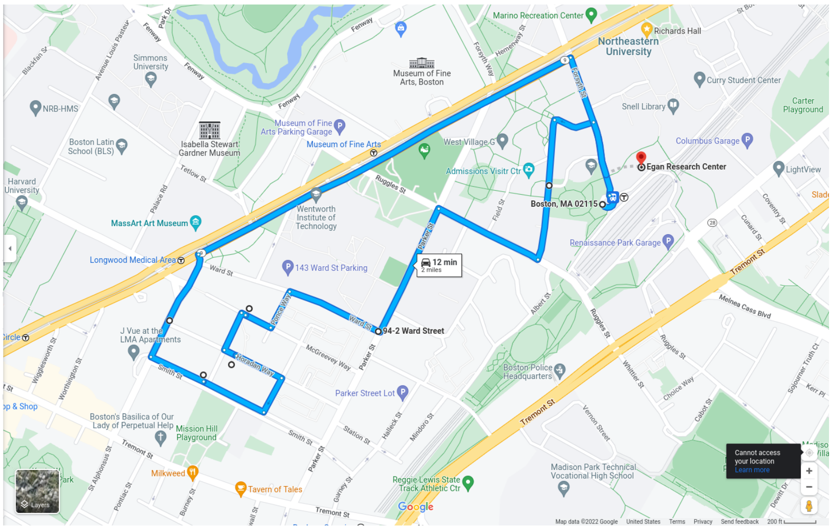
Lab Goal:

Analyze the data collected by VectorNAV-100 IMU and GPS, and use this data to build a navigation stack. Further, understand the relative strengths and drawbacks of each sensor.

Methods:

Our team collected the moving data on Boston Streets while driving in a NUANCE car. The IMU Sensor was mounted on the floor between the front and back seats of the car, and the GPS sensor was mounted on the Roof. The path for navigation is shown in the image below - the path was chosen to have more than ten turns and minimum traffic at that time.



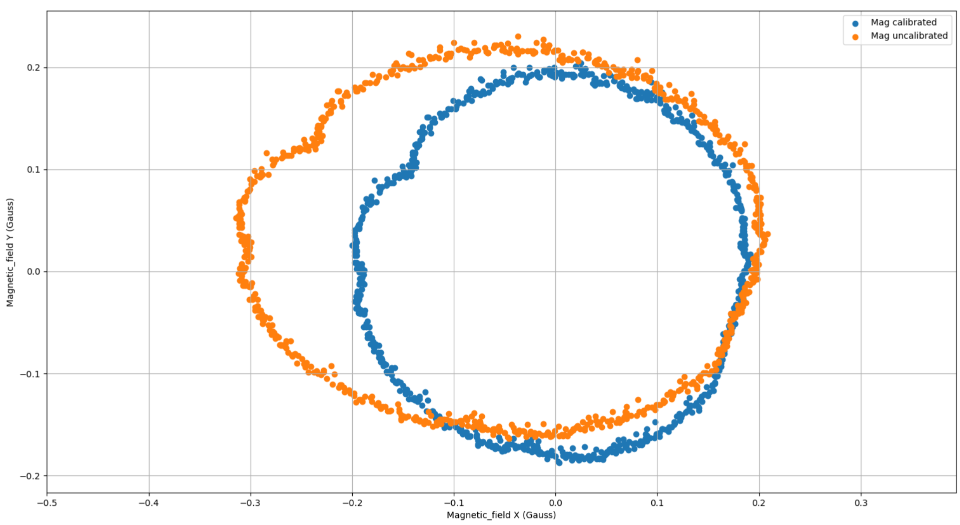
Data Analysis:

We calculated two data sets - Driving in the circle for removing soft and hard iron Bias, Navigating on a predefined path for testing of Sensor Fusion framework.

MAGNETOMETER - Calibration

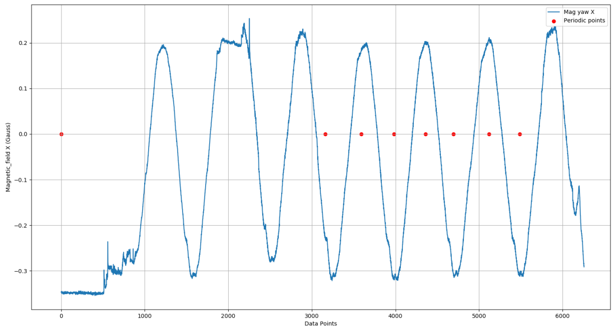
We moved on Ruggles circle eight times to calibrate the magnetometer - Estimating soft and hard iron bias, which were further used in correcting Magnetometer values in the Navigation dataset.

Plot 1: It shows the Mag X vs. Mag Y plot depicting the trajectory of the path taken for calibration - Circle. Also, the plotted magnetometer data before and after Hard-Iron and Soft-Iron calibration.



(Image: Magnetic Field for Calibrated vs. Uncalibrated magnetometer when moving in a circle)

The calibration dataset had some systematic errors due to moving traffic, so we first identified the index in the dataset free of these systematic errors by looking at the time series plot of MagX and getting indexes with slope = 0 after the first three complete rotations. Then, the data points between correct rotation indexes are used to estimate the calibration values and averaged to get the final calibration values.

