EECE 5554 Robotics Sensing and Navigation
Lab-4 Report

Part-A - Data Collection for Dead-Reckoning:

The data collected and the drivers used for this Lab were taken from shirgaonkar.a (Gitlab user name) with Professor Dorsey's permission and mentioning the same as per her advice.

*The analysis of the data was done partially in python (magnetometer soft iron and hard iron corrections) and Matlab, both the scripts are added.

Part-B - Analysis of the data collected:

We have collected IMU and GPS data which had a lot of noise and errors. The main aim is to reduce those errors/noise and plot that data for comparing with other alternatively calculated data(eg. trajectory from imu data and gps data)

1.Estimate the heading (yaw):

The plot in figure-1 shows the comparison between Raw Magnetometer data and corrected Magnetometer data after removing the hard iron and soft iron errors.

Hard iron correction:

Hard-iron distortion is caused by materials that display a constant, additive field to the earth's magnetic field, resulting in a constant additive value to the output of each magnetometer axis.

As observable from Fig-1 the uncorrected magnetic field plot(red circle) has a center at (-0.1,0.32) so to correct that translation we calculate the offset of each axis and subtract it from them. Considering X and Y to be our raw data from magnetic_x and magnetic_y.

X offset = [maximum(x) + minimum(x)]/2

Y offset = [maximum(y) + minimum(y)]/2

Soft iron correction:

Soft-iron distortion is created by material that alters or distorts a magnetic field, but does not necessarily create a magnetic field, and hence is not additive. soft-iron distortion is affected by the material's orientation relative to the sensor and the magnetic field. This distortion is found when an ellipse with an angular rotation is recognized. The following is a procedure for eliminating moderate iron distortion.

We must estimate the length of the primary axis by eyeballing it or using an elliptical fit.

find the maximum and minimum of x and y This gives us the distance of the farthest point (x, y) from the center (0,0). The distance r may be calculated using

$$r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Θ = arcsin(y1/r)

We use the rotation matrix to rotate the ellipse to bring it to position where major axis and minor axis are aligned to the x and y cartesian line.

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

V1 is the product of rotation matrix R and vector components (v) of mag_x,mag_y.

As a gist of how to remove the hard and soft iron errors we have to perform

Subtracting the following offsets from respective raw magnetic-x and magnetic-y values

X offset = [maximum(x) + minimum(x)]/2

Y offset = [maximum(y) + minimum(y)]/2

Followed by rotation and scaling with the scale factor which is the ratio of length of major axis to minor axis.

Plot-1: The magnetometer X-Y plot before and after hard and soft iron calibration:

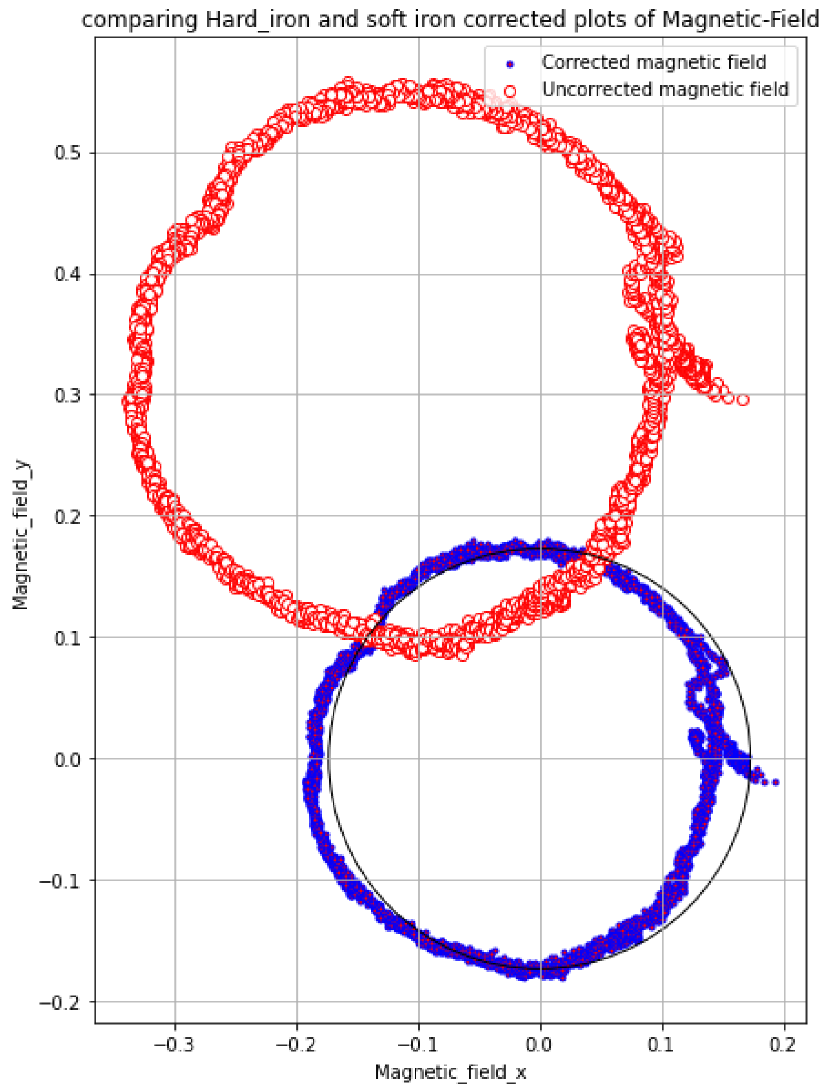


Fig-1 Comparing the Magnetic field in X and Y directions before and after correcting the data.

