nonlinear cancellation. The SMC is a nonlinear controller that drives the states toward the desired states along a sliding surface.

Either of these controllers can be used to point the satellite while satisfying maximum slew rate and torque constraints. Both controllers compute a control torque input

$\mathbf{u}_{\text{wheel}_{3 \times 1}}$ which is generated by an assembly of reaction wheels. The reaction wheels must be oriented such that their combined torques span all three body axes, such as the assembly of three orthogonal reaction wheels on-board ARMADILLO and Bevo-2.

6.2.1 Proportional Derivative Controller

The proportional derivative controller employs the saturation control logic proposed by Wie [26]. The control law is as follows:

$$\mathbf{u}_{\text{cmd}} = \text{sat} \left(-\mathbf{K}_{\text{sat}} (\mathbf{I}_{3 \times 3} \hat{\mathbf{q}}_{\mathbf{E}}, \omega_{\text{max}}) - c \hat{\mathbf{J}} \hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\omega}}^\times (\hat{\mathbf{J}} \hat{\boldsymbol{\omega}} + \mathbf{h}_{\text{wheel}}), u_{\text{max}} \right) \quad (6.19)$$

where \mathbf{K} is the gain matrix, $\hat{\mathbf{q}}_{\mathbf{E}}$ is the error quaternion, and c is a control design parameter that tunes the performance of the controller. Increasing c will demand higher control torques and result in faster controller response. However, an overly high value for c will drive the system unstable, cause ringing, and saturate the control signal. For the TSL satellite parameters, a value of 1 for c was determined to provide desirable performance. Note that this value must be properly adjusted for other satellites with a different moment of inertia.

This is a proportional derivative controller in the sense that the first term is proportional to the attitude error, and the second term is proportional to the slew rate which is essentially the derivative of the attitude. The third term is for cancellation of the known nonlinearities in the attitude dynamics from the gyroscopic torques. This is the same nonlinear term cancellation that was performed for the Sun acquisition controller (refer to Section 4.3).

The error quaternion represents the rotation from the current estimated attitude to

the commanded attitude. The error quaternion can be computed via quaternion multiplication:

$$\hat{\mathbf{q}}_{\mathbf{E}} = \hat{\mathbf{q}} \otimes \mathbf{q}_{\text{cmd}}^{\mathbf{C}} \quad (6.20)$$

where the superscript \mathbf{C} denotes the conjugate quaternion:

$$\mathbf{q}^{\mathbf{C}} = \begin{bmatrix} -\mathbf{e}_{\mathbf{r}} \cdot \sin(\theta_r/2) \\ \cos(\theta_r/2) \end{bmatrix} \quad (6.21)$$

The gain matrix is given by:

$$\mathbf{K} = \text{diag} \left(c \cdot \text{sign}(\delta\alpha) \frac{\hat{\mathbf{q}}_{\mathbf{E}}}{\|\hat{\mathbf{q}}_{\mathbf{E}}\|} \right) \quad (6.22)$$

where the diag function diagonalizes an n -dimensional vector \mathbf{x} according to:

$$\text{diag}(\mathbf{x}) = \begin{bmatrix} x_1 & & & [\mathbf{0}] \\ & x_2 & & \\ & & \ddots & \\ [\mathbf{0}] & & & x_n \end{bmatrix} \quad (6.23)$$

The sign function operates on each component of an n -dimensional vector \mathbf{x} according to:

$$\text{sign}(\mathbf{x}) = \begin{bmatrix} \text{sign}(x_1) \\ \text{sign}(x_2) \\ \vdots \\ \text{sign}(x_n) \end{bmatrix} \quad (6.24)$$

where

$$\text{sign}(x_i) = \begin{cases} -1 & x_i < 0 \\ 0 & x_i = 0 \\ 1 & x_i > 0 \end{cases} \quad (6.25)$$

The vector $\delta\alpha$ is a set of angle errors about each body axis, $[\delta\theta_x \ \delta\theta_y \ \delta\theta_z]^T$. This is the same representation that was used in the MEKF to compute angle corrections to the attitude estimate during the measurement update step (refer to Section 5.2). The angle errors can be computed by transforming the error quaternion into its equivalent Euler angles using Equations 3.17, 3.18, and 3.19. Note that the two-quadrant Euler angle solution for $\delta\theta_x$ over $[-\pi, \pi]$ is sufficient since only the sign of these angle errors is needed by the control law.

The unit error quaternion in the gain matrix \mathbf{K} determines the eigenaxis of rotation, while the angle error sign check ensures that the satellite takes the shorter rotation about this eigenaxis to reach the commanded quaternion. The cascade-saturation functions in the control law prevent the satellite from exceeding the maximum torque and slew rate constraints. The outer saturation function limits the commanded torques to u_{max} , while the inner saturation function limits the maximum slew rate to ω_{max} . While saturation is a source of nonlinearity, the system behaves linearly within the saturation limits due to the nonlinear term cancellation. Further analysis proving that the linear controller provides asymptotic stability, bounds the maximum slew rate to ω_{max} , and monotonically drives the error quaternion to zero can be found in [26].

6.2.2 Sliding Mode Controller

As shown in the controllability analysis, a nonlinear controller can take advantage of the cascade structure of the system dynamics to drive the states as desired. A sliding

mode controller was developed that uses this approach to perform cascade control on the two subspaces of the system using two sliding mode control laws. The quaternion controller computes a slew rate trajectory that will drive the attitude to the commanded quaternion. The slew rate controller then computes the necessary actuator control torques to generate this slew rate trajectory.

6.2.2.1 Quaternion Control

The estimate of the slew rate is used as the effective control input to drive the quaternion to the desired orientation. For convenience, the system model is restated here:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Xi(\hat{\mathbf{q}})\hat{\omega}$$

$$\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{m}}_{1,\mathbf{B}} \\ \tilde{\mathbf{m}}_{2,\mathbf{B}} \end{bmatrix} = \begin{bmatrix} \mathbf{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{1,\mathbf{I}} \\ \mathbf{DCM}(\hat{\mathbf{q}}) \cdot \mathbf{m}_{2,\mathbf{I}} \end{bmatrix}$$

The quaternion is embedded in the output, so differentiating the output once will give a $\dot{\hat{\mathbf{q}}}$ term which contains the effective control, $\hat{\omega}$. Since the control term is recovered after one differentiation of the output, the relative order of the system is 1. Accordingly, a sliding surface of order 1 is constructed as a linear function of the error between the desired and estimated orientation. This error is represented by the vector of angle errors $\delta\alpha$ introduced previously:

$$\mathbf{s}_\omega = \delta\alpha = \begin{bmatrix} \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} \quad (6.26)$$

Once again, these angle errors can be computed by transforming the error quaternion into its equivalent Euler angles using Equations 3.17 through 3.19. In this case, the quadrant ambiguity in the Euler angles must be resolved. The corrected angle error $\delta\alpha_{i,c}$ about the i th body axis can be computed according to:

$$\delta\alpha_{i,c} = \begin{cases} \delta\alpha_i + 2\pi\text{sign}(\delta\alpha_i^-) & |\delta\alpha_i - \delta\alpha_i^-| > \pi \\ \delta\alpha_i & \text{otherwise} \end{cases} \quad (6.27)$$

where $\delta\alpha_i^-$ is a vector of previous angle errors. Since the derivative of the body axis angles is simply the slew rate, the effective control input can be computed by setting the derivative of the sliding surface equal to the desired sliding mode behavior:

$$\dot{\mathbf{s}}_\omega = \omega_{\mathbf{cmd}} = -\omega_{max} \cdot \tanh\left(\frac{\mathbf{s}_\omega}{s_{\omega 0}}\right) \quad (6.28)$$

where $s_{\omega 0}$ is the width of the boundary layer around the sliding surface. The \tanh function saturates at $+/- 1$, so the commanded slew rate never exceeds the ω_{max} constraint. A smooth \tanh function was chosen over the sign function in order to avoid aggressive “infinitely-fast” switching in the control signal. In the real system, this would cause the spacecraft to oscillate back and forth at a high frequency about the desired orientation instead of settling and maintaining it. The boundary layer acts like a low pass filter, effectively smoothing out the control signal and reducing chattering. Increasing $s_{\omega 0}$ amplifies this effect, but at the cost of degraded pointing accuracy. In the ADC system, it is desirable to have minimal chattering and a smooth control signal, but to also maintain high pointing accuracy. To achieve both of these goals, a variable boundary layer was imposed.

$$s_{\omega 0} = \begin{cases} 5 \text{ deg} & \|\delta\alpha\| > 5 \text{ deg} \\ 1 \text{ deg} & \|\delta\alpha\| \leq 5 \text{ deg} \end{cases} \quad (6.29)$$

The larger boundary layer when the spacecraft is far from the desired orientation generates a smoother control history with less chattering, and the smaller boundary layer once the spacecraft is close to commanded quaternion ensures a smaller steady state pointing error [27].

6.2.2.2 Slew Rate Control

A similar sliding mode control law is developed to compute the reaction wheel torques necessary to generate the slew rates commanded by the quaternion controller. For convenience, the system model for this subspace is also restated:

$$\dot{\omega} = \mathbf{J}^{-1}(-\omega^\times(\mathbf{J}\omega + \mathbf{h}_{\text{wheel}}) + \mathbf{u})$$

$$\mathbf{y} = \tilde{\omega} = \hat{\omega} + \hat{\beta}$$

Differentiating the output once will result in an $\dot{\omega}$ term containing the control input \mathbf{u} , so the relative order of this system is also 1. A sliding mode surface of order 1 in the error is defined as:

$$\mathbf{s}_{\mathbf{u}} = \hat{\omega}_{\mathbf{E}} = \omega_{\text{cmd}} - \hat{\omega} \tag{6.30}$$

where ω_{cmd} is the control signal computed by the quaternion sliding mode controller. Differentiating the above equation and setting it equal to a tanh sliding mode control function gives the following expression for $\dot{\mathbf{s}}$:

$$\dot{\mathbf{s}}_{\mathbf{u}} = \dot{\omega}_{\text{cmd}} - \dot{\hat{\omega}} = \dot{\omega}_{\text{cmd}} - \mathbf{J}^{-1}(-\dot{\hat{\omega}}^\times(\mathbf{J}\hat{\omega} + \mathbf{h}_{\text{wheel}}) + \mathbf{u}) = -\eta_u \cdot \tanh\left(\frac{\mathbf{s}_{\mathbf{u}}}{s_{u0}}\right) \quad (6.31)$$

where s_{u0} is the width of the boundary layer. Assuming that the commanded slew rate is roughly constant, solving for the control input \mathbf{u} gives:

$$\mathbf{u}_{\text{cmd}} = \mathbf{J}\eta_u \cdot \tanh\left(\frac{\mathbf{s}_{\mathbf{u}}}{s_{u0}}\right) + \hat{\omega}^\times(\mathbf{J}\hat{\omega} + \mathbf{h}_{\text{wheel}}) \quad (6.32)$$

The saturation limit η_u can be calculated as

$$\eta_u = \|\mathbf{J}\|^{-1} \cdot u_{max} \quad (6.33)$$

such that the saturation limit of the sliding mode term is the saturation limit of the reaction wheels. However, the additional nonlinear gyroscopic terms would cause the torque constraint to be violated. Since the nonlinear terms are much smaller in magnitude than the sliding mode term, they can be treated as additional disturbance torques to be rejected by the controller rather than computing a state-varying η_u that would satisfy the maximum torque constraint. The resulting commanded torque is:

$$\mathbf{u} = u_{max} \cdot \tanh\left(\frac{\mathbf{s}_{\mathbf{u}}}{s_{u0}}\right) \quad (6.34)$$

which does satisfy the torque constraints. The boundary layer width s_{u0} is set to 2 deg/s, which results in fairly smooth control torques and very accurate slew rate control.

