

Spring 2013

NASA Magnetospheric MultiScale Mission TableSat 1C

Joshua Chabot

University of New Hampshire - Main Campus, joshchab@gmail.com

Joseph Kelley

University of New Hampshire - Main Campus, jkrunner@gmail.com

Michael Johnson

University of New Hampshire - Main Campus, mj21181@gmail.com

Follow this and additional works at: <https://scholars.unh.edu/honors>

 Part of the [Navigation, Guidance, Control and Dynamics Commons](#)

Recommended Citation

Chabot, Joshua; Kelley, Joseph; and Johnson, Michael, "NASA Magnetospheric MultiScale Mission TableSat 1C" (2013). *Honors Theses and Capstones*. 121.

<https://scholars.unh.edu/honors/121>

This Senior Honors Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Honors Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

NASA MAGNETOSPHERIC MULTISCALE MISSION TABLESAT 1C

**Joshua A. Chabot^{*}, Joseph J. Kelley[†], Michael A. Johnson[‡],
and May-Win L. Thein[§]**

INTRODUCTION

NASA's Magnetospheric Multiscale (MMS) mission is a Solar Terrestrial Probes Program that is expected to launch in 2014. The objective of this mission is to study the interaction between the Sun's solar winds and the Earth's magnetosphere. The MMS mission is a constellation mission consisting of four spin-stabilized spacecraft that must maintain a tetrahedral formation of varying dimensions to obtain 4-D structures (3-D space and time) of fundamental plasma phenomena.¹ Each spacecraft has six major instrumentation booms: four flexible 60 m Spin-plane Double Probe (SDP) booms with thicknesses of just under 2 mm, and two rigid 12.5 m Axial Double Probe (ADP) booms.³ A scaled image of the MMS spacecraft is shown in Figure 1 illustrating the boom configuration. The SDP booms are of primary focus because of their extreme length and flexibility, and hence their influence on the overall dynamics of the MMS spacecraft. As such, there is much interest in analyzing the spacecraft dynamics under spin rate and nutation control.

The Advanced Controls Lab (ACL) at the University of New Hampshire has constructed a series of table top experimental test beds (TableSats) for MMS spacecraft dynamics, control, and flexible structure analysis. TableSat I, the first generation of TableSats, are limited 3-DOF rotation (full spin and limited nutation) test beds. TableSat IC is the latest design iteration in the TableSat I family and incorporates design changes and updated technologies to more closely mimic MMS spacecraft characteristics. To study the effects of spin rate and nutation control on the experimental spacecraft bus and flexible booms, a PID controller is implemented on TableSat IC using sensor feedback from an on-board Inertial Measurement Unit (IMU).

^{*}Undergraduate Student of Mechanical Engineering at the University of New Hampshire

[†]Undergraduate Student of Mechanical Engineering at the University of New Hampshire

[‡]Undergraduate Student of Mechanical Engineering at the University of New Hampshire

[§]Associate Professor of Mechanical Engineering at the University of New Hampshire

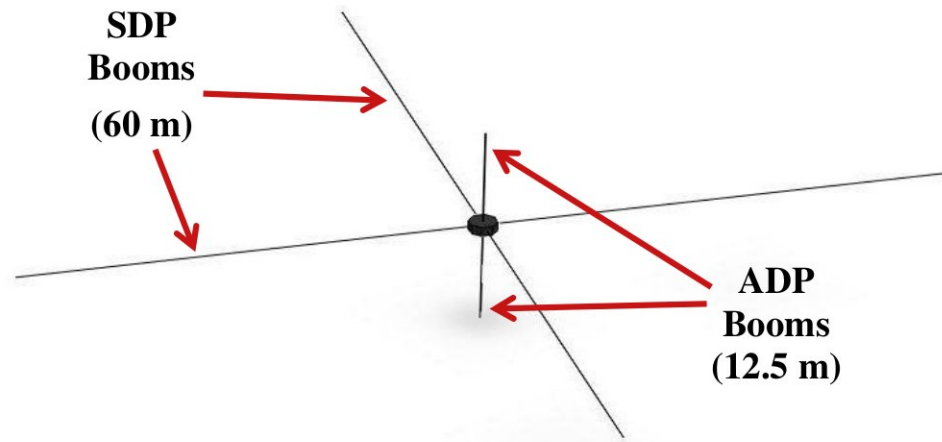


Figure 1: Scaled rendering of MMS spacecraft highlighting the extreme length of the SDP booms

EXPERIMENTAL TEST BED

The TableSat IC test bed shown in Figure 2 was designed to simulate the attitude dynamics of the actual MMS mission spacecraft. To accomplish this, TableSat IC sits atop a spherical air bearing that allows the platform to move with three degrees of rotational freedom with essential no friction. Additionally, the center of gravity and center of rotation of the test bed can be aligned such that torque from gravity is negated, in effect mimicking a weightless body.



Figure 2: MMS TableSat IC experimental test bed on air bearing stand

Thin aluminum booms protrude from the circumference of the test platform, simulating the flexible MMS spacecraft SDP booms. However, the flexible booms on TableSat IC have a higher natural frequency (0.95 Hz) than those found on the actual MMS spacecraft (0.15 Hz) and can only oscillate in the xy -plane due to the geometry of the aluminum sheet metal. The low natural frequency of the 60 m MMS SDP booms is impractical to duplicate in this earthbound setup. In addition to the

flexible SDP booms, a single carbon-fiber boom protrudes axially from the test bed, simulating an ADP boom.

A high pressure nitrogen propulsion system is currently used to actuate the test bed and simulate the thrusters present on the actual MMS spacecraft. However, the majority of the results presented in this report were recorded using the previous CO₂-based system. Six thrust nozzles are situated around the circumference of the test platform and are directed to control nutation and rotation accordingly – four of the thrusters are oriented downwards to control nutation about the x and y axes and two of the thrusters are directed tangentially to provide rotation about the z -axis. This propulsion system is highlighted in Figure 3 where the top cover of the test bed is transparent. A central cannister contains nitrogen gas at 3000 PSI that is used to propel the test bed. This high pressure nitrogen is regulated to approximately 15 PSIG using two regulators and is monitored via an on-board pressure transducer. Flow to these thrust nozzles is controlled via solenoid valves and an 84 MHz Arduino Due microcontroller. The microcontroller regulates the attitude and body rates of the test bed with a PID-based negative feedback control algorithm that receives attitude quaternion and body rate measurements from the IMU. The IMU is composed of a 3-axis magnetometer, accelerometer, and gyroscope and provides body rate and quaternion attitude information.

