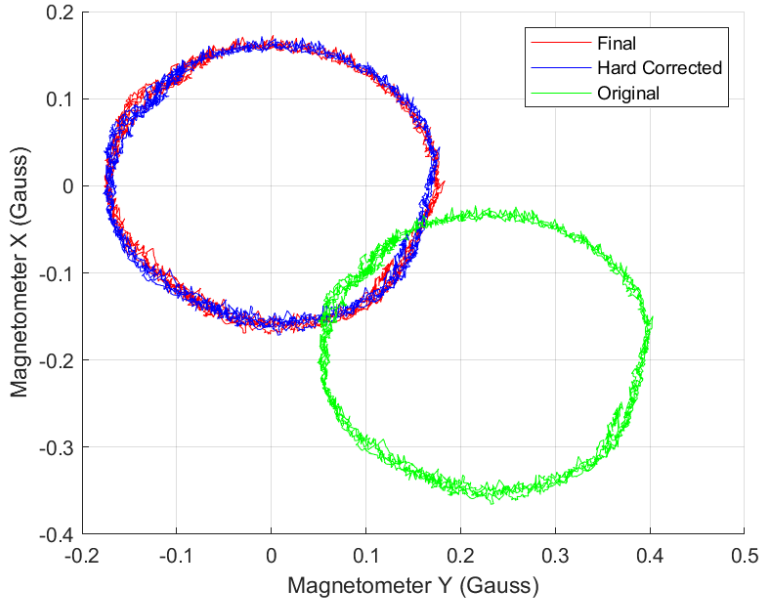
Lab 4

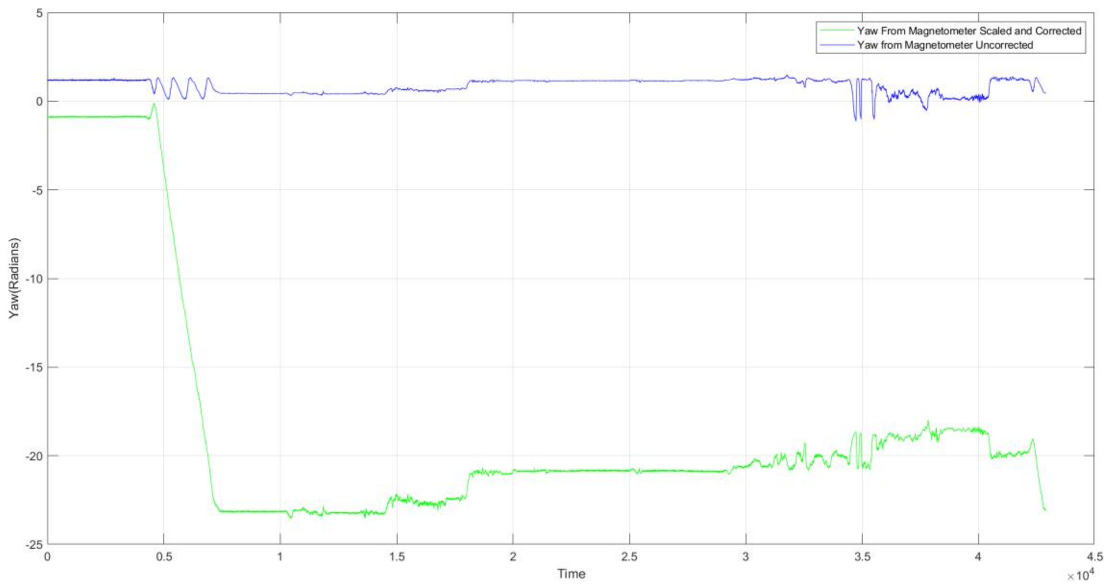
Navigation with IMU and Magnetometer

1. Magnetometer Calibration

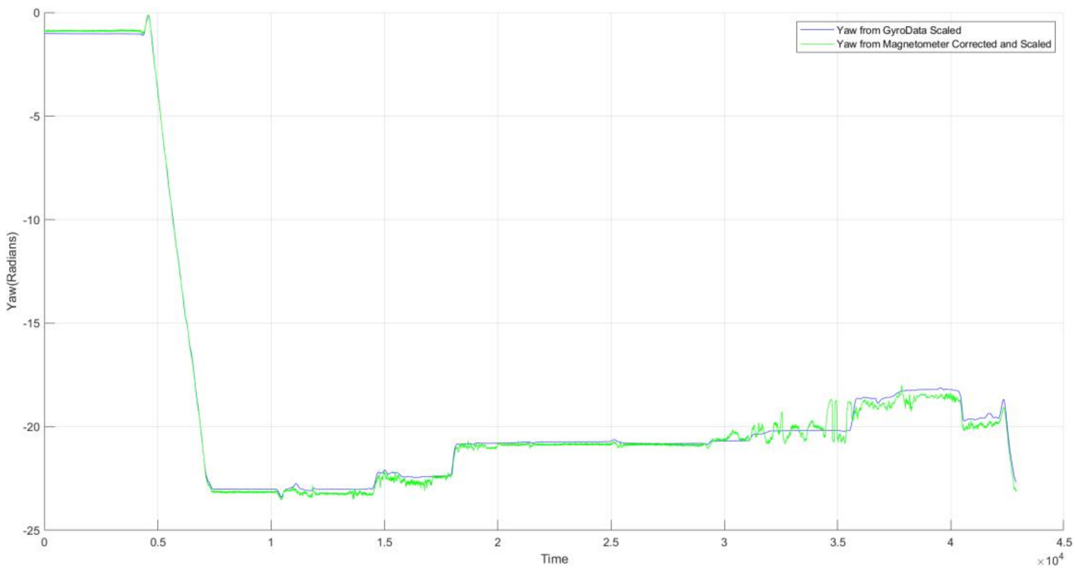


From the data it is observed that the hard iron corrections are very prevalent but the soft iron corrections are very minute. The Scaling Factor comes out to be 1.02 which implies that the