University of Colorado - Boulder

ASEN 3200 - Orbital Mechanics/Attitude Dynamics and Control

Section 011, Group 8

Lab 1: Attitude Sensors and Actuators

Author:
Tomaz Remec[a]

Author:
Hunter Daboll[b]

Author:
Connor O'Reilly[c]

[a]107061117
[b]108976705
[c]107054811

Professors:
Penina Axelrad
Bobby Hodgkinson

09/14/2020

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO BOULDER

The objective of this lab was to collect and characterize and utilize rate gyro data, to calculate Moment of Inertia of a reaction wheel based on lab data, and to analyze and qualitatively comprehend the behavior of spinning objects undergoing external torques (in the form of the control moment and physical rate gyros). It was found that with known data processing methods, the rate gyro could accurately measure angular velocities of a mockup spacecraft and that the reaction wheel of this mockup could absorb up to $89.221\ \frac{Nm}{s}$ of angular momentum. It was also shown that an external torque on a rotating body produces an orthogonal procession, and that this can be utilized to track moving objects from a rotating vantage point.

Introduction

In this lab, the group investigated the principles of gyroscopes and their relationship to attitude control and dynamics. A MEMS rate gyro mounted on a spacecraft mockup was analyzed to determine its effectiveness in determining the angular velocity of the mockup. A reaction wheel on the same spacecraft mockup was also analyzed to quantify its use in controlling the angular position and velocity of the mockup. Then the group moved to the more tactile (and slightly more entertaining) portion of the lab. A small physical rate gyro was qualitatively analyzed to teach the effects of precession and how to determine it numerically. Then finally, a bicycle wheel was used as a control moment gyro (CMG) for a group member standing on a low friction platform to further demonstrate precession and its effects on a large scale. These experiments all relate closely to spacecraft attitude and control as they are used extensively in the real world as control and orientation determination mechanisms on spacecraft where it is easier or more efficient to have a control system that does not burn any fuel or consume any other resources besides electrical power.

I. Preliminary Questions

1. To calculate angular position using rate gyro measurements, you would first need to analyze the voltage values given by the gyro and find what angular velocity corresponds to a given input voltage. Then, you would numerically integrate this velocity set over the given time set in order to get the total change in angular position. To this change you would add the original angular position (known value) to get your new angular position.

2. We know that torque is equal to the moment of inertia of the reaction wheel multiplied by the angular acceleration of the wheel. We also know that the moment of inertia of the wheel is equal to $I\omega$. Substituting and solving for the angular acceleration gives us an equation for the angular acceleration required to counter the moment applied to the spacecraft.

$$\alpha = \frac{M_d}{I\omega} \tag{1}$$

3. When rotating a spinning gyroscope in your hand, you can expect to feel the effects of gyroscopic precession. You will feel the gyroscope want to turn in directions orthogonal to the direction you are trying to turn it.

4. In order to control yourself by using a spinning bicycle wheel as a Control Moment Gyro, you would tilt the wheel from side to side, changing the direction of its angular momentum vector. This will cause you to rotate to following the equation of precession.

$$\vec{M}_G = \frac{d\vec{H}_G}{dt} + \vec{\omega}_{A/I} \times \vec{H}_G \tag{2}$$

Starting with a bicycle wheel spinning with angular momentum vector to the left, you would raise your right hand and lower your left. Doing this will impart a torque unto the wheel whose vector points at you. This will cause precession in the wheel-person system that obeys the above equation, resulting in the system rotating to the left.

II. Experiment

A. MEMS Rate Gyro Characterization

1. Equipment

- ASEN Spin Module
  - myRIO
  - Shaft Encoder
  - Motor Controller and 30W Base Motor
  - 2 XBEE radios
  - Arduino Due
  - DC Voltage Converter and 9 Volt battery pack
  - Full 9 axis IMU (MEMS Gyro)
- Other Equipment
  - Main Computer running LabVIEW VI

The Main computer running the spin module labVIEW.vi software lets the user chose between the manual or automatic input control. If automatic input control is selected the user will chose the desired Control Current (Amps) and Control Frequency(Hz.) which will cause the base motor to move the spacecraft in a sinusoidal pattern. The Control Current and Frequency are passed to the myRIO device and a torque is applied to the the 30W DC motor on the base of the spacecraft using the motor controller. The true rotation rate and position of the spacecraft are measured by the shaft encoder located on the 30W motor and passed back to the main computer through the myRIO device. The full 9 axis IMU located on the top of the spin module includes a MEMS gyro which measures the un-calibrated angular velocity of the spacecraft in rad/s. This is powered by a 9v battery pack which supplies 12v of power to the arduino Due using a voltage converter. The recorded angular velocity and time are sent back to the main computer using an XBEE radio system attached to both the myRIO device and the arduino DUE which relay commands and data between the two devices. If manual control is selected no control current of frequency commands are sent to the spin module and only the shaft encoder and IMU data are recorded. This process is documented in the block diagram for this system, which can be found in Appendix A.

2. Procedure

The spin module labVIEW.vi is opened on the main computer and the gyro tab is selected, allowing gyro experiments to be recorded. Initially, manual experiments are run by setting the VI to "Manual Control"; data is then recorded by selecting the "begin test" button. Using our hands, the top of the spin module is moved with a slow, steady, sinusoidal input for around a minute. The Shaft Encoder and Gyro data is downloaded and exported to a text file, and observations of data collection are noted for different rotation rates.

Following the manual control experiments, the spin module VI was set to "automatic input control," with 9 different inputs for the control current and frequency to produce sinusoidal movement of the spacecraft. Data is then collected, downloaded and exported to separate text files. After the data for both experiments is collected, the bias and scale factor $\boldsymbol{K}$ are computed. These values are used to calibrate the MEMS gyro measurements and used to compute the angular rate and position of the spacecraft based on the MEMS gyro measurements.

3. Theory

To calculate the rate gyro bias and the adjusted scale factor, the MEMS gyro output is compared to the actual motor encoder velocity for a range of different spacecraft angualr velocities. The code used for this analysis can be found in Appendix C. Initially the data is imported into Matlab using built in functions *readtable()* and *table2array* which will create a table by reading column oriented data from a file and then converting it into a matrix for ease of use. Afterwards the MEMS gyro output in RPM is converted to rad/s by multiplying the gyro output by a value defined as $RPM * \frac{2\pi rad}{1 revolution} * \frac{1min}{60sec} \approx 0.104272$. The converted gyro data is then plotted on the y-axis against the encoder velocity on the x-axis using *plot()* and *polyfit()* along with *polyval* are used to fit a line to the data. The y-intercept of each run represents the bias of the gyro, while the scale factor **K** is the average of the ratio between the encoder and gyro angular velocity for all time steps.

$$K = mean\left(\frac{\omega_{encoder}(t)}{\omega_{gyro\_output}(t)}\right)$$

After determining the Scale factor **K** and the gyro bias for every run, the average of both values are used to measure gyro angular velocity

$$\omega_{gyro} = \omega_{gyro\_output} * mean(K) - mean(bias_{gyro})$$

where $mean(K)$ and $mean(bias_{gyro})$ represent the average scale factor and the average gyro bias for all runs. A forward Euler method is then used to determine the angular position of the spacecraft using the measured gyro angular velocity.

$$\theta(t+h) = \theta(t) + \omega_{gyro}(t) * h$$

where h represents the average change in time between each output, $\omega_{gyro}(t)$ and $\theta(t)$ are the measured gyro spin rate and the current angular position at thee current time step, and $\theta(t+h)$ is the angular position at the next time step.

4. Results

Below, we present the mean bias (and its standard deviation) between the angular velocity input and the angular velocity measured by the MEMS gyroscope.

Mean Bias $[kg\ m^2]$	0.0027
Standard Deviation σ_K	0.0019

This averaged value is not incredibly accurate: with five data sets, the standard deviation nearly reached the value of the bias itself. This is likely due in part to the short duration of each data set (on the order of 10 seconds for most data sets), and the significant variation in behavior between data collection runs. Some sets behaved calmly and sinusoidally, while others varied significantly due to changes in the input voltage and frequency.

However, the scale factor (presented below) is significantly more accurate, with a standard deviation multiple orders of magnitude smaller than the value itself.

Mean Scale Factor K	-1.1422
Standard Deviation σ_{SF}	0.0075

These values are expected to be accurate between -10 and +10 $[\frac{rad}{sec}]$, as this is the highest angular velocity obtained (see Appendix B for full plots). Below, some sample data sets are presented as requested, showing only the angular position over time as calculated with gyro output values multiplied by the scale factor and with the bias subtracted.