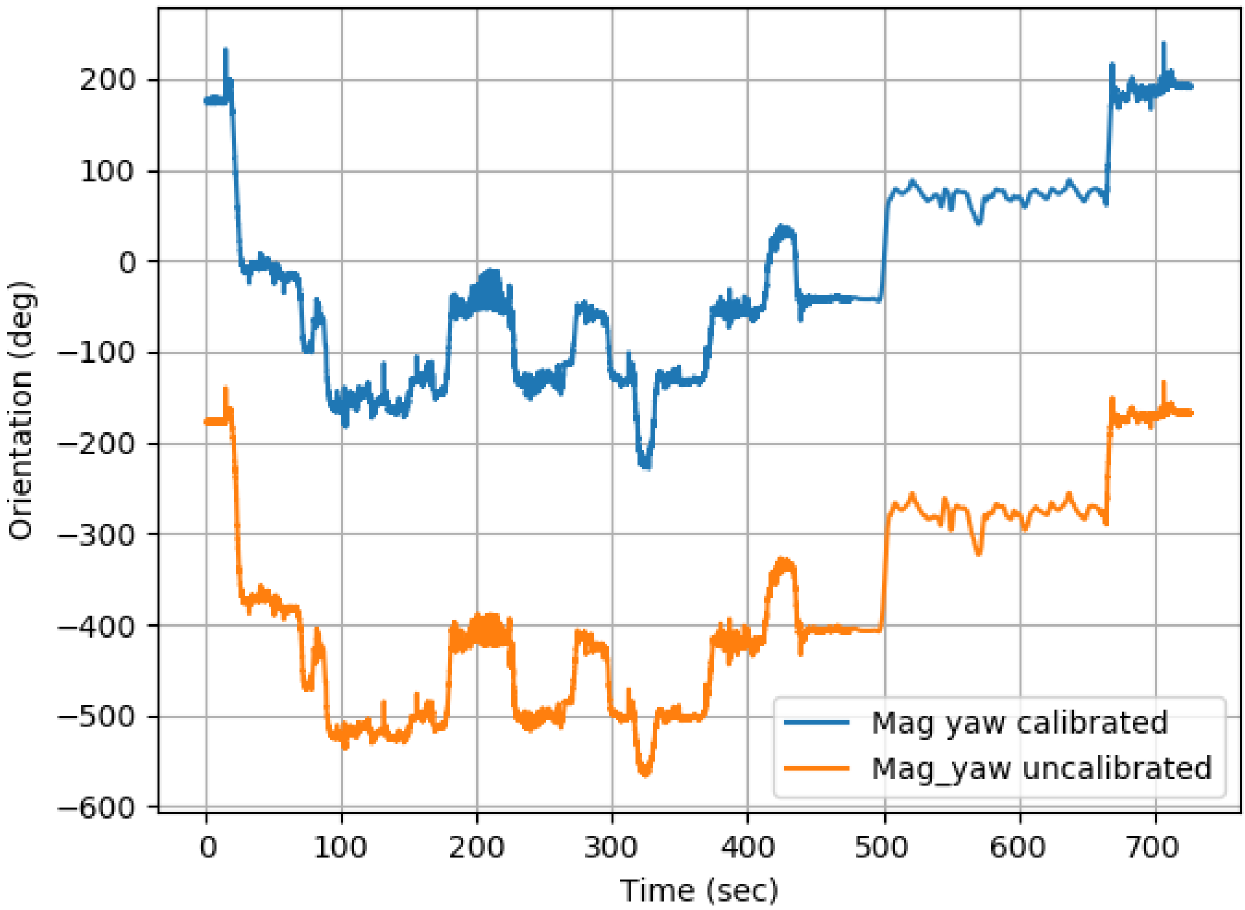


(Image: Alternative red dots signify the beginning and end of one rotation - In our analysis, we used data pts from 2nd-4th Red Dot and 4th-6th Red Dot)

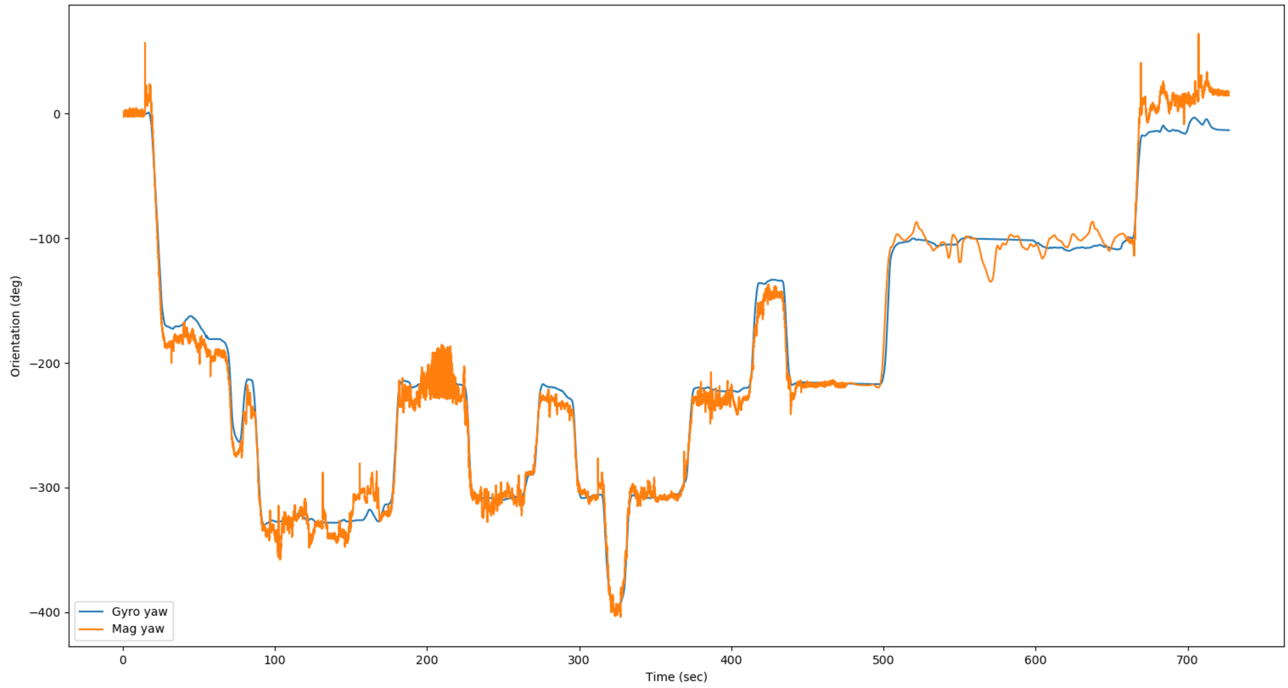
NAVIGATION -

We used navigation data from IMU - Gyroscope, and Magnetometer to get the heading after Sensor Fusion. Further, the heading data generated from Sensor fusion is used with accelerometer data to estimate the displacement curve. Finally, the trajectory data from IMU data is compared with GPS data.