ellipse is almost a circle. This corrected data can be used for further calculations and the data set has been calibrated.

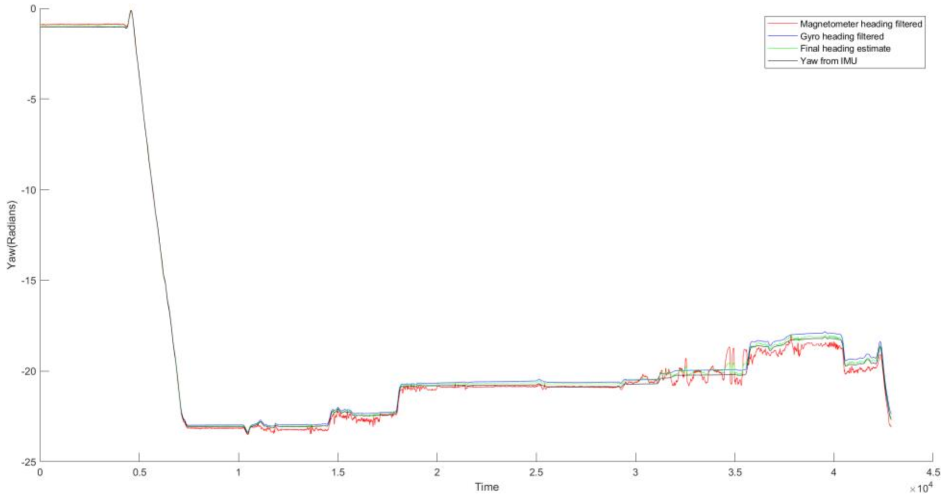
2. Yaw Calculations



From this graph we can see the drastic difference between the yaw calculations from the magnetometer corrected and uncorrected data. This validates the correction, calibration done on the data set