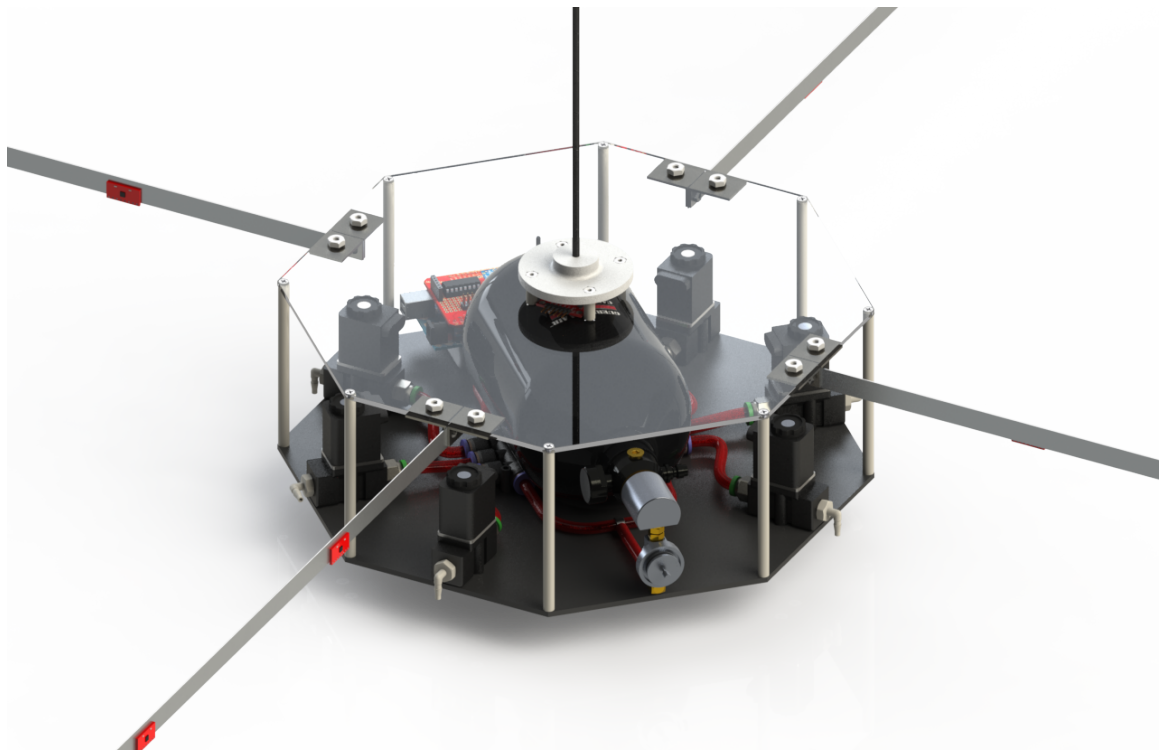


Figure 3: MMS TableSat IC propulsion system shown under clear top

THEORETICAL MODEL AND PID CONTROL DESIGN

Before implementing the experimental PID controller on TableSat IC, the system is first theoretically simulated assuming the test bed is a rigid body with constant mass and that it operates in a frictionless environment. A negative feedback PID control scheme is implemented as shown in Figure 4:

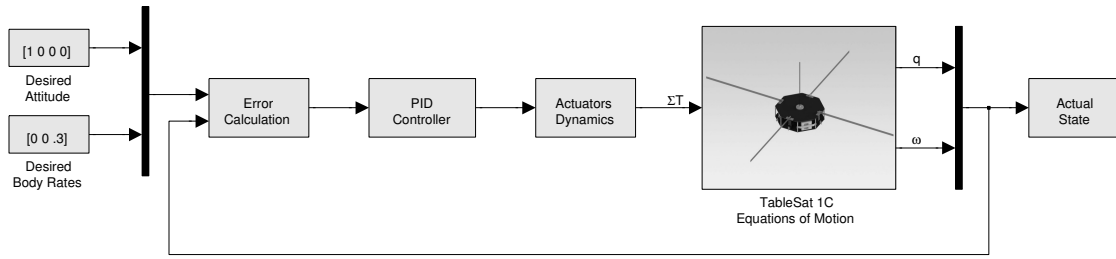


Figure 4: TableSat IC and PID controller simulation block diagram using PID control

In this simulation, the rotational dynamics of TableSat IC are governed by

$$\Sigma \mathbf{T} = \mathbf{I} \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega}) \quad (1)$$

where $\Sigma \mathbf{T}$ is the vector of actuator torques, \mathbf{I} is the moment of inertia tensor, and $\boldsymbol{\omega}$ is the angular velocity vector with respect to the inertial frame.² The moment of inertia tensor of the test platform is theoretically calculated using 3D CAD software and measured mass properties of the components. Unit quaternions were used to represent the attitude of TableSat IC, as given by

$$\dot{q} = \frac{1}{2} \boldsymbol{\omega}' \otimes q, \quad (2)$$

where $\boldsymbol{\omega}' = [0 \ \boldsymbol{\omega}^T]^T$ is the quaternion representation of the angular velocity vector provided by Eq. (8) and q is the attitude quaternion defined by

$$q = q_0 + \mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}. \quad (3)$$

Here, q_0 is the scalar component and \mathbf{q} is the vector component of the quaternion.¹ Body rate errors provided by Equation (8) are calculated such that:

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_d - \boldsymbol{\omega}_m \quad (4)$$

where $\boldsymbol{\omega}_e$ is the body rate error, $\boldsymbol{\omega}_d$ is the desired body rate, and $\boldsymbol{\omega}_m$ is the measured body rate. Additionally, attitude quaternion error is given by

$$q_e = q_d \otimes q_m^* \quad (5)$$

where q_e is the error quaternion, q_d is the desired quaternion state, and q_m is the measured quaternion state. Only the vector component of the attitude quaternion is used in this PID controller because the scalar component is a redundant term. With the error state defined by $e = [q_e \ \boldsymbol{\omega}_e]^T$, the PID control law is given by

$$\mathbf{T}(t) = \mathbf{K}_p e(t) + \mathbf{K}_i \int_0^t e(\tau) d\tau + \mathbf{K}_d \dot{e}(t) \quad (6)$$

where K_p , K_i , and K_d are proportional, integral, and derivative gain matrices, respectively.