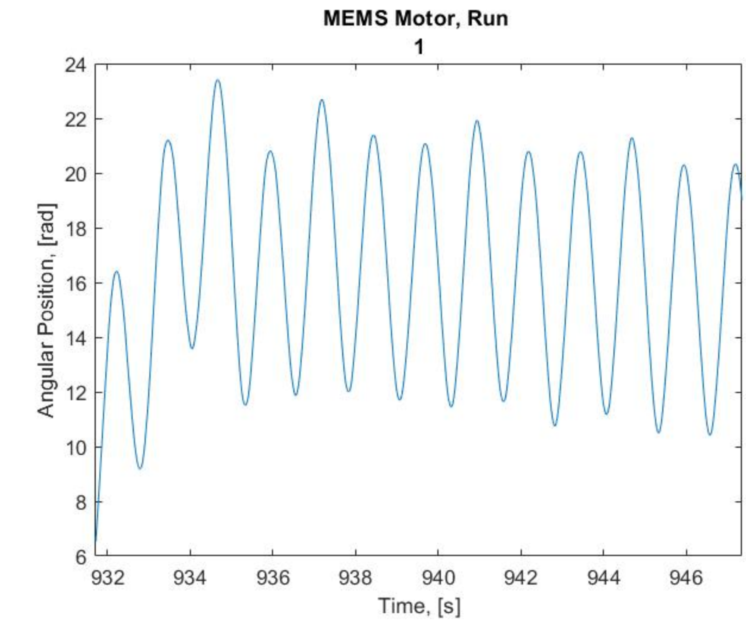


Fig. 1 Angular Position vs Time, Run 1

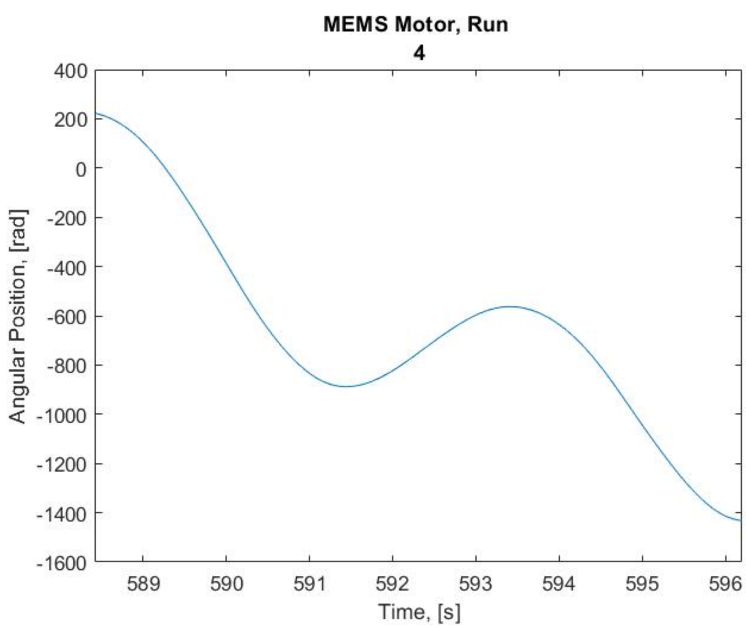


Fig. 2 Angular Position vs Time, Run 4

However, this data is more useful when context is provided with it: specifically, when the angular velocity can be compared with position over time. This side by side comparison is shown below, which was used by the authors to ensure the calculation of angular position behaved correctly.

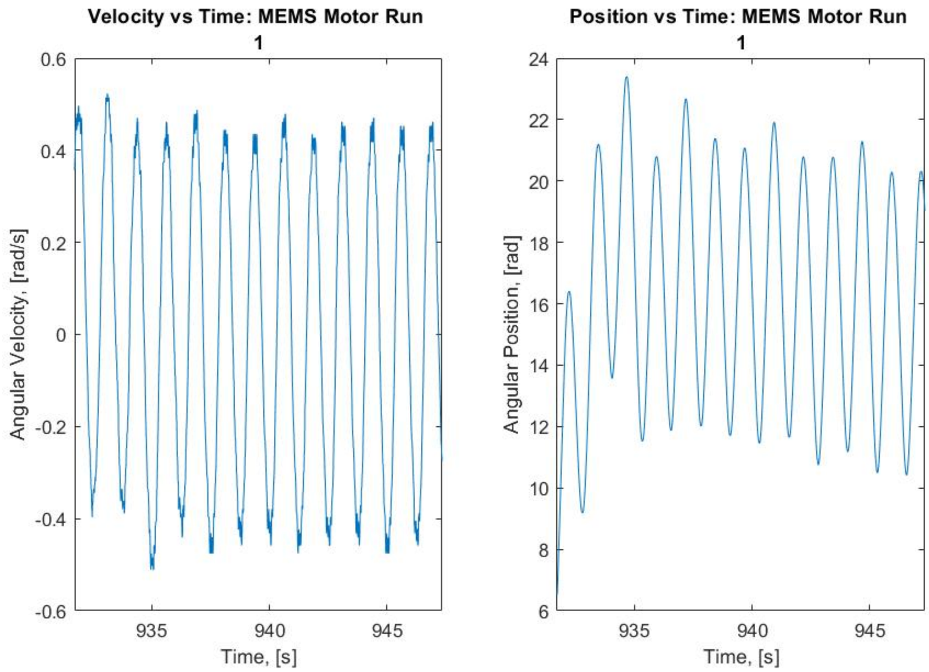


Fig. 3 Angular Velocity and Position vs Time, Run 1

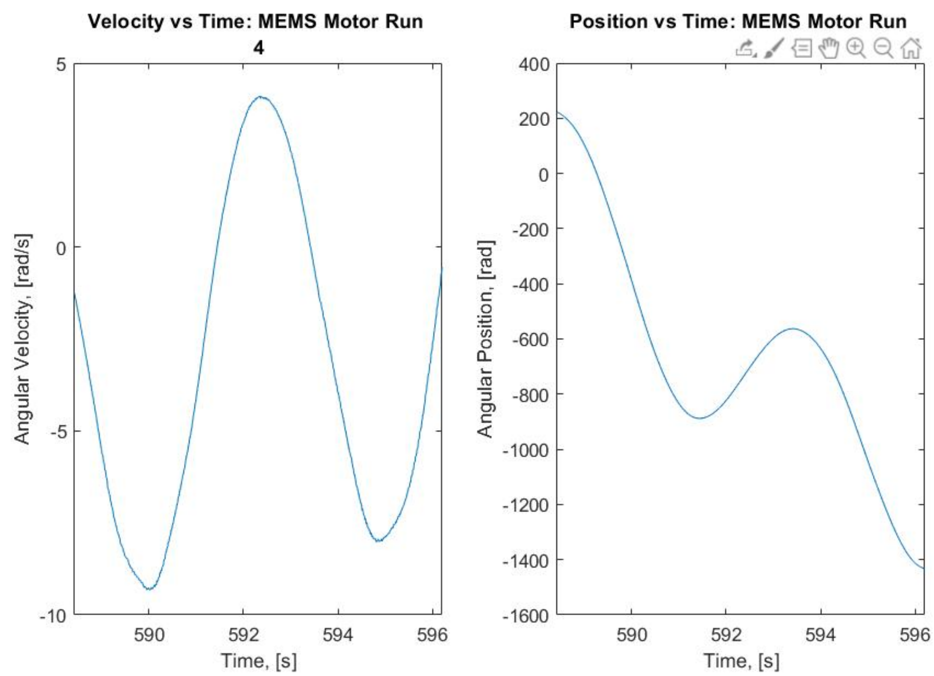


Fig. 4 Angular Velocity and Position vs Time, Run 4

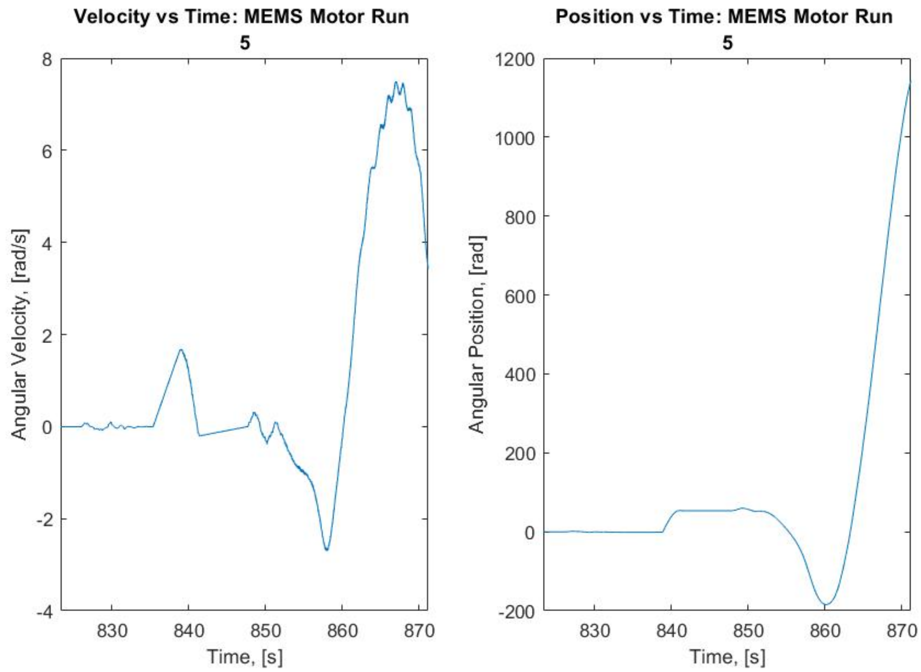


Fig. 5 Angular Velocity and Position vs Time, Run 5

5. Analysis

The side by side analysis of measured angular velocity and the integrated position do indeed behave as anticipated. Figure 4 is a fantastic example, displaying motion in the positive angular direction when velocity is positive, but displaying an overall negative angular position as the angular velocity is primarily in this direction.