The above plot shows the raw scattered plot(red circle) to be translated to a circle translated, rotated and scaled with origin as shifted which is the desired plot after correcting the hard and soft iron errors.

Plot-2:

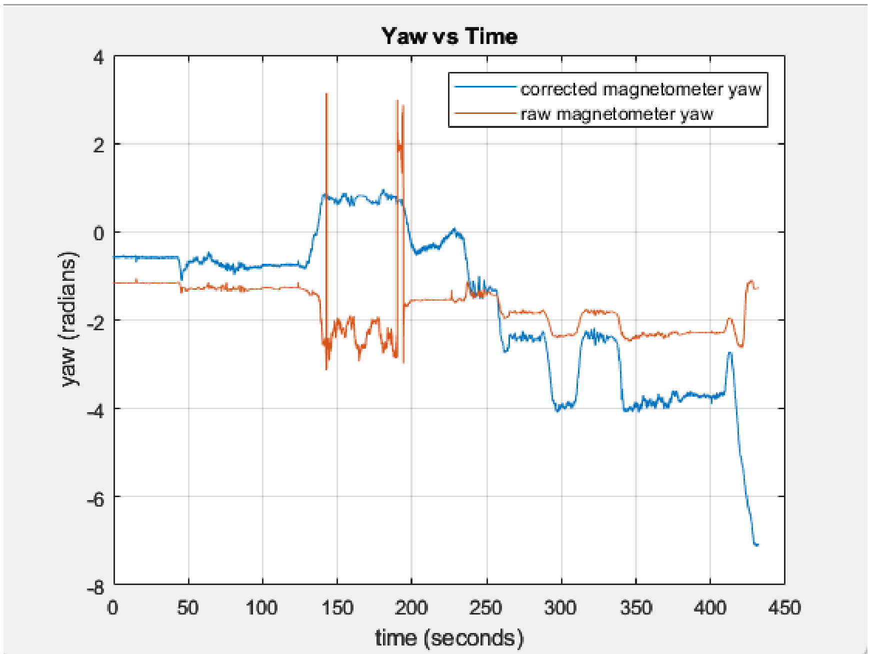


Fig-2 Time series magnetometer data before and after the correction

Plot-3:

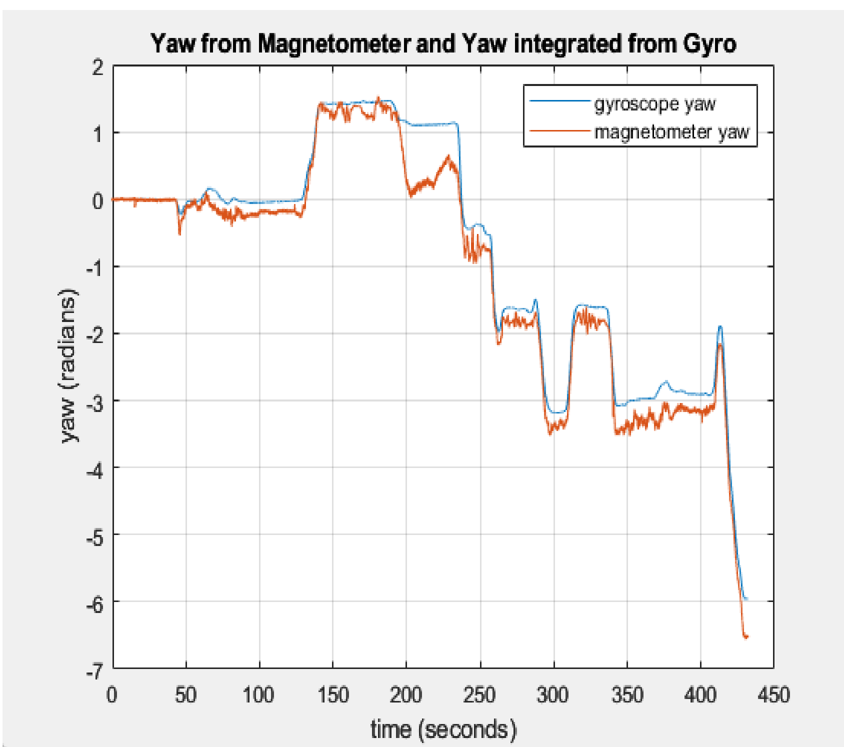


Fig-3 Magnetometer Yaw & Yaw Integrated from Gyro together

Plot-4:

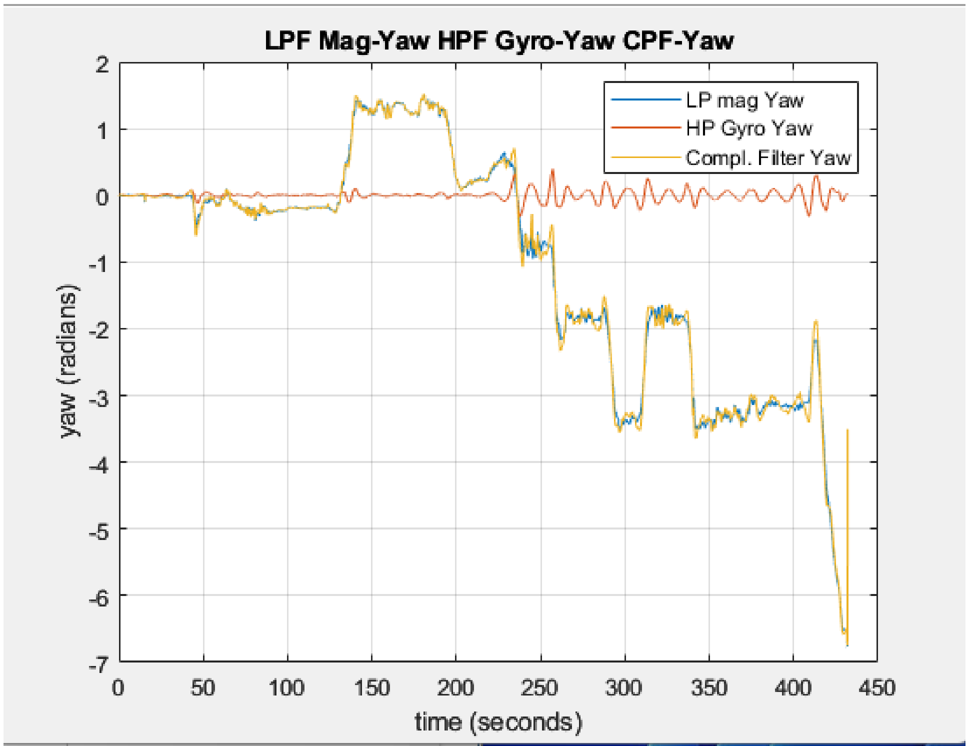


Fig-4

LPF, HPF, and CF plots together

In order to use a complementary filter to find a combined estimate of yaw the following steps were followed:

1. First the yaw is calculated from the magnetic x and y values using the arctan(-y/x). Where the y and x are the corrected Magnetic_Y and Magnetic _X.
2. Now this corrected magnetic yaw is unwrapped and passed through a low pass filter using, y = lowpass(x,fpass,fs) where x is the corrected magnetic yaw that has been sampled at a rate of 40 hertz. fpass is the passband frequency of the filter which was set to 0.0001 hz
3. The output of the LPF is depicted as a blue coloured plot in fig-4.
4. The gyro_z which is the angular velocity in z is integrated and the integrated value is passed through the high pass filter which has been sampled at a rate of 40hz and the pass band frequency of the filter is set to 0.07.

5. Once the lpf and hpf are obtained, the sum of both lpf and hpf has given the complementary filter output.

All the graphs of LPF, HPF, CPF are plotted and plotted in the fig-4

Plot-5:

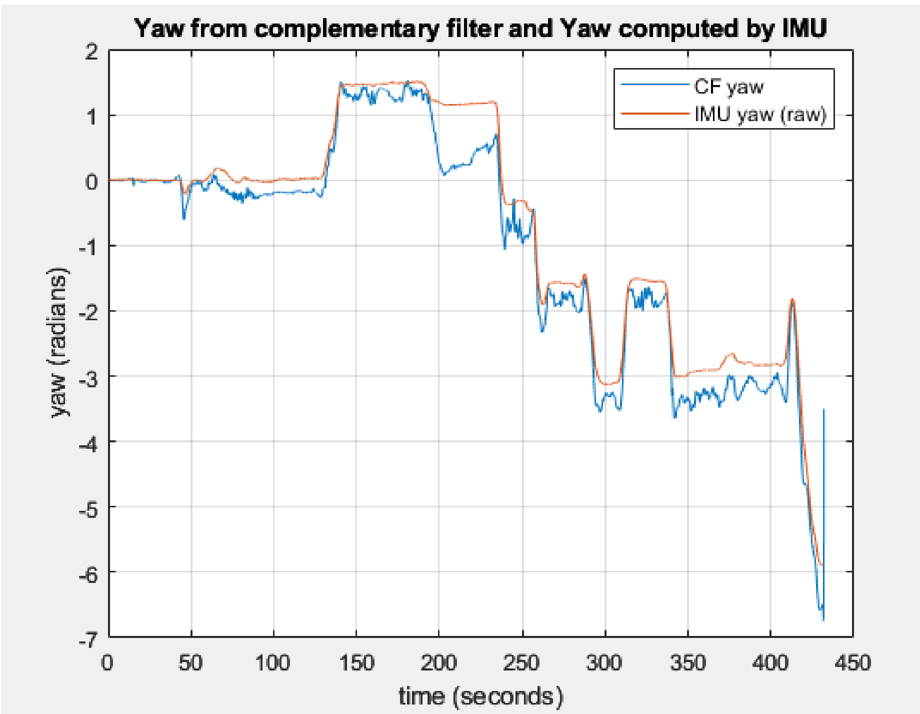
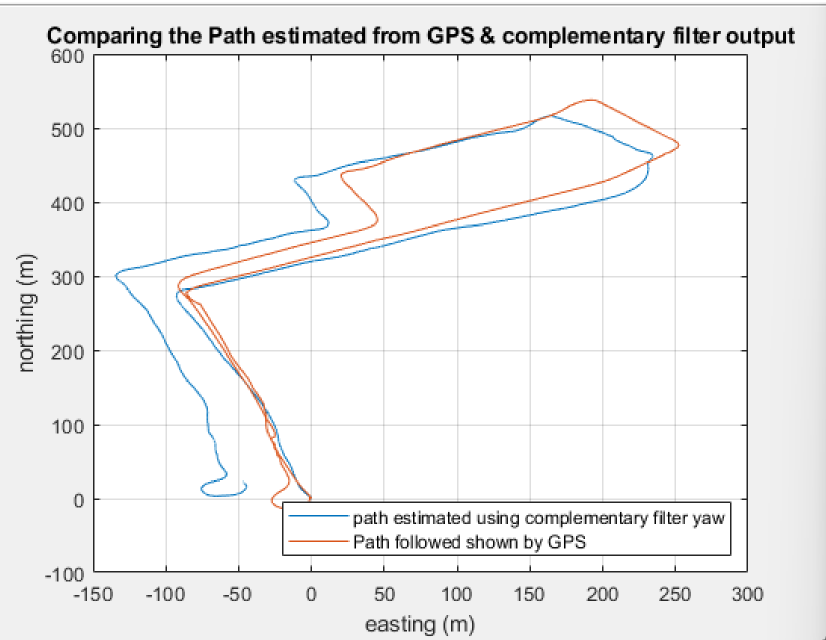