6.2.3 Performance Analysis

A simulation of a 170 degree eigenangle slew maneuver was performed using both pointing controllers. The simulation includes environmental disturbance torques, actuator dynamics, and uncertainty in the moment of inertia of the spacecraft. The simulation also includes MEKF state estimation using simulated sensor measurements with noise and biases. For details on the ADC component models and simulation environment, please refer to Chapters 2 and 3.

The results of the simulation are presented in Table 6.2. The maneuver time is defined as the time taken to decrease the magnitude of the pointing error and slew rate below 0.5 deg and 0.5 deg/s, respectively. The pointing error is the mean angle to the desired attitude after the controller has reached steady state. The pointing error standard deviation is the standard deviation in the angle error about that mean value. As shown in Table 6.2, both controllers achieve very small pointing errors in less than two minutes. The sliding mode controller has better performance since it completes the slew maneuver more quickly and accurately.

As shown in the plots in Figure 6.3, both controllers satisfy the maximum torque constraint of $1mNm$ in each axis. While they also satisfy the maximum slew rate constraint of $2deg/s$, the type of slew rate constraint that is enforced differs between the two controllers. The PD controller enforces this constraint on the total eigenaxis slew rate of the spacecraft, whereas the SMC enforces it on the rotation rate about each individual axis. Thus, SMC allows the spacecraft to rotate faster, explaining why there is a fairly large discrepancy in maneuver time between the two controllers. Care should be taken when selecting a pointing controller to note whether the maximum slew rate limit is specified for an eigenaxis or

body axis rotation rate. For the TSL satellites the distinction is not significant, and either controller may be used. Also note that the price paid by the SMC for greater pointing precision is greater actuator effort, as can be seen in the torque plot comparison.

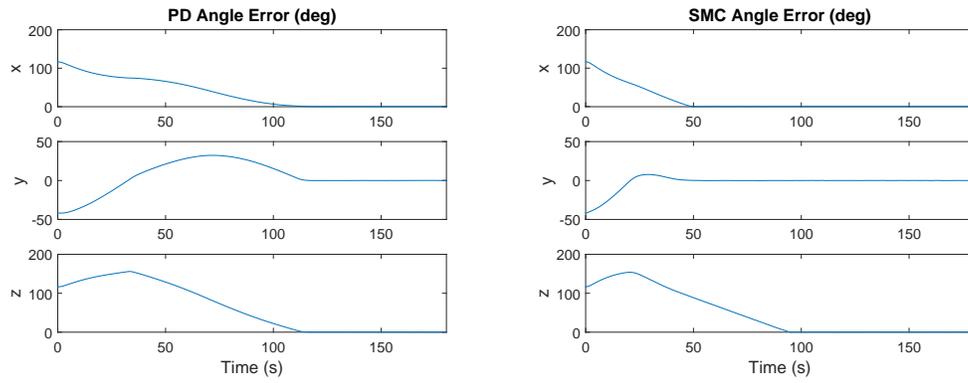
Although the attitude was continuously observable during this simulated maneuver, the unconstrained nature of the pointing system means that no keep-in constraints were enforced to keep the Sun in the FOV of a sun sensor, and no keep-out constraints were enforced to keep bright objects out of the FOV of the star tracker. The lack of attitude constraints means that the sun sensors or star tracker could fail to provide measurement vectors to the navigation filter, and observability could be lost.

This navigation problem can be resolved by using the secondary pointing constraint in the pointing manager to minimize the angle between a navigation sensor and its target (e.g. sun sensor boresight and Sun vector). So long as the primary pointing constraint does not prevent the satellite from also satisfying the navigation requirements, the satellite will regain observability at the end of the maneuver. Although the attitudes along the reorientation trajectory are not constrained, the satellite can navigate through this period of unobservability since the pointing controller maneuver times are less than two minutes (refer to attitude error growth in Section 5.3.1).

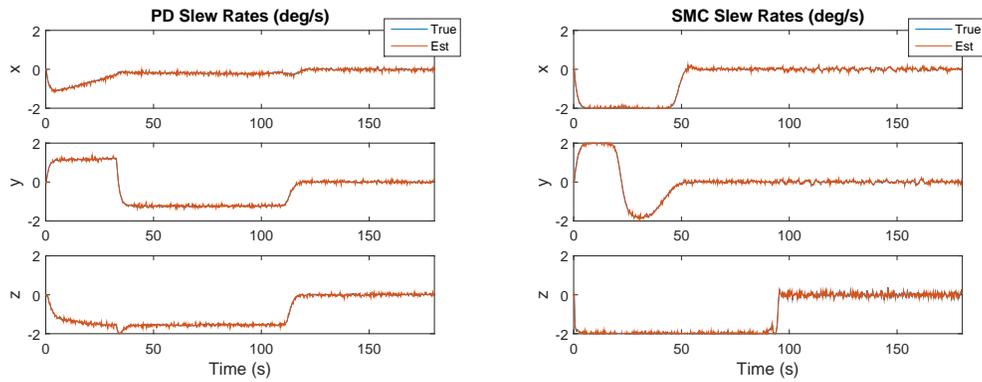
Care must be taken by the satellite operator to avoid commanding a primary pointing direction that prevents the satellite from collecting navigation measurement vectors. For example, on Bevo-2 and ARMADILLO, if the Z body axis is pointed at the Sun vector, the minimum angle to the Sun from the sun sensors on the X-Y plane is 90 degrees, which exceeds the 70 degree half-angle of the sensors. This pointing command would thus cause long-term attitude unobservability, and the satellite will eventually be unable to maintain pointing as the error in the estimated quaternion grows.

Figure 6.3: Pointing Controller Simulation Results

(a) Pointing Angle Error over Time



(b) Slew Rates over Time



(c) Reaction Wheel Torques over Time

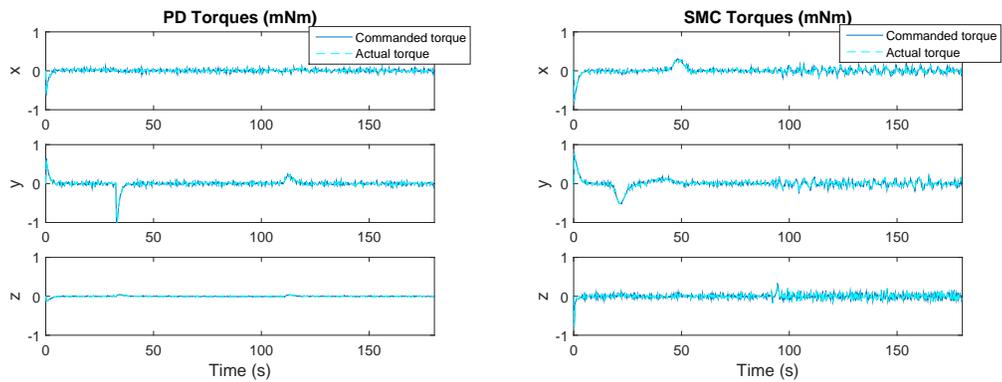


Table 6.2: Controller Performance

Controller	Maneuver Time (s)	Pointing Error (deg)	Standard Deviation (deg)
PD	116.75	0.277795	0.06332
SMC	95	0.139416	0.06255

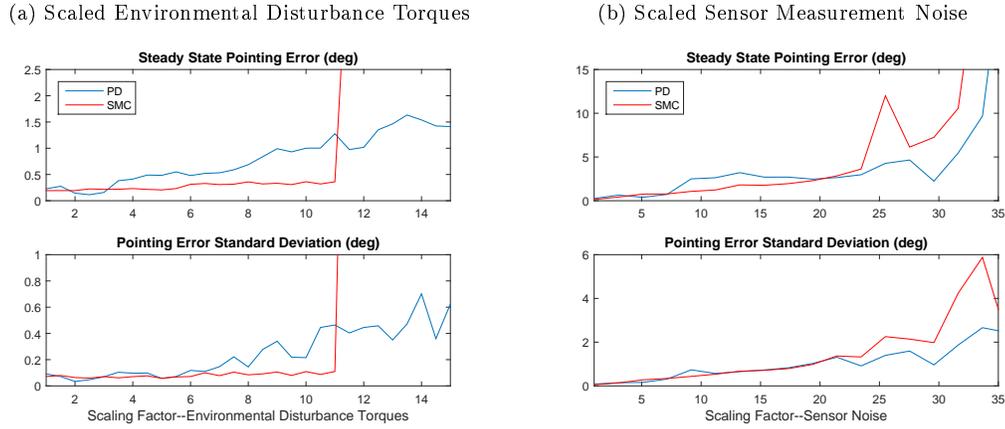
6.2.4 Robustness Analysis

Several studies were performed to analyze the robustness of the controllers to disturbances, sensor noise, and satellite parameter uncertainty. The disturbance robustness study compares the performance of the controllers in a series of simulations with a range of scaling factors from [1, 15] on the nominal environmental disturbance torques. The sensor noise robustness study compares their performance when the MEKF is providing state estimates using sensors with a scaling on the standard deviation in the measurement noise. The scaling factors on the sensor noise ranged from [1, 35], and the biases in the measurements were not altered. For details on the nominal sensor noise and disturbance torque values, refer to Sections 2.1 and 3.2.

