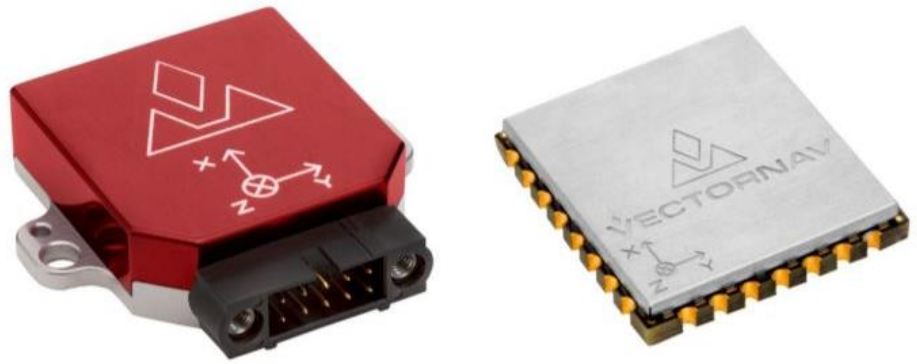
EECE 5554 Robotics Sensing and Navigation LAB 4

Navigation with IMU and Magnetometer

Data Analysis by: Arvinder Singh



Description

In this lab, we are building a navigation stack using two different sensors – GPS and IMU and are trying to understand their relative strengths and drawbacks. The IMU is mounted on the center of the dashboard of car and the GPS is mounted on top of the car. The data is collected in 3 major parts:

1. Stationary Data for 2-3 minutes near Ruggles
2. Circular moving data- To calibrate the magnetometer, car is driven around the round-about 3-4 times
3. Moving data- Car is driven around and data is collected

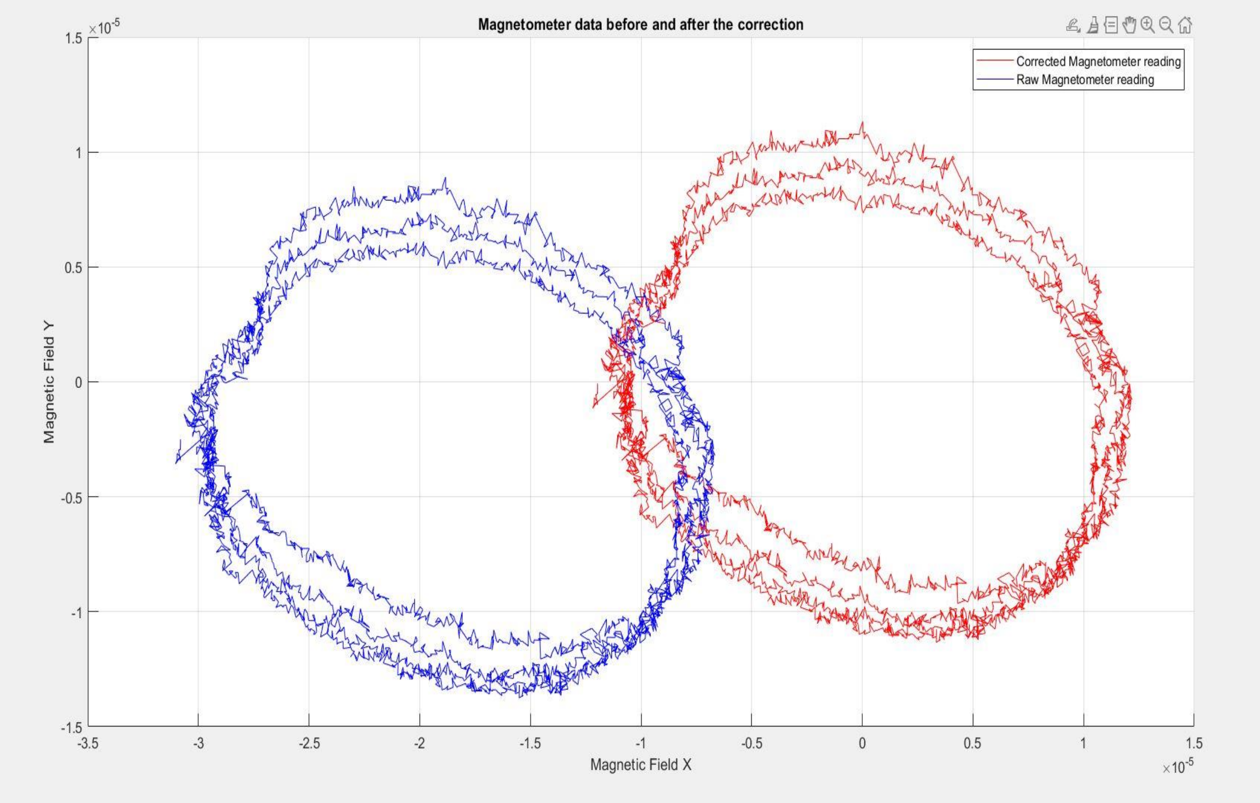
1. Estimate the heading (yaw)

Hard iron distortions will only shift the center of the circle away from the origin, they will not distort the shape of the circle in any way. Soft iron distortions, on the other hand, distort and warp the existing magnetic fields. When plotting the magnetic output, soft iron distortions are easy to recognize as they will distort the circular output into an elliptical shape.