Fig-5 Yaw from the Complementary filter & Yaw angle computed by the IMU together

There were two reasons for considering the yaw from complementary output as better for navigation than the yaw computed by IMU.

1. It is obtained as a sum of both the low pass and high pass filters,signal smoothening due to low-pass filter and signal sharpening due to high-pass filter are observed to the complementary filter output.
2. When trajectory paths were estimated by considering both gyro integrated Yaw and complementary filter output yaw for velocity estimation and there by for path estimation separately and comparing with the gps trajectory, promising results were observed when complementary filter output is considered.This is shown in figure-6.



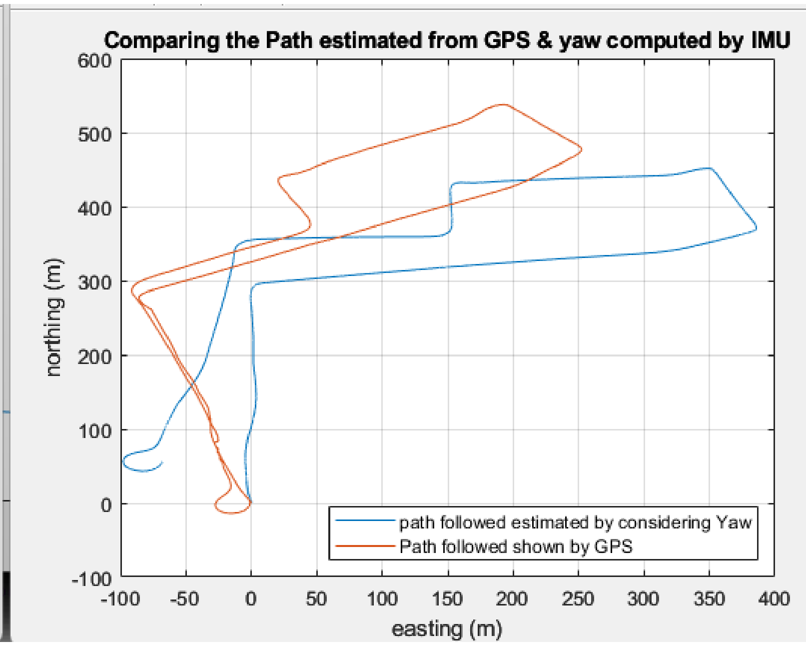
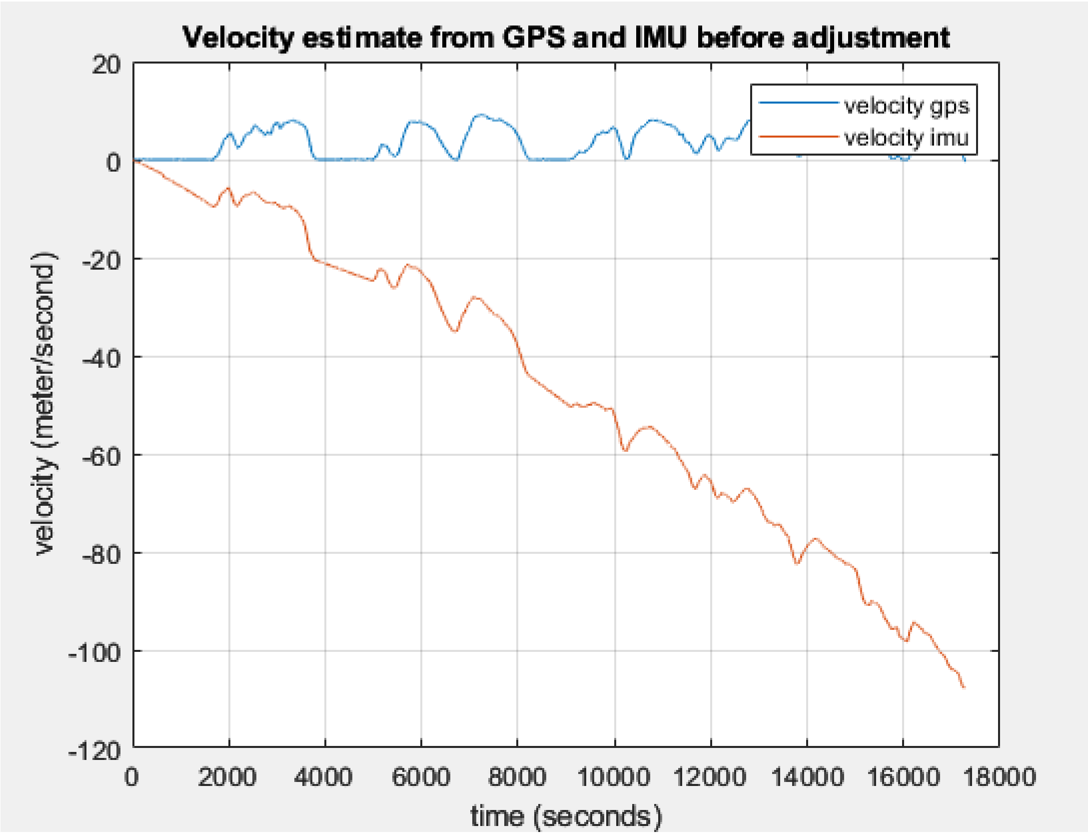


