ECE167L

Xingyu Zhou

Max Dunne

2020/02/23

Lab 3 Report

Introduction:

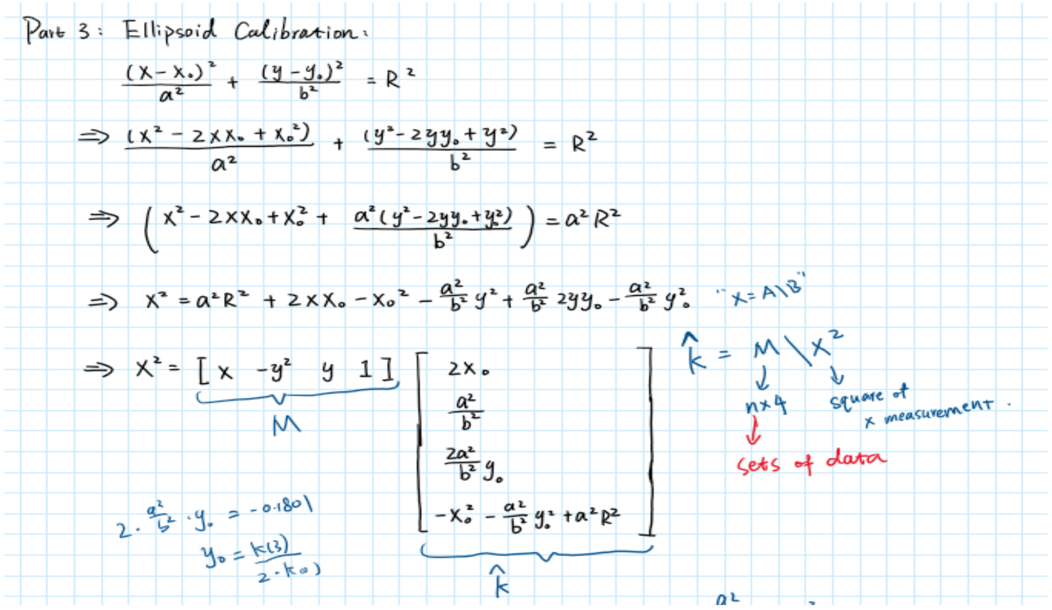
In this lab we are introduced with a new component—the MPU9250 IMU module. This is a device that contains an accelerometer, a gyro and a magnetometer. Our goal in this lab is to get familiar with it and understand how the sensor works. Lastly, we will be using the sensor to take measurements and since the sensor is not perfect, we will need to calibrate it so that the readings are somewhat acceptable.

Part1: Ellipsoid calibration using simulated data

In this part we will be using the simulated data and perform a 2D calibration on the x and y axis of the accelerometer. The first thing we need to do is to expand the ellipse function that the lab manual provided, which is:

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}=R^2$$

Figure 1 shows the complete process of how I got to the final answer in the matrix form.



Part 3: Ellipsoid Calibration:

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}=R^2$$

$$\Rightarrow \frac{(x^2-2xx_0+x_0^2)}{a^2}+\frac{(y^2-2yy_0+y_0^2)}{b^2}=R^2$$

$$\Rightarrow \left(x^2-2xx_0+x_0^2+\frac{a^2(y^2-2yy_0+y_0^2)}{b^2}\right)=a^2R^2$$

$$\Rightarrow x^2=a^2R^2+2xx_0-x_0^2-\frac{a^2}{b^2}y^2+\frac{a^2}{b^2}2yy_0-\frac{a^2}{b^2}y_0^2 \quad \text{"}x=A\backslash B\text{"}$$

$$\Rightarrow x^2=\underbrace{[x \quad -y^2 \quad y \quad 1]}_{M}\underbrace{\begin{bmatrix}2x_0\\ \frac{a^2}{b^2}\\ \frac{2a^2}{b^2}y_0\\ -x_0^2-\frac{a^2}{b^2}y_0^2+a^2R^2\end{bmatrix}}_{\hat{k}}$$

$$\hat{k}=M\backslash x^2$$

M: $n\times4$ → sets of data

x^2: square of x measurement.

$$2\cdot\frac{a^2}{b^2}\cdot y_0=-0.1801$$

$$y_0=\frac{k(3)}{2\cdot k(2)}$$

Figure 1: Ellipse function expansion

After getting the matrix representation, I'm now ready to implement the calibration on MATLAB. According to the lab manual, we need to figure out the following parameters: x_0, a, b and y_0. This can be done by using the function $k = M\backslash X^2$. In this function, X stands for the X axis measurements that are provided by the professor and M is just a set of data. As you can see in figure 1, M is another matrix that contains both my x axis measurements and y measurements, and a lot of 1s. By typing the funtion into MATLAB, I got a 4x1 matrix k which contains the values of the variables showed in figure 1 above. The picture on the right shows the exact value I got for the matrix k. After getting these values, I was then able to solve for x_0, a, b and y_0. The method is straightforward as well—simply match each value together, for example, $2x_0 = 0.7923$ to solve for x_0 and continues from there. The results I got were:

```
k =

    0.7923
    0.2658
   -0.1801
    1.0349
```

$$a = 1.1056, b = 2.1445, x_0 = 0.3962, y_0 = -0.3388$$

The next step is to plot the data, but first we need to perform the calibration. To achieve that, we just need to subtract x_0 from X axis measurements and y_0 from Y axis measurements and that was it. Figure 2 shows the plotted data of both post/pre calculated data.