From experimenting with the mockup spacecraft manually, it was clear that abrupt changes in spin rate were not accurately measured - there was inevitably a delay (caused by the angular momentum of the gyro) and a jump in the measured angular velocity. However, for smaller angular accelerations the gyro could measure angular velocity quite accurately once the data was correctly multiplied by a negative scale factor and the bias was accounted for. The calculation of the bias was not highly repeatable, but was consistently low - indicating that it is less important to drive the stadard deviation of this average to zero.

B. Reaction Wheel Characterization

1. equipment

Equipment List:

- ASEN Spin Module
  - myRIO
  - 2 XBEE radios
  - Arduino Due
  - DC Voltage Converter and 9 Volt battery pack
  - Motor Controller board
  - Reaction Wheel and Hall Effect Sensor
- Other Equipment
  - Main Computer running LabVIEW VI

Similar to the MEMS gyro characterization the spin module is used again. The spin module labVIEW.VI is opened on the main computer and the torque tab is selected. The run arrow is then clicked which begins the selected operation. In the VI the reaction wheel (hall sensor) is selected to read the angular velocity, the reaction wheel is also selected for current sensor and motor for the applied torque. The desired torque in mNm and the durration of the applied torque in seconds is entered in the VI and the apply torque button is pressed. The main computer will send this information to the myRIO device which will then send the torque commands to the arduino DUE located on the top of the space craft using an XBEE radio system. The applied torque commands are then sent to the motor controller board and are then applied to the reaction wheel. The hall Effect Sensor records the angular velocity of the reaction wheel and is communcated back to the main computer using the XBEE radio system and the myRIO device. The arduino due and motor controller are powered by a 9v battery pack which supplies 24v of power to the using a voltage converter.

2. Procedure

The spin module labVIEW.vi software for applying torque was opened and **number of trials run** different torques were applied to the reaction wheel. The spacecraft was held stationary and was prevented from interacting with the wheel as it accelerated. The output from the hall effect sensor from each trial was recorded, downloaded and exported to a matlab script. The recorded motor current in amps was converted to Nm of torque. Using the recorded data the average angular acceleration and an estimate of the moment of inertia for the wheel were computed. The code used for this analysis can be found in Appendix C.

3. Theory

Treating the reaction wheel as a rigid body we can compute the moment of inertia of the wheel. The total angular momentum for a rigid body about its center of mass is equal to the moment of inertia matrix of the body multiplied by the angular velocity vector of the same body.

$$[H_g]_B = [I_G]_B * [\vec{\omega}]_b (1)$$

Using the relations derived in class from the rotational motion EOMS, the time rate of change of the total angular momentum of a rigid body about the center of mass G is equal to the external torque applied about G.

$$[\vec{M}_G] = \frac{d\vec{H}_G}{dt} = [I_G]_B * [\vec{\alpha}]_B (1)$$

Because the reaction wheel only spins in one direction which is out of the table, the reaction wheel will only have a moment of inertia component in the Iz direction and an angular acceleration about the Z axis.

$$[\vec{M}_G] = I_{wheel} * \alpha_{wheel}$$

This allows to estimate the moment of inertia of the by applying a known torque to the wheel and measuring the resulting angular velocity.

$$I_{wheel} = [\vec{M}_G] * \alpha_{wheel}$$

4. Results

The average angular acceleration of the reaction wheel was computed for each applied torque. Using the computed value the average moment of inertia of the reaction wheel was computed with their corresponding average angular velocity. After the average moment of inertia for each trial was calculated the average and standard deviation of was computed. These results are shown below.

Applied Torque [mNm]	Avg. Angular Acceleration [$\frac{rad}{s^2}$]	Avg. MOI [kgm^2]
5	19.4241	0.2574
8	31.9791	0.2502
9	40.8163	0.2205
10	54.1274	0.1847
15	98.6573	0.1520

Table 1 Avg. Angular Acceleration and MOI for each trial

Avg. MOI of all torques	0.2130 $[kgm^2]$
Standard Deviation	0.0445

5. Analysis

My expectations were that the calculated moment of inertia of the reaction wheel would be relatively similar for each torque applied. In reality as the applied torque on the reaction wheel increased the estimated average moment of inertia ended up decreasing in size. In reality, as the applied torque of the reaction wheel increased the average angular Acceleration also increased. Because the computed MOI is inversely proportional to the angular acceleration, it would make sense that the estimated average moi of the reaction wheel would decrease as the applied torque increases.

One major possible source of error would be due to using a hand to hold the spin module stationary, a student can unknowingly interfere with the collected data by allowing the spin module to interfere with the reaction wheel or allowing the spin module to spin.

Rewriting the equation to solve for total angular momentum $\vec{M}_G = I_{wheel} * \alpha_{wheel}$ and using a forward Euler method we can determine the current angular velocity at a certain time t.

$$\omega(t + h) = \omega(t) + \alpha_{avg} * h$$

where the average angular acceleration is calculated by $\alpha_{avg} = \frac{\vec{M}_G}{I_G}$, h represents a set time step, and $\omega(t)$ the angular velocity of the wheel at time t. Using matlab it was determined the time a reaction wheel could resist an aerodynamic torque of $10^{-4}\,[Nm]$ before the reaction wheel speed exceeds the limit of 4000 rpm is 893.100 seconds.The angular momentum capacity of the wheel corresponds to product of the maximum angular velocity of the wheel and the moment of inertia. Converting the provided limit of 4000 rpm to 418.879 rad/s and using the average reaction wheel moment of inertia of 0.213 Nm the angular momentum capacity of the wheel was calculated to be 89.221 $\frac{Nm}{s}$

C. Control Moment Gyro Experiment

1. Experiment

Equipment List:
- Bicycle Wheel
- Low Friction Platform
- Electric Spinner
- Group Member(s)

Procedure: One group member stood on the low friction platform holding the bicycle wheel out in front of them. Another group member then used the electric spinner to spin the bicycle wheel up to a high rate of speed. Then the person standing on the platform applied a torque to the wheel by "turning" it left or right. The resulting rotation of the whole person/bicycle wheel system was then noted. Below is a rough diagram of the experimental setup.

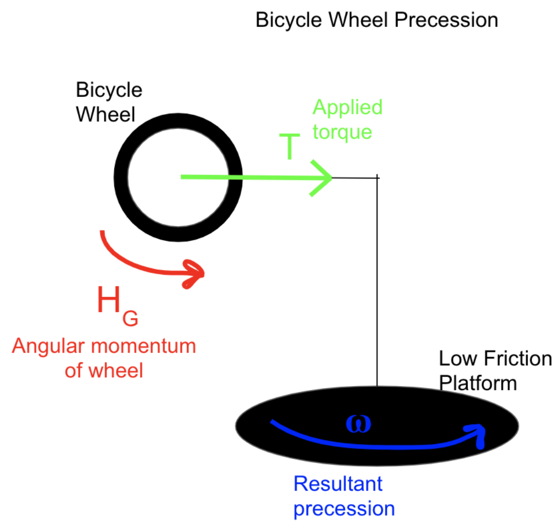


Fig. 6 CMG Experiment

2. Results & Analysis

It was found that with a bicycle wheel spinning "away" from the person on the platform, that is, a wheel with angular velocity to the left from the perspective of the holder, that "turning" the wheel to the right by raising the left hand and dropping the right will cause rotation of the system to the right, or in the counterclockwise direction when viewed from above. This is due to the gyroscopic precession of the bicycle wheel causing a torque on the system according to Equation 2. The same effect can be felt and made useful when riding a bike or motorcycle. When travelling at high speeds, it is dangerous to turn the handlebars as that might quickly cause an accident. However, simply by leaning one way or another, your body is imparting a torque on the wheels, which then precess in the direction of the lean, causing the whole vehicle to turn.

D. Physical Rate Gyro

1. Experiment

A small physical rate gyro consisting of a spinner and motor assembly contained in a box was the subject of this experiment. The spinner was activated and then manipulated in order to observe the effects of different torques and rotations on the gyro. Below is a diagram of the rate gyro used in the experiment.

