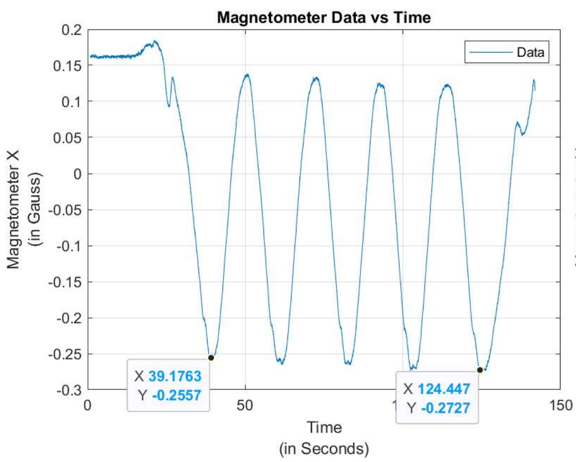
LAB 4: IMU AND GPS SENSOR FUSION

The following document aims to analysis the data collected from an IMU and a GPS sensor using custom drivers written in Python and ran using ROS.

Hard Iron and Soft Iron Corrections:



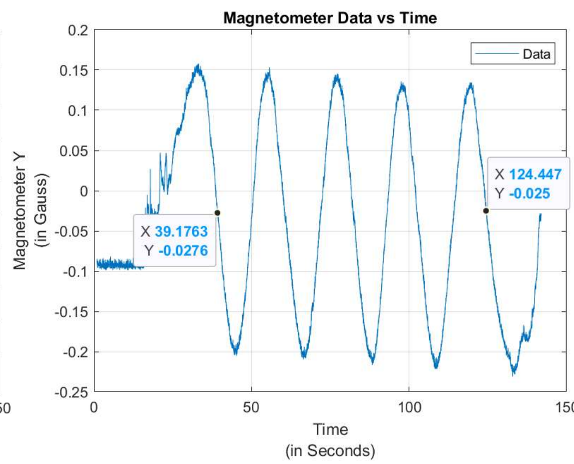
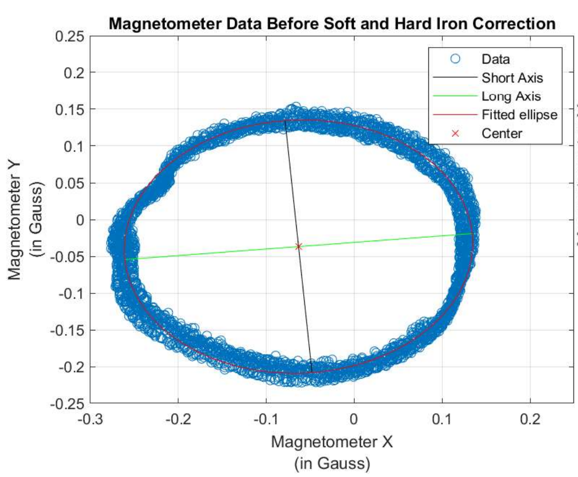


Figure 1. shows Magnetic field in X vs time and Figure 2. Shows the Magnetic field in Y vs time. These two graphs were used to splice the data collected while driving in a circle around the Ruggles circle for 5 rounds. The spliced data is for 4 rounds. While travelling in a circle the X and Y component of the magnetometer form a sine wave. I have considered the data between the time intervals shown in both figures for calibration. This was necessary because the driving data was started from outside the circular path and also stopped outside the circular path.



