Lab 4

This lab is designed to implement sensor fusion and perform dead reckoning using the sensor fusion data and only IMU sensor readings. The lab is performed in two parts –

I. Calculate Yaw angle by only using IMU magnetometer readings and Gyroscope readings, then use a complementary filter to find a resultant best result.
II. Perform dead reckoning with the IMU data and sensor fusion yaw readings and compare it with GPS data for estimating the drift in calculations.

For driver, please refer to the files pushed by Will Tower from our Group, his Gitlab Username is *tower.w@northeastern.edu*

PART I

A. Magnetometer data calibration –

To calibrate the magnetometer data, we rotate our IMU sensor slowly, to perform this we took readings in a car and moved around in a circle. There are two main errors associated with the magnetometer readings which are dependent on external environment; rest might be internal biases which we cannot resolve with this experiment.

The two distortion causing factors are hard iron or soft iron. Hard iron distortions are created by objects that produce a magnetic field. A speaker or piece of magnetized iron for example will cause a hard iron distortion. If the piece of magnetic material is physically attached to the same reference frame as the sensor, then this type of hard iron distortion will cause a permanent bias in the sensor output. Soft iron distortions are considered deflections or alterations in the existing magnetic field. These distortions will stretch or distort the magnetic field depending upon which direction the field acts relative to the sensor. This type of distortion is commonly caused by metals such as nickel and iron. In most cases hard iron distortions will have a much larger contribution to the total uncorrected error than soft iron.

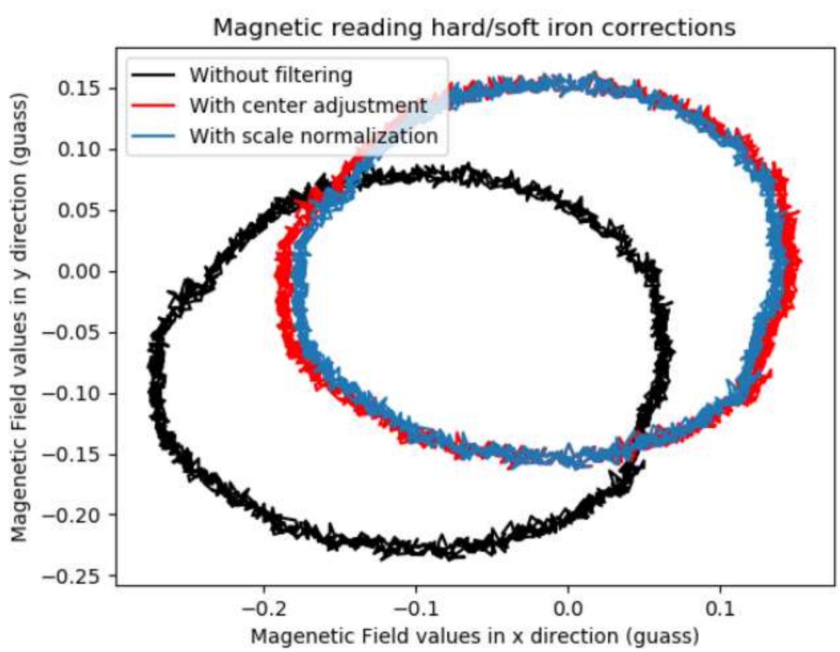


Fig .1. – Magnetic readings in X, Y direction.

From the *Fig. 1.*, we can observe that the hard iron distortions are dominating and there is less overall effect on our readings because of soft iron. To remove this error, the distortion free MagX and MagY plot looks like a circle with its center at (0,0). So, our aim is to find the coefficients of these transformations, so that we can use the same coefficients in the readings from the Boston mini tour. To bring our ellipse to origin, we'll calculate the center of our ellipse and shift it to the origin. To calculate the center, we find r/2 and q/2 from *Fig. 2.*, and add/subtract this value from the minimum of our ellipse readings in X and Y. Since, our shape is not tilted, we do not need to perform rotation of our existing ellipse, but we'll need to reshape the ellipse into a circle for which we define a reshaping factor given by minor axis length/ major axis length (q/r from *Fig. 2.*) of the ellipse. To correct our readings, we divide our Magx readings by reshaping factor.