As shown in the plots in Figure 6.4a, the sliding mode controller is generally much more robust to disturbance torques. This is as expected, since the SMC is designed to handle nonlinearities, whereas the PD controller relies on approximating the system as linear by performing nonlinear cancellation. As the simulated disturbance torques are unknown and cannot be cancelled, increasing the disturbance torques quickly degraded the linear controller's performance. This robustness to unmodeled system dynamics is one of the primary advantages of nonlinear control.

The large spike in error seen in the SMC plot is due to the reaction wheels saturating at their maximum momentum of $10mNm \cdot s$. The sliding mode controller causes wheel saturation before the proportional derivative controller does, since its control law demands

Figure 6.4: Controller Robustness to Disturbance and Sensor Noise

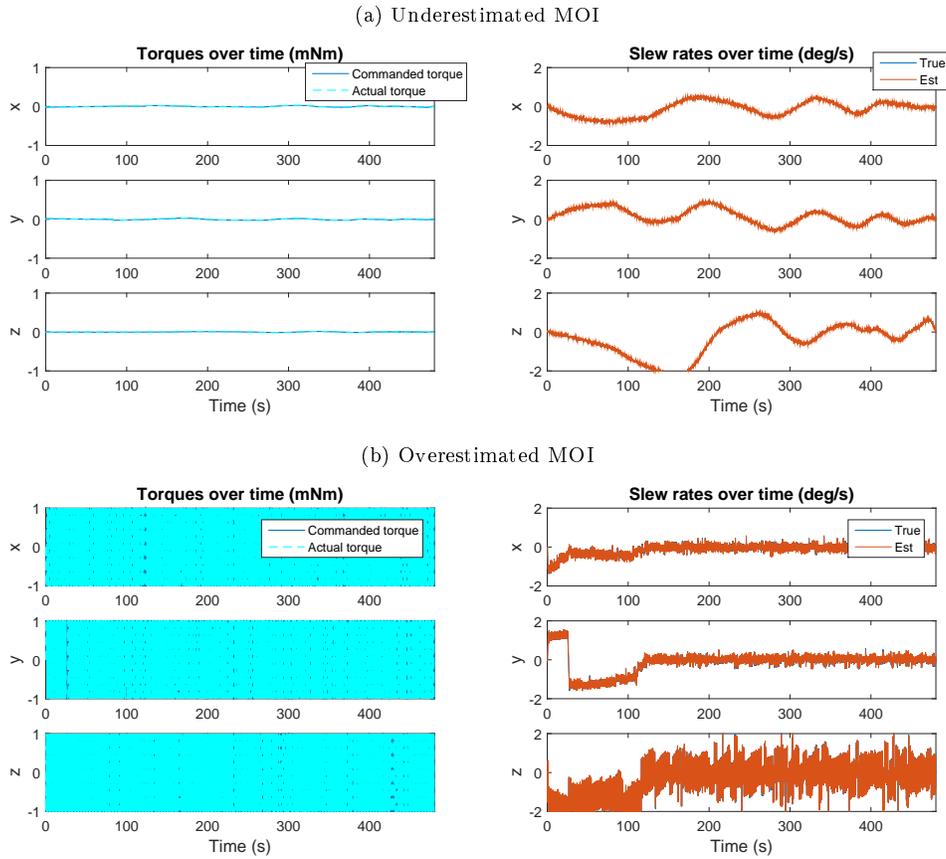


greater control torques from the wheels. Thus, if conserving actuator effort and preventing wheel saturation is a primary concern, the PD controller should be selected or the SMC boundary layer should be adjusted. If the wheels are unlikely to saturate, then the SMC should be selected for greater pointing accuracy.

As shown in Figure 6.4b, the controllers generally have comparable pointing accuracies when the sensor measurement noise is low, while the PD controller is more robust to very noisy measurements. However, the pointing errors of > 5 degrees at those levels of noise are too great for most satellite pointing requirements regardless.

Simulations of the controllers' performance when the moment of inertia (MOI) is under- and over-estimated by a factor of 50 demonstrated that the SMC is much more robust to MOI uncertainty. This is as expected, since the estimated moment of inertia tensor is integrated into the PD control law, whereas the SMC has no dependence on this estimate. The PD controller response plotted in Figure 6.5 shows the poor controller performance in

Figure 6.5: PD Controller Performance with MOI Uncertainty



both of these cases. The SMC response is not plotted, as the estimated moment of inertia has no effect on this control law. Thus, the SMC should be selected if the mass properties of the satellite are not well known. This is often the case for small satellites, which may not have access to the necessary equipment for measuring their mass properties. Without a measured MOI, estimates of the inertia tensor are often poor. For example, on RACE, the estimated moments of inertia about the two shorter axes were about 36% lower than the measured values, and 250% smaller for the moment of inertia about the long axis [28].

Chapter 7

Constrained Attitude Guidance

This chapter presents the guidance system that was developed as a more sophisticated alternative to the system outlined in the previous chapter. As noted above, the unconstrained pointing algorithms have the following limitations:

1. Only one additional attitude constraint can be satisfied via the secondary constraint functionality in the pointing manager;
2. The secondary constraint merely minimizes the angle between a second pair of pointing/target vectors, with no consideration for the field of view of the sensor;
3. Attitude is not constrained along the attitude trajectory during the maneuver—the secondary constraint is only for the final orientation, and the pointing controllers will take any path necessary to get there; and
4. Cannot satisfy keep-out constraints.

These restrictions motivated the development of a more general purpose constrained attitude guidance system without these limitations. Using a convex optimization approach, a CAG system was developed that autonomously reorients the satellite along an attitude trajectory that satisfies an arbitrary number of keep-in and keep-out constraints at all times.

7.1 Convex Optimization Approach

Nonlinear optimization can be applied to solve problems that have nonlinear constraints, as is the case for this problem. However, this approach does not guarantee that it will find a feasible solution even if one exists. The computation time can also be large which prevents it from being flown on a real-time navigation system. On the other hand, convex optimization offers a solution to the CAG problem that guarantees a solution within polynomial computation time if a feasible solution exists. Numerical convex optimization solvers using interior-point methods (IPMs) have proven to be very reliable and can solve problems with hundreds of variables and constraints in tens of seconds [29]. For the CAG problem, there are only a handful of variables and constraints; thus a convex optimization formulation would allow for control updates in real-time, as is necessary for an attitude control system. The difficulty in this approach is in formulating the CAG problem as a well-posed convex optimization problem.

A convex optimization problem has a convex objective function and convex constraints. In the CAG problem, the keep-in and keep-out pointing constraints define a set of permissible orientations which is non-convex. This source of non-convexity can be handled by applying the convex parametrization proposed by Yoonsoo Kim in [30, 31] to transform the attitude constraints into convex quadratic inequalities. For sets of redundant sensors where satisfying a keep-in pointing constraint for one sensor out of the set is sufficient, the constraint becomes a logical combination of the individual keep-in constraints. This can be satisfied by introducing binary variables to convert the set of quadratic inequality constraints into mixed integer convex constraints [32].

A second source of non-convexity is the nonlinear dynamics of the system. The

discretization approach taken by Kim in [30, 31] is adopted to transform the nonlinear kinematic constraints into pointwise-in-time linear constraints. The simplicity of this convexification of the problem comes at the cost of global optimality and feasibility, since this approach requires solving a convex optimization problem at every time step. Even if an admissible path exists over the entire set of permissible orientations, with discretized dynamics it is possible that a feasible solution does not exist at every step. The pointwise-in-time kinematic constraints can prevent the system from finding a path forward that does not violate the pointing constraints. This is addressed by relaxing the pointing constraints so that an admissible path is guaranteed, despite the discretization. Although this relaxation allows pointing constraint violations along the resulting path, this concern can be mitigated by enforcing more conservative pointing constraints.

Discretizing the dynamics also results in a suboptimal trajectory, since the optimization is performed at each step instead of over the entire path. However, a globally optimal trajectory would require recomputing the trajectory at certain time intervals as disturbances and uncertainties that are unmodeled in the system dynamics push the true states further from the predicted states. Discretization offers robustness to these unmodeled dynamics, since the control input is continually being computed based on the current state. The discretized convex optimization algorithm presented here naturally integrates into an ADC system by generating a steering law that acts just like any robust feedback controller while satisfying the pointing constraints.

7.2 Problem Statement

The CAG problem is to compute a control signal, $\mathbf{u}(\mathbf{t})$, that autonomously reorients the spacecraft such that a specified vector in the body frame of the spacecraft is aligned

with an inertial target vector. If \mathbf{p}_B is the unit body vector to point in the direction of \mathbf{t}_I , the unit inertial vector to the target, then the angle between these two vectors must be driven to zero:

$$\arccos(\mathbf{p}_B \cdot \mathbf{t}_B) \rightarrow 0 \quad (7.1)$$

where \mathbf{t}_B is the target vector rotated into the body frame according to Equations 3.14 and 3.15.

$$\mathbf{t}_B = \text{DCM}(\mathbf{q})\mathbf{t}_I \quad (7.2)$$

If the instrument needs to be at rest for mission operations, such as imaging a star, then the slew rate also needs to be driven to zero.

$$\boldsymbol{\omega} \rightarrow \mathbf{0}_{3 \times 1} \quad (7.3)$$

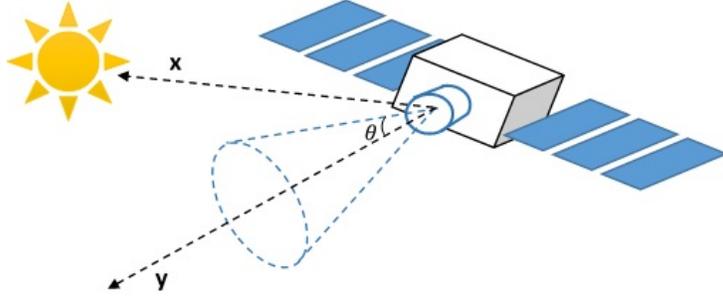
The bounds on control input and slew rate are enforced as follows:

$$\|\mathbf{u}\|_\infty \leq u_{max} \quad (7.4)$$

$$\|\boldsymbol{\omega}\|_\infty \leq \omega_{max} \quad (7.5)$$

where the infinity-norm bounds the control input and slew rates in each axis. If ω_{max} is a limit on the maximum eigenaxis rotation rate, the 2-norm should be used in Equation 7.5 instead. Both norms are convex functions, so these bounds are convex constraints.

