Navigation with IMU and Magnetometer

By- Uday Raghuvanshi
NUID-002102498

SUMMARY

In the given lab, a navigation stack was built using a GPS and IMU and their relative strengths and weaknesses were studied. Device drivers were used to read and parse the messages from GPS and IMU for orientation, magnetic field, linear acceleration and angular velocity in all the three axis, and utm easting and northing.
Data was collected in a car to perform dead reckoning. The IMU was mounted on car's dashboard with x-axis pointing forward and the GPS puck was mounted on the roof. Data collection was done in three phases: 1) When the car was stationary, 2) Driving in a circle 4 times in front of Ruggles and 3) Driving for a longer distance and coming back to the starting point to form a loop.

ANALYSIS

Magnetometer Calibration

The data collected while driving in circles was used to calibrate the magnetometer. The magnetometer readings are affected by the nearby magnetic objects and in order to get accurate heading estimate, it is important to mitigate these affects. Objects that produce a magnetic field create hard iron distortions and objects like metals distort the existing magnetic field and contribute to soft iron distortions.
To visualize the distortions, a 2D graph was plotted between magnetometer readings in x and y directions (circles data). Hard iron distortions cause a permanent bias in magnetic measurements and shift the center of circle from (0,0). Soft iron distortions distort and warp the shape of circle.
In the data collected, the hard iron distortion in x and y was -1.885×10^{-05} T and -2.42×10^{-06} T respectively. The scale factors in x and y were 0.96472 and 1.03795 respectively. Since the scale factors were approximately 1, it indicates that the data had negligible soft iron distortions. The distortions were then eliminated to obtain the corrected magnetometer readings.
Figure (1) shows the magnetometer data before and after the correction

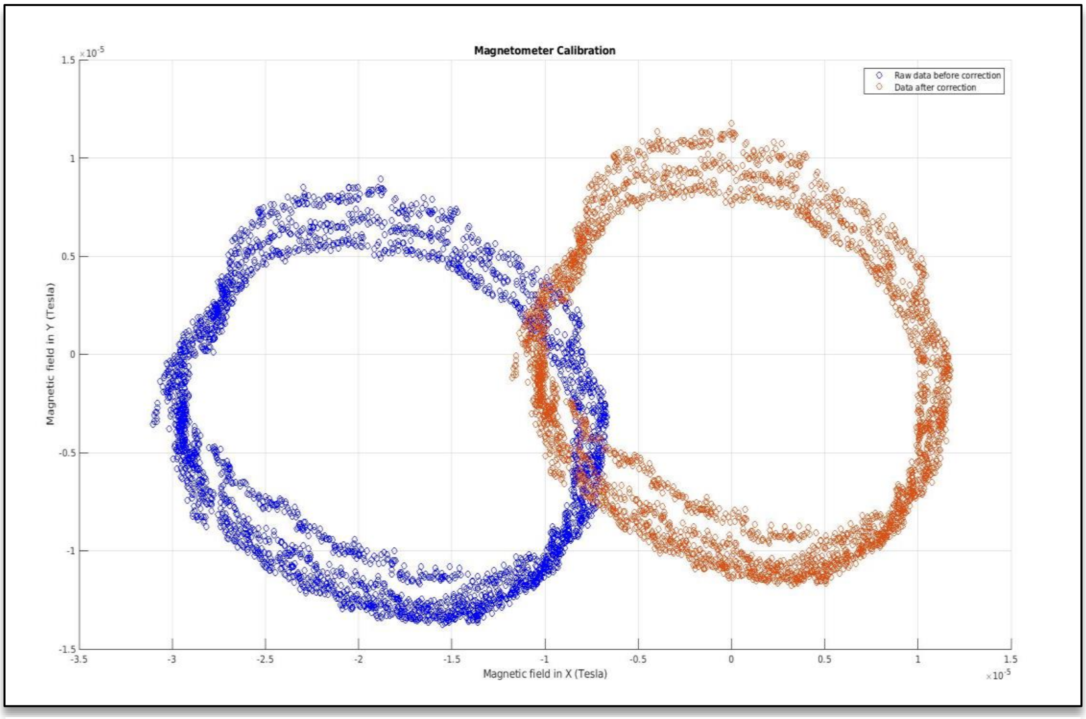

Figure 1

The magnetometer data for moving case was then corrected and yaw angle was calculated. Yaw angle from gyroscope was calculated by integrating the yaw rate in z direction. The integrated yaw from gyroscope has the tendency to drift over time but gives accurate results for shorter time. Yaw from magnetometer does not drift over time. So, for the complimentary filter, an alpha value of 0.985 was used (0.985*yaw-gyro + 0.015*yaw-magno). The filtered yaw is compared with yaw from magnetometer and gyroscope in figure 2. The bias from filtered yaw is removed and then compared with the yaw from imu (converting quaternions to euler angles) in figure 3.
From the figure the filtered yaw is better estimation of the yaw angle. The yaw angle obtained upon filtering is similar to yaw computed by imu.

