Attitude and Formation Control Design and System Simulation for a Three-Satellite CubeSat Mission

by

Austin Kyle Nicholas

B.S., University of Illinois at Urbana-Champaign (2007)

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author……

Department of Aeronautics and Astronautics
May 23, 2013

Certified by……

David W. Miller
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by……

Jeffrey A. Hoffman
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by……

Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Attitude and Formation Control Design and System Simulation for a Three-Satellite CubeSat Mission

by

Austin Kyle Nicholas

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Spacecraft formation flight has been identified as a critical enabling technology for achieving many scientific, commercial, and military objectives. One of the primary challenges of a formation flight mission is the control of the relative motion between spacecraft. Before any flagship missions will launch, technology development missions will be required to demonstrate the utility and functionality of formation flying systems.

This thesis describes the complete attitude and formation control design for the MotherCube formation flight technology demonstration mission in LEO. A model of the spacecraft's sensors and actuators is developed and analyzed. Using curvilinear orbit theory, a simple LQR control law is used to generate a set of desired relative accelerations for formation control. A newly developed two-tier numerical allocation scheme is used alongside an independent PD attitude control law to generate a set of actuator commands which provides 3-axis attitude stabilization as well as formation control with guaranteed feasibility of actuator commands. An Extended Kalman Filter was developed to estimate the system attitude and angular rate from sensor measurements. To test these algorithms, a simulation environment was developed. This environment includes realistic models of space environment and the major perturbation effects which a LEO spacecraft formation would encounter. In order to improve the fidelity, a new intermediate-accuracy method for computing attitude-dependent aerodynamic and solar effects was also developed. Finally, results from the simulation are used numerically validate the dual-allocator approach, assess the performance of the control laws and provide system level metrics such as fuel use and required maneuver time.

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Jeffrey A. Hoffman
Title: Professor of Aeronautics and Astronautics

ACKNOWLEDGEMENTS

There are many people I would like to thank for helping to make completion of this thesis possible. First, my thesis advisors, Jeff Hoffman and Dave Miller, who were patient and supportive throughout my time at MIT and without whom I would not have been successful. Second, my officemate, Alex Buck, who was always willing to be a sounding board for crazy ideas and suggest new ways of approaching challenging problems. Finally, I would like to thank my loving family who have always encouraged me to follow my dreams and without whose unwavering support I never could have made it to MIT or completed this thesis.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant Number 1122374 and by MIT Contract Number AFS12-0207 (Distributed Satellite Systems).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

A distributed satellite system (DSS) is defined as a set of spacecraft which are physically separated but which form part of a bigger system which derives functionality from the contributions of each member. There are two types of distributed satellite systems: constellations and formations. A 'formation' (or 'cluster') is characterized by utilizing control laws which couple the states of the member satellites to achieve the desired behavior. Constellations, on the other hand, may be designed so that their orbits complement the other members, but they are controlled relative to their own pre-defined reference trajectories rather than dynamically based on the behavior of the other members. Historically, constellations have been where one spacecraft is incapable of accomplishing the mission. The best example of a spacecraft constellation is the Global Positioning System (GPS), which uses multiple broadcasting satellites in a variety of orbits so that users can triangulate their position from the time of flight of the broadcast signals. Constellations have also been used to ensure high-availability, global communication (Iridium) or for coordinated remote sensing ('A-Train').

Spacecraft formations, on the other hand, typically seek to replace monolithic spacecraft with a set of smaller spacecraft. This 'fractionation' is expected to improve some of the following categories:

- **Flexibility** – Because the members of a spacecraft formation are not physically connected, the possibility of fast, dynamic reconfiguration allows a single set of hardware to be designed and used for a wide variety of tasks, albeit at a complexity cost [1].
- **Robustness** – A system with many elements is often capable of complete or partial functionality in the case of single element failure. Having a distributed architecture also means that partial upgrade or replacement of an in-place system is possible without needing to replace the entire system all at once.
- **Cost-effectiveness** – One of the inherent benefits of an architecture composed of smaller pieces is the learning curve benefit, where making many similar satellites is thought to be

less costly than making one monolith [2]. Additionally, the flexibility and robustness benefits often also manifest themselves as a cost savings by reducing the requirements for hardware redundancy because the consequence of a single failure is much less severe. It also allows for reuse of common system elements.

DARPA's System F6 concept [3] is an embodiment of these first three benefits of using a formation flying system. Its objective is to demonstrate the fractionation of individual subsystem functions within individual spacecraft which fly in formation and communicate wirelessly within the formation. For instance, there would be one spacecraft dedicated to ground communication, one to power, one for the payload operations, and so on.

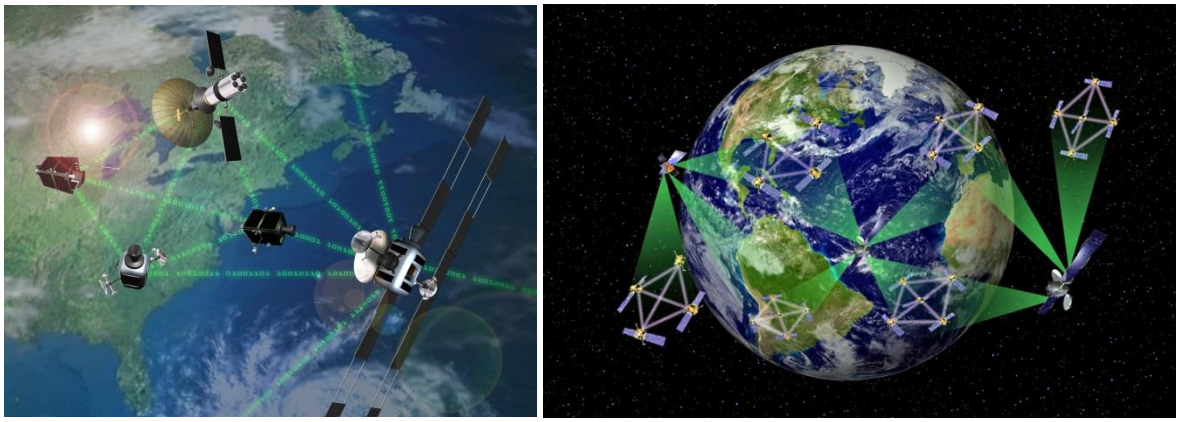

Figure 1.1 - Artist's Depictions of DARPA's System F6 [3]

The other benefit which can be leveraged from a formation flight mission is for payload performance:

- **Performance –** There are several types of spacecraft payloads which benefit from having a long baseline between sensors. Interferometers (often used for astronomy) are one striking example, where using elements separated by a long baseline can yield results equivalent to a filled aperture much larger than what could be physically realized in a space system.

The now-cancelled Terrestrial Planet Finder (TPF) mission [4] is the embodiment of why a formation would be chosen for its enhanced performance. During its design, the TPF team was investigating two separate concepts for the mission to perform spectroscopy on exoplanets. The

first design was a traditional single-aperture monolith, which would have been on the order of 10m in diameters.

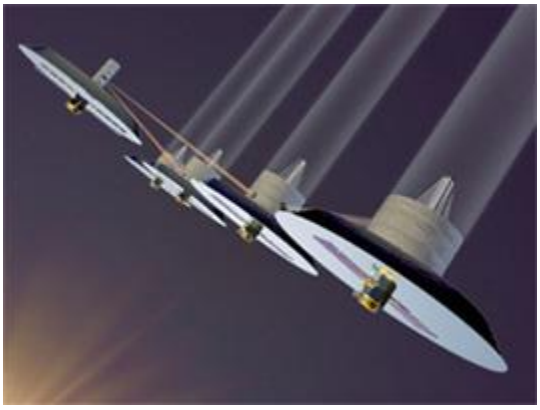

Figure 1.2 - Artist's Rendition of the Terrestrial Planet Finder Mission [5]

The second concept called for four 3.5m diameter apertures flying approximately 1000m in separation. Thus, a much smaller system would be as capable (or perhaps more so) than a monolith.

More recently, the PRISMA mission [6] was launched to demonstrate chemical-based propulsion for microsatellites as well as formation flight, homing, and rendezvous. The formation has two agents, only one of which has translational control.

Of course, using a formation flight architecture also poses some challenges. These challenges include the transfer of information (and potentially power or mass) and for the determination and control of the relative positions. These challenges are augmented by additional mission-specific hardware or operational constraints. This thesis will address the issues surrounding the control of the attitude and relative positions of a formation of Cubesats.

1.2 MOTHERCUBE MISSION OVERVIEW

Although most of the ideas contained in this thesis are generalizable, the need for most of them originated from a requirement or difficulty in designing the Attitude Determination and Control System (ADCS) for the MotherCube mission. Additionally, all of the numerical results are presented

using parameters from this mission. A brief description is given to familiarize the reader with some of the mission-specific objectives and challenges.

1.2.1 Mission Objectives (Relevant to ADCS)

1.2.1.1 Payload Objective: Radio Geolocation

The primary objective of the MotherCube (also called DSS) mission is to identify the location of radio sources by using three formation-flying satellites with receiver antennas to triangulate its position. On this mission, the radio sources will be VHF sources located on Earth, so the term 'geolocation' is often used to describe the payload objective. The mission also aims to demonstrate the on-orbit behavior of electrospray thrusters and satellite formation flight.

1.2.1.2 Demonstrate: Electrospray Thrusters

Electrospray thrusters (a type of colloid thrusters) are a form of electric propulsion [7]. The thrust is produced by electrostatic acceleration of microscopic charged droplets. These systems have several advantages over traditional (chemical) propulsion, which are:

- High Specific Impulse – Electrically accelerating the fuel allows a much higher exhaust velocity than can be achieved using chemical propellants and gas nozzles. Specific impulses in the range of 1000 – 5000 seconds may be obtainable in a fully-developed system.
- Precision Control – The thrust can be controlled to very high precision by varying both the voltage level and the length of the pulse. Most chemical systems have a much coarser level of control due to the 'minimum impulse bit', which is typically driven by the plumbing of the chemical system.
- Scalability – the fundamental unit is a conical emitter tip, which has a diameter on the order of hundreds of micrometers. The thrusters are simply arrays made of thousands of emitter tips. This makes electrospray thrusters uniquely suited to propulsion for small satellites, as the arrays can easily be scaled to fit the smaller form factor where chemical systems become significantly less efficient.
- Simple Plumbing – Electrospray thrusters do not require any moving parts, valves, or pressurized tanks. The fuel is drawn into the thruster via a porous substrate and operates in response to an applied voltage command.

The system also comes with some disadvantages:

- Electric Energy – Unlike chemical propellant where the energy used to accelerate the reaction mass is derived from the chemical potential of the combustion reaction, electric propulsion requires an external energy source.
- Complicated Power System – Electrospray systems require power at very high voltage (>1000 V) and low current (< 1 mA), which requires special electronics to convert and regulate the power.
- Low Thrust – The consequence of the aforementioned energy and electronics requirements generally restrict feasible systems to thrust levels which are many orders of magnitude smaller than chemical systems. This significantly affects the types of trajectories and maneuvers which may be utilized, as impulsive maneuvers have lower delta V than low thrust maneuvers.

1.2.1.3 Demonstrate: Formation Flight

A spacecraft "formation" (or "cluster") is defined as a group of spacecraft operating in close proximity such that control of their relative position is important. This differs from a "constellation" of satellites (such as the GPS satellites) which work together to achieve a common goal but do not actively control their distance from the other satellites in the constellation. Spacecraft formation flight is an important technology required for next-generation space telescopes, such as Terrestrial Planet Finder.

1.2.2 Basic Satellite Hardware Configuration

For this report, the dusk-dawn sun synchronous concept is assumed. At the time of writing, this concept is considered more likely and preferable and it matches the analysis performed thus far. It is possible that the hardware configuration could change if a different orbit is utilized.

The primary components of the MotherCube hardware are:

1. Electrospray Thrusters
2. Solar Panels
3. GPS Antenna
4. Patch Antenna (for ground communication)
5. Payload Antenna
6. Electronics (includes IMU and omni-directional antenna for inter-satellite communications)

7. Magnetic Torque Coils (around the edges of the bus frame and the solar panels)

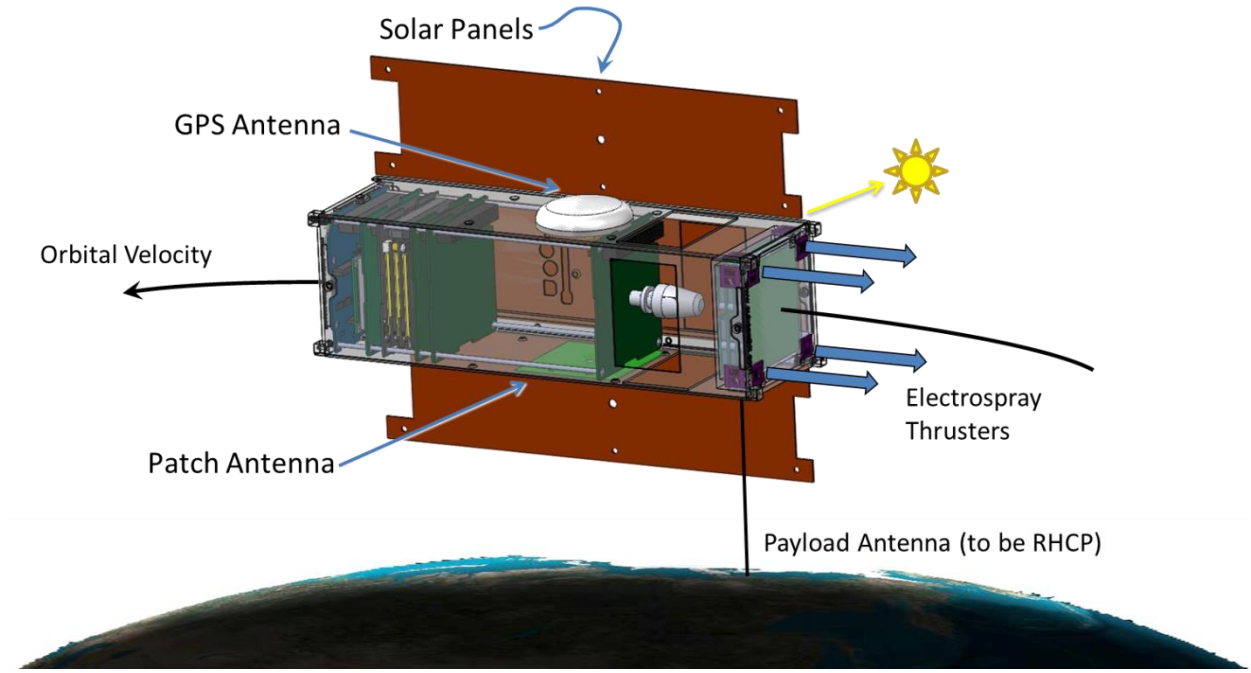

Figure 1.3 - MotherCube Hardware Configuration

1.3 THESIS OVERVIEW

The ultimate goal of the ADCS software design is to craft a set of algorithms which, when implemented onboard the spacecraft, produce the desired behavior. However, in order to evaluate such algorithms, a computer simulation is used to virtually replicate the space environment and its interaction with the spacecraft's actuators and sensors.

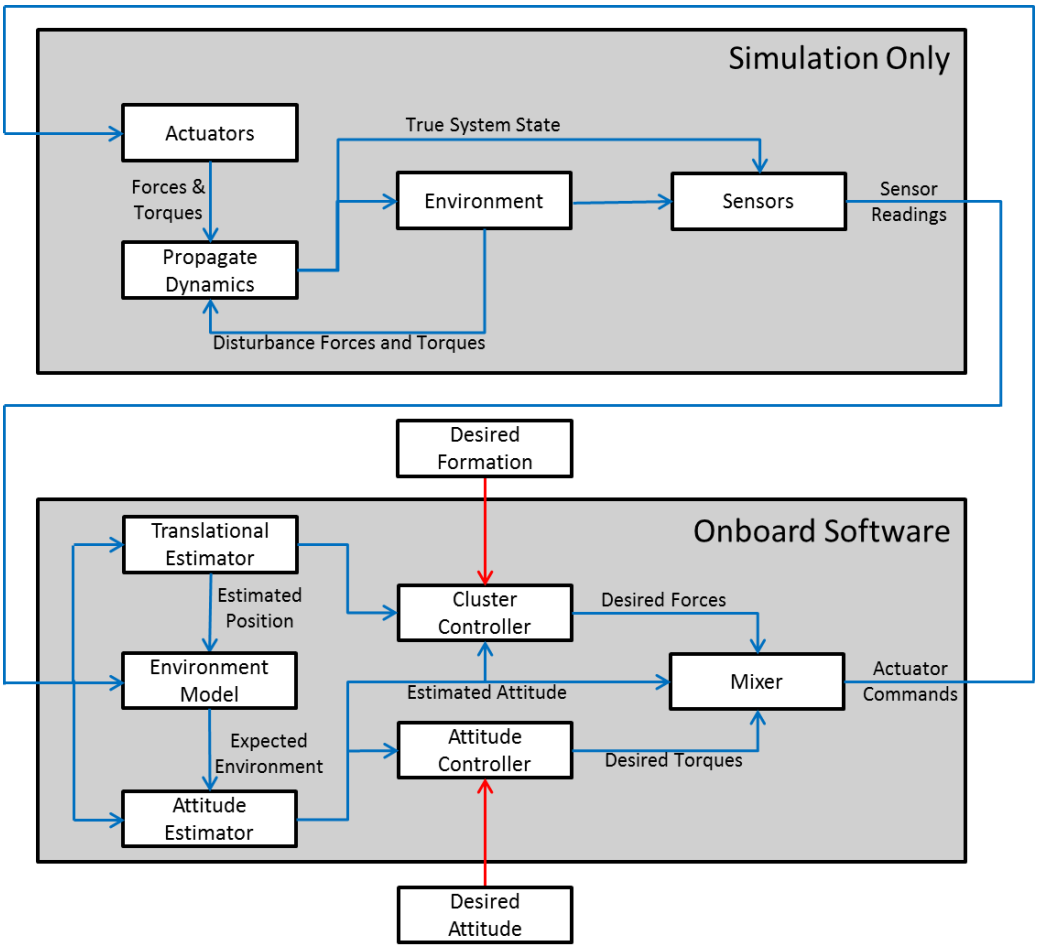

Figure 1.4 - High Level Simulation Diagram

This thesis is divided into 12 chapters. Each chapter is essentially one or two of the blocks as shown in Figure 1.4. Chapter 2 provides a literature review and a gap analysis. Chapter 3 discusses the variety of coordinate frames and other conventions which are used throughout the thesis. Chapter 4 derives the model of the spacecraft's physical properties, actuators, and sensors. Chapter 5 derives the control law for relative formation positional control and the formation allocator which assigns control actions to individual spacecraft in the formation. Chapter 6 presents two control laws for regulating the attitude and attitude rate of the spacecraft. Chapter 7 presents the local allocator (or 'mixer') function which assigns actuator commands based on a set of desired forces and torques. Chapter 8 discusses the estimation of the system state and presents an estimator. Chapter 9 derives an intermediate-accuracy ray-tracing method for computing attitude-dependent forces and torques resulting from aerodynamic or solar effects. Chapter 10 presents an overview of

the simulation effort, including a look at the environmental models which are inputs to the simulation. Chapter 11 presents the simulation results for four use cases. Finally, Chapter 12 concludes the thesis with a summary and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

In the last decade, a great deal of research on the guidance, navigation, and control of spacecraft formation flight missions has emerged due to its identification as a critical technology along the path to many NASA objectives (such as exoplanet detection and characterization [4], synthetic aperture radar [8], and gravitational mapping [9]). This section will discuss some of the relevant literature in the control of formations, which is defined as the generation of control actions to achieve the desired system behavior.

A dedicated two-part survey of formation flight guidance [10] and control [11] by gives an excellent overview of the major types of formation flight control along with numerous references to examples of each. This survey broadly groups formation flight control methodologies into two dynamical regimes. The dynamics can be broadly grouped as:

- **Deep Space (DS)** – This is defined to be a region of space where spacecraft translational dynamics are well approximated by a double integrator model.
- **Planetary Orbital Environment (POE)** – Significant environmental disturbances such as gravity or aerodynamic drag affect spacecraft motion.

The survey further divides the types of control into five categories, unrelated to the dynamic environment:

- **Multi-Input, Multi-Output (MIMO)** – In this architecture, the dynamics for the entire formation are treated as one large MIMO system and modern methods of control are directly applied. The primary benefit of this architecture is its optimality and stability, which are due to the fact that the full state of the entire formation is available. However, the primary downside is high communication requirements and problems with scalability for large formations. The MIMO formulation is not robust with respect to changes in formation size, because the full state is used in the control design.

- **Leader/Follower (L/F)** – Also called Chief/Deputy, this is the most studied type of formation control because it simplifies the control of an entire formation into individual tracking problems by selecting a strict hierarchy whereby Follower satellites control their position relative to a designated Leader(s). The most common architecture is the single-layer L/F, where all spacecraft follow the same leader. Another common L/F architecture is the chain, where each spacecraft follows the preceding one. An important feature of this architecture is the relationship between any two spacecraft MUST be either leader, follower, or unrelated. The primary benefit to this architecture is its robustness to changes in formation. Adding a new member does not affect any of the existing members because the new member will be a Follower. In the event of the failure of a leader, only its followers are affected and they can be reassigned to new leaders. The one major downside to this architecture is that it is not globally optimal.
- **Virtual Structure (VS)** – The objective of this method of formation control is to produce formation behavior which mimics the behavior of objects embedded in a rigid structure such as a truss. The most well-known example of this type of controller is the Terrestrial Planet Finder mission [4].
- **Cyclic** – A cyclic controller is similar to a L/F architecture but without the hierarchal constraint. In this way, two spacecraft can both feedback on their relative state. Many formulations of cyclic control lend themselves to decentralized control by creating relationships among the nearest neighbors in a formation.
- **Behavioral** – A behavioral controller combines the output of multiple controllers designed for different behaviors, where formation control is considered a mandatory behavior. For instance, a L/F architecture could be combined with a collision avoidance algorithm. Rather than a standalone architecture, the behavioral control methodology is an augmentation of one or more of the other control methodologies.

In addition to these five categories, a recent special focus has been centered around 'formation initialization' [12] [13], which seeks to analytically or numerically identify collision- or drift-free trajectories based only on the initial conditions [14]. This strategy typically results in one or many open loop maneuvers which produce acceptable behavior for some time period, at which point the formation is reinitialized. The assumption is typically that impulsive thrust is used to enforce the initialization condition, so it is not suitable for electric propulsion.

2.2 GAP ANALYSIS

One significant limitation in the utility of much of the formation flight literature is that it assumes unconstrained thrust direction while neglecting attitude dynamics. For large systems with impulsive thrust, this is perhaps well justified, as a single-thruster spacecraft could rotate to the desired attitude between maneuvers. However, for electrically propelled formations or for small satellite formations with significant hardware limitations due to size, this is a significant limitation. Some limited work has been done in this area to demonstrate closed periodic trajectories using a single unilateral thruster with passive magnetic stabilization [15] [16].

This thesis seeks to address this limitation – the conversion of desired relative accelerations (between spacecraft) to actuator commands for body-fixed thrusters with saturation constraints. An extension of this allows for the interaction with other spacecraft attitude actuators to simultaneously control the spacecraft attitude according to an independent attitude control law.

CHAPTER 3

COORDINATE SYSTEMS AND NOTATION

3.1 UNITS

All quantities are expressed in metric units. Specifically, the MKS notation is used, which means that distances are expressed in meters, mass is expressed in kilograms, and time is expressed in seconds. All angles are expressed in radians and all coordinate frames are assumed to be right-handed.

3.2 NOTATION

Vector quantities are denoted with bold type. $\boldsymbol{x}$ is a vector, x is not a vector.

Accent characters are used with specific meaning throughout the thesis. These uses are:

$(.)$	True Value
$\overline{(.)}$	Measured Value
$\hat{\boldsymbol{e}}$	Unit Vector
$\widehat{(.)}$	Estimated Value
$\widetilde{(.)}$	Modeled Value
$\dot{(.)}$	Time Derivative of Value

3.3 QUATERNION CONVENTIONS

The default attitude representation used in this report is quaternions because they are computationally faster and have lower storage requirements. This thesis does not give a full overview of quaternion math. This can be found in [17] and [18]. However, the most notable elements will be repeated here for clarity. The most important item is that the quaternions used here have the scalar element fourth.

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = q_1 i + q_2 j + q_3 k + q_4 \tag{3.1}$$

Additionally, because quaternions have a sign ambiguity, the additional constraint of $q_4 \geq 0$ is imposed. If at any point, q4 becomes negative, then the entire quaternion undergoes a sign change, which preserves the physical meaning of the quaternion. Finally, the notation used in this thesis for the skew matrix is as follows:

$$[\boldsymbol{\omega} \times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3.2}$$

3.4 Coordinate Frame Notation and Conventions

This section describes the notations which will be used for all vector quantities which are expressed in a particular coordinate frame. A single superscript is used to denote the frame in which a vector quantity appears. The following table is a complete list of all coordinate frames and their superscript designations:

Table 3.1 - List of Coordinate Frame Superscripts

$(.)^I$	Inertial (ECI)
$(.)^{\mathcal{B}}$	Body
$(.)^E$	ECEF
$(.)^N$	NED
$(.)^{\mathcal{L}}$	LVLH
$(.)^{\mathcal{R}}$	Ray Tracing Body

For quantities which describe the relationship between two coordinate frames, the superscript is expressed in the 'to-from' notation. Direction cosine matrices (also called 'rotation matrices') are typically denoted by a capital 'R' and quaternions are denoted by a lowercase 'q'. Thus, quantities are related in the following way:

$$\boldsymbol{x^{\mathcal{B}}} = R^{\mathcal{B}I}\boldsymbol{x^{I}} \tag{3.3}$$

$\boldsymbol{x^{\mathcal{B}}}$ is a vector expressed in the body frame, $\boldsymbol{x^{I}}$ is the same quantity expressed in the inertial frame, and $R^{\mathcal{B}I}$ is the DCM which relates the two frames. This notation is very intuitive, because equations which appropriately transform the quantities must match adjacent superscripts.

3.5 Earth-Centered Inertial (ECI)

The inertial (also called 'global' or 'fixed') is referenced to the J2000 ECI reference frame. It is defined as the Earth's Mean Equator and Equinox at 12:00 Terrestrial Time on 1 January 2000. The X-axis is aligned with the mean equinox. The Z-axis is aligned with the Earth's spin axis or celestial North Pole. The Y-axis is rotated by 90° East about the celestial equator. This system is shown in Figure 3.1.

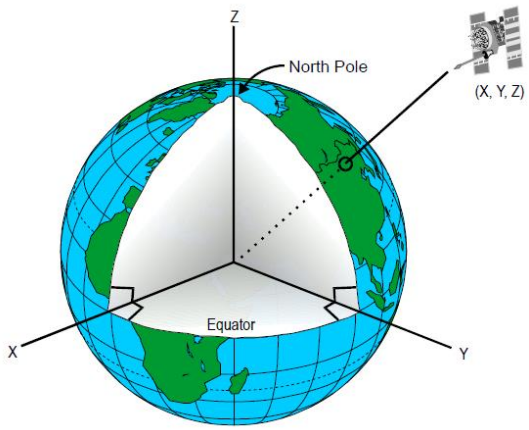

Figure 3.1 - Inertial Coordinate System [Image: Wikimedia]

3.6 Earth-Centered, Earth-Fixed (ECEF)

ECEF coordinates are very similar to ECI coordinates except that the reference frame rotates with the Earth such that the X^{E} axis intersects the sphere of the Earth at 0° latitude and 0° longitude, the Z^{E} axis is aligned with the earth's spin axis and the Y^{E} axis completes the right handle system. The rotation matrix which relates positions in ECEF and ECI is:

$$R^{EI} = \begin{bmatrix} cos\,\Theta & sin\,\Theta & 0 \\ -sin\,\Theta & cos\,\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$\Theta = 280.46061837° + 360.98564736628° * d_{2000} \tag{3.5}$$

Where d_{2000} is the fractional number of days since noon on January 1, 2000.

3.7 North-East-Down (NED)

Also known as geodetic coordinates, this coordinate system is parameterized by latitude (ϕ), longitude (λ) and height or altitude (h). The conversion from these parameters to ECEF coordinates is straightforward:

$$\boldsymbol{r^E} = \begin{bmatrix} x^E \\ y^E \\ z^E \end{bmatrix} = \begin{bmatrix} (N+h)\,cos\,\phi\,cos\,\lambda \\ (N+h)\,cos\,\phi\,sin\,\lambda \\ [N(1-e^2)+h]\,sin\,\phi \end{bmatrix} \tag{3.6}$$

Conversion from NED to ECEF is much less straightforward. One approximation [18] which is valid for heights below 1000 km is:

$$\phi = atan\left(\frac{z^E + \bar{e}^2 b\,sin^3\,\psi}{p - e^2 a\,cos^3\,\psi}\right) \tag{3.7}$$

$$\lambda = atan2(y^E, x^E) \tag{3.8}$$

$$h = \frac{p}{cos\,\phi} - N \tag{3.9}$$

$$p = \sqrt{x^{E\,2} + y^{E\,2}} \tag{3.10}$$

$$N = \frac{a}{\sqrt{1 - e^2\,sin^2\,\phi}} \tag{3.11}$$

$$\bar{e}^2 = \frac{a^2 - b^2}{b^2} \tag{3.12}$$

$$\psi = atan\left(\frac{za}{pb}\right) \tag{3.13}$$

The parameters used to describe the geoid are: $a = 6378137.0\,m,\ b = 6356752.3142\,m,\ e = 0.0818$. Note that N is the Earth's radius at a given latitude.

The rotation matrix which relates the ECEF coordinate frame and the NED coordinate frame is:

$$R^{NE} = \begin{bmatrix} -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi \end{bmatrix} \tag{3.14}$$

3.8 LOCAL-VERTICAL, LOCAL-HORIZONTAL (LVLH)

The most-used coordinate frame in this work is the Local-Vertical Local-Horizontal coordinate frame. This frame rotates with the chief orbit and is always aligned such that the x-direction is in the local vertical ('radial'), the z-direction is in the direction of the orbit normal, and the y-direction forms a right-handed coordinate system. The origin of the system is at the center of mass of the chief. In this work, it is assumed that all orbits are circular, which means that the y-direction is always in the velocity direction (also called 'tangential' or 'alongtrack' direction). This system is shown in Figure 3.2.