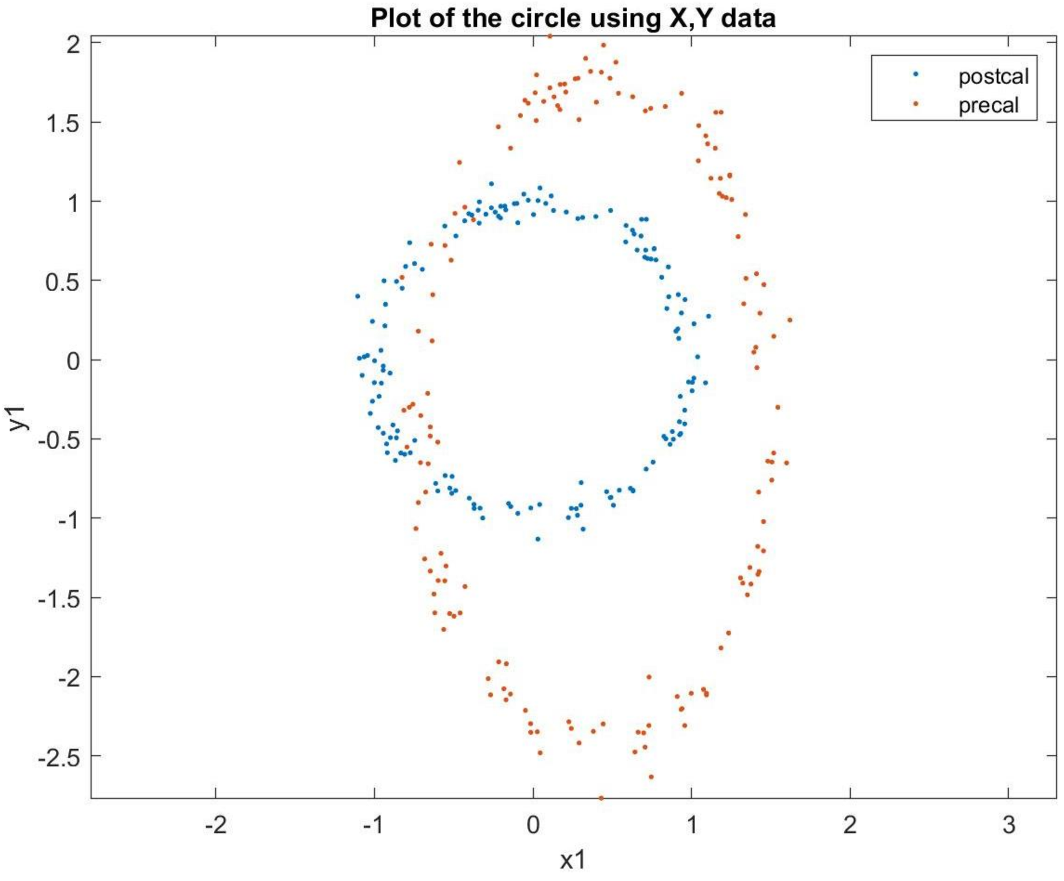


Figure 2: pre vs post calibrated data plot

As you can see in figure 2, the pre-calibrated data looks like an ellipsoid with the data more separated from each other while the post-calibrated data tightens the spread of the data and that's why it looks more like a circle.

Part 4: Naïve Calibration of Magnetometer and Accelerometer

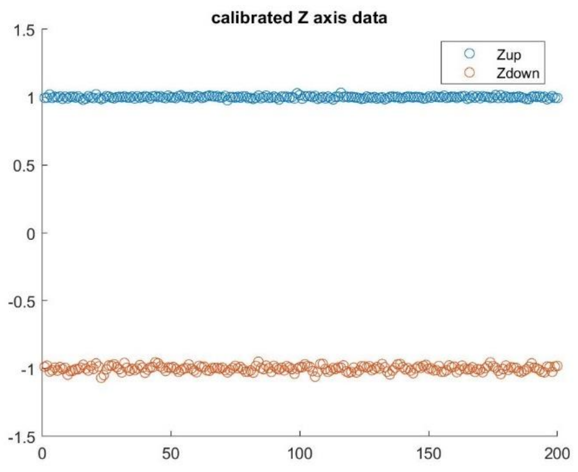
As it's shown in the title, all we need to do in this part is to calibrate the accelerometer as well as the magnetometer (all 3 axis). The first thing we need to do is to calibrate the accelerometer. The method is very simple—on the accelerometer you will see the label telling you the direction of each axis. Then we will need to hold the accelerometer at 90 degrees of both x and y axes (both +90 and -90 degrees) and record the accelerometer readings. For z axis we need to record the value when the sensor is facing straight up and straight down. The lab manual says to record the data at a rate of 50Hz so I added a 20 millisecond delay in my code each time a data is recorded.

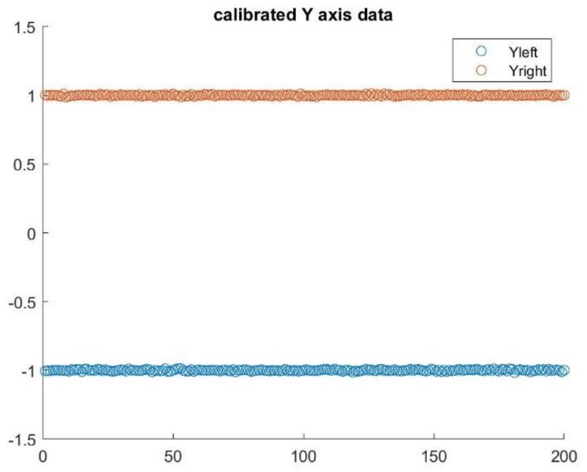
I started by recording the reading for z axis. After getting all the values and loaded them into MATLAB, I took the average of both sets of data (face up\down) since I will be using this value to determine the offset and the scale factor. According to the lab manual, we need to scale our reading into the range of +1 or -1g, which is why I choose this formula $scale = \frac{1}{average}$. the offset is calculated by $offset = \frac{average1+average2}{2}$. average 1 and average 2 represents the average value of the data for both directions. Finally, to get the calibrated data, I just need to simply ***multiply all the raw data by my scale factor*** and plot them out. The steps are the same for both x and y axis but with different direction, so I will be going over the result here directly.

	X	Y	Z
offset	581.74	186.96	1.51e+03
scale	-6.31e-05(up), 5.88e-05(down)	-6.16e-05(left), 6.027e-05(right)	5.5286e-05(up), -6.6385e-05(down)

Table 1: offset and scale for X Y Z axes of the accelerometer

When I plotted out the raw data, I noticed that all three axes were kind of spread out (the range can go from -2 to 2 or maybe 4). By performing the calibration using the values in table 1, here are the results I got for all three axes.