Figure 7.1: Sun Keep-Out Attitude Constraint



To satisfy the keep-out pointing constraints, the angle between the unit boresight vector \mathbf{y} of a sensitive instrument and the unit vector \mathbf{x} to the object to which it is sensitive must exceed the half-angle θ of its exclusion cone, as illustrated in Figure 7.1. This constraint is satisfied for each sensitive instrument by the following condition:

$$\mathbf{x}_{I,i}^T \mathbf{y}_{I,i} \leq \cos \theta_i, \quad i \in [1, n_{out}] \quad (7.6)$$

where n_{out} is the number of sensitive instruments imposing keep-out constraints on the spacecraft. Both vectors are in the inertial frame as indicated by the subscript I . The keep-in constraint is analogous to the keep-out constraint and can be expressed as follows:

$$\mathbf{x}_{I,i}^T \mathbf{y}_{I,i} \geq \cos \theta_i, \quad i \in [1, n_{in}] \quad (7.7)$$

where n_{in} is the number of instruments imposing keep-in constraints, \mathbf{y} is the unit boresight vector of an instrument imposing a keep-in constraint, and \mathbf{x} represents the unit vector to the object which must be kept in its view. The angle between these two vectors must be within the half-angle θ of the inclusion cone of the instrument. If there is a set of redundant sensors for which satisfying one keep-in constraint out of the set is sufficient, the

constraint becomes a logical combination of individual keep-in constraints:

$$\mathbf{x}_{1,I}^T \mathbf{y}_{1,I} \geq \cos \theta_1$$

$$\text{or } \mathbf{x}_{2,I}^T \mathbf{y}_{2,I} \geq \cos \theta_2$$

⋮

$$\text{or } \mathbf{x}_{I,n_r}^T \mathbf{y}_{I,n_r} \geq \cos \theta_{n_r} \quad (7.8)$$

where n_r is the number of redundant sensors. Now the constrained attitude control problem can be summarized as follows:

Compute the control signal $u(t)$ that drives the angle between the specified pointing vector and target vector to zero by minimizing

$$\arccos(\mathbf{p}_B \cdot \mathbf{DCM}(\mathbf{q})\mathbf{t}_I)$$

subject to final conditions

$$\boldsymbol{\omega}(t_f) = \mathbf{0}_{3 \times 1}$$

kinematic constraints

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Xi}(\mathbf{q})\boldsymbol{\omega}, \quad \boldsymbol{\omega} = \mathbf{J}^{-1}(-\boldsymbol{\omega}^\times(\mathbf{J}\boldsymbol{\omega} + \mathbf{h}_{\text{wheel}}) + \mathbf{u})$$

control and slew rate bounds

$$\|\mathbf{u}\|_\infty \leq u_{max}, \quad \|\boldsymbol{\omega}\|_\infty \leq \omega_{max}$$

and pointing constraints

$$\mathbf{x}_{I,i}^T \mathbf{y}_{I,i} \leq \cos \theta_i, \quad i \in [1, n_{out}]$$

$$\mathbf{x}_{I,i}^T \mathbf{y}_{I,i} \geq \cos \theta_i, \quad i \in [1, n_{in}]$$

$$\mathbf{x}_{I,1}^T \mathbf{y}_{I,1} \geq \cos \theta_1 \text{ or } \mathbf{x}_{I,2}^T \mathbf{y}_{I,2} \geq \cos \theta_2 \text{ or } \dots \text{ or } \mathbf{x}_{I,n_r}^T \mathbf{y}_{I,n_r} \geq \cos \theta_{n_r}$$

7.3 Convex Parametrization of Pointing Constraints

The pointing constraints as currently stated are non-convex, but they can be converted into convex quadratic inequalities using the method outlined in [30, 31]. The keep-out pointing constraint is given by Equation 7.6, where the unit vector \mathbf{y}_I is the body vector \mathbf{y}_B rotated into the inertial frame:

$$\mathbf{y}_I = \mathbf{y}_B - 2(\mathbf{q}_v^T \mathbf{q}_v) \mathbf{y}_B + 2(\mathbf{q}_v^T \mathbf{y}_B) \mathbf{q}_v + 2q_4 (\mathbf{y}_B \times \mathbf{q}_v) \quad (7.9)$$

Substituting this expression and performing some algebraic manipulation allows the keep-out constraint for the i th instrument to be expressed as

$$\mathbf{q}^T \mathbf{P}_i \mathbf{q} \leq 0 \quad (7.10)$$

where

$$\mathbf{P}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{b}_i \\ \mathbf{b}_i^T & d_i \end{bmatrix} \quad (7.11)$$

$$\mathbf{A}_i = \mathbf{y}_{B,i} \mathbf{x}_{I,i}^T + \mathbf{x}_{I,i} \mathbf{y}_{B,i}^T - (\mathbf{x}_{I,i}^T \mathbf{y}_{B,i} + \cos \theta) \mathbf{I}_{3 \times 3} \quad (7.12)$$

$$\mathbf{b}_i = \mathbf{x}_{I,i} \times \mathbf{y}_{B,i} \quad (7.13)$$

$$d_i = \mathbf{x}_{I,i}^T \mathbf{y}_{B,i} - \cos \theta \quad (7.14)$$

This quadratic inequality is non-convex, since the matrix \mathbf{P}_i is sign-indefinite. However, Equation 7.10 is equivalent to

$$\mathbf{q}^T [\mathbf{P}_i + \lambda_i \mathbf{I}_{4 \times 4}] \mathbf{q} = \mathbf{q}^T \mathbf{P}_i \mathbf{q} + \lambda_i \|\mathbf{q}\|^2 \leq \lambda_i \quad (7.15)$$

since \mathbf{q} is unity norm. Then, the matrix

$$\hat{\mathbf{P}}_i = \mathbf{P}_i + \lambda_i \mathbf{I}_{4 \times 4} \quad (7.16)$$

is positive definite if λ_i is slightly larger than the most negative eigenvalue of \mathbf{P}_i . This spectral shift makes all eigenvalues of $\hat{\mathbf{P}}_i$ strictly positive, so $\hat{\mathbf{P}}_i$ is a symmetric positive definite matrix. The following expression is thus a convex quadratic inequality constraint.

$$\mathbf{q}^T \hat{\mathbf{P}}_i \mathbf{q} \leq \lambda_i \quad (7.17)$$

An equivalent process can be applied to the keep-in constraint from Equation 7.7. Due to the sign change, the quadratic constraint becomes

$$\mathbf{q}^T \mathbf{P}_i \mathbf{q} = \begin{bmatrix} \mathbf{A}_i & \mathbf{b}_i \\ \mathbf{b}_i^T & d_i \end{bmatrix} \geq 0 \quad (7.18)$$

or

$$\mathbf{q}^T (-\mathbf{P}_i) \mathbf{q} \leq 0 \quad (7.19)$$

which is equivalent to

$$\mathbf{q}^T [-\mathbf{P}_i + \lambda_i \mathbf{I}_{4 \times 4}] \mathbf{q} = \mathbf{q}^T (-\mathbf{P}_i) \mathbf{q} + \lambda_i \|\mathbf{q}\|^2 \leq \lambda_i \quad (7.20)$$

Then the matrix

$$\tilde{\mathbf{P}} = -\mathbf{P}_i + \lambda_i \mathbf{I}_{4 \times 4} \quad (7.21)$$

is symmetric positive definite if λ_i is slightly larger than the most positive eigenvalue of \mathbf{P}_i , and the keep-in constraint has been converted into another convex quadratic inequality constraint.

$$\mathbf{q}^T \tilde{\mathbf{P}}_i \mathbf{q} \leq \lambda_i \quad (7.22)$$

If there is a set of redundant sensors, the keep-in constraint should be satisfied for at least one of the sensors in that set. Enforcing the keep-in constraint for every sensor in the set results in an over-constrained problem that may not be feasible. The logical combination of individual constraints as stated in Equation 7.8 can be satisfied by introducing a binary variable $\mu_i = \{0, 1\}$ for each redundant sensor. The resulting mixed integer constraint can be constructed as follows:

$$\mathbf{q}^T \tilde{\mathbf{P}}_i \mathbf{q} \leq \lambda_i + M \mu_i \quad (7.23)$$

where M is an arbitrary large positive number. Thus, $\mu_i = 0$ means that the keep-in constraint is satisfied for the i^{th} instrument. To ensure that at least one of these binary values is 0, the following additional constraint is enforced on the binary variables:

$$\sum_{i=1}^{n_r} \mu_i \leq n_r - 1 \quad (7.24)$$

The complexity of a mixed integer convex programming (MICP) problem increases exponentially with the number of binary variables; in this case, that is the number of redundant pointing constraints. For any spacecraft system the number of redundancies in a set of sensors is guaranteed to be small, since having many redundant sensors unnecessarily increases the mass and cost of the system. The MICP problem can thus be expected to be limited to at most 5 binary variables, which keeps the computation time low.