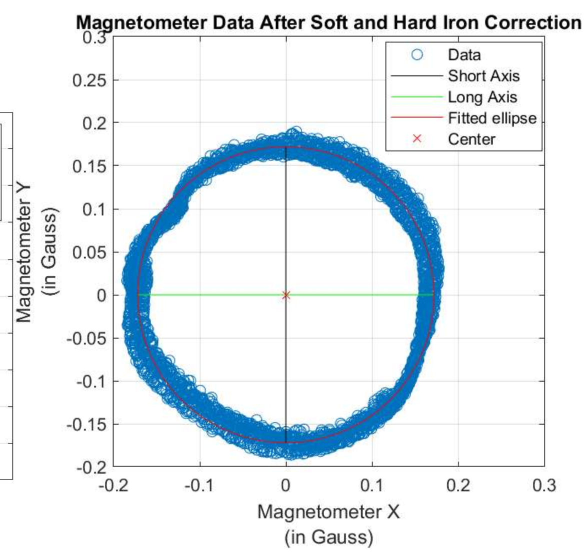


Figure 3. and Figure 4. show the Magnetometer Data before and after Hard and Soft Iron corrections respectively.

According to the datasheet of the IMU, as the skew error is under 10 degrees, we do not need to consider the z-axis data to correct the soft and hard iron data. The Hard Iron Errors can be corrected by the x and y co-ordinates of the centre of the skewed and displaced ellipse. These are used to subtract from the x and y of the magnetic field respectively.

(EllipseCenterX , EllipseCenterY) = (3.0358e-18 , -2.8460e-18)

After correcting the Hard Iron Errors, the ellipse's centre is shifted to the origin. The skew angle for this data is,

-0.0892 rad or -5.110 degrees

To correct the Magnetometer Soft Iron errors, first the ellipse is rotated along the Z-axis. The rotation matrix used is shown below

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} 0.9960 & -0.0891 & 0 \\ 0.0891 & 0.9960 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}$$

Next the Rotated ellipse needs to be scaled into a Circle. To do this, we multiply sigma with the X-coordinate of the data,

$$\text{sigma} = \frac{\text{short axis}}{\text{long axis}}$$

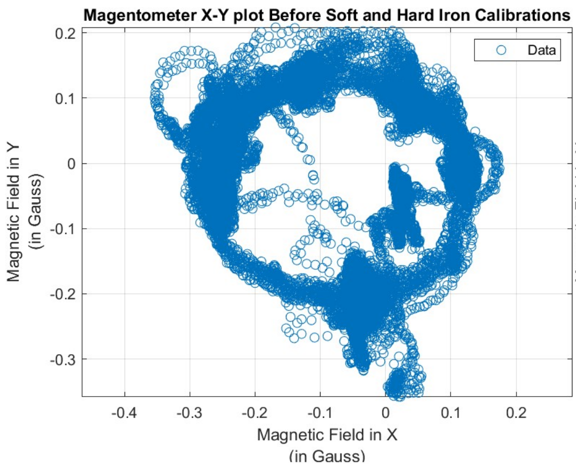
sigma = 0.865

The above-mentioned values are used for calibrating the data used in the analysis of the data recorded by the IMU while driving around Boston.

Q. How did you calibrate the magnetometer from the data you collected? What were the sources of distortion present, and how do you know?

- The magnetometer was calibrated using a rotation matrix as shown above. The sources of distortion are hard and soft iron errors. Hard iron errors are introduced due to the presence of permanent magnets near the IMU. Magnets present in speakers or any instruments inside the car can cause these errors. Soft Iron errors occur due to the presence of metal objects near the IMU that can induce magnetic fields temporarily.

Sensor Fusion:



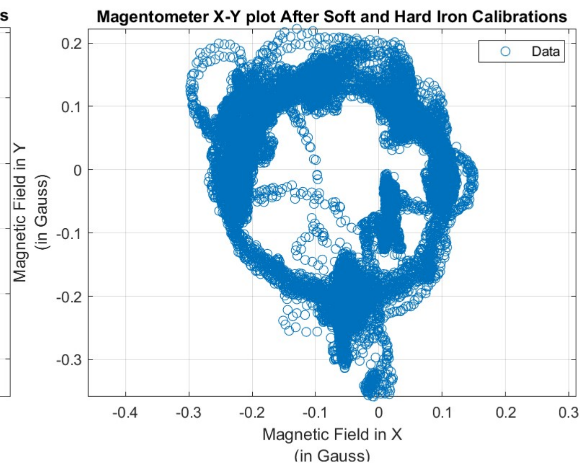


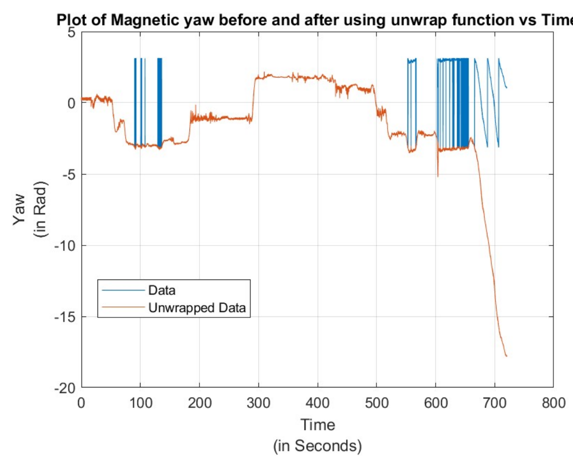
Figure 5 and Figure 6 show the plot of Magnetic Field in X vs Magnetic Field in Y before and after Soft and Hard Iron Calibrations respectively. In order to perform hard and soft iron corrections, the correction data from driving around in circles is used. While collecting the 'mini tour' data, the car was driven around the Ruggles circle two times before coming to a stop. This can be seen clearly in further plots.

In order to find the magnetic yaw, we use the accelerometer data along with the raw magnetometer data. We first calculate the pitch and roll for the accelerometer sensor using,

Pitch = atan2(accx, sqrt(accy^2 + accz^2))

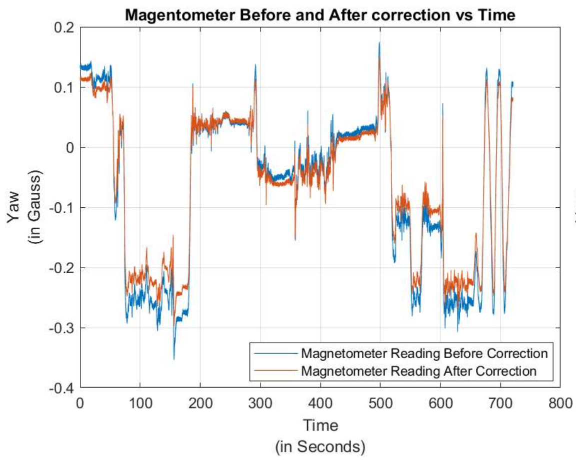
Roll = atan2(accy, sqrt(accx^2 +accz^2))

The Figure 7 shows the plot of Magnetic yaw before and after using the unwrap function. The yaw from magnetometer is calculated using the formula mentioned below.



Yaw = atan2((-ymag*cos(Roll) + zmag*sin(Roll)),(xmag*cos(Pitch) + ymag*sin(Pitch)*sin(Roll)+ zmag*sin(Pitch)*cos(Roll)))

We can see in figure 7 that towards the end of the plot, there are zig-zag lines similar to a sine wave. This indicates that the car was driven in circles as stated earlier. The 'unwrap' function adds all the values above pi and -pi to get a smooth graph. But as the car was driven in circles, we can see a sharp drop in unwrapped data as those rotations have been added. Figure 8 shows the magnetometer data before and after removing the soft and hard iron corrections.