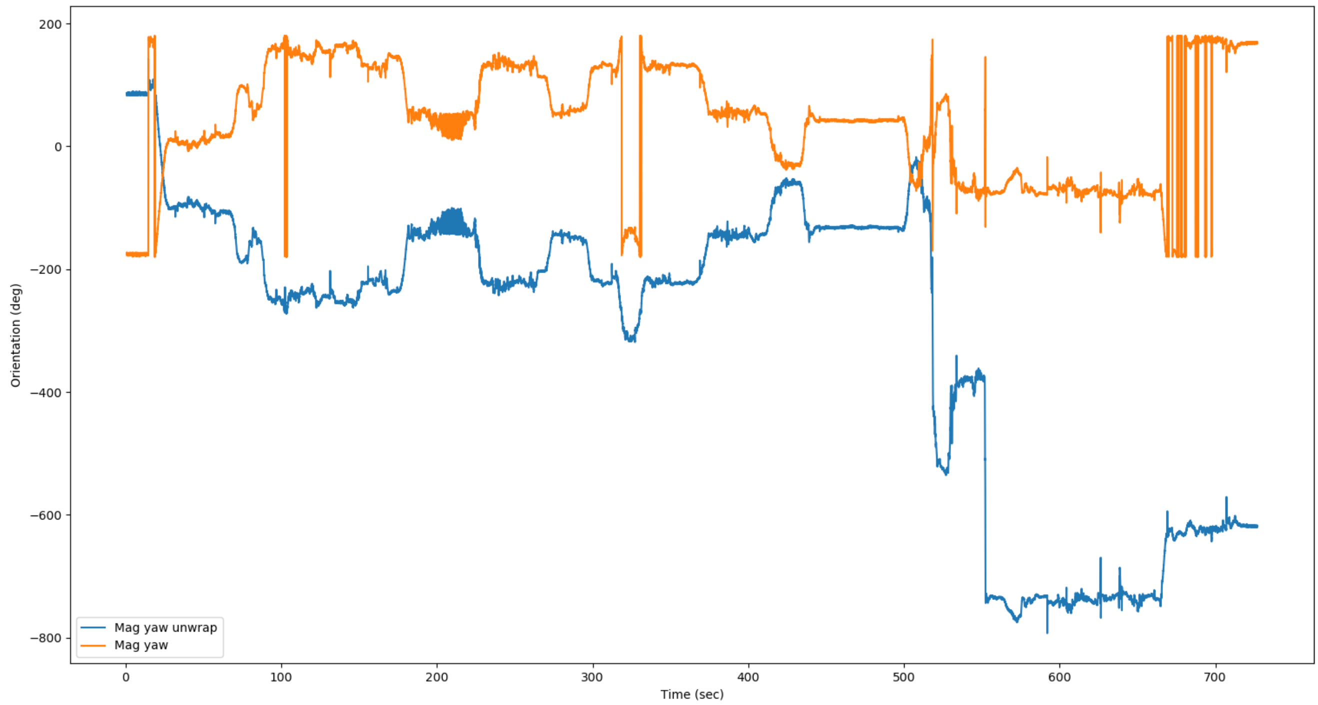
Plot 2: Calibration values from the previous dataset are used to remove hard and soft iron bias from Magnetometer data. This time series plot for magnetometer readings before and after calibration depicts the impact of removing these biases.



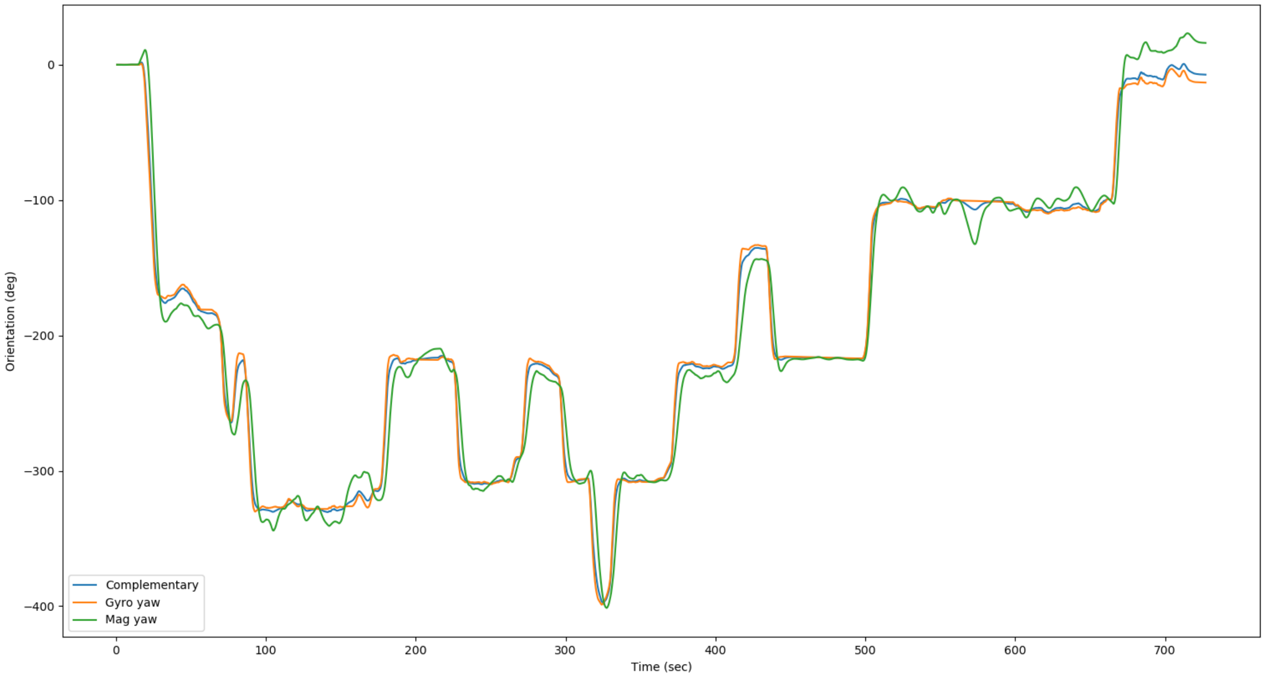
Plot 3: Angular Velocity (z-axis) from Gyroscope is integrated to get yaw or heading, and yaw is calculated from the Magnetometer data using atan2(Mag Y, Mag X). These values are compared with each other to understand the noise and error characteristics of the two and further decide on tuning parameters in Sensor Fusion.