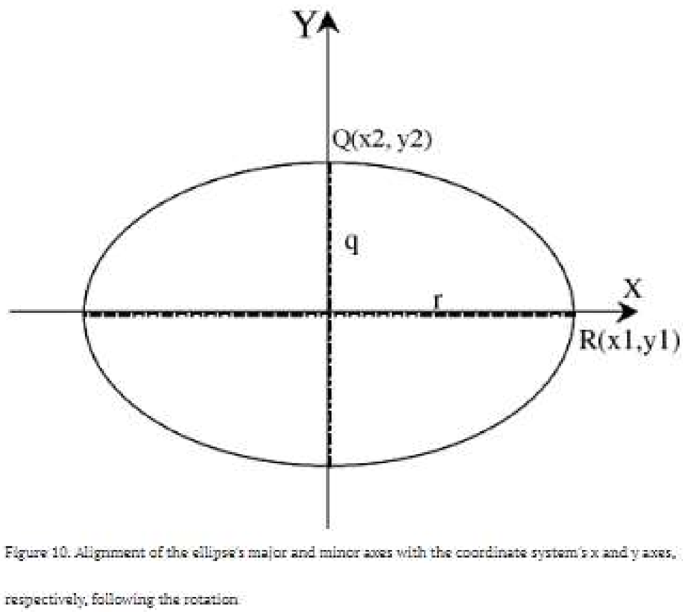


Fig. 2. – For calculating reshaping factor. Source.

After performing step 1 we get the ellipse (red) shown in *Fig. 1.*, and after performing step 2 we get the sort of circle (blue) shown in *Fig. 1.*

We use the values calculated for performing this normalization for our readings in mini-Boston tour.

We use these corrected values of magnetic flux in y direction and x direction to find the angle between them, which is the yaw angle.

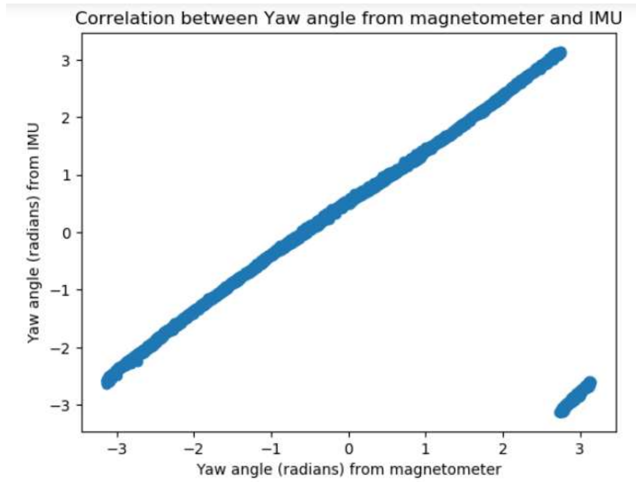


Fig. 3. – Comparison between corrected yaw angle using magnetometer readings and yaw angles from IMU.

From Fig. 3, we can conclude that the yaw angles calculated using magnetometer and the yaw angles got from IMU are directly proportional, which is expected.