Fig-6 Comparing the trajectories obtained considering yaw from complementary filter and yaw from imu

Plot-6:



Plot-7:

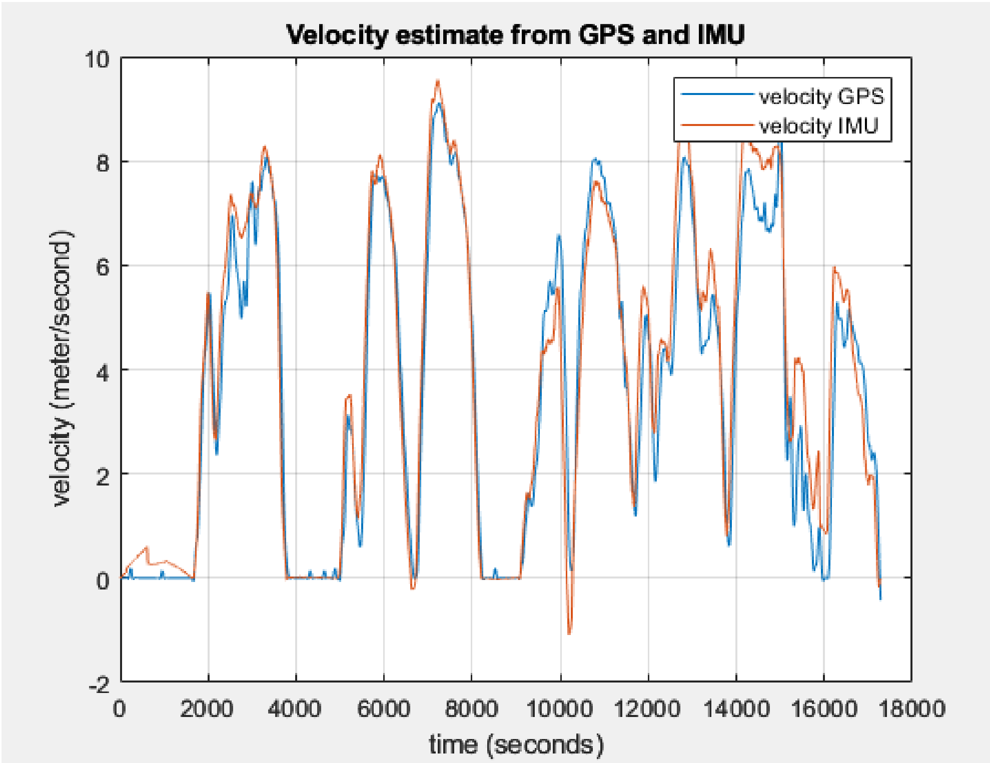


Fig-8 Velocity estimate from the GPS with Velocity estimate from accelerometer after adjustment

.

Before making any adjustments or corrections to the acceleration from IMU (Fig-7) by directly integrating it the obtained velocity shows negative values and even the sections of the plot which were at rest/motion-less depict velocity in negative direction which isn't actually true hence bias was removed from the acceleration and then integrated. The resultant velocity estimate from accelerometer (Fig-8) appears to be in coherence with the gps velocity plot calculated by v=s/t (v is the gps velocity, s is the displacement calculated considering utm easting and utm northing, t is the time interval). The main adjustment made was to subtract the bias's from the data, which in this case was to find regions of graph at rest (no motion) and calculating the mean and subtracting it from the remaining data and repeating this process for all stationary regions in the acceleration plot and then integrate that acceleration to obtain velocity from accelerometer after adjustment.