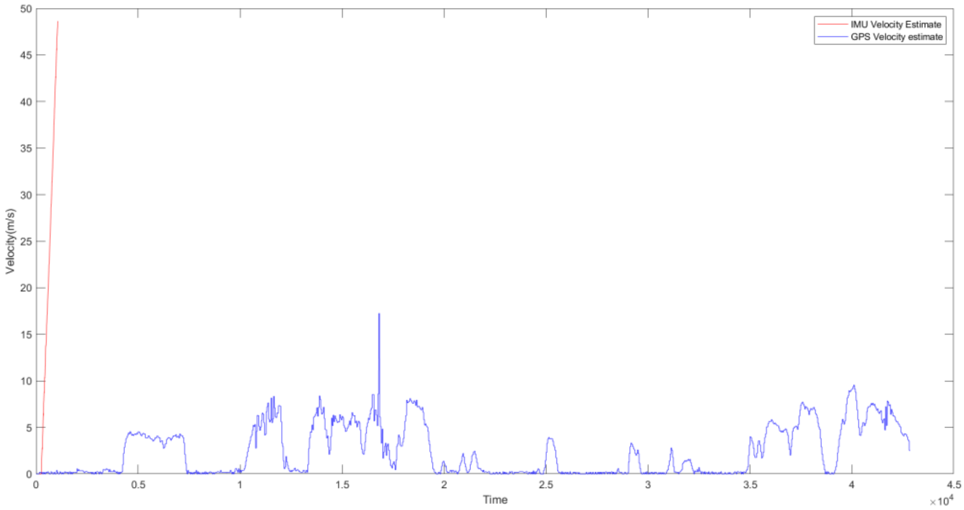


The Yaw/Heading calculated from the GyroData and the Magnetometer data are almost following a similar trend, the magnetometer data seems to have some noise as could be assigned to magnetic fluctuations due to the various power lines,electronics near the sensor and various conditions present that create a noise in the magnetic fields.



After applying the complimentary filter is observed that the Yaw from IMU is similar to the final filtered Yaw estimate for the Magnetometer filtered and Gyro Filtered data. There's a striking resemblance between the Yaw from Gyro and The Yaw from Imu implying that the sensor sends the yaw data doing a similar calculation. Due to combining the magnetometer yaw and gyro yaw we see that the final complimentary filtered data has some noise that is the remnant of the magnetometer noise and can be removed by further filtering.

3. Estimating the Forward Velocity



The IMU velocity overshoots as the initial error seems to be integrated and keeps on integrating the error wheres the GPS Velocity has the correct trend of the trajectory followed.

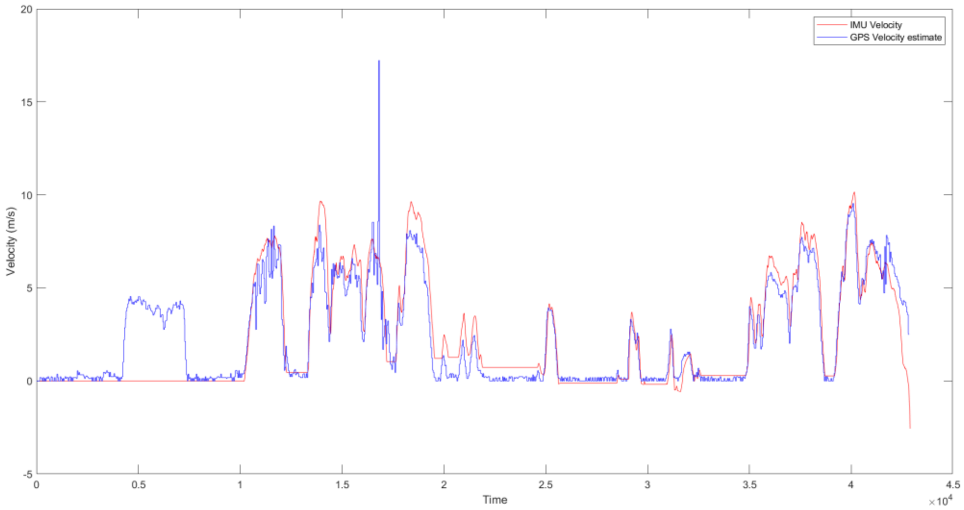
Adjusting the Acceleration Data

We know that the vehicle was stationary when we exited the calibration circles at Ruggles and hence we adjust the data for this bias and set the acceleration to 0 for this time.

We select a time window of 5 seconds and check the acceleration to be less than $0.5 m/s^2$.

This is considered as an acceleration of less than that could be considered to be 0 and is force set to be 0 in that range.

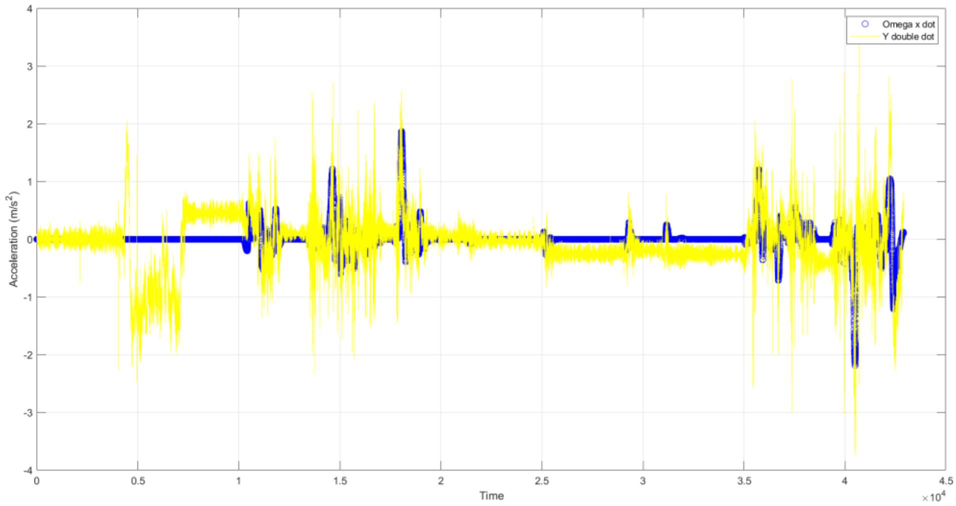
The negative velocity which can be integrated should be set to 0 as the car did not go in reverse.