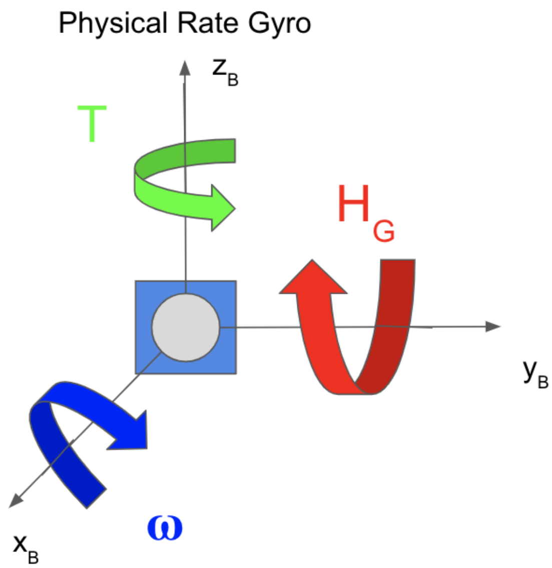


Fig. 7 Physical Rate Gyro Diagram

2. Analysis

A qualitative analysis was first performed on the gyro. It was spun up and then rotated about its three principal axes. It was observed that when rotated about the x axis, the gyro produced a torque about the z axis. This was found to be a product of the precession of the gyro, obeying both Equation 2 and the following Euler equation of motion.

$$M = I\alpha + \omega \times (I\omega) \tag{3}$$

This gyroscopic precession, while initially difficult to comprehend and visualize, is a key aspect of gyroscopes that make them as useful as they are. It allows for the use of gyroscopes in attitude and navigation systems as sensors that are sensitive to torques and changes in orientation, which are of particular use in Inertial Navigation Systems (INS).

III. Conclusions & Recommendations

Dynamic Equations for Rotational Motion, Angular Momentum, and Matlab computations were used together to characterize gyro rate data (obtained from lab data) to investigate the rate gyro and measure its capabilities for sensing angular velocity and position. These were then compared to the true characteristics of the spacecraft. Analysis of the calibrated measured angular velocity showed that abrupt changes in spin rate were not accurately measured but for smaller angular accelerations the gyro proved to be accurate once the scale factor and bias where applied. Additionally the reaction wheel actuator was investigated to compute its angular momentum capacity. The code provided performs this analysis, and produces all plots displayed in this report.

Analysis of both the physical rate gyro and the control moment gyro expanded our understanding of the causes and effects of gyroscopic precession. The physical rate gyro allowed us to determine the axis and direction of precession mathematically by providing visual axes and a single free axis for the gyro to precess around. The use of the bicycle wheel as a control moment gyro allowed us to get an even more hands on experience with the tangible effects of precession. By letting the team get our hands "dirty" so to speak, we were able to gain a much better understanding of how and when precession affects whole systems, and not just the gyroscope itself.

This lab provided significant insight and intuition into the behavior of rotating objects, particularly regarding their interaction with external moments. Some improvements to this lab could include more hands on time with the machinery involved, but that was particularly difficult due to the ongoing COVID-19 pandemic. Overall, this lab provided valuable experience that is sure to follow the group through the rest of our careers.

References

Axelrad, P., "ASEN 3200 Lab A-1: Attitude Sensors and Actuators" Aug. 25, 2020.

Curtis, H., *Orbital Mechanics for Engineering Students*, 3rd ed., Elsevier, 2014.

Acknowledgements

We would like to thank Professor Axelrad, Professor Hodgkinson, and all of the TAs of the ASEN 3200 instructional team for their continued support and guidance throughout this lab.

Team Member Participation

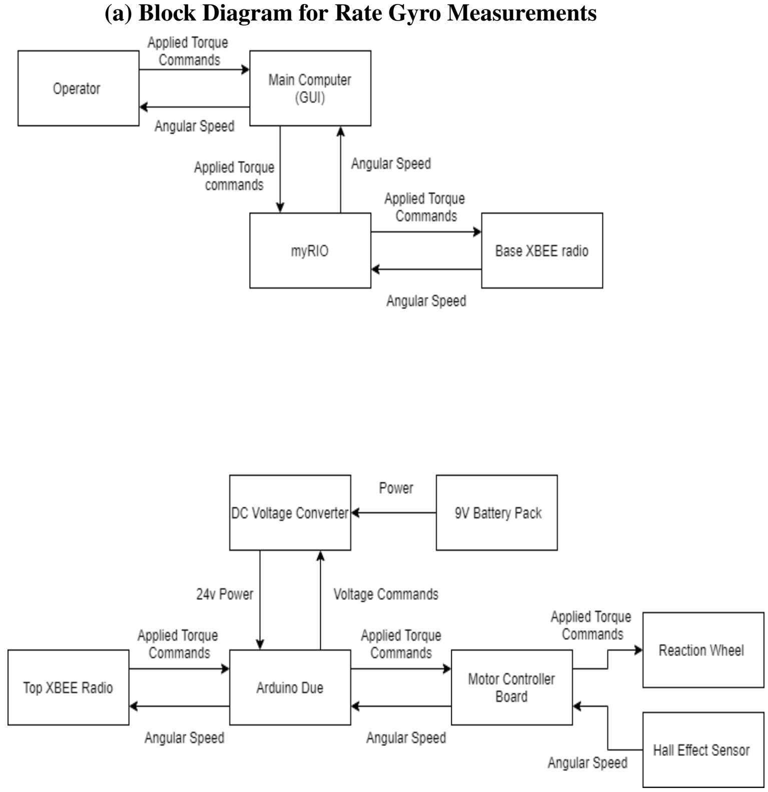
2 - Lead, 1 - Participated/Contributed, 0 - Did not Contribute

Name	Plan	Model	Experiment	Results	Code	Report
Tomaz Remec	2	1	1	2	0	1
Hunter Daboll	1	1	2	1	2	1
Connor O'Reilly	1	2	1	1	0	1