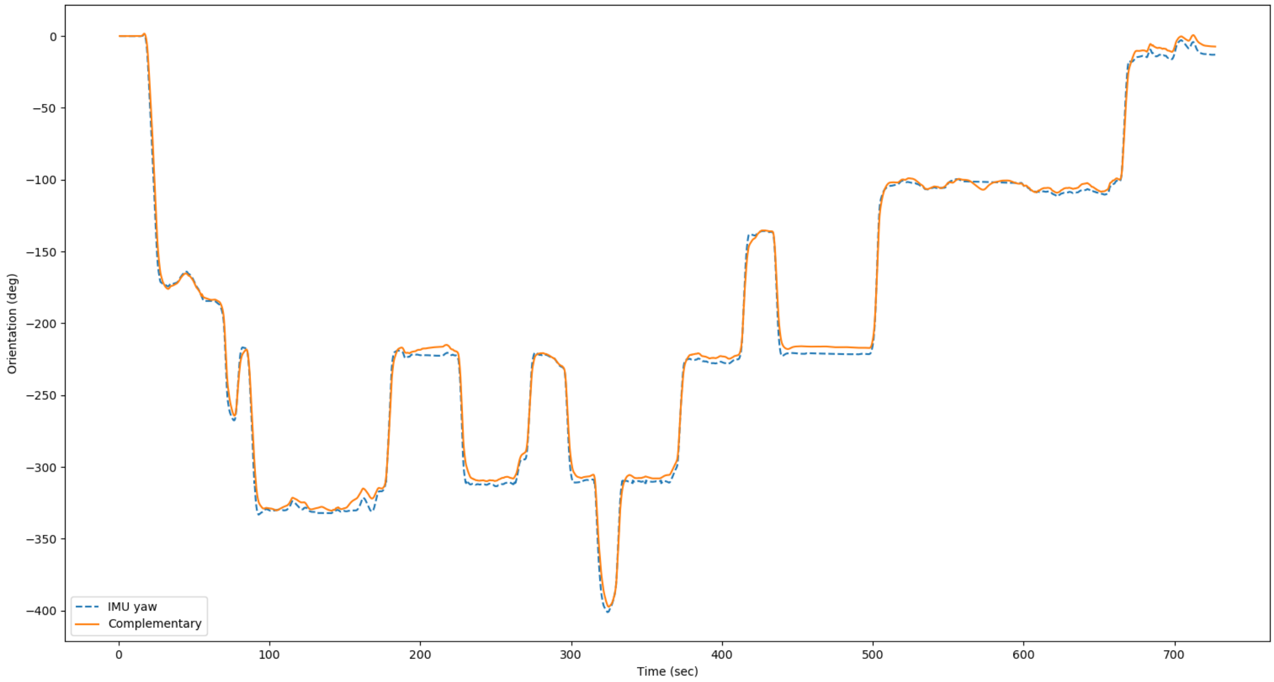
Correction logic: For calculating yaw from Magnetometer in timestamp between 500-700 sec, I have used a moving average filter before unwrapping data to remove significant errors in the unwrapping. A possible source for this error is the large spikes in the raw mag-yaw data, which were caused by high-current in the power lines on the road while 2-trams passed by near us in this period. These errors are reflected in the plot below.



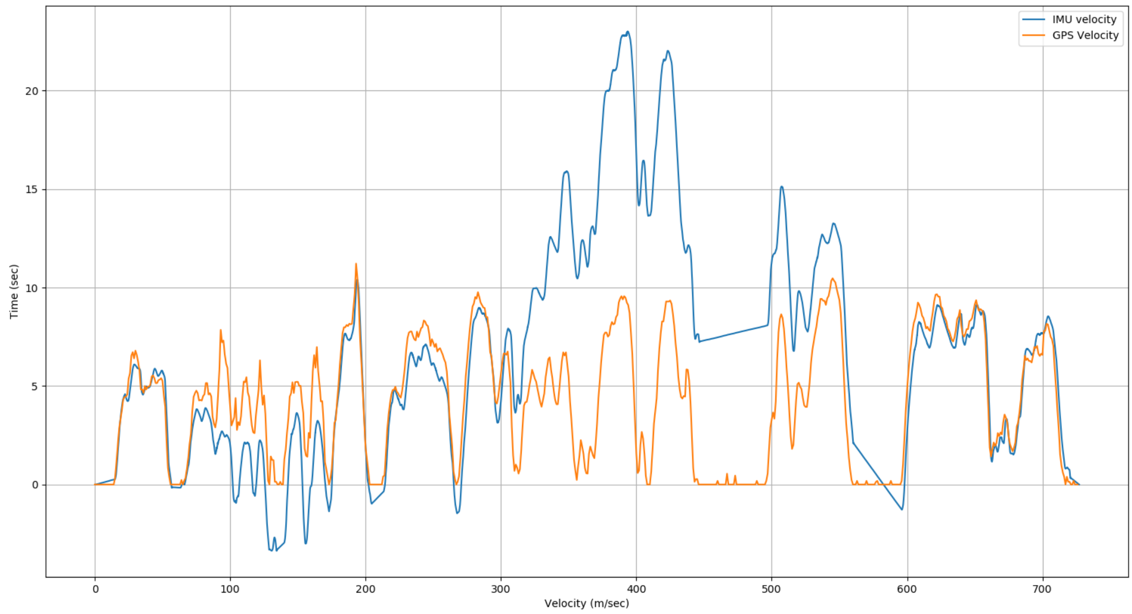
Plot 4: The yaw from the magnetometer has large noises (high-frequency characteristic), while the yaw from Gyroscope has bias/drift (low-frequency characteristic) that causes errors in the yaw to increase with time. To remove these errors, we use a low-pass filter with a Magnetometer and a high-pass filter with Gyroscope data. Further, filtered data for Magnetometer and Gyroscope is combined using complementary filters. **Complementary filter:** $Yaw_{Comp} = \alpha * (Yaw_{Mag}) + (1 - \alpha) * (Yaw_{Gyro})$