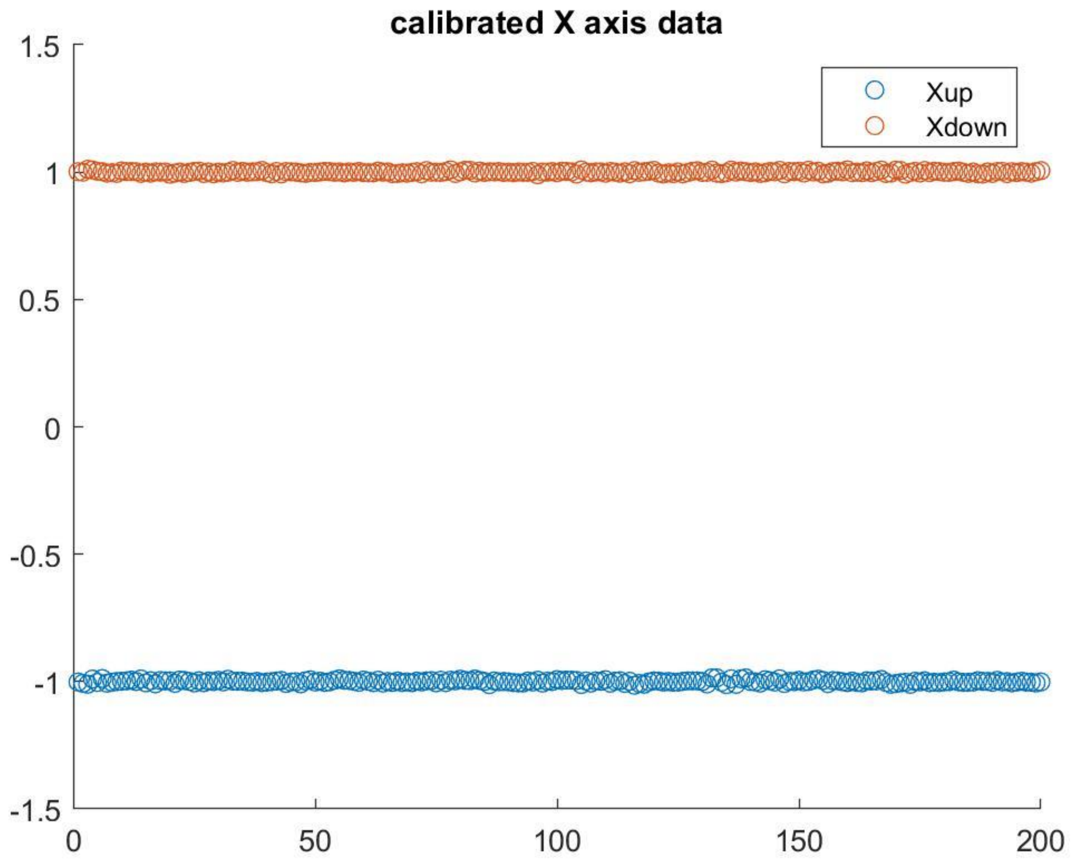


Figure 3: post-calibrated data of X Y Z axes of the accelerometer

As you can see in figure 3 above, after performing the calibration, the data from all three axes seems to be closer to +/- 1g. Which is exactly what the lab manual asks us to do.

The next part is the magnetometer. The magnetometer will be calibrated the same way as we did with the accelerometer, but the data recording process is a little bit different—we need to either rotate the sensor so that the y reading is close to 0 and x reading is the largest (or x close to 0 and y close to the largest), or try to find an angle so that when you are reading the x value, both y and z are 0.

However, one of the tutor told me another method—rotate the sensor to cover more area as much as you can (like a ball shape), then, if I'm doing it correctly, I should be getting a graph similar to figure 4 when I try to plot the raw data out:

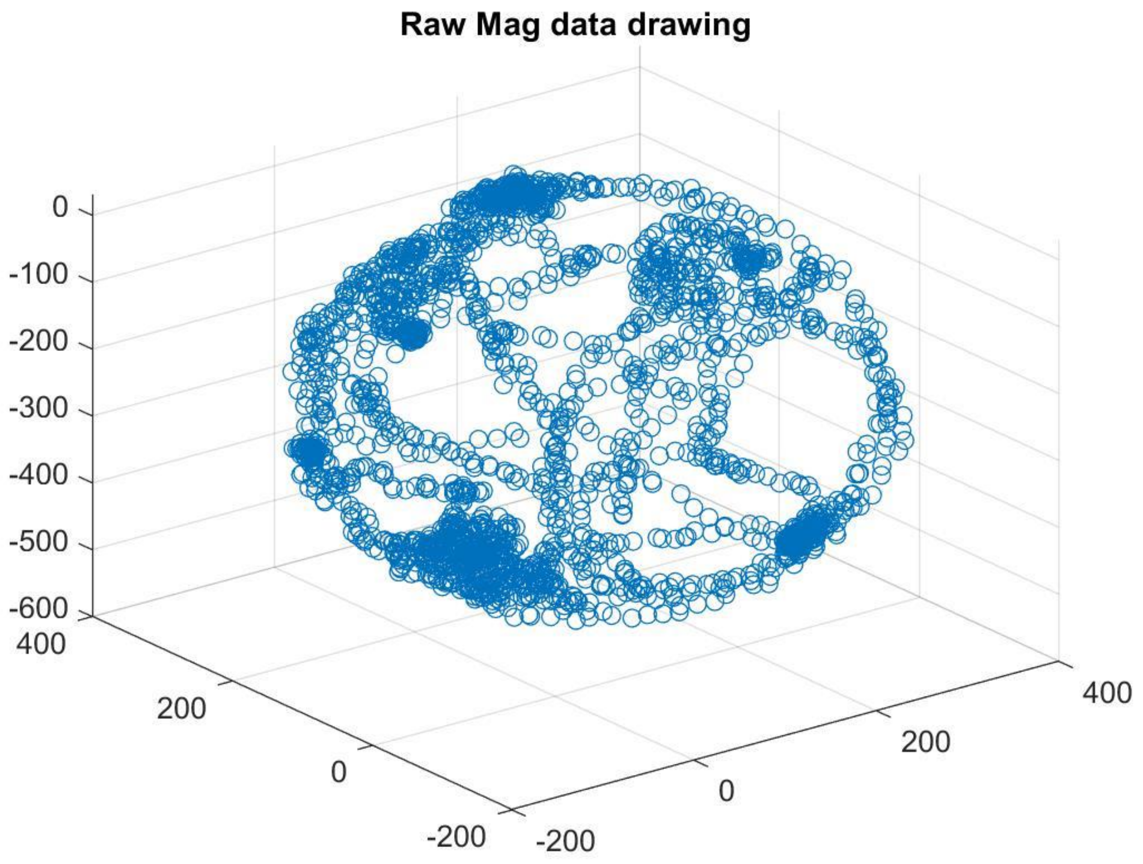


Figure 4: Raw magnetometer data plot

The next thing to do is to figure out the maximum and minimum value for all 3 axes. What I did is to directly take measurements from the graph above (hover the mouse on the dots and you will be able to get the value of each dot), in this case I took 5 samples for both maximum and minimum and I calculate the average values of them. To get the offset, the following formula is used $offset = \frac{avgmax+avgmin}{2}$ and for the scale factor, $scale = \frac{2}{avgmax-avgmin}$ is used. To get the calibrated data, we need to first subtract the offset from the raw reading, and then divide the result by my scale factor. Figure 5 below shows the outcome of the calibration.