Group members have reviewed and approved this table.
Initials: WHD CTO TJR

Appendix A: Block Diagrams

(a) Block Diagram for Rate Gyro Measurements



(b) Block Diagram for Reaction Wheel Measurements

Appendix B: Full Graphs

A. Angular Position vs Time

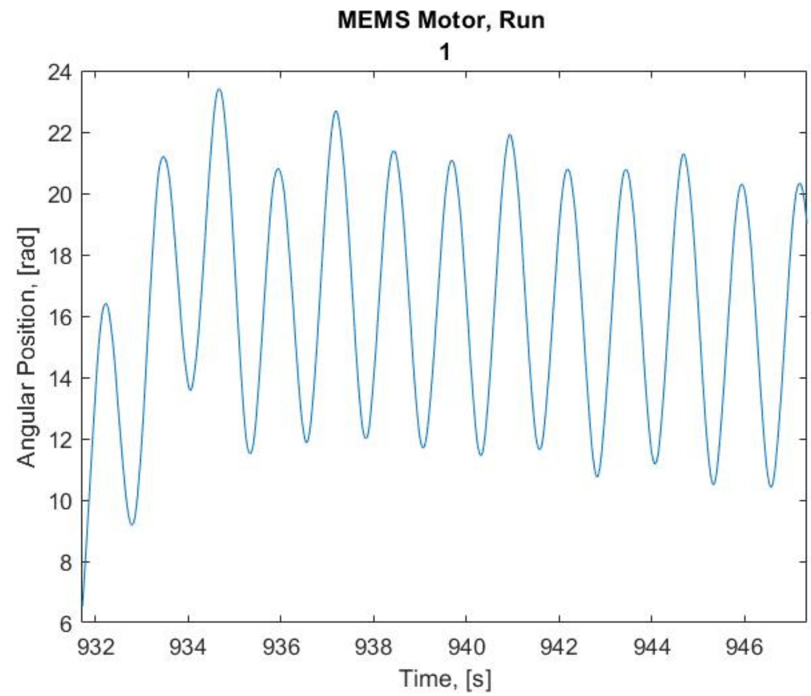


Fig. 9 Angular Position vs Time, Run 1

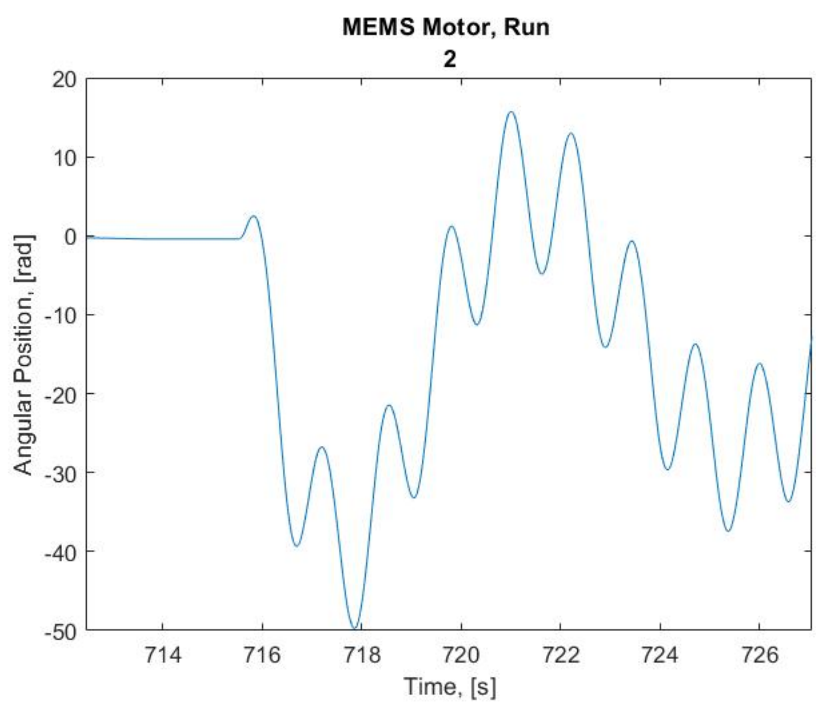


Fig. 10 Angular Position vs Time, Run 2

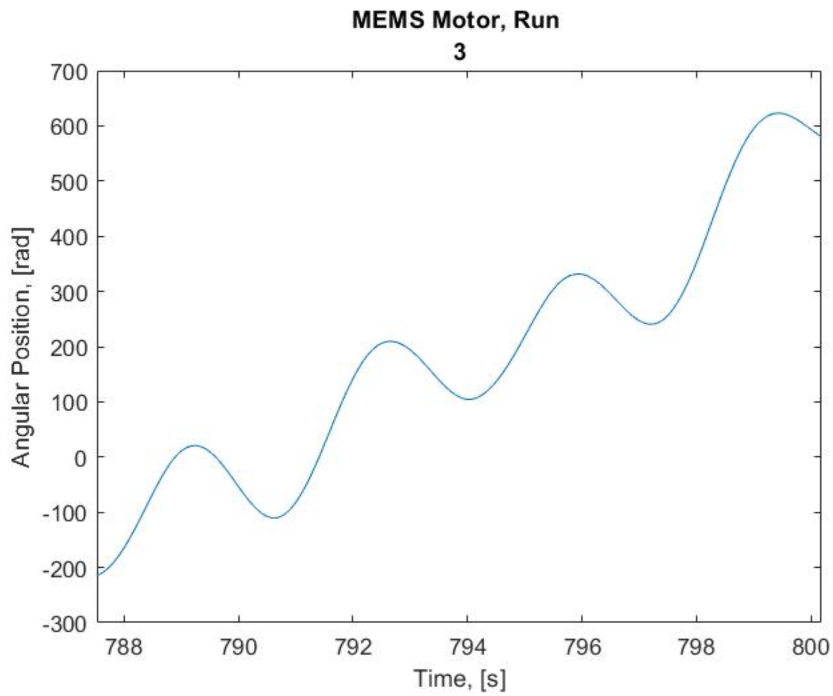


Fig. 11 Angular Position vs Time, Run 3

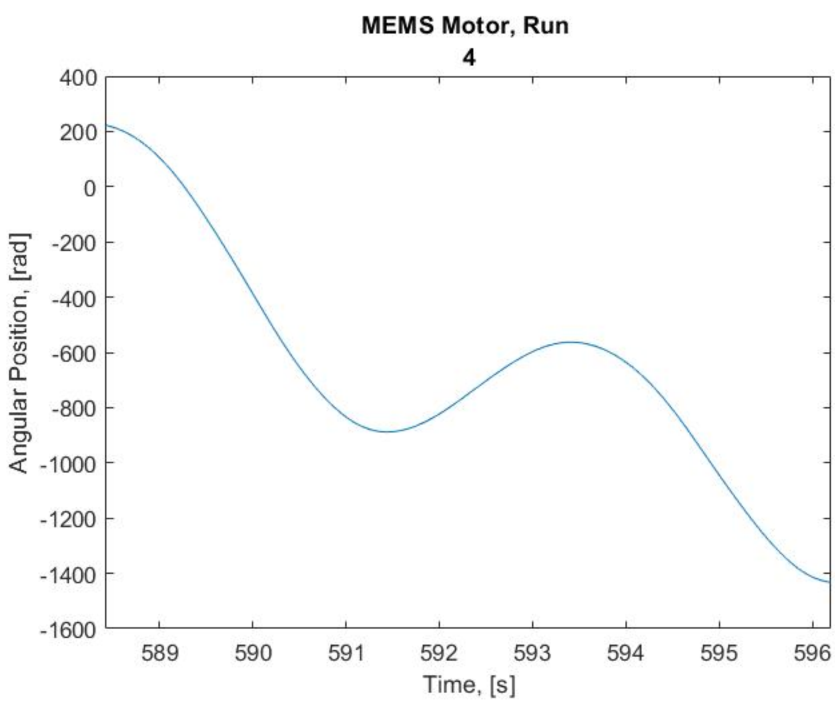


Fig. 12 Angular Position vs Time, Run 4

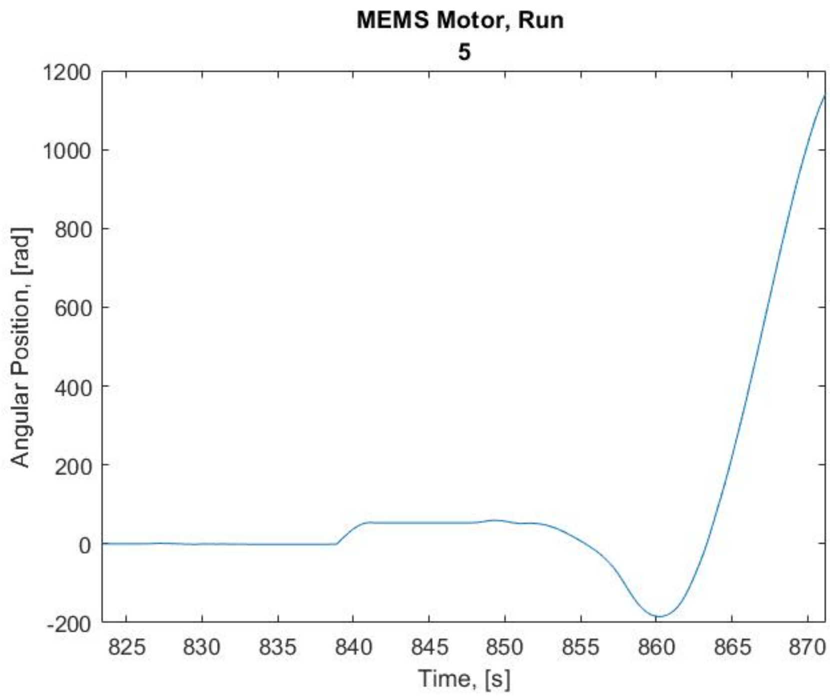


Fig. 13 Angular Position vs Time, Run 5

B. Angular Position and Velocity vs Time

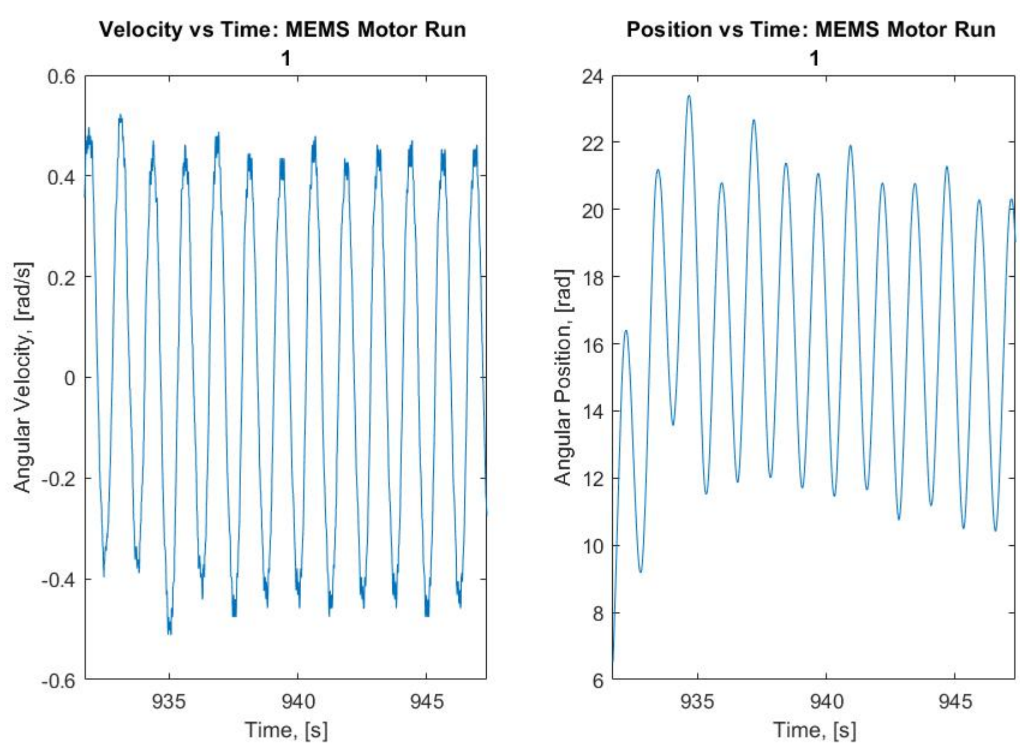


Fig. 14 Angular Position vs Time, Run 1

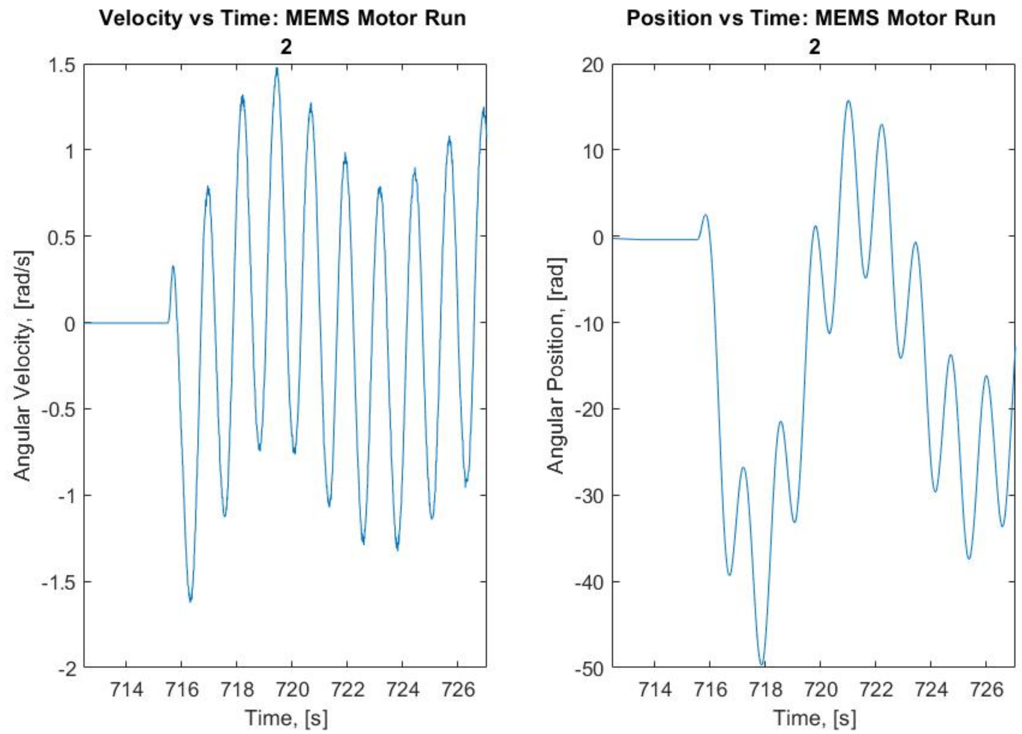


Fig. 15 Angular Velocity and Position vs Time, Run 2

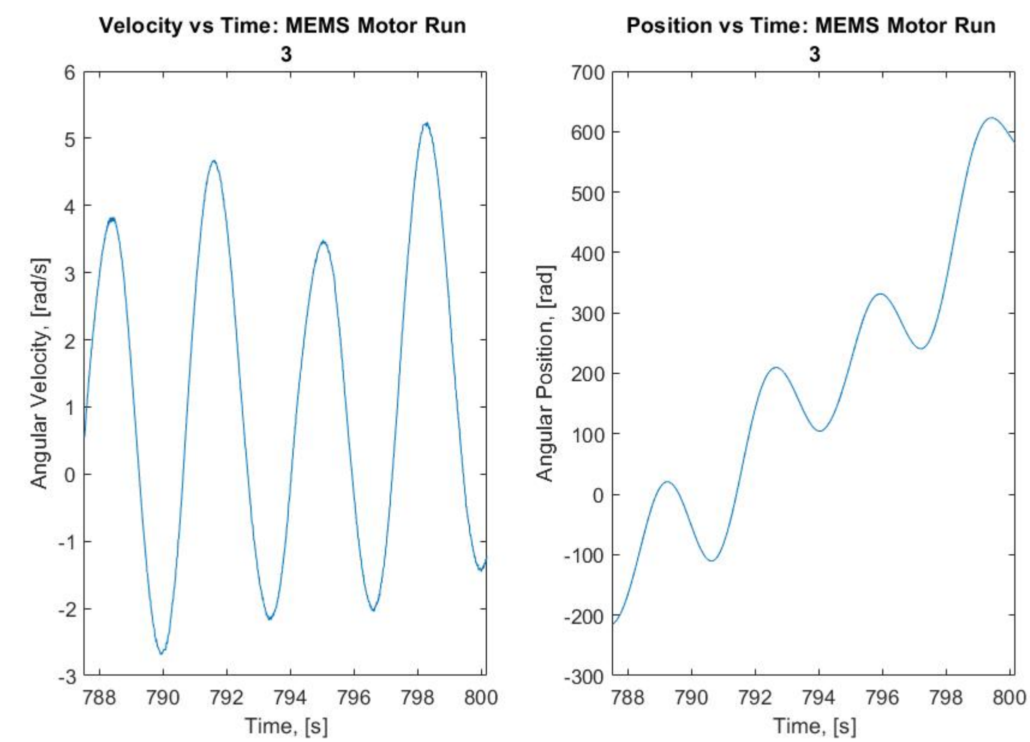


Fig. 16 Angular Velocity and Position vs Time, Run 3

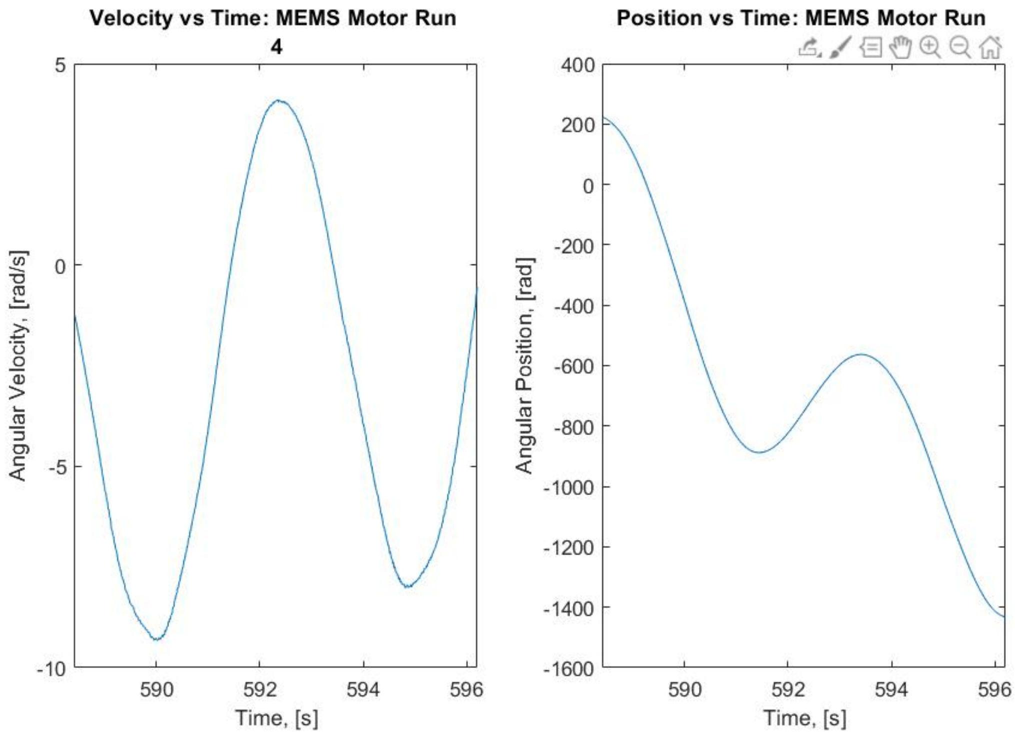


Fig. 17 Angular Velocity and Position vs Time, Run 4

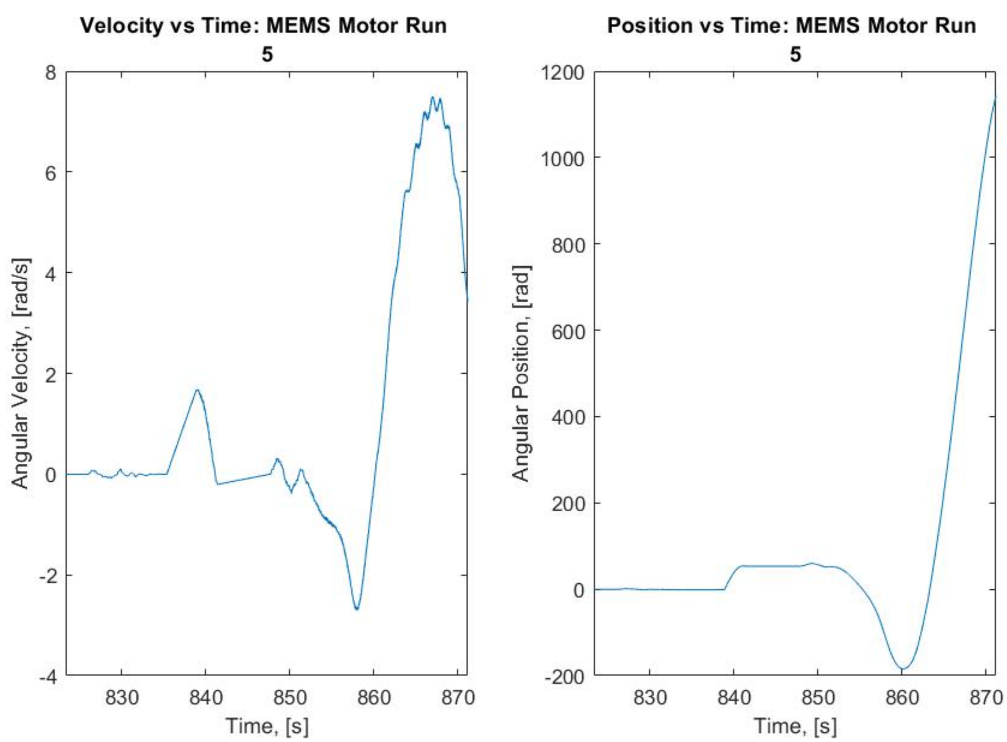


Fig. 18 Angular Velocity and Position vs Time, Run 5

C. Angular Velocity vs Time

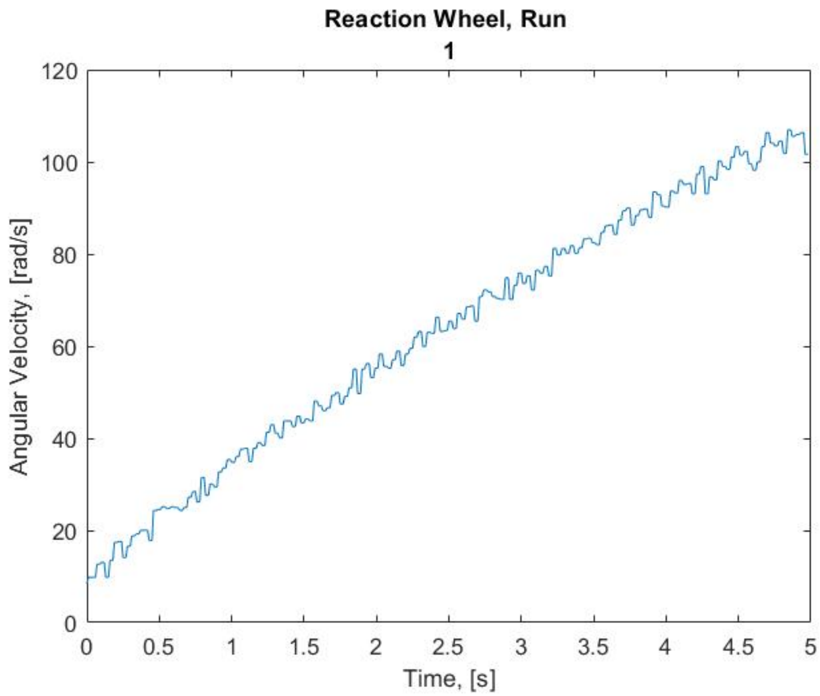


Fig. 19 Angular Velocity vs Time, Run 1

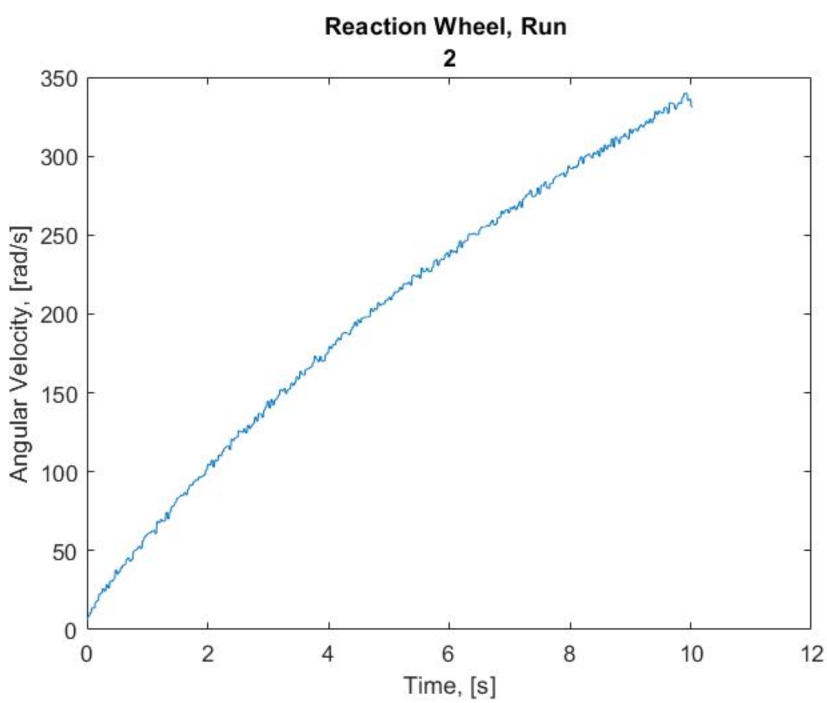


Fig. 20 Angular Velocity vs Time, Run 2

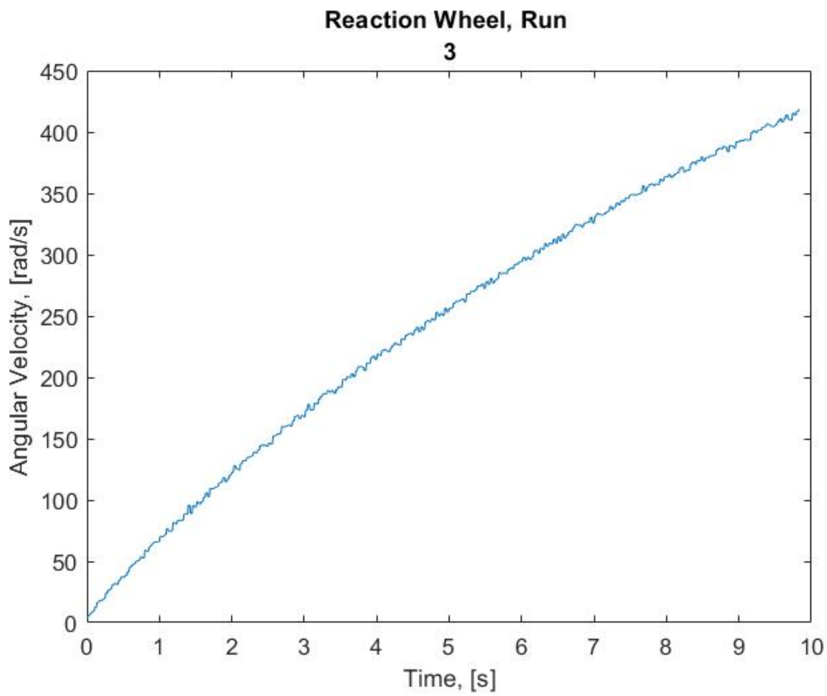


Fig. 21 Angular Velocity vs Time, Run 3

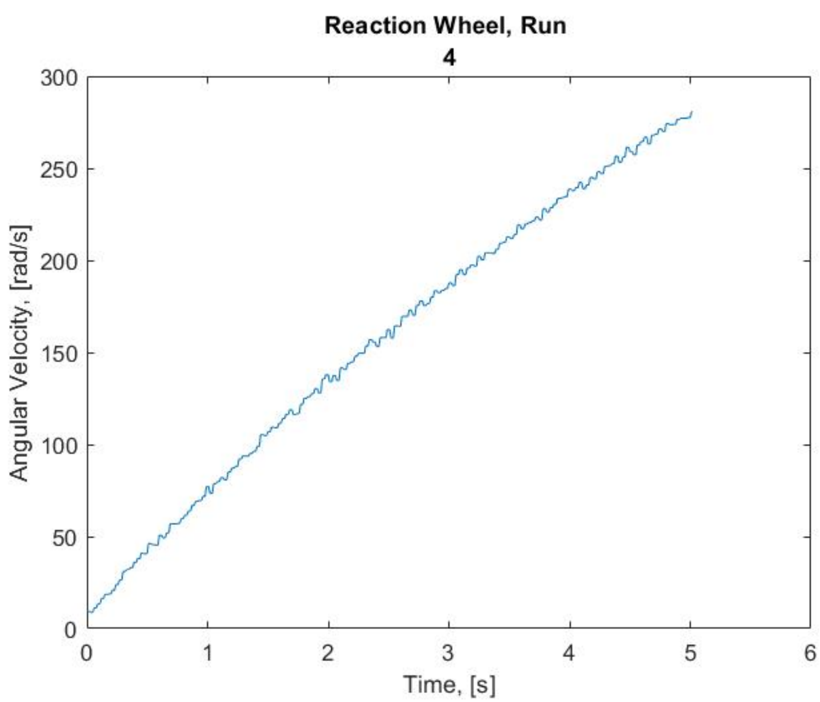


Fig. 22 Angular Velocity vs Time, Run 4

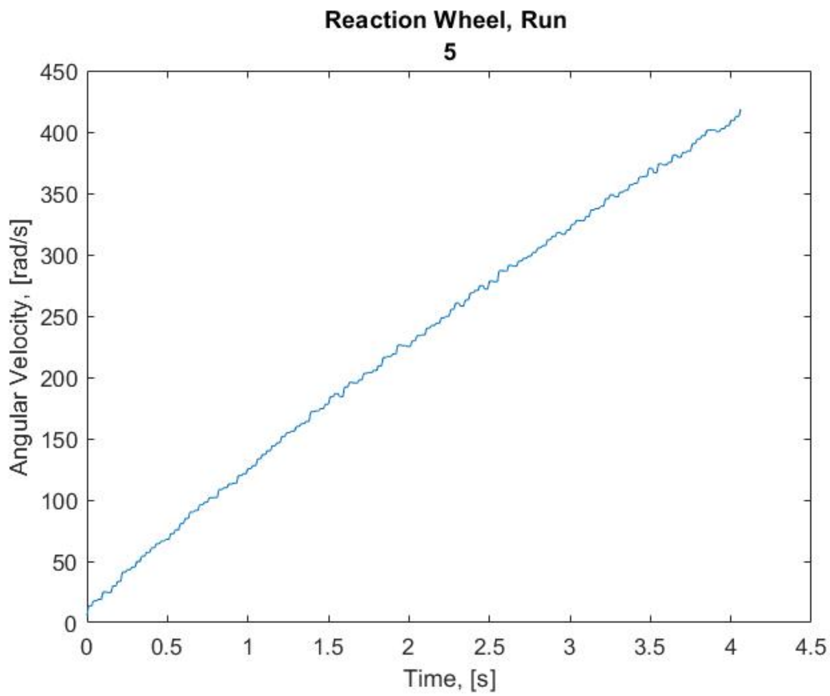


Fig. 23 Angular Velocity vs Time, Run 5

Appendix C: MatLab Code

D. Lab 1 Main Script

```
%% LAB 1, ASEN 3200
%{