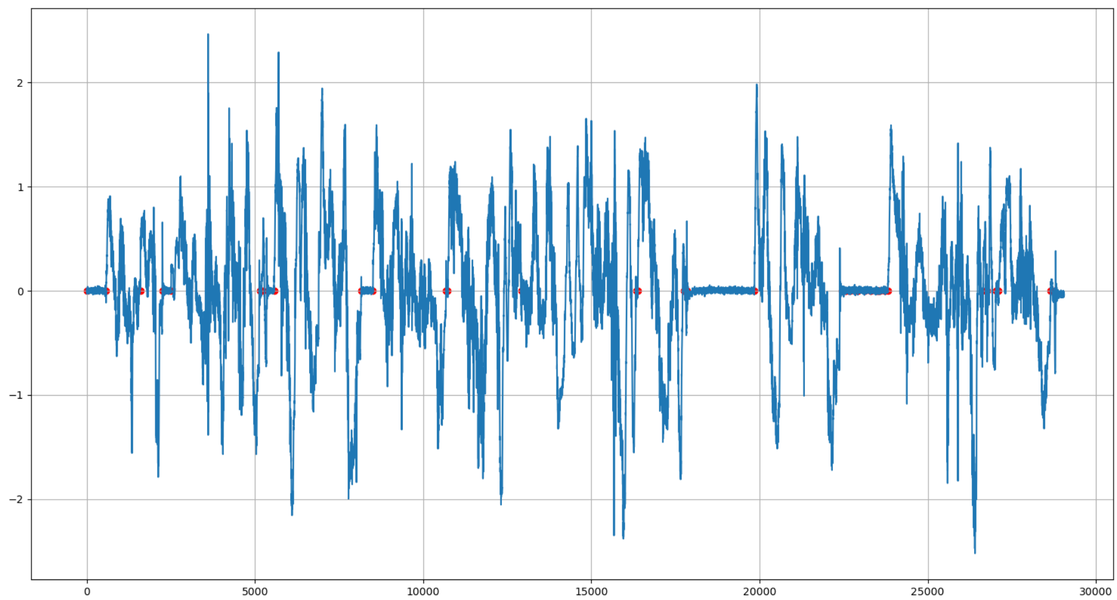
Plot 5: Yaw from IMU is compared with yaw from the complementary filter, and further the complementary parameter is tuned to minimize variations between the two datasets.



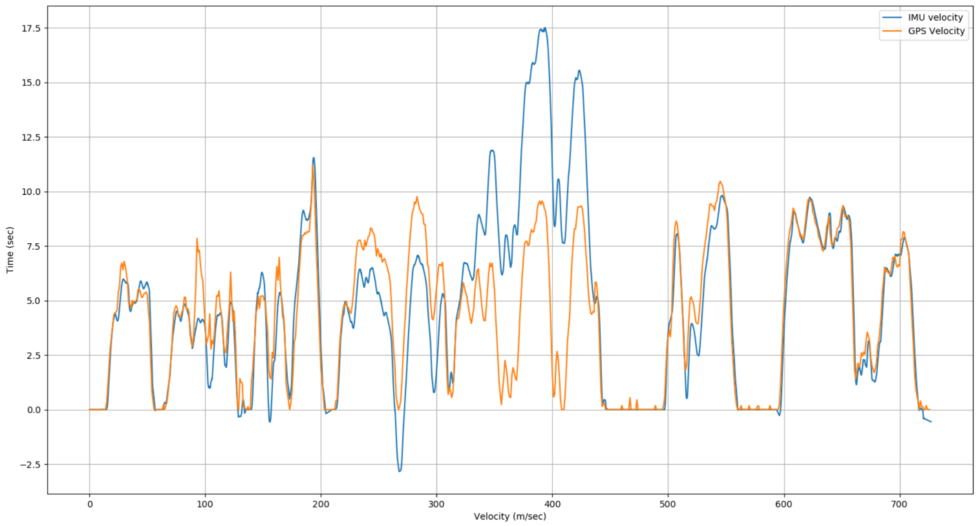
Plot 6: Forward velocity is calculated by integrating linear acceleration along the x-axis, however, there are significant errors in between. The Source of these errors is varying bias and gravity along x on slopes.



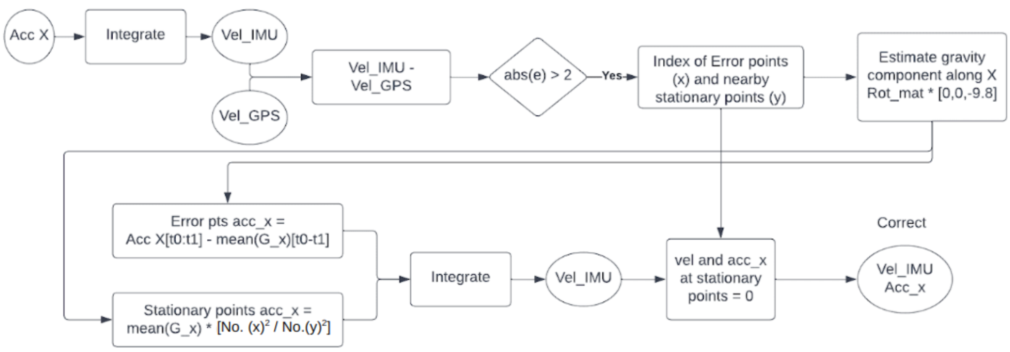
Correction Logic: To correct errors in velocity estimation, we divide the acceleration data into different sectors by finding indexes with Jerk Value < 0.05 for more than 2 seconds. Identified indexes give points where the car was stationary. Mean bias is evaluated for each stationary period and subtracted from all points until the next stationary period starts.