Pulse-width modulation, with a pulse period of 0.1 s, is used to provide variable thrust to the test bed. The thruster dynamics of the system are also included in the analytical simulation, and are modeled as providing a first-order thrust response with an initial reaction delay. The thruster delay time and the response time constant were experimentally found to be 12 ms and 4 ms, respectively, at 15 PSIG operating pressure.

The PID controller gain matrices of Equation (6) are manually tuned until a desired spin rate of 2 rad/s about the z -axis and an attitude of $q_d = [1 \ 0 \ 0 \ 0]^T$ were achieved, given an initial attitude quaternion of $q_0 = [0.98 \ 0.11 \ 0.15 \ 0.11]^T$ ($\omega_0 = [15^\circ \ 15^\circ \ 15^\circ]^T$). These PID gains are detailed in Tables 1, 2, and 3. A spin rate of 2 rad/s is chosen for TableSat IC because of the difficulties associated with controlling the 0.3 rad/s spin rate that is needed on the actual MMS satellites. As can be seen from the results shown in Figure 5, the simulated TableSat IC achieves the desired state in approximately 5 s with minimal control effort, evidenced by the low thruster PWM duty cycle.

Table 1: PID Gains for Roll Actuator

	q_1	q_2	q_3	ω_x	ω_y	ω_z
K_P	-15	0	0	10	0	0
K_I	0	0	0	0	0	0
K_D	10	0	0	0	0	0

Table 2: PID Gains for Pitch Actuator

	q_1	q_2	q_3	ω_x	ω_y	ω_z
K_P	0	-15	0	0	10	0
K_I	0	0	0	0	0	0
K_D	0	10	0	0	0	0

Table 3: PID Gains for Yaw Actuator

	q_1	q_2	q_3	ω_x	ω_y	ω_z
K_P	0	0	0	0	0	15
K_I	0	0	0	0	0	0
K_D	0	0	0	0	0	3

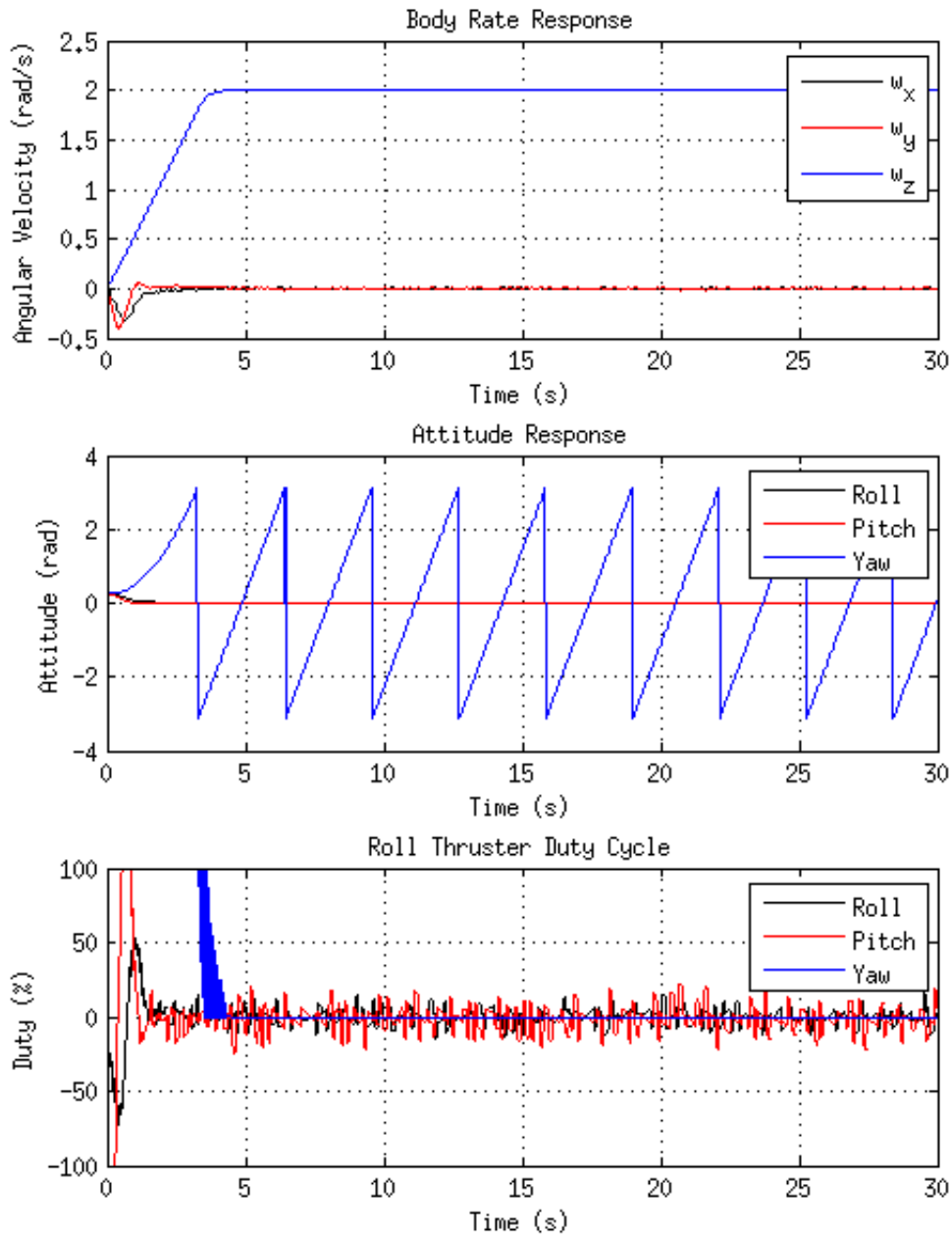


Figure 5: Simulated body rate and attitude response of PID controlled TableSat IC with thruster duty cycle. $\omega_d = [0 \ 0 \ 1]^T$, $q_d = [1 \ 0 \ 0 \ 0]^T$

EXPERIMENTAL PID CONTROLLER

The flight software of the TableSat IC Model is written in C++ for the Arduino platform. The Arduino platform was selected because it is an open platform with a large number of libraries, has a large community, and is supported by a wide range of microprocessors. Object oriented design was used for the flight software. In addition to simplifying the design of the software, the object oriented model closely mimics the Matlab Simulink Blocks that the theoretical model of the controller uses.

