Lab #5

Inertial Sensors

MECHENG 552

Team 4B

Saptadeep Debnath
Manavendra Desai
Ellen Kim
David Radtke

Professor Shorya Awtar
22 November 2019

Department of Mechanical Engineering
University of Michigan, Ann Arbor

Contents

List of Figures

List of Tables

1 Electrical Wiring (10pts)

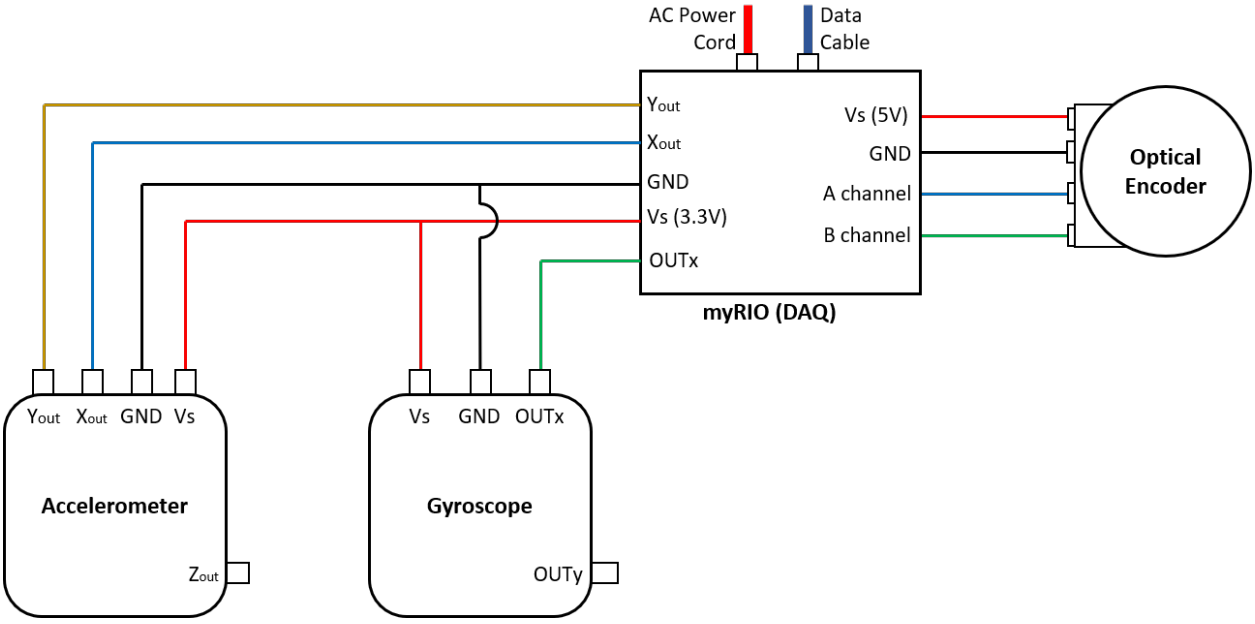

Figure 1: Electrical Wiring Diagram

Figure 1 represents the complete electrical wiring schematic which consists of the four major components - 3-axis accelerometer, dual-axis gyroscope, an optical encoder and the data acquisition unit (myRIO), which are used to estimate the angular position of the pendulum. Three axes measurements are used from this IMU - linear accelerations in x and y from the accelerometer and angular rate in x from the gyroscope. Linear acceleration in z and angular rate in y are not used in this experimental setup. These measurements are fed through to LabVIEW via myRIO which is used as the data acquisition unit. Accelerometer and Gyroscope are powered through the 3.3V power output of myRIO. An optical encoder attached at the pendulum pivot measures *reference* angles, against which the position estimates from the IMU system are compared. The encoder is powered through the myRIO's 5V power output.

2 Low Frequency Characterization (30pts)

(a) Accelerometer in Horizontal Direction

Physical set-up of accelerometer in horizontal direction: A 3-axis accelerometer, the ADXL335, is mounted normal to the axis of rotation via a 3D-printed bracket. The accelerometer individually outputs the readings for each of its axes and the readings in all cases are proportional to the acceleration that axis experiences. One of these axes are sensitive in the direction that is perpendicular to the axis of rotation and parallel to the floor, which we will say is the horizontal direction.

Via kinematic equations, an assumption of low frequency movement, and a sensor placement close to the axis of rotation, we are able to formulate the acceleration the sensor perceives as $gsin(\theta)$. Consequently, the voltage output from the sensor is:

$$V_{out,h} = V_{b,h} + S \cdot gsin(\theta)$$

where $V_{out,h}$ is the voltage output from the horizontal axis, $V_{b,h}$ is the bias voltage of the accelerometer(half of the supply voltage), and S is the sensitivity of the accelerometer. Within LabVIEW, we were able to find the inverse of this function with respect to θ to find our angular position at any given moment. This was formulated as:

$$\theta = sin^{-1}\left(\frac{V_{out,h} - V_{b,h}}{gS}\right)$$

But since the sensitivity is defined in units of $\frac{mv}{g}$, the denominator simplifies to the numeric value of S. // **Maximum range of α motion of accelerometer in horizontal direction**:To measure the maximum useful range of the horizontal axis, the pendulum was rotated from $\pm\frac{\pi}{2}$.

Time domain plot of accelerometer in horizontal direction: When calculating the phase angle and magnitude error of the time domain response, there arises an issue in that it is difficult to pair a point in one wave to its counterpart in the other in order to calculate its respective phase and magnitude error. Typically, one could look at the peaks of the curve and calculate the phase there and assume it is constant over time, since the response time of the sensor should not be changing. In this case, however, the sensitivity of the sensor changes with respect to its position and is at its worse near $\pm\frac{\pi}{2}$. Understanding this, the phase was instead calculated near $\alpha = 0$ and was found to be 0. This makes sense because the accelerometer has a bandwidth of 1600Hz which, if the accelerometer is modeled as a first-order system, would mean the phase lag for a 2Hz signal would be about 0.07 degrees and consequently the time lag for a 2Hz signal would be about 10^{-4} seconds which is much less than our sampling time of 5ms. Therefore our data should not be able to perceive the phase difference between the accelerometer and encoder. As the position nears the bounds of our measurement, it may appear that the phase difference increases, but that is only due to the sensitivity of this measurement scheme changing(note that this is different than the sensitivity of the sensor itself), as discussed in the next section. Accepting all of this, we can calculate the magnitude error of the accelerometer by comparing any point of the accelerometer measurement to the encoder measurement taken at the same time.

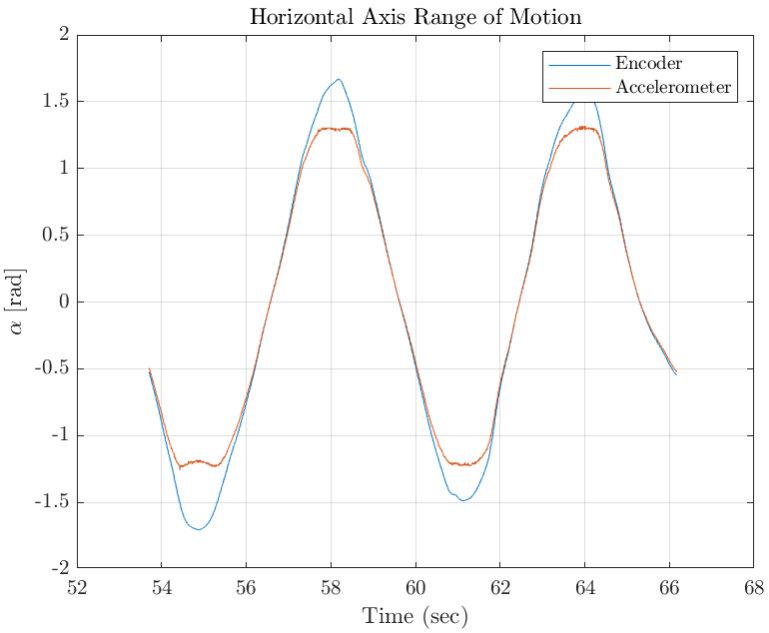

Figure 2: Time domain comparison of angular position derived from horizontal accelerometer compared to angular position measured by encoder.

The magnitude error is calculated as shown in Eq. 1 and then plotted against the angle measured by the encoder.

$$100 * \frac{amp_{acc} - amp_{enc}}{amp_{enc}}\% \tag{1}$$

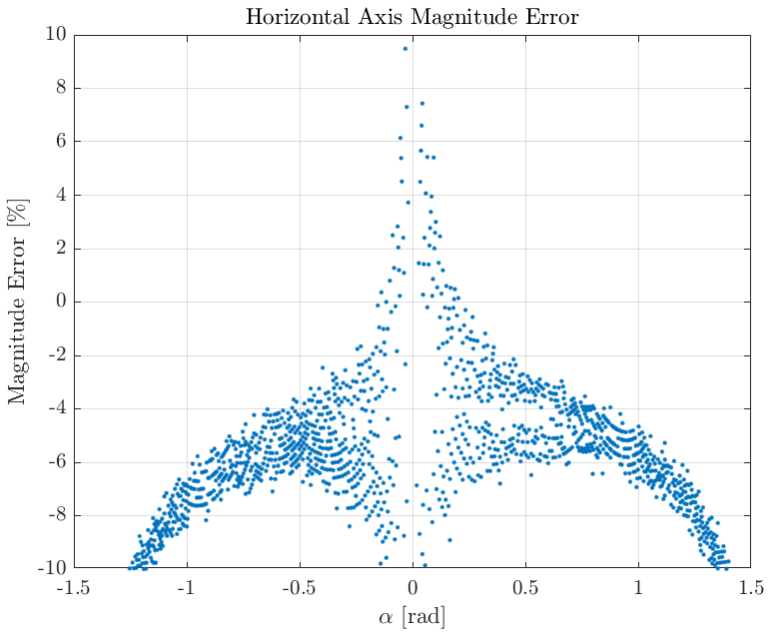

Figure 3: Magnitude Error of Horizontal Axis with Respect to Angle. Note that the error approaches inf. as α approaches 0 for mathematical reasons and not physical reasons.

From fig. 3 we can see that the region of the horizontal sensor that meets the 10% error criteria is approximately ± 1.2 rad .