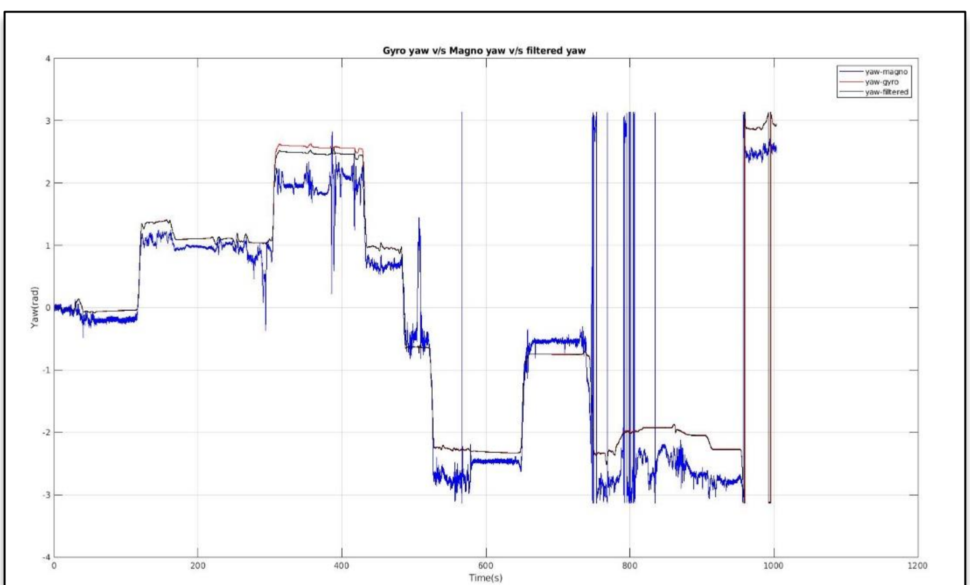

Figure 2

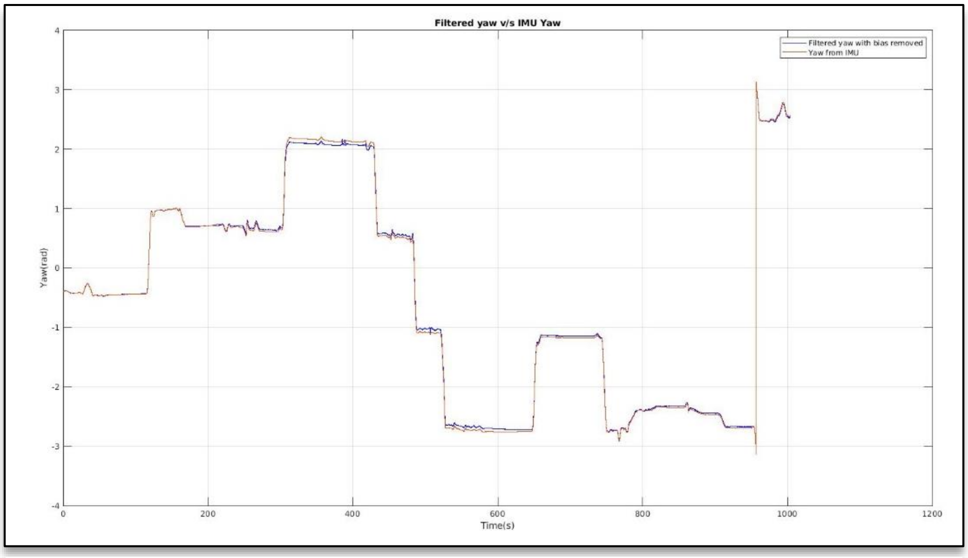

Figure 3

Estimating forward velocity

In this, the acceleration in x direction was integrated to get an estimated forward velocity. Also, the velocity was calculated from GPS measurements which was taken as the ground truth. Upon plotting velocity with time (figure 5), it is observed that velocity drifts overtime. It is because every time acceleration is integrated, it accumulates an error which eventually keeps on increasing with time. Thus, the integrated velocity doesn't make sense (shown in magenta in figure 5). It is important to make adjustments to acceleration to make velocity more reasonable.