To validate that the flight software is functioning correctly, two different methods were used. The first method was to create unit tests that verify the code is working as expected. The second method was to create a Simulink function block that executes the C++ code inside the Simulink simulation. The first method tests the logic of the flight software and ensures that the controller code is calculating the correct values while the control algorithm executes. The second method tests the control algorithm by executing the control algorithm inside the simulation and allows the output to be compared against the theoretical controller output.

Several different methods were investigated for the unit testing. Ideally the unit testing would be performed while the code runs on the Arduino microprocessor. That way the code would be tested while running on the platform that it was intended for, eliminating any possible differences between the software's performance while running on the computer it was developed on and the intended platform. It also eliminates any potential differences in how the compiler generates the byte code for the Arduino and how the development computer's compiler generates its byte code. There is an Arduino library for unit testing that reports the results of the tests out of the serial line. This option was investigated, but ultimately we decided not to use it. The Arduino has a limited amount of RAM. Some of the smaller models only have 2000 bytes, and the Arduino Mega that we used only has 8000. The limited amount of memory limited how much information the unit tests could output because the message strings need to be stored in the Arduino's memory. Having a verbose output quickly limited the size of the tests, and on the other hand if no information was reported about why tests failed the tests could be large but would be of limited usefulness. We decided to use the Google C++ Unit Testing framework for our unit tests. It is a powerful library with more features than the Arduino unit testing library. It is also simple to set up and use and running on the development computer instead of the Arduino allowed the tests to be written faster without having to worry about memory constraints.

The Simulink function block is still a work in progress, but once completed it will allow any algorithms that are implemented on the Arduino to be tested and validated by the simulation. Since Simulink, the development computer, and the Arduino all depend on different libraries there are various `#ifdef` statements that include different libraries depending on what has been defined using `#define`. That way the same code that is running on the Arduino will be the same code tested by Simulink and the unit tests. One challenge to implementing the Simulink function block has been the timing needed for the time based integration and differentiation. Since the simulation is running on a different simulation time scale and not in real time, the Integration and Differentiation classes need to be altered to use the simulation time.

As for the flight software itself, the code has a hierarchy of objects that abstract away trivial tasks so that the controller object needs only to worry about important higher level tasks. On the lowest level there is an Arduino Matrix Math library that contains a set of functions that perform matrix operations on multidimensional arrays. We created a wrapper class called "Matrix" for this Matrix Math library that handles all of the memory allocation for a matrix and implements many

commonly used matrix functions. This wrapper class also overloads many common operators so that the matrix objects are recognized as special data types. For example, with Matrix A and Matrix B, $A * B$ would perform matrix multiplication. By creating this wrapper class it helps to ensure that there are no memory leaks by handling all of the memory internally to the class. For further details please refer to the Appendix.

The classes “Quaternion” and “SpinRate” are both special types of matrix objects. These objects contain the functions for dealing with the attitude and body rates of the experimental controller. The Quaternion class also overloads the operators dealing with special quaternion operations. A State class has both a Quaternion and a SpinRate as fields. This object holds all of the information associated with a certain state to the controller. These classes are not specific to any one controller and can be reused when other algorithms are implemented. There is a Modulator class that takes output from the controller and uses these values to perform pulse width modulation on the thrusters to produce the desired output torque.

The PID class is the class that implements the PID controller itself. It deals with the gains, the algorithm, and storing persistent values needed in future iterations. The class has an Integrator and Differentiator class as fields. These classes deal with integrating and differentiating the states used in the algorithm. Please see the Appendix for further details.

Currently in order to change control algorithms different classes need to be included. We intend to change this using C++’s inheritance features so that the code uses a generic controller class. The specific control algorithm implementations will be child classes of the generic controller class. This way new flight software does not need to be loaded onto the Arduino when we need to test a new algorithm.

EXPERIMENTAL PID CONTROLLER RESULTS

Spin Rate Control

Spin rate control is tested at desired spin rates of 0.3, 2, and 10 rad/s, both with and without the SDP booms. The 0.3 rad/s spin rate – the spin rate of the actual MMS spacecraft – cannot accurately be achieved because the thruster system cannot produce the necessary thrust precision, as can be seen in Figure 6.

Due to the inaccuracies inherent with slower spin rates, higher angular rates of 2 and 10 rad/s are examined. As can be seen from the results shown in Figures 7 and 8, the response times of the test bed under spin rate control for 2 and 10 rad/s are approximately 1 and 10 s, respectively. Once steady-state was achieved, the spin rates for each of these trials oscillate about the setpoint by approximately ± 0.2 rad/s.

Next, the same desired spin rates are tested with the SDP booms removed from the test bed to examine the effects of the flexible bodies on spin rate. These results can be seen in Figures 9 and 10. The response time for the spin rate tests without the SDP booms prove to be much faster than those with the SDP booms, partially due to reduced drag effects. The low natural frequency of the SDP booms also plays a role in the spin rate response of the system, as can be seen by the oscillation occurring in the transient portion of the response that is not present in the boomless responses. The frequency of this oscillation is approximately 1 Hz, which corresponds to the natural frequency of the SDP booms, and is especially noticeable in Figure 8.