7.4 Discretization of Nonlinear Dynamics

The kinematics of the system are non-convex, but they can be converted into pointwise-in-time linear constraints by discretization [30, 31]. Linearizing the quaternion dynamics in Equation 3.6 gives the following quaternion update equation at each time step, k :

$$\mathbf{q}(k+1) = \Delta t \frac{1}{2} \Xi(\mathbf{q}(k)) \boldsymbol{\omega}(k) + \mathbf{I}_{4 \times 4} \mathbf{q}(k) \quad (7.25)$$

where Δt is the time interval between each step. Using forward differences to compute the numerical derivative, the angular acceleration is

$$\dot{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}(k+1) - \boldsymbol{\omega}(k)}{\Delta t} \quad (7.26)$$

Substituting this expression for $\dot{\boldsymbol{\omega}}$ into Equation 3.7, gives the following constraint on the control torques:

$$\mathbf{u}(k) = \mathbf{J} \frac{\omega(k+1) - \omega(k)}{\Delta t} + \omega(k)^\times (\mathbf{J}\omega(k) + \mathbf{h}_{\text{wheel}}(k)) \quad (7.27)$$

Selecting the solution variable

$$\mathbf{s} = \begin{bmatrix} \mathbf{u}(k) \\ \omega(k+1) \\ \mathbf{q}(k+2) \end{bmatrix} \quad (7.28)$$

the dynamic constraints can be enforced through the following constraint:

$$\mathbf{F}(k)\mathbf{s}(k) = \mathbf{z}(k) \quad (7.29)$$

where

$$\mathbf{F}(k) = \begin{bmatrix} \Delta t \mathbf{I}_{3 \times 3} & \mathbf{J} & \mathbf{0}_{3 \times 4} \\ \mathbf{0}_{4 \times 3} & -\frac{\Delta t}{2} \Xi(\mathbf{q}(k+1)) & \mathbf{I}_{4 \times 4} \end{bmatrix} \quad (7.30)$$

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{J}\omega(k) - \Delta t \omega(k)^\times (\mathbf{J}\omega(k) + \mathbf{h}_{\text{wheel}}(k)) \\ \mathbf{q}(k+1) \end{bmatrix} \quad (7.31)$$

These can both be computed at each time step and are independent of the solution variable. Equation 7.29 is thus a linear equality constraint and is subsequently convex.

7.5 Convex Objective Function

To autonomously satisfy the mission objectives, the guidance system needs to minimize

$$\psi = \arccos(\mathbf{p}_B \cdot \mathbf{DCM}(\mathbf{q})\mathbf{t}_I) \quad (7.32)$$

where \mathbf{t}_I is the target vector in the inertial frame, \mathbf{p}_B is the body vector to point at the target, and ψ is the angle between the two vectors. This can be formulated as a convex cost function by applying the methods used in the convexification of keep-in pointing constraints. Consider the keep-in constraint

$$\mathbf{q}^T \bar{\mathbf{P}} \mathbf{q} \leq \lambda \quad (7.33)$$

where $\bar{\mathbf{P}}$ is defined according to Equations 7.11-7.14 with $\mathbf{x}_I = \mathbf{t}_I$, $\mathbf{y}_B = \mathbf{p}_B$, and θ is some small angle. Smaller values of λ impose tighter keep-in constraints on the system and cause the inclusion cone to become smaller and smaller, until the angle between the target and pointing vectors is zero for $\lambda = 0$. Thus, the target and pointing vectors are aligned when

$$\mathbf{q}^T \bar{\mathbf{P}} \mathbf{q} = \left\| \bar{\mathbf{P}}^{1/2} \mathbf{q} \right\|^2 = 0 \quad (7.34)$$

The angle ψ is driven to zero by minimizing the cost function

$$cost = \left\| \bar{\mathbf{P}}^{1/2} \mathbf{q} \right\| \quad (7.35)$$

This is a convex objective function, since norms are convex. If the reorientation maneuver needs to end at rest, the cost function can be modified according to:

$$cost = k_q \left\| \bar{\mathbf{P}}^{1/2} \mathbf{q} \right\| + k_\omega \|\omega\|^2 \quad (7.36)$$

where the k_q and k_ω are weights on the pointing error and slew rates, respectively.

This is also convex, since it is an affine combination of convex functions [29]. k_ω is defined by the switching function

$$k_\omega = \begin{cases} 0; & \psi < \psi_{small} \\ k_{w,c} < k_q; & \psi > \psi_{small} \end{cases} \quad (7.37)$$

so that the slew rate is only driven to zero at the end of the maneuver. So long as the slew rate remains within its imposed bounds, slewing more slowly would only hurt the system by increasing the maneuver time. The weight on the slew rates should always be less than the weight on the pointing error, as the system would otherwise prioritize staying at rest over decreasing the pointing error. The choice of $k_{w,c}$ is fairly trivial, as any small weight will suffice. Its inclusion in the cost function is sufficient to drive the final angular rates to zero, provided the weight is not vanishingly small with respect to k_q .

As mentioned in Section 7.1, the discretization can cause the solver to fail to find a feasible solution. In order to guarantee a feasible solution, the pointing constraints are relaxed:

$$\mathbf{q}^T \hat{\mathbf{P}}_i \mathbf{q} \leq \lambda_i + \nu_i, \quad i \in [1, n_{out}] \quad (7.38)$$

$$\mathbf{q}^T \tilde{\mathbf{P}}_i \mathbf{q} \leq \lambda_i + \nu_i, \quad i \in [1, n_{in}] \quad (7.39)$$

where ν is a relaxation variable. Pointing constraint violations are then penalized in the cost function, where M is some arbitrary large positive number:

$$cost = k_q \left\| \bar{\mathbf{P}}^{1/2} \mathbf{q} \right\| + k_\omega \|\omega\|^2 + M \|\nu\|^2 \quad (7.40)$$

The relaxation of the constraints can also be offset by adding conservatism back into the pointing constraints:

$$\mathbf{q}^T \hat{\mathbf{P}}_i \mathbf{q} \leq (1 - \delta) \lambda_i + \nu_i, \quad i \in [1, n_{out}] \quad (7.41)$$

$$\mathbf{q}^T \tilde{\mathbf{P}}_i \mathbf{q} \leq (1 - \delta) \lambda_i + \nu_i, \quad i \in [1, n_{in}] \quad (7.42)$$

where δ is some small number much smaller than 1. The larger this value, the more conservative the constraint becomes. This parameter should be adjusted via numerical simulation to determine an acceptable degree of pointing constraint violation or conservatism. This relaxation also offers the advantage of not requiring the initial attitude of the satellite to satisfy the pointing constraints. The big-M penalty on constraint violations will drive the attitude into the set of permissible orientations as quickly as possible.

Computing a control signal by solving the convex optimization problem using Equation 7.40 would generate a slew maneuver that satisfies the constraints. However, there is no consideration of control effort in this formulation; as such, the resulting control signal will rapidly oscillate between its maximum and minimum bounds and act like a bang-bang control law. To prevent this behavior, a penalty for control effort is appended to the cost function:

$$cost = k_q \left\| \bar{\mathbf{P}}^{1/2} \mathbf{q} \right\| + k_\omega \|\omega\|^2 + k_u \|u\|^2 + M \|\nu\|^2 \quad (7.43)$$

The weights k_q and k_u are parameters that must be carefully selected by the control designer. Increasing k_q will decrease the maneuver time and improve pointing accuracy, but

at the cost of greater control effort. The weight k_q is best defined by another switching function:

$$k_q = \begin{cases} k_{q, far}; & \psi > \psi_{small} \\ k_{q, close} > k_{q, far}; & \psi < \psi_{small} \end{cases} \quad (7.44)$$

Increasing the weight when the maneuver is almost complete causes the final pointing error to be small. The smaller weight when ψ is large results in lower control expenditure over large slew maneuvers. Combining the results from Section 7.3 through the current section gives the following MICP formulation of the CAG problem:

At every time step, k , solve the MICP problem for the solution variables

$$\mathbf{s} = \begin{bmatrix} \mathbf{u}(k) \\ \omega(k+1) \\ \mathbf{q}(k+2) \end{bmatrix}, \quad \nu = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_{n_{in}+n_{out}} \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{n_r} \end{bmatrix}, \quad \mu_i = \{0, 1\}$$

Minimize

$$cost = k_q \left\| \overline{\mathbf{P}}^{1/2} \mathbf{q}(k+2) \right\|^2 + k_\omega \|\omega(k+1)\|^2 + k_u \|\mathbf{u}(k)\|^2 + M \|\nu\|^2$$

subject to the control and slew rate bounds

$$\|\mathbf{u}(k)\|_\infty \leq u_{max}, \quad \|\omega(k+1)\|_\infty \leq \omega_{max}$$

pointing constraints

$$\mathbf{q}(k+1)^T \hat{\mathbf{P}}_i \mathbf{q}(k+1) \leq \lambda_i, \quad i \in [1, n_{out}]$$

$$\mathbf{q}(k+1)^T \tilde{\mathbf{P}}_i \mathbf{q}(k+1) \leq \lambda_i, \quad i \in [1, n_{in}]$$

$$\mathbf{q}(k+1)^T \tilde{\mathbf{P}}_i \mathbf{q}(k+1) \leq \lambda_i + M \mu_i, \quad i \in [1, n_r]; \quad \sum_{i=1}^{n_r} \mu_i \leq n_r - 1$$

and kinematic constraints

$$\mathbf{F}(k) \mathbf{s}(k) = \mathbf{z}(k)$$

$$\mathbf{F}(k) = \begin{bmatrix} \Delta t \mathbf{I}_{3 \times 3} & \mathbf{J} & \mathbf{0}_{3 \times 4} \\ \mathbf{0}_{4 \times 3} & -\frac{\Delta t}{2} \Xi(\mathbf{q}(k+1)) & \mathbf{I}_{4 \times 4} \end{bmatrix}$$

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{J}\omega(k) - \Delta t \omega(k)^\times (\mathbf{J}\omega(k) + \mathbf{h}_{wheel}(k)) \\ \mathbf{q}(k+1) \end{bmatrix}$$

7.6 Simulation Results

To demonstrate the system, a simulation of a ram-pointing mission in which the -Z body axis of the Bevo-2 satellite must be aligned with the instantaneous velocity vector is presented. The simulation includes environmental disturbance torques, actuator dynamics, and uncertainty in the moment of inertia of the spacecraft. The simulation also includes MEKF state estimation using simulated sensor measurements with noise and biases. For details on the ADC component models and simulation environment, please refer to Chapters

2 and 3.

For the ram-pointing objective, the pointing vector and target vectors are:

$$\mathbf{p}_B = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}; \mathbf{t}_I = \mathbf{v}_I$$

where \mathbf{v}_I is the current inertial velocity vector.