When driving in circles, the soft and hard iron corrections calculated are:

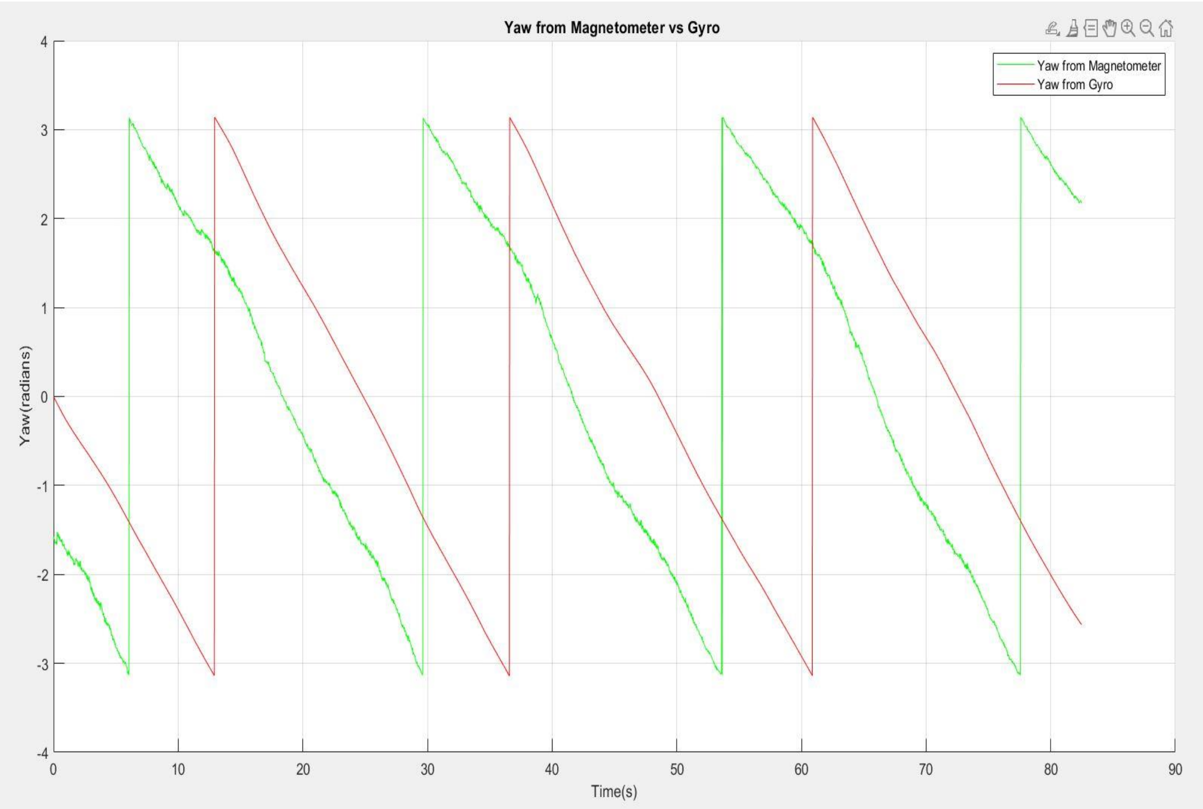
- Soft Iron - No effect (Scale X, Scale Y =1)
- Hard Iron – Offset X = -1.885e-05, Offset Y = -2.42e-06

This means the shape of the circle is not distorted but the center of the circle is shifted due to the hard iron effects. To account for these errors, I subtracted the offset from magnetometer values and then multiplied it by the scale (which is 1 in our case) for both X &Y. The resulting plot for corrected and raw magnetometer readings is as follows:

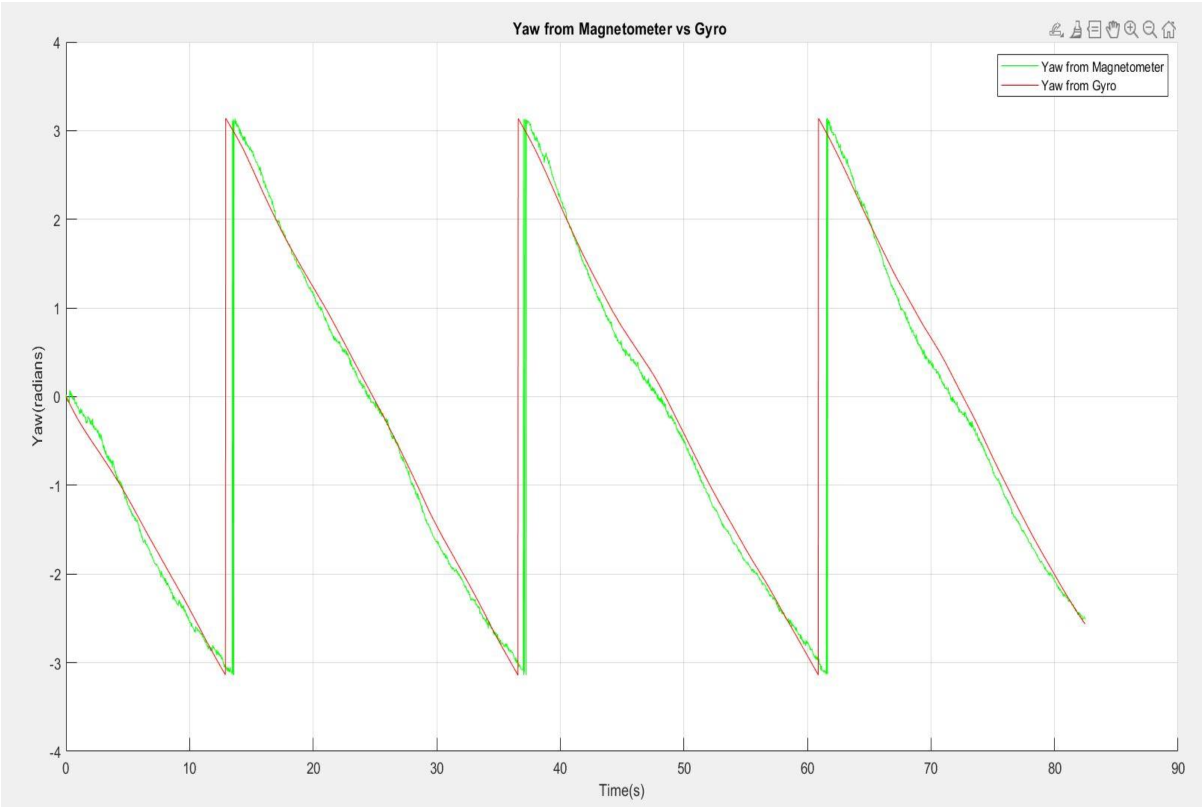


Yaw from Magnetometer vs. Yaw Integrated from Gyro

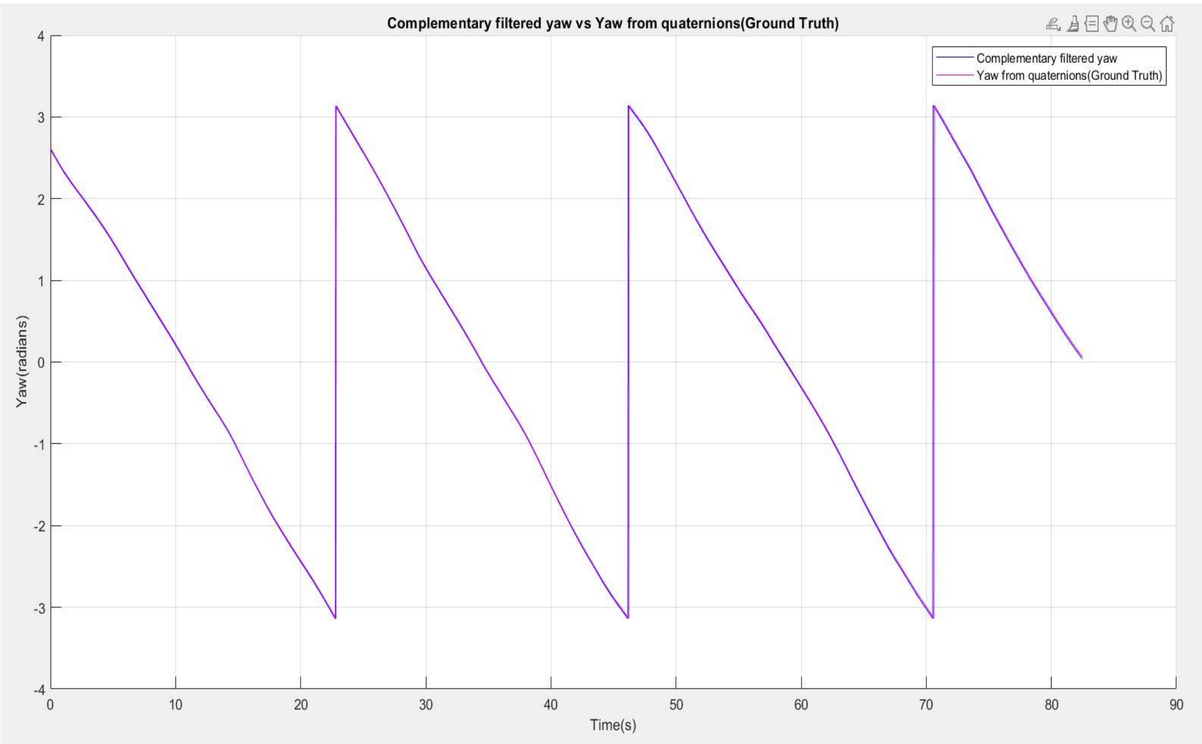
Below we can see the plot for yaw from magnetometer and yaw integrated from Gyro with different starting point (Integrated yaw from gyro starts from 0)



If we make the starting point for both the yaw's (from magnetometer and gyro) same, the plot looks like:-



We can see that the integrated yaw from gyro drifts over time and the yaw from magnetometer has bias. To get a better result we will use a complementary filter to combine the measurements from the magnetometer and yaw rate to get an improved estimate of yaw angle. Below is the plot for complementary filtered yaw and the ground truth or the yaw from quaternions.