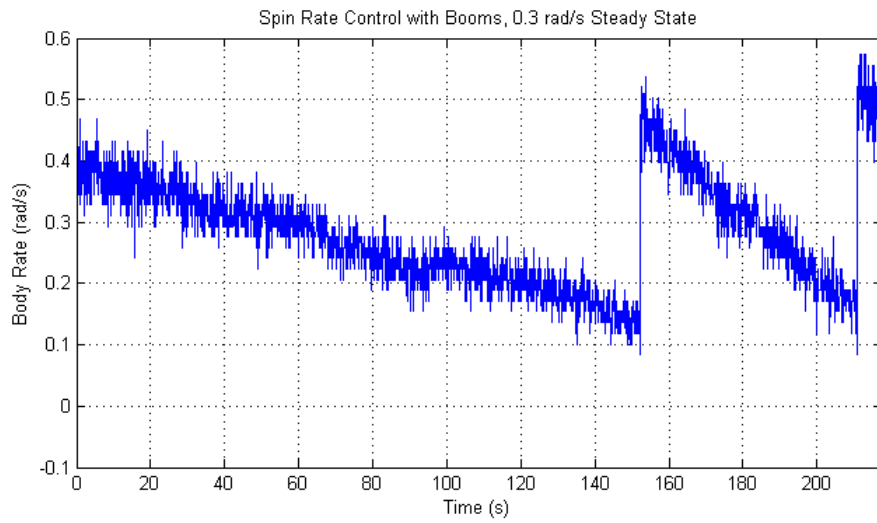


Figure 6: 0.3 rad/s spin rate control displaying insufficient thruster resolution

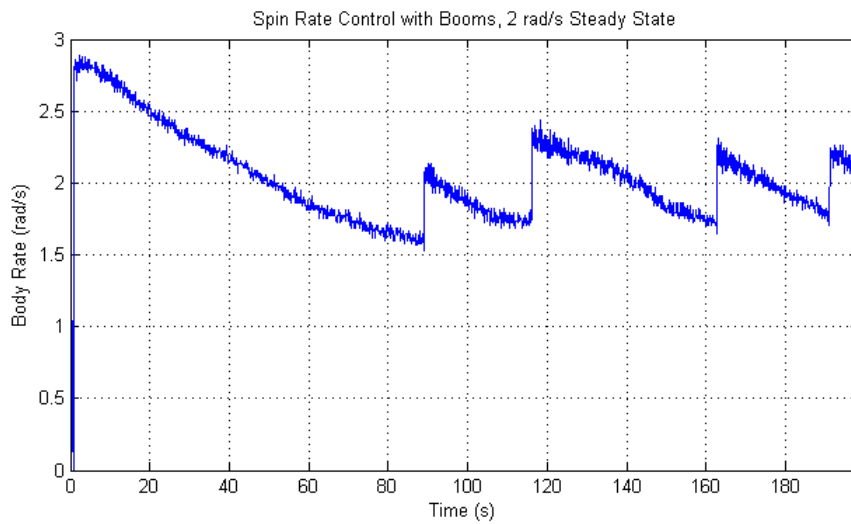


Figure 7: Spin rate control of 2 rad/s with SDP booms
 $P = 10$, $I = 1$, and $D = 5$

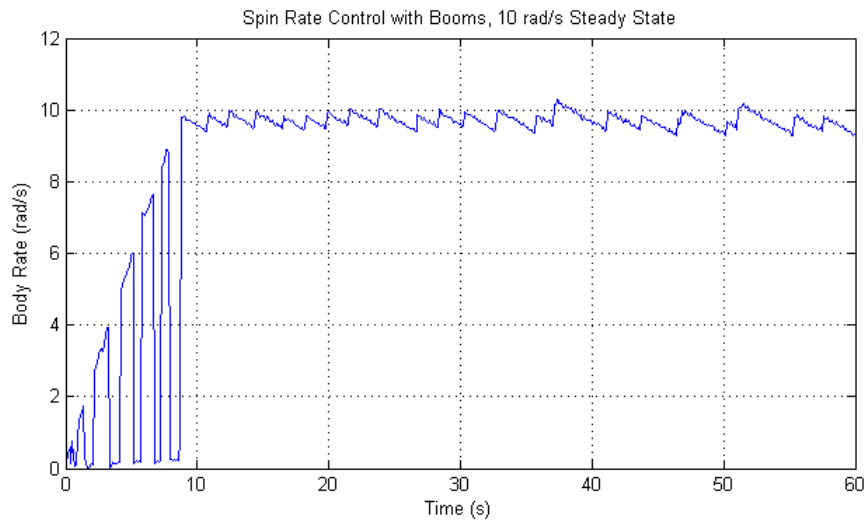


Figure 8: Spin rate control of 10 rad/s with SDP booms
 $P = 15$, $I = -0.15$, and $D = 5$