Sensitivity of accelerometer in horizontal direction: The sensitivity of this measurement scheme(as opposed to the sensitivity of the sensor itself) in the horizontal direction is defined as the change of voltage output by the horizontal sensor compared to the change of position. It can be mathematically formulated as $\frac{\delta V_{out,h}}{\delta \alpha}$. To understand this better, we plotted the sensitivity of the scheme versus the angle measured by the encoder for the same time domain measurements referred to in the last section, as shown in fig. 4.

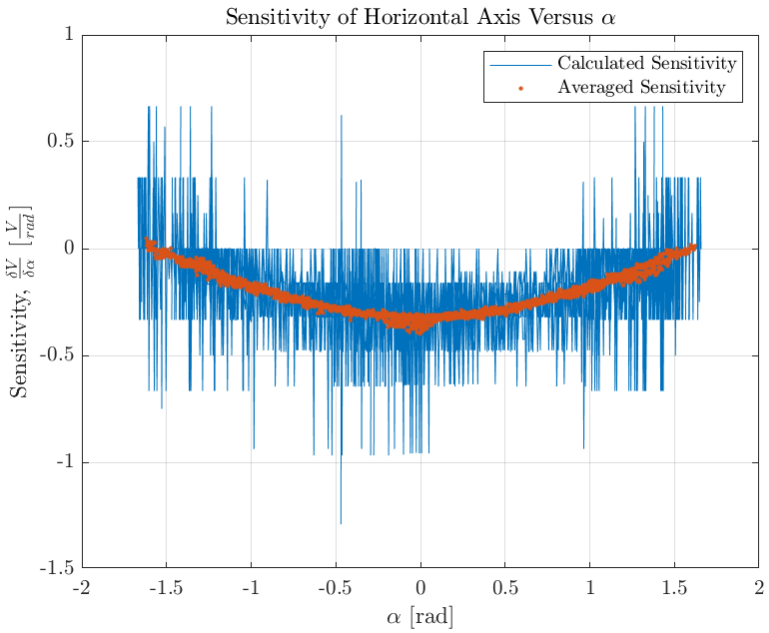

Figure 4: Sensitivity of the Accelerometer Mounted in the Horizontal Direction. Note that the sensitivity of the scheme appears to be erratic because taking the derivative of the noise from the accelerometer and the quantitization from the encoder both lead to erratic results. In order to get a better picture of the sensitivity, a moving mean was taken.

From fig. 4 the sensitivity of the scheme appears to range from completely insensitive(0) near the bounds of the measurement to -0.3 at $\alpha = 0$. This in agreement with our expectations as the sensor advertises its sensitivity as $300\frac{mv}{g}$.

Peak-to-peak noise level of accelerometer in horizontal direction: The voltage noise and angular position noise were both measured while $\alpha = 0$. As shown in fig. 5, the peak-to-peak noise amplitude for the voltage output of the sensor is about 0.003 volts, which creates a peak-to-peak position noise of about 0.9 radians.

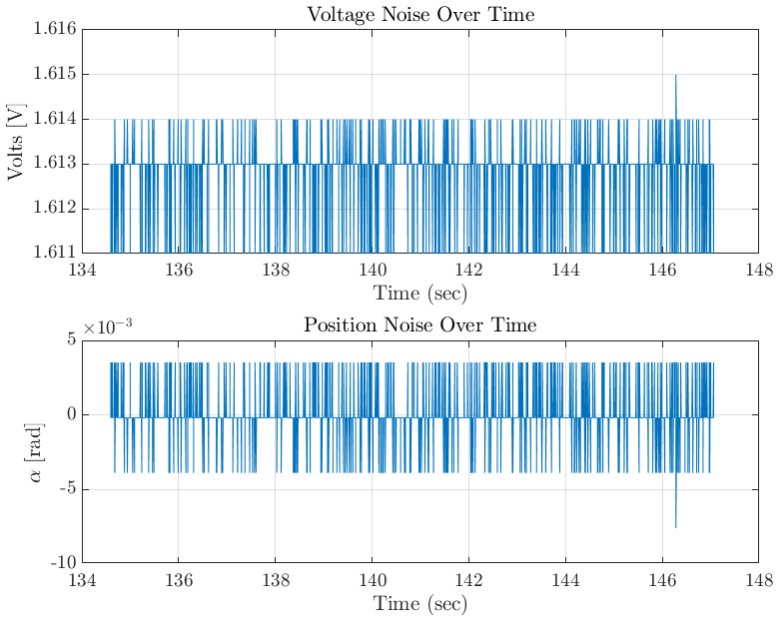

Figure 5: Voltage and angular position noise for horizontally mounted accelerometer.

Filter of accelerometer in horizontal direction: In order to smooth out the amount of noise in the position measurements, we decided to add a first-order filter. From an analytical standpoint, a filter of the form $\frac{1}{\tau s+1}$ will always increase phase lag and increase magnitude error as the signal approaches the cut-off frequency $\frac{1}{\tau}$. For our sensing scheme, this proved to be true as shown in fig. 6. Furthermore, by zooming in on the data we could also see that the amount of noise also reduced as the filter strength increases, as we expected.

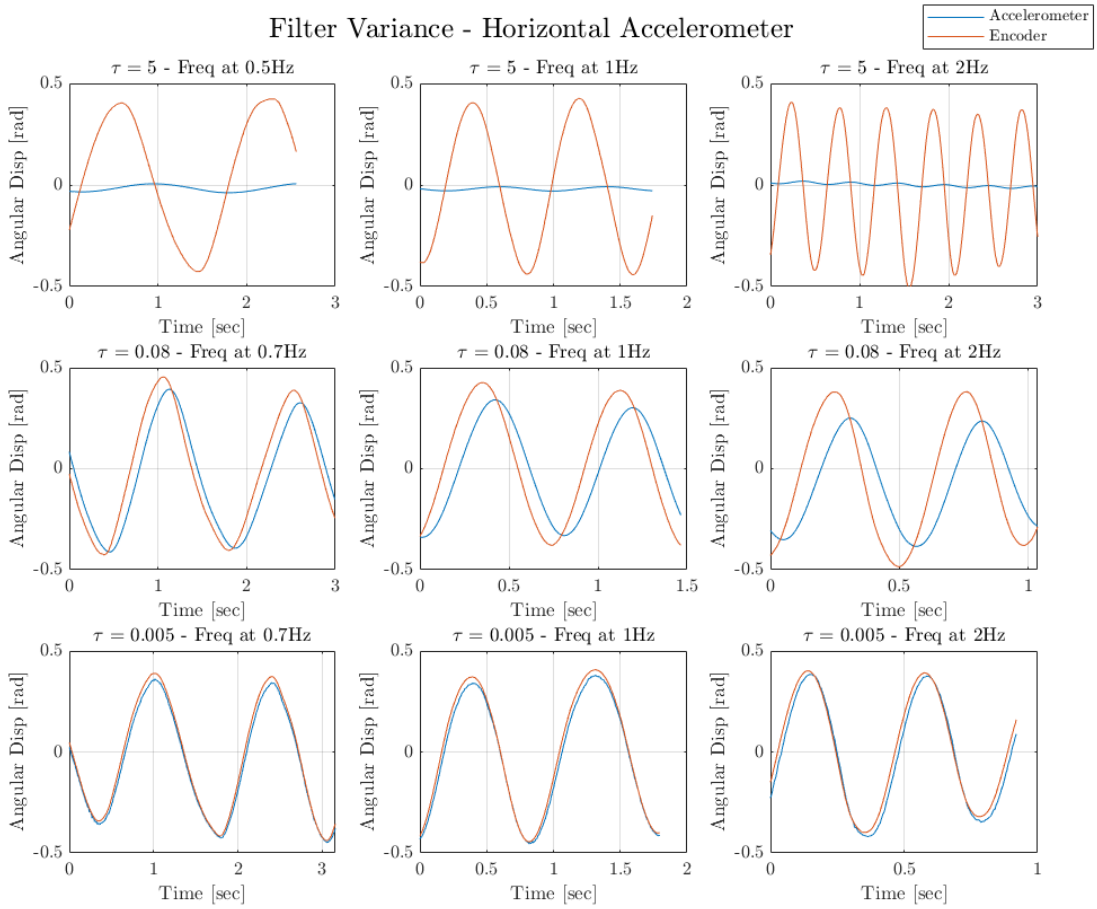

Figure 6: Horizontal accelerometer filters evaluated for various low frequencies. The filters are compared for the frequencies of approximately 0.5 Hz, 1Hz, and 2Hz.

Table 1: Magnitude Error and Phase Lag associated with Filtering of Horizontal Accelerometer

Filter τ [Sec]	Pendulum Swing [Hz]	Magnitude Error [%]	Phase Lag [deg]
5	0.5	95.7	68
5	1	96.5	73.8
5	2	98.5	97.2
0.08	0.7	13.1	21.4
0.08	1	19.1	28.8
0.08	2	38.0	46.9
0.005	0.7	9.0	3.8
0.005	1	8.8	3.6
0.005	2	4.3	7.2

Overall the middle filter, where $\tau = 0.08$, seems to give the best compromise of noise reduc-

tion and error of the designs tested. The filter where $\tau = 0.005$ clearly gives quicker and more accurate results, but there is still noticeable noise.

(b) Accelerometer in Vertical Direction

Physical set-up of accelerometer in vertical direction: Like the accelerometer mounted in the horizontal direction, the accelerometer in the vertical direction is also a part of the ADXL335 that is mounted normal to the axis of rotation. In this context the vertical direction is the direction that is perpendicular to both the axis of rotation and the ground.
In a similar fashion to the horizontal accelerometer, we can relate the acceleration perceived by the vertical accelerometer to the angular position, α, via kinematic relationships and then some simplifying assumptions. In this case however the acceleration the accelerometer perceives and consequently the voltage it outputs is dependent on $cos(\theta)$. We find:

$$V_{out,v} = V_{b,v} + S \cdot gcos(\alpha)$$

where $V_{out,v}$ is the voltage output from the vertical axis, $V_{b,v}$ is the bias voltage of the accelerometer(half of the supply voltage), and S is the sensitivity of the accelerometer. And further we can find α as a function of $V_{out,v}$ via:

$$\alpha = cos^{-1}\left(\frac{V_{out,v} - V_{b,v}}{gS}\right)$$

Maximum range of α motion of accelerometer in vertical direction: To measure the maximum useful range of the vertical axis, the pendulum was rotated in the ranges of $[0, \pi]$ and $[-\pi, 0]$.