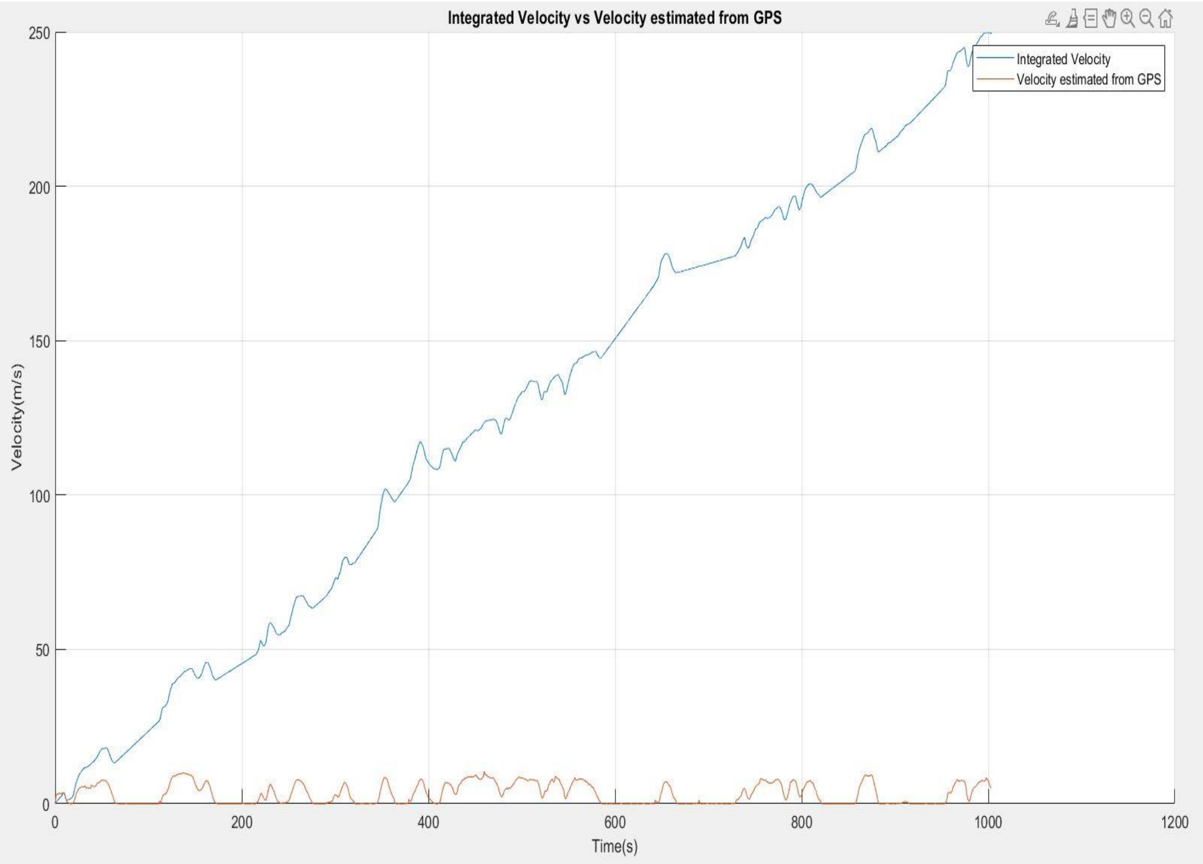
As we can see in the above plot, after using a complementary filter to combine both the yaw's, the Complementary filtered yaw is exactly the same as the yaw obtained from quaternions or the ground truth yaw. Complementary filter as discussed in the class works best when we have combination of two sensors with different time response, as a result the complementary filtered yaw is exactly the same as the ground truth yaw or the yaw obtained from quaternions. On the short term, we use the data from the gyroscope, because it is very precise and not susceptible to external forces. On the long term, we use the data from the magnetometer, as it does not drift.