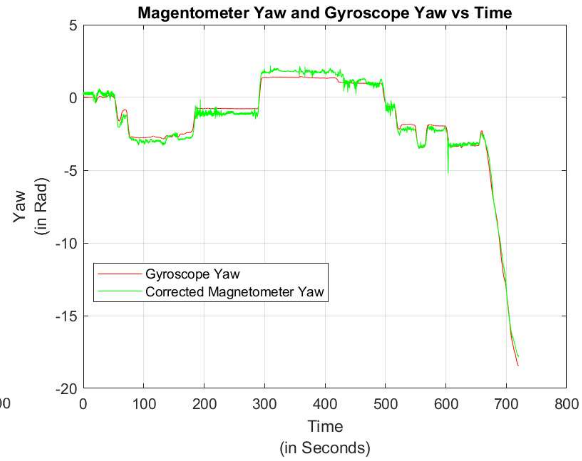
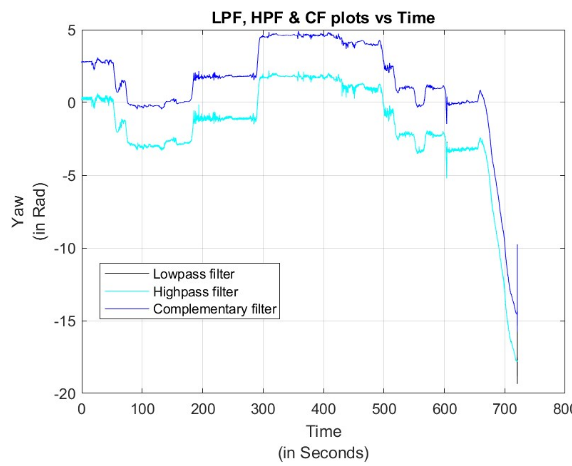


Figure 9 shows the plot of corrected Magnetometer Yaw and Yaw calculated from the gyroscope vs Time. We can see that the calculated magnetometer yaw matches the gyroscope yaw data.

Figure 10 shows the Magnetometer Yaw and Corrected Magnetometer Yaw vs Time. As the data collected did not have a lot of hard and soft iron errors, the plots almost overlap each other.

Figure 11 shows the plot of the output of the Magnetometer passed through the Low pass filter, Gyroscope yaw passed through the High pass filter and the resultant complementary filter.



Arguments for Filters			
Argument	**LPF**	**HPF**	**CF**
Passband Frequency fpass (Hz)	5	0.000001	0.78
sample rate Fs (Hz)	40	40	40

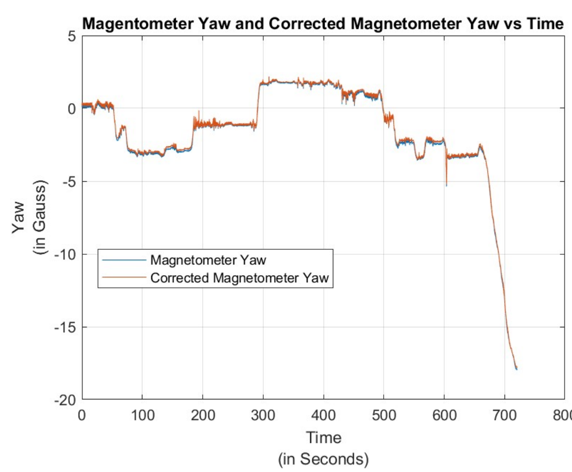
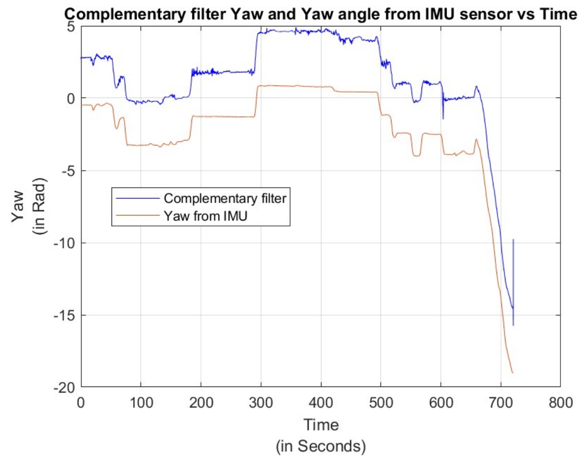


Figure 12 shows the Complementary filter and Yaw angle from IMU sensor vs Time. Here is can be seen that the data calculated for the magnetometer yaw follows the same path as that of the data collected by the Yaw of the IMU orientation. There is some offset in this data.



