ASEN 3801 Attitude Lab: Attitude Sensors, Actuators, and Spacecraft Pitch Controls

Charles Absher, Eric Egaas, Jared Steffen, Matt Turner

October 27, 2023

1 Preliminary Questions

1.1 Lab Task 1

1.1.1 Preliminary Question 1

To calculate the angular position of the spacecraft at time t from gyro measurements (angular velocities about principle axes) given in RPM, one would first translate the values from RPM to rad/sec by multiplying by $\frac{2\pi}{60}$, then integrate the data from $t = 0$ to t with given time step dt.

1.2 Lab Task 2

1.2.1 Preliminary Question 2

The angular acceleration the wheel needs to balance out the drag force is equal to the moment of the drag, divided by the wheels inertia, but in the opposite direction.
$\alpha = \frac{-M_d}{I}$

1.2.2 Preliminary Question 3

If you are holding a gyroscope and it is rotating horizontally in a counter clockwise direction, if you tilt your hand around the y-axis, you will feel a resistance from the gyroscope as it tries to stay level and it will begin to precess.

1.2.3 Preliminary Question 4

In order to rotate to the left (spin axis up), you would rotate the wheel so it would be rotating to the right (spin axis down)

1.3 Lab Task 3

1.3.1 Preliminary Question 5

Since $\tau = I\alpha$, if the torque being applied is known and the angular rate is measured with spaced time increments of dt seconds, you could integrate the angular rate over time and then find I by diving the torque by α.

1.3.2 Preliminary Question 6

See Figure 1.

1.3.3 Preliminary Question 7

See Figure 2.

1.3.4 Preliminary Question 8

See Figure 3.

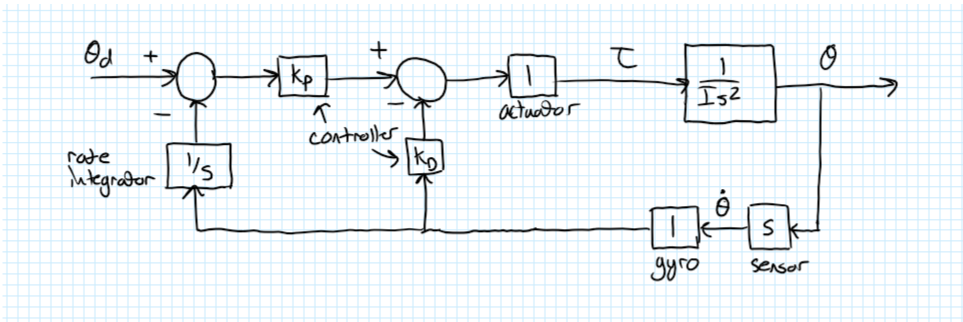

Figure 1: Control Block Diagram

$$I\ddot{\theta} = \tau = k_d(\underbrace{\cancel{\dot{\theta}_d}}_{0} - \dot{\theta}) + k_p(\theta_d - \theta)$$

$$I\ddot{\theta} = -k_d\dot{\theta} + k_p(\theta_d - \theta)$$

$$I\ddot{\theta} + k_d\dot{\theta} + k_p\theta = k_p\theta_d$$

$$I\ddot{\theta} + k_2\dot{\theta} + k_1\theta = k_1\theta_d$$

$$\ddot{\theta} + \frac{k_2}{I}\dot{\theta} + \frac{k_1}{I}\theta = \frac{k_1}{I}\theta_d$$

Figure 2: Derived Dynamical System Equation

Max Overshoot < 10%   5% settling time < 1.5s

$$0.1 = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\ln(0.1) = \frac{-\zeta\pi}{\sqrt{1-\zeta^2}}$$

$$(\ln(0.1))^2 = \frac{\zeta^2\pi^2}{1-\zeta^2}$$

$$5.3019(1-\zeta^2) = \zeta^2\pi^2$$

$$5.3019 - 5.3019\zeta^2 = \zeta^2\pi^2$$

$$5.3019 = 15.1715\zeta^2$$

$$\zeta^2 = 0.349$$

$$\boxed{\zeta = 0.591}$$

$$e^{-\zeta\omega_n t} = \frac{\%}{100}$$

$$-1.5\zeta\omega_n = \ln(0.05)$$

$$\zeta\omega_n = 1.99715$$

$$\boxed{\omega_n = 3.379 \text{ rad/s}}$$

$$\lambda = -\omega_n\zeta \pm i\omega_n\sqrt{1-\zeta^2}$$

$$\boxed{\lambda = -1.996 \pm 2.725i}$$

$$\omega_n = \sqrt{\frac{K_1}{I_{spin}}}$$

$$K_1 = \omega_n^2 I_{spin}$$

$$\boxed{K_1 = 11.419\, I_{spin}}$$

$$\frac{K_2}{I_{spin}} = 2\zeta\omega_n$$

$$K_2 = 2\zeta\omega_n I_{spin}$$

$$\boxed{K_2 = 3.9939\, I_{spin}}$$

Figure 3: Poles, Natural Frequency, Damping Ratio and Gains

2 Lab Task 1 - Physical Rate Gyro

2.1 Experiment

1a.

a. When rotating the spinning gyro around the different axes, we can notice a few consistent patterns. For example, when we rotate the gyro around the +x and +y there is no external effect on the rotating disc. When rotating about the z axis there is a observed upward motion of the rotating disc when you spin the box in the clockwise direction and a downward motion when you spin the box in the counter clockwise.

b. When rotating the spinning gyro around the +x and +y axes I do not feel any torques from the gyro. This changes when I rotate around the +z axis. With a sudden acceleration around this axis, there is a noticeable force in the opposite direction.

1b. By inspection when we rotate the box counterclockwise, around the +z axis, there is a downward precession. Due to right hand rule, in order for the downward precession to make sense, there must be a counter clockwise rotation of our disk. The same thing happens when we rotate the box clockwise around the +z axis. By the right hand rule, in order to get an upward precession, our disk must be spinning clockwise.

2.2 Results/Analysis

1c. When the physical rate gyro rotates around the z axis there is a clear perturbation around the x axis. Using Euler's rotational dynamics equations of motion given to us by $\dot{\omega}_B = \mathbf{I}_B^{-1} * [-\tilde{\omega}_B(\mathbf{I}_B * \omega_B) + \mathbf{G}_B]$. Although I won't show explicitly how to solve for the angular velocity of our spinning disk, we can use the sign of our output angular velocity to get a sense of the direction. The main point behind this equation is to show that the angular velocities need to work out so that our total angular momentum is conserved. That means if a positive angular velocity around the z axis corresponds to a positive theta about the x axis, and a negative angular velocity in the z axis corresponds to a negative theta about the x axis, we must have a negative angular velocity about the y axis, which confirms that our disc is rotating clockwise.

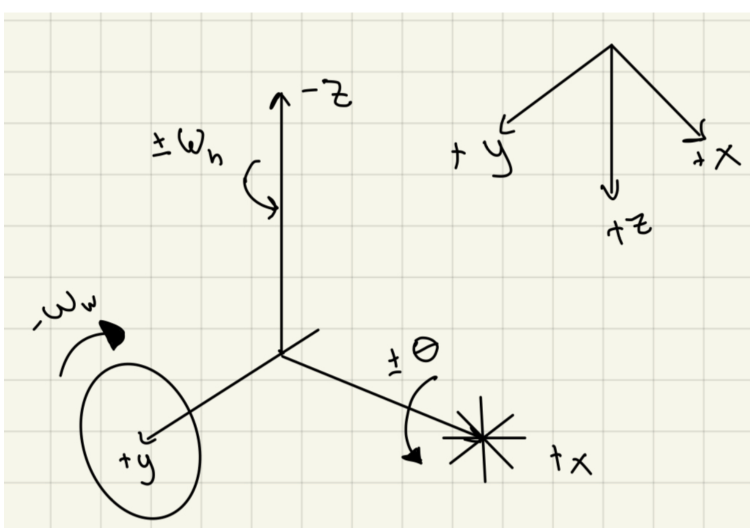

Figure 4: Diagram of Physical Rate Gyro

2.3 What Did We Learn?

We learned that the gyro's spinning axis responds perpendicular to an applied force. Another concept we learned is that gyros have high stability, but are prone to move to even the smallest of perturbations.

2.4 Recommendations

No recommendations as the experiment was easy and straightforward. Lab Task 3 does a sufficient job of extending the learning objectives.

3 Lab Task 2 - Control Moment Gyro Experiment

3.1 Experiment

3.1.1 Equipment List

- Bicycle wheel with handle
- Handheld electric spin-up motor
- Swivel board

3.1.2 Procedure

1. Have one team member stand on the swivel board with the bicycle wheel aligned vertically.
2. Have another team member use the handheld electric spin-up motor to make the bicycle wheel spin.
3. Begin to rotate the bicycle wheel so it is no longer vertical and observe the rotational motion of the bicycle wheel compared to the team member on the swivel board holding the bicycle wheel.
4. Have the team member with the bicycle wheel attempt to track a moving target (another team member).