For the complementary filter we are using α(alpha) = 0.01,

where, complementary filtered yaw = (1-α)*yaw_from_gyro + α*yaw_from_magnetometer

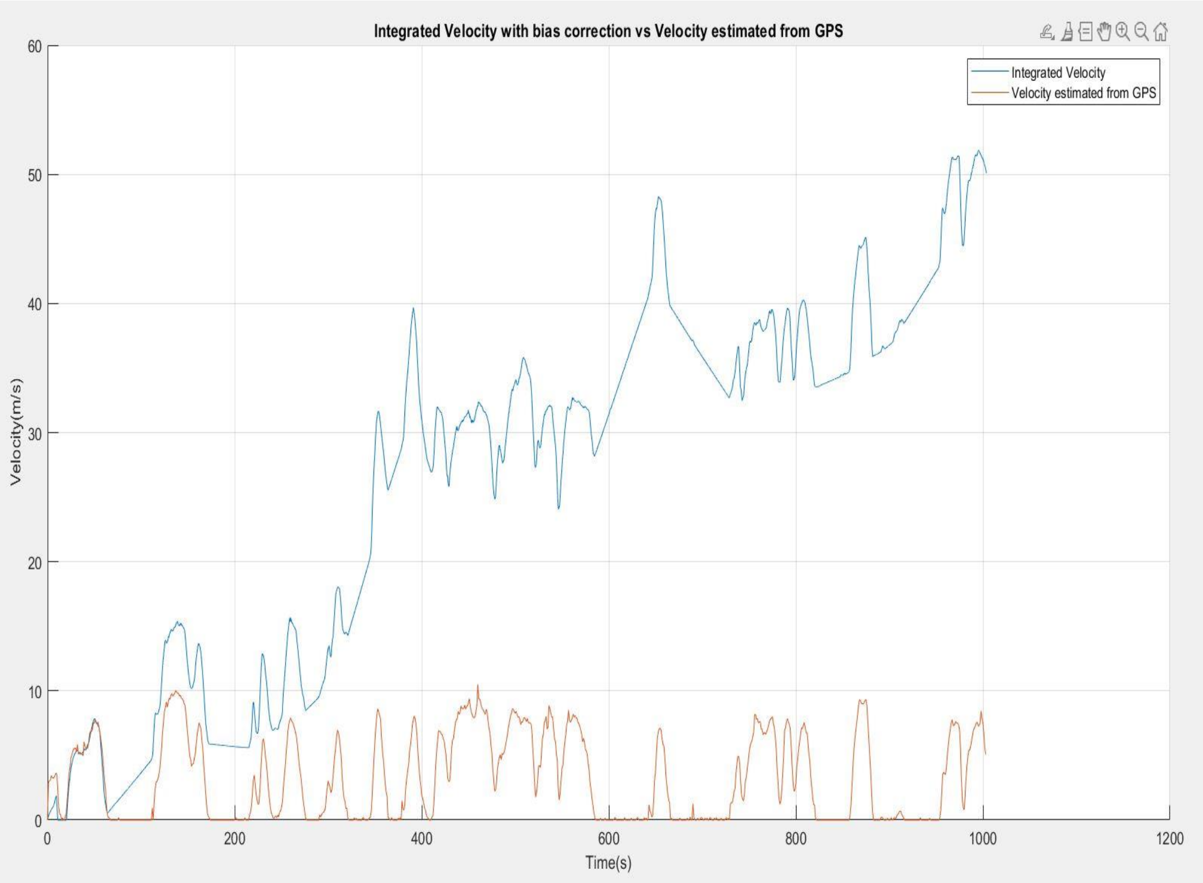
2. Estimate the forward velocity

Below is the plot for forward velocity obtained from integrating forward acceleration and estimating the velocity from our GPS measurements.



The integrated velocity makes no sense at all, it keeps increasing which is due to the acceleration data being integrated which results in drift getting accumulated as the data is integrated. This is why the drift keeps adding with time and we have a very large velocity at the end.