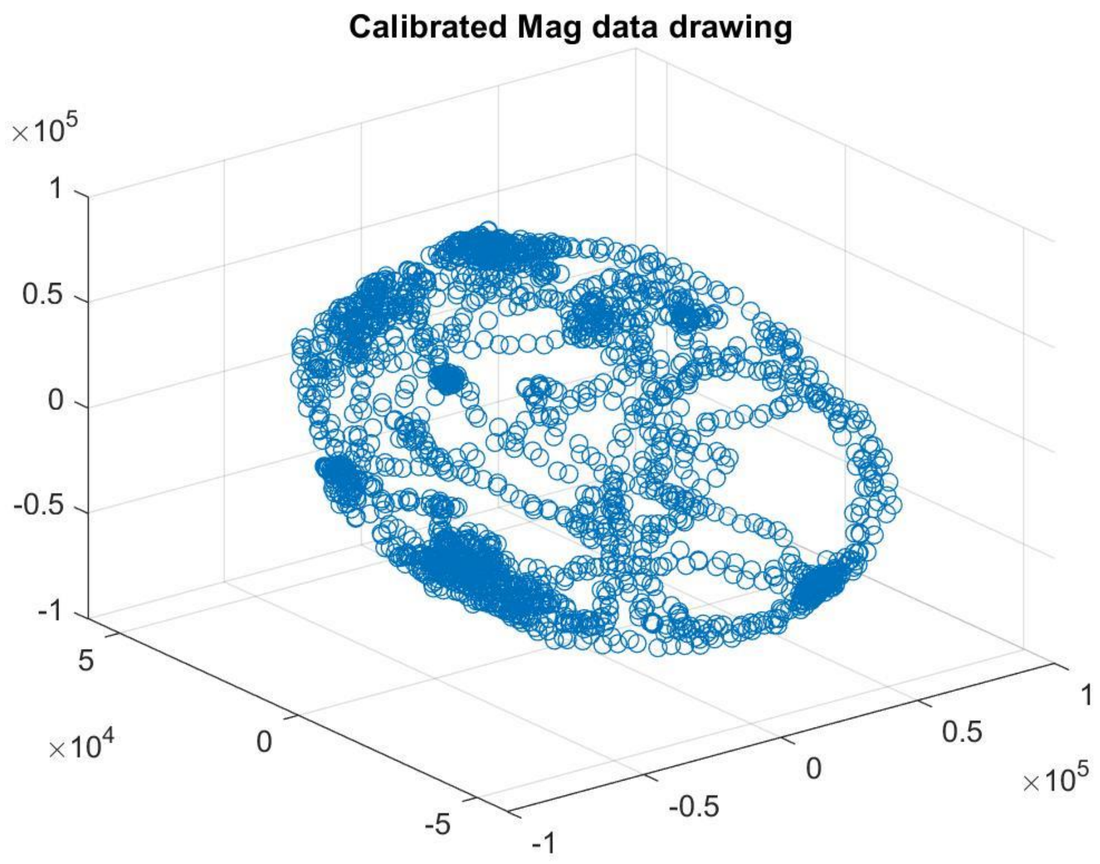


Figure 5: calibrated magnetometer data plot

As you might have noticed in figure 4, the range of my data can go from -200 to 400 or 0 to -600, but after the calibration, you will notice that the range has been limited to -1 to 1 and -5 to 5, which in this case, means that my calibration is successful.

Part 5: Gyro Bias and Bias Drift

In this part we will be working with gyroscopes. Since the sensor that we are given is not very good, when measuring the turning rates (with the sensor is just sitting still on the table), there will be a drift, which means that the gyro is actually turning (the turning rate is changing) even thought it's just sitting on the table. Our job in this part of the lab is to find out the bias as well as the drift for each axis. To begin with, we will need to take a set of data for 10 seconds. This set of data will then be used to determine the bias—it's determined by taking the average of the 10-second data for each axis. Next, we need to let the gyroscope sit still and log the data for an hour. After getting the hour long data, we need to subtract the bias that we got from the raw, hour-long data of each axis and divided all of the result by 131, because we want to convert the raw reading into degree/second, which is the angle rate.

Since we want to see the total change in angles over the length of an hour, we need to integrate the readings for each axis separately. The way to do it is simple—first, we need to multiply the angle rate that we got from above with the sampling rate (in my case is 0.01, which is 100Hz). Then, we need to add the result together so that we can see a graduate change in angles over time. Figure 6 shows my bias drift over the duration of an hour. As it's shown in the figure, there are some significant drift for x and z axis in an hour but since the maximum drift rate on the data sheet is 250 degree/s, I can confirm that my gyro is still working.