The initial conditions are:

$$\mathbf{q}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \omega(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The slew rate and control bounds are given by:

$$u_{max} = 1e - 3N; \omega_{max} = 2deg/s$$

The keep-out constraints are imposed by the sensitive star tracker on the -Z axis of the spacecraft.

$$\mathbf{x}_1 = \mathbf{r}_{sun}; \mathbf{x}_2 = \mathbf{r}_{moon}$$

$$\mathbf{y}_1 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}; \mathbf{y}_2 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

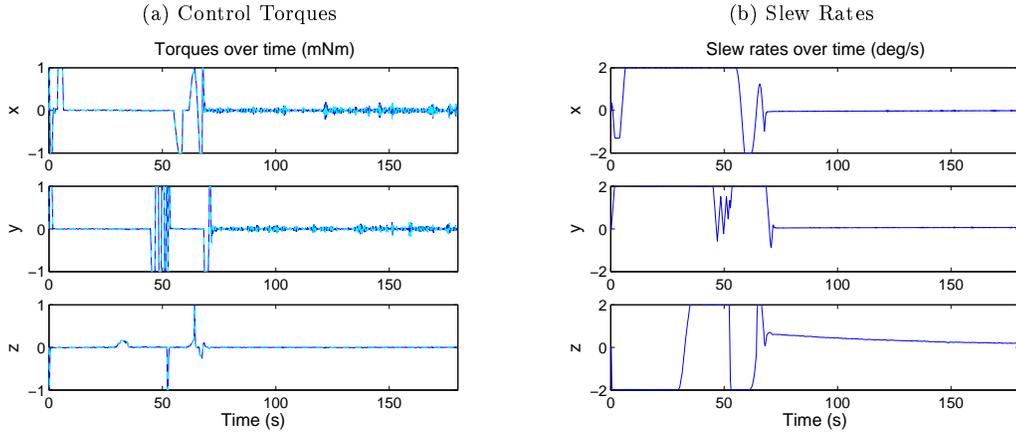
The keep-in constraints are OR constraints imposed by the two sun sensors on the +X and +Y axes of the spacecraft.

$$\mathbf{x}_3 = \mathbf{r}_{\text{sun}}; \mathbf{x}_4 = \mathbf{r}_{\text{sun}}$$

$$\mathbf{y}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \mathbf{y}_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

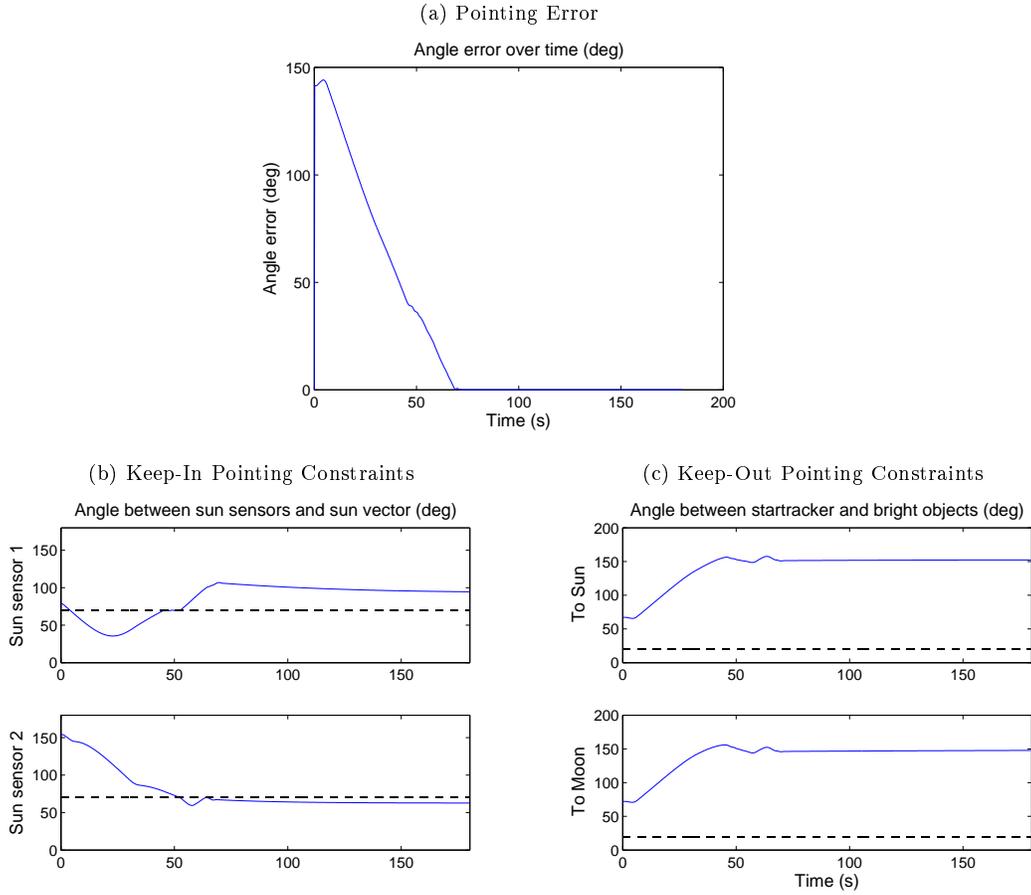
This MICP is solved at time intervals of 0.25 seconds using the CVX convex optimization modeling toolbox and Gurobi optimization software [33, 34].

Figure 7.2: Bounded Control and Slew Rates



The control torques and slew rates saturate at their respective bounds, as illustrated in Figure 7.2. Note that Figure 7.3a demonstrates that the mission objective of aligning the -Z body axis with the instantaneous velocity vector is achieved. The maneuver time is about a minute, and the final pointing error is less than 0.05 degrees. The sun sensor angles with respect to the Sun vector are plotted in Figure 7.3b. The mixed integer formulation successfully imposes the OR condition on the keep-in constraints, as is evident in the manner in which the sun sensors hand off responsibility for providing a Sun vector measurement. When the Sun vector leaves the inclusion cone of the first sensor at around 50 seconds, it

Figure 7.3: Pointing Error and Pointing Constraints



enters the inclusion cone of the second. Thus, a Sun vector measurement is always available to the navigation filter. Figure 7.3c shows that both the Sun and the Moon stay outside of the exclusion cone of the sensitive star tracker.

Chapter 8

Momentum Management

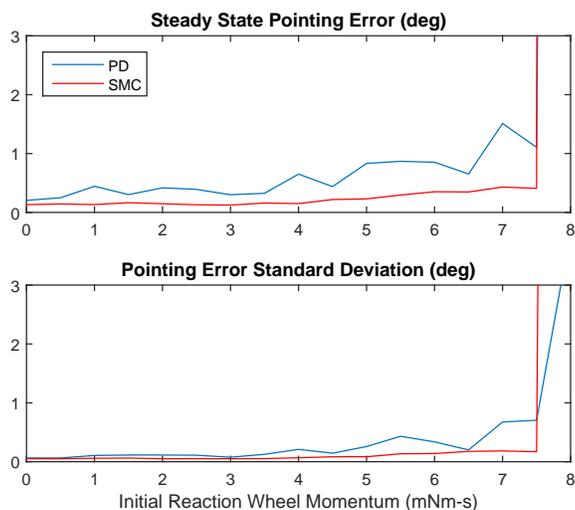
This chapter discusses the magnetorquer controller that is used to manage the momentum level in the reaction wheels. A discussion of the importance of momentum management is first presented to motivate the development of a momentum management system, followed by a description of the controller and its performance.

8.1 Reaction Wheel Saturation

The change in the combined momentum of the spacecraft and wheel assembly is equal to the external torques on the system. Thus, the wheels will gradually accumulate angular momentum from the environmental torques acting on the satellite. However, the reaction wheels can only store a finite amount of angular momentum. If a reaction wheel controller commands a torque that would result in exceeding this maximum momentum level, the reaction wheel will saturate at this limit and be unable to provide the torque commanded. This lack of controllability would prevent the Sun acquisition controller from spinning the satellite at the desired rates, and prevent the pointing controllers and CAG system from reorienting the satellite into the desired orientation. Thus, another set of control devices which can unload the momentum in the wheels is essential for preventing wheel saturation and maintaining controllability.

Momentum management is also useful for improving pointing accuracy. The gyroscopic torque terms scale with the reaction wheel spin rate, and reaction wheel jitter torques

Figure 8.1: Pointing Accuracy with Varying Wheel Momentum



scale with the wheel speed squared. Keeping the reaction wheel speeds low therefore minimizes these perturbing torques on the satellite. A simulation of the PD and SMC pointing controllers was performed with varying levels of initial momentum in the wheels to illustrate this point.

The simulation includes environmental disturbance torques, actuator dynamics, and uncertainty in the moment of inertia of the spacecraft. The simulation also includes MEKF state estimation using simulated sensor measurements with noise and biases. For details on the ADC component models and simulation environment, please refer to Chapters 2 and 3. The simulation results are presented in Figure 8.1. As shown in these plots, the pointing accuracy degrades at increased momentum levels, and the ADC system loses controllability entirely when the momentum level in the reaction wheels is above $7.5\text{mNm}\cdot\text{s}$ due to wheel saturation.

8.2 Momentum Controller

The momentum in the wheels is managed by commanding a magnetic dipole μ_{cont} from the magnetorquers according to:

$$\mu_{\text{cont}} = \text{sat}(k_m(\Delta\mathbf{H} \times \mathbf{B}_{\mathbf{B}}), \mu_{\text{max}}) \quad (8.1)$$

where k_m is the controller gain, $\mathbf{B}_{\mathbf{B}}$ is the local magnetic field in the body frame, and $\Delta\mathbf{H}$ is the desired change in momentum given by:

$$\Delta\mathbf{H} = \mathbf{h}_{\text{wheel}} - \mathbf{h}_{\text{cmd}} \quad (8.2)$$

where \mathbf{h}_{cmd} is the commanded level of momentum in the reaction wheels. The commanded wheel momentum should be much less than \mathbf{h}_{max} to prevent the wheels from saturating. However, the wheels also have a small deadband region, so the commanded momentum level should be slightly offset from zero. The controller gain k_m should be selected such that the commanded magnetic dipole saturates at high values of $\Delta\mathbf{H}$ and scales accordingly as it decreases to zero. For the magnetorquers on-board the TSL satellites, a gain value of $1e7$ was selected. The torques on the satellite due to the magnetic dipole of the rods are given by Equation 2.20, which is restated here:

$$\mathbf{u}_{\text{mag}} = \mu_{\text{mag}} \times \mathbf{B}_{\mathbf{B}}$$

If the magnetorquers on-board the satellite are bang-bang devices, the continuous control law becomes:

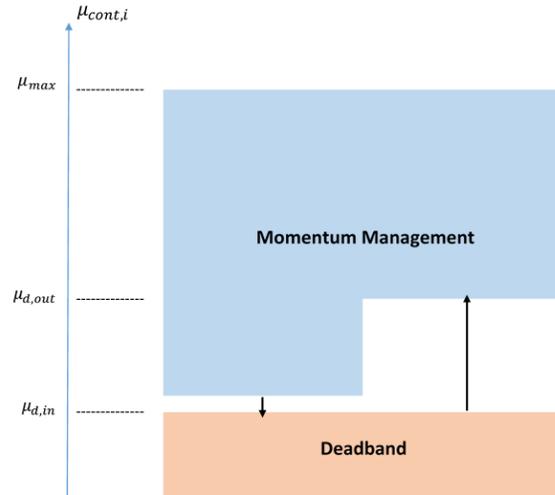
$$\mu_{\mathbf{mag}} = \text{sign}(\mu_{\mathbf{cont}}) \quad (8.3)$$

where the sign function is defined in Equations 6.24 and 6.25.