The only discrepancy observed in velocity estimated from IMU acceleration is the presence of negative velocity at certain regions of the graph which isn't observed in the velocity estimated from GPS. The one main reason to take the velocity estimate from gps to be promising is the perfect zero velocity at points where there is halt or a stop in the motion of the car. The velocity estimate from IMU shows some error which needs a lot more high level filtering, but would be helpful in approximate indoor navigation.

Dead Reckoning with IMU:

Plot-8:

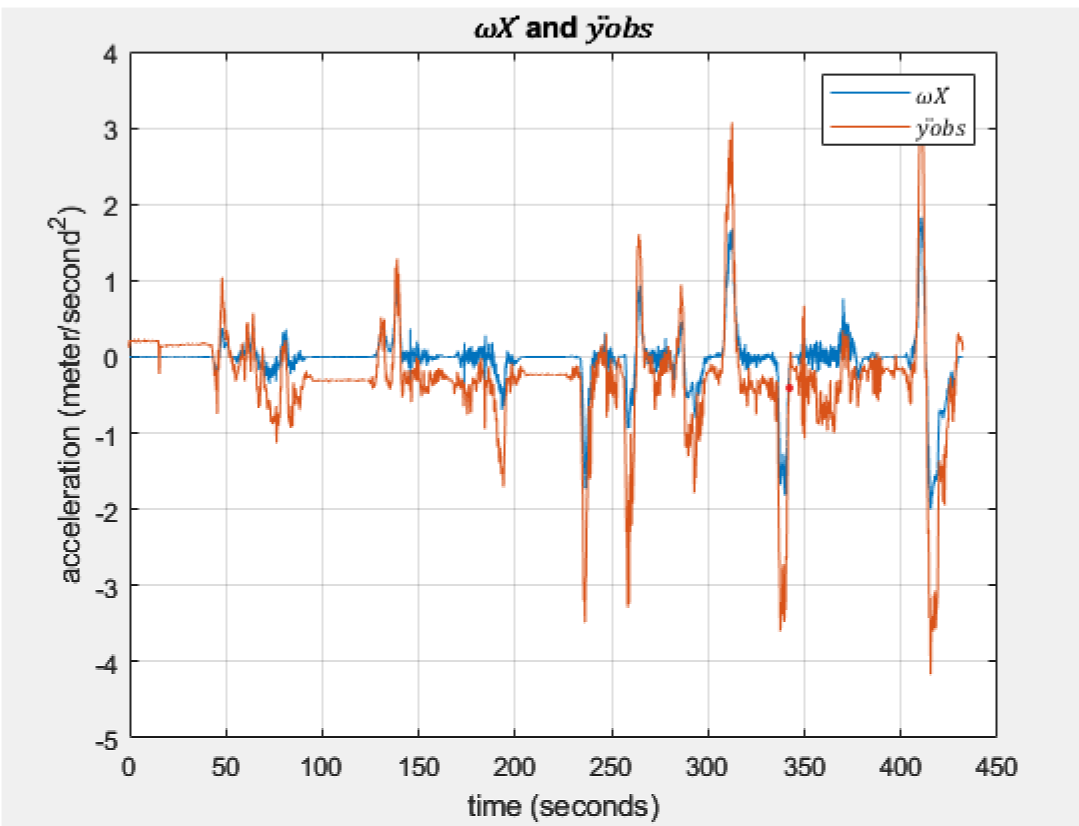


Fig-8 ωX **and** $\ddot{y}obs$ **plotted together**

The ωX and $\ddot{y}o$bserved are obtained from the formula

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

Notation taken for the position of the center-of-mass (CM) of the vehicle by (X,Y,0) and its rotation rate about the CM by (0,0,ω). We denote the position of the inertial sensor in space by (x,y,0) and its position in the vehicle frame by (xc,0,0) and assuming that Y = 0 (that is, the vehicle is not skidding sideways) and ignore the offset by setting xc = 0 (meaning that the IMU is on the center of mass of the vehicle, i.e. the point about which the car rotates). Then the first equation above reduces to $\ddot{X} = \ddot{x}obs$. So now by integrating $\ddot{X}$ we get $\dot{X}$ which is the velocity of the vehicle w.r.t the center of mass, we can use it to compute ωX and compare it to $\ddot{y}obs$. This corresponding plots can be observed in fig-8. ωX is the the product of velocity obtained from linear acceleration in x direction to angular velocity in z direction. And $\ddot{y}o$bserved is the linear acceleration in y direction. From fig 8 we can see that the noise in $\ddot{y}o$bserved is much greater than to that of the ωX. This is because when integrating the error also gets integrated with the original data causing more noise. But for ωX, the X velocity obtained when multiplied with ω will make the angular rotation compensation and reduces the error .