The above graph shows the IMU velocity and the GPS velocity following the same trend and hence validates the corrections done to the acceleration data. Some bias is present and can be scaled to remove the bias but these results are satisfactory enough for velocity estimation.

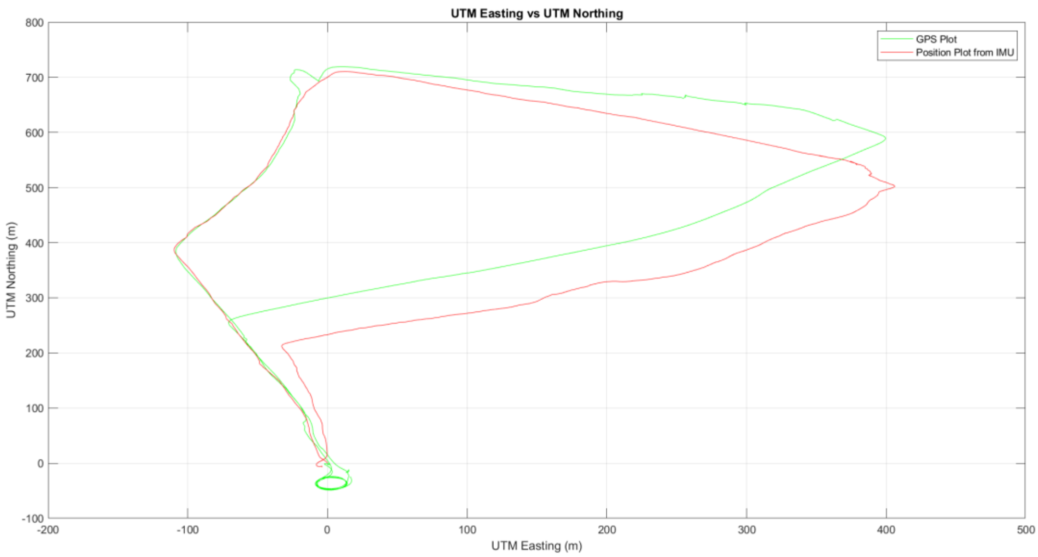
4. Dead Reckoning with IMU

a. Omega x Dot and Y Double Dot



We see that both the Omega X Dot and the Y Double Dot have a similar trend but the y double dot data is very noisy as we have considered the xc to be zero but the factual xc is at a different physical location an hence a bias can be seen. The noise can be attributed to the vibrations the sensor feels as well as some movements in the wire connected to the sensor and being disturbed by the car moving as well as by human error. Even though that's the case the avg accl comes to be about 0 which is correct and the y double dot follows the trend similar to omega x dot.

b. Compare IMU and GPS Trajectory.



From the above graph it can be seen that the displacement calculated from IMU data and GPS data is very similar in the beginning but as the integration increases the graph starts to create a bias after the second 90^{O} turn leading to an offset. A Rotationary matrix was applied to the data set to conform to the GPS data with a theta of 140 and the scaling factor of 0.8. The Magnetometer filtered data and imu filtered data gave us a final yaw which was used to estimate this position.

c. Estimating Xc

Using the equation in the Lab Document we assume y_doubledot to be 0 as the car is not skidding sideways so y(obs)_doubledot = omega*x_dot + omega_dor*x_c

And x_c = (y(obs)_doubledot – omeg*x_dot)/(omega_dot)

We ignore the values where omega_dor is zero as x_c tends to be infinite.

From matlab, the calculated mean of x_c comes to be -0.133114039396424 m ie 13cm

We collected the data in a personal car and the sensor was on the dashboard. A negative sign says that **the cars centre of mass is behind the sensor by 13cm .**