To continually manage the momentum level in the wheels and prevent them from saturating, this controller should be used whenever the reaction wheels are on. However, these magnetic dipole torques essentially become disturbance torques from the perspective of the reaction wheel controllers. Generally, these torques are so small that these disturbances are easily rejected by the reaction wheel controllers. If $\Delta\mathbf{H}$ is small, however, this results in the continuous dipole $\mu_{\mathbf{cont}}$ rapidly changing signs as $\Delta\mathbf{H}$ oscillates about zero. Since the magnetorquers are bang-bang devices, this will result in constant, abrupt switching in the commanded magnetic dipoles from full on positive to negative. This rapid switching will cause oscillatory pointing error in the reaction wheel controllers.

To address this problem, a tiered deadband region is imposed around the continuous dipole $\mu_{\mathbf{cont}}$ in each axis. Within the deadband region, the commanded dipole is zero. The deadband entry condition occurs when $\mu_{cont,i}$ falls below an inner deadband limit, $\mu_{d,in}$. The exit condition occurs when $\mu_{cont,i}$ rises above an outer deadband limit, $\mu_{d,out}$. This logic is illustrated in Figure 8.2. Imposing this deadband on the continuous dipole before applying the sign function to obtain the bang-bang control law prevents the rapid magnetic dipole switching scenario described above. The inner and outer deadband limits were set to $\mu_{max}/50$ and $\mu_{max}/5$, respectively, which provides sufficient distance between the deadband limits to prevent the system from constantly entering and exiting the deadband region.

Figure 8.2: Momentum Management Deadband Logic



8.3 Simulation Results

A simulation of the momentum in the reaction wheels being lowered from an initial momentum level of $5mNm \cdot s$ in each wheel is presented to demonstrate the momentum management capabilities of the system. The momentum management is performed concurrently with the reaction wheels being used by the PD pointing controller to maintain a commanded attitude. The same simulation environment described above was used to generate these results.

The desired momentum level to which the system is driven is $0.5mNm \cdot s$. This is far from the saturation limit of $10mNm \cdot s$, but offset from zero to avoid the reaction wheel deadband region. The simulated momentum management scenario uses Bevo-2's configuration of 2 bang-bang magnetorquers, so there is no magnetic dipole commanded from the Z body axis. A satellite with three orthogonal torque rods would simply have better performance than this baseline. As shown in Figure 8.3a, the momentum in the wheels is

successfully lowered to $0.5mNm \cdot s$ within one orbit. The plot of the pointing angle error in Figure 8.3b also illustrates the decrease in pointing angle error as the reaction wheel momentum level is lowered. The commanded magnetic dipoles as plotted in Figure 8.3c demonstrate that the controller gain was appropriately tuned, as the continuous dipole only saturates at high levels of momentum in the wheels. The tiered deadband structure is also shown to successfully prevent the bang-bang magnetorquers from overly rapid, continuous switching. The transitions between on and off are on the order of tens of seconds. This magnetic torque profile is easily rejected by the pointing controller, which is evident in the small pointing angle errors.

8.4 Lifetime Momentum Management

Satellites in LEO experience significant drag which gradually decays the orbit and lowers the altitude of the satellite until it disintegrates in Earth's atmosphere. As the satellite's altitude decreases, the density of the atmosphere increases exponentially. This causes greater drag torques acting on the satellite, resulting in greater momentum accumulation. If the momentum accumulation due to the external torques on the satellite exceeds the amount of momentum that the torque rods can unload from the wheels, controllability will be lost. To assess the momentum management capabilities of the ADC system over the lifetime of the satellite, the momentum accumulation from environmental torques over one orbital period was calculated for a circular orbit at altitudes of 400 to 150km. For the worst-case momentum accumulation scenario, the simulation assumes maximum solar activity and a satellite orientation that maximizes the projected area into the velocity vector. The maximum amount of momentum that can be unloaded from the reaction wheels by the magnetorquers is given by:

Figure 8.3: Momentum Management Simulation Results

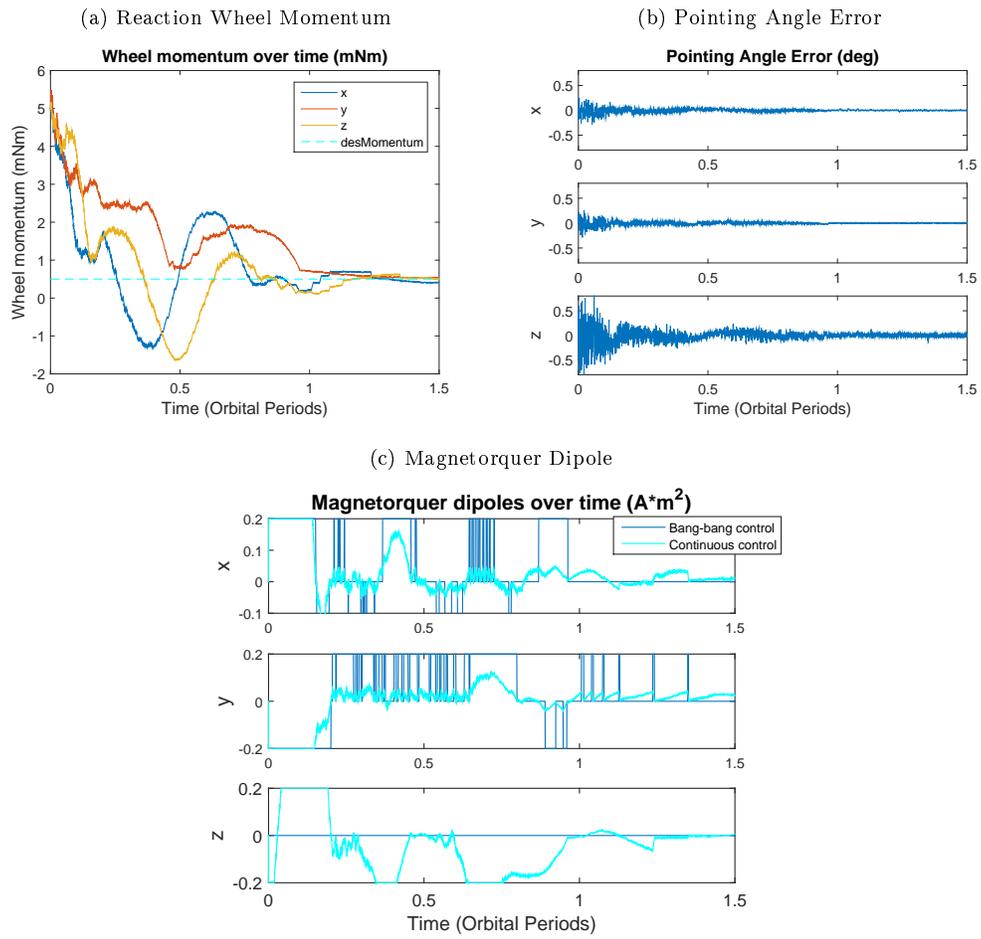
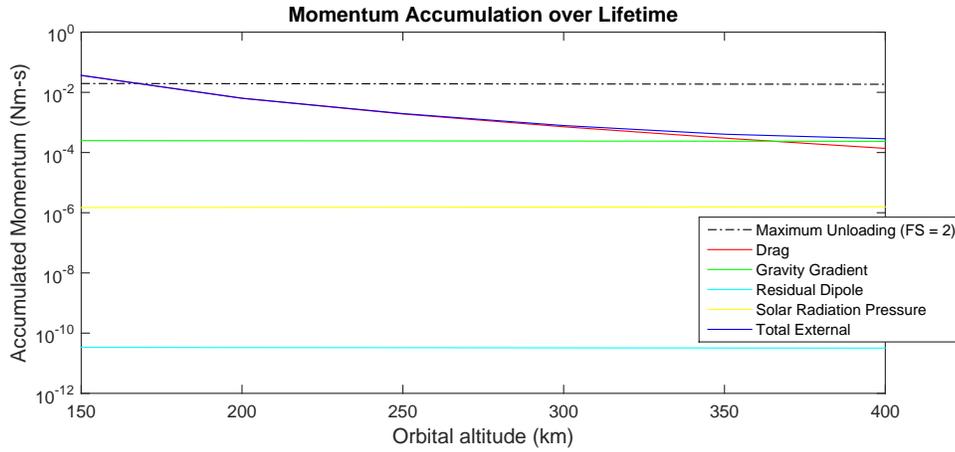


Figure 8.4: Momentum Accumulation over an Orbit at Varying Altitudes



$$\Delta H_{max} = \mu_{max} \|\mathbf{B}\| \quad (8.4)$$

where \mathbf{B} is the local magnetic field. This is plotted as a black, dashed line in Figure 8.4 with a factor of safety of two. As the altitude of the satellite decreases, ΔH_{max} increases due to the increased strength of the magnetic field at lower altitudes. However, this effect is far outweighed by the exponential increase in drag. As shown in Figure 8.4, the momentum accumulated over an orbit exceeds the momentum management capabilities of the satellite at an altitude of roughly 175 km. At this point, the reaction wheels will start accumulating momentum that cannot be unloaded until they are saturated and controllability is lost. However, the altitude at which this occurs is so low that the satellite cannot maintain this orbit, and it will quickly deorbit. This ADC system therefore maintains controllability over the entire duration of the satellite's lifetime.

Chapter 9

Conclusions

With the miniaturization of space instrumentation, small satellites are being increasingly called upon to collect data, perform experiments, and relay information at a fraction of the price. As these small satellite platforms gain popularity, they also gain responsibility for missions of greater complexity. In view of these demands, a flexible, robust attitude determination and control system was developed to meet a variety of mission objectives.