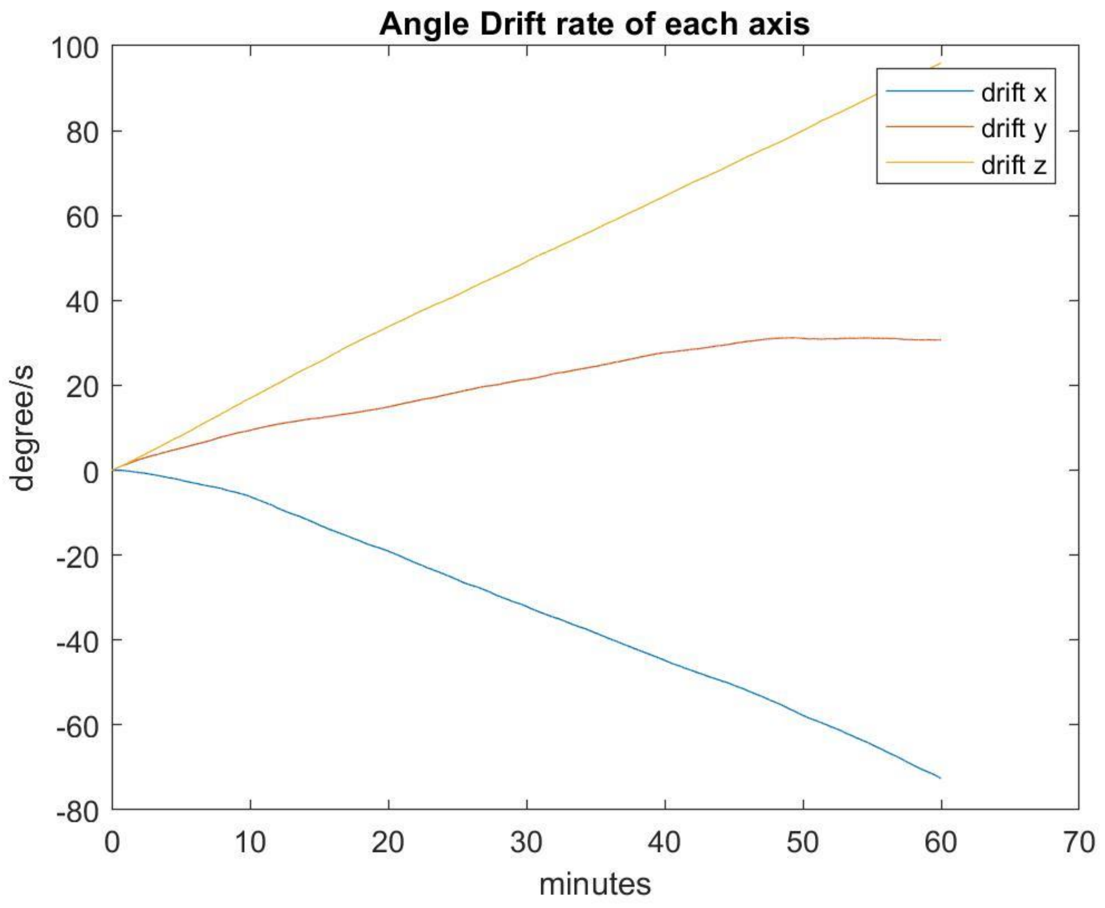


Figure 6: Angle Drift rate of each axis

<u>Part 6: Gyro Scale Factor via Angle Integration</u>

In this part, we need to rotate our sensor at a certain degree and see if the output is the same as we expect it to be. To begin with, we need to take a set of data for 10 seconds and set up bias for each axis. This is the same as part 5 so please refer to part 5 for the process. Then we are asked to rotate the sensor on each axis for a certain degree for a couple of times and record the reading. I choose to rotate my sensor for +/- 90 degrees on each axis since the cable isn't long enough for me to perform a +/- 180-degree rotation.

Then to calculate the scale factor, I used the following equation:

$$scale = \frac{(rawdata - bias) * sample\ rate}{sens\ scale}$$

If you divide the raw data by the scale calculated above, you will get the actual angle of rotation. However, when I'm using the original sens scale, which is 131 (provided by the data sheet), the highest angle I can reach is only 25 degree. I thought it was an error when taking measurements but after recording some more data I was still getting around 25 degree of rotation at most. Thus, I have to change the sens scale so that I can reach the 90 degree that I was expecting. After a few experiments I have found my X axis sens scale to be 400, Y axis sens scale to be 1000 and Z axis sens scale to be 500. Figure 7 shows the change of angles after applying the correct sens scale. There might be some overshoot in the graph but that's because I might have turned the sensor a little bit too far sometimes.

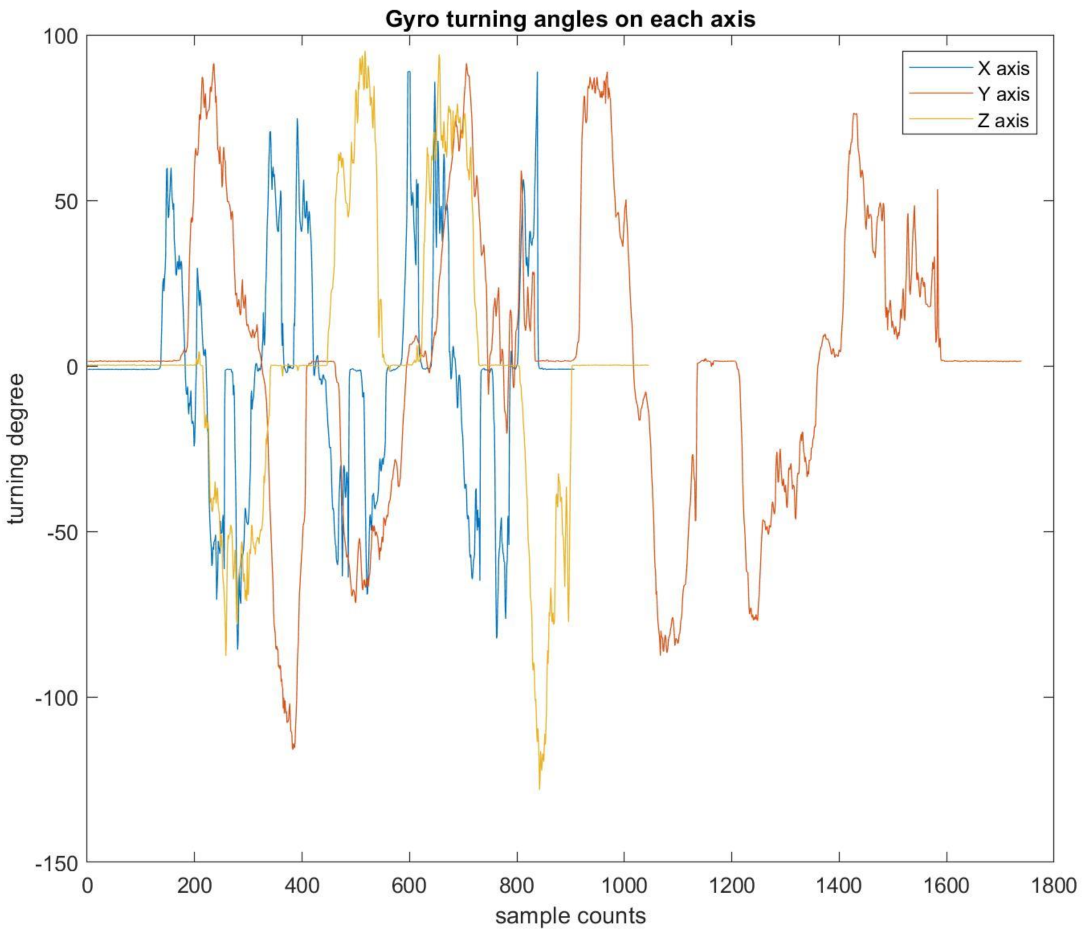


Figure 7: Gyro turning angles on each axis

<u>Part 7: Tumble Test simulation</u>

This is the part which I'm most confused about. I assume what we need to do is first generate the simulated data using the function CreateRandomAttitude.m that is provided to us, and then calibrate the data using both the naïve method and the calibration function provided. The first thing I do is to run the createrandomattitude.m. This function will output the simulated data for both the accelerometer and the magnetometer. Then I go ahead and perform the naïve calibration for both sensors—it's almost the same as previous parts but instead of using the average values, I directly use the maximum and minimum values, so the formulas will be:

$$offset = \frac{(max + \min)}{2}, scale = \max - offset$$

To perform the naïve calibration, all I do is subtract the offset from all my raw readings and divide the result by my scale factor. Figure 8 is the result of the naïve calibration of the magnetometer, as you can see, the data falls mostly into +/-1., which is what the lab manual specified.