Plot 7: Integrating the acceleration data after removing varying biases generates the following velocity time series plot. However, in the period between 200-300 sec, there is a negative bias, and in data between 300-450 sec, there is a positive bias, which is not resolved by our previous approach. The source of these biases is the road inclinations - positive and negative slope that occurs between two stopping periods.



Correction Logic - The above biases are corrected using pitch and roll data, which is used to reduce the gravity component along X-axis on slopes. However, correcting the data for the above biases led to addition / reduction in distance traveled (given by area under vel-time curve). To correct for these, the basic principle I am using is conservation of distance traveled which is given by 0.5*Acc_x*t^2. Total distance added or reduced due to gravity component, is added / subtracted in the stationary phase of the car. After correcting for this error, the velocity and acceleration values in the stationary period are reduced back to 0. Implemented logic is further expressed below:



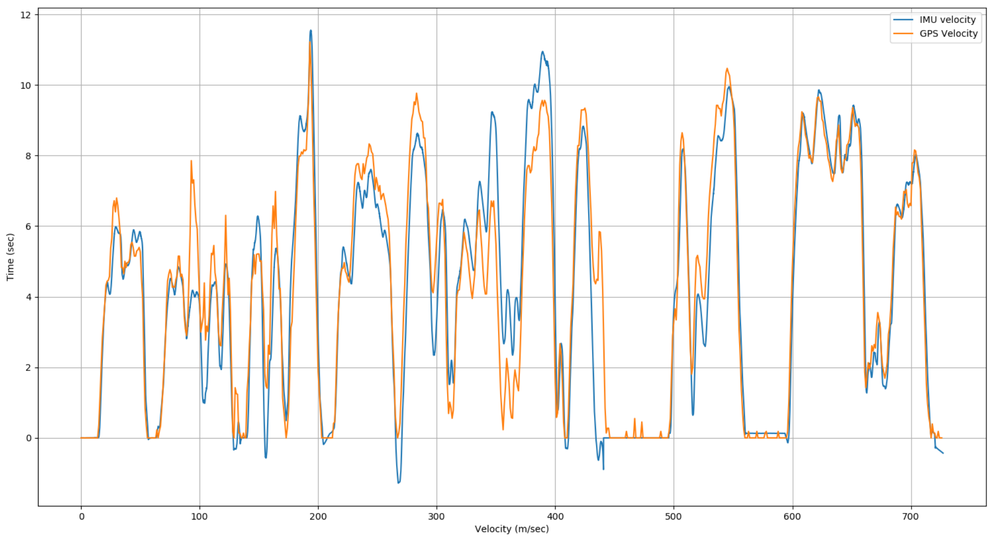
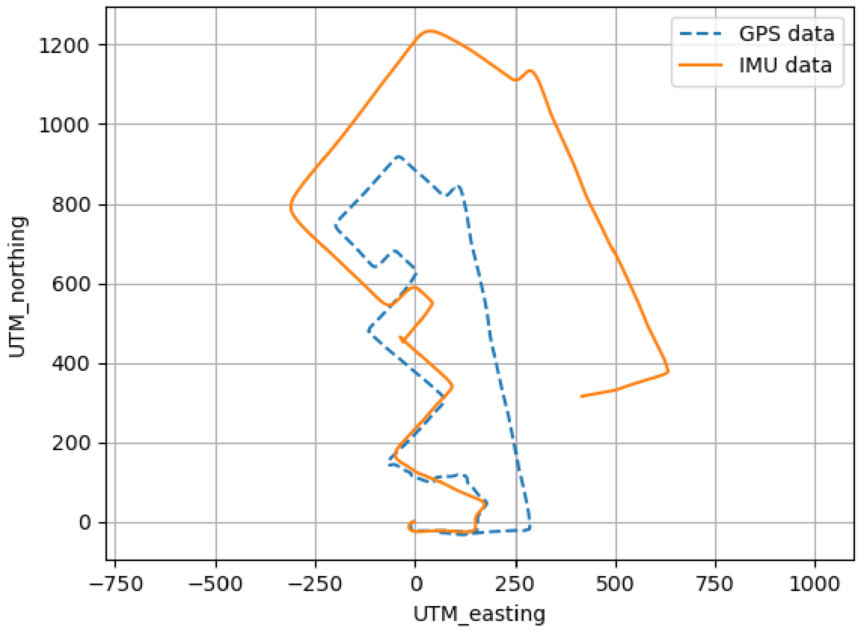


Image: Corrected Velocity plot

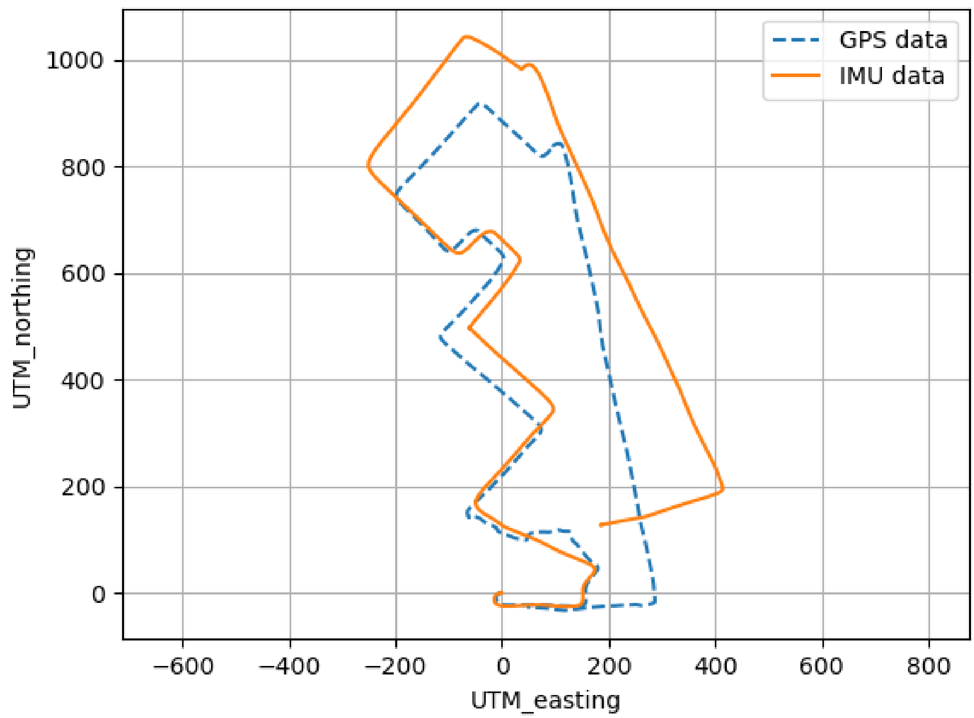
Plot 8: Forward velocity is divided into components along Easting and Northing direction using yaw angle