3.2 Results and Analysis

b. While there were slight differences between team members holding the gyro, it was relatively easy to track the team member walking around the gyro wheel up to a certain point. About halfway through tracking a full circle it became harder to track the moving team member due to the swivel board not being fricitonless.

c. When inverting the wheel, the person holding the gyro would begin to spin in the opposite direction that the wheel was spinning. The spin rate could be changed by changing how tilted the bicycle wheel was. In order to turn directions, the person holding the bicycle wheel would change the direction that they would tilt the bicycle wheel. For example, if the person holding the wheel wanted to turn counter clockwise, they would rotate the wheel so it was spinning clockwise and vice versa.

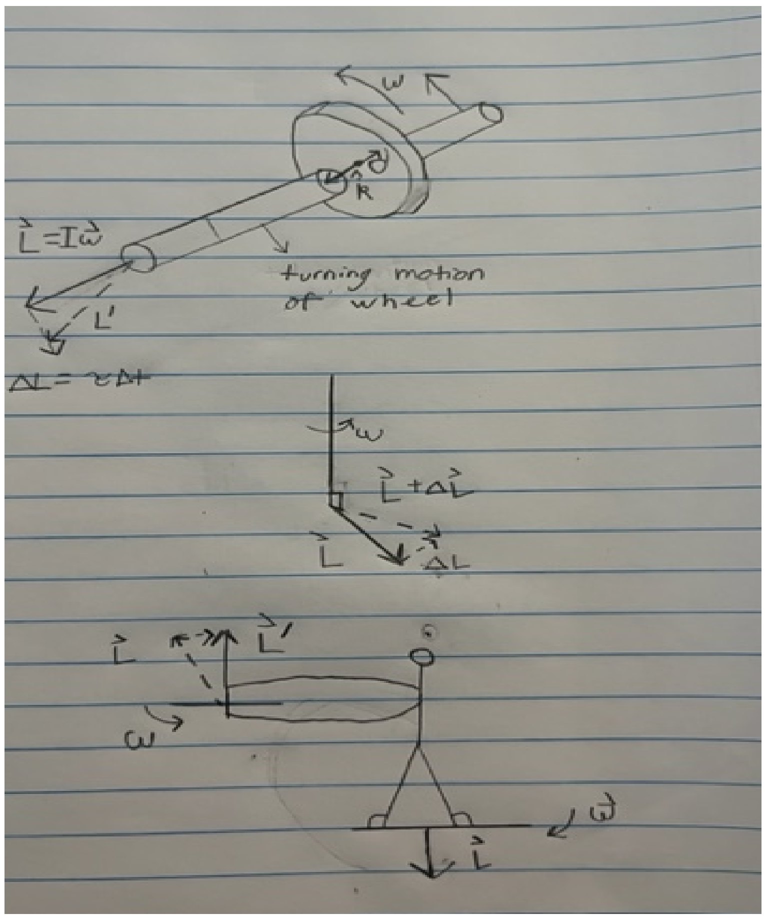

Figure 5: Momentum Vector and Vector Rate of Change

4 Lab Task 3 - Spin Module Experiments

4.1 MEMS Rate Gyro Characterization

4.1.1 Block Diagram

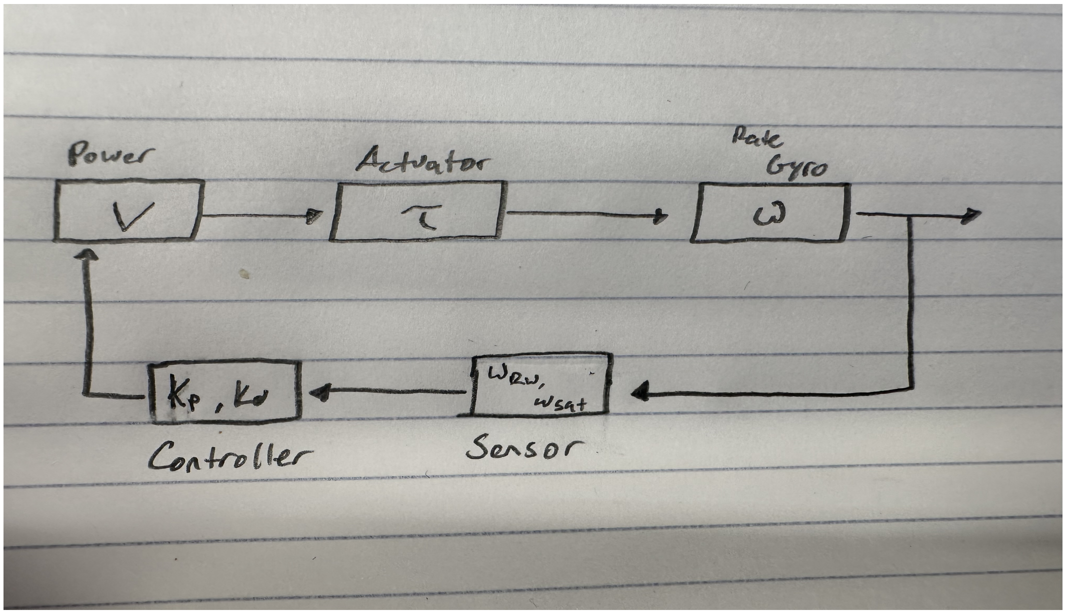

Figure 6: Functional Block Diagram of Rate Gyro

Functionally, the system is powered by a battery, this electric energy is converted into mechanical energy (torque) by the actuator. This torque is applied on the reaction wheel. The angular velocity of both the wheel and the satellite itself are measured by the sensor, fed through the controller that applies gains, relaying the necessary change in electrical power to the battery. The cycle repeats.

4.1.2 Manual Inputs

When moving the module manually, we see that if we move it back and forth there is a sinusoidal wave both the input and output sections. They are also out of phase, which is expected because when we spin the module one way, we expect the gyro to spin the other way. The largest sources of error are due to human input to the system.

4.1.3 Bias and Adjusted Scale Factor

When plotting the time history of the angular rates of the input and gyro output, Figure 8 in Appendix A shows that they are 90^o out of phase with one another, but the magnitudes are very similar, if not the exact same. In order to adjust this, some MATLAB code was used to find the bias and adjusted scale factor of the data sets. To do this, the input rate was plotted against the output rate, and the results can be seen in Figure 9 of Appendix A. The bias tells us how much of a consistent offset there is within the gyro. In order to calculate this, the average of the output rate was taken. The values for the gyro bias, as well as their mean and standard deviation, with various frequencies and current inputs can be seen in the table below. The adjusted scale factor was then calculated and used in order to get the input and gyro output in phase with each other. To calculate the adjusted scale factor, a best fit line was applied to the data. The slope of this best fit line serves as the adjusted scale factor. The values for adjusted scale factor, along with their mean and standard deviation, can also be seen in the table below. Figures 10 and 15 in Appendix A show the graphs for the time history of the input and adjusted gyro output rates.

Test Number	Frequency $[Hz]$	Current $[A]$	Adjusted Scale Factor $[rad/s^2]$	Bias $[rad/s]$
1	0.2	0.5	-0.8586	-0.0879
2	1.0	1.0	-0.8689	-0.0159
3	0.5	0.75	-0.8718	-0.0075

Scale Factor Mean $= -0.8664 rad/s^2$
Bias Mean $= -0.0107 rad/s$
Scale Factor Standard Deviation $= 0.0069 rad/s^2$
Bias Standard Deviation $= 0.0045 rad/s$

The gyro angular position can also be found with the data that was taken. To do this, we simply adjusted the time data to start at zero and increment accordingly, and then multiplied the input and newly adjusted gyro output by the time vector. The graphs for adjusted angular rates, adjusted angular position, their respective errors, and angular position as a function of input rate can all be seen in Figures 10-19 in Appendix A for a frequency of $0.2Hz$ and $0.5A$ as well as $1.0Hz$ and $1.0A$. No data was removed from these sets and the only thing that was adjusted was the time vector so it would start at zero. These specific data sets were used because they were the ones with the lowest and highest frequencies and current used respectively. This shows that the methods used to manipulate and display that data work with any reasonable frequency and current applied to the system. This also shows the importance of calibrating gyros before use. It can be used to ensure consistency and reduce error in attitude control.

4.1.4 Rate Gyro Analysis

The sensitivity of the rate gyro is able to be determined through the angular rate measurement error graphs (Figures 11 and 16). The graphs show that error is $\pm 1 rad/s$ from the actual input rate. The graphs also show that that the higher rate actually has less error than the slower rate. However, once the rates start getting high enough, this would probably not be the case and the error between input and output would be increase as rate increases. When exposed to abrupt changes in spin rate, the MEMS gyro doesn't perform as well as expected. This could bee seen in the manual experiments and is most likely due to the vibrations that are introduced with abrupt changes, Since the gyro is spinning relatively fast, any introduced vibrations would have a large effect on the gyros output measurements. The bias for the gyro was easily repeatable, but the values for bias were not very close and were sometimes a whole magnitude different. It is important for the bias to be determined because in an actual spacecraft, attitude control is extremely important due to the fact that most goals of spacecraft include being able to have extremely accurate attitude control. While the angular rates error was relatively small, the same cannot be said for angular position error. Figures 13 and 18 show that the angular position error was quite large ($\pm 30 rad$ was largest seen in our data). This is most likely due to the bias of the gyro itself. The graphs also show that as time goes on, the error keeps growing. This means that the drift is increasing with time and could be another explanation for why the position errors grow to large values even with small rate errors.