This script reads in the data gathered by Group 8 for ASEN 3200 Lab 1, then
analyzes that data and generates the plots requested in the lab document

Author: W. Hunter Daboll
Co-Authors: Connor O'Reilly, Tomaz Remec
Last Edited: 9/8/2020
%}
%% Housekeeping
clear all; clc; close all;

%% Part 1 - C
disp("Part 1 - C ------------------------------------------------------------------------------------------");
plotRawData = 'False'; % Boolean variable which toggles a figure for each data set,
% plotting the raw encoder velocity against the velocity measured by the
% MEMS gyro, and displays the line fitted to this data
showAngVel = "True"; % Boolean variable which toggles a subplot on the graphs of angular position,
% optionally displaying angular velocity side by side for comparison (when value == "True")
numFiles = 5;
%baseName = 'Data/Unit7_Gyro_Auto_';
baseName = 'Data/G8_UnitUnknown_MEMSMOTOR_RUN0';
[bias, bias_std, scaleFactor, sf_std] =memsMotor(numFiles, baseName, plotRawData, showAngVel);

%% Part 2 - C
disp("Part 2 - C ------------------------------------------------------------------------------------------");
numFiles = 5;
baseName = 'Data/G8_UnitUnknown_RWHEEL_RUN0';
[MOI, MOI_std] = rWheel(numFiles,baseName);
```

```
fprintf("Moment of Inertia of the system is %f with a standard deviation of %f\n",MOI,MOI_std);
```

E. MEMS Motor Analysis

```
function [bias, bias_std, SF, sf_std] = memsMotor(numFiles,baseName, PRD, showAngVel)
%Reads in data from MEMSMOTOR data set, produces bias and scale factor,
%then produces graphs of angular rate and position as a function of time
%based on adjusted gyro measurements
bias = zeros(1,numFiles); % Preallocate bias array
SF = zeros(1,numFiles); % Preallocate Scale Factor array