Time domain plot of accelerometer in vertical direction: Using the same arguments as for the horizontal direction, we can assume that the phase error is 0 for the low frequency measurements taken for the vertical direction and that the magnitude error can be calculated by comparing measurements taken at the same time. The time domain data taken for the vertical direction is shown in fig. 7

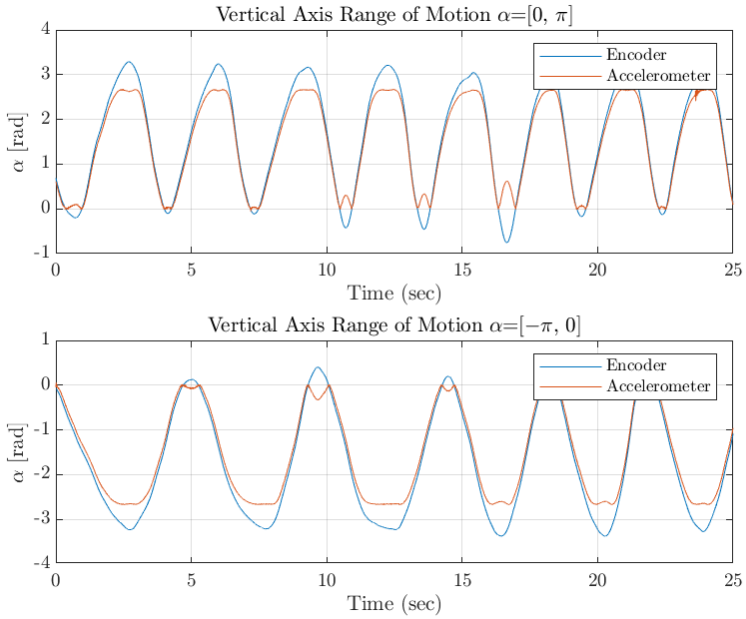

Figure 7: Time domain comparison of angular position derived from vertical accelerometer compared to angular position measured by encoder.

The magnitude error is calculated as shown in Eq. 1 and then plotted against the angle measured by the encoder.

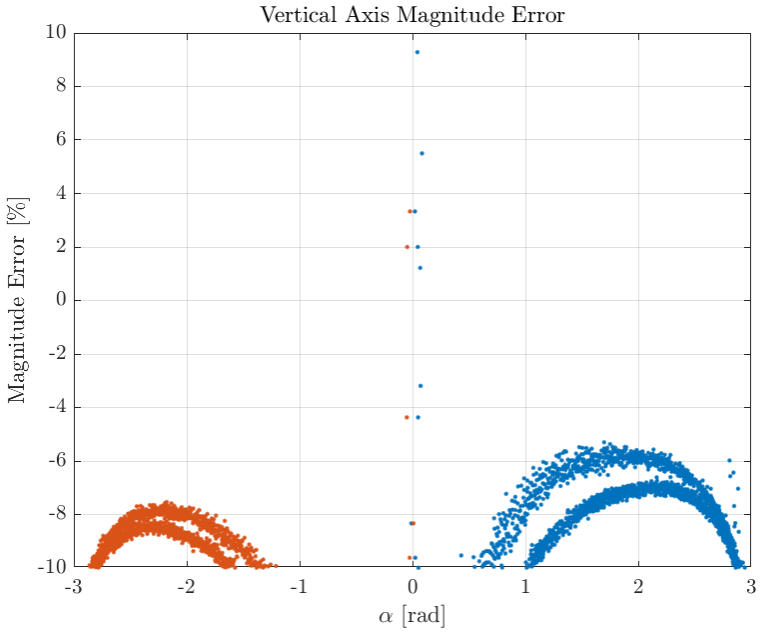

Figure 8: Magnitude Error of Vertical Axis with Respect to Angle. Note that the error approaches inf. as α approaches 0 for mathematical reasons and not physical reasons.

From 8 we can see that the applicable region of the vertical sensor is approximately in the

ranges of [-2.7,-1.2] and [1, 3] radians.

Sensitivity of accelerometer in vertical direction: Again, we calculate sensitivity of this scheme as $\frac{\delta V}{\delta \alpha}$ as is shown in fig. 9. Like before, we took a moving mean to understand the sensitivity separate from noise.

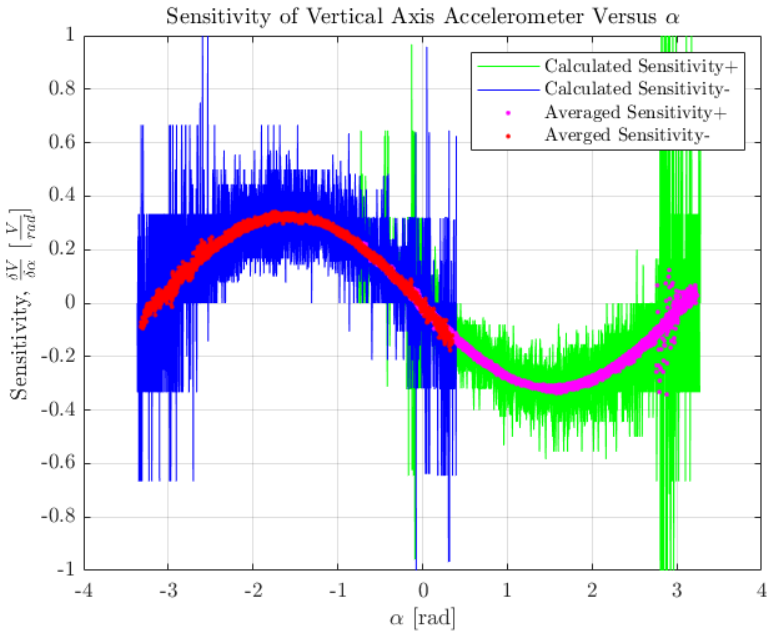

Figure 9: Sensitivity of the Accelerometer Mounted in the Vertical Direction. Note that the sensitivity of the scheme appears to be erratic because taking the derivative of the noise from the accelerometer and the quantitization from the encoder both lead to erratic results. In order to get a better picture of the sensitivity, a moving mean was taken.

From this we can see that the sensitivity ranges from ± 0.3, which is the max offered by the sensor, to 0.

Peak-to-peak noise level of accelerometer in vertical direction: The voltage noise and angular position noise where both measured while $\alpha = 0$. As shown in 10, the peak-to-peak noise amplitude for the voltage output of the sensor is about 0.004 volts, which creates a peak-to-peak position noise of about 1 radians.

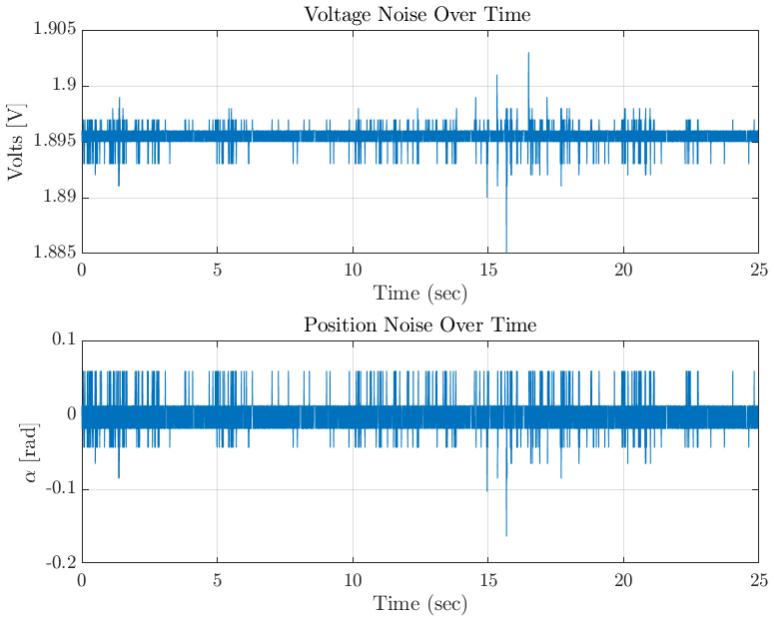

Figure 10: Voltage and angular position noise for vertically mounted accelerometer.

(c) Combination of Accelerometer in Horizontal and Vertical Directions

Physical set-up of accelerometer in horizontal and vertical direction: Accelerometers sensitive in the X (Accelerometer X) and Y (Accelerometer Y) directions will be used together to provide a larger range of angle measurement for pendulum link angle α. Their mounting and physical set-up is the same as that in Q2(a) and Q2(b).

Maximum range of α motion of accelerometer in horizontal and vertical direction: Once the raw voltages from Accelerometers X and Y have been converted to the respective angles (θ_X, θ_Y), and the angles have been corrected for sign and phase (θ_X^{cor}, θ_Y^{cor}), the next step is to develop a logic that helps switch the sensor being used for pendulum link angle estimation. Based on the error graphs shown in Q2(a) and Q2(b) and physical trials on the setup, it was observed that Accelerometer X tracks α (measured by the encoder) well for $\alpha \epsilon$ [-0.75,1.2], [1.94,3.14] and [-2.5,-3.14] radians. Outside these ranges, Accelerometer Y tracks α well. Therefore, the following logic was used. As long as θ_X^{cor} lies within the above ranges, it will be passed as α. If θ_X^{cor} lies outside the above ranges, then θ_Y^{cor} will be passed as α.

Time domain plot of accelerometer in horizontal and vertical direction: From Figure 11 we see that for the Accelerometer X+Y sensor, the range of pendulum link angles that provides accuracy of magnitude error $< 10\%$ and phase lag $< 10\%$ is $[-3, 3]$ radians.

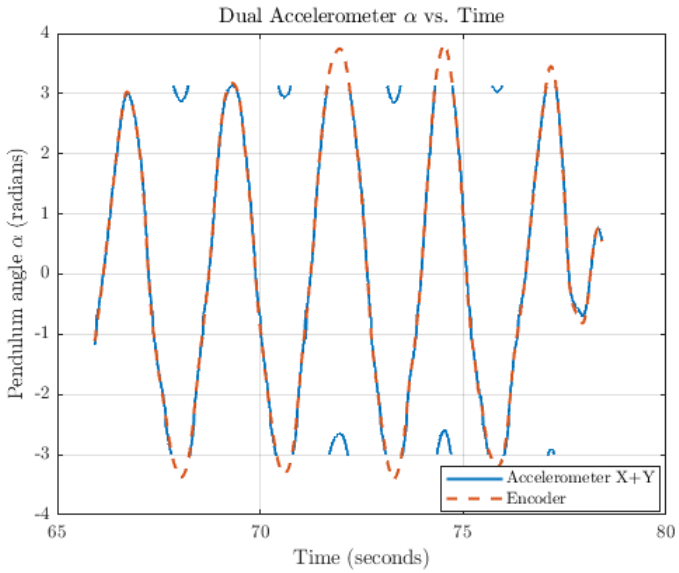

Figure 11: Time domain plot of the Accelerometer X and Y combination sensor compared with angle measurements from an incremental encoder.