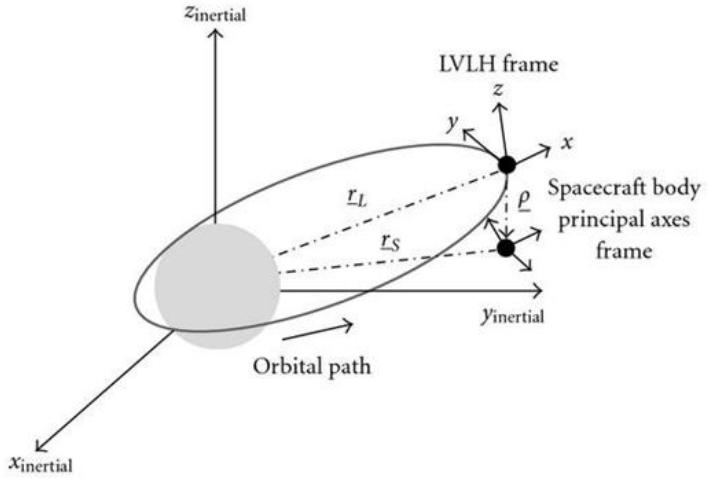

Figure 3.2 - LVLH Coordinate System

The rotation matrix between the LVLH frame and the inertial frame can be computed from the inertial position and velocity vectors of the chief as follows:

$$\hat{\boldsymbol{e}}_{\boldsymbol{R}}^{\mathcal{L}} = \frac{\boldsymbol{r}^{\boldsymbol{I}}}{\|\boldsymbol{r}^{\boldsymbol{I}}\|} \tag{3.15}$$

$$\hat{\boldsymbol{e}}_{\boldsymbol{N}}^{\mathcal{L}} = \frac{\boldsymbol{r}^{\boldsymbol{I}} \times \boldsymbol{v}^{\boldsymbol{I}}}{\|\boldsymbol{r}^{\boldsymbol{I}} \times \boldsymbol{v}^{\boldsymbol{I}}\|} \tag{3.16}$$

$$\hat{\boldsymbol{e}}_{\boldsymbol{T}}^{\mathcal{L}} = \hat{\boldsymbol{e}}_{\boldsymbol{N}}^{\mathcal{L}} \times \hat{\boldsymbol{e}}_{\boldsymbol{R}}^{\mathcal{L}} \tag{3.17}$$

$$R^{I\mathcal{L}} = \begin{bmatrix} | & | & | \\ \hat{\boldsymbol{e}}_{\boldsymbol{R}}^{\mathcal{L}} & \hat{\boldsymbol{e}}_{\boldsymbol{T}}^{\mathcal{L}} & \hat{\boldsymbol{e}}_{\boldsymbol{N}}^{\mathcal{L}} \\ | & | & | \end{bmatrix} \tag{3.18}$$

To be very clear, the position of a spacecraft in the LVLH frame can be written in the following ways:

$$\boldsymbol{r}^{\mathcal{L}} = x\hat{\mathbf{e}}_{\mathbf{R}}^{\mathcal{L}} + y\hat{\boldsymbol{e}}_{\boldsymbol{T}}^{\mathcal{L}} + z\hat{\boldsymbol{e}}_{\boldsymbol{N}}^{\mathcal{L}} = R^{\mathcal{L}I}\boldsymbol{r}^{\boldsymbol{I}} \tag{3.19}$$

It is worth noting that all of the deputy spacecraft use the same LVLH coordinate frame – relative to the chief. This transformation changes as the chief moves through the orbit. Because the chief's state is not always known to the deputies, it is often the case that the transformation between

inertial and LVLH coordinates is computed using the deputy's inertial states instead of the chief's, introducing some error.

3.9 Body Coordinate System

The body coordinate system is defined such that the GPS antenna is on the +X face of the main bus body, the solar panels are the on the +Y face, and the electrospray thruster slice is located on the –Z face. The origin is on the located at the (-X, -Y, -Z) corner of the thruster slice.

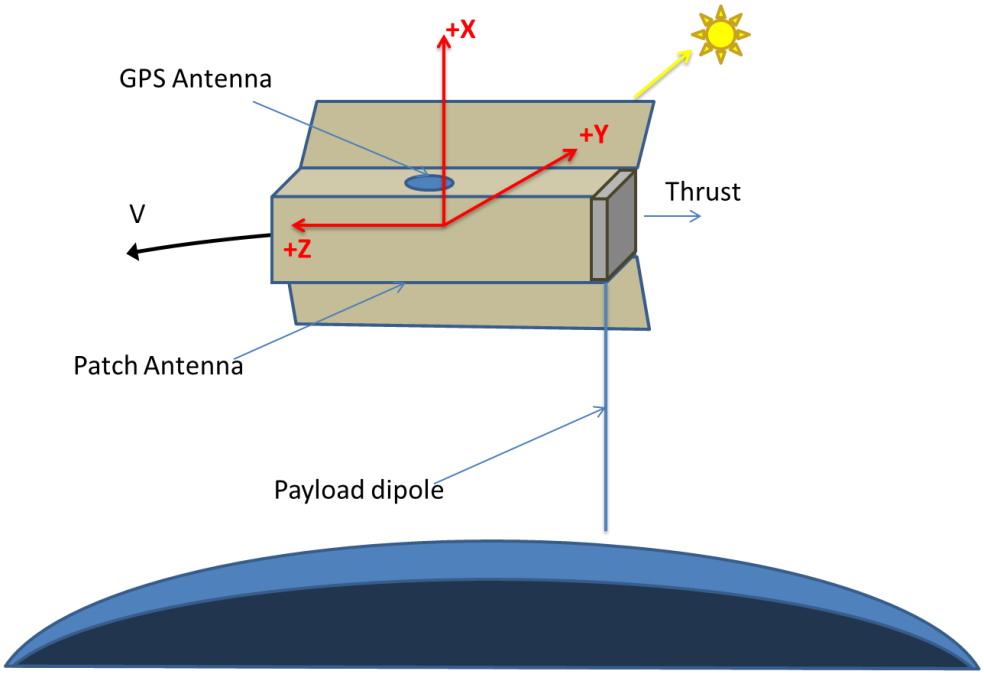

Figure 3.3 - Body Coordinate System during Nominal Mission

In the nominal configuration, the +X body axis is aligned with the +x LVLH axis, the +Y body axis is aligned with the –z LVLH axis, and the +Z body axis is aligned with the +y LVLH axis. Thus the nominal relation between the body and LVLH frames is given by:

$$R_{nominal}^{\mathcal{BL}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.20}$$

CHAPTER 4

SPACECRAFT MODEL

4.1 PHYSICAL PROPERTIES

The spacecraft are 3U Cubesats, which means the mass is limited to 4kg and the envelope dimensions are 10cm-by-10cm-by-34cm. The spacecraft can essentially be considered to come in two pieces: the solar panels and the spacecraft bus. A 3U-by-3U solar panel is attached to the +Y face of the spacecraft and a 3U-by-1U solar panel is attached to the –Y face of the spacecraft. The solar panels are assumed to have an areal density of 0.5 g/cm^2, yielding a total solar panel mass of 600g. Due to the center of mass requirements, it is assumed that mass balancing will be used to bring the spacecraft center of mass to the geometric center of the spacecraft bus. Because the internal configuration of the spacecraft is still unknown, the remainder of the mass will be assumed to be evenly distributed in the spacecraft bus. Thus, the inertia tensor taken at the spacecraft center of mass can be approximated as:

$$J = \begin{bmatrix} 0.03258 & 0 & 0 \\ 0 & 0.03383 & 0 \\ 0 & 0 & 0.01087 \end{bmatrix} kg * m^2 \tag{4.1}$$

The center of mass is assumed to be located at the geometric center.

$$\boldsymbol{x_{CM}} = \boldsymbol{x_{GC}} = [50 \quad 50 \quad 150]^T \, cm \tag{4.2}$$

4.2 ELECTROSPRAY THRUSTERS

The spacecraft derive translational and attitude actuation from the electrospray thruster slice.

4.2.1 Geometry

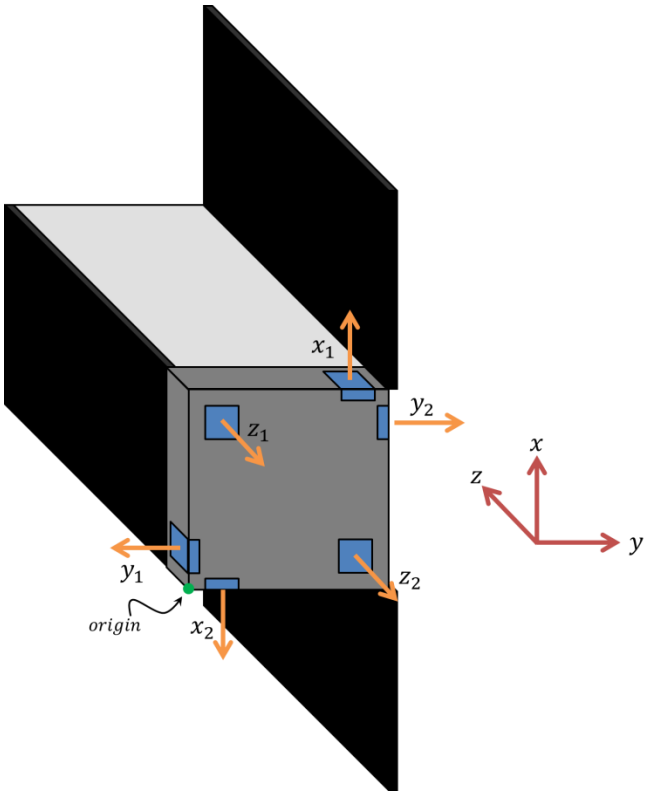

Figure 4.1 - Electrospray Thruster Configuration

The thrusters are assumed to produce a point force located at their center. The thrust vector is assumed to be normal to the thruster slice. These properties are tabulated in:

Table 4.1 - Thruster Geometry

Thruster	Index	Location [cm] ($\boldsymbol{l}$)	Thrust Direction ($\boldsymbol{d}$)
x_1	1	$[100 \quad 85 \quad 15]^T$	$[\ 1 \quad 0 \quad 0]^T$
x_2	2	$[\ 0 \quad 15 \quad 15]^T$	$[-1 \quad 0 \quad 0]^T$
y_1	3	$[15 \quad 0 \quad 15]^T$	$[0 \quad -1 \quad 0]^T$
y_2	4	$[85 \quad 100 \quad 15]^T$	$[0 \quad 1 \quad 0]^T$
z_1	5	$[85 \quad 15 \quad 0\ ]^T$	$[0 \quad 0 \quad -1]^T$
z_2	6	$[15 \quad 85 \quad 0\ ]^T$	$[0 \quad 0 \quad -1]^T$

It should be noted that the force produced by each thruster is in the opposite direction from its thrust vector. The moment arm for the i'th thruster is defined as:

$$\boldsymbol{\rho_i} = \boldsymbol{l_i} - \boldsymbol{x_{CM}} \tag{4.3}$$

Thus, the force and torque produced by the i'th thruster can be given by:

$$\boldsymbol{F_i} = -u_i \boldsymbol{d_i} \tag{4.4}$$

$$\boldsymbol{\tau_i} = \boldsymbol{\rho_i} \times -u_i \boldsymbol{d_i} \tag{4.5}$$

With u_i being the thrust produced by the i'th thruster.

4.2.2 Thrust Modeling

As mentioned in Section 1.2.1.2, one of the main benefits of the electrospray thruster technology is that it can be precisely controlled by varying the input voltage and duty cycles - much more finely than possible using a pulsed chemical system. For this reason, it is assumed that any thrust level up to the thruster saturation limit can be commanded. It is also (obviously) the case that the thrusters cannot produce negative thrust. Thus, the constraints on each thruster can be expressed as:

$$0 \leq u_i \leq u_{sat} \;\; \forall \, i \tag{4.6}$$

The electronics required for power conversion and control place some limitations on the thrust levels which can be produced. It is out of scope to discuss the details of the electronics design and why these limitations exist, but the result on the thruster is that there is a limit on the total thrust which can be produced by the system at any one time.

$$\sum_{i=1}^{6} u_i \leq u_{max} \tag{4.7}$$

For this hardware configuration, each thruster can produce a maximum of 50 μN and the total thrust output is limited to 100 μN. Due to the hardware configuration, this yields the following limiting values assuming maximum control effort and an ideal center of mass:

Table 4.2 - Body Axis Electrospray Thruster Force and Torque Limits

Body Axis	Thruster Force Limits [μN]	Thruster Torque Limits [μNm]
$X^{\mathcal{B}}$	-50 to +50	-8.5 to +8.5
$Y^{\mathcal{B}}$	-50 to +50	-8.5 to +8.5
$Z^{\mathcal{B}}$	0 to +100	-3.5 to 3.5

However, it should be noted that forces and torques about the $X^{\mathcal{B}}$ and $Y^{\mathcal{B}}$axes cannot be produced without generating disturbance forces and/or torques about the other axes. However, forces and torques about the $Z^{\mathcal{B}}$ axis can be produced independently and without introducing other disturbance torques.

4.3 TORQUE COILS

The spacecraft is also equipped with magnetic torque coils (abbreviated MTQ) for attitude actuation. Magnetic torque coils function via the Lorentz force, which is the force which acts on moving charges in an electric field. For a straight wire, the force can be expressed as:

$$\boldsymbol{F} = I\boldsymbol{l} \times \boldsymbol{B_m} \tag{4.8}$$

Where F is the force [N], I is the current in the wire [A], B_m is the magnetic field vector [T], and l is the wire vector [m], containing the magnitude and direction of the wire. If the wire is instead formed as a closed coil, the net force is zero but a torque is produced. Specifically:

$$\boldsymbol{\tau_{MTQ}} = N_{MTQ}A_{MTQ}\boldsymbol{I_{MTQ}} \times \boldsymbol{B_m} = \boldsymbol{\mu_{MTQ}} \times \boldsymbol{B_m} = -\boldsymbol{B_m} \times \boldsymbol{\mu_{MTQ}} \tag{4.9}$$

Where N_{MTQ} is the number of turns of wire, A_{MTQ} is the cross-sectional area of the coil [m^2], $\mathbf{I_{MTQ}}$ has a magnitude proportional to the current in a single wire and a direction which aligns with the normal vector of the coil. $\boldsymbol{\mu_{MTQ}}$ is known as the dipole moment of an MTQ, and units of Am^2.

There are three sets of torque coils – one mounted in line with each of the spacecraft body axes. The dipole moment can be varied simply be increasing or decreasing the current through the coils. Thus, the dipole moment can effectively be slewed to any direction at very high rate. Each coil is assumed to be separately controllable up to its maximum thrust. The maximum current in this case is determined by thermal concerns related to the wiring of the coils themselves. In order to prevent

overheating and possible damage, the current must be limited. Thus, each axis of the torque coil is limited as follow:

$$-\mu_{sat} \leq \mu_{MTQ_i} \leq \mu_{sat} \tag{4.10}$$

The preliminary mission study indicated that torque coils with a maximum dipole moment of 0.42 Am^2 would be sufficient to overcome the expected disturbances. Because the final design is not yet known, this is assumed to be the maximum dipole moment of the coil set around each axis. It should be noted that the torque coils must be unpowered when the magnetometer is taking data to prevent interference, so the actual coils will be sized such that their time-averaged maximum dipole moment includes this unpowered time.

This yields the following maximum torque values (assuming averaged magnetic field strength):

Table 4.3 - Body Axis Magnetic Torque Coil Torque Limits

Body Axis	Thruster Torque Limits [μNm]
$X^{\mathcal{B}}$	-18.9 to +18.9
$Y^{\mathcal{B}}$	-18.9 to +18.9
$Z^{\mathcal{B}}$	-18.9 to +18.9

The major downside to magnetic torque coils is that torque cannot be produced around the magnetic field vector, which means only two degrees of freedom are instantaneously controllable.

4.4 Rate Gyros

The spacecraft will have tri-axial digital gyroscopes for sensing the spacecraft's angular rate of its body frame with respect to the inertial frame. The sensor which is tentatively selected for this task is a component of the Analog Devices ADIS16488 [19].

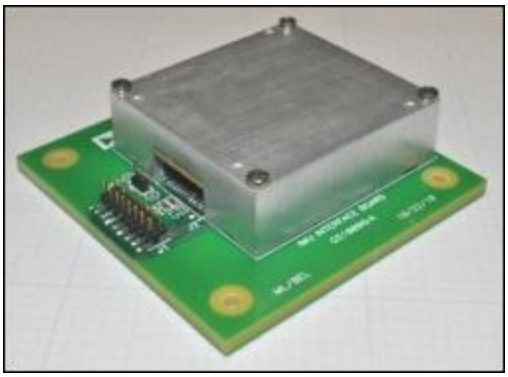

Figure 4.2 - ADIS16488 Inertial Measurement Unit

4.4.1 Sensor Model

Because this mission expects a relatively stable temperature profile and does not anticipate high acceleration, the parameters which are most likely to affect the spacecraft estimation are the initial bias error, the in-run bias stability, and the angular random walk. The gyros are modeled as a first order Markov process with the following model:

$$\bar{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta_v} \tag{4.11}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta_u} \tag{4.12}$$

$$\boldsymbol{\beta}(t_0) = \boldsymbol{\beta_0} \tag{4.13}$$

Where $\boldsymbol{\beta}$ is the gyro bias vector, $\boldsymbol{\beta_0}$ is the initial bias error, and $\boldsymbol{\eta_v}$, $\boldsymbol{\eta_u}$ are zero-mean Gaussian white-noise with spectral densities of $\sigma_v^2 I_{3x3}$ *and* $\sigma_u^2 I_{3x3}$, respectively.

4.4.2 Expected Use

The intended use of this sensor is to measure the spacecraft's angular rate for attitude determination. Upon ejection from the P-POD, it is likely that the spacecraft may be spinning at a rate of up to 3 degrees per second. However, in the nominal case, the spacecraft will be rotating at a rate of once per orbit, or 0.06 degrees per second.

4.4.3 Derivation of Numerical Values

Unfortunately, the sensor specifications are not given in terms of white noise. Because this gyro is a "smart" gyro, the device samples the gyro much more frequently than the user sample rate and

performs some internal filtering to give a more accurate sensor measurement at the lower sample rate requested by the user to reduce the noise in the measurement. This is characterized by an Angular Random Walk parameter with units of degrees per $\sqrt{\text{time}}$. This can be converted to white noise if the user's sampling time of the sensor is known by:

$$\sigma_v = \frac{ARW}{\sqrt{\Delta t}} = \frac{0.3°/\sqrt{hr}}{\sqrt{0.25\ sec}} = 0.01°/s \tag{4.14}$$

The white noise on the bias drift is computed from the in-run bias stability assuming that it can also be modeled as a random walk process and converted to a white noise process in a similar way. In this case, the sensor specification lists a standard deviation on the bias (units of degree/time). However, if it is assumed that this bound was derived by testing the performance for a finite time interval, it can be conservatively converted into an random walk process. Thus:

$$\sigma_u = \frac{in-run\ bias\ stability}{\sqrt{test\ time}} = 6.25\ °/hr \tag{4.15}$$

4.5 MAGNETOMETER

The ADIS16488 also has a tri-axial magnetometer which measures the magnetic field in the body frame.

4.5.1 Sensor Model

The sensor imperfection which will affect the system most is the output sensor noise. The magnetometer output is modeled as:

$$\overline{\boldsymbol{B}}_{\boldsymbol{m}}^{\mathcal{B}} = \boldsymbol{B}_{\boldsymbol{m}}^{\mathcal{B}} + \boldsymbol{\eta_m} \tag{4.16}$$

Where $\boldsymbol{\eta_m}$ is a zero mean Guassian white noise process with a spectral density of $\sigma_m^2 I_{3x3}$.

4.5.2 Expected Use

The intended use of this sensor is to measure the Earth's magnetic field for comparison with an onboard model for use in attitude determination. The Earth's magnetic field varies in intensity from approximately 30-60 μT. Because the magnetic field varies significantly over the orbit, the magnetic field may be oriented in any direction with respect to the body, even during nominal operations.

4.5.3 Derivation of Numerical Values

However, the output noise is specified as a noise density which can be converted to a white noise standard deviation as:

$$\sigma_m = \frac{Output\ Noise\ Density}{\sqrt{sampling\ time}} = \frac{0.054\ mgauss/\sqrt{Hz}}{\sqrt{0.25\ sec}} = 10.8\ nT \tag{4.17}$$

The amount of error introduced in this way is similar to the least significant bit, which is 10 nT, and represent an error of approximately 0.03% of the expected magnetic field.

4.6 SUN SENSORS

The sun sensors to be used on the spacecraft are the Space Micro Medium Sun Sensor [20]. These sun sensors are capable of measuring the direction to the sun in the body frame. However, the sensors can only determine two degrees of freedom, as the degree of freedom around the sun vector is not measurable. The spacecraft is expected to have two of these sensors, one centered on the $+Y^{\mathcal{B}}$ face and pointing along the $Y^{\mathcal{B}}$axis. The other will be centered on the $+Z^{\mathcal{B}}$ face and pointing along the $+Z^{\mathcal{B}}$ axis.

Figure 4.3 - Space Micro Coarse Sun Sensor

4.6.1 Sensor Model

Because the sun sensor only measures the direction of the sun, it can be modeled as a unit vector. Its error can be modeled by rotating the true sun vector. For small angles, this can be approximated as:

$$\overline{\boldsymbol{S}}_{\boldsymbol{v}}^{\mathcal{B}} = \begin{bmatrix} 1 & \eta_{z_{ss}} & \eta_{y_{ss}} \\ \eta_{z_{ss}} & 1 & \eta_{x_{ss}} \\ \eta_{y_{ss}} & \eta_{x_{ss}} & 1 \end{bmatrix} \boldsymbol{S}_{\boldsymbol{v}}^{\mathcal{B}} \tag{4.18}$$

Where $\boldsymbol{S_v^{\mathcal{B}}}$ is the unit vector pointing in the direction of the sun, expressed in the body frame. $\eta_{x_{ss}}$, $\eta_{y_{ss}}$, and $\eta_{z_{ss}}$ are independent zero-mean white noise processes with a variance of σ_{ss}^2.

In order for a valid measurement to be made, the sun must be in the field of view of the sensor. Most sensors have cone-shaped baffles, so a simple check for sensor validity is:

$$\boldsymbol{S_v^{\mathcal{B}}} \cdot \boldsymbol{d_{ss}} \geq cos(FOV) \tag{4.19}$$

Where $\boldsymbol{d_{ss}}$ is the direction vector of the sun sensor and FOV is the half-cone angle of the field of view of the sensor.

4.6.2 Expected Use

The intended use of the sun sensor is to measure the direction of the sun for comparison with an onboard model for use in attitude determination. In nominal operations, the sun sensors on the $+Y^{\mathcal{B}}$ face should have a continuous view of the sun. If the attitude is off-nominal, it is possible that none of the sun sensors will be able to see the sun.

4.6.3 Derivation of Numerical Values

The sensor's data sheet gives centroid error values for the entire field of view of the sensor. The majority of the field of view has an error less than 1°, so the sensor's angular standard deviation is conservatively assumed to have $\sigma_{ss} = 1°$.

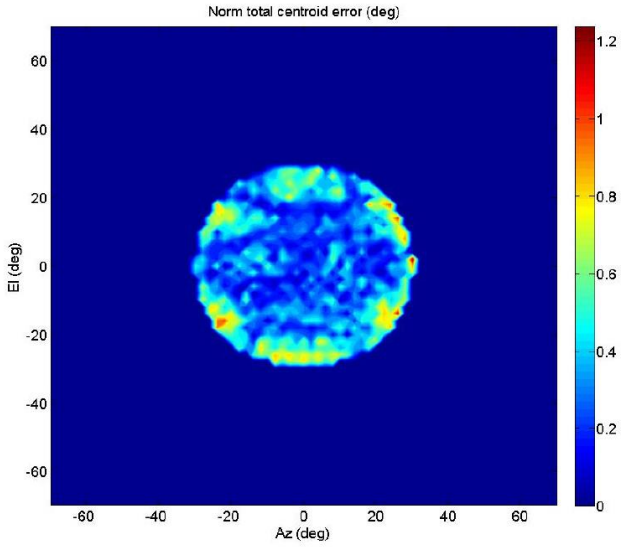

Figure 4.4- Sun Sensor Error Plot [20]

CHAPTER 5

FORMATION CONTROL DESIGN

5.1 ORBITAL STATE EQUATIONS

The starting point for the control design is the Hill-Clohessy-Wiltshire (HCW) equations for relative orbital motion [21]. These are a linearized set of equations for the motion of a member of the formation (a 'deputy') relative to an imaginary, uncontrolled object in a perfectly circular orbit (the 'chief'). The equations for a single satellite are given below (all quantities in LVLH coordinates):

$$\ddot{x} - 2\omega\dot{y} - 3\omega^2 x = u_x \tag{5.1}$$

$$\ddot{y} + 2\omega\dot{x} = u_y \tag{5.2}$$

$$\ddot{z} + \omega^2 z = u_z \tag{5.3}$$

Where u_x, u_y, and u_z are external accelerations in the LVLH x, y, and z directions, respectively. ω (not to be confused with the angular rate vector $\boldsymbol{\omega}$) is the angular rate of the chief's orbit, and can be calculated as:

$$\omega = \sqrt{\frac{\mu_E}{a^3}} \tag{5.4}$$

Where μ_E is the gravitational constant of the Earth, and a is the semi-major axis of the reference orbit. Rewriting these equations in matrix form yields:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{5.5}$$

Where:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix};\ B = \begin{bmatrix} 0_{3x3} \\ I_{3x3} \end{bmatrix};\ \boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix};\ \boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{5.6}$$

However, because the forces are produced by thrusters fixed to the spacecraft body, it is more useful to describe the control vector in terms of the body axes, since the body z-axis thruster is the only one which can fire without creating disturbance torques. This is important to distinguish so that this control can be weighted later. This changes the equations to:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B^{\mathcal{B}}\boldsymbol{u}^{\boldsymbol{\mathcal{B}}} \tag{5.7}$$

$$B^{\mathcal{B}} = \begin{bmatrix} 0_{3x3} \\ R^{\mathcal{LB}} \end{bmatrix};\ \boldsymbol{u}^{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} u_x^{\mathcal{B}} \\ u_y^{\mathcal{B}} \\ u_z^{\mathcal{B}} \end{bmatrix} \tag{5.8}$$

For expressing the full state of the system, the states of each satellite are simply concatenated to form another equation of the form:

$$\dot{\boldsymbol{X}} = \bar{A}\boldsymbol{X} + \bar{B}^{\mathcal{B}}\boldsymbol{U}^{\mathcal{B}} \tag{5.9}$$

$$\bar{A} = \begin{bmatrix} A_1 & 0_{6x6} & 0_{6x6} \\ 0_{6x6} & A_2 & 0_{6x6} \\ 0_{6x6} & 0_{6x6} & A_3 \end{bmatrix};\ \bar{B}^{\mathcal{B}} = \begin{bmatrix} B_1^{\mathcal{B}} & 0_{6x3} & 0_{6x3} \\ 0_{6x3} & B_2^{\mathcal{B}} & 0_{6x3} \\ 0_{6x3} & 0_{6x3} & B_3^{\mathcal{B}} \end{bmatrix};\ \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \end{bmatrix};\ \boldsymbol{U}^{\mathcal{B}} = \begin{bmatrix} \boldsymbol{u_1^{\mathcal{B}}} \\ \boldsymbol{u_2^{\mathcal{B}}} \\ \boldsymbol{u_3^{\mathcal{B}}} \end{bmatrix} \tag{5.10}$$

Because all the satellites are referenced to the same circular orbit, $A_1 = A_2 = A_3 \triangleq A$.

In the HCW formulation, it is possible to achieve Passive Relative Orbits (PROs) if certain relationships are held between states. In these orbits, the satellites will oscillate periodically about a center, which will not drift over time. These types of orbits are the foundation for fuel-efficient formation flight. In general, randomly selected conditions will not yield PROs. The condition which yields a PRO with a common center is the following (note that string of pearls does NOT obey this):

$$\dot{x} = \frac{1}{2}\omega y \tag{5.11}$$

The characteristic shape shared by all PRO's is a 1:2 ellipse projected in the x-y plane. Because the z-direction is decoupled in the HCW formulation, the normal-direction motion of the satellite is not constrained by the PRO condition.

5.2 Relative State Equations

Because the electrospray thrusters cannot create thrust in the –z body direction, it is not possible to directly use continuous control methods and directly command all three spacecraft relative to the same fixed orbit. However, because the primary concern for mission success is relative positioning, the center of the cluster is actually not important. Therefore, the system state can be defined relative to one of the actual spacecraft instead of to an imaginary, uncontrolled spacecraft. Thus, the chief is now a controllable spacecraft instead of an imaginary point. This allows the size of system positional state vector to be reduced by six. The relative state vector is therefore defined as the state of spacecraft 2 and 3 relative to spacecraft 1. Thus the relative states and controls can be defined as:

$$\boldsymbol{x_{21} = x_2 - x_1;} \quad \boldsymbol{x_{31} = x_3 - x_1;} \quad \boldsymbol{u_{21} = u_2 - u_1;} \quad \boldsymbol{u_{31} = u_3 - u_1} \tag{5.12}$$

These states have dynamics

$$\boldsymbol{\dot{X}_{rel}} = A_{rel}\boldsymbol{X_{rel}} + B_{rel}\boldsymbol{U_{rel}} \tag{5.13}$$

$$A_{rel} = \begin{bmatrix} A & 0_{6x6} \\ 0_{6x6} & A \end{bmatrix};\ B_{rel} = \begin{bmatrix} B & 0_{3x3} \\ 0_{3x3} & B \end{bmatrix};\ \boldsymbol{X_{rel}} = \begin{bmatrix} \boldsymbol{x_{21}} \\ \boldsymbol{x_{31}} \end{bmatrix};\ \boldsymbol{U_{rel}} = \begin{bmatrix} \boldsymbol{u_{21}} \\ \boldsymbol{u_{31}} \end{bmatrix} \tag{5.14}$$

5.3 Cluster Control Law

5.3.1 Relative Acceleration Command Generation

Because the system uses differential GPS, it will have relative position and velocity known to high accuracy. Therefore, it is appropriate to assume the full relative state is available and to use linear full-state feedback techniques for the control design.