4.2 Reaction Wheel Characterization

4.2.1 Experiment

4.2.2 Equipment List

- Spacecraft mock up
- Computer with LabVIEW VI

4.2.3 Procedure

1. In the LabVIEW VI, open the torque section.
2. Select "Reaction Wheel" to specify that torque is being applied to the reaction wheel.

3. Enter a value of $10mNm$ of torque to be applied.
4. Hold onto the spacecraft and run the program for a desired time (we used 5 seconds).
5. Repeat steps 1-4 with varying torque values.

4.2.4 Theory

From the measurements taken in the experiment, we can back out the wheels MOIs using the formula $\tau = I\alpha$. The torque, τ, can be manually changed in the user interface of the reaction wheel, and angular acceleration, α, is found by taking the slope of the wheels angular velocity over time. Solving using coaxial values gives $I = \frac{\tau}{\alpha}$.

4.2.5 Block Diagram

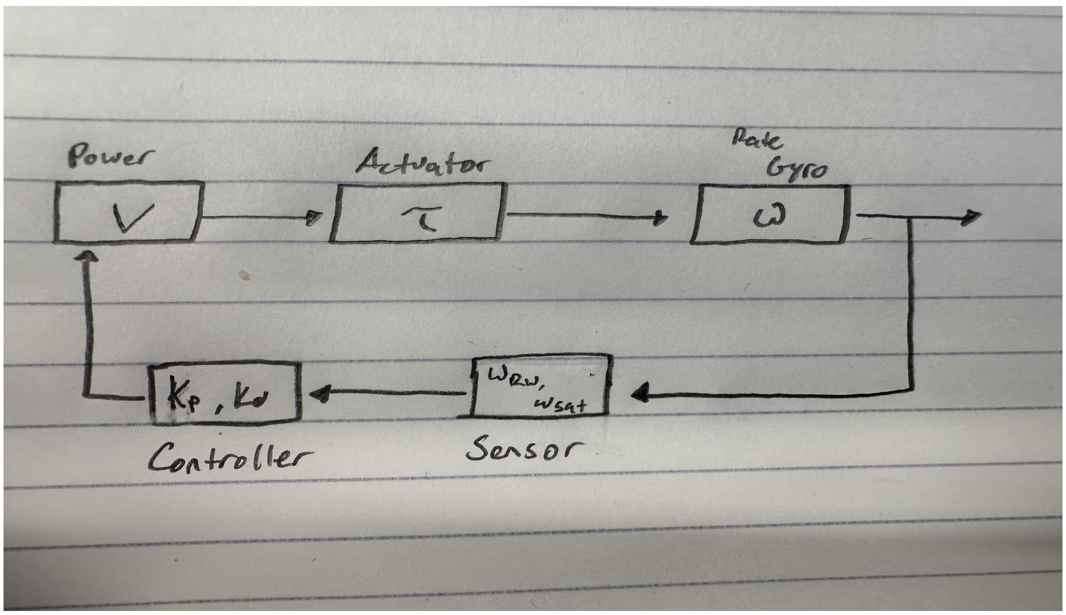

Figure 7: Functional Block Diagram of Rate Gyro (Same as Figure 6)

4.2.6 Results

Test Number	Torque $[mNm]$	Angular Acceleration $[rad/s^2]$	Moment of Inertia $[gm^2]$
1	6	24.72	0.2427
2	8	40.27	0.1986
3	10	54.89	0.1822
4	12	70.05	0.1713
5	14	85.04	0.1646

MOI Mean = $0.1919gm^2$
MOI Standard Deviation = $0.0312gm^2$

4.2.7 Analysis

Figure 20 in Appendix A shows the time history of the angular rate of the reaction wheel with $6mNm$ of torque being applied and the best fit line that was used to determine angular acceleration. Using the equation $I = \frac{\tau}{\alpha}$, this data was used to solve for the moment of inertia of the reaction wheel for all trials conducted. There was data that was removed from each trial as well. The data removed was data where the current being applied to the reaction wheel was not at its maximum value (found through inspection of the data). This is because if the current being applied was not at its maximum, then the torque being applied was not as well. If a torque of $10^{-4}Nm$ or $0.1mNm$ was applied to the system, it would take around $800s$ to reach the maximum value of $4000rpm$. The approximate angular momentum capacity is $0.08Nms$. The largest source of error when conducting these experiments come from having to hold he spacecraft still while torque was applied to the reaction wheel. As torque values increased, it became harder to hols the spacecraft exactly still. Despite this, however, we believe our data for the reaction wheel to be fairly accurate given the constraints.

4.3 Spacecraft Controls

4.3.1 Moment of Inertia

The moment of inertia of the spacecraft was found by dividing the torque applied by the angular acceleration. The torque applied was known as it was one of the inputs into the LabVIEW VI. Once the data was taken for applying a known torque to the spacecraft, it was imported into MATLAB. In MATLAB, after removing data where the current being applied to the motor of the spacecraft was not at the maximum value, the angular rate was plotted over time. A best-fit line was then applied to the data, and the slope of this best-fit line is equivalent to the angular acceleration. After dividing the torque by the angular acceleration magnitude, the moment of inertia was found to be $6.5011gm^2$

4.3.2 Explore Proportional Control Gains

When lower proportional gains were used (less than $75mNm/rad$), and the system was perturbed, the spacecraft would often be able to return to its initial position after some time.However, it would often oscillate back and forth for a good amount before being able to do so. If the perturbation was big enough, regardless of the proportional gains, the system was unable to return to its initial state. When larger proportional gains were used (more than $75mNm/rad/s$), the system was not able to return to its initial position, even with very small perturbations.

4.3.3 Controlling Angular Speed

When using a proportional control gain of $100mNm/rad$, the system would oscillate back and forth around the desired angular position for quite awhile. Due to the fact that no integral or derivative control was present, there was a large amount of error in getting to the desired position, hence the oscillations for a long duration.

4.3.4 Explore PD Gains

When using a proportional control gain of $100mNm/rad$ and a derivative control gain of $15mNm/rad/s$, the system was able to get to the desired position much faster than when using only proportional control. The oscillations were much smaller and went on for a much shorter duration. This is due to the addition of derivative control, which focuses on the past rate of change of error, effectively reducing the current error in the system.

4.3.5 Control Design

From the results of preliminary question 8, $\zeta = 0.591$. With $0 < \zeta < 1$, we know that the poles take the form of $\lambda = -\zeta\omega_n \pm i\omega_n\sqrt{1-\zeta^2}$. Given the requirements for maximum overshoot and settling time, ζ and ω_n were able to be determined and used to find the poles of the closed loop system. Figure 2 in Appendix A show the expected step response plotted in MATLAB. These poles tell us that the expected step response does in fact have a maximum overshoot of 10% and a 5% settling time within $1.5s$ and meets the required criteria.

4.3.6 Control Results

The results from the control design can be seen in figure 22 of Appendix A. We see that the maximum overshoot and settling time are within the goals of preliminary question 8. The only difference is that the spacecraft does not settle at the desired position of $0.5rad$, but is instead slightly lower at around 0.45rad. This is most likely due the physical hardware not being perfect, so it settled at a slightly lower position than the reference position.

4.3.7 Control Analysis/Comparison

Figures 21 and 22 in Appendix show the expected step response and actual step response respectively. Both the expected and actual results show a maximum overshoot of 10% and a 5% settling time within 1.5s. this shows that our calculated gains meet the design requirements presented in preliminary question 8. The measured motion about the vertical axis shows that the reaction wheel initially spins up in order to get the spacecraft to its desired position of 0.5rad. It then corrects itself once it overshoots to bring it back to 0.45rad. As mentioned above, this discrepancy in desired position is most likely due to the actual physical hardware having some imperfections.

4.3.8 Reaction Wheel Behavior

The behavior of the reaction wheel during the step response is that it speeds up at first to get to the desired reference height. Once it slightly overshoots, the reaction wheel makes a slight adjustment to correct and go back to the desired reference position. Figure 22 in Appendix A shows the actual control design response. This makes sense due to the fact that the actual torque being applied at first is not the maximum due to the reaction wheel having to spin up and matches the data for the actual time history of the torque. The initial current being applied to the system has to increase over time, which corresponds to the torque increasing over time.