Sensitivity of accelerometer in horizontal and vertical direction: Sensitivity of the Accelerometer X and Y combination will be the same as that of Accelerometers X and Y, when operating in regions sensitive to the respective accelerometer.

Peak-to-peak noise level of accelerometer in horizontal and vertical direction: Since $\alpha = 0$ degrees corresponds to the sensitive region of Accelerometer X, peak to peak noise for the Accelerometer X and Y combination will be the same as that of Accelerometer X, i.e, *enter here.*

(d) Single Rate-Gyro

Physical set-up of rate-gyro: A single rate-gyro, LPR410AL, is a dual-axis micromachined gyroscope that measures the the angular rate with respect to the pitch and roll axis. To measure the pendulum angle using a single rate-gyro, the sensor was mounted parallel to the plane of pendulum's rotation at the pendulum's center of rotation. The rate-gyro produces voltage and the pendulum's angular velocity $\dot{\alpha}_{e/r}$ is calculated by

$$\dot{\alpha}_{e/r} = \frac{V_{out/r} - V_{b/r}}{S_r} + C_1$$

where $V_{out/r}$ is the electrical signal from the gyro, $V_{b/r}$ is the bias voltage of the rate-gyro (half of the supply voltage), S_r is the sensitivity of the rate-gyro, and C_1 is a constant error of the rate-gyro. To determine the pendulum's angular position using the rate-gyro, the $\dot{\alpha}_{e/r}$ is integrated producing

$$\alpha_{e/r} = \frac{1}{s}\dot{\alpha}_{e/r} + C_1 t + C_2$$

where C_2 is a product from the integration. Since the rate-gyro was manufactured for measuring angular velocity, when we use it for measuring angular displacement, there is a drift

caused from the $C_1 t$ term. Thus, in the LabVIEW, after transforming the electrical signal to the pendulum's angular velocity, the C_1 term is removed by measuring the error experimentally and the term is integrated to find the pendulum's angular displacement.

Maximum range of α motion of rate-gyro: To measure the maximum α range, the pendulum was rotated from $\pm\pi$.

Time domain plot of rate-gyro: The time domain plots of the rate-gyro with the encoder is shown in figure 12 with two experiments showing the pendulum rotating separately toward π and $-\pi$. The plot shows that the encoder and the rate-gyro have a relatively close match during its first cycle; however, as time increases, the rate-gyro results drift away. This is due to the C_1 error not being perfectly removed and causing a slow change in angular displacement with respect to time. The maximum range of α with less than 10% error in magnitude and 2 deg lag in phase, is the full range from $\pm\pi$ for the initial cycle. The magnitude error seen at the maximum pendulum angle of $-\pi$ is 7% and π is 0%. The phase lag seen at $-\pi$ is 1.02 deg and at π is 1.42 deg. However, due to the error constant explained earlier, the rate-gyro results do begin to drift causing an increase in magnitude error after the first cycle of greater than 10%.

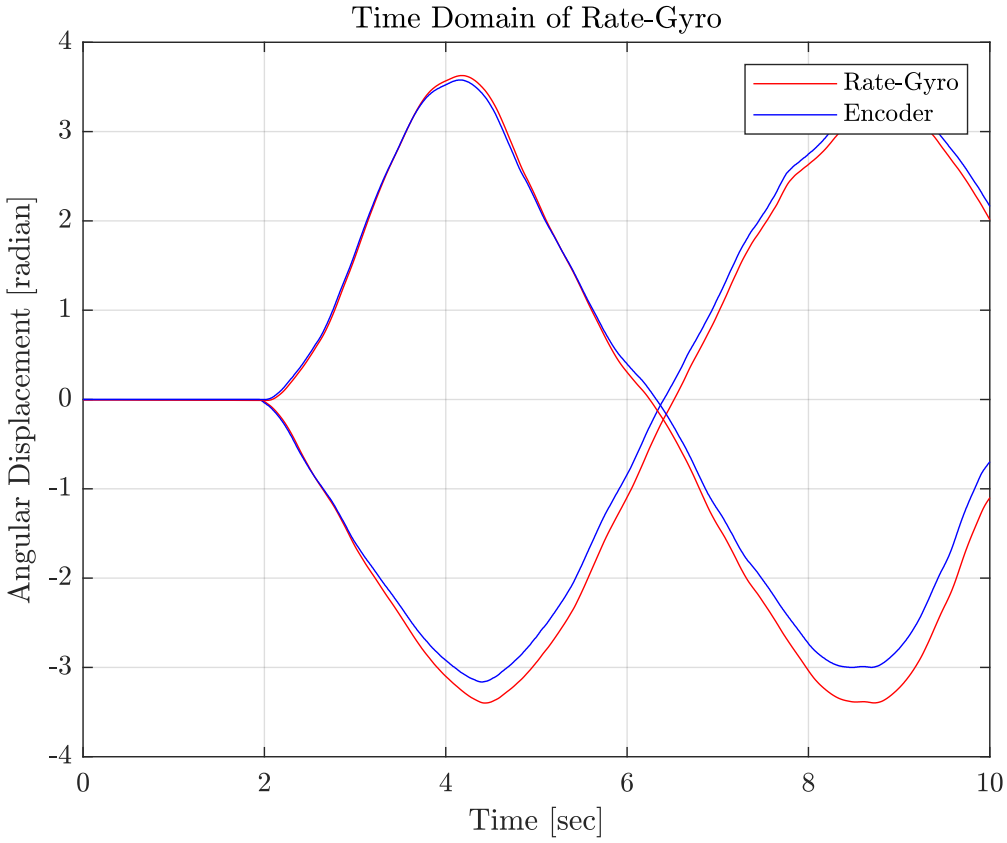

Figure 12: The time domain plot for the rate-gyro. As the pendulum rotates to $\pm\pi$ approximately, the rate-gyro and the encoder show a relatively close match. However, as time increases, the rate-gyro drifts away from the encoder result.

Sensitivity of rate-gyro: To calculate the sensitivity of the rate-gyro between the measured angle and output voltage, the slope of the voltage to angle was taken from the linear portion of one of the experiments, the pendulum approaching π was taken to be 0.001V/degree.

Peak-to-peak noise level of rate-gyro: When the pendulum angle was set to 0, the rate-gyro showed noise in its voltage measurements and fluctuation in the angular measurement is as shown in figure 13. The peak to peak noise level of the voltage measurement is 0.003V and the corresponding angular displacement fluctuation is 1.43 degree.

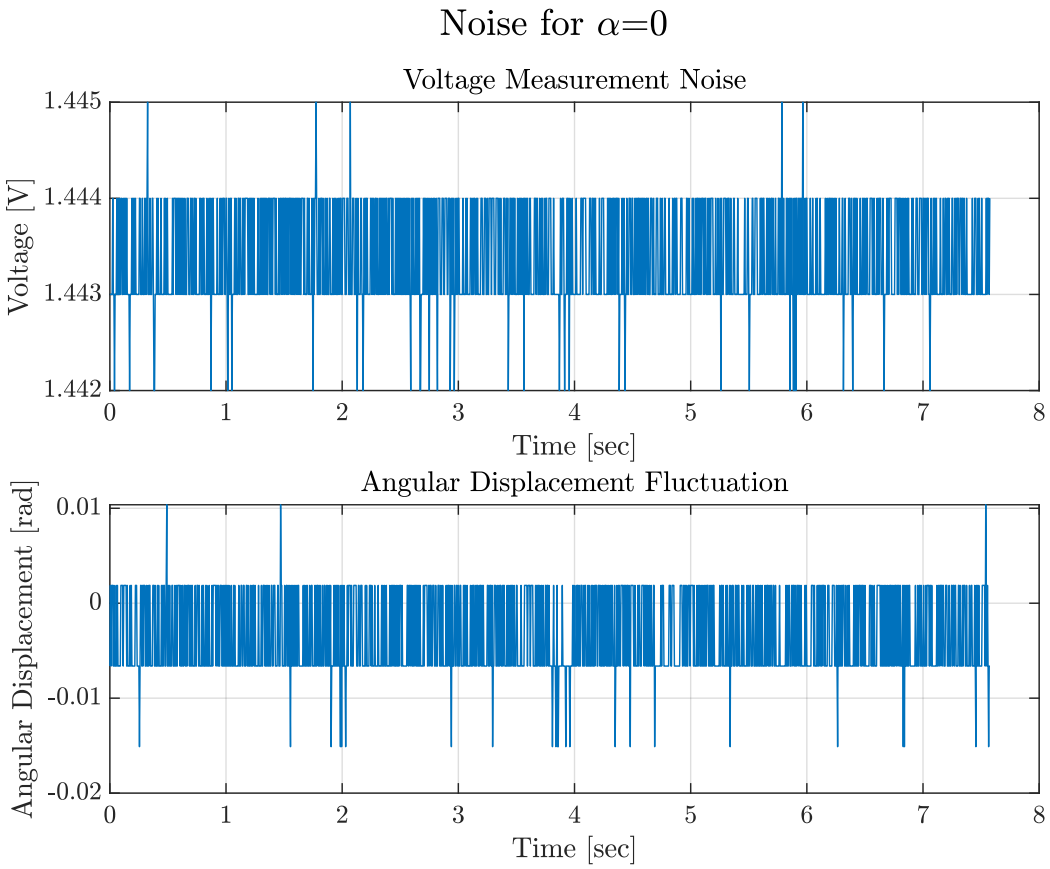

Figure 13: Noise from the rate-gyro. The noise in the rate-gyro voltage measurements and the corresponding fluctuations in the angular measurements are shown in the above plots with fluctuation of 0.003V and 0.004 radians respectively.