V(easting) = V*Cosine(Yaw angle) | V(northing) = V*Sin(Yaw angle)

Velocity components are integrated to estimate northing and easting coordinates for each time stamp, thus giving the navigation trajectory. Further, this trajectory is compared with the trajectory from GPS.

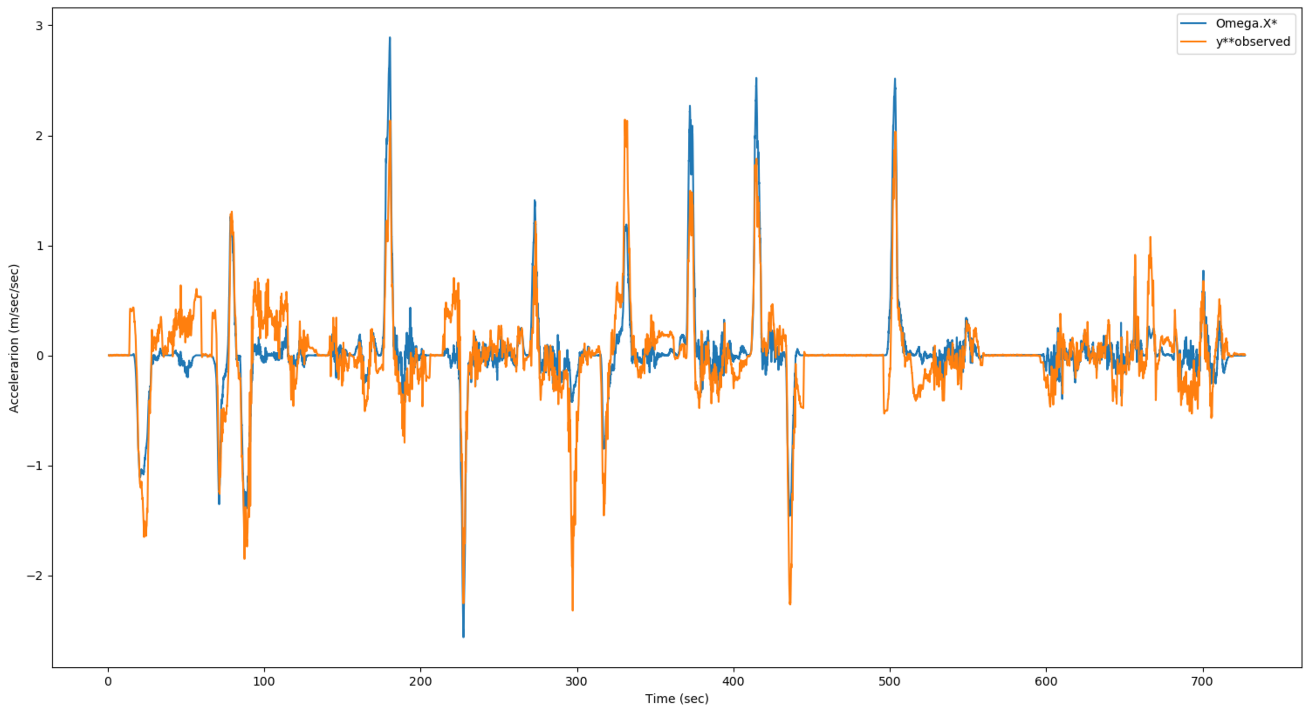


(Image: Easting and Northing plot for velocity plot with no corrections for slopes during movement)



(Image: Easting and Northing plot for final corrected velocity data)

Plot 9: Plot for linear acceleration(y) observed by IMU vs. linear acceleration(y) calculated by $(\omega * \frac{dx}{dt})$



Discussion:

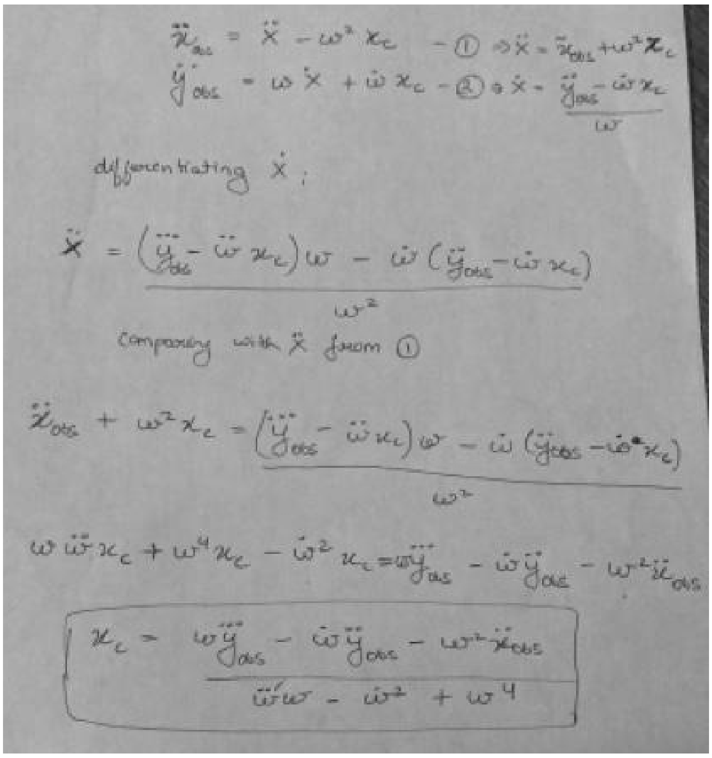
Developing a navigation stack using IMUs requires calibration of Magnetometers, removing Biases from Accelerometers and Gyroscopes to estimate the heading and attitudes of the IMU or System with mounted IMU. Calibration of the magnetometer was done by driving the car in a full circle and then mapping the trajectory generated by the magnetometer. Differences like warping and rotation in observed and driven trajectories give calibration values for soft-iron errors, and mean for MagX and MagY gives calibration values for hard-iron errors. In our experiment where an IMU was mounted inside a car, sources for Hard-iron errors may include amplifiers or speakers in the cars, and Soft-iron errors may include the iron and nickel in the car body. Calibration values are used for correcting magnetometer data from Navigation Dataset to remove known biases of the system (Car, Robot, Machine, etc.).