9.1 Summary of Work

Using a suite of commercial-off-the-shelf sensors and actuators which fit within the 3U CubeSat form factor, the ADC system presented in this thesis is capable of precisely estimating and controlling the attitude of the spacecraft. The system includes a multiplicative extended Kalman filter for attitude estimation using measurements from the sensors on-board the satellite. A Sun acquisition controller restores observability so that the navigation filter has enough measurement vectors to update the attitude estimate.

The ADC software also includes an unconstrained pointing system that reorients the spacecraft to align a commanded pointing vector with a given target. The pointing system can accept a secondary pointing constraint to minimize the angle between another pair of vectors without violating the primary constraint. This functionality can be used to maintain observability in the system to satisfy navigation requirements. The pointing system also includes a choice of two pointing controllers – a proportional derivative controller, and

a nonlinear sliding mode controller. Both of these controllers satisfy the maximum torque and slew rate constraints imposed by the ADC hardware.

A constrained attitude guidance system is proposed as an improvement upon the unconstrained system. Using convex optimization, the CAG system can enforce an arbitrary number of keep-in and keep-out constraints at every point along the attitude trajectory. It can also satisfy logical combinations of the attitude constraints, such as keeping the Sun within the FOV of either of the two sun sensors on-board the TSL satellites.

Finally, a momentum management system is demonstrated to successfully dissipate the momentum stored in the reaction wheels at any point during the lifetime of the satellite. Every part of the ADC system was extensively tested in simulation using realistic physical parameters based on the TSL satellites. It was demonstrated to be robust to environmental disturbance torques, sensor noise, and actuator dynamics. The system delivers arc-minute level pointing precision in the presence of these perturbations and uncertainties. Although the ADC system was specifically designed for use on-board ARMADILLO and Bevo-2, the algorithms developed are general-purpose and flexible enough to be used on the vast majority of satellite missions. The pointing system can accept pointing targets in a variety of frames, and the CAG system can handle an arbitrary set of attitude constraints. The control algorithms are also tunable, so the gains and controller parameters can be adjusted for different satellites as needed.

9.2 Current and Future Work

The majority of the ADC algorithms presented in this thesis have been integrated into the flight software on Bevo-2 and ARMADILLO. Both satellites make use of the unconstrained pointing system, as the CAG system has not been integrated for flight. They

also use the sequential optimal attitude recursion (SOAR) filter rather than the MEKF. The SOAR filter has improved performance with noisier sensor measurements, but the performance of the two filters is nearly identical when the sensor noise is low [21]. The performance of the system will therefore be as good or better with the SOAR filter as compared to the results presented in this thesis.

As of Spring 2015, both of these satellites are about to be delivered and flown within the year. Extensive software-in-the-loop testing was performed using the simulation environment while running the ADC flight software on the ADC flight computer. Hardware-in-the-loop testing is currently being performed on the ADC engineering design unit (EDU), shown in Figure 9.1a. The engineering design unit has three orthogonal torque rods, two sun sensors, three gyroscopes, a magnetometer, and one reaction wheel. The test setup is illustrated in Figure 9.1b. The ADC graphical user interface (GUI) is used to send commands to the ADC computer on the EDU which is running the flight software. These commands switch the ADC system between the following modes: turn actuators off, spin the reaction wheel at a set speed, use the sliding mode controller, use the pointing controller, acquire the Sun, and turn on the torque rods. The ADC GUI can also send commands to turn momentum management on or off, as well as set all of the controller parameters (e.g. gains, deadband limits, and boundary layer widths).

The MATLAB simulation environment is used to generate simulated sensor measurements corresponding to the satellite's simulated state in space. These simulated sensor measurements are consumed by the navigation filter rather than the true measurements produced by the sensors sitting on the ground. Depending on the mode set by the ADC GUI, the actuators will respond as commanded, and their outputs will be fed into the MATLAB simulation to propagate the attitude with the appropriate actuator torques. All of the

Figure 9.1: ADC Hardware-in-the-Loop Testing

(a) Engineering Design Unit



(b) Testing Set up



telemetry data from the ADC is displayed in the GUI, so the operator can verify that the states of the system are as expected.

Future work includes further hardware and systems-level testing to prepare the system for delivery. Rather than simulating commands from the command and data handling (CDH) subsystem using the ADC GUI, the interface between the ADC and CDH computers needs to be implemented and tested. If time permits, the CAG system will also be implemented on the flight system before delivery. The ADC system will then be flown for the first time on Bevo-2 and ARMADILLO. The data collected during these missions will be essential for on-orbit validation and assessment of the performance of the system.

Bibliography

- [1] S. Lee, A. Hutputanasin, A. Toorian, W. Lan, R. Munakata, J. Carnahan, D. Pignatelli, and A. Mehrparvar, “Cubesat design specification revision 13,” The CubeSat Program at Cal Poly SLO, <http://www.cubesat.org/index.php/documents/developers>, February 2014, [cited 12 Dec 2014].
- [2] M. Grewal and A. Andrews, “How Good Is Your Gyro?” *IEEE Control Systems Magazine*, February 2010, [cited 4 April 2015]. [Online]. Available: <http://ieeecss.org/CSM/library/2010/feb10/06-AsktheExperts.pdf>
- [3] H. Kjellberg, “Design of a CubeSat Guidance, Navigation, and Control Module,” Master’s thesis, The University of Texas at Austin, 2011.
- [4] D. J. Weeks, A. B. Marley, and J. R. L. III, “SMDC-ONE: An Army Nanosatellite Technology Demonstration,” *Proceedings of the 23rd Annual AIAA/USU Conference on Small Satellites*, vol. SSC09-X-3, Aug 2009.
- [5] “U.S. Army Taps Andrews for Second Kestrel Eye Satellite,” SpaceNews, February 2013, [cited 25 March 2015]. [Online]. Available: <http://spacenews.com/us-army-taps-andrews-for-second-kestrel-eye-satellite/>
- [6] M. D. Lemonick, “How a Pocket-Size Satellite Could Find Another Earth,” *Time Inc.*, May 2012. [Online]. Available: <http://content.time.com/time/health/article/0,8599,2114158,00.html>

- [7] S. Clark, "Earth observation satellites deployed from space station," *Spaceflight Now*, March 2014, [cited 25 March 2015]. [Online]. Available: http://spaceflightnow.com/station/exp38/140304flock/#.VRNuhPnF_GA
- [8] "Isipod cubesat deployer brochure version 7.11," Innovative Solutions in Space, http://www.isispace.nl/brochures/ISIS_ISIPOD_Brochure_v.7.11.pdf, December 2014, [cited 6 April 2015].
- [9] S. Arestie, B. Hudson, and E. G. Lightsey, "Development of A Modular, Cold Gas Propulsion System for Small Satellite Applications," *Journal of Small Satellites*, vol. 2, no. 2, pp. 63–74, November 2012.
- [10] K. Brumbaugh, H. Kjellberg, and E. G. Lightsey, "In-Situ Sub-Millimeter Space Debris Detection Using CubeSats," *2012 AAS Guidance and Control Conference*, February 2012.
- [11] A. J. Joplin, "Development and Testing of a Miniaturized, Dual-Frequency, Software-Defined GPS Receiver for Space Applications," Master's thesis, The University of Texas at Austin, 2011.
- [12] "Smart Digital Magnetometer HMR2300," Honeywell International Inc., January 2012, [cited 4 April 2015]. [Online]. Available: www.magneticsensors.com
- [13] "SS-411 Two-Axis Digital Sun Sensors," Sinclair Interplanetary, 2011, [cited 4 April 2015]. [Online]. Available: <http://www.sinclairinterplanetary.com/digitalsunsensors>
- [14] C. McBryde, "A Star Tracker Design for CubeSats," Master's thesis, The University of Texas at Austin, 2012.

- [15] W. J. Larson and J. R. Wertz, Eds., *Space Mission Analysis and Design*, 3rd ed. Space and Technology Library, 1999, vol. 8.
- [16] “Spice toolkit, version n65,” NASA’s Navigation and Ancillary Information Facility, <http://naif.jpl.nasa.gov/naif/index.html>, 2014.
- [17] C. D. Brown, *Elements of Spacecraft Design*. Reston, VA: American Institute of Aeronautics and Astronautics, 2002, pp. 265–269.
- [18] “International geomagnetic reference field,” IAGA Division V-MOD Geomagnetic Field Modeling, <http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html>, 2010.
- [19] P. Navarez and R. Kuberry, “Magnetic characterization of radiometer atmospheric cube-sat experiment (race),” NASA Jet Propulsion Laboratory, Interoffice Memorandum 5137-14-014, 2014.
- [20] J. K. Hedrick and A. Girard, “Control of Nonlinear Dynamic Systems: Theory and Applications,” pp. 62–83, 2005.
- [21] J. A. Christian and E. G. Lightsey, “Sequential optimal attitude recursion filter,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, Nov-Dec 2010.
- [22] F. L. Markley, “Attitude error representations for kalman filtering,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, March-April 2003.
- [23] D. M. Roger Bate and J. White, *Fundamentals of Astrodynamics*. New York: Dover Publications, Inc., 1971.
- [24] “World geodetic system 1984 (wgs 84),” 2004. [Online]. Available: <http://earth-info.nga.mil/GandG/wgs84/index.html>

- [25] “Systems tool kit 10.0,” Analytical Graphics, Inc., <http://www.agi.com/products/stk/>, 2012.
- [26] B. Wie, *Space Vehicle Dynamics and Control*, 2nd ed. AIAA Education Series, 2008, pp. 425–444.
- [27] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Prentice Hall, 2002, pp. 601–617.
- [28] K. McDonald, “Magnetic control for spinning 3-unit science cubesat,” Master’s thesis, The University of Texas at Austin, 2014.
- [29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004, pp. 7–8;21–174.
- [30] Y. Kim and M. Mesbahi, “Quadratically Constrained Attitude Control via Semidefinite Programming,” *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 731–735, May 2004.
- [31] Y. Kim, M. Mesbahi, G. Singh, and F. Y. Hadaegh, “On the Convex Parameterization of Constrained Spacecraft Reorientation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1097–1109, July 2010.
- [32] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, “Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming,” *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, July-August 2002.
- [33] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, 2014.
- [34] “Gurobi optimizer, version 6.0,” Gurobi Optimization, <http://www.gurobi.com>, 2014.