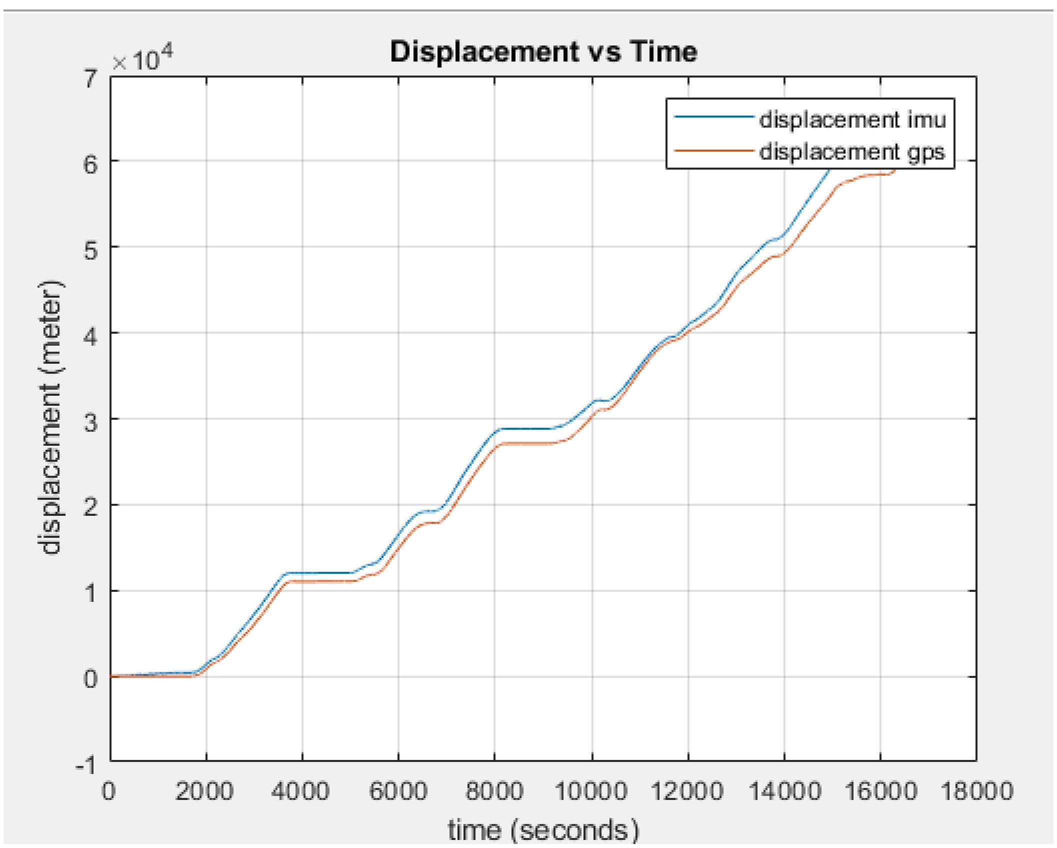


Fig-9 Comparing Displacement vs Time graph measured by IMU and GPS

Plot-9:

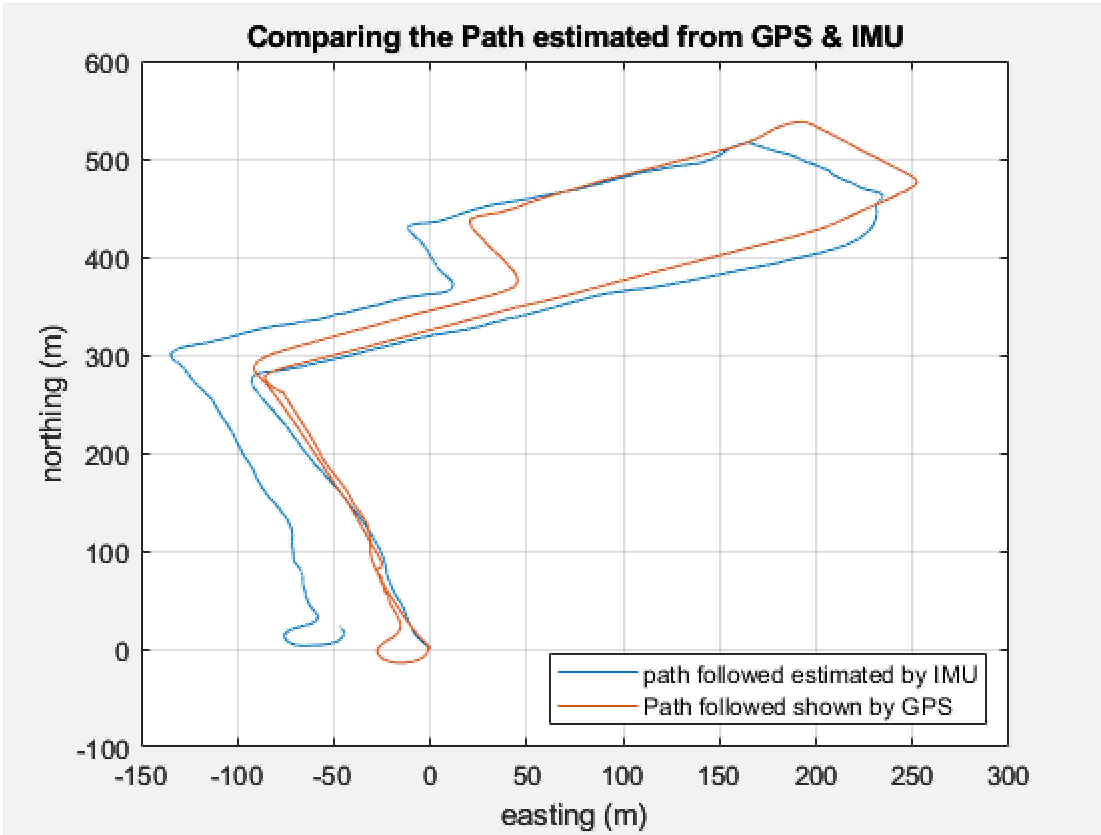


Fig-10 Trajectories measured by the IMU and GPS data

The trajectories xe and xn are calculated by integrating the velocities estimated from gps and imu.Respective graphs are plotted. A very good trajectory was plotted with gps data, nevertheless trajectory using the complementary filter output was coinciding with gps data with slight variation.