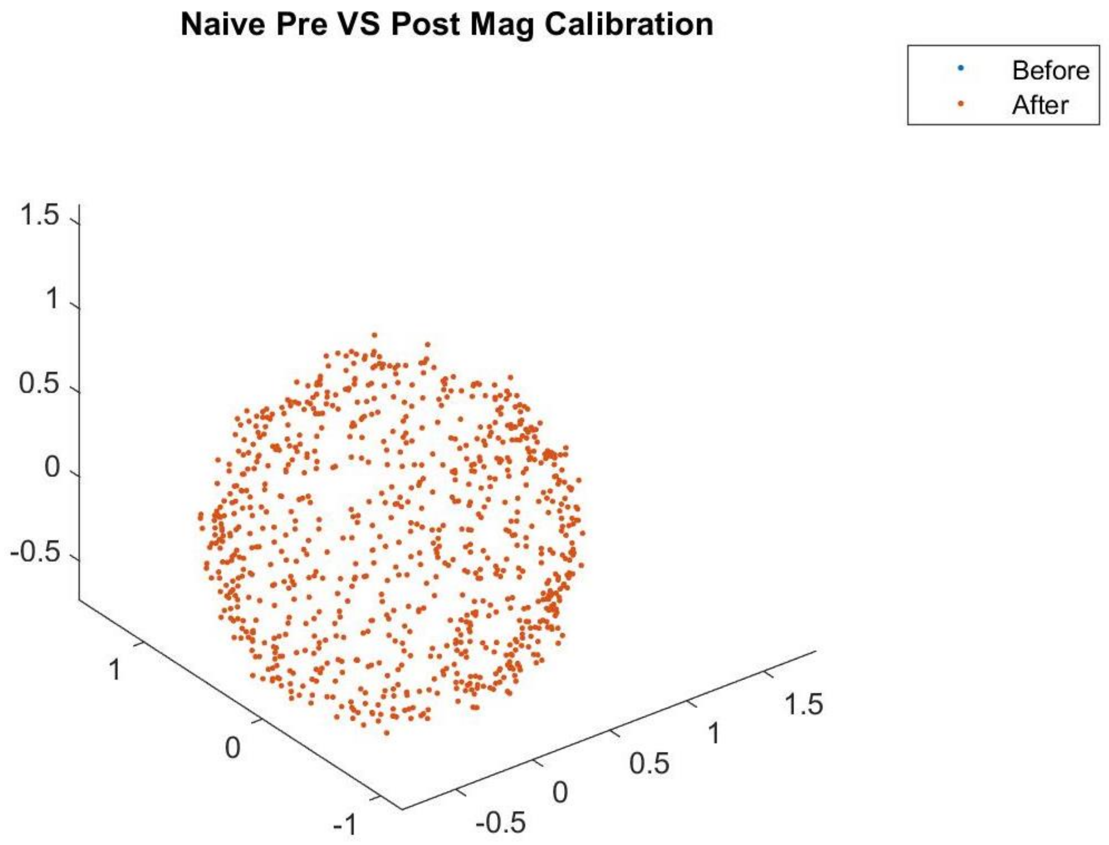


Figure 8: Naïve pre vs post calibration for magnetometer

Once we have the calibrated data, we will then need to feed those data into a function called calibratellipsoiddata3D, this function will give us both Ã and Btilde. These two matrixes will be fed into another function called correctellipsoiddata3d along with my calibrated X Y Z axes data. Finally, what we will get out of the last function is the correct X, Y, Z data that we should expect. Figure 9 is the plot of the calibrated magnetometer using the correctellipsoiddata3d function.