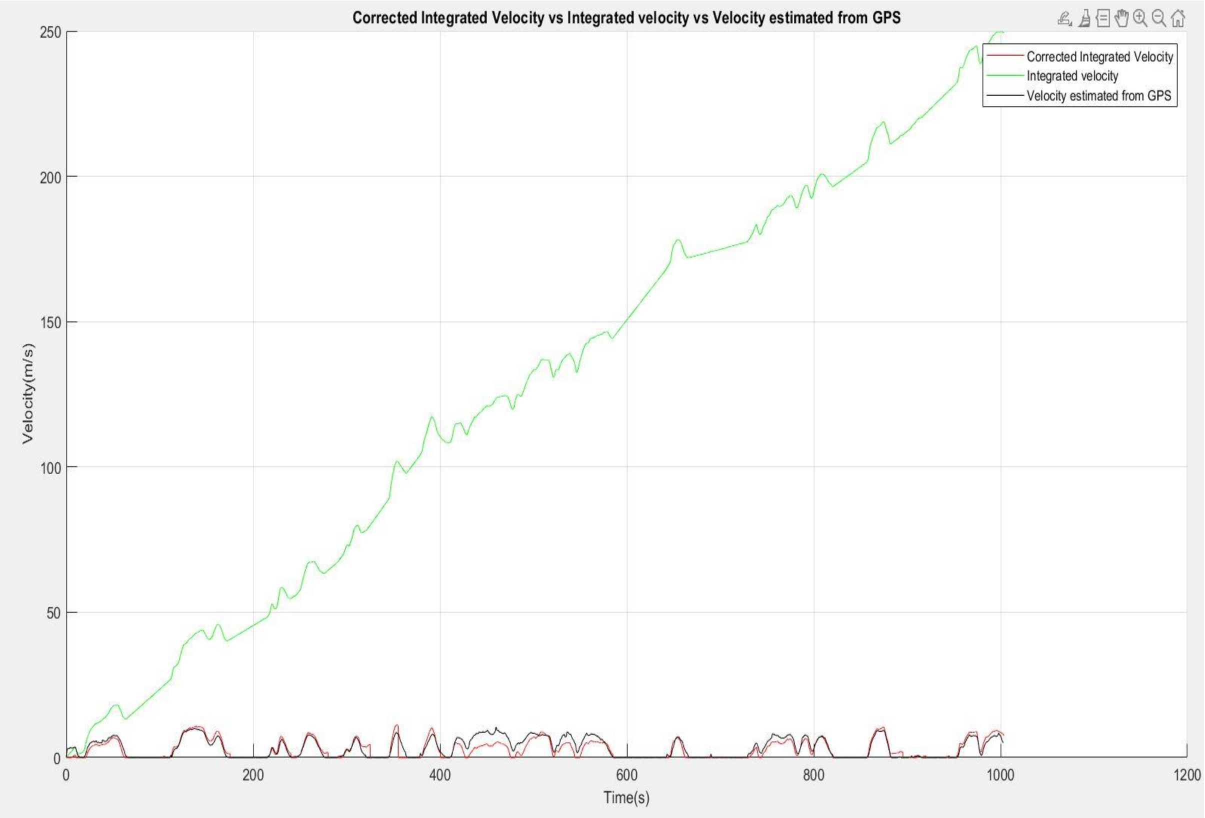
To make the plot more reasonable, we should first subtract the constant bias which is present in our data, this can be calculated from the stationary data. So, I used the stationary IMU data we collected and calculated the mean acceleration in the X direction from the data and subtracted it from the moving acceleration data. The mean acceleration bias in the X direction comes out to be = 0.1986 m/s^2. After subtracting the mean acceleration bias, we still have drift in our data, the plot is as follows:



After bias correction, the plot looks more reasonable but to further improve the integrated velocity estimate, I have further filtered the acceleration data. As discussed in class I reduced the acceleration to 0 where we are stopped. The idea is to check 200 acceleration data points before the current (this doesn't make the system anti-causal) data point and see if the standard deviation of those 200 points is less than the standard deviation of the stationary acceleration data. If yes, then the 200 points are reduced to 0. 200 is chosen as we are getting data at 40Hz so the car would atleast stop for 5secs.

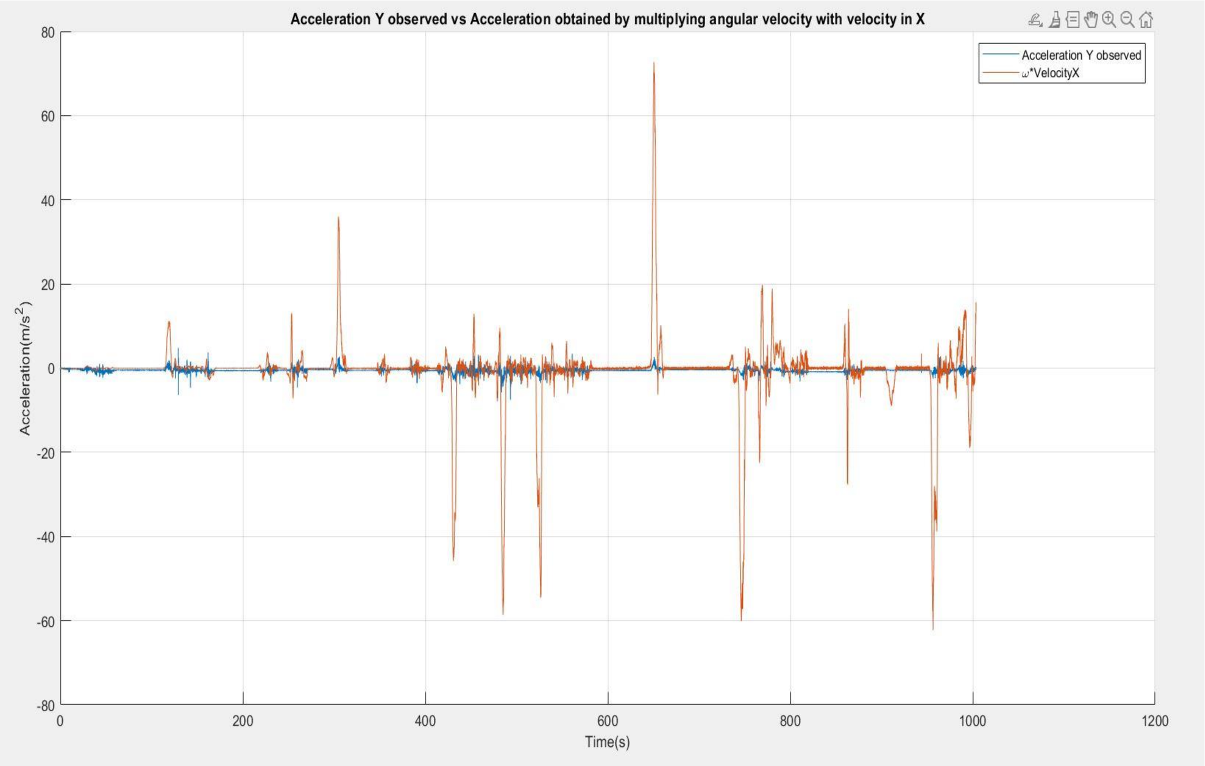
Also, since the drift is not constant, I am also checking 19 points from the current velocity data point to see if their standard deviation is less than 0.0001 which would again mean the car is stopped, If, yes then I calculated the mean of the data which will be our error at the same time I reduced those 19 points to 0. This mean/error gets subtracted from the following data points until we have a data set which has standard deviation less than 0.0001. Our error will keep getting updated as we encounter these data points with standard deviation less than 0.0001 and we keep subtracting the updated error from the following data points. This way our system doesn't get anti-causal since we are using the prior values to update our posterior. As a result, the velocity plot obtained is very close to the ground truth or the plot obtained from GPS data.