5 Acknowledgements

Name	Plan	Model	Experiment	Results	Report	Code	ACK
Jared Steffen	1	1	1	2	2	2	X
Charles Absher	1	2	2	1	1	1	X
Eric Egaas	2	1	2	1	1	1	X
Matt Turner	2	2	1	1	2	1	X

6 Appendix

6.1 Appendix A: Graphs

6.1.1 Bias and Adjusted Scale Factor

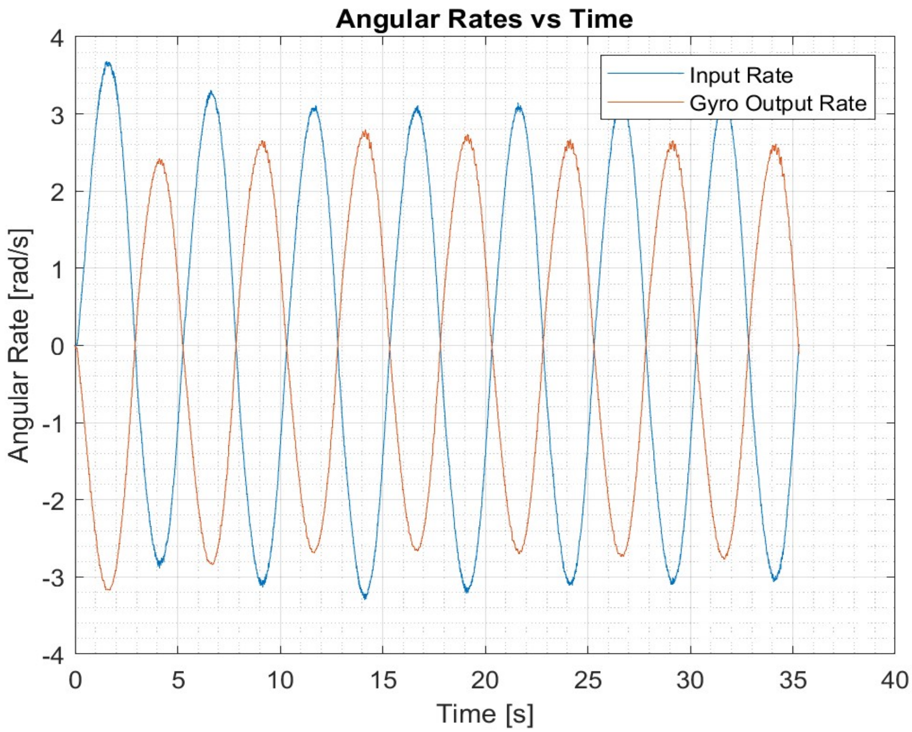

Figure 8: Time History of Angular Rates

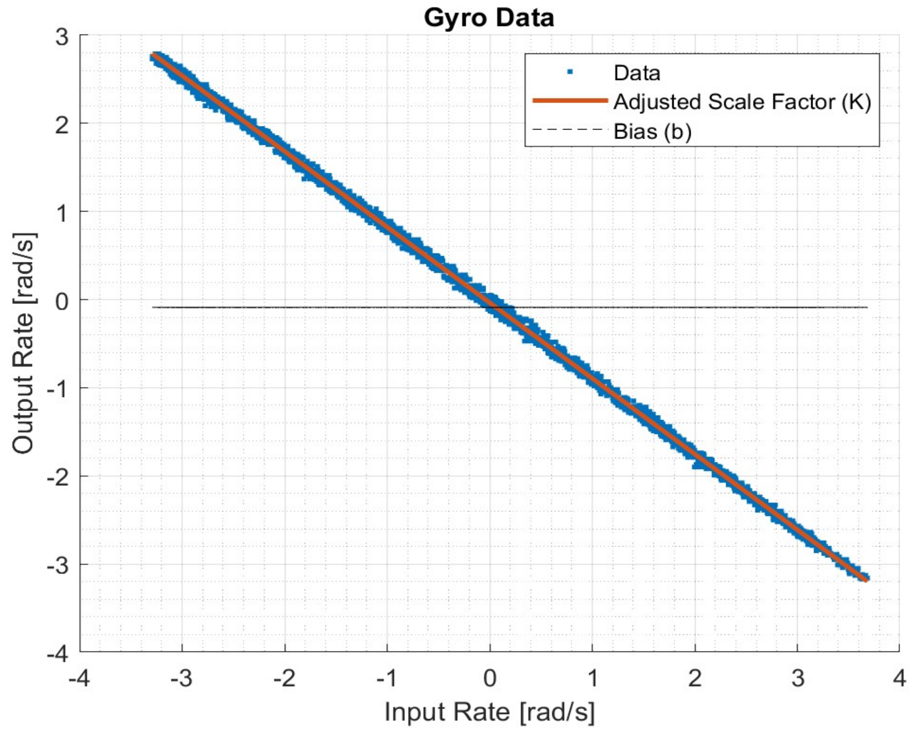

Figure 9: Input Rate vs Gyro Output

6.1.2 Angular Rates and Position with 0.2 Hz and 0.5 Amps

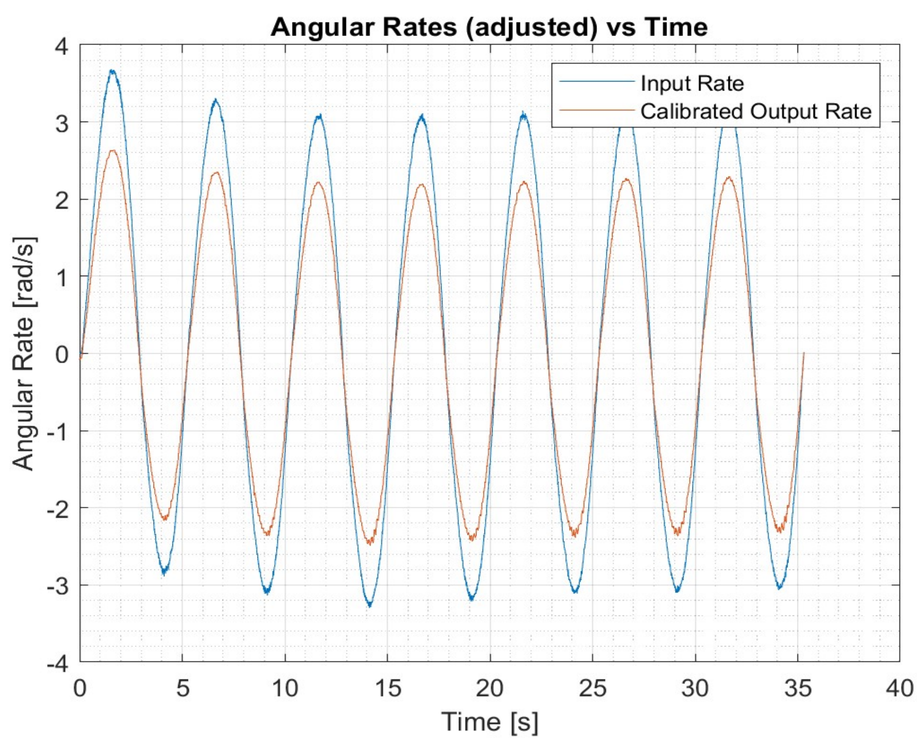

Figure 10: Time History of Adjusted Angular Rates

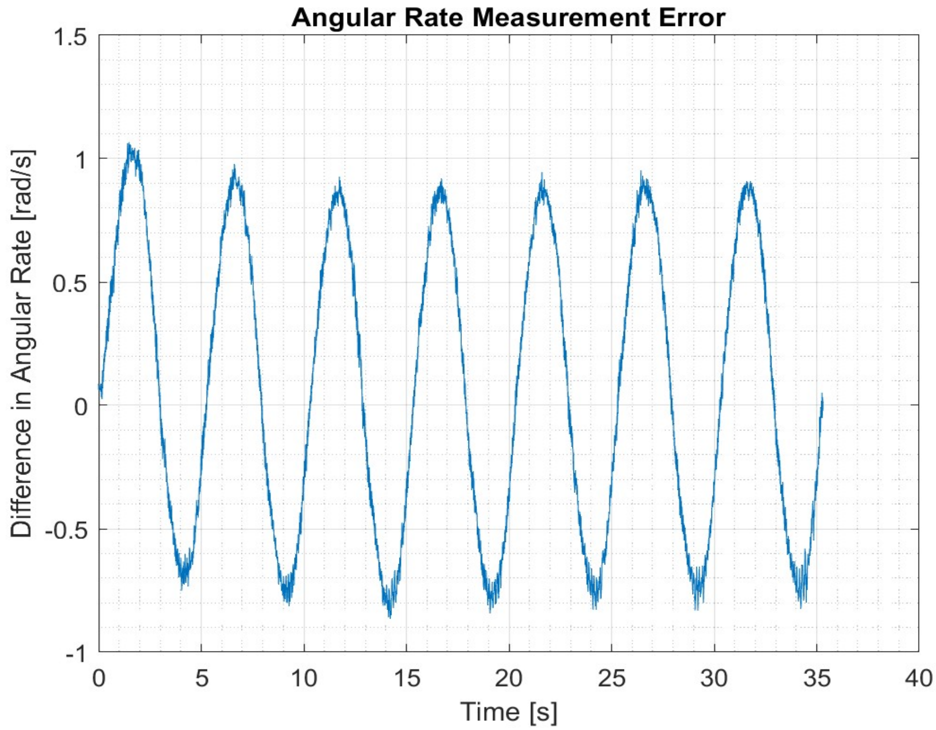

Figure 11: Time History of Adjusted Angular Rate Error

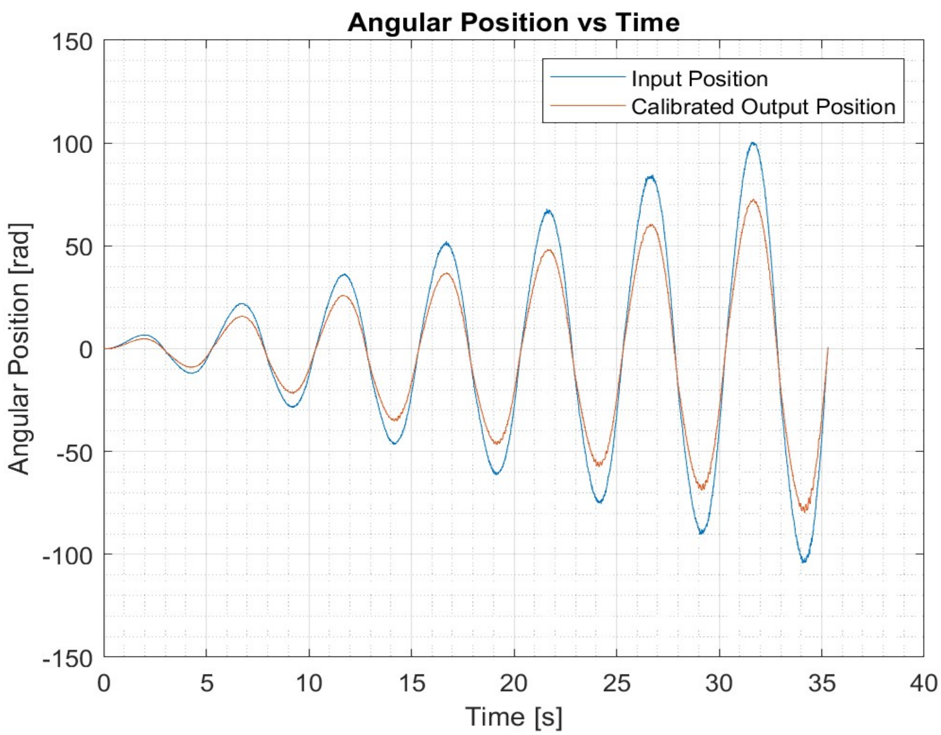

Figure 12: Time History of Adjusted Angular Position

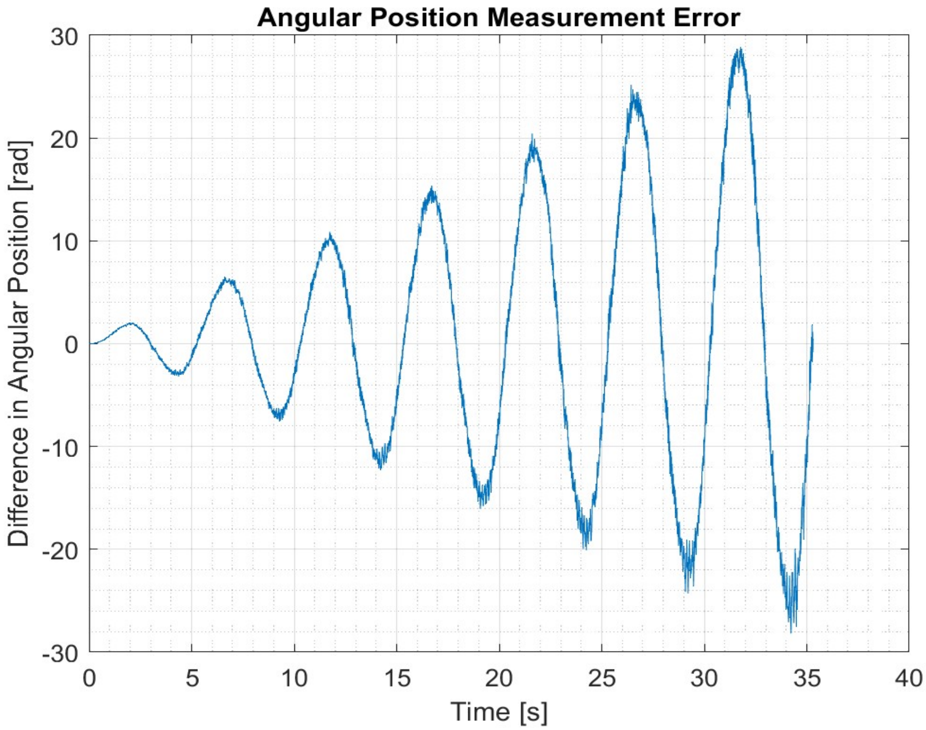

Figure 13: Time History of Adjusted Angular Position Error

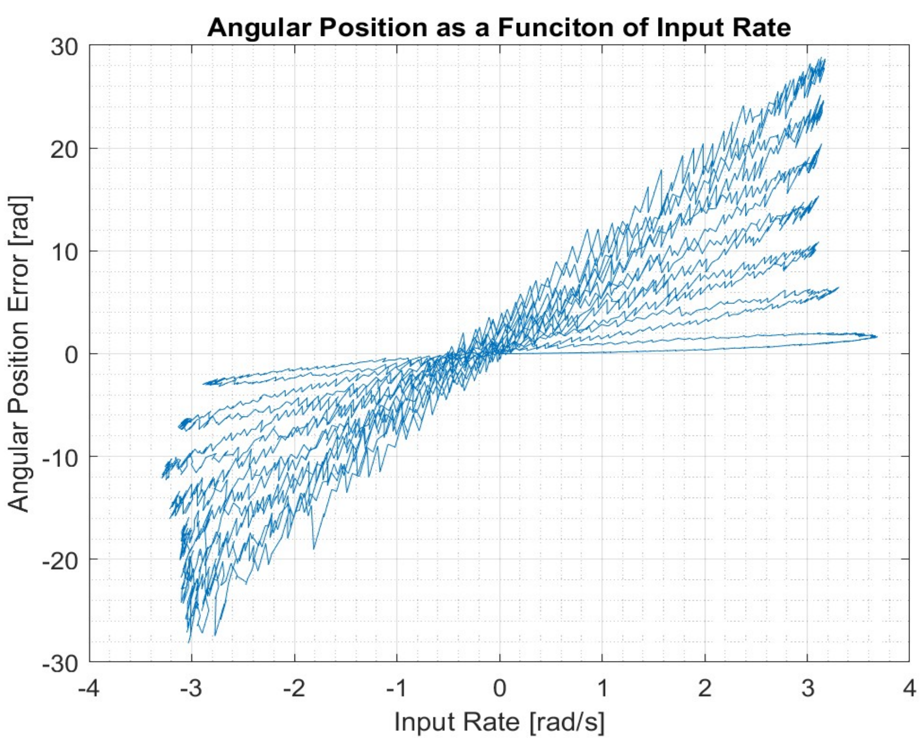

Figure 14: Angular Position Error as a Function on Input Rate

6.1.3 Angular Rates and Position with 1 Hz and 1 Amp

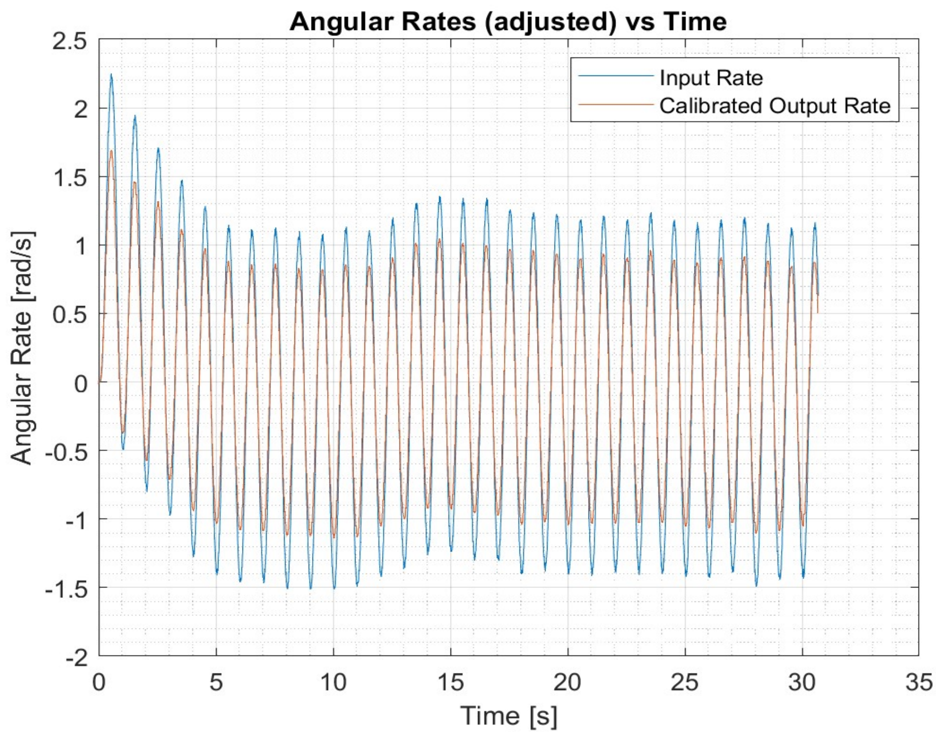