Filter of rate-gyro: Since the rate-gyro is also an analog sensor, it requires a filter to get clean results. The strength of the filter is varied from 0.2, 2, and 30 s, while the frequency at which the pendulum rotated was varied from approximately 0.5, 1, and 2 Hz. These results are shown in figure 14. As the cutoff frequency is increased, generally the rate-gyro is able produce better match to the encoder values as the magnitude error and phase lag decrease. This is shown through the columns. As we go up along the columns, the rate-gyro produces better results. Similarly, as the pendulum rotation frequency increase along the rows from

left to right, the rate-gyro produces smaller magnitude error and phase lag. Considering that the trend indicates better match to result as we go up along the column, the first upper left graph for filter of 0.2 Hz at a frequency of 0.5Hz, does not fit the data well compared to those below in its column; however, this graph data were taken at 0.5Hz instead of 0.7 Hz rotation that the other two graphs on the same column were taken. Thus, the general trend stated before matches for all the graphs. Table 2 provides the magnitude error and phase difference for each of the plots shown in figure 14. Thus, the rate-gyro produces the best results for a high pass filter of 30 Hz with high frequency (2Hz) rotations of the pendulum. The rate-gyro works well with a high pass filter because we calculate angular displacement by integrating angular velocity, which cause the noise to be reduced. Only when there is high frequency signal does the angular displacement found from the integral of angular velocity, result in any substantial data as shown through the figure below. Due to the filter, the signal noise from the sensors have been reduced and are not present in the data collected. There is a difference in the peak to peak fluctuation of the sinusoidal curve; however, this is due mainly to the pendulum rotating manually.

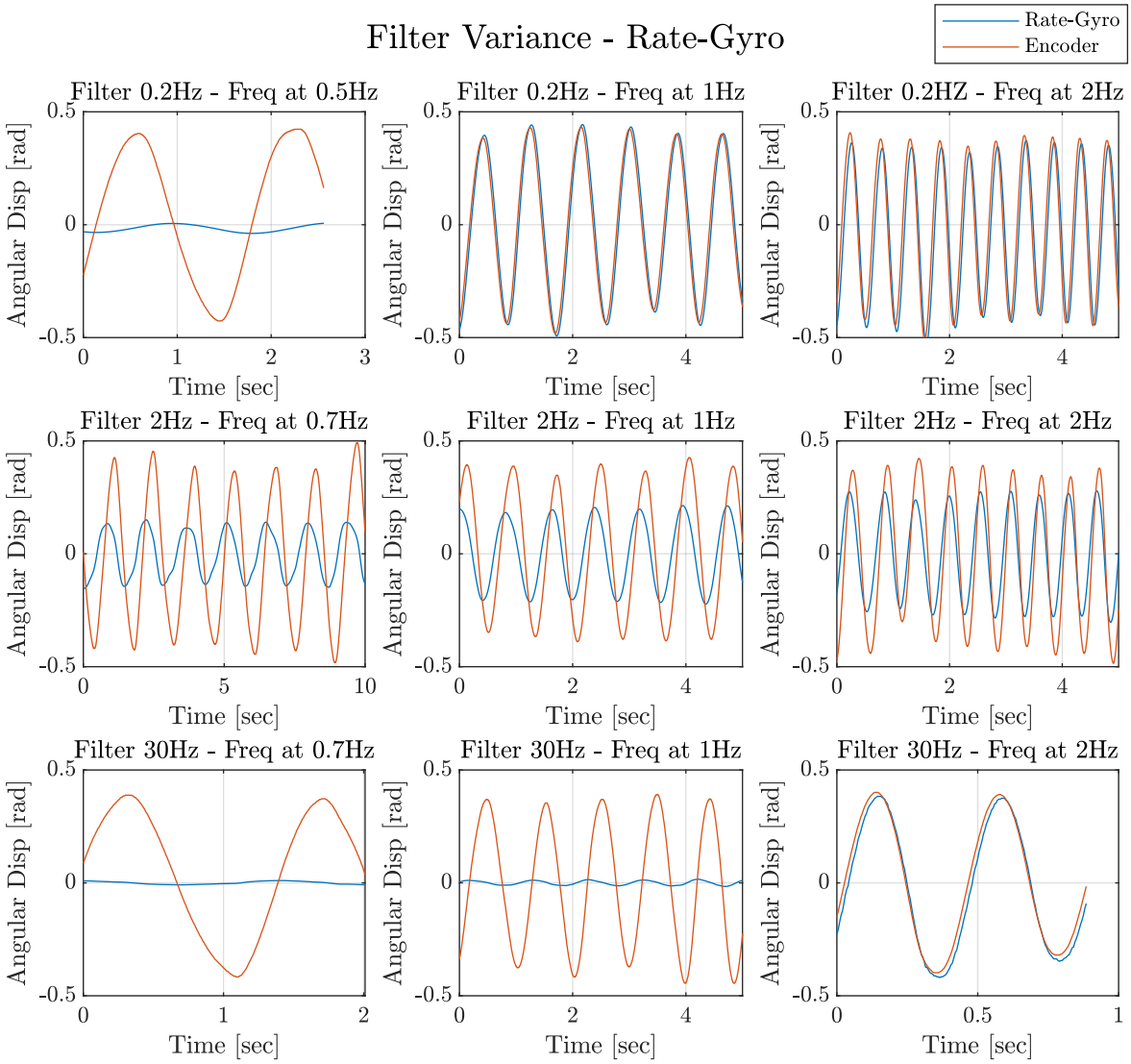

Figure 14: The time domain plots for the rate-gyro with encoder as the filter time is changed and the frequency at which the pendulum rotates is varied as well. The rows represent the different filter times of 0.2, 2, 30 Hz and the columns represent the different frequency at which the pendulum rotates, approximately at 0.5, 1, and 2 Hz.

Table 2: Magnitude Error and Phase Lag associated with Rate-Gyro Filter

Filter [Hz]	Pendulum Swing [Hz]	Magnitude Error [%]	Phase Difference [deg]
0.2	0.5	98.7	83.7
0.2	1	-2.6	10.8
0.2	2	9.4	18
2	0.7	69.98	-66.78
2	1	50	-36
2	2	27.26	-25.2
30	0.7	94.7	-70
30	1	97	-81
30	2	3.8	7.2

3 High Frquency Characterization (10pts)

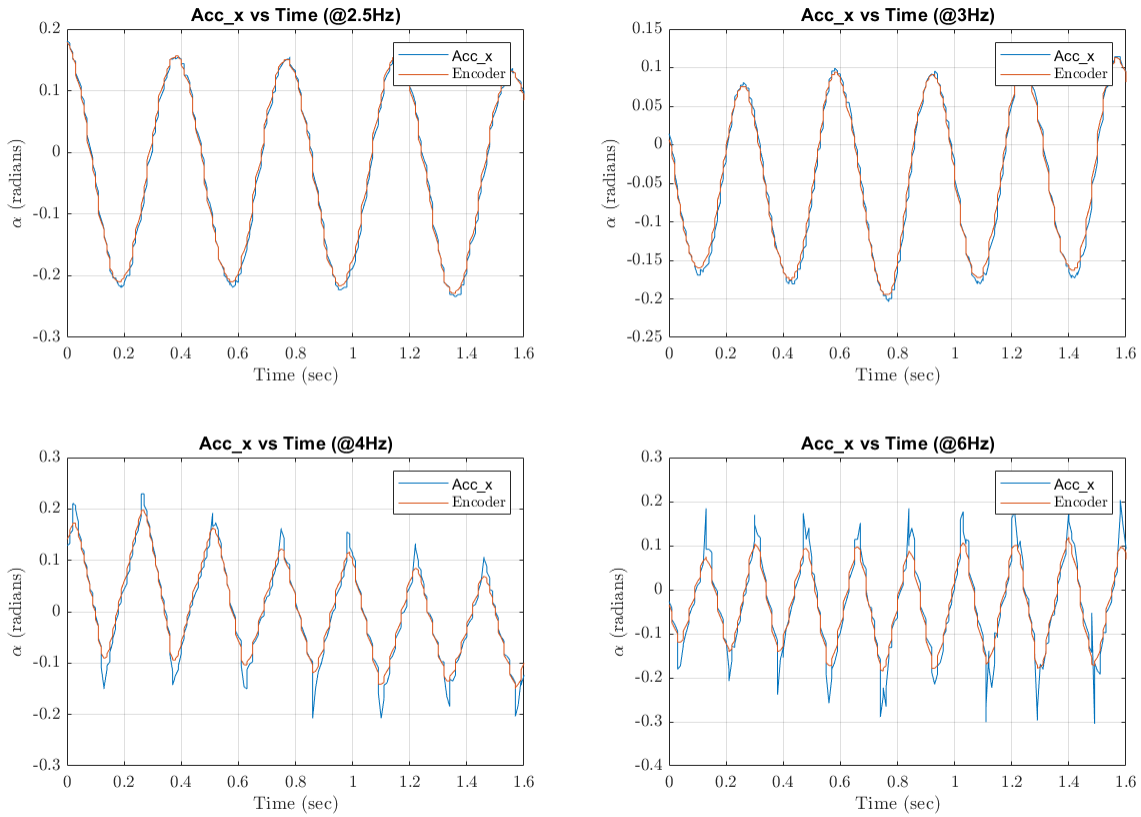

Figure 15: High frequency characterization for accelerometer oriented in horizontal direction without filter.

To analyze the performance of the accelerometer when oriented in horizontal direction, the pendulum is oscillated at high frequencies (> 2Hz) and the sensor output is recorded. Sensor performance is plotted at 2.5Hz, 3Hz, 4Hz and 6Hz. As can be seen in figure 15, the accelerometer matches the command really well at 2.5Hz and 3Hz, with magnitude error being less than 10% and phase lag being well within range of 10°. At frequencies higher than that, i.e. at 4Hz and 6Hz it generates noise at the extreme positions, hence generating a magnitude error of 17% at 4Hz and 6Hz. This performance is as expected, as the accelerometer is supposed to work for low frequencies because of the small angle approximations used for calculating angle when the accelerometer is used as a tiltmeter.