for i = 1:numFiles

    filename = strcat(baseName,int2str(i));
    currentData = readtable(filename);
    encoderVel = table2array(currentData(:,2));
    gyroOutput = table2array(currentData(:,3))* 0.10472 ; % Convert RPM to rad/s
    if (PRD == "True") % PRD stands for Print Raw Data
        figure();
        plot(encoderVel,gyroOutput);
        xlabel("Encoder Velocity, [rad/s]")
        ylabel("Gyro Velocity, [rad/s]")
    end
    P = polyfit(encoderVel, gyroOutput, 1);
    gyroEstimate = polyval(P,encoderVel);
    if PRD == "True"
        hold on;
        plot(encoderVel,gyroEstimate,'LineWidth',1.5)
        hold off
    end
    % Bias - y intercept
    bias(i) = P(2);

    SF(i) = P(1);


end


mean_bias = mean(bias);
bias_std = std(bias);
fprintf("The mean bias is: %.4f with a standard deviation of %.4f \n",mean_bias, bias_std);

mean_SF = mean(SF);
std_SF = std(SF);
fprintf("The mean scale factor is: %.4f with a standard deviation of %.4f\n\n",mean_SF, std_SF);


for i = 1:numFiles
    filename = strcat(baseName,int2str(i));
    currentData = readtable(filename);
    encoderVel = table2array(currentData(:,2));
    gyroOutput = table2array(currentData(:,3))* 0.10472 ; % Convert RPM to rad/s
    time = table2array(currentData(:,1));
    delta_t = mean(diff(time)); % Find the average change in time between indecies

    gyro_Position = zeros(1,length(gyroOutput));
    calculated_Gyro_Rate = gyroOutput.*mean_SF - mean_bias;
    for j = 1:length(calculated_Gyro_Rate)
        if j>1
            gyro_Position(j) = gyro_Position(j-1) + calculated_Gyro_Rate(j)*delta_t;
        end
    end
    figure();

    if showAngVel == "True"
```

```
        set(gcf,'Position',[100 200 800 500])