Below we can see the integrated velocity plot after bias and drift correction, the ground truth velocity or the GPS velocity and the integrated velocity with bias and drift.



3. Dead Reckoning with IMU

1. Assumed that $\dot{Y} = 0$ (that is, the vehicle is not skidding sideways) and the offset $x_c = 0$. Below is the plot for $\omega\dot{X}$ vs $\ddot{y}_{obs}$:

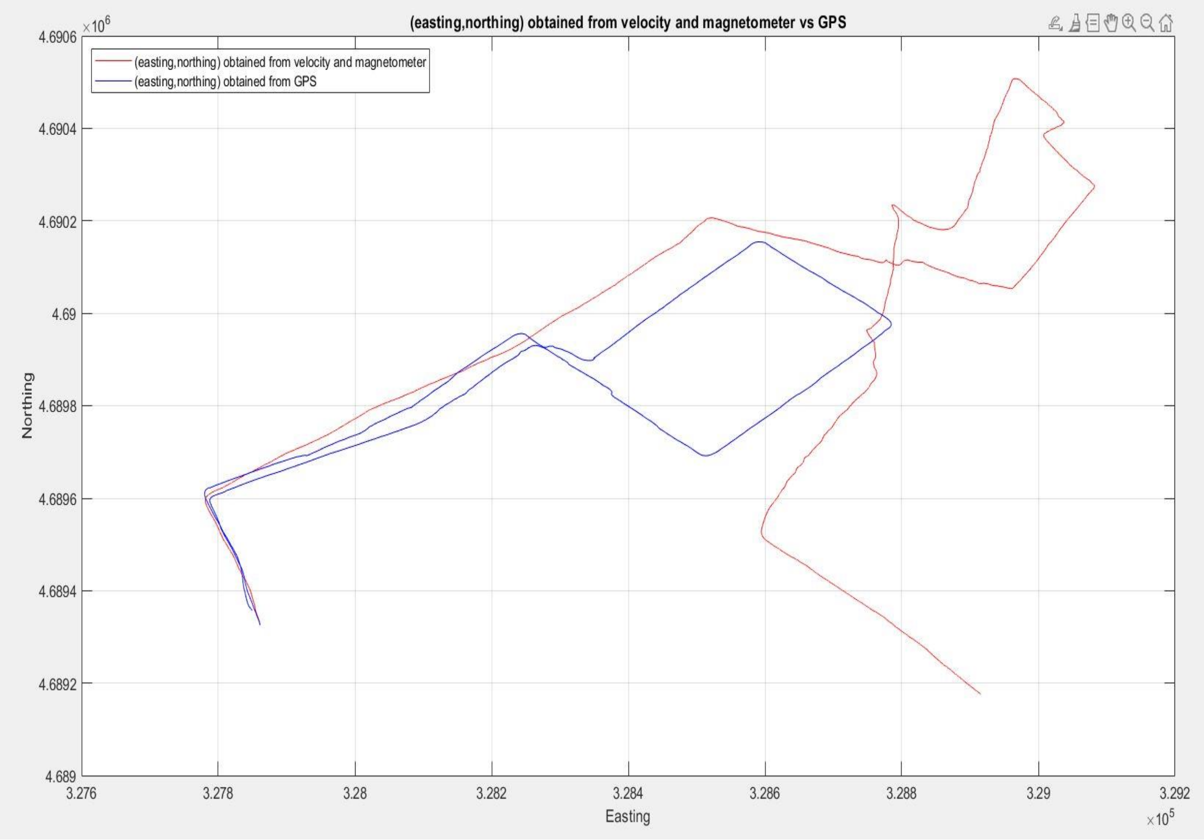


The observed acceleration Y has a bias present in the data due to which it is shifted below the acceleration obtained by multiplying the angular velocity with velocity in X $(\omega\dot{X})$. If we subtract the mean of stationary acceleration in Y then the bias is removed.