Having created the relative state matrices as described in the previous section, the controller design is quite straightforward. The controller used is an infinite-horizon continuous linear-quadratic regulator (LQR). This controller minimizes the following error:

$$J_{LQR} = \int_0^\infty \boldsymbol{X_{rel}^T} Q_{LQR} \boldsymbol{X_{rel}} + \boldsymbol{U_{rel}^T} R_{LQR} \boldsymbol{U_{rel}}\, dt \tag{5.15}$$

The feedback control law which minimizes this cost is given by:

$$\boldsymbol{U_{rel,\ desired}} = -K_{LQR} \boldsymbol{X_{rel}} \tag{5.16}$$

$$K_{LQR} = R_{LQR}^{-1} \boldsymbol{B_{rel}^T} P_{LQR} \tag{5.17}$$

P_{LQR} is found by solving the continuous time algebraic Ricatti equation:

$$A_{rel}^T P_{LQR} + P_{LQR} A_{rel} - P_{QRL} \boldsymbol{B_{rel}} R_{LQR}^{-1} \boldsymbol{B_{rel}^T} P_{LQR} + Q_{LQR} = 0 \tag{5.18}$$

This can be generated using the standard MATLAB function '*lqr*' with the following simple syntax:

$$K_{LQR} = lqr(A_{rel}, B_{rel}, Q_{LQR}, R_{LQR}) \tag{5.19}$$

Q_{LQR} *and* R_{LQR} are the weighting matrices on state error and control effort, respectively. Because the system dynamics couple the position and velocity through the characteristic orbital rate, small velocity errors will eventually propagate into much larger position errors. Therefore, the state weighting matrix is chosen to more heavily penalize errors in velocity. Thus:

$$Q_1 = \begin{bmatrix} I_{3x3} & 0_{3x3} \\ 0_{3x3} & \frac{1}{\omega^2} I_{3x3} \end{bmatrix};\ Q_{LQR} = \begin{bmatrix} Q_1 & 0_{6x6} \\ 0_{6x6} & Q_1 \end{bmatrix} \tag{5.20}$$

The final tuning parameter is the control weighting matrix, R_{LQR}. Because the desired control action is expressed as a relative acceleration in the LVLH frame (independent of the orientation of the spacecraft bodies), it is not desirable to penalize any axis of control more heavily. Thus, the control weighting matrix is:

$$R_1 = I_{3x3};\ R_{LQR} = \rho \begin{bmatrix} R_1 & 0_{3x3} \\ 0_{3x3} & R_1 \end{bmatrix} \tag{5.21}$$

Thus, the tuning of the controller can be largely accomplished by varying the scalar value of ρ. It is always a good idea in beginning a control design to relate the parameters of the controller to parameters of the system. In this case, it was decided that a position error of 1km should yield

maximum control effort in the axial thruster. This yields a value of ρ =10^{14}. Subsequent tuning reveals that ρ values in the range of 10^{14} to 10^{16} are typically appropriate for most maneuvers.

To prevent saturation effects from causing drastically unwanted behavior, the entire relative acceleration vector is scaled down if the magnitude of any of its components exceed the maximum acceleration producible by the thrusters:

$$\boldsymbol{U_{rel,\ desired}} = \frac{\boldsymbol{U_{rel,\ desired}}}{\left\|\boldsymbol{U_{rel,\ desired}}\right\|} * min\ \{u_{sat}, \left\|\boldsymbol{U_{rel,\ desired}}\right\|\} \qquad (5.22)$$

5.3.2 Addition of a Reference Trajectory

Because full-state feedback is used, constant reference commands can be easily added:

$$\boldsymbol{U_{rel}} = K_{LQR}(\boldsymbol{r} - \boldsymbol{X_{rel}}) \qquad (5.23)$$

The most common formation examined is the 'string of pearls' which is a constant separation in the velocity direction. The design reference trajectory for this mission is the 10km string of pearls configuration. This is represented as:

$$\boldsymbol{r_{SOP}} = \begin{bmatrix} 0 \\ 10 * 10^3 \\ 0_{5x1} \\ -10 * 10^3 \\ 0_{4x1} \end{bmatrix} \qquad (5.24)$$

5.3.3 Allocation of Control to Individual Spacecraft

This control law generates a set of desired relative accelerations$\boldsymbol{U_{rel,\ desired}}$, expressed in the LVLH frame of the chief. Because communication is assumed to be infrequent, it is desirable for centralized command generation to be infrequent. For the design reference mission, the spacecraft should all be nominally oriented and would therefore have a constant orientation in the LVLH frame. However, in the interest of having a more robust control design which can accommodate attitude control modes such as inertial pointing or ground station tracking, or continue to operate in case of attitude actuator failure, a variable LVLH attitude is considered.

In order to produce the control desired by the LQR controller, all three spacecraft must fire their thrusters such that the net relative acceleration is equal to the desired value. To accomplish this,

control is allocated to each satellite as a desired LVLH force which must be maintained until the next centralized control update. Each satellite must then calculate its own thruster firings based on its instantaneous attitude in order to produce that LVLH force (or as close as possible) at every control cycle.

However, because of the limitation that the spacecraft cannot produce any thrust in the $-Z^{\mathcal{B}}$ direction, the control allocator must consider the spacecraft's attitude to some extent in order to ensure that the allocated control is feasible based on its current attitude. Therefore, the assumption is made that the LVLH attitude is nearly constant between centralized control cycles (though possibly varying over longer timescales).

The method used for control allocation is constrained linear least squares optimization. The problem statement for this type of problem is:

$$\min_{U_L} \|C_L \boldsymbol{U_L} - \boldsymbol{d_L}\|^2 ; \quad subject\ to: \quad A_L \boldsymbol{U_L} < \boldsymbol{b_L}\ and\ \boldsymbol{L_L} \leq \boldsymbol{U_L} \leq \boldsymbol{H_L} \tag{5.25}$$

In this case, the optimization variables are the actuator forces:

$$\boldsymbol{U_L} = m * \boldsymbol{U}^{\mathcal{B}} = m * \begin{bmatrix} \boldsymbol{u_1^{\mathcal{B}}} \\ \boldsymbol{u_2^{\mathcal{B}}} \\ \boldsymbol{u_3^{\mathcal{B}}} \end{bmatrix} \tag{5.26}$$

From the previously discussed thruster models, the solution space for the thrust commands can be obtained:

$$\begin{gathered} \boldsymbol{L_L} = u_{sat} * [-1 \quad -1 \quad 0 \quad -1 \quad -1 \quad 0 \quad -1 \quad -1 \quad 0]^T \\ \boldsymbol{H_L} = u_{sat} * [1 \quad 1 \quad 2 \quad 1 \quad 1 \quad 2 \quad 1 \quad 1 \quad 2]^T \end{gathered} \tag{5.27}$$

The total thrust constraint imposed by the thrust slice electronics is expressed via linear system inequalities ($A_L \boldsymbol{U_L} < \boldsymbol{b_L}$). Although each thruster can only fire in the positive direction, the spacecraft can have net body $X^{\mathcal{B}}$ and $Y^{\mathcal{B}}$ thrusts as positive or negative. Thus, the total thrust constraint can only be expressed in a linear system by considering all cases of positive and negative $X^{\mathcal{B}}$ and $Y^{\mathcal{B}}$. $Z^{\mathcal{B}}$ thrust can only be in the positive direction. Each spacecraft is therefore constrained by:

$$A_{L_1} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix}; \quad \boldsymbol{b_{L_1}} = u_{max} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{5.28}$$

Thus, the constraint on the overall system is given by:

$$A_L = \begin{bmatrix} A_{L_1} & 0_{4x3} & 0_{4x3} \\ 0_{4x3} & A_{L_1} & 0_{4x3} \\ 0_{4x3} & 0_{4x3} & A_{L_1} \end{bmatrix}; \quad \boldsymbol{b_L} = \begin{bmatrix} \boldsymbol{b_{L_1}} \\ \boldsymbol{b_{L_1}} \\ \boldsymbol{b_{L_1}} \end{bmatrix} \tag{5.29}$$

The actuator models from the previous sections can be combined into a linear system as:

$$\boldsymbol{U_{rel}^{\mathcal{L}}} = B_{rel}^{\mathcal{B}} \boldsymbol{U^{\mathcal{B}}} = \begin{bmatrix} -R_1^{\mathcal{LB}} & R_2^{\mathcal{LB}} & 0_{3x3} \\ -R_1^{\mathcal{LB}} & 0_{3x3} & R_3^{\mathcal{LB}} \end{bmatrix} \boldsymbol{U^{\mathcal{B}}} \tag{5.30}$$

Because the objective is to ensure a target relative acceleration, it could easily be the case that a solution which exactly produces the required relative acceleration has two spacecraft producing thrusts which cancel each other out, thus wasting fuel. In order to minimize the total fuel use, this system is additionally augmented with a set of equations to penalize control effort. Thus:

$$C_L = \begin{bmatrix} -R_1^{\mathcal{LB}} & R_2^{\mathcal{LB}} & 0_{3x3} \\ -R_1^{\mathcal{LB}} & 0_{3x3} & R_3^{\mathcal{LB}} \\ I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & I_{3x3} \end{bmatrix}; \quad \boldsymbol{d_L} = \begin{bmatrix} m * \boldsymbol{U_{rel,desired}^{\mathcal{L}}} \\ 0_{9x1} \end{bmatrix} \tag{5.31}$$

In order to ensure that this fuel-minimization takes second priority to actually producing the desired relative accelerations, a weighting matrix comprised of only positive diagonal elements is applied to the optimization:

$$\min_{U_L} \| W_L C_L \boldsymbol{U_L} - W_L \boldsymbol{d_L} \|^2 \tag{5.32}$$

With:

$$W_L = \frac{1}{u_{max}} \begin{bmatrix} I_{6x6} & 0_{6x9} \\ 0_{6x9} & \rho_L * I_{9x9} \end{bmatrix} \tag{5.33}$$

This problem is solved using an active-set quadratic programming algorithm (MATLAB *lsqlin* function or using the method from [22]), yielding the optimized $\boldsymbol{U_L}$. ρ_L=10^{-3} was used. Recall that

this is a set of body forces. In order for formation control to be achieved, the relative accelerations must be those specified by the LQR control law. In order to maintain these relative accelerations with a variable attitude between centralized control cycles, the body forces are rotated back to LVLH forces which must then be regulated by the individual spacecraft as its attitude changes. Thus, the output of the cluster controller to each spacecraft is a force specified in the LVLH frame it must maintain continuously for the duration of the central control cycle. The commanded force on the i'th spacecraft can then be obtained from the least-squares algorithm solution as:

$$\boldsymbol{F}_{\boldsymbol{i,cmd}}^{\mathcal{L}} = m * R_i^{\mathcal{LB}} \boldsymbol{u}_{\boldsymbol{i}}^{\mathcal{B}} \tag{5.34}$$

Using this formulation of control allocation is appropriate because it explicitly accounts for the thruster saturations and will therefore only produce feasible solutions. However, the obvious downside of this method is that the control commands may not exactly produce the desired relative accelerations. An additional downside is that the solving the constrained linear least squares problem requires software libraries which can solve quadratic programs. This may be too computationally intensive for some formation flight missions.

5.4 CURVILINEAR MODIFICATION

5.4.1 Coordinate System

One of the major issues with the application of this control methodology is the conversion of the spacecraft's actual state to the relative frame where the linearized equations hold. The HCW equations and formulation yield a rotating Cartesian coordinate system. Because the circular orbit is being approximated by a rectilinear coordinate system, the error introduced by the linear model increases with the distance from the chief. Note that in this section the '0' subscript refers to the chief orbit and the 'i' subscript refers to the i'th spacecraft. The chief, in this case, is one of the spacecraft.

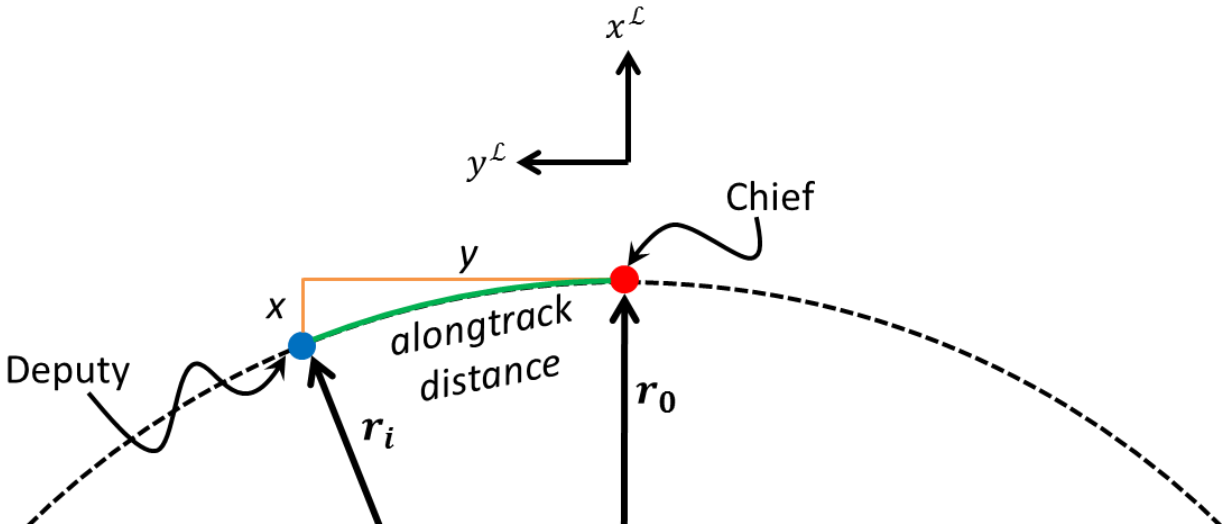

Figure 5.1 - Error in Cartesian LVLH Frame (exaggerated)

For a spacecraft in a true string of pearls formation, this error has the effect of underestimating the alongtrack distance (an arclength) and introducing an error in the radial position. A similar image can be constructed for the out of plane error. This is because the string of pearls aims to emulate a differential true anomaly, which is an angular separation along a constant radius.

These types of errors can be reduced by switching to a curvilinear LVLH coordinate system, directly incorporating the fundamentally circular nature of orbits. The rotating system is established using two sequential rotations:

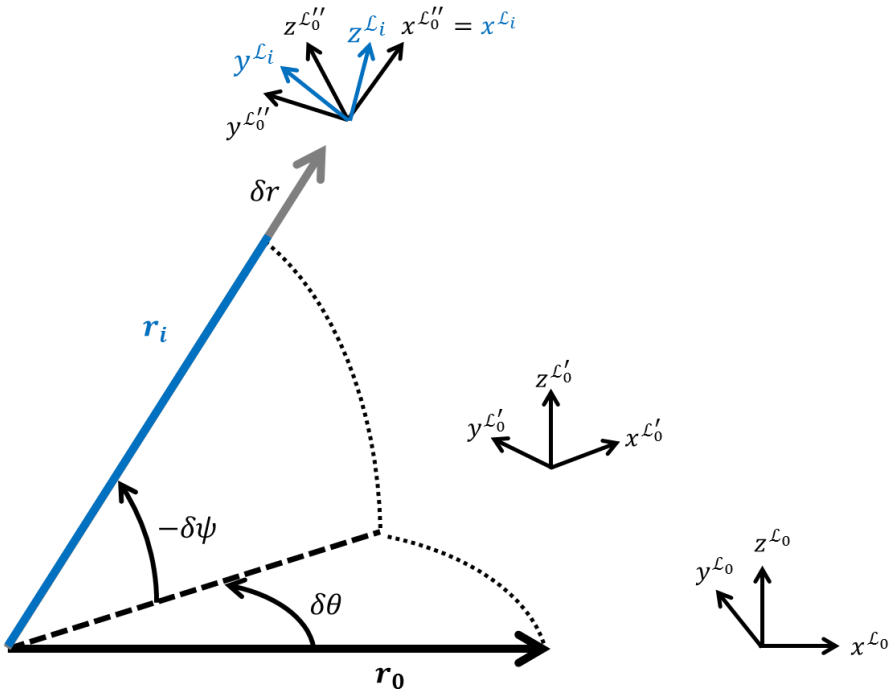

Figure 5.2 - Curvilinear Coordinate Diagram

The first rotation is by an angle $\delta\theta$ counterclockwise about the $\hat{\boldsymbol{e}}_N^{\mathcal{L}_0}$ axis, which is the orbit normal. This creates the intermediate $\mathcal{L}_0'$ frame. The second rotation is by an angle $-\delta\psi$ about the $\hat{e}_T^{\mathcal{L}_0'}$ axis which yields the $\mathcal{L}_0''$ coordinate system illustrated in Figure 5.2. This coordinate system represents the curvilinear LVLH frame of the chief evaluated at the position of the deputy. The coordinate directions are given by the following equations:

$$\hat{\boldsymbol{e}}_R^{\mathcal{L}_0''} = \cos\delta\psi\ \cos\delta\theta\ \hat{\boldsymbol{e}}_R^{\mathcal{L}_0} + \cos\delta\psi\sin\delta\theta\ \hat{\boldsymbol{e}}_T^{\mathcal{L}_0} - \sin\delta\psi\ \hat{\boldsymbol{e}}_N^{\mathcal{L}_0} \tag{5.35}$$

$$\hat{\boldsymbol{e}}_T^{\mathcal{L}_0''} = -\sin\delta\theta\ \hat{\boldsymbol{e}}_R^{\mathcal{L}_0} + \cos\delta\theta\ \hat{\boldsymbol{e}}_T^{\mathcal{L}_0} \tag{5.36}$$

$$\hat{\boldsymbol{e}}_N^{\mathcal{L}_0''} = \sin\delta\psi\cos\delta\theta\ \hat{\boldsymbol{e}}_R^{\mathcal{L}_0} + \sin\delta\psi\cos\delta\theta\ \hat{\boldsymbol{e}}_T^{\mathcal{L}_0} + \cos\delta\psi\ \hat{\boldsymbol{e}}_N^{\mathcal{L}_0} \tag{5.37}$$

This can be more compactly represented by the rotation matrix $R^{\mathcal{L}_0''\mathcal{L}_0}$ which transforms the Cartesian coordinates expressed in the basis of $\mathcal{L}_0$ evaluated at the location of the chief to the basis of $\mathcal{L}_0$ evaluated at the location of the deputy:

$$R^{\mathcal{L}_0''\mathcal{L}_0} = \begin{bmatrix} \cos\delta\psi\cos\delta\theta & \cos\delta\psi\sin\delta\theta & -\sin\delta\psi \\ -\sin\delta\theta & \cos\delta\theta & 0 \\ \cos\delta\theta & \sin\delta\psi\sin\delta\theta & \cos\delta\psi \end{bmatrix} \tag{5.38}$$

The position vector can then be expressed as:

$$\boldsymbol{r_i} = r_i \hat{\boldsymbol{e}}_R^{\mathcal{L}_0''} = (r_0 + \delta r) \begin{bmatrix} \cos\delta\psi \ \cos\delta\theta \\ \cos\delta\psi \sin\delta\theta \\ -\sin\delta\psi \end{bmatrix} \tag{5.39}$$

The other frame illustrated is the LVLH frame of the deputy itself, which is denoted as $\mathcal{L}_i$. Because the radial vector of the deputy's LVLH frame is defined by the direction of its inertial position vector:

$$\hat{\boldsymbol{e}}_R^{\mathcal{L}_0''} = \hat{\boldsymbol{e}}_R^{\mathcal{L}_i} \tag{5.40}$$

However, the other two axes are not necessarily aligned.

5.4.2 Dynamics

The dynamics can be obtained by taking the time derivative of the deputy position vector in the curvilinear LVLH coordinates of the chief (which are time varying according to the motion of the chief) and equating that to the force of gravity plus the control accelerations. These equations are then linearized to obtain a new system. This derivation is performed in [23] for a slightly different coordinate system. However, the results are easily extended to this formulation with the following result:

$$\delta\ddot{r} - 2a_0\omega\delta\dot{\theta} - 3\omega^2\delta r = u_R \tag{5.41}$$

$$a_0\delta\ddot{\theta} + 2\omega\delta\dot{r} = u_T \tag{5.42}$$

$$-a_0\delta\ddot{\psi} - a_0\omega^2\delta\psi = u_N \tag{5.43}$$

This system can be expressed as a standard linear system as follows:

$$\boldsymbol{x_c} = A_c\boldsymbol{x_c} + B_c\boldsymbol{u_c} \tag{5.44}$$

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix};\ B_c = \begin{bmatrix} 0_{3x3} \\ I_{3x3} \end{bmatrix};\ \boldsymbol{x_c} = \begin{bmatrix} \delta r \\ a_0\delta\theta \\ -a_0\delta\psi \\ \delta\dot{r} \\ a_0\delta\dot{\theta} \\ -a_0\delta\dot{\psi} \end{bmatrix};\ \boldsymbol{u_c} = \begin{bmatrix} u_R \\ u_T \\ u_N \end{bmatrix} \tag{5.45}$$

This yields the exciting result that this system is exactly the same as the HCW system previously used for control design! This means that curvilinear coordinates can be used without redoing the control design. The only change necessary is to redefine the transform from the global system state (ECI frame) to the LVLH frame using the new curvilinear relations. This has been implemented throughout the control design.

5.4.3 Converting from ECI to Curvilinear LVLH

Because this conversion only affects the cluster control calculation, it is reasonable to assume central knowledge of the all spacecraft ECI states (which is required for the cluster control calculations anyway). Note that this algorithm DOES allow for the chief to be in a non-circular orbit in order to obtain more accurate differential values. Note that this is by far the more important operation, because this is the direction the state representation takes at every control cycle in order to be controlled. The problem statement for this operation is as follows:

$$Given\ \boldsymbol{r_0^I}, \boldsymbol{v_0^I}, \boldsymbol{r_i^I}, \boldsymbol{v_i^I};\quad Find\ \delta r, \delta\theta, \delta\psi, \delta\dot{r}, \delta\dot{\theta}, \delta\dot{\psi} \tag{5.46}$$

The differential radius is trivially calculated as:

$$\delta r = r_i - r_0 \tag{5.47}$$

First, the position vector must be expressed in the chief's Cartesian coordinates. The transformation matrix $R^{\mathcal{L}_0 I}$ is obtained in the usual way for the Cartesian LVLH frame about the chief.

$$\boldsymbol{r_i^{\mathcal{L}_0}} = R^{\mathcal{L}_0 I}\boldsymbol{r_i^I} \tag{5.48}$$

The differential out-of-plane angle can then be calculated as:

$$\delta\psi = sin^{-1}\left([0 \quad 0 \quad -1] * \frac{\boldsymbol{r_i^{\mathcal{L}_0}}}{r_i}\right) \tag{5.49}$$

The alongtrack angle can be computed in similar fashion:

$$\delta\theta = sin^{-1}\left([0 \quad 1 \quad 0] * \frac{\boldsymbol{r_i^{\mathcal{L}_0}}}{r_i\, cos\, \delta\psi}\right) \tag{5.50}$$

The velocity is slightly more complicated due to the conversion from inertial to rotating reference frame. First, the inertial components must be expressed in the chief's frame evaluated at the deputy's position. This is accomplished by the following transformation:

$$\breve{\boldsymbol{v}}_{\boldsymbol{i}} = R^{\mathcal{L}_0''\mathcal{L}_0} R^{\mathcal{L}_0 I} \boldsymbol{v}_{\boldsymbol{i}}^{\boldsymbol{I}} \tag{5.51}$$

The unusual notation is because this is an inertial vector expressed in the coordinates of a rotating frame, which would be very confusing to express using the usual notation. The same transform is applied to the ECI velocity of the chief, but evaluating the coordinate frame at the origin:

$$\breve{\boldsymbol{v}}_{\boldsymbol{0}} = R^{\mathcal{L}_0 I} \boldsymbol{v}_{\boldsymbol{i}}^{\boldsymbol{I}} \tag{5.52}$$

The differential radial component is trivially obtained by:

$$\dot{r}_i = [1 \quad 0 \quad 0] * \breve{\boldsymbol{v}}_{\boldsymbol{i}} \tag{5.53}$$

$$\dot{r}_0 = [1 \quad 0 \quad 0] * \breve{\boldsymbol{v}}_{\boldsymbol{0}} \tag{5.54}$$

$$\delta\dot{r} = \dot{r}_i - \dot{r}_0 \tag{5.55}$$

The differential normal velocity can be quickly obtained because the normal velocity of the chief in its own LVLH frame is zero by definition. Therefore:

$$\delta\dot{\psi} = [0 \quad 0 \quad -1] * \frac{\breve{\boldsymbol{v}}_{\boldsymbol{i}}}{r_i} \tag{5.56}$$

The differential tangential velocity can be similarly expressed as:

$$\dot{\theta}_i = [0 \quad 1 \quad 0] * \frac{\breve{\boldsymbol{v}}_{\boldsymbol{i}}}{r_i \cos \delta\psi} \tag{5.57}$$

$$\dot{\theta}_0 = [0 \quad 1 \quad 0] * \frac{\breve{\boldsymbol{v}}_{\boldsymbol{i}}}{r_0} \tag{5.58}$$

$$\delta\dot{\theta} = \dot{\theta}_i - \dot{\theta}_0 \tag{5.59}$$

Note that because $\dot{\theta}_0 \approx \omega$, this step is where the primary rotation of the frame is removed.

5.4.4 Converting from Curvilinear LVLH to ECI

In order to convert from curvilinear LVLH coordinates to ECI, a slightly different process must be used because of the information available. The problem statement for this operation is:

$$Given\ \boldsymbol{r_0^I}, \boldsymbol{v_0^I}, \delta r, \delta\theta, \delta\psi, \delta\dot{r}, \delta\dot{\theta}, \delta\dot{\psi}\ ; \quad Find\ \boldsymbol{r_i^I}, \boldsymbol{v_i^I} \tag{5.60}$$

The first step is to compute the radius:

$$r_i = r_0 + \delta r \tag{5.61}$$

The position expressed in the LVLH frame can then be easily computed as:

$$\boldsymbol{r_i^{\mathcal{L}_0}} = r_i \begin{bmatrix} cos\,\delta\psi\ cos\,\delta\theta \\ cos\,\delta\psi\ sin\,\delta\theta \\ -sin\,\delta\psi \end{bmatrix} \tag{5.62}$$

The inertial position is easily computed as:

$$\boldsymbol{r_i^I} = R^{I\mathcal{L}_0}\boldsymbol{r_i^{\mathcal{L}_0}} \tag{5.63}$$

The inertial velocity expressed in the $\mathcal{L}_0''$ frame can be computed as:

$$\breve{\boldsymbol{v}}_{\boldsymbol{i}} = \begin{bmatrix} \delta\dot{r} + \dot{r}_0 \\ r_1 \cos\delta\psi\left(\delta\dot{\theta} + \dot{\theta}_0\right) \\ -r_1\delta\dot{\psi} \end{bmatrix} \tag{5.64}$$

$\dot{r}_0$ and $\dot{\theta}_0$are computed as in the previous section. Finally, the velocity must be rotated into the inertial frame.

$$\boldsymbol{v_i^I} = R^{I\mathcal{L}_0}R^{\mathcal{L}_0\mathcal{L}_0''}\breve{\boldsymbol{v}}_{\boldsymbol{i}} \tag{5.65}$$

$R^{I\mathcal{L}_0}$ and $R^{\mathcal{L}_0\mathcal{L}_0''}$ are also computed as given in the previous section.

CHAPTER 6

ATTITUDE CONTROL DESIGN

6.1 ATTITUDE REPRESENTATION

The attitude representation of the system which is used in the truth model and onboard is the transformation from the inertial frame to the body frame. This is represented by the DCM $R^{\mathcal{BJ}}$ or the quaternion $q^{\mathcal{BJ}}$. Because these quantities come up a lot, this superscript is assumed if no other is given.

The angular rate which is tracked and used for control and estimation is the rate of change of the body frame with respect to the inertial frame, expressed in body frame coordinates, which is represented symbolically as $\boldsymbol{\omega}^{\mathcal{JB},\mathcal{B}}$. This is also assumed in the absence of other coordinate frame notation on any angular rate terms.

6.2 DETUMBLE CONTROL LAW

Because it is possible for the spacecraft to be tumbling too quickly to obtain good GPS or sun sensor data, it is desirable to have a control mode which functions in the absence of these sensors to reduce angular velocity until the sensors can operate. One such control mode is known as 'B-dot' control, which only requires a magnetometer.

This control mode essentially assumes that the rate of change in the magnetic field is due solely to the angular motion of the spacecraft. While this is not strictly true (because the spacecraft's translation also causes a change in the magnetic field), the approximation is valid for high spacecraft angular rates. Thus,

$$\dot{\boldsymbol{B}} \approx -\boldsymbol{\omega} \times \boldsymbol{B} \tag{6.1}$$

Assuming a control law of:

$$\boldsymbol{\mu} = -\frac{K_{Bdot}}{\boldsymbol{B}^T\boldsymbol{B}}\dot{\boldsymbol{B}} \tag{6.2}$$

Yields the following:

$$\boldsymbol{\tau_{Bdot}} = \boldsymbol{\mu} \times \boldsymbol{B} \approx \frac{K_{Bdot}}{\boldsymbol{B}^T\boldsymbol{B}}(\boldsymbol{\omega} \times \boldsymbol{B}) \times \boldsymbol{B} \tag{6.3}$$

Finally,

$$\boldsymbol{\tau_{Bdot}} \approx -K_{Bdot}\boldsymbol{\omega} + K_{Bdot}\frac{\boldsymbol{\omega}^T\boldsymbol{B}}{\boldsymbol{B}^T\boldsymbol{B}}\boldsymbol{B} \tag{6.4}$$

The first term looks like a standard angular rate damper and the second term is the projection of the angular rate vector onto the magnetic field vector. The second term exists because it is impossible to create torque along the magnetic field vector using torque coils.

6.3 Attitude Stabilize Control Law

The attitude control law is used to obtain 3-axis stabilization and is based on one presented in [17]. It is essentially a PD control law based on quaternion error and angular rate error. The control law is:

$$\boldsymbol{\tau_{desired}} = 2\boldsymbol{K_q^T}\begin{bmatrix} q_{1E} \\ q_{2E} \\ q_{3E} \end{bmatrix} q_{4E} + \boldsymbol{K_\omega^T}(\boldsymbol{\omega_{desired}} - \boldsymbol{\omega}) \tag{6.5}$$

Where $\boldsymbol{K_q}$ *and* $\boldsymbol{K_\omega}$ are gain vectors, $\boldsymbol{\omega_{desired}}$ is the desired angular rate, and $\boldsymbol{q_E}$ is defined as:

$$\boldsymbol{q_E} = \boldsymbol{q_{desired}} \otimes \boldsymbol{q^{-1}} \tag{6.6}$$

To stabilize in the LVLH frame, the attitude used should be $\boldsymbol{q^{\mathcal{BL}}}$**.** Because the goal is to stabilize the spacecraft relative to the LVLH frame, the desired angular velocity of the spacecraft should be the rotation of the LVLH orbital frame. Note the subtle difference between the spacecraft angular velocity vector ($\boldsymbol{\omega}$) and the orbital rate (ω). Thus,

$$\boldsymbol{\omega_{desired}^{I\mathcal{B},\mathcal{L}}} = \boldsymbol{\omega^{I\mathcal{L},\mathcal{L}}} = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \tag{6.7}$$

Therefore,

$$\boldsymbol{\omega}_{\boldsymbol{desired}}^{\mathcal{IB},\mathcal{B}} \triangleq \boldsymbol{\omega}_{\boldsymbol{desired}} = R^{\mathcal{BL}}\boldsymbol{\omega}^{\mathcal{IL},\mathcal{L}} = R^{\mathcal{BL}}\begin{bmatrix}0\\0\\\omega\end{bmatrix} \tag{6.8}$$

Although it is not anticipated for use in this mission, the same control law can be adapted for inertial pointing by using $\boldsymbol{q}^{\mathcal{BI}}$ and zero rates about every axis.