Figure 15: Time History of Adjusted Angular Rates

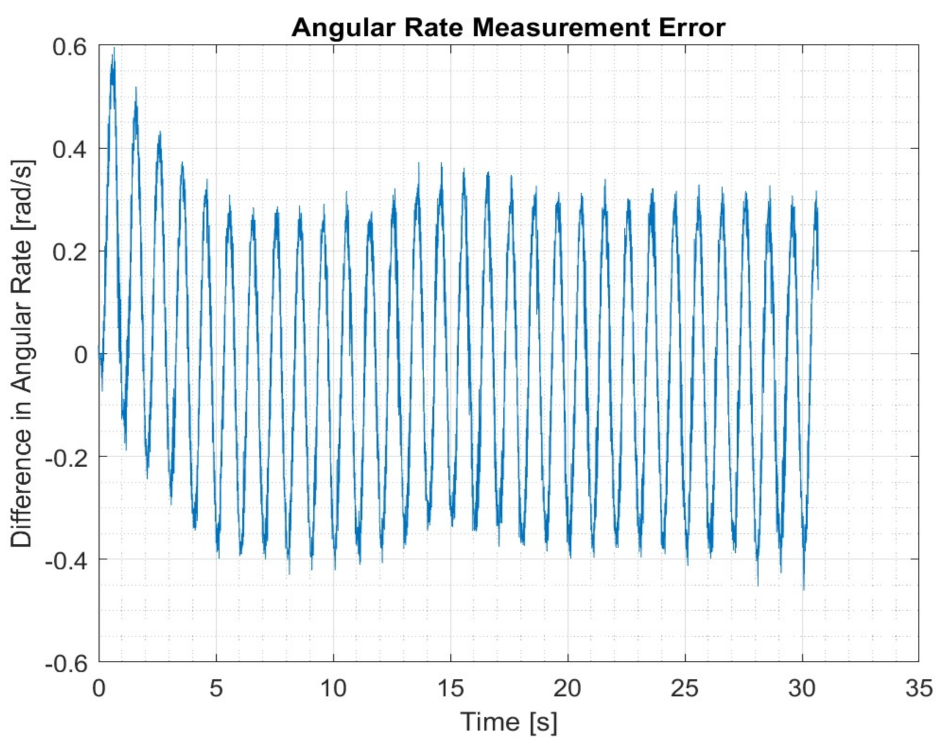

Figure 16: Time History of Adjusted Angular Rate Error

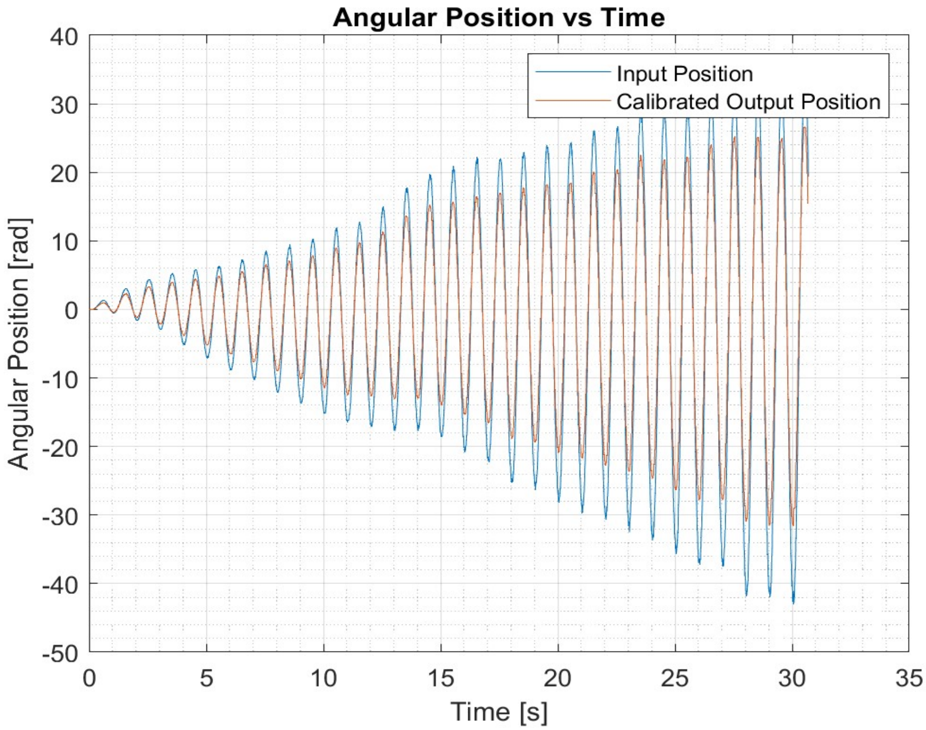

Figure 17: Time History of Adjusted Angular Position

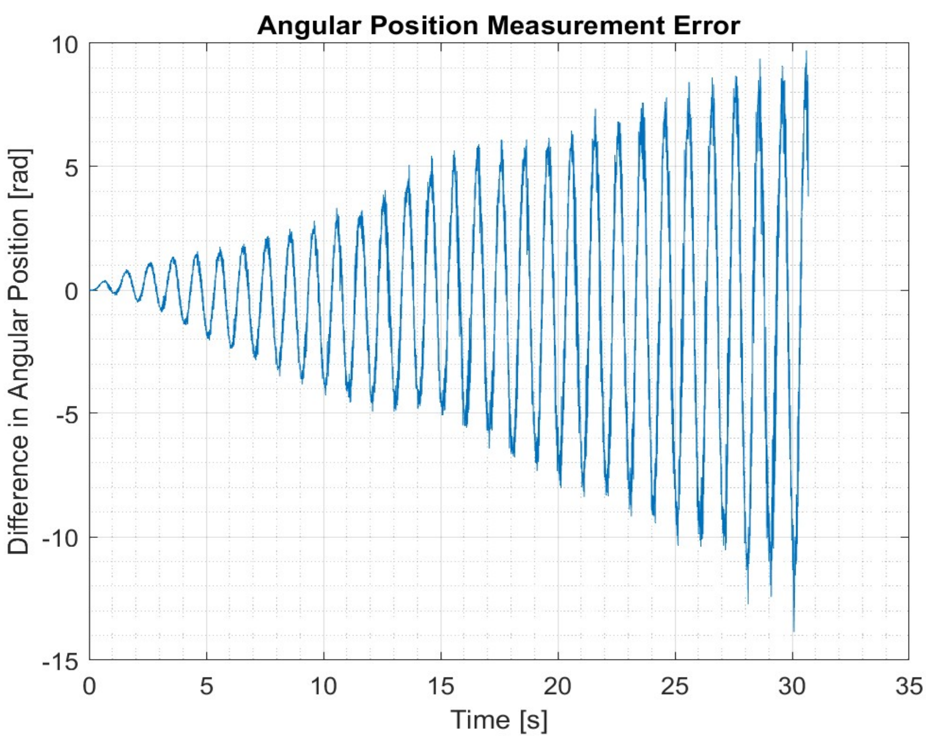

Figure 18: Time History of Adjusted Angular Position Error

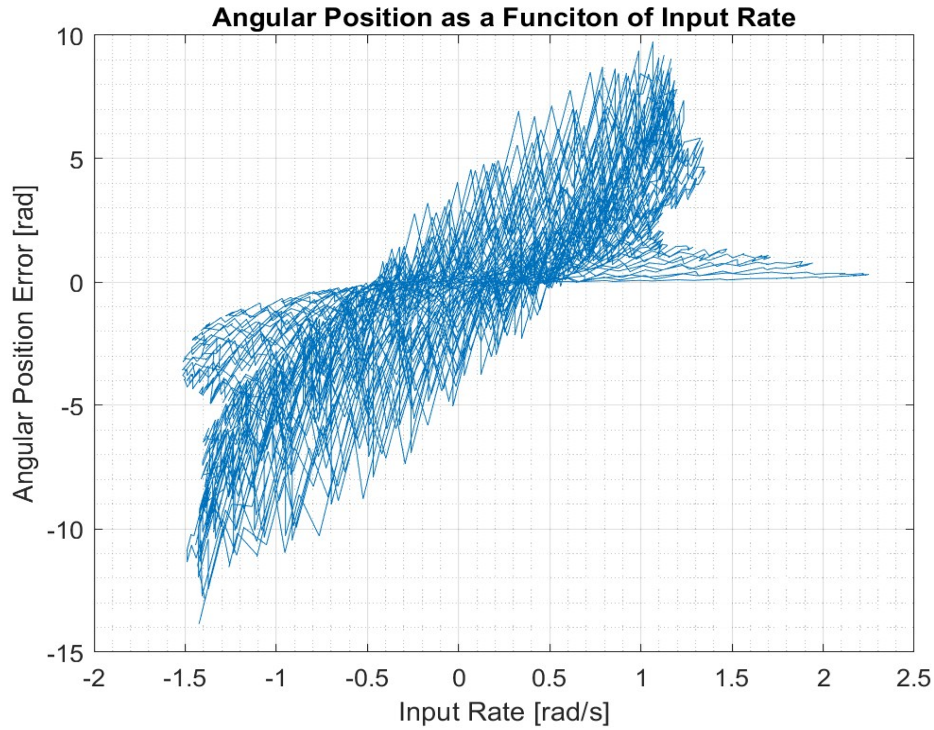

Figure 19: Angular Position Error as a Function on Input Rate

6.1.4 Reaction Wheel Characterization

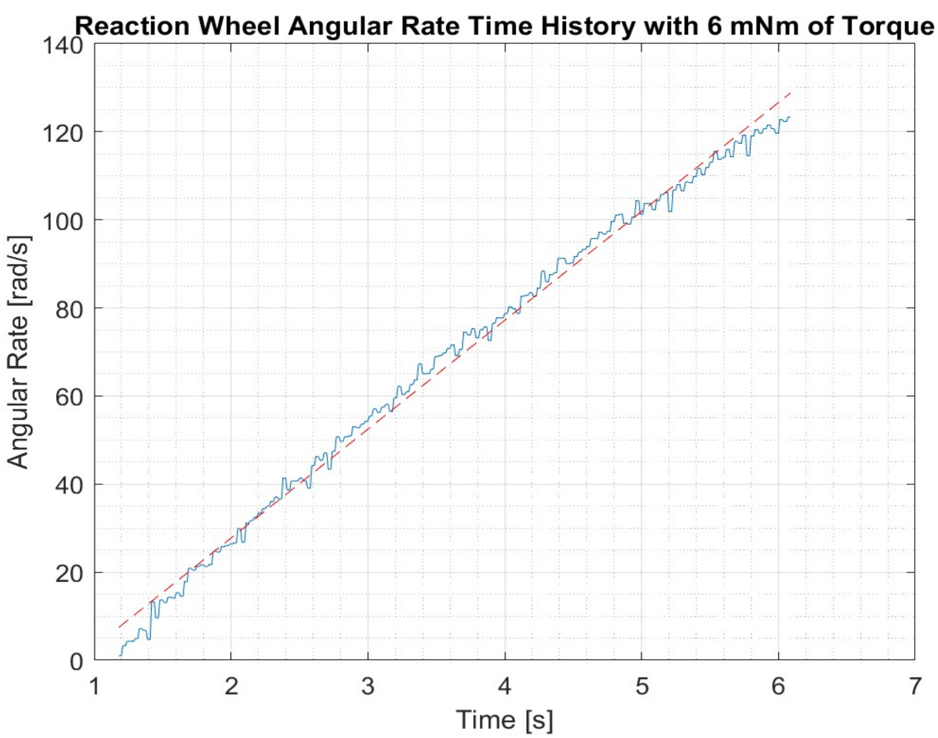

Figure 20: Time History of Reaction Wheel with $6mNm$ of Torque

6.1.5 Designed Control Gains

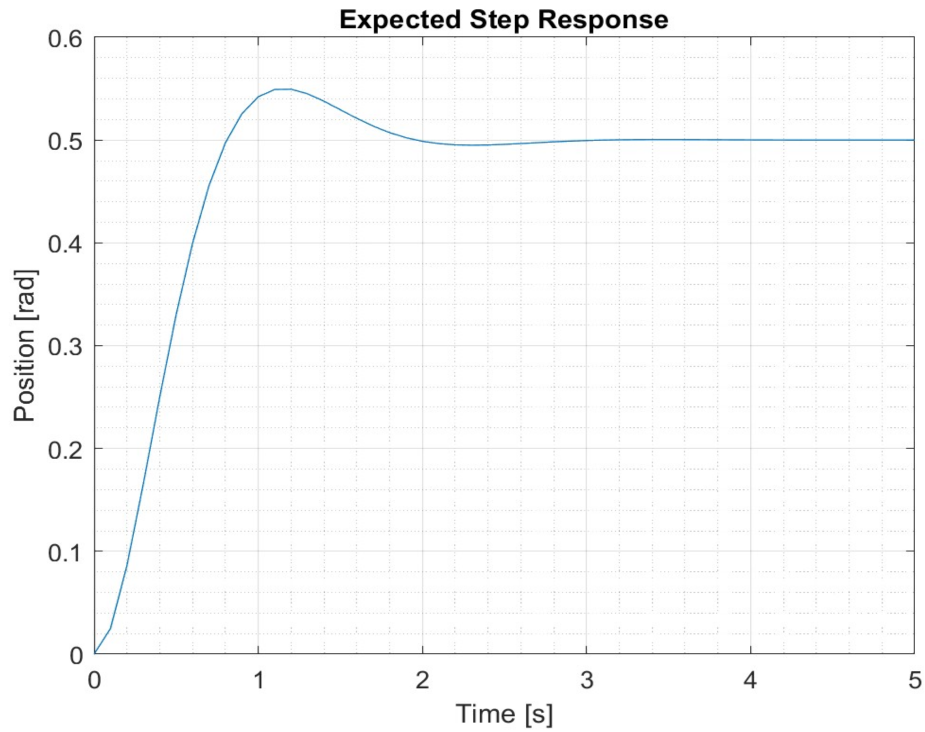

Figure 21: Expected Step Response Using MATLAB Step Function

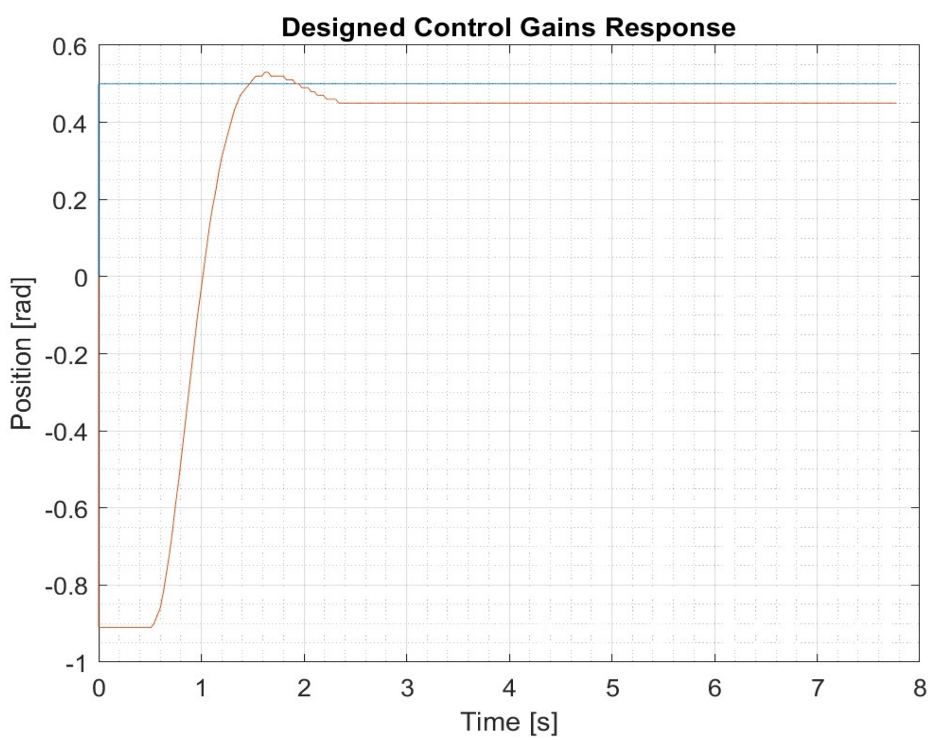

Figure 22: Actual Step Response

6.2 Appendix B: MATLAB Code

6.2.1 Angular Rates and Position with 0.2 Hz and 0.5 Amps

```
clc; clear

% Load Data
data = load('2023_10_13_002_RWHEEL_05_02');
time = data(:,1); %[s]
gyro_output = data(:,2); %[rad/s]
input = data(:,3); %[rpm]

% Adjust Time Vector