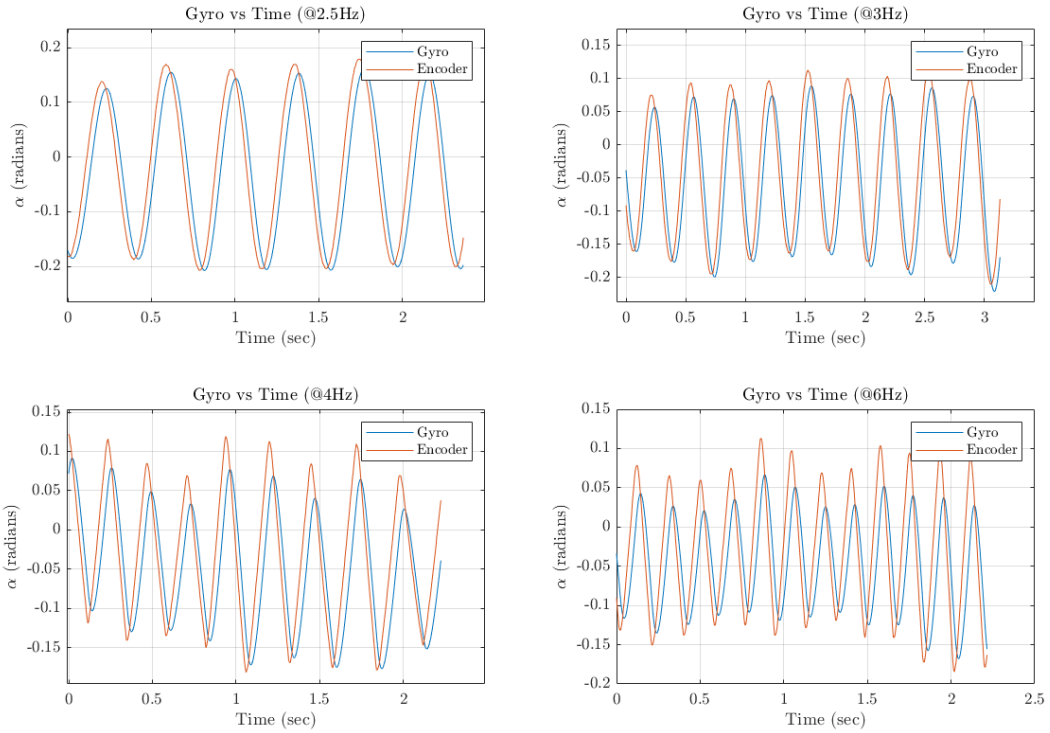

Figure 16: High frequency characterization for single-rate gyroscope without filter.

For the second phase of experiment to test the high frequency performance, the sensor output from the gyroscope are observed. As shown in figure 16, the gyroscope readings are taken at 2.5Hz, 3Hz, 4Hz and 6Hz, same as was for accelerometer. It is observed that for frequencies 2.5Hz and at 3Hz the gyroscope performs really well, with less magnitude error ($< 10\%$) and phase lag ($< 10°$). It starts to deteriorate as the frequency is increased, with the output not matching the commanded angle and the phase lead increasing with increasing operating frequency. This performance of the gyroscope at higher frequencies is not as expected, as the angle measurement of the gyroscope linearly depends on the frequency of oscillations of the proof mass. But in this case of our experiment, the gyroscope fails to perform well at high frequencies. The rationale of the gyroscope to fail is that the gyroscope is not a sensor to directly measure the angle, but rather the rate of the angle. The rate of the angle

or the angular velocity is then integrated to estimate the angle. This method adds in a drift-component which adds up (integrates) over time if there is even a small change of angle in the beginning of the experiment, which was 0.017 radians for this experiment. This offset gets added up over time and can be observed as the plots seems to drift in the y-axis.

4 Sensor Fusion (14pts)

(a) Sensor Fusion Sensing Scheme

The sensor fusion is a method of using complementary sensors such as the accelerometer and the rate-gyro sensors. As seen in the previous sections, an accelerometer provides good angle measurement at low frequencies and a rate-gyro provides a good angle measurement at high frequencies. By combining the two sensors, the sensor fusion is able to provide good angle measurement at a wide range of frequencies. The accelerometer is physical mounted perpendicular to the plane of pendulum's rotation near the pendulum's center of rotation. Similarly, the rate-gyro is mounted parallel to the plane of pendulum's rotation at the pendulum's center of rotation. Thus, the accelerometer collects the voltage of the x and y angular displacements, while the rate-gyro collects the voltage of the angular velocity of the sensor. The data collected from each of the sensors are manipulated into the angular displacements through the equations mentioned in previous sections using their sensitivities from the data sheets. To effectively use the complimentary sensors, the angular displacements of each the sensors are passed through a filter and then summed. The accelerometer is used with a low-pass filter because noise is amplified at high frequency, and the rate-gyro is used with a high-pass filter because rate-gyro angular displacements are reduced at low frequency. Thus, the advantage of each sensor is maximized at different frequency ranges.

(b) Sensor Fusion Block Diagram

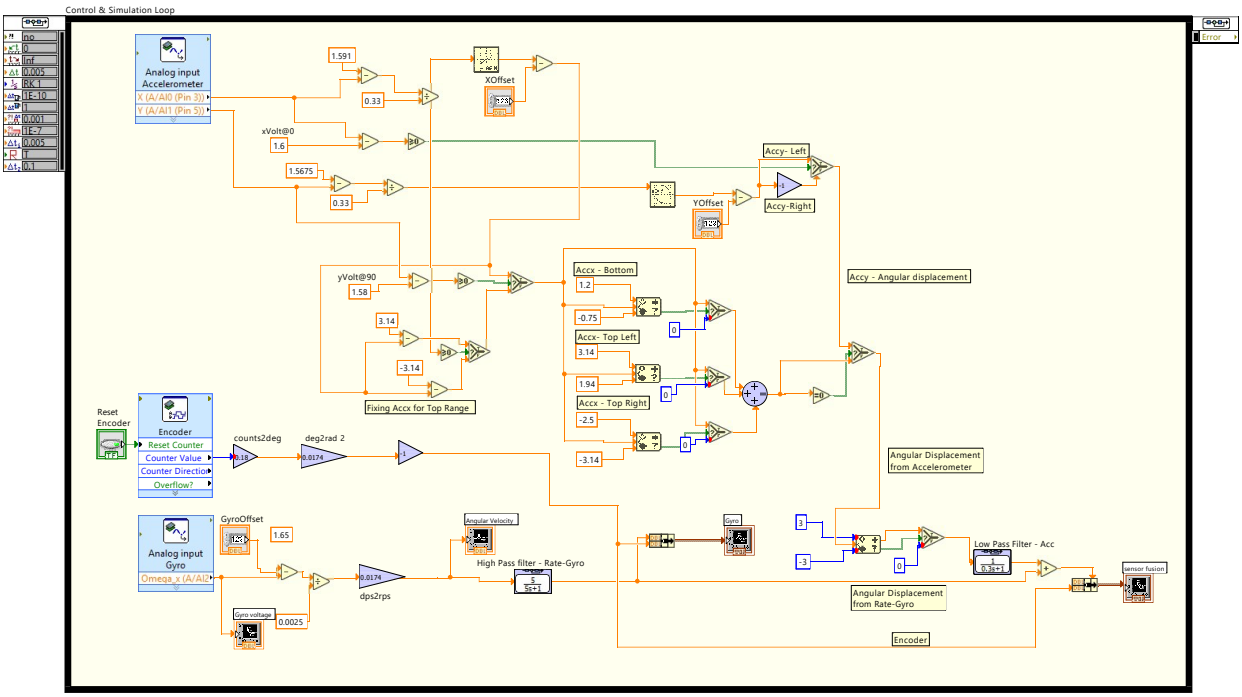

Figure 17: Sensor Fusion Block Diagram.

Fig. 17 shows the sensor fusion block diagram as implemented in LabVIEW. It operates first by selecting the accelerometer, either horizontal or vertical, based on what the perceived position of is using the ranges of motion determined for question 2. If the perceived position does not fall into any of the ranges deemed acceptable for the horizontal accelerometer, the signal defaults to the vertical accelerometer. Once the proper accelerometer is selected it is passed through a low-pass filter such as $\frac{1}{\tau s+1}$. Simultaneously, the angular position integrated from the gyro measurement is passed through a high-pass filter such as $\frac{\tau s}{\tau s+1}$. It is important that the τ values from both the filters match or else the complementary sensing will not work as expected. Once these signals are filtered they are added which results in $\frac{\tau s+1}{\tau s+1}$. Theoretically, this should mean that if both sensors are perfectly accurate in their respective frequency ranges that the sum of their filtered signals would equal the absolute position α of the pendulum arm.

(c) Design and tuning of filter for sensor fusion

We chose the filter to be first order. Generally speaking, accelerometers provide reasonable angle estimations up to frequencies of ~ 2 Hz, while gyroscopes begin providing reasonable angle estimations at frequencies $\sim O(10^{-1})$ Hz. Hence, we chose a reference frequency of 1 Hz (or a time constant of 0.15s/rad-s^{-1}) to begin tuning the filters for our sensor fusion scheme. Time constants (τ) of 0.05s/rad-s^{-1}, 0.2s/rad-s^{-1} and 0.25s/rad-s^{-1} were considered too. Each filter thus obtained was tested at a pendulum link oscillation frequency of 2Hz.

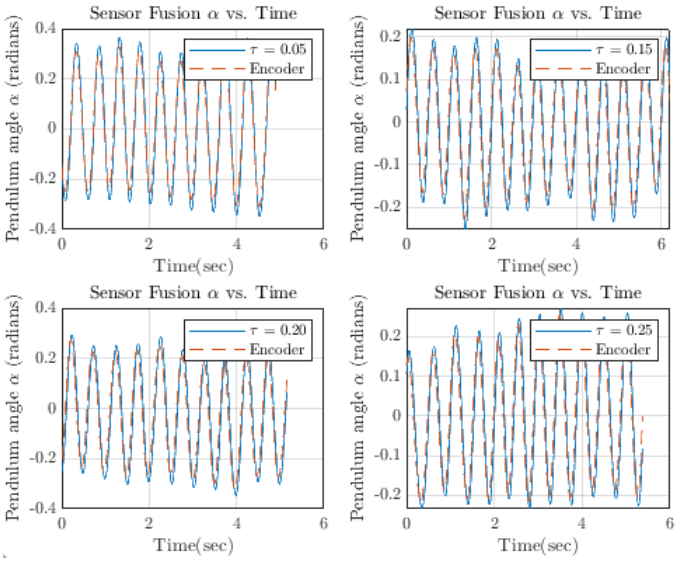

Figure 18: Time domain plots for sensor fusion tuning. Time constants of 0.05s/rad-s^{-1}, 0.15s/rad-s^{-1}, 0.2s/rad-s^{-1} and 0.25s/rad-s^{-1} are considered. During the tests, the pendulum link was oscillated at a frequency of 2 Hz. Unit of τ is s/rad-s^{-1}.