The control gains which have been found to work best are K_q=3*10^{-5} and Kω = 3*10^{-3}. The max control torque which can be provided by the torque coils is on the order of 10^{-5} Nm, so these gains correspond to actuator saturation around 20° of attitude error or 0.2°/s of attitude rate error.

CHAPTER 7

ACTUATOR COMMAND GENERATION

Having generated a set of desired forces and torques from the cluster control law and the attitude control law, it is now necessary to compute the set of actuator commands which will be issued from the computer in order to effect the desired motion in each satellite. This operation is called the 'local allocator' because it occurs in each satellite independently of the others. This type of operation is often called the 'mixer'. The mixer has four competing goals:

1. Minimize the error between the actual and desired forces
2. Minimize the error between the actual and desired torques
3. Minimize thruster utilization (to conserve fuel)
4. Minimize torque coil utilization (to conserve electric power)

The method used to balance these competing goals is the same as the control allocation task from the cluster control algorithm – constrained linear least squares optimization. The problem statement for this type of problem is:

$$\min_{U_M} \|W_M C_M \boldsymbol{U_M} - \boldsymbol{d_M}\|^2 ; \quad subject\ to: \quad A_M \boldsymbol{U_M} < \boldsymbol{b_M}\ and\ \boldsymbol{L_M} \leq \boldsymbol{U_M} \leq \boldsymbol{H_M} \tag{7.1}$$

In this case, the optimization variables are the actuator inputs:

$$\boldsymbol{U_M} = [u_{x1} \quad u_{x2} \quad u_{y1} \quad u_{y2} \quad u_{z1} \quad u_{z2} \quad \mu_x \quad \mu_y \quad \mu_z]^T \tag{7.2}$$

From the stated actuator limitations before, the solution space for the actuator commands can be obtained:

$$\boldsymbol{L_M} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\mu_{sat} \quad -\mu_{sat} \quad -\mu_{sat}]^T \tag{7.3}$$

$$\boldsymbol{H_M} = [u_{sat} \quad u_{sat} \quad u_{sat} \quad u_{sat} \quad u_{sat} \quad u_{sat} \quad \mu_{sat} \quad \mu_{sat} \quad \mu_{sat}]^T \tag{7.4}$$

The total thrust constraint imposed by the thrust slice electronics is expressed in the A,b constraints:

$$A_M = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]; \; b_M = u_{max} \tag{7.5}$$

Note that this is only so easily expressed as a linear equation because all of the thruster commands are required to be expressed as a nonnegative number by design.

As before, the system is augmented with a control penalty on each actuator to ensure the solution uses minimum control effort. The actuator models from the previous sections can be combined with the control penalty equations as:

$$C_M = \begin{bmatrix} F_M & 0_{3x3} \\ T_M & [B_m^{\mathcal{B}} \times] \\ I_{6x6} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \tag{7.6}$$

$$F_M = \begin{bmatrix} | & | & & | \\ -\boldsymbol{d_1} & -\boldsymbol{d_2} & \dots & -\boldsymbol{d_6} \\ | & | & & | \end{bmatrix} \tag{7.7}$$

$$T_M = \begin{bmatrix} | & | & & | \\ \boldsymbol{\rho_1} \times -\boldsymbol{d_1} & \boldsymbol{\rho_2} \times -\boldsymbol{d_2} & \dots & \boldsymbol{\rho_6} \times -\boldsymbol{d_6} \\ | & | & & | \end{bmatrix} \tag{7.8}$$

The F_L describes the forces produced by the thrusters, the T_L matrix describes the torques produced by the thrusters. The solution to the linear system is composed of the desired forces and torques expressed in the body frame augmented with zeros for the control penalty row:

$$\boldsymbol{d_M} = \begin{bmatrix} \boldsymbol{F_{des}^{\mathcal{B}}} \\ \boldsymbol{T_{des}^{\mathcal{B}}} \\ 0_{6x1} \\ 0_{3x1} \end{bmatrix} = \begin{bmatrix} R^{\mathcal{BL}} \boldsymbol{F_{des}^{\mathcal{L}}} \\ \boldsymbol{T_{des}^{\mathcal{B}}} \\ 0_{6x1} \\ 0_{3x1} \end{bmatrix} \tag{7.9}$$

In this case, the weighting function must relate the relative value of force vs torque accuracy as well as actuator output vs control effort.

$$W_M = \begin{bmatrix} \frac{1}{u_{max}} I_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & \frac{\rho_\tau}{\mu_{max}\|B_m^{\mathcal{B}}\|} I_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & \frac{\rho_u}{u_{max}} I_{6x6} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & \frac{\rho_u \rho_{MTQ}}{\mu_{max}} I_{3x3} \end{bmatrix} \tag{7.10}$$

Each element is normalized by a bounding maximum value in order to bring the magnitude of each equation to roughly unity, so that the tuning parameters are more intuitive. Three tuning parameters can be used to tune the mixer to achieve the desired performance.

The first tuning parameter is ρ_τ which represents the relative weighting of force to torque. Increasing this parameter causes torque accuracy to become more important than force accuracy.

The second tuning parameter is ρ_u which represents the relative weighting of control effort to force and torque accuracy. Increasing this parameter causes force and torque accuracy to become less important than minimizing control effort.

The final tuning parameter is ρ_{MTQ} which represents the relative weighting of the thruster and torque coil control actions. Increasing this parameter causes torque coil control actions to be penalized more than thruster control actions.

For this system, the stable tuning parameters were generally found to obey the following relationship:

$$\rho_\tau > 1 \gg \rho_u; \quad \rho_{MTQ} \ll 1 \tag{7.11}$$

Because the system is not power limited, the torque coil control has very little cost to the system, so it is penalized very little relative to the thruster. However, the thruster itself is also quite efficient with respect to fuel consumption but its low power means that it has limited ability to effect the system state. Therefore, it is important to match the desired force and torque as closely as possible. Finally, torque is prioritized over force because the system must maintain nominal attitude to obtain continuous sensor data, especially during payload operations.

CHAPTER 8

ESTIMATION

8.1 POSITION AND VELOCITY

The GPS units come with simple onboard filtering, so no estimation of the GPS signal is required. Furthermore, the GPS units also come with differential corrections, so getting relative positions of the spacecraft is also simple and accurate. Therefore, the estimation has been focused almost entirely on the spacecraft attitude.

8.2 ATTITUDE ESTIMATION

The attitude quantity being estimated is the quaternion which describes the rotation from the inertial frame to the body frame. This is represented notationally as $\boldsymbol{q^{BI}}$. The estimator developed for this application was based primarily on [18], leveraging similarity with other estimators developed for Cubesat applications at MIT. The implemented estimator is an Extended Kalman Filter (EKF). The estimator error dynamics are assumed to be of the form:

$$\boldsymbol{\Delta \dot{x}}(t) = F_e(t)\boldsymbol{\Delta x}(t) + G_e(t)\boldsymbol{w}(t) \tag{8.1}$$

Where $\boldsymbol{\Delta x}(t)$ is the estimator error state and $\boldsymbol{w}(t)$ is the process noise. Measurements are assumed to occur at discrete times. The measurement model at the k'th measurement is given by:

$$\boldsymbol{y_k} = h_k(\boldsymbol{\hat{x}_k}) + \boldsymbol{v_k} \tag{8.2}$$

Where $\boldsymbol{y_k}$ is the measurement, $\boldsymbol{v_k}$ is the measurement noise and $\boldsymbol{h_k}(\boldsymbol{\hat{x}_k})$ is the sensitivity matrix evaluated at the current estimated state.

8.2.2 Derivation of Error Dynamics

The EKF uses a multiplicative quaternion error model, which means that

$$\boldsymbol{\Delta q} = \boldsymbol{q} \otimes \widehat{\boldsymbol{q}}^{-1} \tag{8.3}$$

Assuming small angular errors between the estimated and actual quaternion, the quaternion can be approximated as:

$$\boldsymbol{\Delta q} \approx \begin{bmatrix} \boldsymbol{\Delta \alpha / 2} \\ 1 \end{bmatrix} \tag{8.4}$$

Where $\boldsymbol{\alpha}$ represents the angles around the body x, y, and z axes. Thus, the estimator only requires three states for a quaternion. The EKF formulation used treats the gyro noise as process noise (instead of measurement noise) and estimates the biases alongside the attitude. Recall that the gyros are modeled as:

$$\bar{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta_v} \tag{8.5}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\eta_u} \tag{8.6}$$

The estimated angular velocity is given by:

$$\widehat{\boldsymbol{\omega}} = \bar{\boldsymbol{\omega}} - \widehat{\boldsymbol{\beta}} \tag{8.7}$$

Because the gyro drift's time derivative is assumed to be zero-mean white noise, it is assumed that the estimated gyro biases do not drift. Thus:

$$\dot{\widehat{\boldsymbol{\beta}}} = \mathbf{0} \tag{8.8}$$

Therefore, the angular rate error is given by:

$$\boldsymbol{\Delta \omega} = \boldsymbol{\omega} - \widehat{\boldsymbol{\omega}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta} - \boldsymbol{\eta_v} = -(\boldsymbol{\Delta \beta} + \boldsymbol{\eta_v}) \tag{8.9}$$

Finally, the state error vector and process noise vectors are:

$$\boldsymbol{\Delta x} = \begin{bmatrix} \boldsymbol{\Delta\alpha} \\ \boldsymbol{\Delta\beta} \end{bmatrix}; \quad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{\eta_v} \\ \boldsymbol{\eta_u} \end{bmatrix} \tag{8.10}$$

The linearized dynamics can be shown to be:

$$P\boldsymbol{\Delta\dot{\alpha}} = [-\boldsymbol{\omega}\times]\boldsymbol{\Delta\alpha} - (\boldsymbol{\Delta\beta} - \boldsymbol{\eta_v}) \tag{8.11}$$

$$\boldsymbol{\Delta\dot{\beta}} = \boldsymbol{\eta_u} \tag{8.12}$$

The estimator's error model is now given by:

$$\boldsymbol{\Delta\dot{\hat{x}}} = \hat{F}_e(t)\boldsymbol{\Delta\hat{x}} + \hat{G}_e(t)\hat{\boldsymbol{w}}(t) \tag{8.13}$$

$$\hat{F}_e(t) = \begin{bmatrix} [-\hat{\boldsymbol{\omega}}(t)\times] & -I_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix}; \; \hat{G}_e(t) = \begin{bmatrix} -I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \tag{8.14}$$

8.2.3 Measurement Step

The EKF is simplified by pre-computing a measurement quaternion from the sensors using the triad method. This greatly simplifies the computations in the EKF itself at a small loss of accuracy due to discarding some redundant information from the sensors. The triad method is a basic two-sensor fusion algorithm which produces a direction cosine matrix from two unit vector measurements in two frames. This is possible because a quaternion is composed of three independent pieces of information relating the two frames (the fourth element results from the unit length constraint). Similarly, a unit vector contains two pieces of information relating the two frames (third element results from unit length constraint). Thus, assuming the vectors are not parallel, two vector measurements contain more than enough information to extract a quaternion.

The triad algorithm requires two vectors expressed in two frames. Let these vectors be called $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. Let the two frames be called $\mathcal{C}$ and $\mathcal{D}$. The vectors are related by the following identities:

$$\boldsymbol{v_1^{\mathcal{C}}} = R^{\mathcal{CD}}\boldsymbol{v_1^{\mathcal{D}}} \tag{8.15}$$

$$\boldsymbol{v_2^{\mathcal{C}}} = R^{\mathcal{CD}}\boldsymbol{v_2^{\mathcal{D}}} \tag{8.16}$$

The method relies on the following identity:

$$\boldsymbol{v_1^{\mathcal{C}}} \times \boldsymbol{v_2^{\mathcal{C}}} = R^{\mathcal{CD}}\left(\boldsymbol{v_1^{\mathcal{D}}} \times \boldsymbol{v_2^{\mathcal{D}}}\right) \tag{8.17}$$

The vector which is more trusted should be used as $\boldsymbol{v_1}$. The steps in the algorithm are as follows:

$$\boldsymbol{S^{\mathcal{C}}} = \frac{\boldsymbol{v_1^{\mathcal{C}}}}{\left|\left|\boldsymbol{v_1^{\mathcal{C}}}\right|\right|}; \qquad \boldsymbol{S^{\mathcal{D}}} = \frac{\boldsymbol{v_1^{\mathcal{D}}}}{\left|\left|\boldsymbol{v_1^{\mathcal{D}}}\right|\right|} \tag{8.18}$$

$$\boldsymbol{M^{\mathcal{C}}} = \frac{\boldsymbol{v_1^{\mathcal{C}}} \times \boldsymbol{v_2^{\mathcal{C}}}}{\left|\left|\boldsymbol{v_1^{\mathcal{C}}} \times \boldsymbol{v_2^{\mathcal{C}}}\right|\right|}; \qquad \boldsymbol{M^{\mathcal{D}}} = \frac{\boldsymbol{v_1^{\mathcal{D}}} \times \boldsymbol{v_2^{\mathcal{D}}}}{\left|\left|\boldsymbol{v_1^{\mathcal{D}}} \times \boldsymbol{v_2^{\mathcal{D}}}\right|\right|} \tag{8.19}$$

Repeated application of the previously stated identities results in the following system of equations:

$$\begin{bmatrix} | & | & | \\ \boldsymbol{S^{\mathcal{C}}} & \boldsymbol{M^{\mathcal{C}}} & \boldsymbol{S^{\mathcal{C}}} \times \boldsymbol{M^{\mathcal{C}}} \\ | & | & | \end{bmatrix} = R^{\mathcal{CD}} \begin{bmatrix} | & | & | \\ \boldsymbol{S^{\mathcal{D}}} & \boldsymbol{M^{\mathcal{D}}} & \boldsymbol{S^{\mathcal{D}}} \times \boldsymbol{M^{\mathcal{D}}} \\ | & | & | \end{bmatrix} \tag{8.20}$$

Using the equivalence of inverses and transposes for direction cosine matrices, the direction cosine matrix relating the two frames can then be calculated as:

$$R^{\mathcal{CD}} = \begin{bmatrix} | & | & | \\ \boldsymbol{S^{\mathcal{C}}} & \boldsymbol{M^{\mathcal{C}}} & \boldsymbol{S^{\mathcal{C}}} \times \boldsymbol{M^{\mathcal{C}}} \\ | & | & | \end{bmatrix} \begin{bmatrix} | & | & | \\ \boldsymbol{S^{\mathcal{D}}} & \boldsymbol{M^{\mathcal{D}}} & \boldsymbol{S^{\mathcal{D}}} \times \boldsymbol{M^{\mathcal{D}}} \\ | & | & | \end{bmatrix}^T \tag{8.21}$$

In this case, the sun sensor is considered to be significantly more accurate than the magnetometer. Although the magnetometer is expected to be a more accurate measurement of the actual magnetic field (within 0.12°), the ability of the onboard model to predict the magnetic field can be in error by as much as 10° [24]. The sun sensor yields a less accurate measurement of the actual sun vector (1°), but the onboard model of the sun's motion is accurate to 0.02° [25].

Therefore, the measured quaternion can be estimated by:

$$\overline{\boldsymbol{S}}^{\boldsymbol{\mathcal{B}}} = \frac{\overline{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{\mathcal{B}}}}{\left|\left|\overline{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{\mathcal{B}}}\right|\right|}; \quad \tilde{\boldsymbol{S}}^{\boldsymbol{I}} = \frac{\tilde{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{I}}}{\left|\left|\tilde{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{I}}\right|\right|} \tag{8.22}$$

$$\bar{\boldsymbol{M}}^{\boldsymbol{\mathcal{B}}} = \frac{\bar{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{\mathcal{B}}} \times \bar{\boldsymbol{B}}_{\boldsymbol{m}}^{\boldsymbol{\mathcal{B}}}}{\left|\left|\bar{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{\mathcal{B}}} \times \bar{\boldsymbol{B}}_{\boldsymbol{m}}^{\boldsymbol{\mathcal{B}}}\right|\right|}; \qquad \tilde{\boldsymbol{M}}^{\boldsymbol{I}} = \frac{\tilde{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{I}} \times \tilde{\boldsymbol{B}}_{\boldsymbol{m}}^{\boldsymbol{I}}}{\left|\left|\tilde{\boldsymbol{S}}_{\boldsymbol{v}}^{\boldsymbol{I}} \times \tilde{\boldsymbol{B}}_{\boldsymbol{m}}^{\boldsymbol{I}}\right|\right|} \tag{8.23}$$

$$\bar{R}^{\mathcal{B}I} = \begin{bmatrix} | & | & | \\ \bar{\boldsymbol{S}}^{\boldsymbol{\mathcal{B}}} & \bar{\boldsymbol{M}}^{\boldsymbol{\mathcal{B}}} & \bar{\boldsymbol{S}}^{\boldsymbol{\mathcal{B}}} \times \bar{\boldsymbol{M}}^{\boldsymbol{\mathcal{B}}} \\ | & | & | \end{bmatrix} \begin{bmatrix} | & | & | \\ \tilde{\boldsymbol{S}}^{\boldsymbol{I}} & \tilde{\boldsymbol{M}}^{\boldsymbol{I}} & \tilde{\boldsymbol{S}}^{\boldsymbol{I}} \times \tilde{\boldsymbol{M}}^{\boldsymbol{I}} \\ | & | & | \end{bmatrix}^{T} \tag{8.24}$$

$\bar{\boldsymbol{q}}^{\boldsymbol{\mathcal{B}I}}$ is then calculated from this direction cosine matrix. The measured quaternion is the conversion of this direction cosine matrix.

Because it is assumed that the gyro measurement is incorporated into the dynamics via the bias equation, the only measurements occurring are those which measure attitude. Furthermore, because the triad method is used to generate a complete quaternion, the sensitivity matrix becomes very simple, because the full state is being measured. Therefore:

$$h_k(\hat{\boldsymbol{x}}_{\boldsymbol{k}}) \equiv H = [I_{3x3} \quad 0_{3x3}] \tag{8.25}$$

8.2.4 Update Step

The Update process incorporates the sensed quaternion to update the state estimate and covariance. As used here a 'minus' superscript indicates the pre-update value while the 'plus' subscript indicates the post-update value. The subscript indicates the timestep.

The first step is to determine the innovation in the error vector given by the measurement. This is found by computing the error between the measured and estimated quaternions:

$$\boldsymbol{\Delta\bar{q}} = \bar{\boldsymbol{q}} \otimes \hat{\boldsymbol{q}}_{\boldsymbol{k}}^{-1} \tag{8.26}$$

The elements of the innovation derived from the measurement can then be extracted as:

$$\boldsymbol{\Delta\bar{q}} = \begin{bmatrix} \boldsymbol{\Delta\bar{\alpha}/2} \\ \sim \end{bmatrix} \tag{8.27}$$

The Kalman gain can be computed via:

$$L_k = Q_k^- H^T [H Q_k^- H^T + R]^{-1} \tag{8.28}$$

The Error State Covariance can then be updated:

$$Q_k^+ = (I_{6x6} - L_k H) Q_k^- \tag{8.29}$$

The error state can then be updated:

$$\boldsymbol{\Delta\hat{x}} = L_k \boldsymbol{\Delta\bar{\alpha}} \tag{8.30}$$

The updates to the estimator state can then be extracted:

$$\boldsymbol{\Delta\hat{x}} = \begin{bmatrix} \boldsymbol{\Delta\hat{\alpha}} \\ \boldsymbol{\Delta\hat{\beta}} \end{bmatrix} \tag{8.31}$$

This allows the calculation of the estimated error quaternion:

$$\boldsymbol{\Delta\hat{q}} = \begin{bmatrix} \boldsymbol{\Delta\hat{\alpha}}/2 \\ \sqrt{1 - \dfrac{\boldsymbol{\Delta\hat{\alpha}^T \Delta\hat{\alpha}}}{4}} \end{bmatrix} \tag{8.32}$$

This allows the computation of the updated estimated quaternion:

$$\boldsymbol{\hat{q}_k^+} = \boldsymbol{\delta\hat{q}} \otimes \boldsymbol{\hat{q}_k^-} \tag{8.33}$$

The gyro biases can also be updated:

$$\boldsymbol{\hat{\beta}_k^+} = \boldsymbol{\hat{\beta}_k^-} + \boldsymbol{\Delta\hat{\beta}} \tag{8.34}$$

This allows the computation of the estimated angular rates from the gyro measurements:

$$\boldsymbol{\hat{\omega}_k} = \boldsymbol{\hat{\omega}_k} - \boldsymbol{\hat{\beta}_k^+} \tag{8.35}$$

8.2.5 Predict Step

The prediction process is used to predict the state of the estimated variables at the next time step. To reduce computation, a discrete propagation is used and the update equations are derived using power series expansion. The predicted quaternion is given by:

$$\boldsymbol{\hat{q}_{k+1}^-} = \underline{\Omega}(\boldsymbol{\hat{\omega}}_k) \boldsymbol{\hat{q}_k^+} \tag{8.36}$$

$$\underline{\Omega}(\widehat{\boldsymbol{\omega}}_k) = \begin{bmatrix} cos\left(\frac{1}{2}\|\widehat{\boldsymbol{\omega}}_k\|\Delta t\right)\boldsymbol{I}_{3\times3} - \left[\underline{\widehat{\boldsymbol{\psi}}}_k \times\right] & \underline{\widehat{\boldsymbol{\psi}}}_k \\ -\underline{\widehat{\boldsymbol{\psi}}}_k^T & cos\left(\frac{1}{2}\|\widehat{\boldsymbol{\omega}}_k\|\Delta t\right) \end{bmatrix} \tag{8.37}$$

$$\underline{\widehat{\boldsymbol{\psi}}}_{\boldsymbol{k}} = \frac{sin\left(\frac{1}{2}\|\widehat{\boldsymbol{\omega}}_k\|\Delta t\right)\widehat{\boldsymbol{\omega}}_k}{\|\widehat{\boldsymbol{\omega}}_k\|} \tag{8.38}$$

The covariance is propagated as:

$$Q_{k+1}^- = \Phi_k Q_k^+ \Phi_k^T + W_k \tag{8.39}$$

$$\Phi_{k_{11}} = I_{3\times3} - [\widehat{\boldsymbol{\omega}}_k \times]\frac{sin(\|\widehat{\boldsymbol{\omega}}_k\|\Delta t)}{\|\widehat{\boldsymbol{\omega}}_k\|} + [\widehat{\boldsymbol{\omega}}_k \times]^2\frac{1 - cos(\|\widehat{\boldsymbol{\omega}}_k\|\Delta t)}{\|\widehat{\boldsymbol{\omega}}_k\|^2} \tag{8.40}$$

$$\Phi_{k_{12}} = -\Delta t\boldsymbol{I}_{3\times3} + [\widehat{\boldsymbol{\omega}}_k \times]\frac{1 - cos(\|\widehat{\boldsymbol{\omega}}_k\|\Delta t)}{\|\widehat{\boldsymbol{\omega}}_k\|} + [\widehat{\boldsymbol{\omega}}_k^+ \times]^2\frac{\|\widehat{\boldsymbol{\omega}}_k\|\Delta t - sin(\|\widehat{\boldsymbol{\omega}}_k\|\Delta t)}{\|\widehat{\boldsymbol{\omega}}_k\|^3} \tag{8.41}$$

$$\Phi_k = \begin{bmatrix} \Phi_{k_{11}} & \Phi_{k_{12}} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{8.42}$$

$$W_k = \begin{bmatrix} -I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix}\begin{bmatrix} \left(\sigma_v^2\Delta t + \frac{1}{3}\sigma_u^2\Delta t^3\right)I_{3x3} & \frac{1}{2}\sigma_u^2\Delta t^2 I_{3x3} \\ \frac{1}{2}\sigma_u^2\Delta t^2 I_{3x3} & \sigma_u^2\Delta t I_{3x3} \end{bmatrix}\begin{bmatrix} -I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix}^T \tag{8.43}$$

It is useful to note that the diagonal elements of Q_k^+ directly represent the estimated variance of the estimator error about each of the spacecraft body axes. This is used to derive the estimated 3-sigma bounds on estimator error.

CHAPTER 9

ATTITUDE-DEPENDENT DISTURBANCE MODELING

9.1 INTRODUCTION

One of the objectives of the mission is to characterize the performance of the electrospray thruster package. However, because the IMU package is not sensitive enough to directly measure the extremely low thrust levels of the thrusters, it will be necessary to characterize the thrust levels over some interval by examining the dynamic response of the system to known actuator commands. An additional factor complicating this approach is that the magnitude of the disturbances (especially aerodynamic at low altitudes) is roughly the same order as the thrusters. Thus, accurately characterizing the thruster performance requires a good model of the disturbance forces and torques.

The Knudsen number is a measure of ratio between the molecular free path length and a representative length scale. In all but the lowest of LEO orbits, the Knudsen number is finite and greater than unity, indicating that the molecular free path length is greater than the length of the object. Using 'normal' CFD techniques (Navier-Stokes equations) is not appropriate. In these flow regimes, one widely used solution method is Direct Simulation Monte Carlo (DSMC). This method simulates the physical motion of a finite number of simulated molecules which represent a large number of real molecules. This method models molecule-molecule and molecule-surface collisions are modeled in a variety of different ways with varying complexity and accuracy.

However, most DSMC modeling software is extremely complex and generally only used for much bigger projects. In an effort to model attitude-dependent effects without the full rigor of DSMC methods, a less accurate method which models the most important effects was developed. This method, hereafter referred to as the Ray Tracing Method (RTM) makes a few assumptions about the flow characteristics to simplify the solution.

1. There are no molecule-molecule collisions. While this isn't strictly accurate, the majority of the collisions are molecule-surface collisions for high Knudsen numbers.
2. The incoming flow can be approximated as a collimated (all particles moving in one direction). Because the random motion of molecules in most LEO orbits is on the order of a few hundred m/s compared to orbital velocity of 7.8 km/s, this is a valid approximation.
3. All surface-molecule collisions result in perfectly elastic reflection, uniformly diffuse reflection, or adsorption. While the method may be extendable to directionally non-uniform reflection, at present it does not include these effects.

Under these assumptions, it is only necessary to create one finite plane of particles, normal to the direction of the flow. The plane need only be large enough to ensure that the projection of the body onto the plane does not exceed the bounds of the plane.

Furthermore, it is not necessary to accurately propagate the particles dynamically through time since there are no molecule-molecule collisions to consider and the flow is assumed collimated. In this way, the full trajectories of all particles can be solved in parallel knowing only the initial position of each particle, the flow direction, and the geometry of the body.

9.2 COORDINATE SYSTEM

The coordinate system for this method is based on the Local-Vertical, Local-Horizontal frame for circular orbits such that the X axis is in the direction of zenith, the Y axis is in the direction of the orbital velocity and the Z axis is in the direction of the orbit normal vector. The body frame is denoted by the notation $\mathcal{R}$, to allow for different coordinate systems in the definition of the body axes in the RTM (because it may be derived from a CAD model, etc.). It is possible and desirable that $\mathcal{R} = \mathcal{B}$.

Because the flow is assumed to be collimated, there are only two degrees of freedom for the flow plane orientation with respect to the body. For this method, elevation (ϵ) and azimuth (α) angles of the flow to the body have been chosen to parameterize this space. The specific rotation matrix is:

$$R^{\mathcal{LR}} = \begin{bmatrix} cos(\epsilon) & -cos(\alpha)\,sin(\epsilon) & -sin(\alpha)\,sin(\epsilon) \\ sin(\epsilon) & cos(\alpha)\,cos(\epsilon) & sin(\alpha)\,cos(\epsilon) \\ 0 & -sin(\alpha) & cos(\alpha) \end{bmatrix} \tag{9.1}$$

Additionally,

$$-\frac{\pi}{2} \leq \epsilon \leq \frac{\pi}{2}, -\pi \leq \alpha \leq \pi \tag{9.2}$$

The direction toward the flow plane can be expressed in $\mathcal{R}$ as:

$$\hat{e}_{flow}^{\mathcal{R}} = R^{\mathcal{RL}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \sin\epsilon \\ \cos\alpha\,\sin\epsilon \\ \sin\alpha\,\cos\epsilon \end{bmatrix} \tag{9.3}$$

Figure 9.1 shows a flow plane, the sphere bounding a geometric model of a MotherCube satellite and the ray propagation of all rays which hit the satellite.

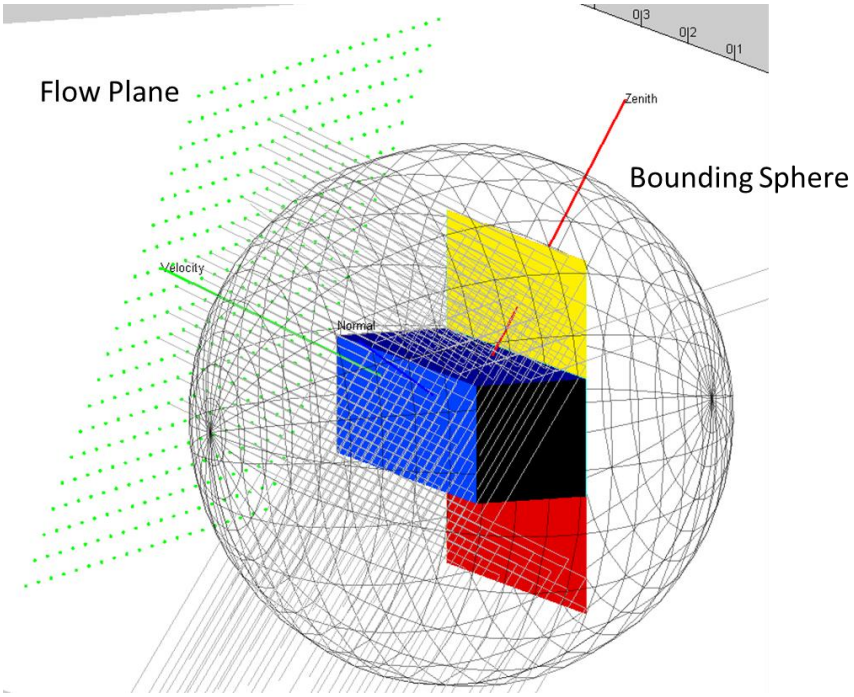

Figure 9.1 - Graphical Example of Ray Tracing

9.3 INPUTS

The required inputs for this method are the geometry and surface properties of the body. This is expressed as a series of 3-point faces, each of which is assumed to be of uniform material property. Currently, these faces are generated manually as a series of points, but this could be extended to

interface with CAD programs. Importantly, the mass properties are NOT required for the ray tracing step.