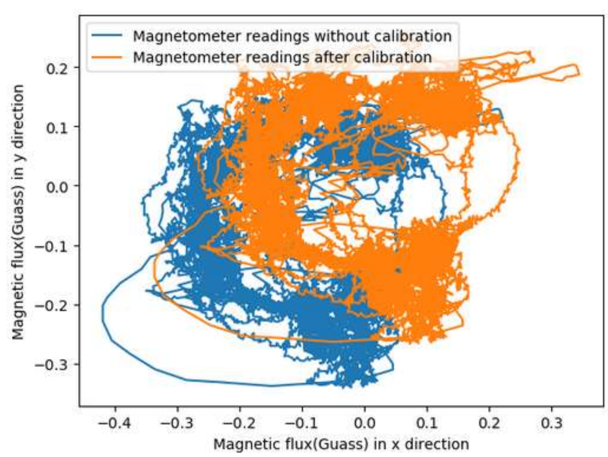
PART II

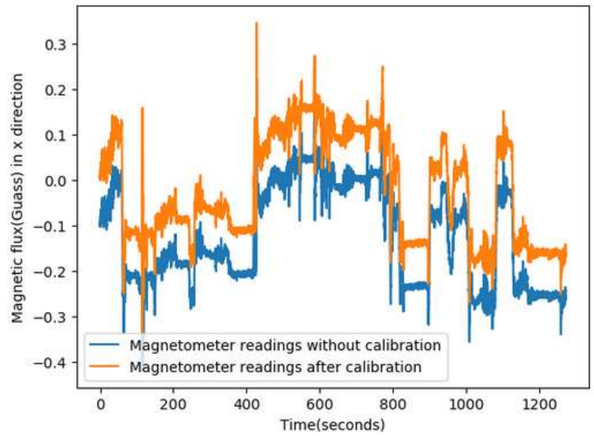
From here on analysis, calculations are done on the Boston mini tour data.

B. Complementary filter –

1. Calculation of yaw angle from magnetometer

To calculate yaw from magnetometer, we first remove the distortions in our readings using the coefficients from PART I. After correcting these readings, we calculate the yaw estimate from magnetometer by finding the angle between the magnetometer readings in y and magnetometer readings in x. After performing all the above stated steps, we get the calibrated magnetic flux in X and Y direction as shown below in Fig. 4.





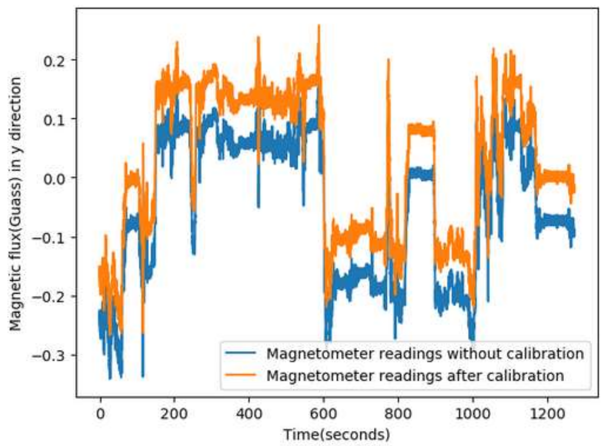


Fig. 4. – All plots after performing calibrations in the magnetic flux readings in the Boston mini tour bag file.

2. Calculation of yaw angle from gyroscope

We calculate the yaw angle from gyroscope by integrating the angular velocity in Z direction with respect to time recorded while collecting the data. Please refer to the plot below for comparison between magnetometer calculated Yaw and integral calculated Yaw from gyroscope.

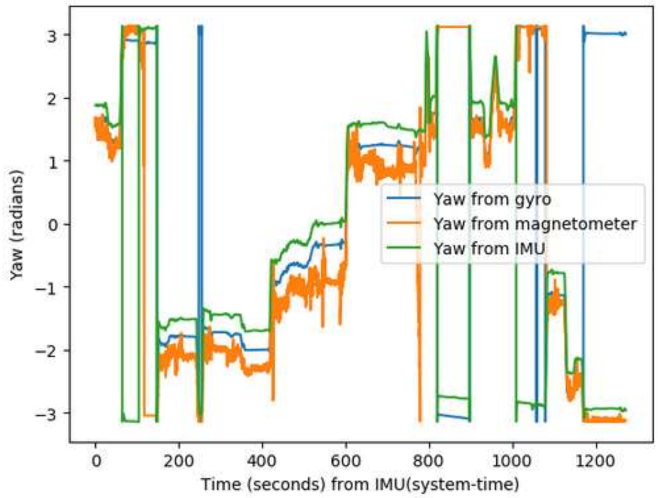


Fig. 5. – Yaw comparison between different readings.