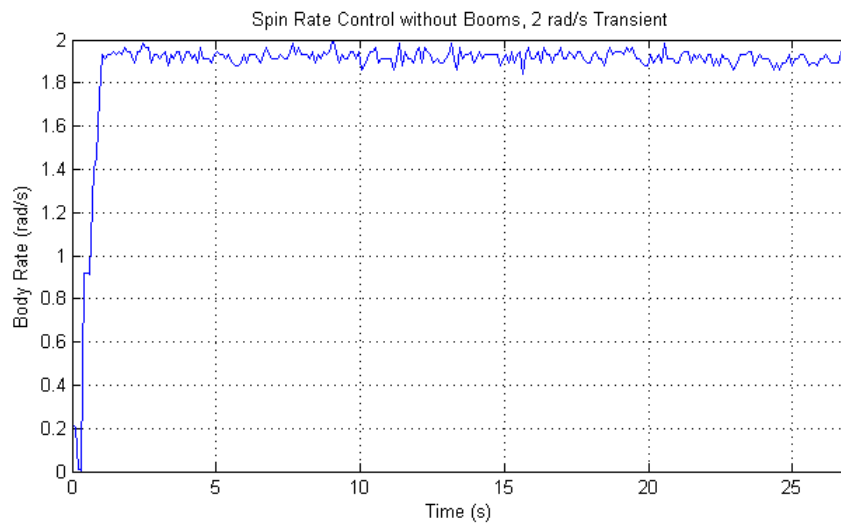


Figure 9: 2 rad/s. $P = 1$, $I = 0$, $D = 0$

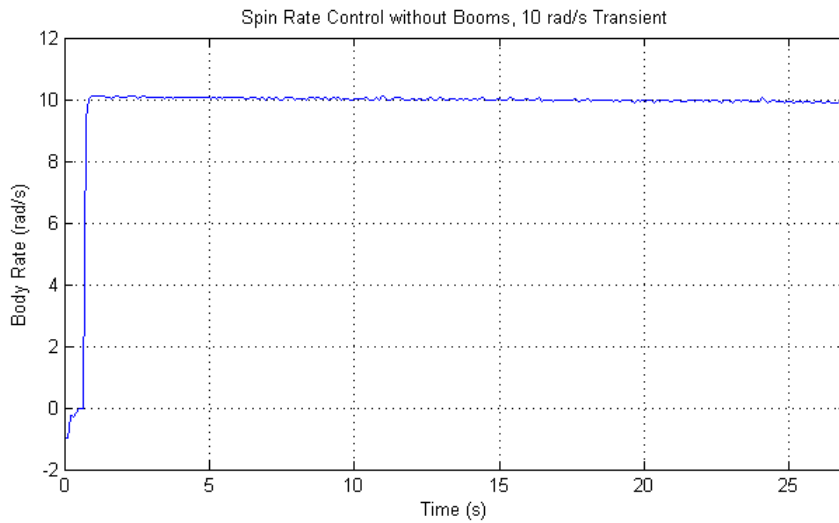


Figure 10: 10 rad/s. $P = 1$, $I = 1$, $D = 1$

Attitude Control

Nutation control is independently implemented on TableSat IC, with the center of gravity below the point of rotation to create a self-stabilizing system. The free nutation response of the system is measured and compared against the controlled response to examine the effectiveness of the nutation controller. This comparison can be seen in Figures 11 and 12.

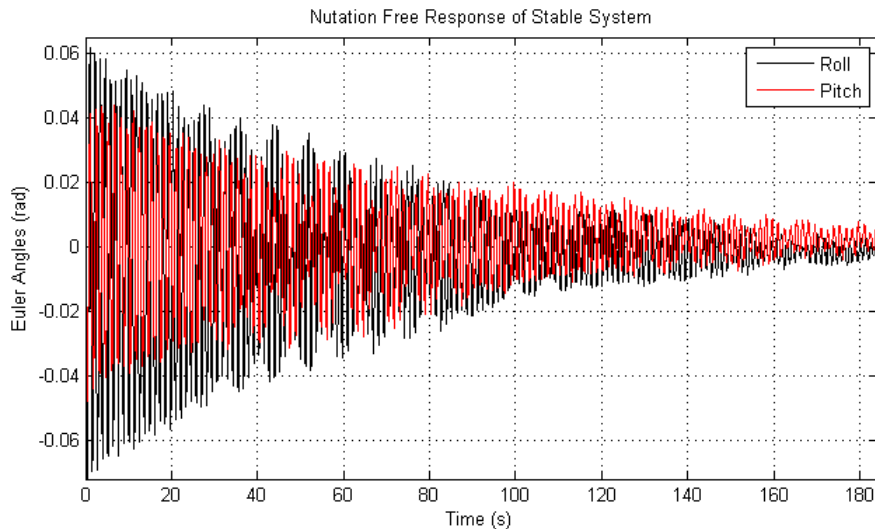


Figure 11: Nutation free response of self-stabilizing system

The free response takes approximately 140 s to damp to a magnitude of ± 0.015 rad ($\pm 0.86^\circ$), whereas the controlled response achieves this magnitude in only about 15 s. At steady-state, the controlled response continues to oscillate instead of damping out due to error introduced by the test bed's own thrusters. As mentioned previously, the thrusters are only capable of producing a certain minimum torque. Therefore, if the thrust needed is smaller than this minimum, the thruster

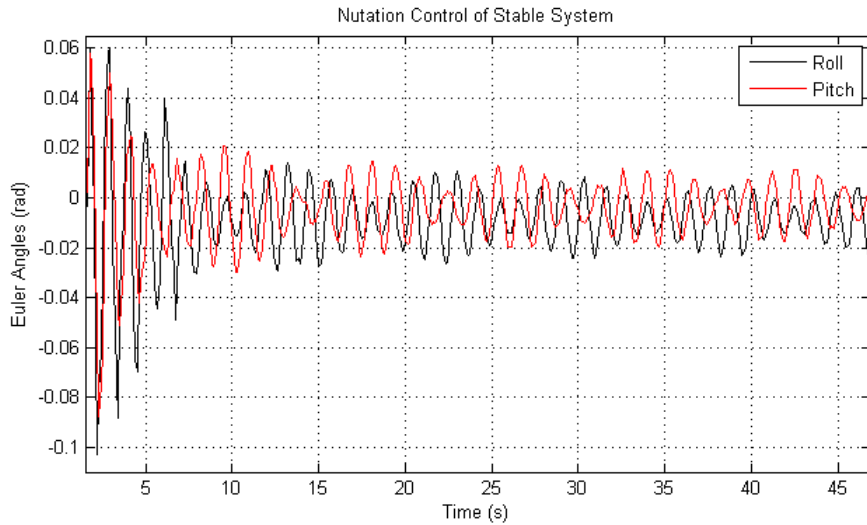


Figure 12: Nutation control of self-stabilizing system

produces further error that the controller will attempt to compensate for. This causes a controlled oscillation about the set point.

Next, attitude control is examined with the setpoint at a non-zero nutation angle. This produces a constant external torque on the system, in turn forcing the platform away from the setpoint. For this test, the desired roll and pitch are 0 and -8.6° , respectively. As can be seen in Figure 13, the platform achieves steady-state in under a second, but continues oscillates about the setpoint.