The material properties required for each material are coefficients of elastic reflection, diffuse reflection, and adsorption, which must be positive and sum to unity.

$$C_{elastic} + C_{diffuse} + C_{adsorp} = 1 \tag{9.4}$$

9.4 Ray Tracing Calculations

The core of the method is the tracing and reflecting of rays. This process can be broken down into the following steps for each bounce of each ray:

1. Identify all planes which intersect the ray
2. Identify which plane is intersected first along the ray
3. Identify the impact point of the ray and record
4. Update the starting position, slope and Intensity of the ray for the next iteration
5. Update the coefficients of Force and Torque

9.4.1 Definitions

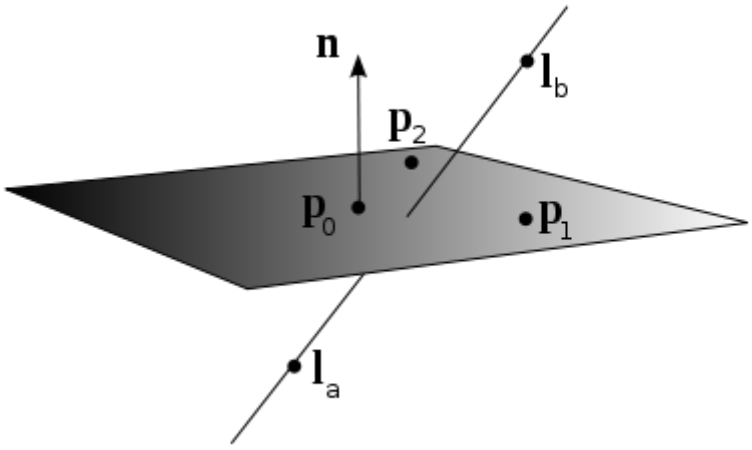

Figure 9.2 - Line Plane Intersection Diagram

Assume the plane is finite and bounded by the three points $\boldsymbol{p_0}, \boldsymbol{p_1}, \boldsymbol{p_2}$and the line segment is finite and bounded by the points l_a and l_b. All of these quantities are assumed to be members of $\mathbb{R}^3$. The points interior to the plane are then given by:

$$\boldsymbol{p_0} + (\boldsymbol{p_1} - \boldsymbol{p_0})u + (\boldsymbol{p_2} - \boldsymbol{p_0})v, \qquad u, v \in [0,1] \; and \; (u+v) \leq 1 \tag{9.5}$$

Similarly, the points on the line segment are given by:

$$\boldsymbol{l_a} + (\boldsymbol{l_b} - \boldsymbol{l_a})t, \;\; t \in [0,1] \tag{9.6}$$

The normal vector can be computed as:

$$\boldsymbol{n} = (\boldsymbol{p_1} - \boldsymbol{p_0}) \times (\boldsymbol{p_2} - \boldsymbol{p_0}) \tag{9.7}$$

It should be noted that this is not necessarily an outward-facing normal.

9.4.2 Step 0 – Initialization (first iteration only)

The first point for each ray is in the original flow plane. The slope of this line is the flow direction. The Intensity of this ray is unity. Note that the Intensity represents only the elastic portion of the ray.

9.4.3 Step 1 – Intersection Identification

The next step is to identify all of the possible intersections of this ray with the planes in the body. A method for determining whether a line segment intersects the plane is to solve for the intersection of a hypothetical infinite plane and line and check to see if the intersection lines on the line segment and within the finite plane. This is solved by the following equation:

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \begin{bmatrix} x_a - x_b & x_1 - x_0 & x_2 - x_0 \\ y_a - y_b & y_1 - y_0 & y_2 - y_0 \\ z_a - z_b & z_1 - z_0 & z_2 - z_0 \end{bmatrix}^{-1} \begin{bmatrix} x_a - x_0 \\ y_a - y_0 \\ z_a - z_0 \end{bmatrix} \tag{9.8}$$

As long as $t \in [0,1]$ and $u, v \in [0,1]$ and $(u+v) \leq 1$, then the line segment intersects the plane. It is useful to check for parallelism of the line and the plane prior to solving this matrix equation to avoid singularities. Additionally, it is useful to use small tolerances around the conditions on t, u, v due to numerical issues. This computation must be performed for each face in the body.

9.4.4 Step 2 – Intersection Selection

In the likely case that the line intersects multiple faces, the intersection which occurs closest to the origin of the ray must be selected. Fortunately, this is simply the face with the smallest positive value for t. A complication arises when a ray exactly intersects the edge or corner of a face, as multiple faces will then have the same intersection point. This is handled by discarding corner hits (which are very rare) and by discarding any edge hits in which both faces do not have the same normal vector.

9.4.5 Step 3 – Impact Point Identification

The point of impact (if it exists) is computed as:

$$\boldsymbol{x_i} = \boldsymbol{l_a} + (\boldsymbol{l_b} - \boldsymbol{l_a})t \tag{9.9}$$

9.4.6 Step 4 – Update for Next Iteration

The starting point for the next iteration is simply the impact point of the current iteration. The slope for the subsequent iterations is given by the specular reflection of the ray from the intersected face. Because it is assumed that all bodies are closed, this can be computed as:

$$\boldsymbol{s^{k+1}} = \frac{\boldsymbol{s^k} - 2(\boldsymbol{s^k} \cdot \boldsymbol{n})\boldsymbol{n}}{\left|\left|\boldsymbol{s^k} - 2(\boldsymbol{s^k} \cdot \boldsymbol{n})\boldsymbol{n}\right|\right|} \tag{9.10}$$

Because the ray tracing only needs to trace the path of the elastic reflections, the Intensity can be updated simply by multiplying by the coefficient of elastic collision of the intersected face.

$$I^{k+1} = I^k C_{elastic} \tag{9.11}$$

For complicated geometry which produces many iterations of reflections, it is useful to remove particles below a certain intensity threshold in order to speed up computation without loss of accuracy.

9.4.7 Step 5 – Update Force and Torque Coefficients

Each point in the flow plane represents a certain volume of molecules. If the flow is assumed to have a uniform density of molecules and a uniform flow velocity, then these variables can be

removed from the computation. Force and torque coefficients can be calculated instead and then these coefficients can be used to compute the forces and torques for any flow density and velocity. Assume the square flow plane area (A) is divided equally into N smaller areas, each of which is approximated by a particle at the center of the area.

The force imparted on the body by the i'th ray on the k'th bounce (k superscript is assumed on all quantities unless otherwise stated) is the change in momentum which results from the interaction with the surface. This method models three types of interactions:

9.4.7.1 Elastic Reflection

Simple elastic collision results in a change in momentum proportional to the change in slope of the ray:

$$\boldsymbol{C_{Fi,elastic}} = \frac{AI^k}{N}\left(\boldsymbol{s^k} - \boldsymbol{s^{k+1}}\right)C_{elastic} \tag{9.12}$$

9.4.7.2 Diffuse Reflection

Assuming uniform diffuse reflection and neglecting the compound reflections (i.e. the reflections from the diffuse reflection), the net result on the body is simply:

$$\boldsymbol{C_{Fi,diffuse}} = -\frac{2}{3}\frac{AI^k}{N}\boldsymbol{n}\, C_{diffuse} \tag{9.13}$$

9.4.7.3 Adsorption

Adsorption assumes that the molecules simply stick to the surface. Thus, the entirety of the momentum is transferred to the body:

$$\boldsymbol{C_{Fi,adsorp}} = \frac{AI^k}{N} C_{adsorp}\boldsymbol{s^k} \tag{9.14}$$

Thus, the total force imparted by the i'th particle is:

$$\boldsymbol{C_{Fi}} = \boldsymbol{C_{Fi,elastic}} + \boldsymbol{C_{Fi,diffuse}} + \boldsymbol{C_{Fi,adsorp}} \tag{9.15}$$

The total force coefficient is then expressed as:

$$\boldsymbol{C_F} = \sum_{k=1}^{k_{max}} \sum_{i}^{N} \boldsymbol{C_{Fi}} \tag{9.16}$$

The summation over 'k' indicates the summation of all the bounces of each ray, and the summation over 'i' indicates the summation over all the particles in the flow field. The actual force is computed as:

$$\boldsymbol{F} = \rho V^2 \boldsymbol{C_F} \tag{9.17}$$

where ρ is the average atmospheric density [kg/m^3]and V is the velocity of the incoming flow [m/s]. The torque coefficient for each particle is similarly expressed as:

$$\boldsymbol{C_{Ti}} = \boldsymbol{x_i} \times \boldsymbol{C_{Fi}} \tag{9.18}$$

$$\boldsymbol{C_T} = \sum_{k=1}^{k_{max}} \sum_{i}^{N} \boldsymbol{C_{Ti}} \tag{9.19}$$

The torque can be reconstructed:

$$\boldsymbol{\tau} = \rho V^2 \left(\boldsymbol{C_T} - (\boldsymbol{x_{CG}} \times \boldsymbol{C_F}) \right) \tag{9.20}$$

Where x_{CG} is the center of mass of the body expressed in the same coordinates as the geometry. This is especially interesting because the location of the CG is not necessary during the ray tracing. This means that the location of the CG can be changed without needing to re-run the ray tracing and force/torque coefficient computation.

9.5 Solar Forces and Torques

This method is extendable to calculate solar forces and torques if one assumes that the flow field represents a collimated light source instead of an aerodynamic flow. In fact, the RTM is probably more accurate for solar forces and torques because light does not experience many particle-particle interactions and the flow field (e.g. from the sun) will in fact be very nearly collimated. In order to compute the solar force and torque coefficients, a second set of optical surface material properties is required for each face, similar to the physical properties specified for the aerodynamic portion:

$$C_{S,specular} + C_{S,diffuse} + C_{S,absorb} = 1 \tag{9.21}$$

The equations for specular reflection, diffuse reflection and absorption mimic the equations for elastic, diffuse, and adsorption behavior from the aerodynamic method. Crucially, adding solar force and torque computation does not require a second set of line-plane intersection calculations, which are the majority of the computation, although it does require bookkeeping a second Intensity value for the optical reflections. The solar forces and torques are normalized by the solar radiation pressure, which is related to the Solar Irradiance as:

$$P_{rad} = \frac{I_S}{c} \tag{9.22}$$

At Earth's distance from the sun, this value is approximately 4.6 μPa. This yields a slightly different final step:

$$\boldsymbol{F_{solar}} = \frac{I_S}{c}\boldsymbol{C_{T_{solar}}} \tag{9.23}$$

$$\boldsymbol{\tau_{solar}} = \frac{I_S}{c}\left(\boldsymbol{C_{T_{solar}}} - \left(\boldsymbol{x_{CG}} \times \boldsymbol{C_{F_{solar}}}\right)\right) \tag{9.24}$$

9.6 Solar Power (and Other Visibility Metrics)

Because the RTM already calculates which rays fall upon which face, it is possible to compute the attitude-dependent power generation using the same line-plane intersection computations. Models vary on how to convert the incidence angle of light into solar power, but for now a binary hit/no-hit model is used. This could also be used to easily show which sun sensors would have valid measurements based on attitude or which solar panels would be generating power.

9.7 Implementation

The RTM was implemented mainly in MATLAB with the line-plane-intersection algorithm coded in MEX-C to improve the runtime. As mentioned previously, the RTM creates a set of coefficients which are dependent only on the following parameters:

1. Orientation of the flow plane with respect to the body (2 degrees of freedom)
2. Body geometry (physical shape)

3. Body surface properties (physical and optical)

Notably, the coefficient solution produced by the RTM does NOT depend on any of the following:

4. Moments of inertia
5. Center of mass
6. Altitude
7. Flow speed/light intensity
8. Solar flux (and its associated change in atmospheric density)
9. Time of year/position of earth
10. Position of body in orbit

Fortunately, items 2 and 3 typically do not change much for most satellites over their operational lifetimes. This means that the relevant information can be computed for a 2D grid of flow plane orientations and used as a lookup table both in simulation and on-orbit at extremely low computational burden by converting these coefficients to actual forces and torques using the instantaneous values of items 4-10.

9.8 VALIDATION

In order to check the validity of the algorithm, it was compared to a similar study performed using DSMC. The study [26] aimed to evaluate the possibility of using the geometry of the Cubesat to provide aerodynamic attitude stabilization. In order to evaluate their designs, they developed a 2D model which only modeled pitching torques. Their simplified model is shown below. Note that there are differences in variable names and definitions between this report and [26].

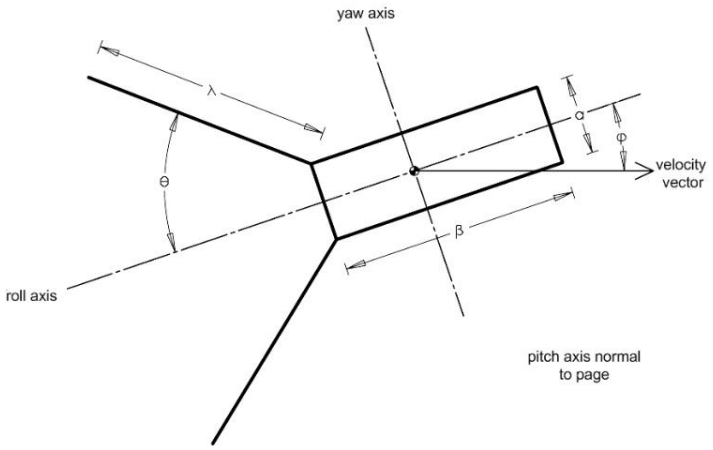

Figure 9.3 - Pitch Torque Model from [26]

To emulate this model, a similar 3D model of a 3U Cubesat with variable solar panel length and angle was creating using the RTM tool. This model is shown below in Figure 9.4 with a 2U length and a 60 degree angle:

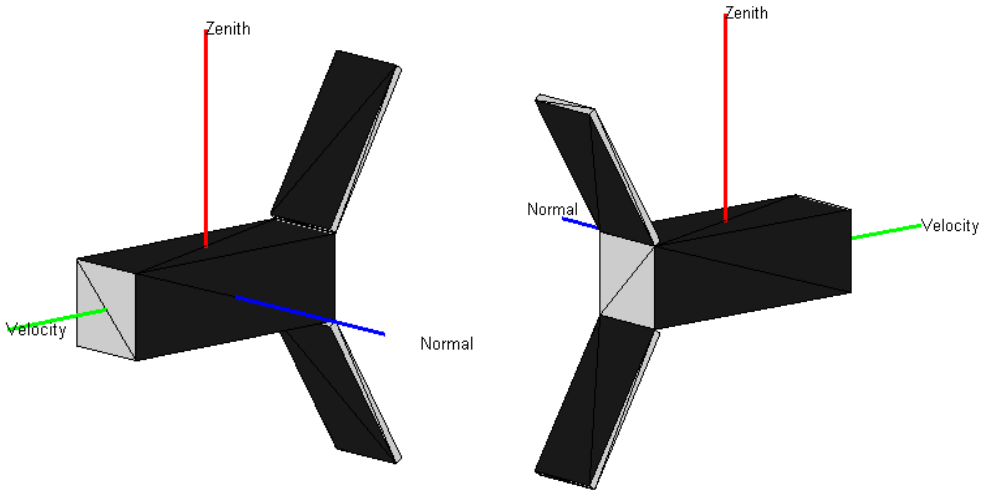

Figure 9.4 - RTM Geometric Model for Pitch Test Validation

One of their case studies was to model the pitch torque as a function of pitch angle for various solar panel angles. This was repeated as closely as possible using the RTM tool. The study [26] specified a 2U solar panels and three solar panel angles (10, 30, 50 degrees) and also specified a 400km circular orbit. However, the study did not specify the CG location or the atmospheric model parameters. Since the RTM only produces a set of coefficients, the air density was used as a fitting parameter to match the magnitudes of the two methods. The parameters required to match magnitudes are similar to mean atmospheric conditions in 2009, which is reasonable since this is the date when the study was published. Since the study discusses the limitation of the Cubesat launch CG requirement, it was assumed that the CG was as far aft as possible. This yields the following comparison in Figure 9.5 (study on left, RTM on right).

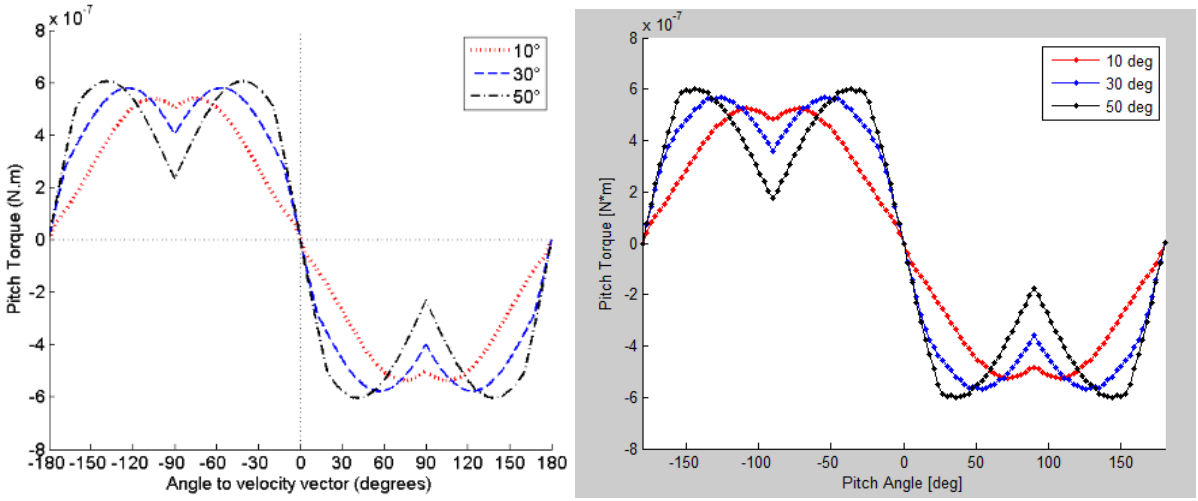

Figure 9.5 - Pitch Test Validation

Although there are some small differences, especially around the peak torque for the 50 degree case, in general the two methods are quite similar, exhibiting many of the same features. Notably, both models predict that the maximum torque increases with increasing solar panel angle and the pitch angle at which the maximum torque occurs also increases at greater pitch angles. Both models also predict a steeper slope near zero pitch angle for higher solar panel angles.

9.9 RESULTS

There are still some uncertainties regarding the satellite configuration, specifically the angle of the solar panels. For the purpose of this report, a set of sample results was generated using a perpendicular solar panel configuration, shown below. In these images, the solar panels are darkened and the metal surfaces are lightened. All surfaces are assumed to have a physical and optical diffuse reflection coefficient of 0.1. All surfaces have a physical adsorption coefficient of 0.1 and an elastic reflection coefficient of 0.8. The solar panel surfaces have a specular reflection coefficient of 0.1 and an absorption coefficient of 0.8. The metal surfaces have a specular reflection coefficient of 0.8 (shiny, but not polished), and an absorption coefficient of 0.1.

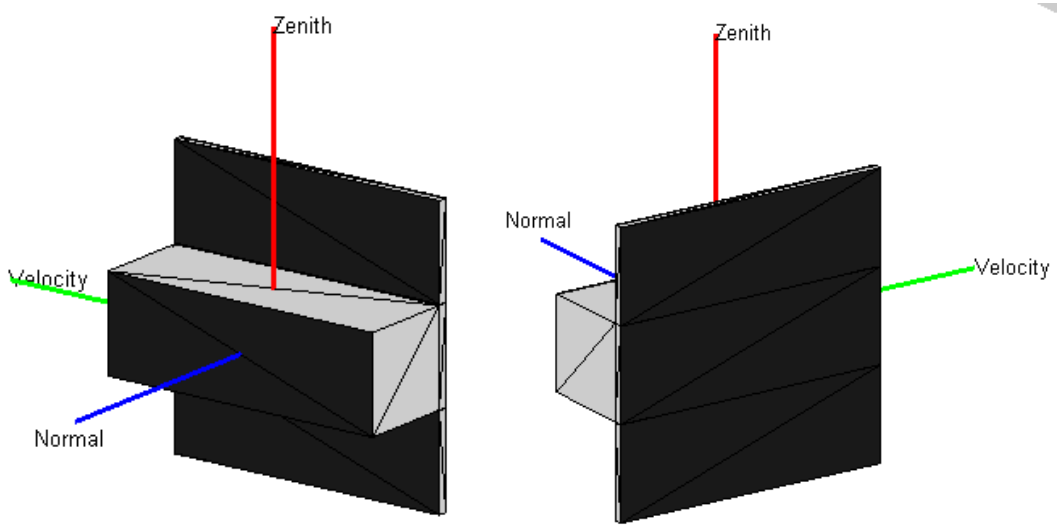

Figure 9.6 - MotherCube RTM Model

Probably the most interesting result is the pure drag result, which measures the forces on the body in the direction of the flow. This is what determines orbit degradation (and therefore satellite lifetime) and differential drag between the different satellites in the cluster. This quickly demonstrates how differential attitude can translate into differential position via aerodynamic drag.

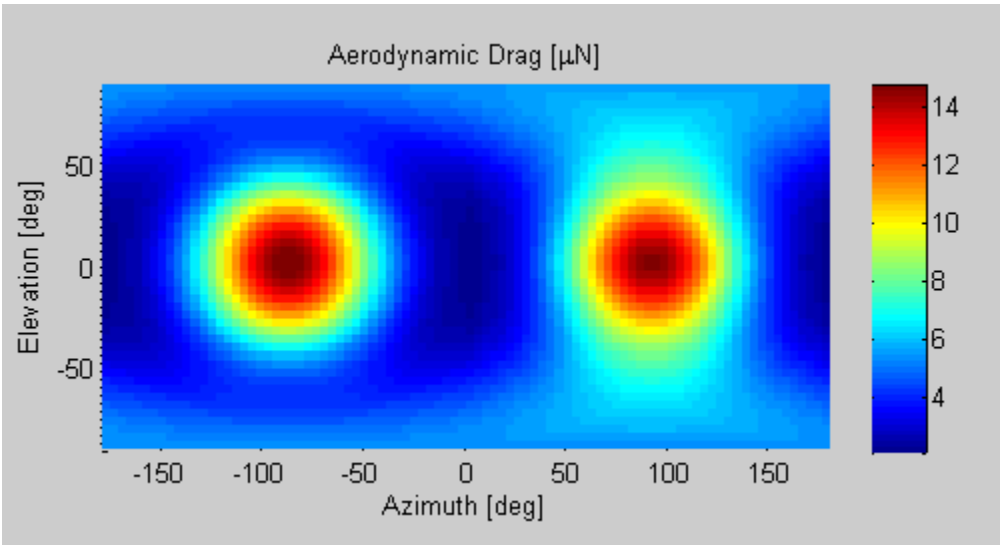

Figure 9.7 - Aerodynamic Drag (RTM Result)

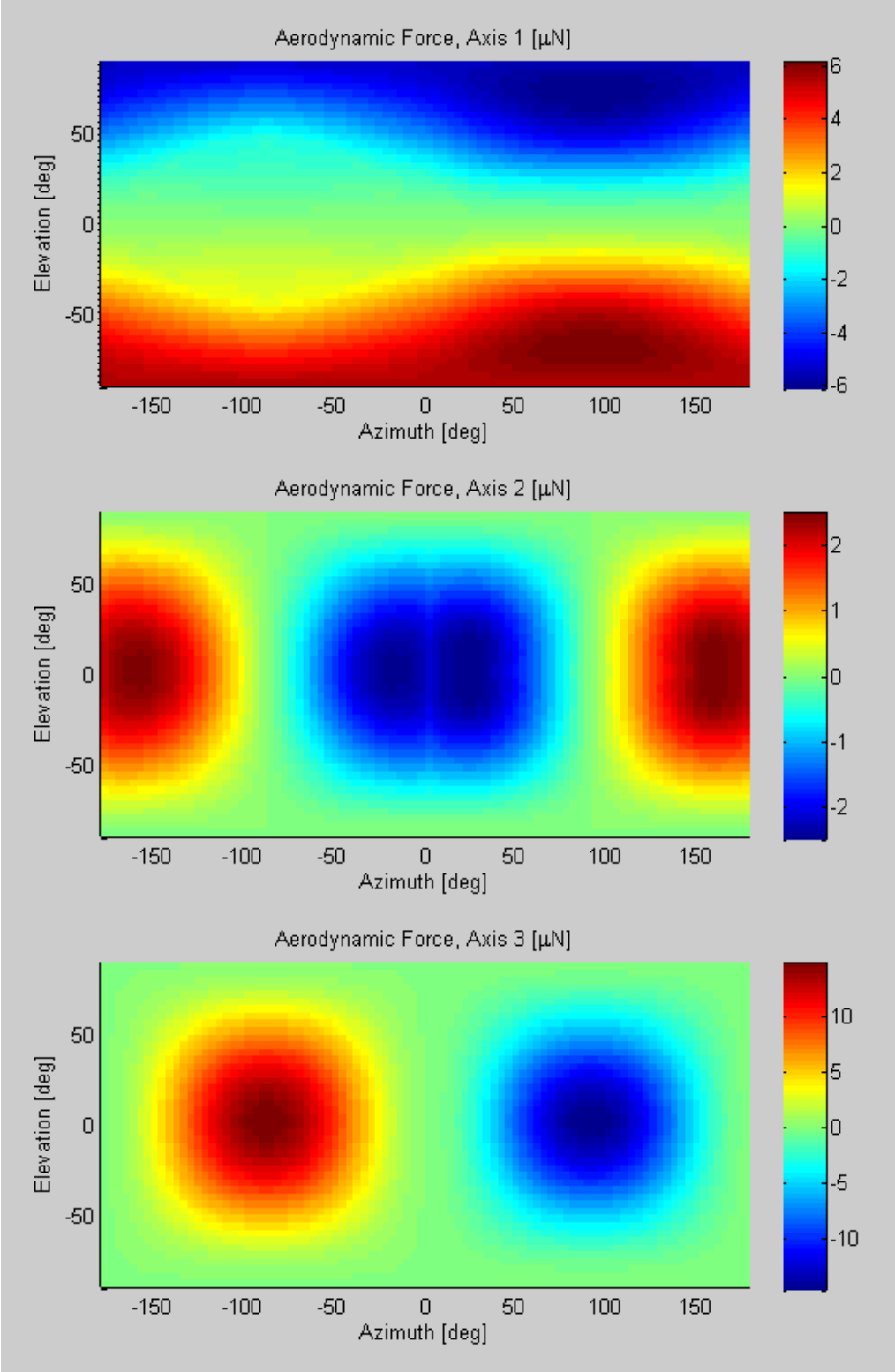

Figure 9.8 - Aerodynamic Forces (RTM Result)

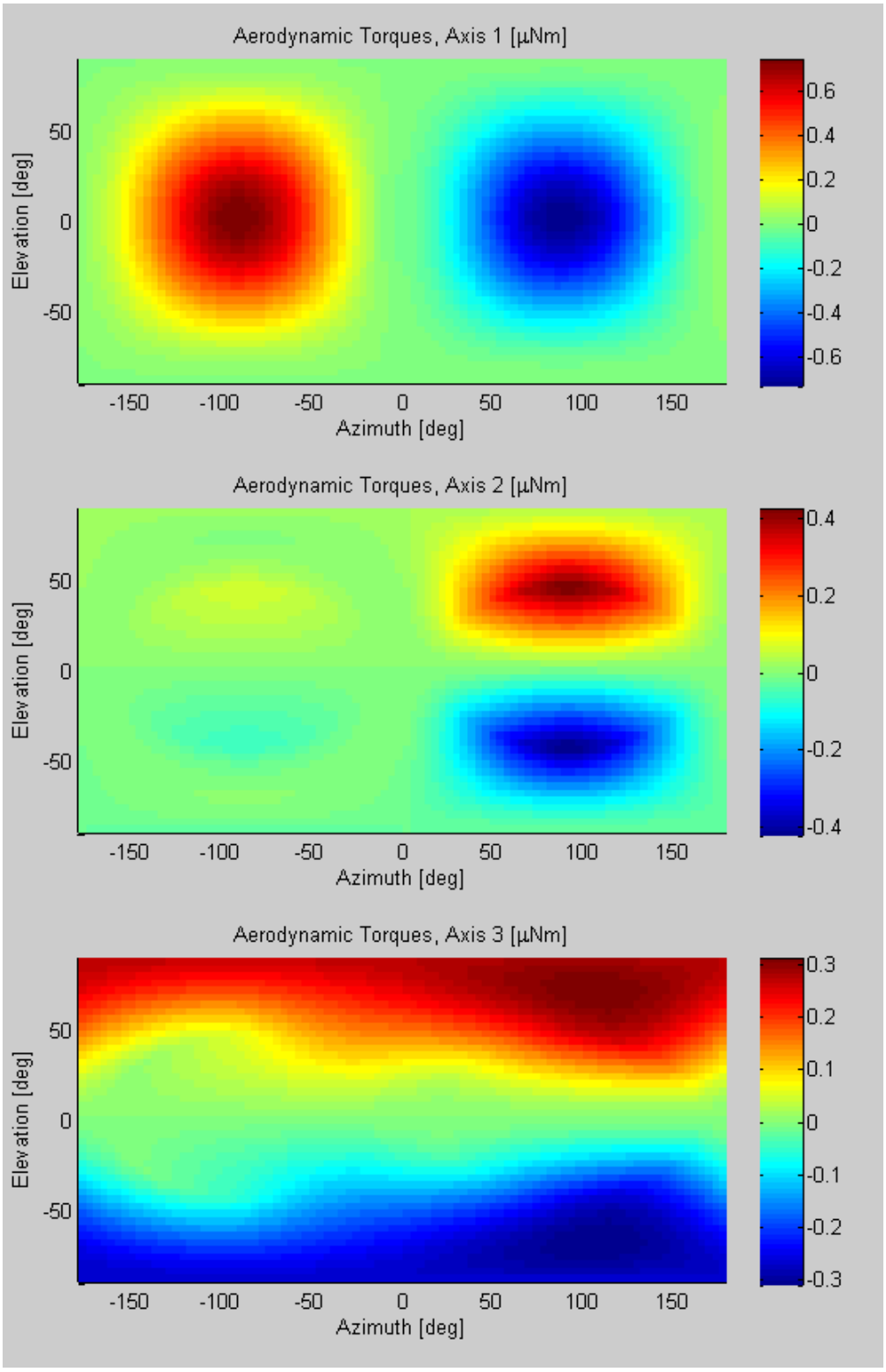

Figure 9.9 - Aerodynamic Torques (RTM Result)