Yaw / Heading values are calculated from the corrected magnetometer data (atan2(MagY , MagX)). Also, the change in yaw with time can be estimated by integrating the angular velocity from the Gyroscope. In Sensor fusion, we implement a complementary filter to estimate yaw by combining yaw data from Magnetometer and Gyroscope - $Yaw_{Comp} = \alpha * (Yaw_{Mag}) + (1 - \alpha) * (Yaw_{Gyro})$. However, before passing the yaw values from Mag and Gyro we need to pass these through a low-pass and high-pass filter respectively to remove high-frequency noises and drift-biases from these signals. Cut-off frequencies for these filters are estimated through trial-and-error - Low-pass: 0.08Hz, High-pass: 1e-6Hz. As per the observation, data from Gyro gives a better result than Magnetometer, so based on this, we selected an alpha(α): 0.2 for our complementary filter. Better results from gyro can be mapped to the fact that magnetometer data is highly affected by the surrounding environment while gyroscopes are not much affected by the changing environment. For example, in our experiment, a magnetic field from the power lines of nearby tram lines generated considerable errors in our dataset, which were difficult to be characterized.

For mapping the trajectory of the navigated map, we need to estimate the forward velocity along with the heading calculated by sensor fusion. Forward velocity is calculated by integrating the linear acceleration component along the forward direction. However, the varying biases and unaccounted gravity components on the slope leading to large errors in estimated velocities. To account for varying biases, we estimate the mean errors at stationary points and subtract these from data points up to the next stationary period. There can still be additional errors, as observed in Plot 7, which are left unaccounted for as we move through inclined roads between two stationary periods. To account for gravity components on such slopes, we can use the Roll, Pitch, and Yaw from IMU to get acceleration due to gravity in the moving direction and subtract these from acceleration values.

The estimated forward velocity, along with the heading from the complementary filter, is used to map the trajectory of the path navigated by our car. When the estimated trajectory from IMU is compared with the measured trajectory by GPS, we observe the deviation between the two trajectories increases over time. The cause for this increasing deviation can be mapped to the bias-instability of IMUs.

The above calculation for dead-reckoning with inertial sensors assumes that the sensor is placed at the center of mass - about which the car rotates. In cases where the offset between the COM of the car and the IMU position is less, the calculation error is minimal and can be ignored. However, this offset needs to be considered in critical applications like space vehicles. For example, in Plot 9, where we estimated the value for linear acceleration along y for IMU using the equation:
$d^2y_{obs} / dt^2 = d^2Y_{car} / dt^2 + \omega * dX/dt + d\omega/dt * x_c$ and compared it to the observed acceleration in y, estimates agree well with the observation; however, there are some discrepancies which may be due to the assumption that offset is 0. We can estimate the value of the offset from our observed and estimated values using the equations in the image shown below. In our case, the value for the offset comes to be approximately **-0.42m** which suggests that the IMU was placed at 42cm offset from COM (between the car front seats) in the opposite direction to the moving axis.



$\ddot{x}_{obs} = \ddot{X} - \omega^2 x_c$ — ① $\Rightarrow \ddot{X} = \ddot{x}_{obs} + \omega^2 x_c$

$\ddot{y}_{obs} = \omega \dot{X} + \dot{\omega} x_c$ — ② $\Rightarrow \dot{X} = \frac{\ddot{y}_{obs} - \dot{\omega} x_c}{\omega}$

differentiating $\dot{X}$:

$$\ddot{X} = \frac{(\dddot{y}_{obs} - \ddot{\omega} x_c)\omega - \dot{\omega}(\ddot{y}_{obs} - \dot{\omega} x_c)}{\omega^2}$$

comparing with $\ddot{X}$ from ①

$$\ddot{x}_{obs} + \omega^2 x_c = \frac{(\dddot{y}_{obs} - \ddot{\omega} x_c)\omega - \dot{\omega}(\ddot{y}_{obs} - \dot{\omega} x_c)}{\omega^2}$$

$$\omega \ddot{\omega} x_c + \omega^4 x_c - \dot{\omega}^2 x_c = \omega \dddot{y}_{obs} - \dot{\omega} \ddot{y}_{obs} - \omega^2 \ddot{x}_{obs}$$

$$\boxed{x_c = \frac{\omega \dddot{y}_{obs} - \dot{\omega} \ddot{y}_{obs} - \omega^2 \ddot{x}_{obs}}{\ddot{\omega}\omega - \dot{\omega}^2 + \omega^4}}$$

Result:

We observe the dead-reckoning to work effectively for initial data sets; however, with each passing turn, the error in the measurements from IMU keeps on increasing. The primary cause of this error is the Bias-instability of the Gyro, which gives drifting error in the estimated trajectory compared with GPS. Another major source of error in trajectory estimation is the unaccounted slopes in the moving period of IMUs that add a gravity component to the moving axis. The best performance was observed for initial data points when the car moved straight on a leveled road. The image shown below gives the net displacement measured by IMU vs. GPS represents error in measurements with time. For initial 350 seconds, IMU and GPS values match for our dead reckoning approach, however, the effect of unaccounted slope and drift from Gyros led to a final displacement error of around 300 meters.