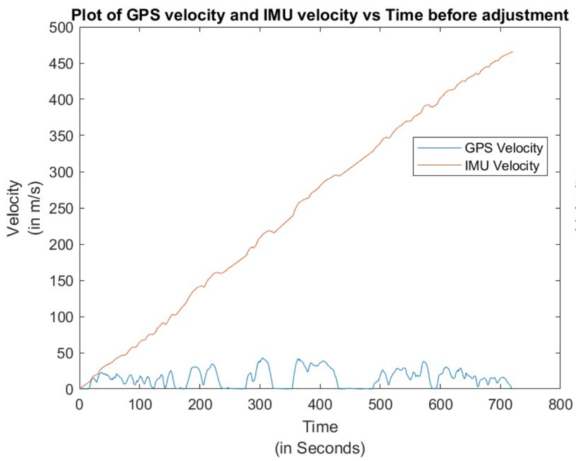
Q. How did you use a complementary filter to develop a combined estimate of yaw? What components of the filter were present, and what cutoff frequency(ies) did you use?

- To use the complementary filter on the IMU data I first used a low pass filter on the magnetic field to filter out the high frequencies in the data. Then I used the High Pass filter on the data from the gyroscope. Magnetic Field is a good indicator of orientation in static conditions while Gyroscope gives a good indicator of tilt in dynamic conditions. The complimentary filter is the combination of both low pass and high pass filter. The cut-off frequency for Low pass filter used was 5Hz and the cut-off frequency for high pass filter used was 0.00001 Hz and the sampling rate for both was 40. The value of alpha is very close to 1 hence, the complementary filter mostly consists of the Low pass filter data.

Q. Which estimate or estimates for yaw would you trust for navigation? Why? (Your answer must not be the Yaw computed by the IMU)

- The estimate of yaw from the Complementary filter is very close to the actual yaw measured by the IMU. This can be quite reliable for navigation.

Estimate of Forward Velocity



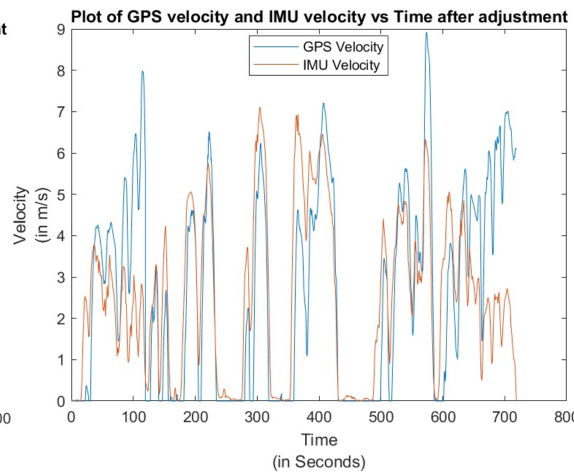


Figure 13 and Figure 14 shows the plot of GPS velocity and velocity calculated from IMU data vs time before and after adjustment respectively.

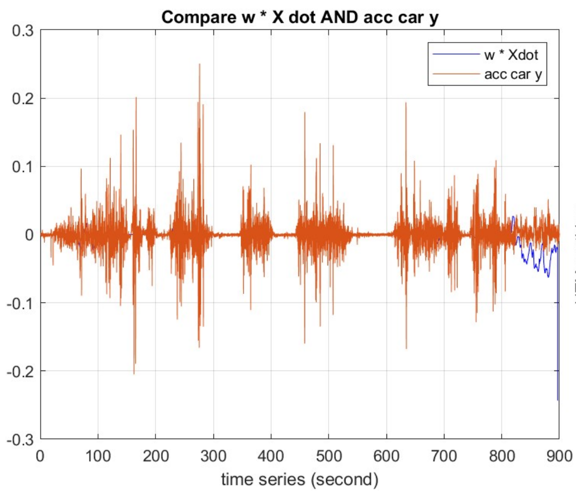
Q. What adjustments did you make to the forward velocity estimate, and why?

- First I calculated the jerk of in the data by finding the derivative against time for the IMU velocity. I then identified different sections in the data using loops. Next, I used this data to scale down the stationary parts to zero or close to zero, using thresholds. After this, I had used the best fit plot of the different sections to scale down the 'in-motion' parts of the velocity graph. Lastly the plot with GPS was plotted as shown in figure 14.