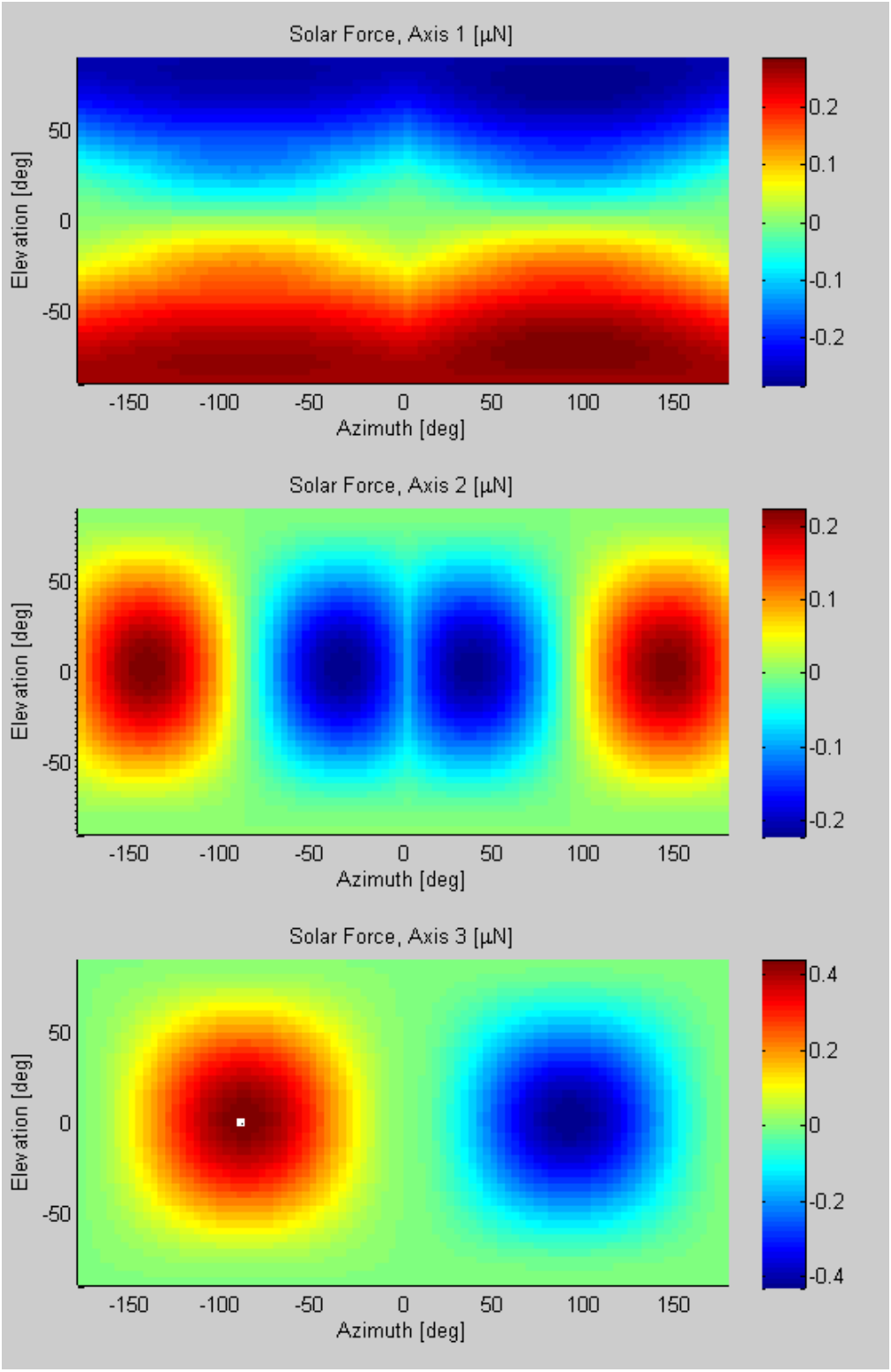

Figure 9.10 - Solar Forces (RTM Result)

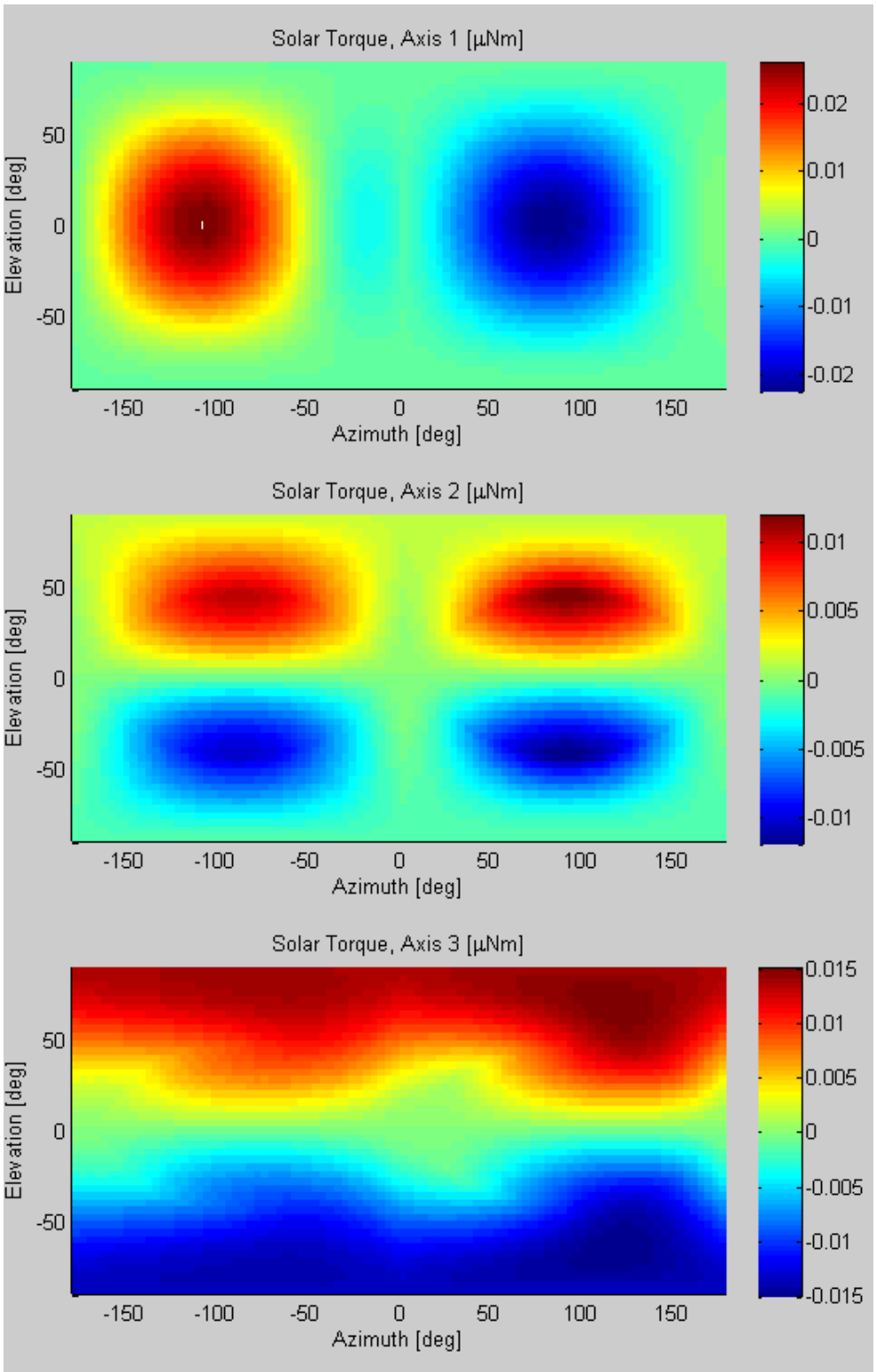

Figure 9.11 - Solar Torques (RTM Result)

It should also be noted that the solar panel model used here is very simple and used to demonstrate that this type of information can be easily extracted from the RTM calculation with very little additional computation.

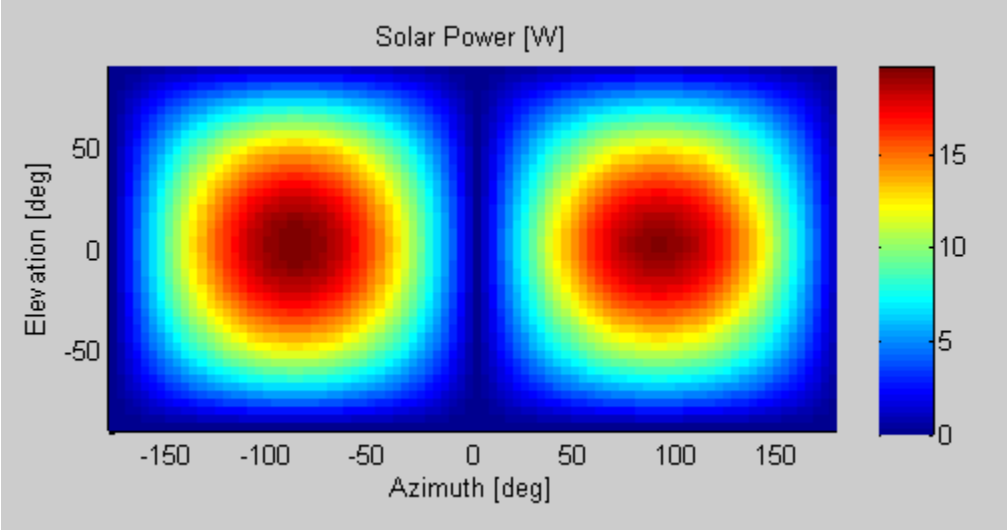

Figure 9.12 - Solar Power (RTM Result)

CHAPTER 10

SIMULATION

10.1 ORBITAL PROPAGATION

No thesis in science or engineering is complete without referencing Newton's laws. The fundamental propagator for the position and velocity states is Newton's second law:

$$\boldsymbol{F} = m\boldsymbol{a} \tag{10.1}$$

The full positional state differential equations are then:

$$\begin{bmatrix} \dot{\boldsymbol{r}}^{\boldsymbol{I}} \\ \dot{\boldsymbol{v}}^{\boldsymbol{I}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^{\boldsymbol{I}} \\ \boldsymbol{F}^{\boldsymbol{I}}/m \end{bmatrix} \tag{10.2}$$

The force from spherical gravity is:

$$\boldsymbol{F}_{\boldsymbol{g}}^{\boldsymbol{I}} = -\frac{\mu\, m\, \boldsymbol{r}^{\boldsymbol{I}}}{r^3} \tag{10.3}$$

The positional state equations are propagated forward in time using MATLAB's *ode45* differential equator solver, which is a variable timestep explicit Runge-Kutta solver. All forces except gravity are assumed to be constant in magnitude and inertial direction over the course of one simulated timestep (typically 0.25 sec).

10.2 ATTITUDE PROPAGATION

Propagation of the spacecraft attitude is accomplished by numeric integration of the quaternion dynamical equations and the Euler differential equations of attitude dynamics.

$$\dot{\boldsymbol{q}} = \Omega'(\boldsymbol{\omega})\boldsymbol{q} \tag{10.4}$$

$$\Omega'(\boldsymbol{\omega}) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{10.5}$$

$$\dot{\boldsymbol{\omega}} = J^{-1}[-\boldsymbol{\omega} \times (J\boldsymbol{\omega}) + \boldsymbol{\tau}^{\mathcal{B}}] \tag{10.6}$$

The attitude state equations are propagated forward in time using MATLAB's *ode45* differential equator solver. All torques are assumed to be constant in magnitude and body-axis direction over the course of one simulated timestep (typically 0.25 sec).

10.3 Environment Models

10.3.1 Atmospheric Density

In order to compute drag forces and torques using the RTM method described in Chapter 9, it is necessary to know the average atmospheric density. While there exist many complicated models, a relatively simple one is used here because the atmospheric effects enter as disturbance forces and torques and are not considered explicitly in either the control or estimation. The model used [27] computes the average density as a function of altitude, the solar radio flux index (F10.7), and the geomagnetic index (Ap). The density is calculated using an exponential model with a variable scale height based on average molecular composition and solar activity. The model is valid only for orbits in LEO at 500km or lower.

The F10.7 index measures the solar radio flux at a wavelength of 10.7cm and is considered to be a good indicator of overall solar activity. It is measured in solar flux units (1 SFU = 10^{-22} W m^{-2} Hz^{-1}). The F10.7 index ranges between approximately 70 SFU at solar minimum to 220 SFU or higher at solar maximum.

The Ap geomagnetic index is a somewhat subjective index which is based on the relative magnetic disturbances measured over a worldwide network which are then translated to a global disturbance level, measured in nT. Average values for Ap are less than 20 nT, but it can reach up to 400 nT during magnetic storms.

A set of predicted F10.7 and Ap values is given in [28]. By way of example, the month of January 2014 is used to give some bounding cases on the model:

Table 10.1 - Predicted F10.7 and Ap Indices for January 2014 [28]

F10.7 Index Percentile (Average)			A_p Index Percentile (Average)		
95%	50%	5%	95%	50%	5%
162.0	117.4	84.8	23.3	16.3	10.9

The units of each intermediate quantity are listed for clarity. The mean asymptotic temperature of the exosphere is approximated as:

$$T_{atm}\,[Kelvin] = 900 + 2.5 * (F10.7 - 70) + 1.5A_p \tag{10.7}$$

The mean molecular mass is approximated as:

$$m_{atm} = 27 - 0.012(h - 200) \tag{10.8}$$

The scale height is then given as:

$$H_{atm}[km] = \frac{T_{atm}}{m_{atm}} \tag{10.9}$$

Finally, the density can be computed:

$$\rho\left[\frac{kg}{m^3}\right] = 6 * 10^{-10} e^{-\frac{h-175}{H}} \tag{10.10}$$

Using this model and the predicted bounds on the F10.7 and Ap indices for January 2014, the density at LEO altitudes can be calculated. This is plotted in Figure 10.1.

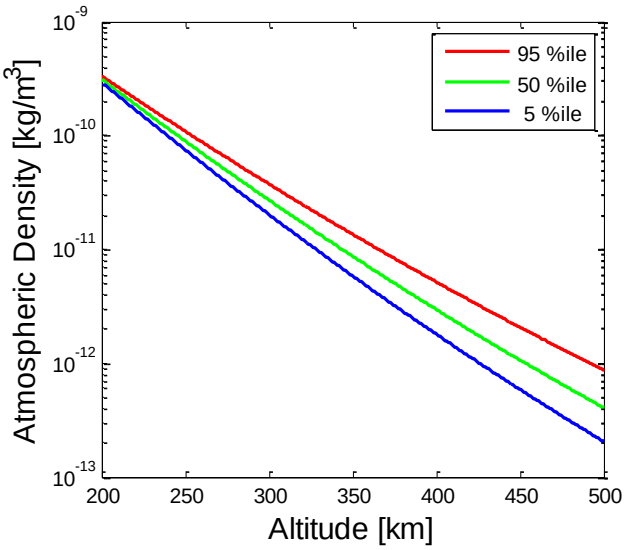

Figure 10.1 - Average Atmospheric Density vs. Altitude (Modeled)

Clearly, the density predicted by this model can vary by a factor of up to five at high altitude based on the solar and geomagnetic activity. The variability of atmospheric density is one of the primary challenges in modeling atmospheric drag.

10.3.2 Magnetic Field

The IGRF-11 magnetic field model is used for the generation of the true magnetic field. The specific implemented used is the MATLAB function '*igrf11magm*'. Based on [24], the IGRF-11 model has an rms error of about 10 nT plus an additional 20 nT/year for extrapolated years beyond 2010. Because the mission is being expected to launch no earlier than 2014, the expected rms error is therefore 90 nT. Additional sources of error suggested in [29] such as contributions from the Earth's crust (as opposed to the core) as well as variations on short time scales may contribute another 60 nT rms globally. If the error is assumed to be Gaussian and zero-mean, these rms noise values are also the standard deviations. If the magnetic field vector at some instant as produced by the IGRF-11 model is denoted as $\widetilde{\boldsymbol{B}}_{\boldsymbol{m}}$, then the simulated truth model for the magnetic field is given by:

$$\boldsymbol{B}_{\boldsymbol{m}} = \widetilde{\boldsymbol{B}}_{\boldsymbol{m}} + \boldsymbol{\eta}_{\boldsymbol{m}} \tag{10.11}$$

Where $\boldsymbol{\eta}_{\boldsymbol{m}}$ is zero-mean Gaussian white-noise with a spectral densities of $\sigma_m^2 I_{3x3}$. σ_m is assumed to have a value of 150 nT. To speed up computation, this was implemented as a 2D lookup table with interpolation over latitude and longitude with data points evenly spaced every 1°, or about 64,000

points. This is reasonable because the orbit is circular (so the height can be easily approximated to high accuracy for a given latitude and longitude). Informal testing shows that this causes about 0.5% error on average. Most simulations will run for no longer than 300 orbits or about 20 days. Because the parameters of the IGRF model vary over a 5 year timespan, the maximum simulation time is less than 1% of the time over which the model varies, which is an acceptable amount of an error given the level of accuracy used in other parts of the simulation.

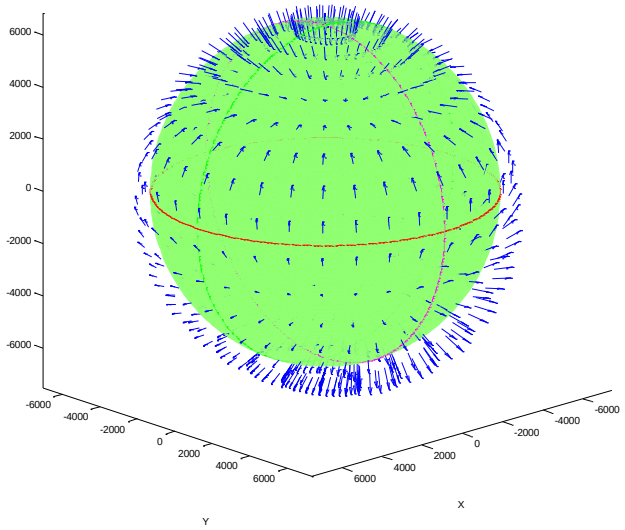

Figure 10.2 - Magnetic Field Vectors at 500km Orbital Radius

10.3.3 Sun Model

The sun model used in the simulation is derived from the Astronomical Almanac [25]. This reference explains the method in great detail and need not be repeated here. The solar flux is assumed to be the average value of 1366 W/m^2. For simplicity, the truth model is also assumed to be given exactly by this method. Thus,

$$\boldsymbol{S}_v = \tilde{\boldsymbol{S}}_v \tag{10.12}$$

The sun's position is not subject to much uncertainty, so it's quite reasonable to assume that the spacecraft will have an onboard model of the sun's position which exceeds the accuracy of the other models, especially the magnetic field.

10.4 OTHER DISTURBANCES

10.4.1 Gravity Gradient

For a rigid body in orbit around a body with uniform spherical gravity, the gravity gradient torque is:

$$\boldsymbol{\tau}_{\boldsymbol{gg}}^{\mathcal{B}} = \frac{3\mu}{r^3}\hat{\boldsymbol{e}}_{\boldsymbol{nadir}}^{\boldsymbol{\mathcal{B}}} \times J\hat{\boldsymbol{e}}_{\boldsymbol{nadir}}^{\boldsymbol{\mathcal{B}}} \tag{10.13}$$

$\boldsymbol{e}_{\boldsymbol{nadir}}^{\boldsymbol{\mathcal{B}}}$ is a unit vector in the body frame pointing toward local nadir. In the LVLH frame, nadir is always located in the –X direction, so the body-frame nadir direction can be easily computed as:

$$\hat{\boldsymbol{e}}_{\boldsymbol{nadir}}^{\boldsymbol{\mathcal{B}}} = R^{\mathcal{BL}}\begin{bmatrix}-1\\0\\0\end{bmatrix} \tag{10.14}$$

10.4.2 Higher Order Earth Gravity

The EGM2008 gravity model [30] was used to compute the forces resulting from non-spherical gravity. Terms up to degree and order 20 were used. The MATLAB function '*gravitysphericalharmonic*' was used for this purpose.

10.4.3 Aerodynamic

The aerodynamic modeling using the Ray Tracing Method is discussed extensively in Chapter 9. This method is used to generate a 2D lookup table and the spacecraft attitude at each timestep is used to interpolate the aerodynamic forces and torques on the spacecraft. The flow vector is the LVLH Y direction expressed in the body frame.

10.4.4 Solar

The aerodynamic modeling using the Ray Tracing Method is discussed extensively in Chapter 9. This method is used to generate a 2D lookup table and the spacecraft attitude at each timestep is used to interpolate the solar forces and torques on the spacecraft. The flow vector is the sun vector expressed in the body frame.

CHAPTER 11

SIMULATION RESULTS

The simulation is useful for generating a wide range of information about the dynamic performance of the system with all of its components properly coupled together. To demonstrate the different aspects of the design discussed in the previous chapters, four test cases have been selected. In all cases the chief begins in an orbit described by the following orbital elements:

$$a = 6878\ km;\ \ e = 1 * 10^{-3};\ \ i = 97°; \omega = 0°; \Omega = 0°; \nu = 0° \qquad (11.1)$$

In all four test cases, the simulation is run at full fidelity, which means that the propagators are non-linear and all disturbance forces and torques are included realistically as discussed in previous sections.

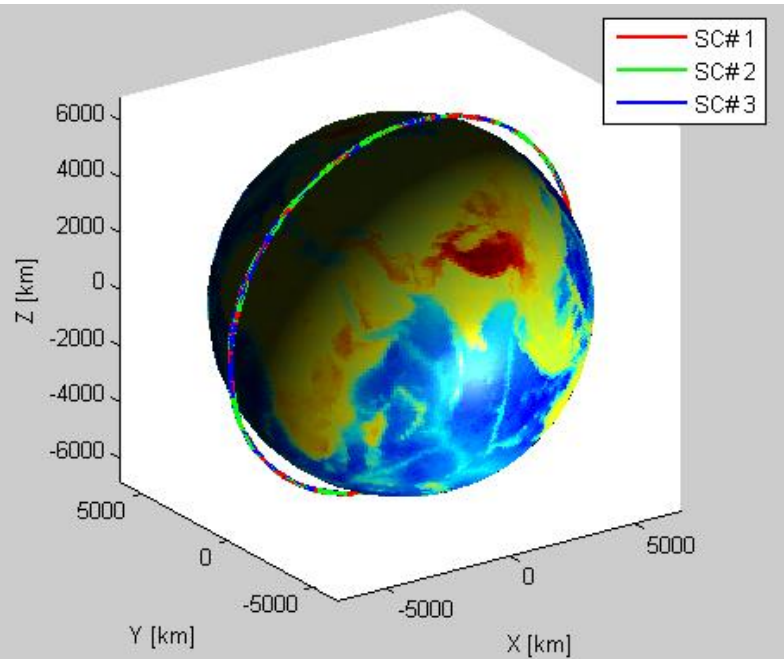

Figure 11.1 - Sun Synchronous Orbit (ECI Frame)

11.2 ATTITUDE CONTROL

In this section, the attitude behavior of a single spacecraft is examined, since the attitude behavior is not coupled between spacecraft.

11.2.1 Case 1: 180° Flip

Initial Conditions:

$$\boldsymbol{q^{BI}} = [-0.99315 \quad 0.0060947 \quad -0.099647 \quad 0.060744]; \ \boldsymbol{\omega} = [0 \quad -\omega \quad 0]^T \tag{11.2}$$

The initial condition for this case is that the spacecraft is oriented correctly in the LVLH frame except for a 180° rotation about the $x^{\mathcal{B}}$ axis, which means the thrusters are facing in the $+Y^{\mathcal{L}}$ direction. This can be considered a very large step response for the attitude controller. It should be noted that the spacecraft is also performing cluster control simultaneously, so the attitude maneuver must also incorporate thrust commands into the mixer.

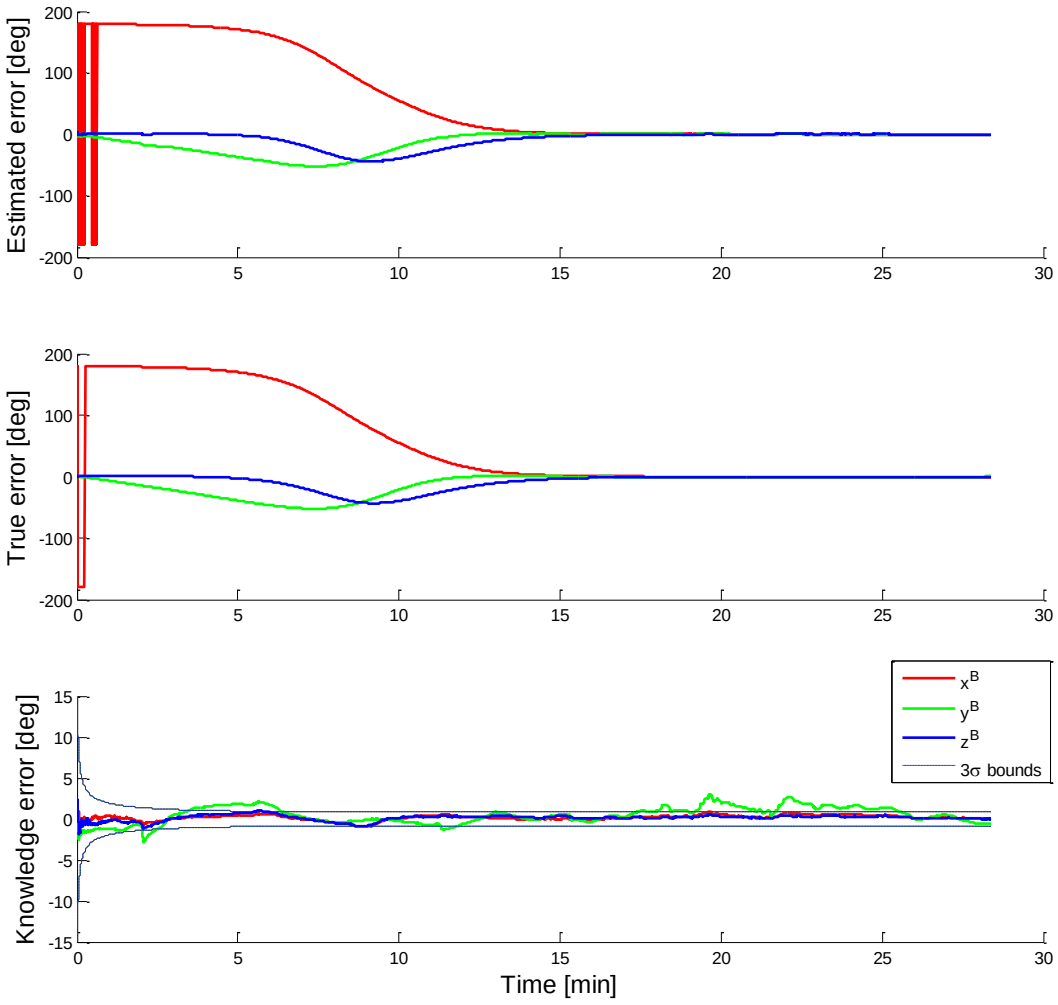

Figure 11.2 - Case 1 State Response

There are several items which are immediately noticeable from this simulated result. The first observation is that the spacecraft did not simply rotate through 180° about the body x axis but instead did some of the turn, did a twist maneuver in the middle (resulting in the non-x axis errors), and then nulled the resulting angular and angular rate errors. This motion also makes intuitive sense because the moment of inertia about the body z axis is much lower, so it should be faster and lower cost to rotate around this axis. The second notable feature is that the error about $Y^{\mathcal{B}}$ is significantly greater than the other two axes. This is because the sun sensor which points toward the sun in the nominal configuration points along the $Y^{\mathcal{B}}$ axis, so it cannot give any information about the error in that axis and must rely on the less accurate magnetometer.