Based on the results shown in Figure 18 , we observed that $\tau = 0.25$s/rad-s^{-1} worked best for our set-up and satisfied the constraints of magnitude error $< 10\%$ and phase lag $< 10\%$.

(d) Effectiveness of Sensor Fusion

Using the filter chosen above and the sensor fusion of the accelerometer and rate-gyro produced time domain graphs shown in figure 19. The effectiveness of the sensors are described in table 3 with pendulum rotation frequency ranging from 0.5 to 3Hz to validate the sensor fusion range. All four frequencies tested in the range satisfy the magnitude error of less than 10% and the phase lag of less than 10 deg. The angular displacement results produced after the filter show very little noise and is not distinguishable in the plots produced. Especially for high frequencies, the encoder began to show quantization, while the sensor fusion produced smooth results. At α=0 in figure 20, the angular displacement fluctuation was 0.057deg.

Comparing the results from the accelerometer and rate-gyro, the sensor fusion provides the best results. The accelerometer produced best results at low frequency, while the rate-gyro

produced best results at high frequency. Even in their respective frequency ranges, the sensors for both the accelerometer and rate-gyro provided magnitude error. By combining the accelerometer and rate-gyro for the sensor fusion, the magnitude error and the phase lag were reduced.

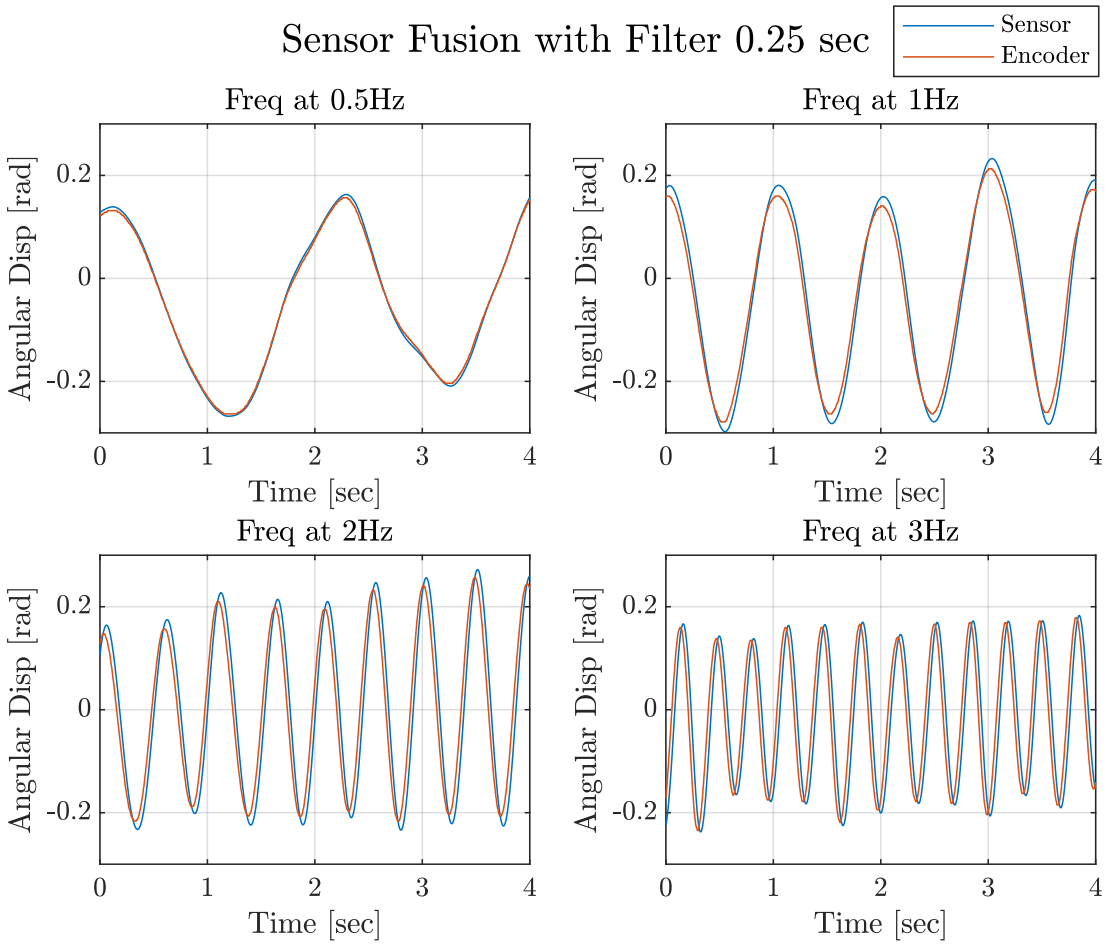

Figure 19: Sensor Fusion Scheme. The effectiveness of the sensor fusion scheme is shown for low and high frequency ranges from 0.5 to 3Hz.

Table 3: Magnitude Error and Phase Difference for Sensor Fusion

Pendulum Rotation [Hz]	Magnitude Error [%]	Phase Difference [deg]
0.5	3.96	3.6
1	9.25	5.4
2	8.1	9.72
3	2.4	9.97

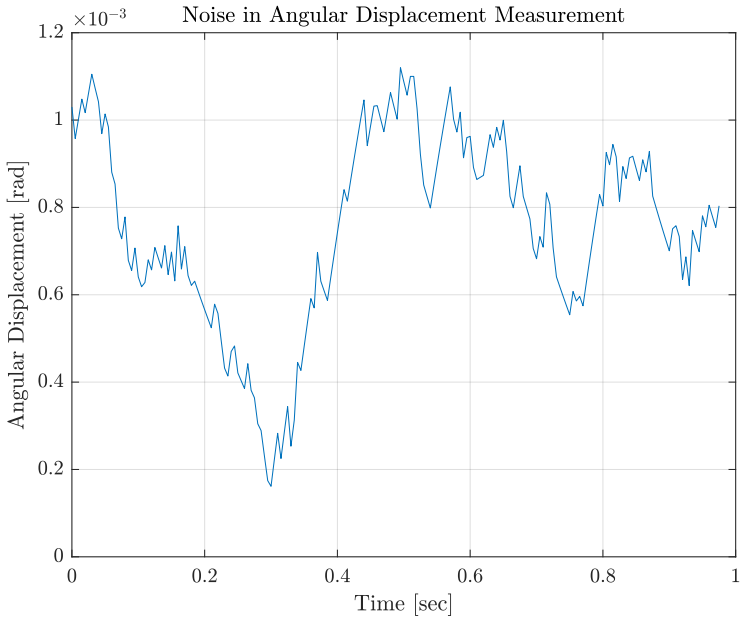

Figure 20: Sensor Fusion Noise for voltage and angular displacement at $\alpha = 0$.