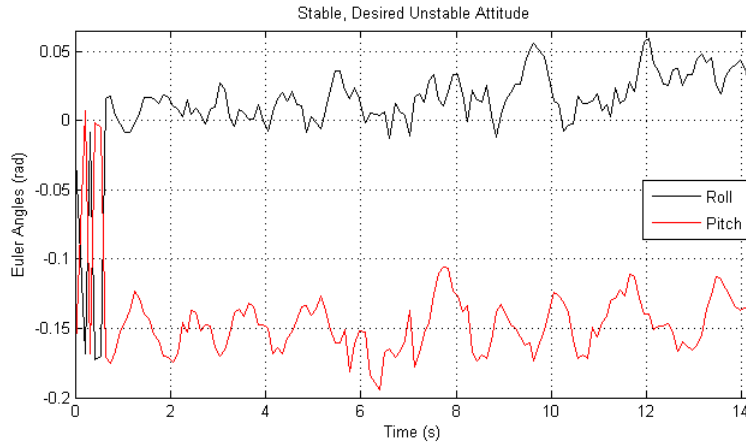


Figure 13: Nutation control of self-stabilizing system with setpoint not at equilibrium

$$P_x, P_y = 5, I_x, I_y = 0, D_x, D_y = 1$$

The final nutation test is performed by raising the center of gravity of the test platform creating a less stable system. The location of the center of gravity of the test bed is manipulated by adding and subtracting spacers between the air bearing and the platform. These spacers effectively change the location of the center of rotation of the test bed relative the center of mass. The free and controlled response of this system from a disturbance torque can be seen in Figures 14 and 15. Compared

to the stable system, the frequency of oscillation is much lower and takes significantly longer to settle. The controlled attitude response of the neutrally stable system achieves steady-state after approximately 4 s. At this point, it oscillates about the setpoint because it is extremely sensitive to torques caused by the thrusters and sloshing propellant.

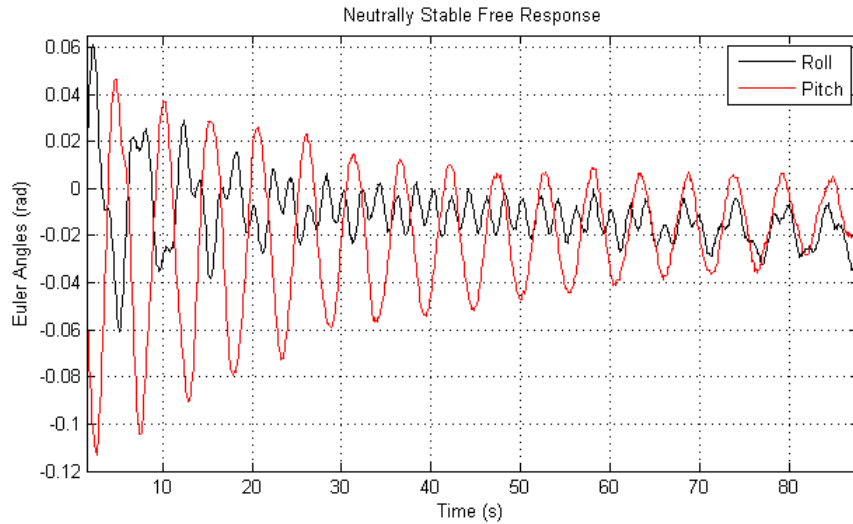


Figure 14: Nutation free response of self-stabilizing system

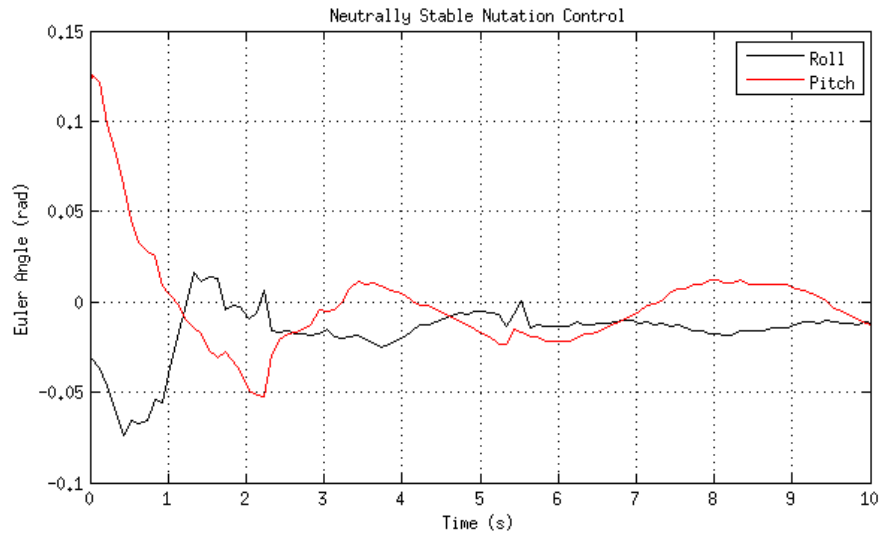


Figure 15: Nutation control of self-stabilizing system

LQR CONTROLLER DESIGN

To implement a Linear Quadratic Regulator (LQR) control algorithm on TableSat 1C, the spacecraft equations of motion are linearized using a first-order Taylor series approximation and put into state space form. The first-order Taylor series approximation is given by

$$P(\mathbf{x}) = f(\mathbf{a}) + Df(\mathbf{a})(\mathbf{x} - \mathbf{a}) \quad (7)$$

where P is the first-order linear approximation of the multivariable function f and \mathbf{a} is the vector about which the function is linearized. Once in state space form, the state space matrices are used to calculate an LQR gain matrix that controls the linearized system.

Linearization

To begin, the dynamic equations of motion of a spacecraft given by Equation 8 may be rewritten as

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} (\mathbf{I}\boldsymbol{\omega} \times \boldsymbol{\omega} + \mathbf{M}) \quad (8)$$

where \mathbf{M} is the vector of external moments, \mathbf{I} is the moment of inertia tensor, and $\boldsymbol{\omega}$ is the vector of angular rates. With more detail, Equation 8 is

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1} \left(\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \right) \quad (9)$$

Applying the Taylor series approximation described by Equation 7 to Equation 8 about the point $\mathbf{a} = [\omega_x = 0 \ \omega_y = 0 \ \omega_z = \omega_0]^T$ produces

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} \left(\omega_0 \begin{bmatrix} I_{yx} & (I_{yy} - I_{zz}) & 2I_{yz} \\ (I_{zz} - I_{xx}) & -I_{xy} & -2I_{xz} \\ -I_{yz} & I_{xz} & 0 \end{bmatrix} \boldsymbol{\omega} + \begin{bmatrix} M_1 - I_{yz} \omega_0^2 \\ M_2 + I_{xy} \omega_0^2 \\ M_3 \end{bmatrix} \right) \quad (10)$$

Attitude Linearization

In addition to the spacecraft dynamics, the quaternion kinematic equations of motion of a spacecraft given by Equation 2 are also linearized. In more detail, Equation 2 is given by