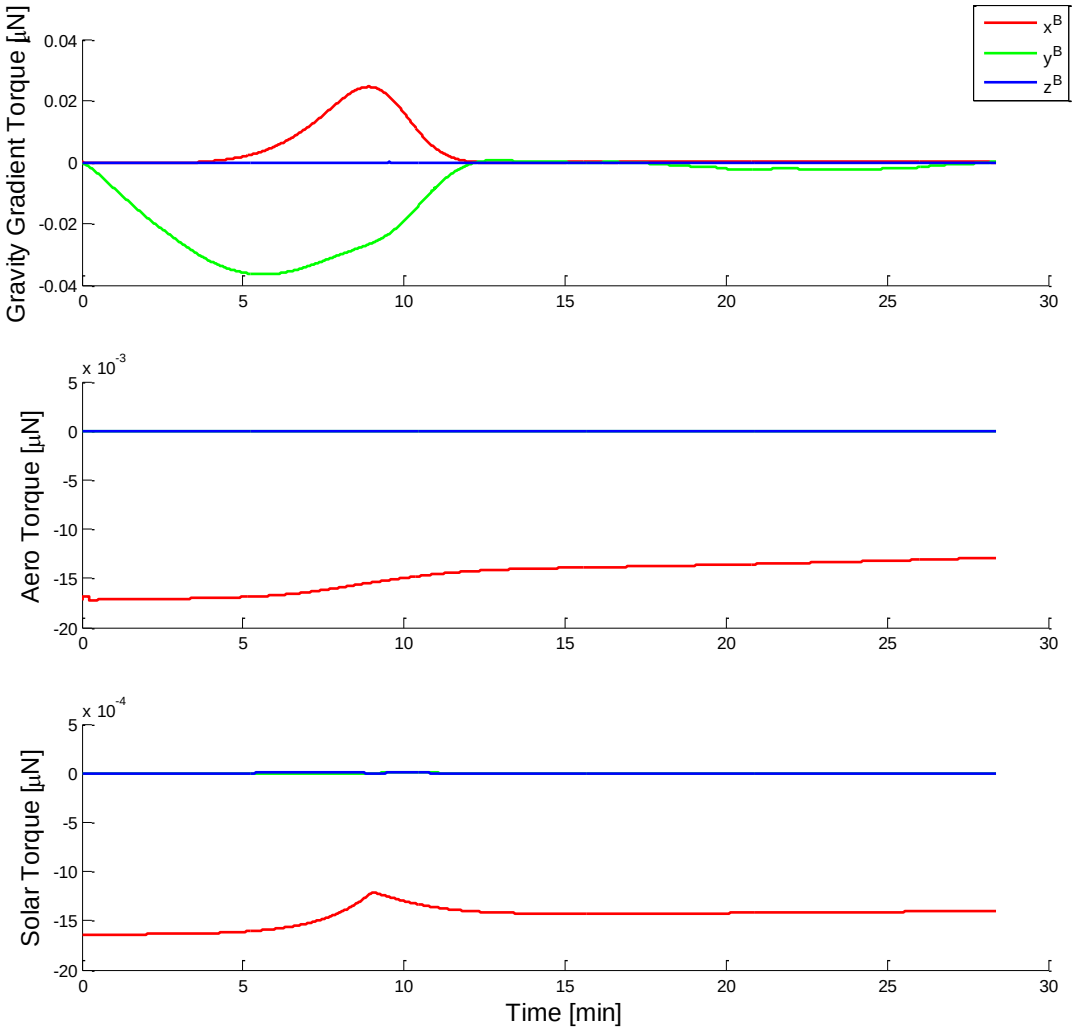

Figure 11.3 - Case 1 Disturbance Torques

Figure 11.3 shows the disturbance torques acting on the spacecraft during this maneuver. There are two main conclusions which can be derived from this. The first is that the attitude disturbances in this orbit are not very impactful on the total solution, providing less than 1% of the force of the actuators. The second conclusion which can be drawn is that all of the models are functioning in response to changing attitude, as that's the only thing currently modeled that could have changed the magnitude of those particular vectors.

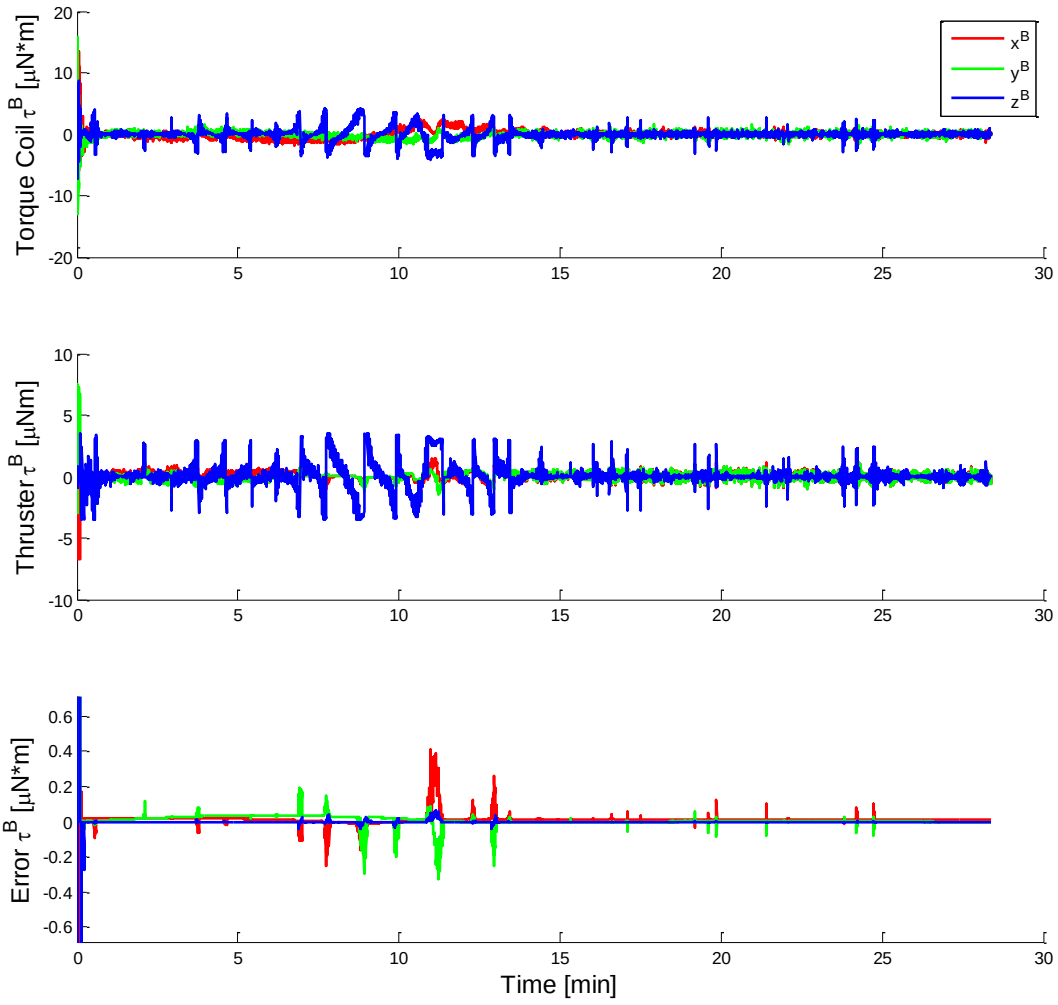

Figure 11.4 - Case 1 Torques

Finally, Figure 11.4 shows the torques from the thrusters and the torque coils. The lowest plot shows the error between the applied torque and the desired torque generated from the attitude control law. Aside from the very high control action demanded in the seconds following the step input, the control allocator is able to produce all of the desired torques to within 0.4 μNm and most more accurately than that.

This maneuver required about 25 minutes to reach the target attitude. A total of 0.025 m/s of delta V was expended over the total maneuver (including deltaV for cluster control), or about 50% average duty cycle over the maneuver.

11.2.2 Case 2: Detumble

The initial condition for this case is that the spacecraft is in a random orientation while spinning at a very high rate of 10° per second with a random initial orientation of the angular velocity vector. The purpose of this maneuver is to simulate deployment, where high angular rates are expected. While rates above 3° per second are not expected in the actual mission, this provides a conservative estimate on time and fuel cost for performing this maneuver.

In this case, the Bdot controller is active until the estimator's covariance indicates 3σ confidence of convergence to within 10°. It is assumed that the sun sensors are not able to take an accurate measurement while the spacecraft is spinning more quickly than 1° per second, so the spacecraft must detumble until the spacecraft is spinning slowly and has the sun in its field of view. Once this occurs, the estimate should rapidly converge to within a couple of degrees. In actual operation, the switching of control modes may be handled manually from the ground to ensure a smooth transition.

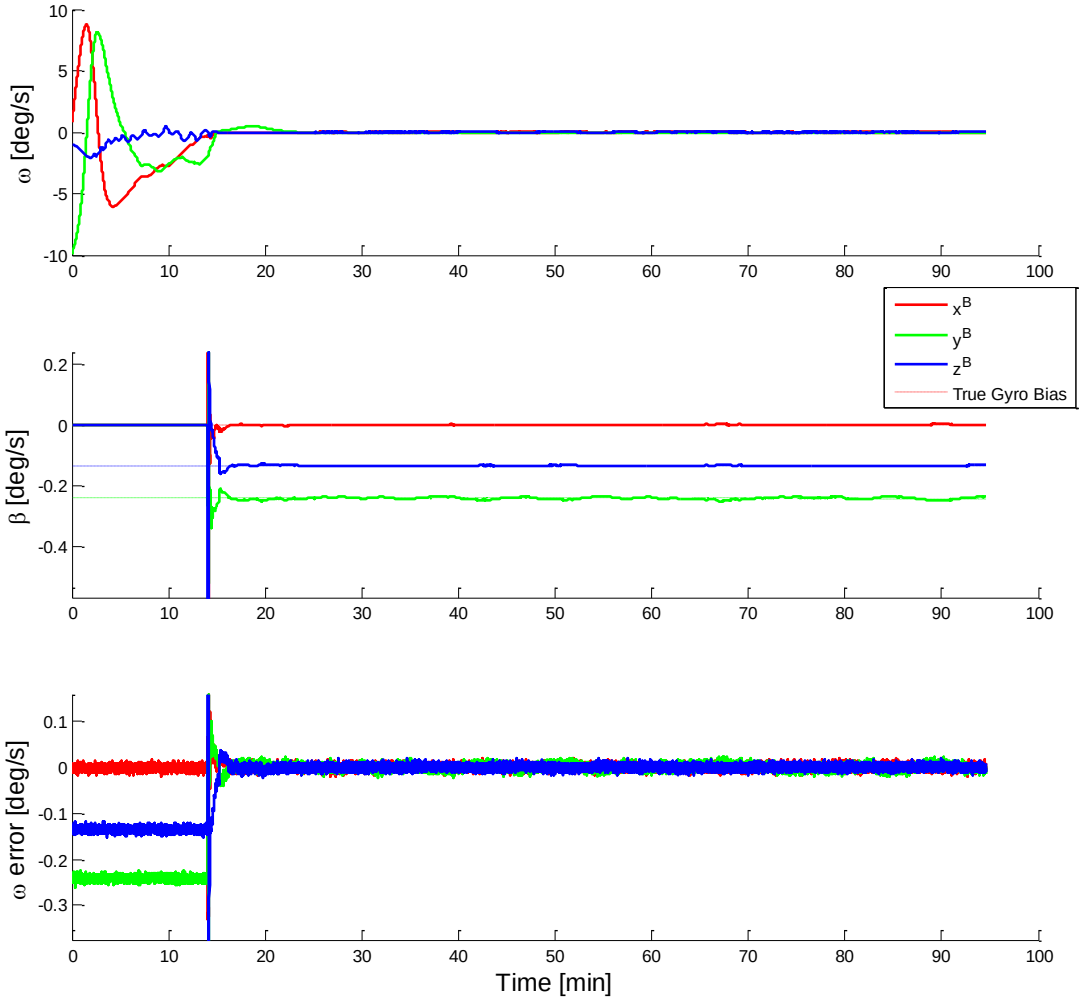

Figure 11.5 - Case 2 Angular Rate Behavior

Using the Detumble control law, the rates in all three axes are damped to within about 1° per second within 15 minutes. The spacecraft gets the first valid sun sensor reading at 14.1 minutes and the estimator converges to the mode transition criteria within one minute. With the sun sensors in view, the estimator is able to estimate the initial gyro biases within 4 minutes of beginning estimation. Note that this does not mean the estimator is fully converged, just sufficiently converged to begin using the Stabilize control law, which relies on the accuracy of the estimated attitude and angular rate. The angular states are plotted in Figure 11.6. The transition to Stabilize mode is quite apparent here.

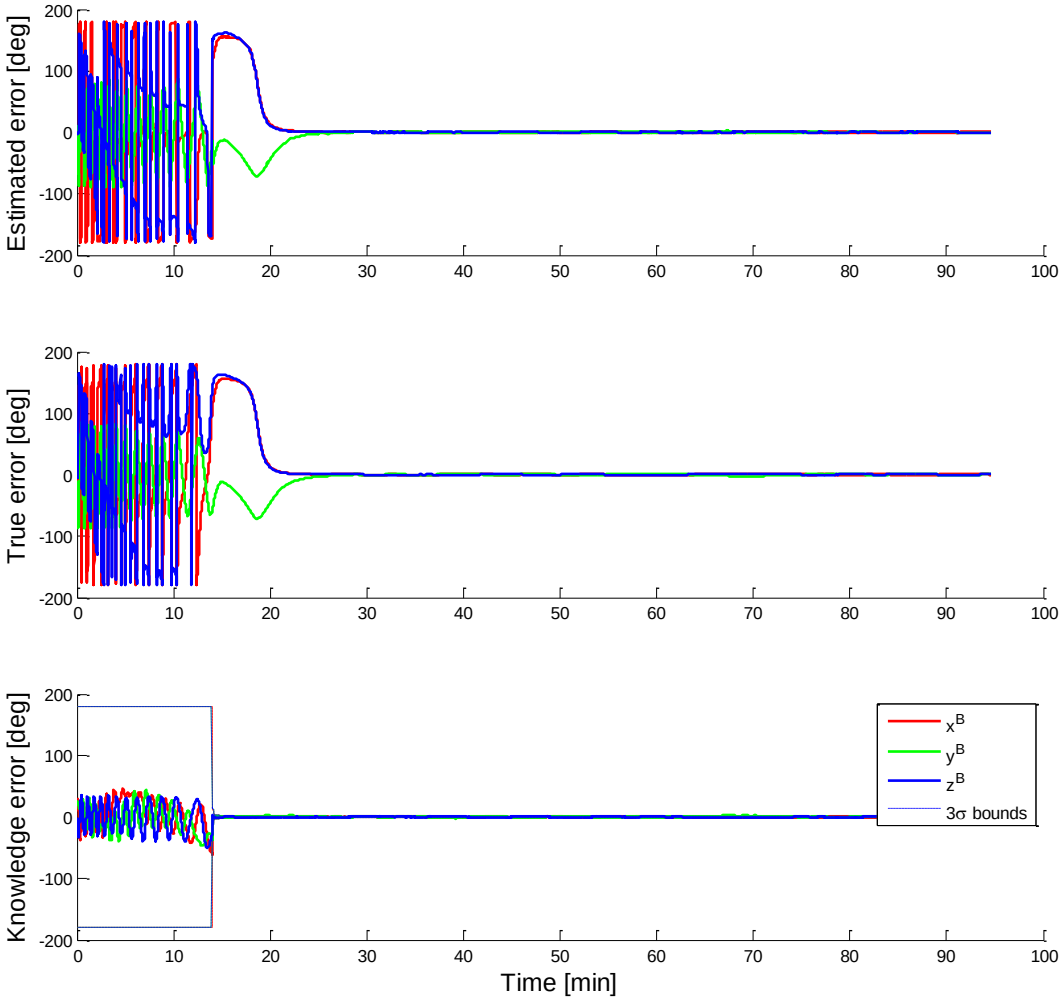

Figure 11.6 - Case 2 Angular State and Error

11.3 Formation Control

In this section, the behavior of the formation will be emphasized, as the attitude performance was addressed directly in the previous section. Because formation control maneuvers take many orbits, a larger timestep of 5 seconds was used to speed up computation. Since 60 second formation control cycles are being assumed, this has no impact on the validity of the simulation for assessment of formation control performance. However, the attitude controller cannot recover

from high angular velocity conditions when a 5 second timestep is used, so all simulations are initialized with LVLH stable attitude. Interestingly, the attitude controller is stable even with a 5 second timestep as long as it keeps a low angular velocity. Note that this does not mean that perfect attitude is being assumed for the whole simulation! The controller must still compensate for disturbance torques generated by the environment and the thrusters.

11.3.1 Case 3: Formation Resizing

This case is intended to capture the fuel cost required to reconfigure the cluster using the current control law and nested allocation setup. Because one of the major functions of the mission is to demonstrate a variety of missions, this type of maneuver will be fairly common. The initial conditions for this case are:

$$\boldsymbol{x_{21}}(t_0) = [0\,m \quad 0\,m \quad 8000\,m \quad 0\,m/s \quad 0\,m/s \quad 0\,m/s]^T \tag{11.3}$$

$$\boldsymbol{x_{31}}(t_0) = [0\,m \quad 0\,m \quad -8000\,m \quad 0\,m/s \quad 0\,m/s \quad 0\,m/s]^T \tag{11.4}$$

The desired final condition is the reference 'string of pearls' configuration.

$$\boldsymbol{r_{SOP}} = \begin{bmatrix} 0\text{ m} \\ 10{,}000\text{ m} \\ 0_{5x1} \\ -10{,}000\text{ m} \\ 0_{4x1} \end{bmatrix} \tag{11.5}$$

This case was simulated for 15 orbits, or about 1 day. The reconfiguration completes in roughly 5 orbits, followed by 10 orbits of stationkeeping, which is used to estimate steady-state fuel consumption using this control scheme.

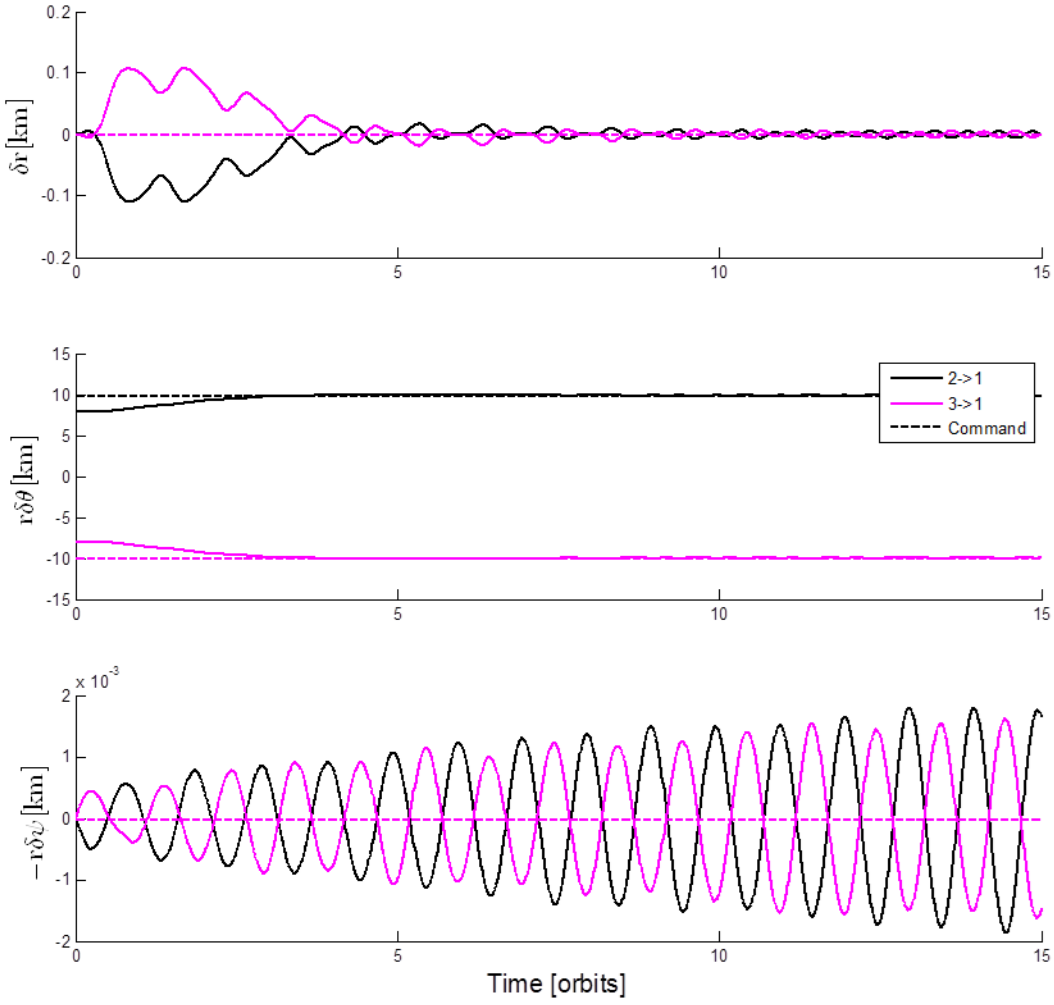

Figure 11.7 - Case 3 State History

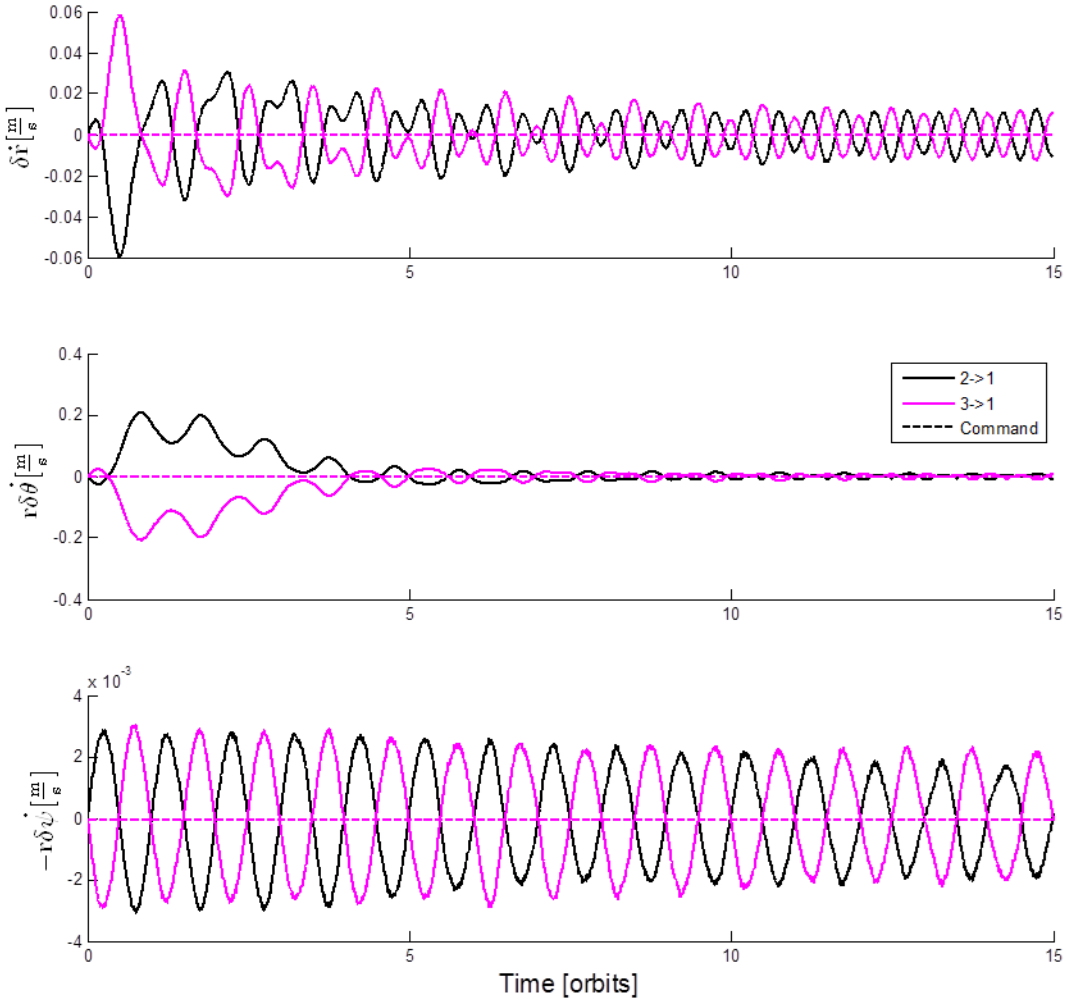

Figure 11.8 - Case 3 Velocity State History

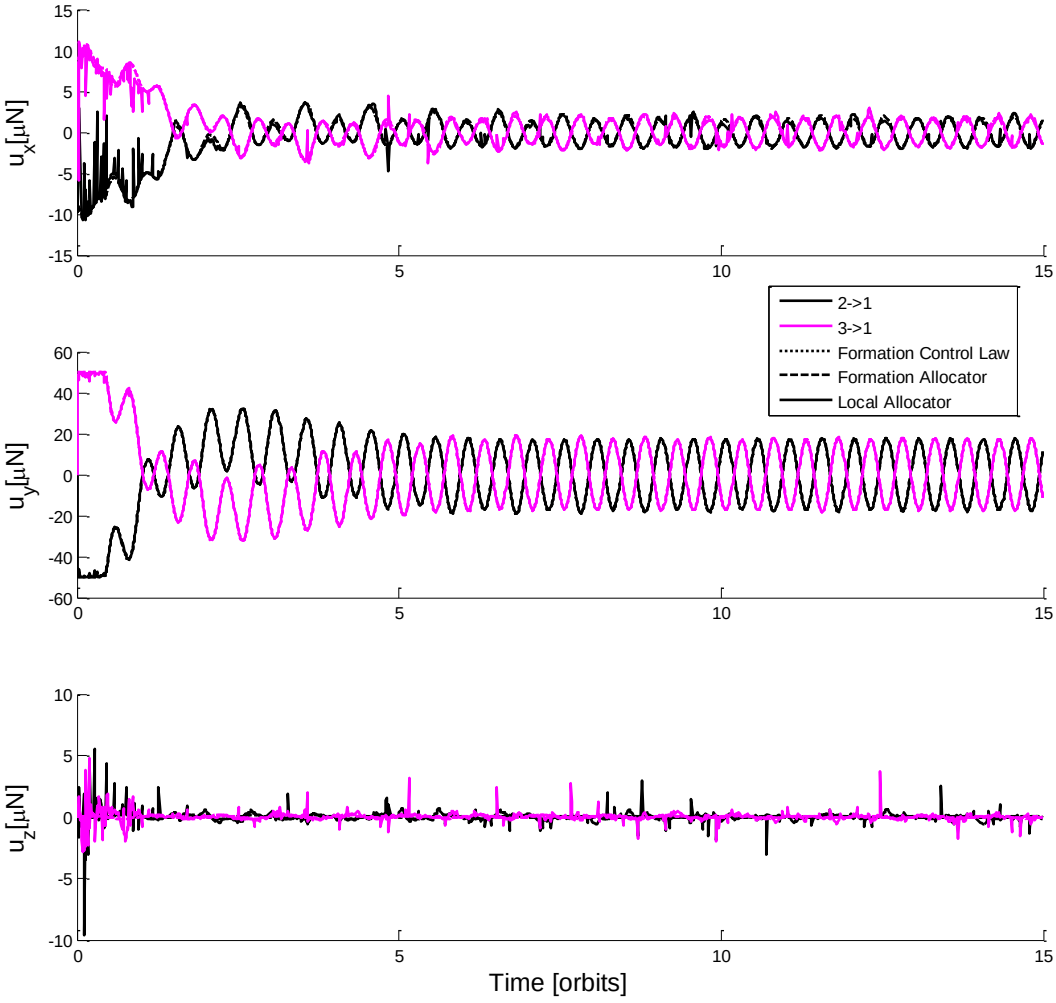

Figure 11.9 - Case 3 Relative Force History

One useful way to measure the total formation error is by using the LQR weighting matrices to define a weighted error:

$$E_W = \sqrt{(\boldsymbol{X_{rel}} - \boldsymbol{r})^T Q_{LQR} (\boldsymbol{X_{rel}} - \boldsymbol{r})} \tag{11.6}$$

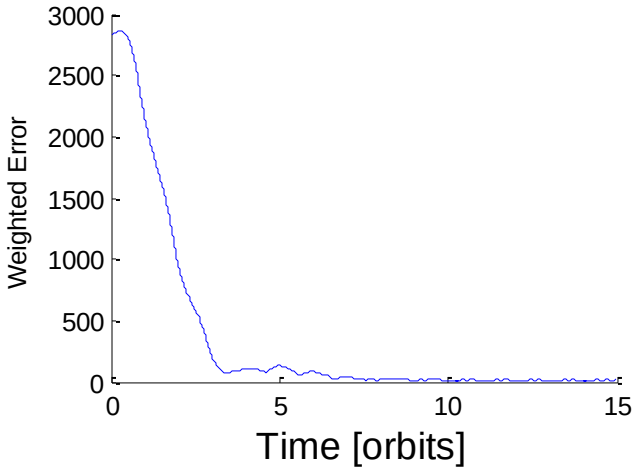

Figure 11.10 - Case 3 Weighted Error

As designed, the weighted error is driven to very nearly zero.

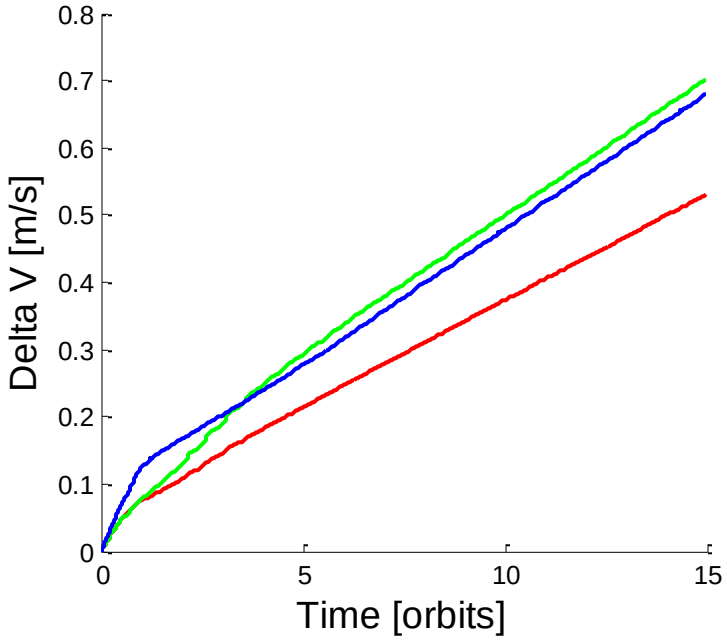

Figure 11.11 - Case 3 Delta V

However, the very good performance does come with a fuel cost. The fuel cost for the maneuver is approximately 0.25 m/s on average for each spacecraft and takes 5 orbits, or roughly 8 hours to complete. The stationkeeping fuel cost for this scenario is about 0.6 m/s per day. Assuming a specific impulse of 2000 sec, this means the fuel budget for stationkeeping and attitude maintenance for 1 year is approximately 45 grams, or 1.1% of the expected satellite mass.

11.3.2 Case 4: P-POD Ejection

This case tests the ability of the formation to re-convene after the large dispersal event which may occur upon P-POD ejection, which may impart velocities up to 1.5 m/s. For this case, the initial conditions are given with respect to the P-POD ejector device, which is assumed to be in a perfectly circular orbit. Because the orientation of the P-POD is unknown, the spacecraft are assumed to be ejected in different directions, which is a worst-case scenario. The initial conditions are given in curvilinear LVLH coordinates:

$$\boldsymbol{x_1}(t_0) = [0\,m \quad 0\,m \quad 0\,m \quad 1\,m/s \quad 0\,m/s \quad 0.5\,m/s]^T \tag{11.7}$$

$$\boldsymbol{x_2}(t_0) = [0\,m \quad 0\,m \quad 0\,m \quad -1\,m/s \quad 0\,m/s \quad -0.5\,m/s]^T \tag{11.8}$$

$$\boldsymbol{x_3}(t_0) = [0\,m \quad 0\,m \quad 0\,m \quad 0\,m/s \quad 1\,m/s \quad 0\,m/s]^T \tag{11.9}$$

The desired final condition is the reference 'string of pearls' configuration.