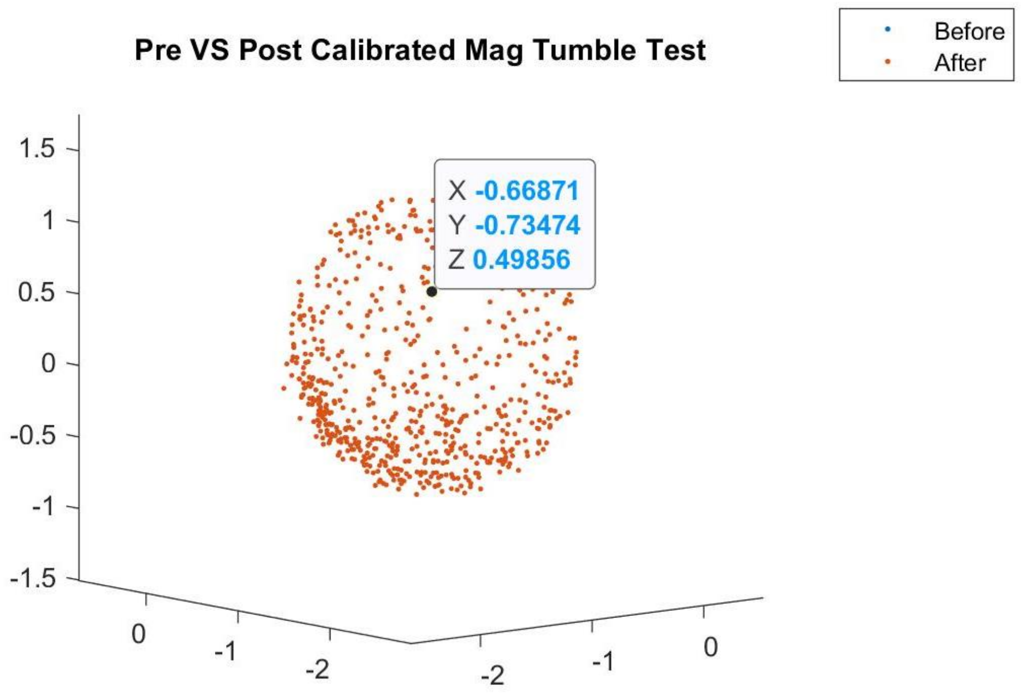


Figure 9: calibrated magnetometer data using the given function

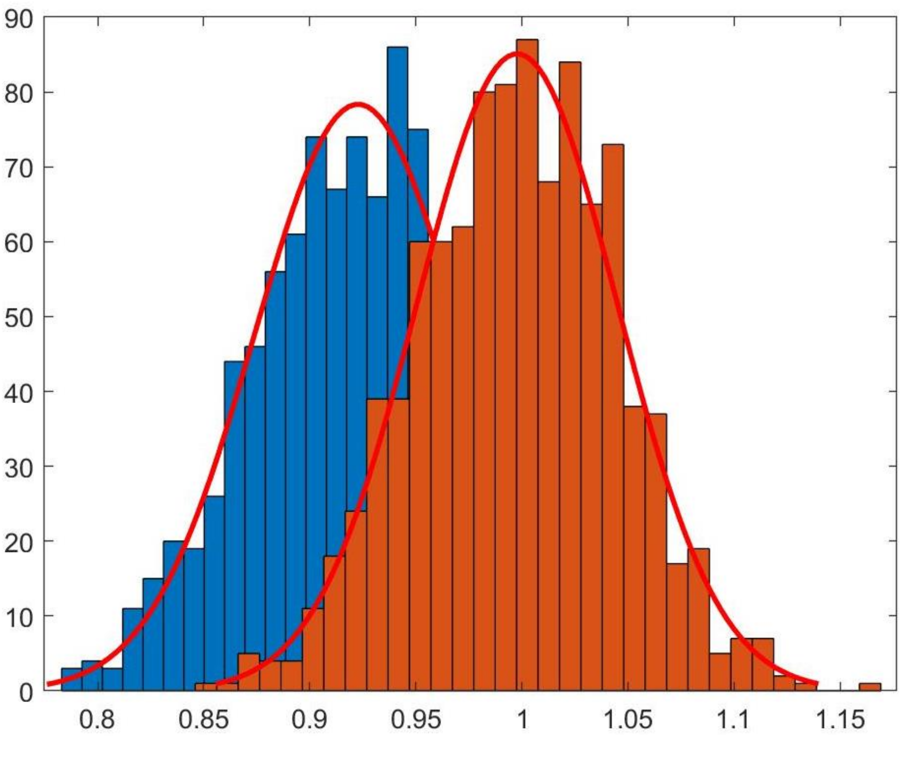


Figure 10: histogram of the accelerometer

The last thing we need to do in this lab is to calculate both the mean and standard deviation for the simulated data, and I will attach the result at the end of the lab report.

Part 8: Tumble Test for Accelerometer and Magnetometer

This is the last part of lab3, and the goal is to run the tumble test again, but instead of using the simulated data, we will be using the actual data that we get from the accelerometer and magnetometer. To begin with, we need to record the data from both the accelerometer and the magnetometer. I just simply rotate the sensor in a circular motion as if it's in an actual tumble test and record the readings of all three axes.

Next, I use all the raw data as inputs to the function *clibrateEllipsoidData3D* as well as the *CorrectEllipsoidData3D*, this will give me the calibrated data using the method that is provided. Since we've already done the calibration in part4 using my own method, I can directly apply it on part 8.

Finally, I plot all of them out and in the following figures you will be able to see the difference between the raw data, the calibrated data by myself as well as the calibrated data by the provided functions.

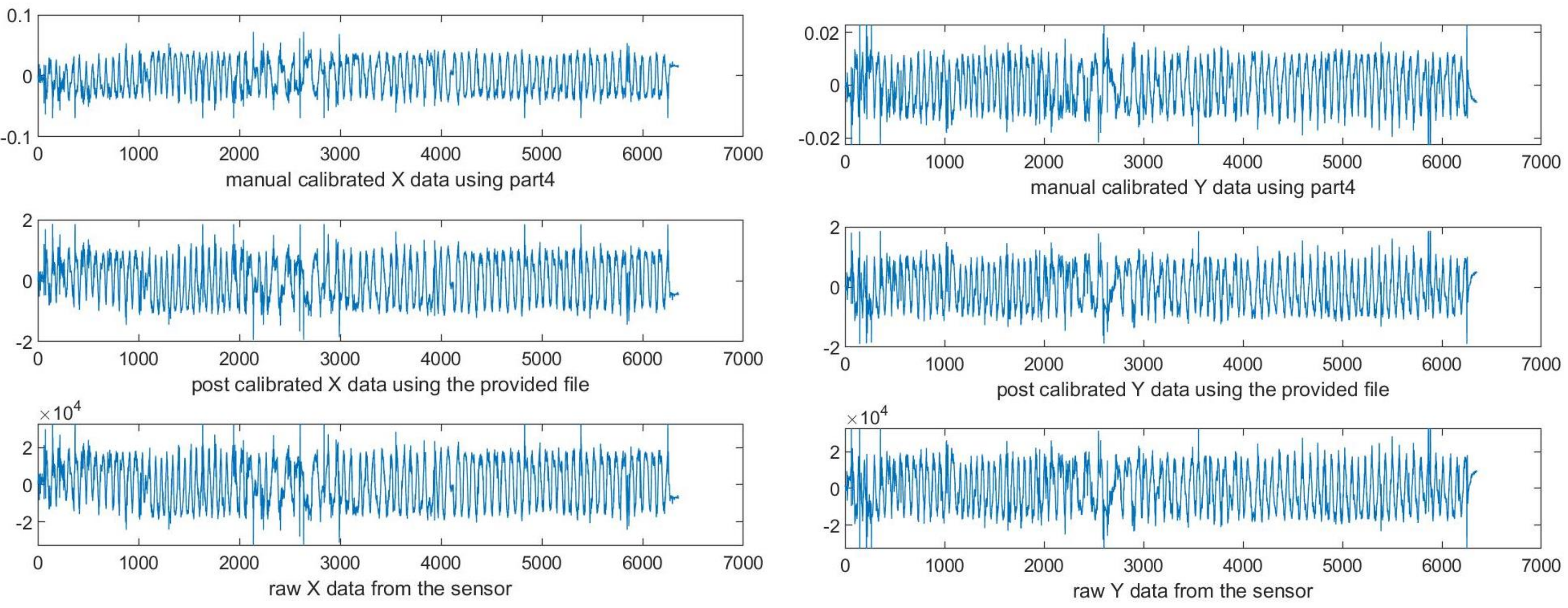


Figure 11: Comparison of X Y axis data calibrated with different method (accelerometer)