Also, there are sharp peaks in the $\omega\dot{X}$ plot. This is due to drifting of integrated data as we calculate $\dot{X}$ by integrating $\ddot{X}$. Due to this we are getting sharp peaks when the acceleration in y is not equal to 0. Even for very small values of acceleration in Y, the $\omega\dot{X}$ plot has sharp peaks which is due to drift in integrated data which when multiplied by gyro results in sharp peaks. If gyro is very small then the $\omega\dot{X}$ becomes small but if there is small increase in gyro, it gets multiplied with even larger $\dot{X}$ due to drifting it results in sharp peaks.

2. Dead Reckoning-

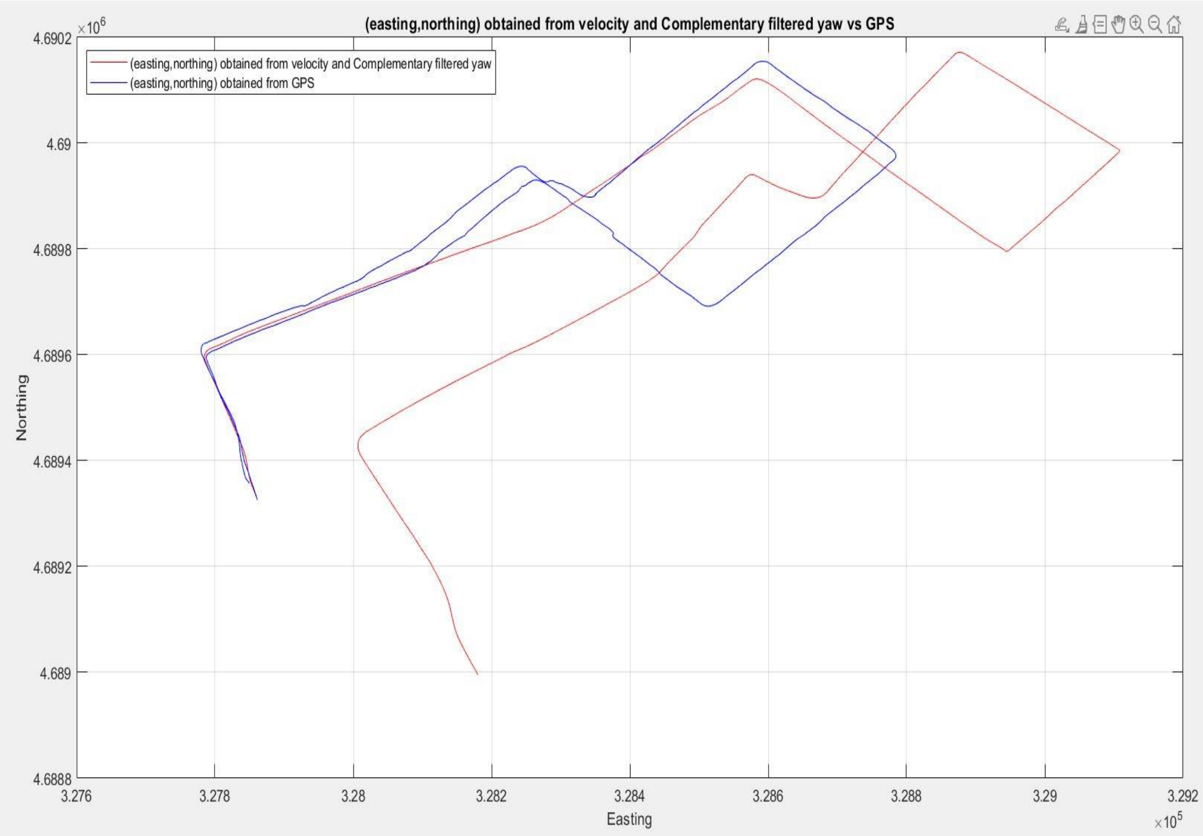
Below is the plot obtained from easting, northing obtained from magnetometer yaw and velocity data:



As we can see, if we just use the corrected magnetometer yaw and velocity to obtain easting and northing data, the plot drifts a lot and the orientation is also not the same for some regions. The magnetometer and velocity northing, easting plot is scaled by 1.4180733. Also, to adjust the first straight line, 1.73 rad is subtracted from the heading so that the first straight line is oriented the same as that of GPS plot.

A better plot would be if we use the complementary filtered yaw and velocity to calculate the easting and northing data.

Below is the plot for easting, northing obtained from complementary filtered yaw and velocity vs the easting, northing from GPS. Scaling factor is 1.41582 for this plot. Also, to adjust the first straight line, 1.45 rad is subtracted from the heading so that the first straight line is oriented the same as that of GPS plot. Now we can see that the orientation is very similar to that of the ground truth. Also, the drift in this plot is way less than the magnetometer yaw- velocity plot.



3. **Estimate x_c**

   The velocity of the inertial sensor is

$$v = V + \omega \times r,$$
$$r = (x_c, 0, 0)$$

using eqn $\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega}x_c$ and solving it in MATLAB

Estimated $\mathbf{x_c}$ **= 0.3463 m**