3. *Complementary filter*

As from Fig. 5., we can see that the magnetometer readings have high fluctuations because of vibrations and for velocity calculated from integrating the angular velocity the bias/initial error gets integrated and increases slowly with time. Hence, the magnetometer is good for long term readings as it does not have a drift but has high frequency in fluctuations; and the gyroscope is good for short term readings but has a drift because of integration of biases. To resolve this, we use complementary filter where we pass the magnetometer calculated yaw readings through low pass (cutoff frequency – 0.003 Hz) and pass the gyroscope calculated yaw readings through high pass filter (cutoff frequency - 0.00001 Hz), we calculated this cutoff frequency by trial and error in such a way that it fits our unfiltered data without having too many fluctuations and without loss of information. After filtering we get the following graphs –

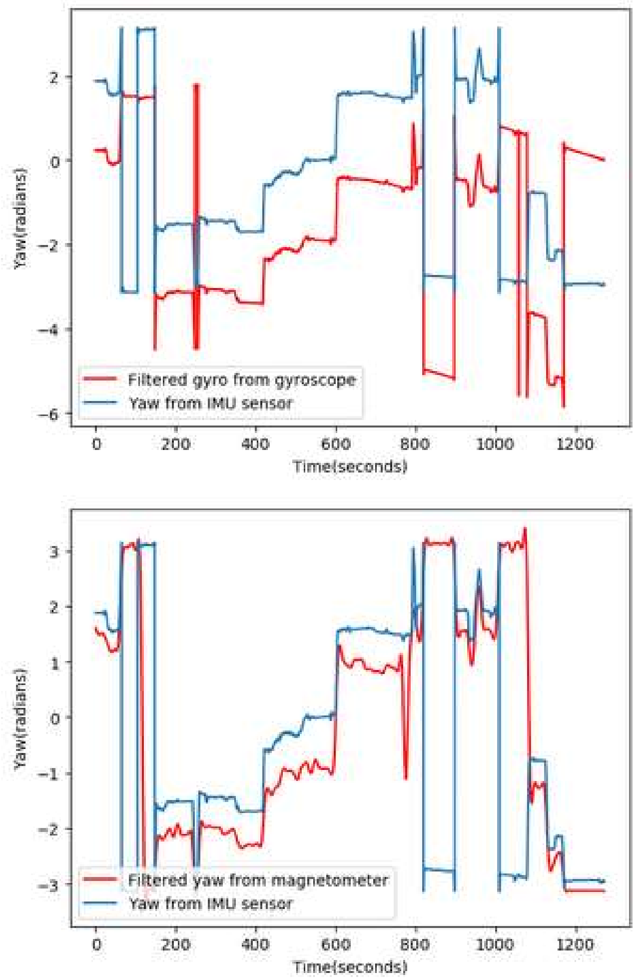


Fig. 6. – Yaw angles after filtering – 1st is a high pass filter for Gyroscope calculated yaw angles and 2nd is a low pass filter for Magnetometer calculated yaw angles.

For calculating a complementary filter, we merge the results by an equation –

```
Yaw_comp = YAW_gyro_filtered * alpha + YAW_mag_filtered[1:] * (1 - alpha)
```

As we did not calibrate the gyroscope which has lots of internal errors like random walk, bias instability, etc. While for magnetometer we at least corrected the errors caused by external sources, and for internal error the magnetometer is already calibrated to some extent by VN-100. Hence, we'll assign higher weight to the value of yaw calculated via gyroscope and the value of alpha taken is 0.95, which is done by trial and error for optimum fit.

The resultant complementary filtered Yaw is –

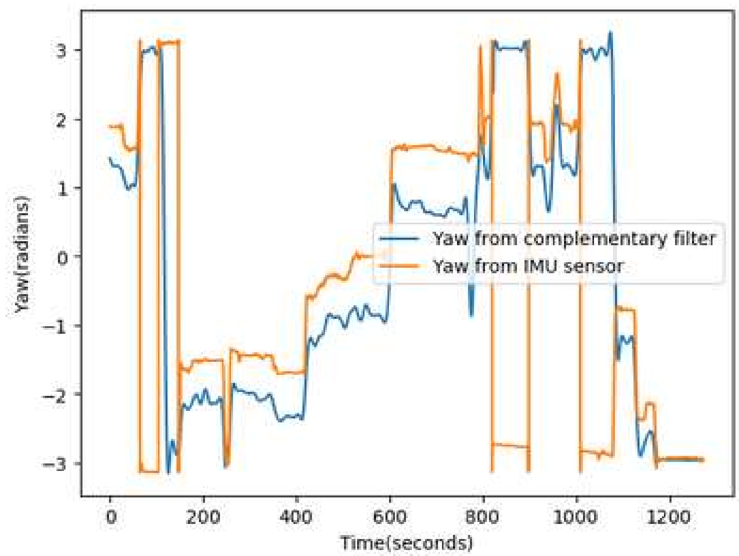


Fig. 7. Yaw readings from Complementary filter.