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (11)$$

However, because q_0 and q_3 represent the yaw of the spacecraft and because the yaw is continually changing under spin rate control, these elements will not be directly controlled and can therefore be set to zero. Applying the Taylor series approximation given by Equation 7 to Equation 11 about $\mathbf{a} = [q_1 = p_1 \ q_2 = p_2 \ \omega_x = 0 \ \omega_y = 0 \ \omega_z = \omega_0]^T$ produces

$$\dot{q} = \frac{1}{2} \left(\begin{array}{cccccc} \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & -p_1 & -p_2 & 0 \\ 0 & 0 & \omega_0 & 0 & 0 & 0 & p_2 \\ 0 & -\omega_0 & 0 & 0 & 0 & 0 & -p_1 \\ 0 & 0 & 0 & 0 & -p_2 & p_1 & 0 \end{array} \right] & \begin{bmatrix} 0 \\ q_1 \\ q_2 \\ 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} 0 \\ -p_2\omega_0 \\ p_1\omega_0 \\ 0 \end{bmatrix} \end{array} \right) \quad (12)$$

State Space Representation

Finally, to implement LQR control the linearized equations of motion must be put into state space form, which is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \quad (13)$$

where \mathbf{x} is the state vector, \mathbf{A} is the state matrix, \mathbf{u} is the input vector, and \mathbf{B} is the input matrix. Combining and arranging the linear equations given by Equations 10 and 12 produces

$$\mathbf{x} = \begin{bmatrix} 0 \\ q_1 \\ q_2 \\ 0 \\ \omega_x \\ \omega_y \\ \omega_z \\ 1 \end{bmatrix}$$

for a state array. Note that an eighth state was added to the system to include the constants produced by Equations 10 and 12. The remaining state space components are defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{1}{2}p_1 & -\frac{1}{2}p_2 & 0 & 0 \\ 0 & 0 & \omega_0 & 0 & 0 & 0 & \frac{1}{2}p_2 & -\frac{1}{2}p_2\omega_0 \\ 0 & -\omega_0 & 0 & 0 & 0 & 0 & -\frac{1}{2}p_1 & \frac{1}{2}p_1\omega_0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2}p_2 & \frac{1}{2}p_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yx} & (I_{yy} - I_{zz}) & 2I_{yz} & -I_{yz}\omega_0^2 \\ 0 & 0 & 0 & 0 & \omega_0 \mathbf{I}^{-1} (I_{zz} - I_{xx}) & -I_{xy} & -2I_{xz} & I_{xy}\omega_0^2 \\ 0 & 0 & 0 & 0 & -I_{yz} & -I_{xz} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}$$

The \mathbf{A} and \mathbf{B} are then used to calculate the LQR gain matrix after suitable \mathbf{Q} and \mathbf{R} matrices are chosen.

CONCLUSION

The chosen analytical PID gains differ from those found for the experimental platform. The difficulty in controlling the nutation and spin rate of the test bed comes from three major sources: uncertainties in the inertia tensor, uncertainties in the thruster force, and angular velocity noise from the gyroscope.

Uncertainties in the inertia tensor stem from the fact that the moments of inertia of the test bed are theoretically calculated and not physically measured. Additionally, as the test platform expels propellant, its mass changes, therefore changing the inertia tensor. With these uncertainties in inertia, the center of gravity is difficult to align with the center of rotation, introducing unwanted external torques on the platform that the controller may not be able to overcome. The new nitrogen-based propulsion system will reduce uncertainties in the inertia tensor due to the fact that the nitrogen tank is more symmetric and thanks to the fact that nitrogen won't slosh like the liquid CO₂ does.

The phase change of the CO₂ propellant introduces further uncertainty in that it causes pressure fluctuations in the system, and therefore, fluctuations in the force produced by the nozzles despite the use of regulators. This issue is no longer a major problem since upgrading to a high pressure nitrogen system, which does not have the inherent phase change.

Finally, noise in the angular velocity measurements made introducing derivative gains difficult. A low pass filter is applied to the gyroscope in an attempt to reduce this noise but is not sufficient for full attitude control. A more advanced filter is needed to reduce these adverse effects.

ACKNOWLEDGMENTS

Funding for this work was granted by NASA Goddard Space Flight Center's Magnetospheric MultiScale (MMS) mission through the Flight Dynamics Analysis Branch of the NASA Goddard Space Flight Center, the New Hampshire Space Grant Consortium, and the Hamel Center for Undergraduate Research at the University of New Hampshire. Special acknowledgments go to Josephine San (Former NASA MMS Attitude Control Systems Lead), Sam Placanica (NASA MMS Attitude Control Systems Lead), Dean Tsai (NASA MMS Attitude Control Systems Member and Aerospace Engineer), and all other members of the NASA MMS Attitude Control Systems Group. Additionally, special acknowledgments are extended to Dan Couture, Joe Jourdain, Sheldon Parent, and Bob Champlin for their assistance and advice on this project and to Tom Gross and Robin Reed at NiPro Optics for their help and generosity in obtaining a precision mirror.

REFERENCES

- [1] B. Jenkins, *Accelerometer Calibration for NASA's Magnetospheric MultiScale Mission Spacecraft*. Durham: University of New Hampshire, 2011.
- [2] M. Kaplan, *Modern Spacecraft Dynamics and Controls*. New York: Wiley, 2nd ed., 1989.
- [3] N. Mushaweh, *An Observer-Based Attitude and Nutation Control and Flexible Dynamic Analysis for the NASA Magnetospheric Multiscale Mission*. Durham: University of New Hampshire, 2007.
- [4] B. Wie, *Space Vehicle Dynamics and Control*. Reston: American Institute of Aeronautics and Astronautics, 1998.
- [5] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Dordrecht: Reidel, 1978.
- [6] T. Roemer, N. Aubut, W. Holmes, J. Chabot, A. Jenkins, M. Johnson, and M. L. Thein, "Using the Magnetospheric MultiScale (MMS) TableSat IB for the Analysis of Attitude Control and Flexible Boom Dynamics for MMS Mission Spacecraft," *Advances in Astronautical Sciences*, AAS 12-244, pp. 2115-24.