Appendix A: Source Code

Listing 1: MATLAB code for Accelerometer Characterization.

```
%% range of motion x time
close all
dataAX = Q22iAccxv3.accx;
dataEX = Q22iAccxv3.enc;
timesX = Q22iAccxv3.VarName1;

figure();
hold on
plot(timesX,dataEX, timesX, dataAX);
xlabel('Time (sec)','interpreter','latex');
ylabel('$\alpha$ [rad]','interpreter','latex');
legend({'Encoder','Accelerometer'},'interpreter','latex');
title('Horizontal Axis Range of Motion','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'xtimedomain.png');

figure();
hold on
%border = 10.*ones(size(dataEX));
%rn = linspace(min(dataEX), max(dataEX), size(dataEX, 1));
plot(dataEX, 100.*((dataAX-dataEX)./dataEX), 'LineStyle', 'none', '
   Marker', '.')
%plot(rn, border, rn, -1.*border);
xlabel('$\alpha$ [rad]','interpreter','latex');
ylabel('Magnitude Error [\%]','interpreter','latex');
ylim([-10, 10])
title('Horizontal Axis Magnitude Error','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'xmagerr.png');

%% Sensitivity of X direction
close all
dataAX = Q22iAccxv3.voltage;
dataEX = Q22iAccxv3.enc;
timesX = Q22iAccxv3.VarName1;
dA = diff(dataAX);
dE = diff(dataEX);
sens = dA./dE;
m = movmean(sens, 20);
%sens = diff(dataAX)./diff(dataEX);
```

```
figure();
hold on
plot(dataEX(2:end), sens);
plot(dataEX(2:end), m, 'LineStyle', 'none', 'Marker', '.');
xlabel('$\alpha$ [rad]','interpreter','latex');
ylabel('Sensitivity, $\frac{\delta V}{\delta \alpha}$  $[\frac{V}{
   rad}]$','interpreter','latex');
title('Sensitivity of Horizontal Axis Versus $\alpha$','interpreter'
   ,'latex');
legend({'Calculated Sensitivity','Averaged Sensitivity'},'
   interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'xsens.png');

%% Noise in X direction
close all
dataAXv = Q22iAccxvoltage.voltage;
%dataAXv = dataAXv - nanmean(dataAXv);
dataAX = Q22iAccxvoltage.accx;
dataAX = dataAX - nanmean(dataAX);
dataEX = Q22iAccxvoltage.enc;
timesX = Q22iAccxvoltage.VarName1;
figure();
hold on
subplot(2, 1,1)
plot(timesX, dataAXv);
xlabel('Time (sec)','interpreter','latex');
ylabel('Volts [V]','interpreter','latex');
title('Voltage Noise Over Time','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on

subplot(2, 1, 2)
plot( timesX, dataAX);
xlabel('Time (sec)','interpreter','latex');
ylabel('$\alpha$ [rad]','interpreter','latex');
title('Position Noise Over Time','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'xnoise.png');


%% range of motion y time
```

```
close all
dataAY = Q22iAccyv3pi.accypi;
dataEY = Q22iAccyv3pi.enc;
timesY =  linspace(0, 25, size(dataAY, 1));
dataAYN = Q22iAccyv3pin.accypi;
dataEYN = Q22iAccyv3pin.enc;
timesYN =  linspace(0, 25, size(dataAYN, 1));
figure();
hold on
subplot(2, 1,1)
plot(timesY,dataEY, timesY, dataAY);
xlabel('Time (sec)','interpreter','latex');
ylabel('$\alpha$ [rad]','interpreter','latex');
title('Vertical Axis Range of Motion $\alpha$=[0, $\pi$]','
   interpreter','latex');
legend({'Encoder','Accelerometer'},'interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on

subplot(2, 1, 2)
plot(timesYN,dataEYN, timesYN, dataAYN);
xlabel('Time (sec)','interpreter','latex');
ylabel('$\alpha$ [rad]','interpreter','latex');
title('Vertical Axis Range of Motion $\alpha$=[$-\pi$, 0]','
   interpreter','latex');
legend({'Encoder','Accelerometer'},'interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'ytimedomain.png');

figure();
hold on
% border = 10.*ones(100);
% rn = linspace(min(dataEYN), max(dataEY), 100);
plot(dataEY, 100.*((dataAY-dataEY)./dataEY), 'LineStyle', 'none', '
   Marker', '.')
plot(dataEYN, 100.*((dataAYN-dataEYN)./dataEYN), 'LineStyle', 'none'
   , 'Marker', '.')
%%plot(rn, border, rn, -1.*border);
xlabel('$\alpha$ [rad]','interpreter','latex');
ylabel('Magnitude Error [\%]','interpreter','latex');
title('Vertical Axis Magnitude Error','interpreter','latex');
ylim([-10, 10])
grid on
set(gca,'ticklabelinterpreter','latex');
```

```
box on
saveas(gca,'ymagerr.png');

%% Sensitivity of Y direction
close all
dataAY = Q22iAccyv3pi.voltage;
dataEY = Q22iAccyv3pi.enc;
dataAYN = Q22iAccyv3pin.voltage;
dataEYN = Q22iAccyv3pin.enc;
dA = diff(dataAY);
dE = diff(dataEY);
sens = dA./dE;
dAN = diff(dataAYN);
dEN = diff(dataEYN);
sensN = dAN./dEN;
m = movmean(sens, 20);
mN = movmean(sensN, 20);
%sens = diff(dataAX)./diff(dataEX);
figure();
hold on
plot(dataEY(2:end), sens, 'b');
%plot(dataEY(2:end), m);
plot(dataEYN(2:end), sensN, 'b');

plot(dataEY(2:end), m, 'LineStyle', 'none', 'Marker', '.', '
   MarkerFaceColor', 'g', 'MarkerEdgeColor', 'g');
plot(dataEYN(2:end), mN, 'LineStyle', 'none', 'Marker', '.', '
   MarkerFaceColor', 'g', 'MarkerEdgeColor', 'g');
xlabel('$\alpha$ [rad]','interpreter','latex');
ylabel('Sensitivity, $\frac{\delta V}{\delta \alpha}$  $[\frac{V}{
   rad}]$','interpreter','latex');
title('Sensitivity of Vertical Axis Accelerometer Versus $\alpha$','
   interpreter','latex');
legend({'Calculated Sensitivity','Averaged Sensitivity'},'
   interpreter','latex');
ylim([-1, 1]);
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'ysens.png');

%% Noise in Y direction
close all
dataAYv = Q22iAccyvoltage.AmplitudePlot2;
dataAY = Q22iAccyvoltage.AmplitudePlot0;
%dataAYv = dataAYv - nanmean(dataAYv);
dataAY = dataAY - nanmean(dataAY);
```

```
%dataEY = Q22iAccyvoltage.enc;
timesY = [0:0.005:25];
timesY = timesY(1:size(dataAYv,1));
figure();
hold on
subplot(2, 1,1)
plot(timesY, dataAYv);
xlabel('Time (sec)','interpreter','latex');
ylabel('Volts [V]','interpreter','latex');
title('Voltage Noise Over Time ','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on

subplot(2, 1, 2)
plot( timesY, dataAY);
xlabel('Time (sec)','interpreter','latex');
ylabel('$\alpha$ [rad]','interpreter','latex');
title('Position Noise Over Time','interpreter','latex');
grid on
set(gca,'ticklabelinterpreter','latex');
box on
saveas(gca,'ynoise.png');


%% filter strengths
%data 0 is accelerometer, data1 is encoder
T02 = 5;
T30 = 0.005;
T2 = 0.08;
dataA = cf02Accx05Hz.AmplitudePlot0;
dataE = cf02Accx05Hz.AmplitudePlot1;
times = cf02Accx05Hz.TimePlot0;
[peaksA, locsA] = findpeaks(dataA - nanmean(dataA), "MinPeakDistance
   ", 100);
[peaksE, locsE] = findpeaks(cf02Accx05Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
magl5 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
periodl5 = mean((dataA(locsE(2:end))-dataA(locsE(1:end-1))));
phasel5 = 360*(dataA(locsA(1))- dataA(locsE(1)))/periodl5;

[peaksA, locsA] = findpeaks(cf02Accx1Hz.AmplitudePlot0 - nanmean(
   cf02Accx1Hz.AmplitudePlot0), "MinPeakDistance", 200);
[peaksE, locsE] = findpeaks(cf02Accx1Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
```

```
magm5 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
periodm5 = mean((cf02Accx1Hz.TimePlot0(locsE(2:end))-cf02Accx1Hz.
   TimePlot0(locsE(1:end-1))));
phasem5 = 360*(cf02Accx1Hz.TimePlot0(locsA(1))- cf02Accx1Hz.
   TimePlot0(locsE(1)))/periodm5;

[peaksA, locsA] = findpeaks(cf02Accx2Hz.AmplitudePlot0  - nanmean(
   cf02Accx2Hz.AmplitudePlot0), "MinPeakDistance", 200);
[peaksE, locsE] = findpeaks(cf02Accx2Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
magh5 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
periodh5 = mean((cf02Accx2Hz.TimePlot0(locsE(2:end))-cf02Accx2Hz.
   TimePlot0(locsE(1:end-1))));
phaseh5 = 360*(cf02Accx2Hz.TimePlot0(locsA(1))- cf02Accx2Hz.
   TimePlot0(locsE(1)))/periodh5;



[peaksA, locsA] = findpeaks(cf2Accx07Hz.AmplitudePlot0 - nanmean(
   cf2Accx07Hz.AmplitudePlot0), "MinPeakDistance", 200);
[peaksE, locsE] = findpeaks(cf2Accx07Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
magl08 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
periodl08 = mean((cf2Accx07Hz.TimePlot0(locsE(2:end))-cf2Accx07Hz.
   TimePlot0(locsE(1:end-1))));
phasel08 = 360*(cf2Accx07Hz.TimePlot0(locsA(1))- cf2Accx07Hz.
   TimePlot0(locsE(1)))/periodl08;

[peaksA, locsA] = findpeaks(cf2Accx1Hz.AmplitudePlot0 - nanmean(
   cf2Accx1Hz.AmplitudePlot0), "MinPeakDistance", 200);
[peaksE, locsE] = findpeaks(cf2Accx1Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
magm08 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
periodm08 = mean((cf2Accx1Hz.TimePlot0(locsE(2:end))-cf2Accx1Hz.
   TimePlot0(locsE(1:end-1))));
phasem08 = 360*(cf2Accx1Hz.TimePlot0(locsA(1))- cf2Accx1Hz.TimePlot0
   (locsE(1)))/periodm08;

[peaksA, locsA] = findpeaks(cf2Accx2Hz.AmplitudePlot0  - nanmean(
   cf2Accx2Hz.AmplitudePlot0), "MinPeakDistance", 200);
[peaksE, locsE] = findpeaks(cf2Accx2Hz.AmplitudePlot1, "
   MinPeakDistance", 100);
numPeaks = min(size(peaksA), size(peaksE));
magh08 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
```

```
247 periodh08 = mean((cf2Accx2Hz.TimePlot0(locsE(2:end))-cf2Accx2Hz.
       TimePlot0(locsE(1:end-1))));
248 phaseh08 = 360*(cf2Accx2Hz.TimePlot0(locsA(1))- cf2Accx2Hz.TimePlot0
       (locsE(1)))/periodh08;
249
250
251
252 [peaksA, locsA] = findpeaks(cf30Accx07Hz.AmplitudePlot0 - nanmean(
       cf30Accx07Hz.AmplitudePlot0), "MinPeakDistance", 200);
253 [peaksE, locsE] = findpeaks(cf30Accx07Hz.AmplitudePlot1, "
       MinPeakDistance", 100);
254 numPeaks = min(size(peaksA), size(peaksE));
255 magl005 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
256 periodl005 = mean((cf30Accx07Hz.TimePlot0(locsE(2:end))-cf30Accx07Hz
       .TimePlot0(locsE(1:end-1))));
257 phasel005 = 360*(cf30Accx07Hz.TimePlot0(locsA(1))- cf30Accx07Hz.
       TimePlot0(locsE(1)))/periodl005;
258
259 [peaksA, locsA] = findpeaks(cf30Accx1Hz.AmplitudePlot0 - nanmean(
       cf30Accx1Hz.AmplitudePlot0), "MinPeakDistance", 200);
260 [peaksE, locsE] = findpeaks(cf30Accx1Hz.AmplitudePlot1, "
       MinPeakDistance", 100);
261 numPeaks = min(size(peaksA), size(peaksE));
262 magm005 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
263 periodm005 = mean((cf30Accx1Hz.TimePlot0(locsE(2:end))-cf30Accx1Hz.
       TimePlot0(locsE(1:end-1))));
264 phasem005 = 360*(cf30Accx1Hz.TimePlot0(locsA(1))- cf30Accx1Hz.
       TimePlot0(locsE(1)))/periodm005;
265
266 [peaksA, locsA] = findpeaks(cf30Accx2Hz.AmplitudePlot0  - nanmean(
       cf30Accx2Hz.AmplitudePlot0), "MinPeakDistance", 100);
267 [peaksE, locsE] = findpeaks(cf30Accx2Hz.AmplitudePlot1, "
       MinPeakDistance", 100);
268 numPeaks = min(size(peaksA), size(peaksE));
269 magh005 = mean(peaksA(1:numPeaks)./(peaksE(1:numPeaks)));
270 periodh005 = mean((cf30Accx2Hz.TimePlot0(locsE(2:end))-cf30Accx2Hz.
       TimePlot0(locsE(1:end-1))));
271 phaseh005 = 360*(cf30Accx2Hz.TimePlot0(locsA(1))- cf30Accx2Hz.
       TimePlot0(locsE(1)))/periodh005;
```