We can see how we implemented our complementary filter from the graph below –

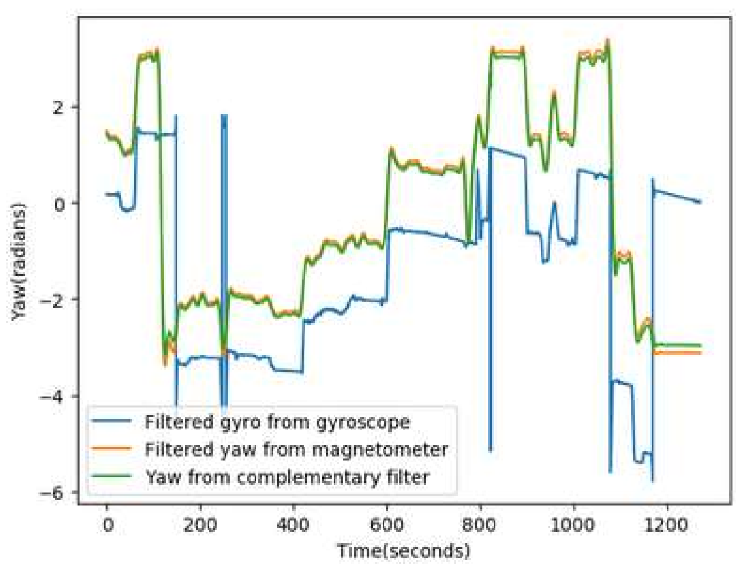


Fig. 8. Yaw readings with different filters

4. *Estimating Forward Velocity*
   For estimating the forward velocity, we integrate the forward acceleration (in x direction) (this will give a graph shown in Fig. 9.) of our IMU but as we start integrating this acceleration, we need to subtract the gravity component which forms the pitch angle with our forward acceleration. After integration we notice a drift in the acceleration which is because of integrated bias. This drift can be removed by fitting a line to this drift velocity curve and then bringing that line to x-axis by subtraction of each point of our velocity curve with the corresponding point of this straight line.

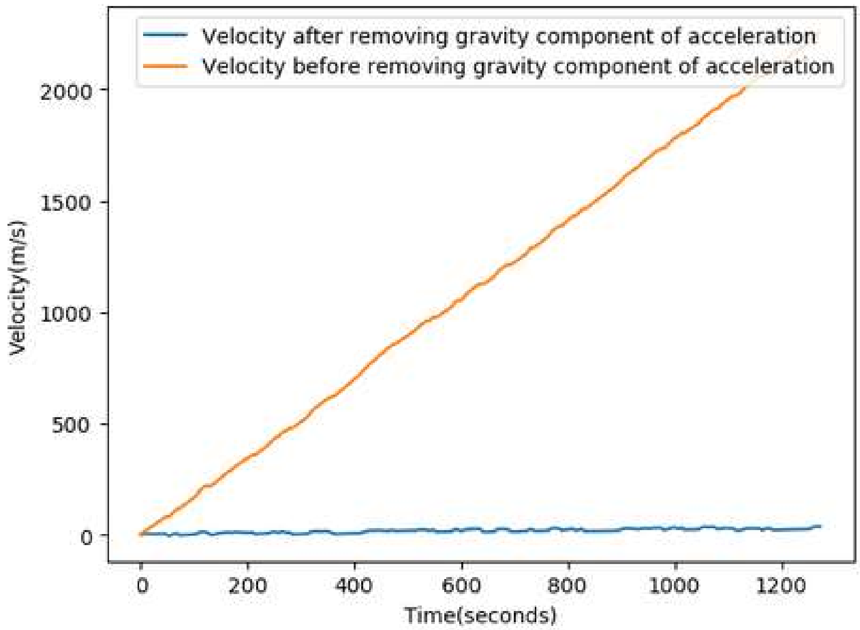


Fig. 9. Velocity before and after removing gravity component from heading acceleration.