Xc calculations:

Using the equations in the below figure

$$v_{sensor}^{U} = v_{car}^{U} + \omega \times \rho_{sensor}^{R}$$

$$a_{sensor}^{U} = a_{car}^{U} + \dot{\omega} \times \rho_{sensor}^{R} + \omega \times (\omega \times \rho_{sensor}^{R})$$

$${}_{T}^{R}R^{T} a_{sensor}^{R} = a_{car}^{U} + \dot{\omega} \times \rho_{sensor}^{R} + \omega \times (\omega \times \rho_{sensor}^{R})$$

$${}_{T}^{R}R^{T} \begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix} = \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\omega} \end{bmatrix} \times \begin{bmatrix} x_c \\ 0 \\ 0 \end{bmatrix}^{R} + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times \begin{bmatrix} x_c \\ 0 \\ 0 \end{bmatrix}^{R} \right)$$

$$\begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix}^{R} = {}_{U}^{R}R \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix}^{U} + \begin{bmatrix} 0 \\ \dot{\omega} x_c \\ 0 \end{bmatrix}^{R} + \begin{bmatrix} -\omega^2 x_c \\ 0 \\ 0 \end{bmatrix}^{R}$$

$$\begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix}^{R} - {}_{U}^{R}R \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix}^{U} = \begin{bmatrix} -\omega^2 \\ \dot{\omega} \\ 0 \end{bmatrix}^{R} x_c$$

Then solving for Xc the obtained value is approximately 55cm .

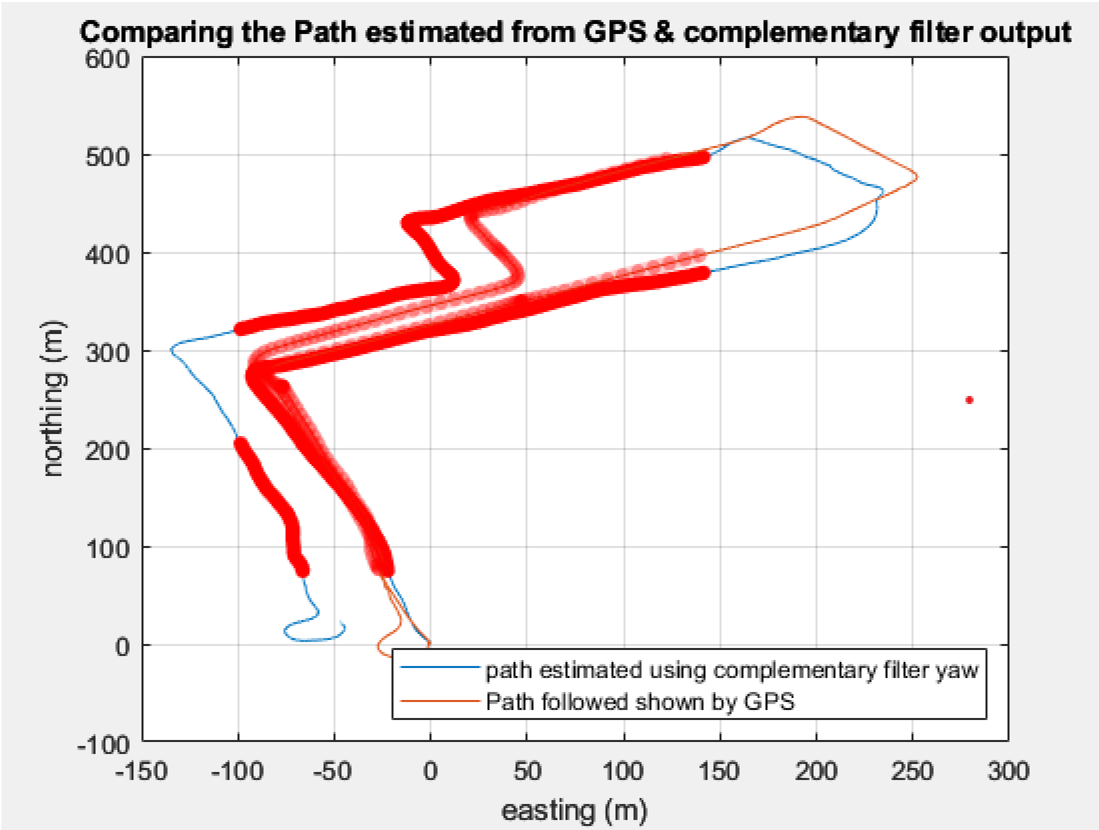


Fig-11 comparing the paths and highlighting the most matching regions

The marked regions in the graph (Fig-11)seem to have a close match in the trajectories and the calculating the time period by (distance /vel) its about 350 sec. During this period the navigation is almost as accurate as GPS data, but after this mark there is a gradual decline in navigation accuracy.