Q. What discrepancies are present in the velocity estimate between accel and GPS. Why?

- There are a few discrepancies present in the velocity calculated from the IMU data. This could be due to the integration adding up all the errors.

Dead Reckoning



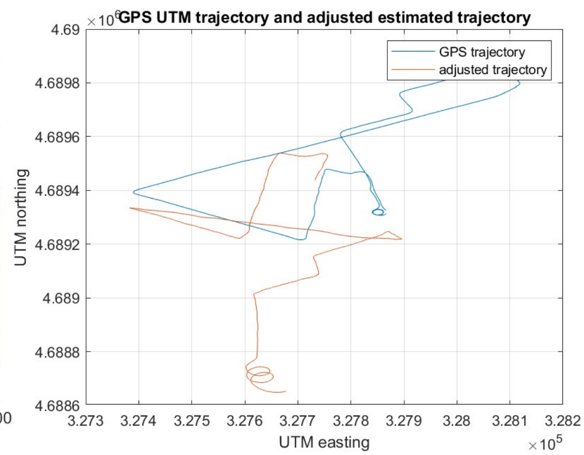
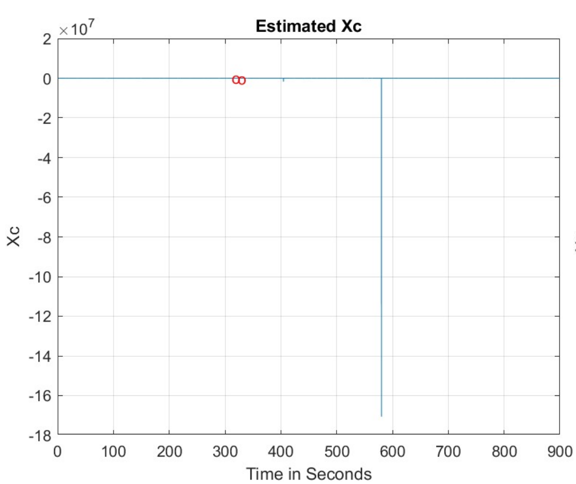
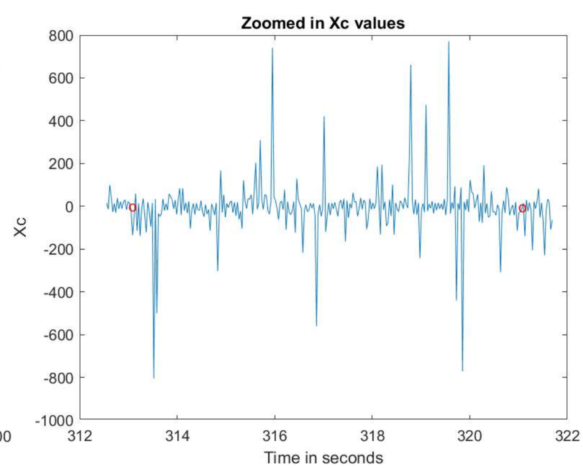


Figure 15 show the comparison between w multiplied with X and acc in Y. Figure 16 shows the final comparison between the GPS trajectory and the trajectory calculated from the IMU.

Q. Compute ωX and compare it to $\ddot{y}_{obs}$. How well do they agree? If there is a difference, what is it due to?

- The figure 15 shows the computed wX which is compared with Yobs. The Yobs overlaps the wX almost throughout the data.





Q. Estimate xc and explain your calculations.

- The estimated distance between inertial sensor and the CM point is shown in figure 17. To better estimated the x_c value, I calculate the mean value of x_c at time period 312 second to 323 second (shown as two red circle) and zoomed in shown in Figure 18 that the car is moving at a nearly constant velocity during this period. The mean was calculated for this period and it came to be **5.5cm**. I used this formula to estimate the Xc.

Xc = (Acc in X – Acc drift – Acc calibrated – (gyro yaw * IMU linear acc in X) / w_dot + w.*w