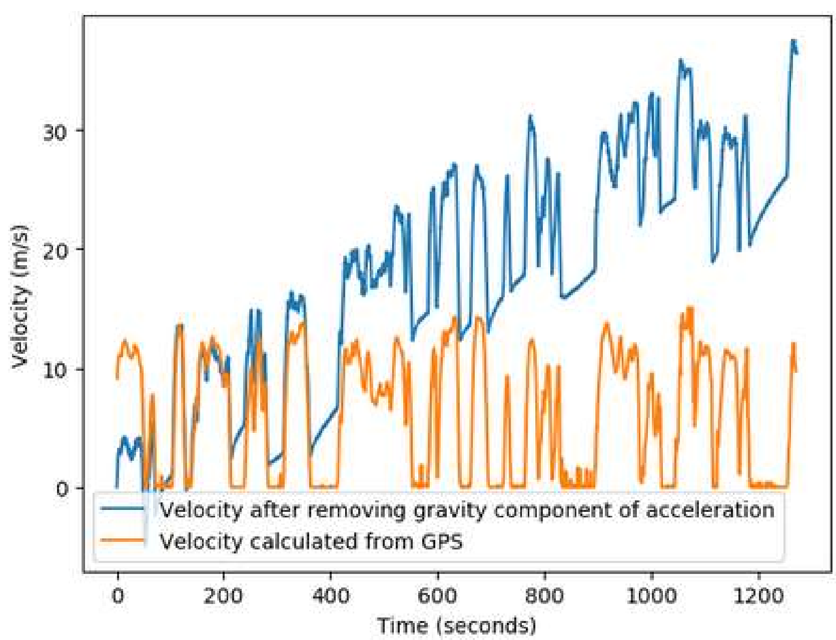


Fig. 10. Velocity comparison between biased velocity from accelerometer v/s GPS velocity.

After removing this bias as discussed earlier, we get –

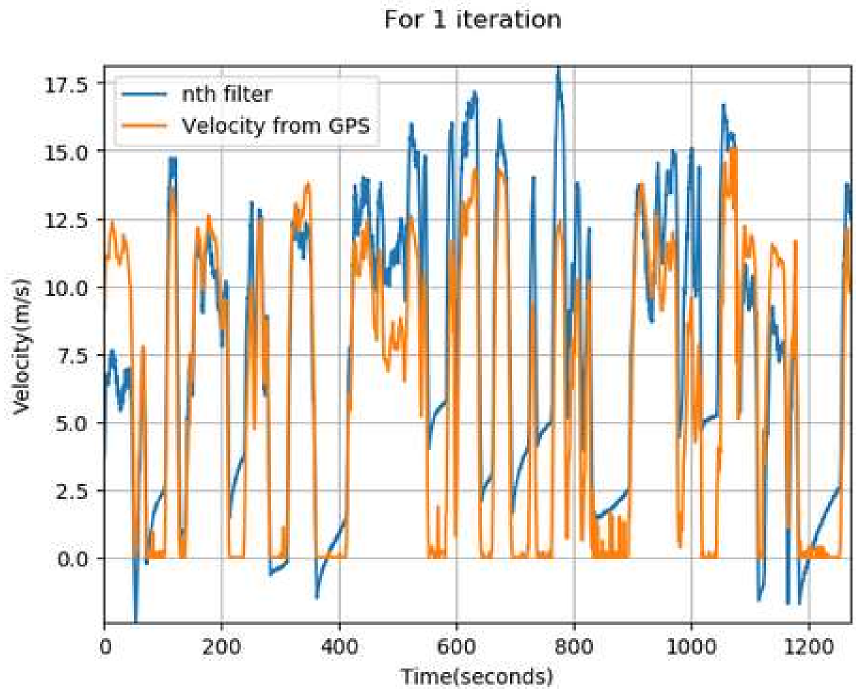


Fig. 11. Linear regression fit of IMU velocity

To further filter this, we observe that the bias has introduced slopes of constant angle in the graph. We also know that these slopes should be indeed zero, and hence we'll further filter with bringing these sections of velocity down on these slopes with the minimum value on the slope of this line. We further remove any negative velocity which can be seen in the graph as we did not take any reverse during our tour. Hence, the final filtered graph without using any knowledge from velocity of GPS is –