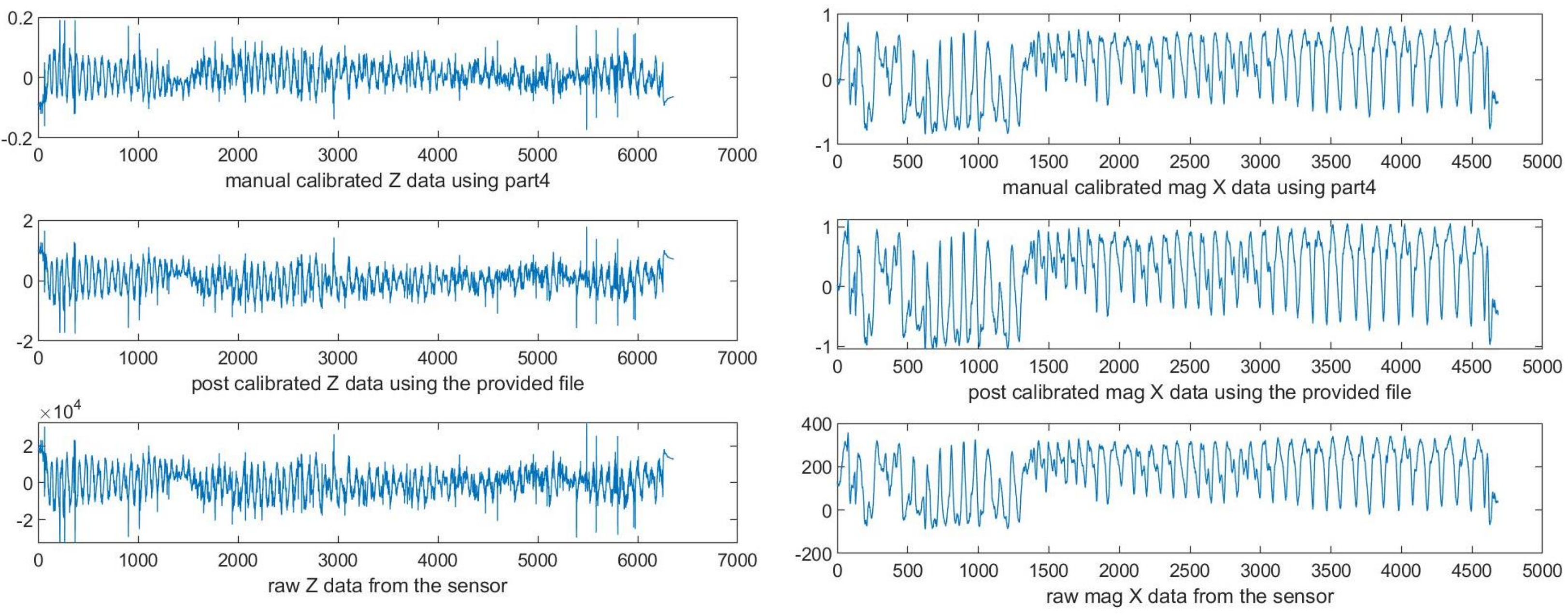


Figure 12: Comparison of Z X axis data calibrated with different method

Accelerometer(left), Magnetometer (right)

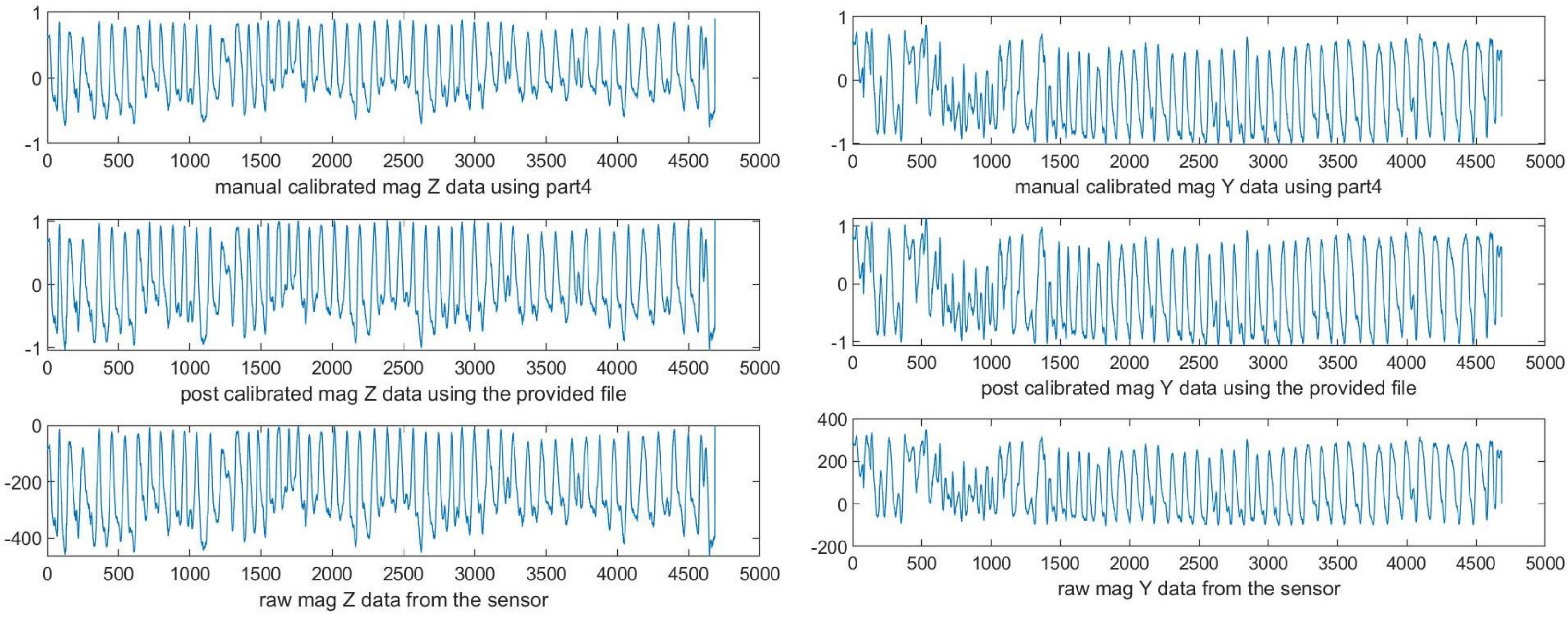


Figure 13: Comparison of Y Z axis data calibrated with different method

(Magnetometer)

Conclusion:

This lab is not as time consuming as the previous lab, but I did spend a lot of time on it. Mostly because I'm not very familiar with MATLAB and I spend some time trying to figure out how it works. Other than that, the lab itself is ok. I learn a lot on how the sensor calibration works, especially the 3D calibration part. It's also interesting to see how bad my sensor is since I have an almost 100-degree drift in an hour with the sensor laying flat on the table. I would still spend some extra time in this lab trying to get everything figured out before advancing to lab4 since they are related.