time = time - 13523.26;
time(1) = 0;

% Convert rpm -> rad/s
input = input./60; %[rps]
input = input * 2*pi;

% Plot vs Time
figure();
plot(time,input)
hold on
plot(time,gyro_output)
hold off
grid on;grid minor
xlabel('Time [s]')
ylabel('Angular Rate [rad/s]')
legend('Input Rate','Gyro Output Rate')
title('Angular Rates vs Time')

% Plot Input vs Output
figure();
scatter(input,gyro_output,'.')
hold on
xlabel('Input Rate [rad/s]')
ylabel('Output Rate [rad/s]')
title('Gyro Data')
grid on; grid minor

% Determine K
p = polyfit(input,gyro_output,1);
K = p(1).*input + p(2);

% Plot K
plot(input,K,'LineWidth',2)

% Detrmine b
b = mean(gyro_output);
b = b + zeros(length(input),1);

% Plot b
plot(input,b,'--k')
legend('Data','Adjusted Scale Factor (K)','Bias (b)')
hold off
```

```
% Adjust Gyro Measurements
adj_gyro = p(1)*gyro_output + b;

% Plot New vs Time
figure();
plot(time,input)
hold on
plot(time,adj_gyro)
legend('Input Rate','Calibrated Output Rate')
hold off
grid on; grid minor
xlabel('Time [s]')
ylabel('Angular Rate [rad/s]')
title('Angular Rates (adjusted) vs Time')

% Find and Plot Differences - Rate
difference_rate = input - adj_gyro;

figure();
plot(time,difference_rate)
title('Angular Rate Measurement Error')
xlabel('Time [s]')
ylabel('Difference in Angular Rate [rad/s]')
grid on; grid minor;

% Find and Plot Angular Position
input_pos = input .* time;
adj_gyro_pos = adj_gyro .* time;

figure();
plot(time,input_pos)
hold on
plot(time,adj_gyro_pos)
hold off
grid on; grid minor
title('Angular Position vs Time')
ylabel('Angular Position [rad]')
xlabel('Time [s]')
legend('Input Position','Calibrated Output Position')

% Find and Plot Differences - Position
difference_pos = input_pos - adj_gyro_pos;

figure();
plot(time,difference_pos)
title('Angular Position Measurement Error')
xlabel('Time [s]')
ylabel('Difference in Angular Position [rad]')
grid on; grid minor

% Angular Position as a Funciton of Input Rate
figure();
plot(input,difference_pos)
grid on; grid minor
title('Angular Position as a Funciton of Input Rate')
xlabel('Input Rate [rad/s]')
```

```
ylabel('Angular Position Error [rad]')
```