        subplot(1,2,1);
        plot(time,calculated_Gyro_Rate);
        xlim([time(200),time(end)]);
        xlabel("Time, [s]");
        ylabel("Angular Velocity, [rad/s]");
        title(["Velocity vs Time: MEMS Motor Run " num2str(i)]);
        subplot(1,2,2);
        plot(time,gyro_Position);
        xlim([time(200),time(end)])
        xlabel("Time, [s]");
        ylabel("Angular Position, [rad]");
        title(["Position vs Time: MEMS Motor Run " num2str(i)]);
    else
        plot(time,gyro_Position);
        xlim([time(200),time(end)])
        xlabel("Time, [s]");
        ylabel("Angular Position, [rad]");
        title(["MEMS Motor, Run " num2str(i)]);
    end
end
bias = mean_bias;
bias_std = bias_std;
SF = mean_SF;
sf_std = std_SF;
end
```

F. Reaction Wheel Analysis

```
function [mean_MOI, MOI_std] = rWheel(numFiles,baseName)
%Plots the angular velocity of each data file vs time, finds average
%angular acceleration and estimates moment of inertia: averages and find
%the standard deviation of the moment of inertia calculations
MOI = zeros(1,numFiles);
for i = 1:numFiles

    filename = strcat(baseName,int2str(i));
    currentData = readtable(filename);
    time = table2array(currentData(:,1))./1000;
    time = time - time(1);
    torque = table2array(currentData(:,2));
    angVel = table2array(currentData(:,3))* 0.10472; % Convert RPM to rad/s

    P = polyfit(time, angVel, 1);
    angAccel = P(1);

    MOI(i) = mean(nonzeros(torque))/angAccel;

    figure();
    plot(time,angVel);
    xlabel("Time, [s]");
    ylabel("Angular Velocity, [rad/s]");
    title(["Reaction Wheel, Run " num2str(i)]);
end
mean_MOI = mean(MOI);
MOI_std = std(MOI);
end
```

G. Reaction Wheel Analysis