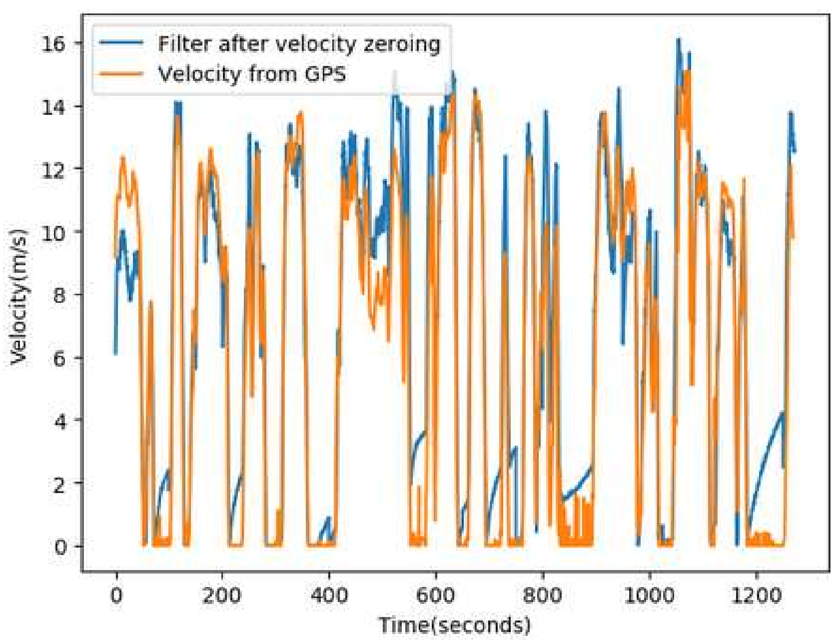


Fig. 12. Bin classified filtering of velocity

We can still see those slopes which should be zero, and hence our filter can be further adjusted. But this is a lengthy process, hence we take this velocity obtained as our reference for car velocity.

5. *Dead Reckoning –*

To calculate the distance covered we further integrate our velocity derived from IMU and to obtain the angular velocity we integrate the acceleration in y direction and multiply with angular velocity in z direction.