6.2.2 Angular Rates and Position with 1 Hz and 1 Amp

```
clc; clear

% Load Data
data = load('2023_10_13_002_RWHEEL_1_1');
time = data(:,1); %[s]
gyro_output = data(:,2); %[rad/s]
input = data(:,3); %[rpm]

% Adjust Time Vector
time = time - 13937.54;
time(1) = 0;

% Convert rpm -> rad/s
input = input./60; %[rps]
input = input .* 2*pi;

% Plot vs Time
figure();
plot(time,input)
hold on
plot(time,gyro_output)
hold off
grid on;grid minor
xlabel('Time [s]')
ylabel('Angular Rate [rad/s]')
legend('Input Rate','Gyro Output Rate')
title('Angular Rates vs Time')

% Plot Input vs Output
figure();
scatter(input,gyro_output,'.')
hold on
xlabel('Input Rate [rad/s]')
ylabel('Output Rate [rad/s]')
title('Gyro Data')
grid on; grid minor

% Determine K
p = polyfit(input,gyro_output,1);
K = p(1).*input + p(2);

% Plot K
plot(input,K,'LineWidth',2)

% Detrmine b
b = mean(gyro_output);
b = b + zeros(length(input),1);

% Plot b
plot(input,b,'--k')
legend('Data','Adjusted Scale Factor (K)','Bias (b)')
hold off
```

```
% Adjust Gyro Measurements
adj_gyro = p(1)*gyro_output + b;

% Plot New vs Time
figure();
plot(time,input)
hold on
plot(time,adj_gyro)
legend('Input Rate','Calibrated Output Rate')
hold off
grid on; grid minor
xlabel('Time [s]')
ylabel('Angular Rate [rad/s]')
title('Angular Rates (adjusted) vs Time')

% Find and Plot Differences - Rate
difference_rate = input - adj_gyro;

figure();
plot(time,difference_rate)
title('Angular Rate Measurement Error')
xlabel('Time [s]')
ylabel('Difference in Angular Rate [rad/s]')
grid on; grid minor;

% Find and Plot Angular Position
input_pos = input .* time;
adj_gyro_pos = adj_gyro .* time;

figure();
plot(time,input_pos)
hold on
plot(time,adj_gyro_pos)
hold off
grid on; grid minor
title('Angular Position vs Time')
ylabel('Angular Position [rad]')
xlabel('Time [s]')
legend('Input Position','Calibrated Output Position')

% Find and Plot Differences - Position
difference_pos = input_pos - adj_gyro_pos;

figure();
plot(time,difference_pos)
title('Angular Position Measurement Error')
xlabel('Time [s]')
ylabel('Difference in Angular Position [rad]')
grid on; grid minor

% Angular Position as a Funciton of Input Rate
figure();
plot(input,difference_pos)
grid on; grid minor
title('Angular Position as a Funciton of Input Rate')
xlabel('Input Rate [rad/s]')
```

```
ylabel('Angular Position Error [rad]')
```

6.2.3 Reaction Wheel Characterization

```
clc
clear

% Load Data
data1 = load("2023_10_13_002_RWHEEL_INERTIA_6");
data2 = load("2023_10_13_002_RWHEEL_INERTIA_8");
data3 = load("2023_10_13_002_RWHEEL_INERTIA_10");
data4 = load("2023_10_13_002_RWHEEL_INERTIA_12");
data5 = load("2023_10_13_002_RWHEEL_INERTIA_14");

% Times [ms] and Angular Rates [rpm]
time1 = data1(:,1);
time2 = data2(:,1);
time3 = data3(:,1);
time4 = data4(:,1);
time5 = data5(:,1);

ang_rate1 = data1(:,3);
ang_rate2 = data2(:,3);
ang_rate3 = data3(:,3);
ang_rate4 = data4(:,3);
ang_rate5 = data5(:,3);

% Adjust Units for Time [s] and Angular Rates [rad/s]
time1 = time1./1000;
time2 = time2./1000;
time3 = time3./1000;
time4 = time4./1000;
time5 = time5./1000;

ang_rate1 = (2 * pi .*ang_rate1) ./ 60;
ang_rate2 = (2 * pi .*ang_rate2) ./ 60;
ang_rate3 = (2 * pi .*ang_rate3) ./ 60;
ang_rate4 = (2 * pi .*ang_rate4) ./ 60;
ang_rate5 = (2 * pi .*ang_rate5) ./ 60;

% Remove Data
time1 = time1(118:609);
ang_rate1 = ang_rate1(118:609);
time2 = time2(118:609);
ang_rate2 = ang_rate2(118:609);
time3 = time3(118:609);
ang_rate3 = ang_rate3(118:609);
time4 = time4(118:609);
ang_rate4 = ang_rate4(118:609);
time5 = time5(118:609);
ang_rate5 = ang_rate5(118:609);

% Find Angular Accelerations
p1 = polyfit(time1,ang_rate1,1);
p2 = polyfit(time2,ang_rate2,1);
p3 = polyfit(time3,ang_rate3,1);
p4 = polyfit(time4,ang_rate1,1);
```

```
p5 = polyfit(time5,ang_rate1,1);

% Plot 1 Trial and Best Fit Line
figure();
plot(time1,ang_rate1)
hold on
K = p1(1).*time1 + p1(2);
plot(time1,K,'--r')
xlabel('Time [s]')
ylabel('Angular Rate [rad/s]')
grid on; grid minor
title('Reaction Wheel Angular Rate Time History with 6 mNm of Torque')

% Calculate RW MOI
torque1 = 6; %[mNm]
RW_MOI_1 = torque1/p1(1); %[gm^2]
torque2 = 8; %[mNm]
RW_MOI_2 = torque2/p2(1); %[gm^2]
torque3 = 10; %[mNm]
RW_MOI_3 = torque3/p3(1); %[gm^2]
torque4 = 12; %[mNm]
RW_MOI_4 = torque4/p4(1); %[gm^2]
torque5 = 14; %[mNm]
RW_MOI_5 = torque5/p5(1); %[gm^2]

% Calculate MOI STD and Mean
MOI_vec = [RW_MOI_1 RW_MOI_2 RW_MOI_3 RW_MOI_4 RW_MOI_5];
stand_dev = std(MOI_vec)
avg = mean(MOI_vec)
```

6.2.4 Spacecraft MOI Calculation

```
clc; clear

% Load and Break Up Data
data = load("2023_10_13_002_SC_INERTIA");
time = data(:,1); %[ms]
angular_vel = data(:,3); %[rpm]

% Convert Data
time = time ./ 1000; %[s]
angular_vel = angular_vel ./ 60;
angular_vel = 2 * pi .* angular_vel; %[rad/s]

% Remove Data Before 1 second and Post 6 Seconds
time = time(101:601);
angular_vel = angular_vel(101:601);

% Best Fit Line
p = polyfit(time,angular_vel,1);

% Angular Acceleration (slope)
angular_accel = abs(p(1)) %[rad/s^2]

% Torque
torque = 10 %[mNm]
```

```
SC_MOI = torque/angular_accel
```

6.2.5 Spacecraft Controls

```
clc
clear

% Expected Result in MATLAB
num = 5.709;
den = [1 3.9938 11.4181];
sys = tf(num,den);
t = [0:0.1:5];
figure();
plot(t,step(sys,t))
ylabel('Position [rad]')
xlabel('Time [s]')
grid on; grid minor
title('Expected Step Response')

% Load Data
data = load("Tested_Gains3");
time = data(:,1); %[ms]
ref_pos = data(:,2); %[rad]
meas_pos = data(:,3); %[rad]

% Adjust/Convert Time Vector
time = time - time(2);
time(1) = 0;
time = time ./ 1000; %[s]

% Plot Data
figure();
plot(time,ref_pos)
hold on
plot(time,meas_pos)
hold off
ylabel('Position [rad]')
xlabel('Time [s]')
grid on; grid minor
title('Designed Control Gains Response')
```