Essentially, these conditions translate into spacecraft 1 and 2 being in orbits with opposing differential inclination, and arguments of perigee, but with the same orbital energy. Because spacecraft 3 has excess alongtrack velocity, it actually has a higher semimajor axis and therefore will have a secular drift with respect to the other two spacecraft. Additionally, because it has no out-of-plane differential velocity, spacecraft 3 will also be in a different inclination than either of the other spacecraft. Because the maximum thrust is so low, the time it takes to arrest this small secular drift rate will yield a large alongtrack error. The system response to these conditions was simulated over a time period of 200 orbits, or roughly two weeks.

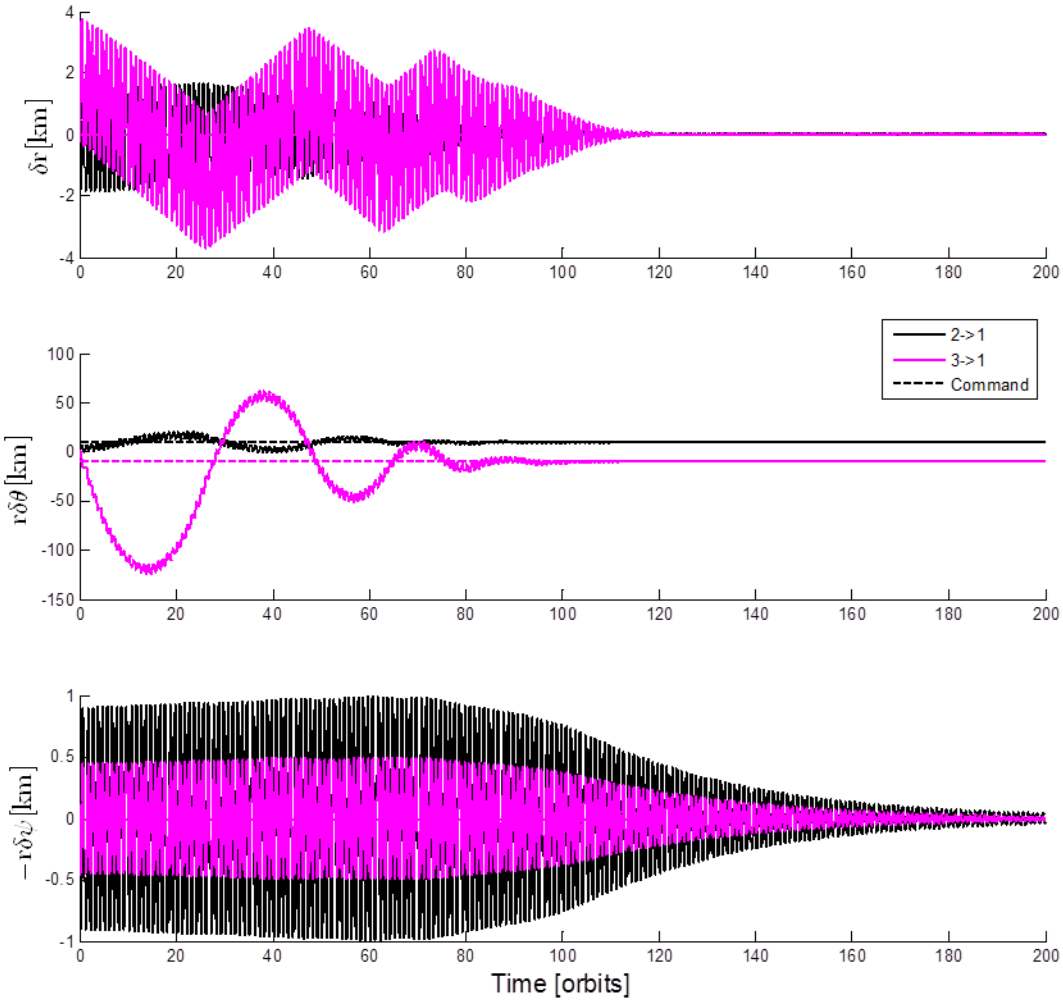

Figure 11.12 - Case 4 Position Relative Position State History

As expected, spacecraft 3 has a very large alongtrack displacement. Note that this error would continue to grow in the absence of control. Also, note that the time scale is in orbits, with each orbit having a length of about 95 minutes, so this is quite a long maneuver. One of the most interesting results from this case is that the curvilinear coordinates allow this motion to be controlled accurately using a linear feedback law with a very low-thrust propulsion system when the underlying dynamics are nonlinear.

The out-of-plane motion is initially undamped because the control effort is being directed to correct the more problematic in-plane states. In fact, there is some growth in the out-of-plane relative

position which peaks around 70 orbits. This is because the in-plane states are much further from their reference values, so the control is directed toward fixing those errors first. Once the in-plane error is on the same order of magnitude as the out of plane errors (around 100 orbits), the out of plane error again begins to decrease more rapidly. It is difficult to tell from the plot, but by the end of the maneuver the in-plane position error has been reduced to about 20m from the reference and the out-of-plane error has been reduced to about 50m from reference.

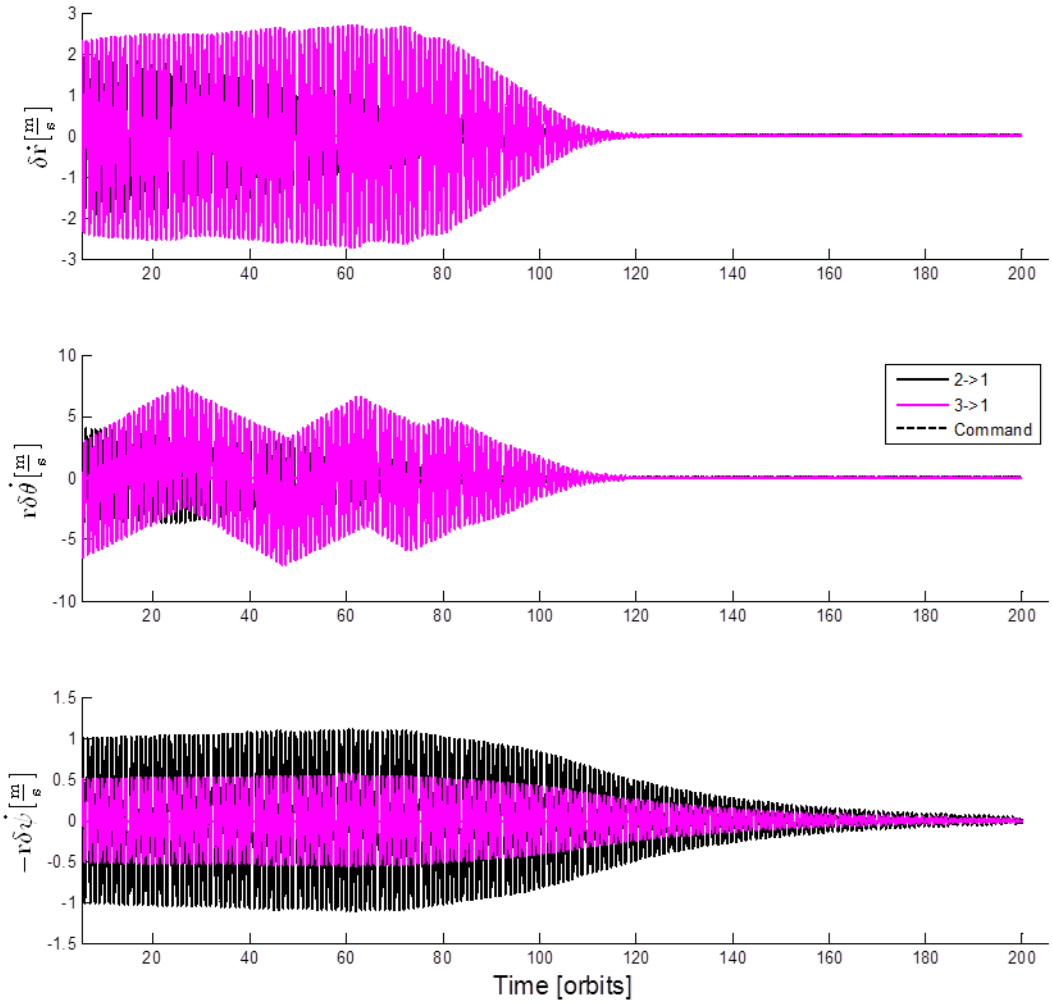

Figure 11.13 - Case 4 Velocity State History

Figure 11.14 shows the relative forces between the spacecraft over the course of the maneuver at various points through the control algorithm. The first step is the formation control law (which

includes the simple scaling for saturation). The second step is the relative acceleration resulting from the formation-level allocator. The final step is the relative acceleration which results from the actual actuator commands which are generated after the local allocator. It is difficult to discern these three stages along the control cycle from these plots due to the time scaling. Figure 11.15 gives a closer look at a 1.5 orbit section where it is possible to observe where the local allocator has deviated from the command from the formation allocator. This is because the local allocator prefers reducing attitude error, so the thrusters may be used to generate torques. The actual thrust commands in each spacecraft's body frame are shown in Figure 11.16 for the same time segment.

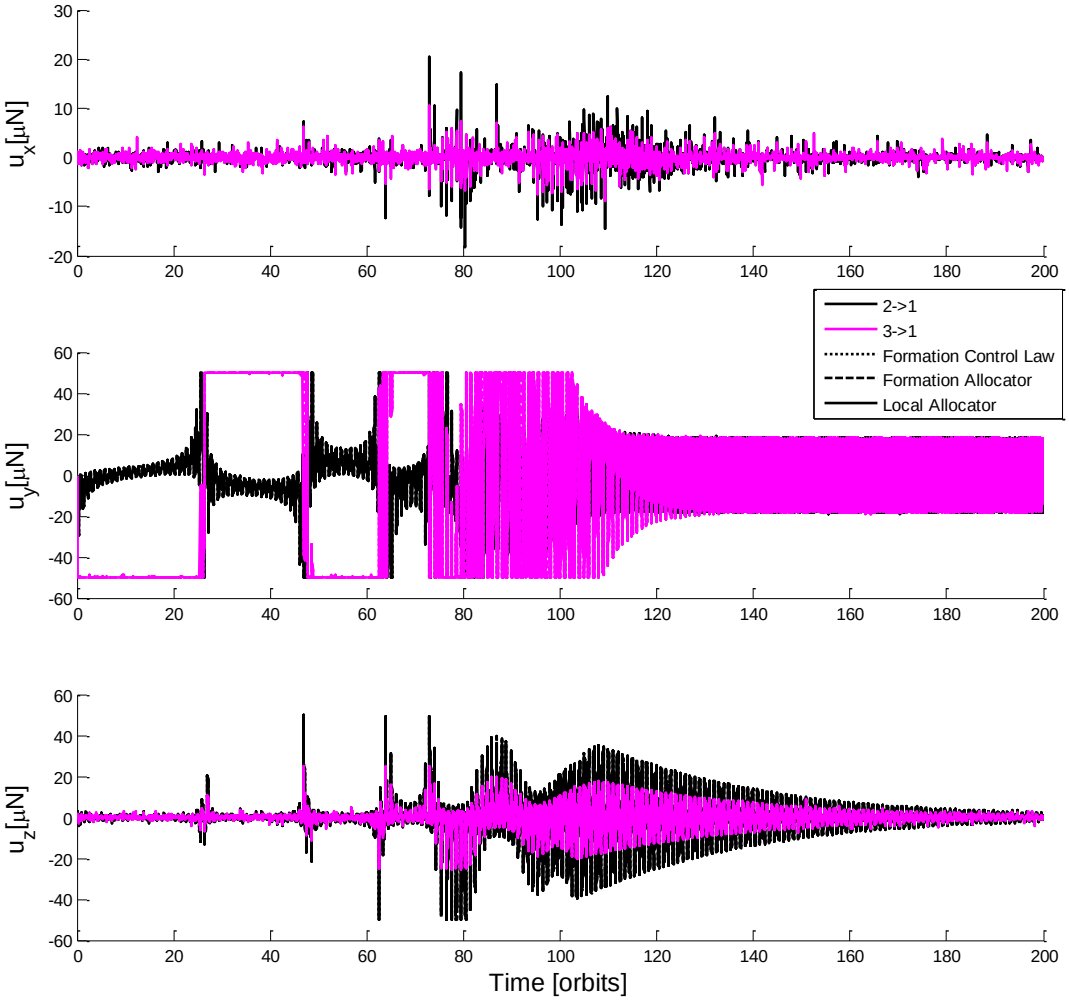

Figure 11.14 – Case 4 Relative Force Plot

It is interesting to observe the steady state non-zero thrust which is being performed in the LVLH Y direction. This is a result of the LQR control law attempting to fight the perceived disturbances resulting from the nonlinear dynamics and disturbance forces, especially higher order gravity.

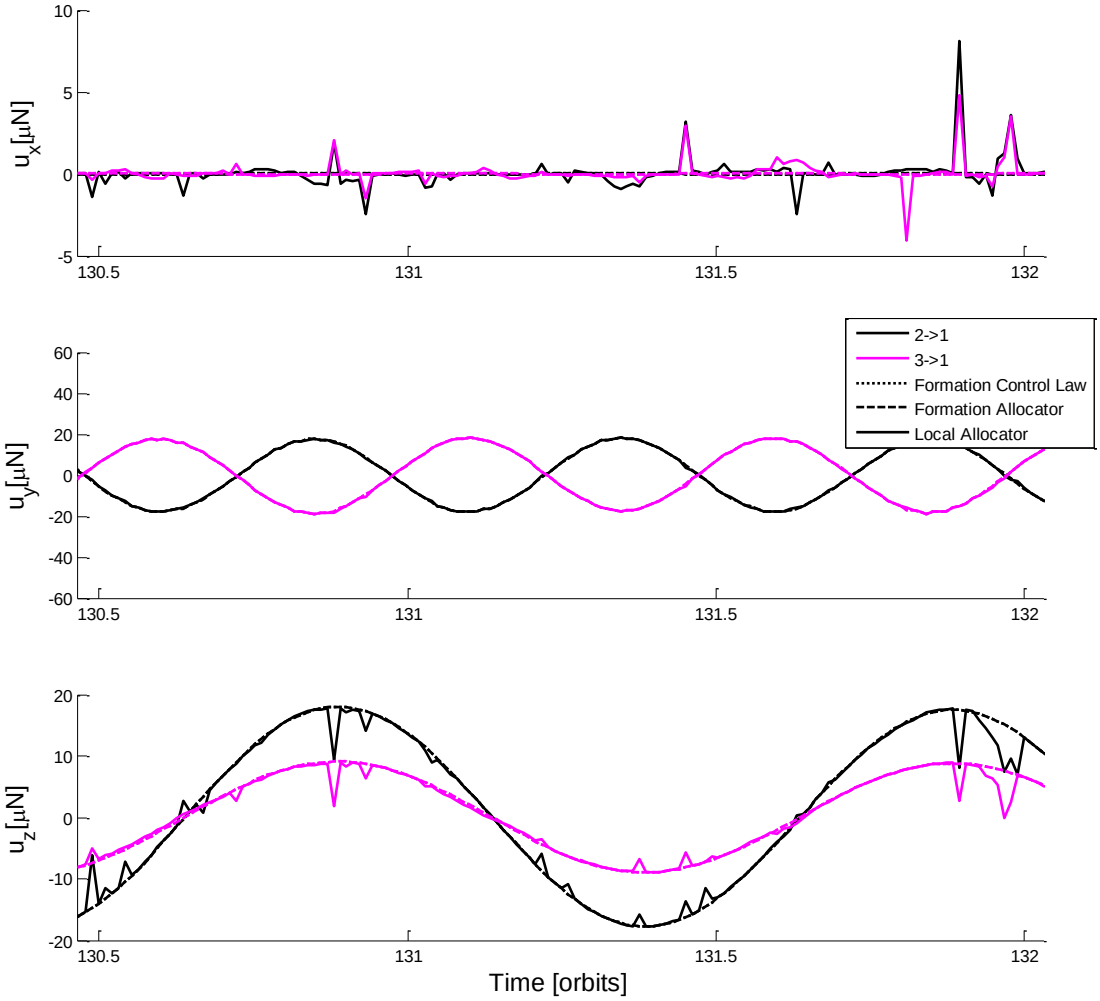

Figure 11.15 – Case 4 Relative Force Plot Zoomed In

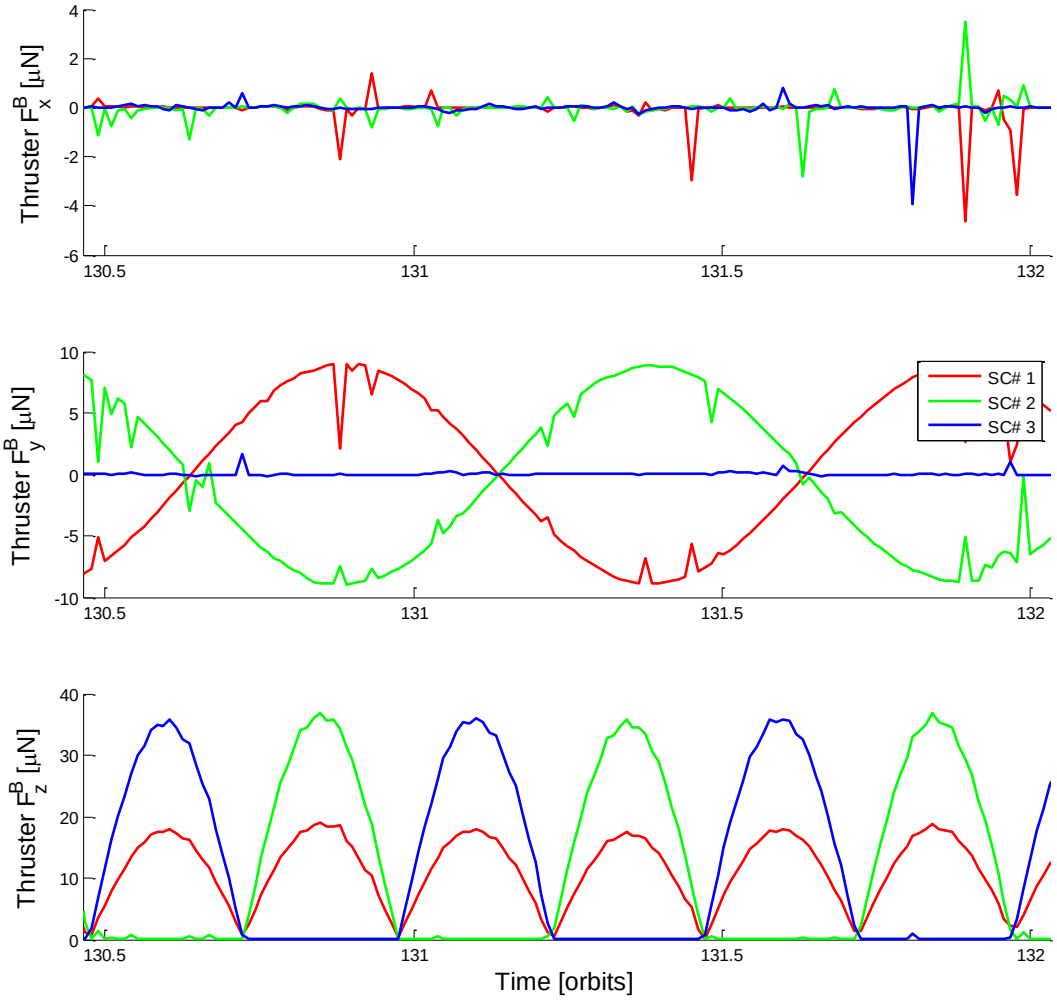

Figure 11.16 - Case 4 Body Frame Force Commands after Local Allocation

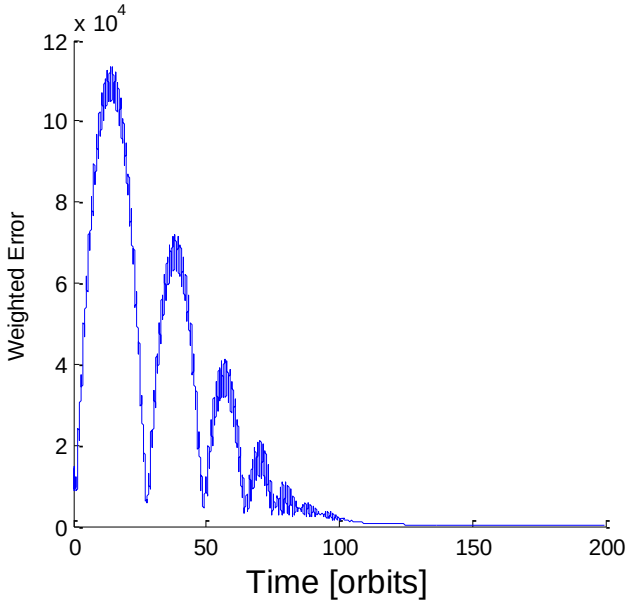

Figure 11.17 - Case 4 Weighted Formation Error

As designed, Figure 11.17 shows that the weighted formation error is decreasing with time, approaching the desired reference configuration. One of the interesting metrics to examine for a maneuver like this is fuel expenditure. The cumulative delta V is shown in Figure 11.18. This shows that this maneuver can be completed with 12 m/s or less of Delta V in each spacecraft.

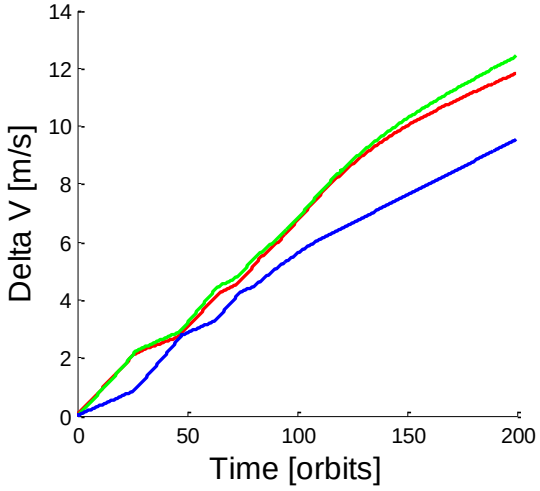

Figure 11.18 - Case 4 Cumulative Delta V

CHAPTER 12

CONCLUSION

12.1 SUMMARY

The overall objective of this thesis was to design and analyze the attitude and formation control algorithms for a 3 satellite Cubesat mission. To evaluate the performance of these algorithms, a virtual environment was created to simulate the space environment and its interaction with the spacecraft's sensors and actuators.

Due to the processing limits posed by the Cubesat platform, a straightforward LQR feedback control law based on the relative states between spacecraft is used to generate a generalized set of relative accelerations between spacecraft. A formation-level allocator then distributes generalized thrust commands to the individual spacecraft in order to obtain the desired relative acceleration (or as close to it as possible). This formation-level allocator indirectly considers thruster saturation by imposing constraints on the generalized force command based on the state of the spacecraft at the time of the formation control calculation.

A separate attitude control law is used to generate a set of desired torques. A local-level allocator ('mixer') explicitly considers the saturation constraints of each actuator and generates the set of actuator commands which best match the desired generalized forces and torques. Because the actuator constraints are explicitly considered at the level of the local allocator, the issued commands are guaranteed to be feasible.

A discrete EKF based on sun sensor-magnetometer fusion is used to estimate the attitude and angular rates of each spacecraft.

Additionally, a new intermediate-accuracy, intermediate-complexity method using ray tracing to calculate aerodynamic and solar forces and torques as well as other visibility metrics such as solar power generation was presented.

Finally, the various elements of the simulation are combined to evaluate the performance of the MotherCube mission. Four case studies demonstrate that the control framework is capable of simultaneous three-axis attitude stabilization and formation control. Further, these case studies provided important performance metrics such as fuel use during maneuvers and the time required to complete them.

12.2 FUTURE WORK

The following list provides possible areas of extension to the work presented in this thesis:

- Full state knowledge of the inertial position and velocity of all satellites is assumed to be known to high accuracy before using the LQR control law to generate the generalized relative accelerations. Because the MotherCube mission will use GPS in LEO, this assumption is well-founded. However, such precise knowledge of the inertial state is not guaranteed on all formation missions. One area of further investigation would be the application of this method to less observable formation concepts, such as those which have only inter-spacecraft sensing.

- The fuel use for stationkeeping is relatively high. This is likely because the controller attempts to reject periodic disturbances (which result mostly from higher-order-gravity) resulting in fuel expenditure. One potential way to avoid this effect is by explicitly including higher order gravity (at least J2) in the analytic model [31] and designing a controller with that in mind.

- One concept for reducing the fuel required for formation control is to use differential aerodynamic drag to create relative acceleration between spacecraft [32] [33]. It would be interesting to combine these theories with hierarchal-allocator approach for command generation and the RTM for calculation of the aerodynamic forces and torques.

CHAPTER 13

REFERENCES

[1] C. Mathieu and A. L. Weigel, "Assessing the Flexibility Provided by Fractionated Spacecraft," in *Space 2005*, Long Beach, CA, 2005.

[2] O. Brown and P. Eremenko, "The Value Proposition for Fractionated Space Architectures," in *AIAA Space 2006*, San Jose, CA, 2006.

[3] DARPA, "System F6," [Online]. Available: http://www.darpa.mil/Our_Work/TTO/Programs/System_F6.aspx. [Accessed 22 May 2013].

[4] P. R. Lawson, "The Terrestrial Planet Finder," Pasadena, CA, 2001.

[5] NASA, "Space Science & Technology," 2 April 2004. [Online]. Available: http://www.nasa.gov/missions/science/f_lbt1mirror.html. [Accessed 22 May 2013].

[6] OHB-Sweden, "Prisma Satellites," [Online]. Available: http://www.lsespace.com/?id=16291. [Accessed 22 May 2013].

[7] P. Lozano, M. Martínez-Sánchez and J. M. Lopez-Urdiales, "Electrospray emission from nonwetting flat dielectric surfaces," *Journal of Colloid and Interface Science,* vol. 276, no. 2, pp. 392-399, 2004.

[8] G. Krieger, I. Hajnsek, K. P. Papathanassiou, M. Younis and A. Moreira, "Interferometric Synthetic Aperture Radar (SAR) Missions Employing Formation Flying," in *Proceedings of the IEEE*, 2010.

[9] K. Danzmann, "LISA mission overview," *Advances in Space Research,* vol. 25, no. 6, pp. 1129-1136, 2000.

[10] D. P. Scharf., F. Y. Hadaegh and S. R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," in *American Control Conference*, Denver, CO, 2003.

[11] D. P. Scharf, F. Y. Hadaegh and S. R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control," in *American Control Conference*, Boston, MA, 2004.

[12] L. M. Mailhe and J. J. Guzman, "Initialization and resizing of formation flying using global and local optimization methods," in *IEEE Aerospace Conference*, 2004.

[13] R. Burns, C. A. McLaughlin and M. M. Leitner, "TechSat 21: formation design, control, and simulation," in *IEEE Aerospace Conference*, Big Sky, MT, 2000.

[14] D. Morgan, S.-J. Chung, L. Blackmore, A. Behcet, D. Bayard and F. Y. Hadaegh, "Swarm-Keeping Strategies for Spacecraft Under J2 and Atmospheric Drag Perturbations," *Journal of Guidance, Control, and Dynamics,* vol. 35, no. 5, 2012.

[15] A. Guerman, M. Ovchinnikov, G. Smirnov and S. Trofimov, "Closed Relative Trajectories for Formation Flying with Single-Input Control," *Mathematical Problems in Engineering,* vol. 2012, 2012.

[16] A. Guerman, M. Ovchinnokiv, G. Smirnov and S. Trofimov, "High-precision single-input control of relative motion in spacecraft formation," *Acta Astronautica,* 2013.

[17] M. J. Sidi, Spacecraft Dynamics & Control - A Practical Engineering Approach, Cambridge, MA: Cambridge University Press, 1997.

[18] J. Crassidis and J. Junkins, Optimal Estimation of Dynamics Systems, Boca Raton, FL: CRC Press, 2012.

[19] Analog Devices, "ADIS16488 Data Sheet," Norwood, MA, 2011.

[20] AeroAstro Space Micro, "Medium Sun Sensors: Data Sheet," San Diego, CA.

[21] J. A. Prussing and B. A. Conway, "Orbital Mechanics," 1993.

[22] C. M. Pong, "Autonomous thruster failure recovery for spacecraft formations," Massachusetts Institute of Technology, Cambridge, MA, 2010.

[23] E. Gill, F. de Bruijn and J. How, "Comparative Analysis of Cartesian and Curvilinear Clohessy-Wiltshire Equations," *Journal of Aerospace Engineering, Sciences and Applications,* vol. 3, no. 2, pp. 1-15, 2011.

[24] C. C. Finlay et al., "Internation Geomagnetic Reference Field: the eleventh generation," Lincoln, Nebraska, 2010.

[25] Nautical Almanac Office (U.S.), The Astronomical Almanac, United Kingdom Hydrographic Office, 2013.

[26] S. Rawashdeh, D. Jones, D. Erb, A. Karam and J. Lumpp, "Aerodynamic Attitude Stabilization for a Ram-Facing Cubesat," *American Astronomical Society,* 2009.

[27] The Australian Space Weather Agency, "Satellite Orbital Decay Calculations," Sydney, Australia, 1999.

[28] K. O. Niehuss and J. K. Owens, "Future SOlar Activity Estimates for Use in Prediction of Space Environmental Effects on Spacecraft," Huntsville, AL, 1999.

[29] C. Finlay, "The International Geomagnetic Reference Field: A "Health" Warning," 2010. [Online]. Available: http://www.ngdc.noaa.gov/IAGA/vmod/igrfhw.html. [Accessed 20 May 2013].

[30] N. K. Pavlis, S. A. Holmes, S. C. Kenyon and J. K. Factor, "The development and evaluation of the Earth Gravitational Model 2008 (EGM2008)," *Journal of Geophysical Research: Solid Earth,* vol. 117, no. B4, 2012.

[31] S. A. Schweighart and R. J. Sedwick, "High Fidelity Linearized J2 Model for Satellite Formation Flight," *Journal of Guidance Control and Dynamics,* 2002.

[32] C. L. Leonard, W. M. Hollister and E. V. Bergmann, "Orbital Formationkeeping with Differential Drag," *J. Guidance,* vol. 12, no. 1, 1987.

[33] B. S. Kumar, A. Ng, K. Yoshihara and A. De Ruiter, "Differential Drag as a Means of Spacecraft Formation Control," in *Aerospace Conference, 2007 IEEE*, Big Sky, MT, 2007.