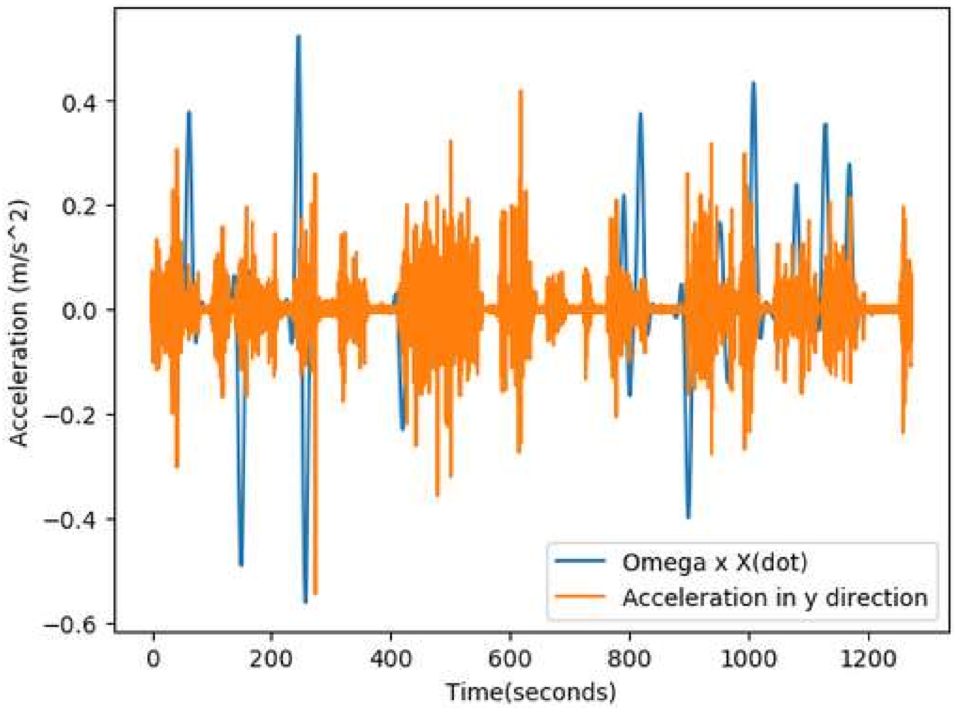


Fig. 13. Angular parameters w.r.t time.

The calculated acceleration is different from the is different from the IMU acceleration in y direction. This is because of the gravity influencing component which is present in the y direction. This can be mitigated by subtracting the gravity component from the calculated angular acceleration.

To find the path traversed by our IMU in east and north we treat our traversed path as a 2D map and have yaw angle as the angle between the heading direction and easting reading. We can later scale and rotate our IMU mapping to fit the GPS mapping. After doing this we get a map generated by IMU and we rescale and rotate this map, offset its initial reading to match the initial reading from our GPS. The scaling factors include, bringing the initial reading together, flipping, rotating, and then again brining the initial reading together. The values for these can be accessed through the Jupyter notebook file attached with the report.

```
The scaling factor in east :0.8325660586501613

The scaling factor in north :0.8418220080050847
```

The angle of rotation is -100 degrees about the initial reading and the flipping is done on this point only.

After performing these transformations, we get a mapping shown below –

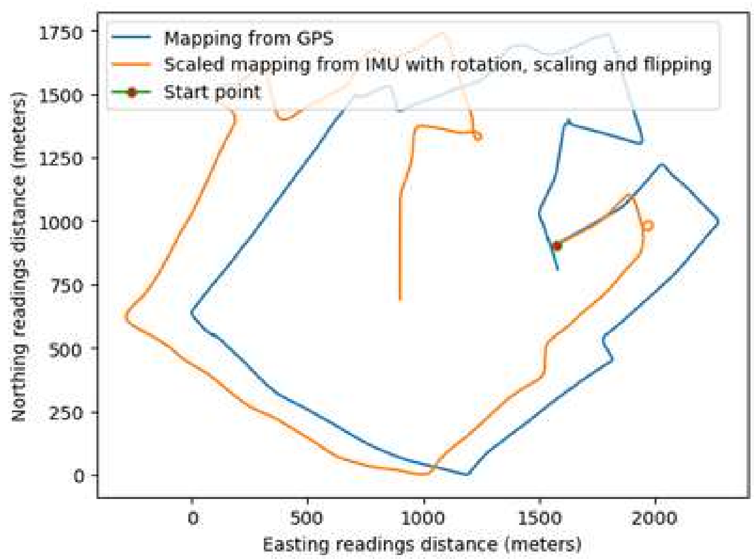


Fig. 14. Dead Reckoning

As we can see from the Fig. 14., we can infer that the readings are approximately accurate till ~ 350 meters. But the accuracy within range of 2m* is not there for even the next point of comparison. This is because of various reasons.

i. GPS puck and IMU were not at same position, so there is initial bias.
ii. We performed dead reckoning with yaw and velocity calculated from magnetometer and accelerometer which have internal errors and the integral of accelerometer introduces a drift error which further magnifies the error.
iii. The GPS puck itself has a steady state error which we have seen in lab1.

From VectorNav datasheet the magnetic heading accuracy is 2 degrees and gyroscope's noise density is 0.0035 deg/sec/root(Hz). With these accuracies the mapping should map better with the GPS reading. However, we are comparing the mapping performed with yaw and velocity which is integrated -> in Yaw the internal errors from magnetometer will introduce a deflection while the distance calculated has errors from accelerometer as well as bias errors introduced during integration. Hence, we had to scale down our readings. However, we have a same trend in both as observed from Fig. 14. which is a good estimate of mapping.

***All questions (except bonus) have been answered in their respective sections with relevant plots.**