First, a constant bias is removed from the collected data. The mean value of acceleration in x direction when the car was stationary was the bias that had to be subtracted from the moving data. The bias was not removed for some initial data points as the car was stationary.

Upon plotting acceleration in x direction with time, it can be seen that the points where car was stopped and acceleration should be 0, the acceleration obtained is not 0. So, an algorithm was written to filter the acceleration such that if the absolute value of acceleration obtained is less than 0.55 for 200 continuous points, then acceleration should be set to 0.

Figure 4 depicts the adjusted acceleration (red) and the points where the car stopped (blue).

In order to filter it more, an algorithm was written that made sure that for the points where the car stopped, the velocity should be put to zero. To do so, the adjusted velocity data was checked in batches of 50 and if the standard deviation of all those 50 points was 0, then the new velocity was put to 0. Also, the mean of all those 50 values was taken as an error that was subtracted from all the further readings until a new batch of 0 standard deviation was seen (the final adjusted velocity obtained is shown in blue in figure 5). The final velocity obtained has the least drift.

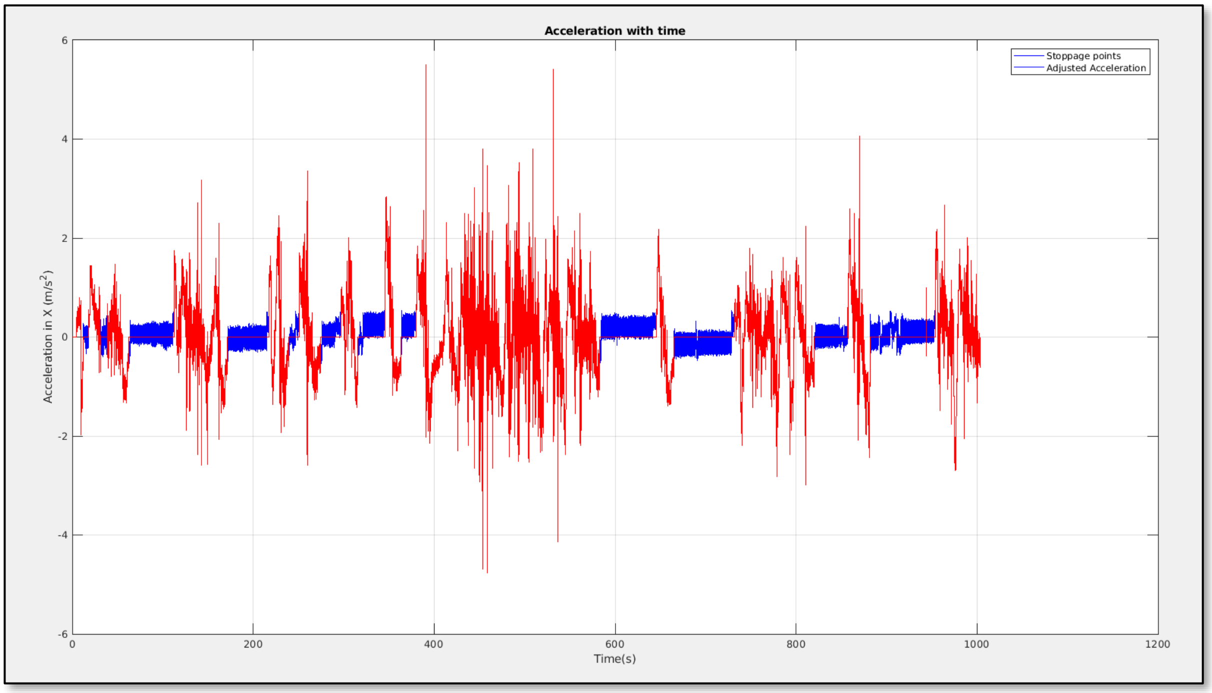

Figure 4

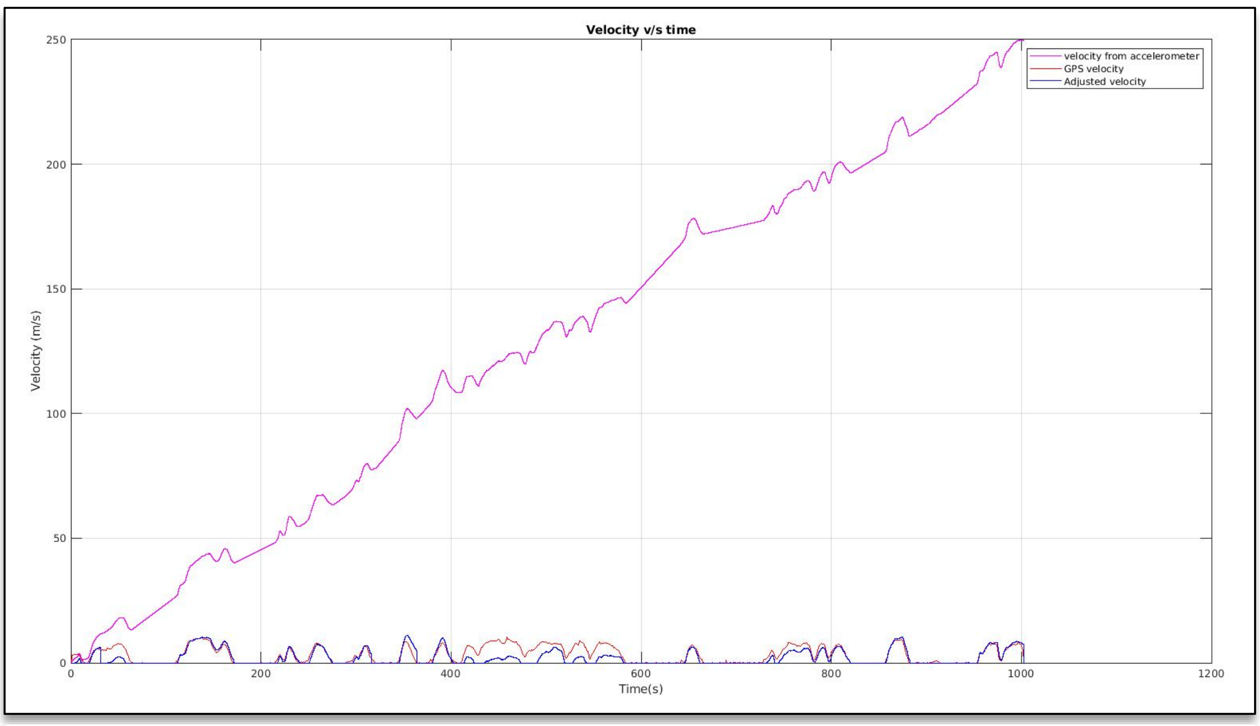

Figure 5

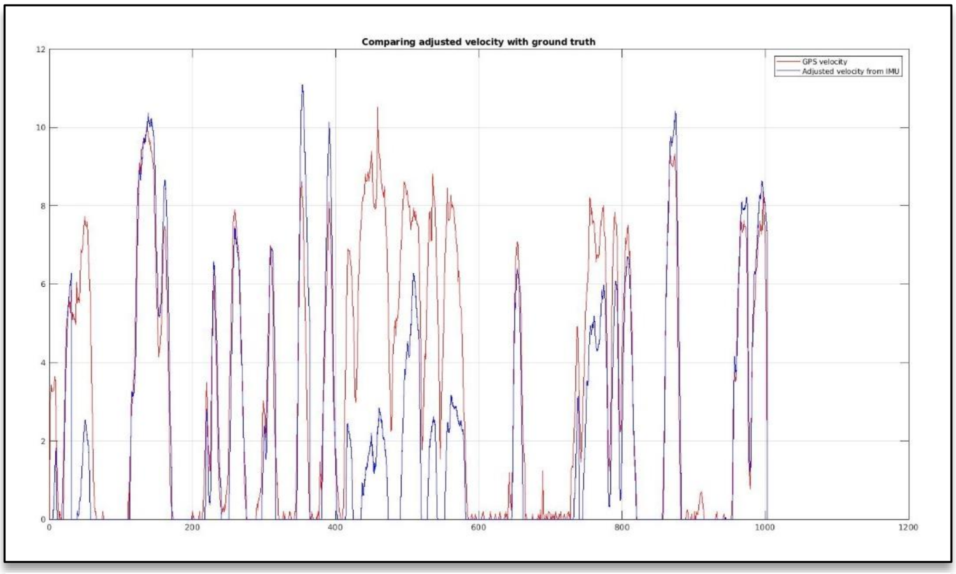

Figure 6

This figure has time on x axis and velocity on y axis.

Dead Reckoning with IMU

For the first part, acceleration in x obtained from imu is integrated to get X´. It is then multiplied by the angular rate in z direction and then plotted with the acceleration in y. Figure 6 shows the plot. There is a difference in the value obtained by multiplying X´ by gyro-z and y-acceleration because of the integration that is performed on x-acceleration. It causes error to accumulate. The peaks are sharper and larger for gyro-z*X'.

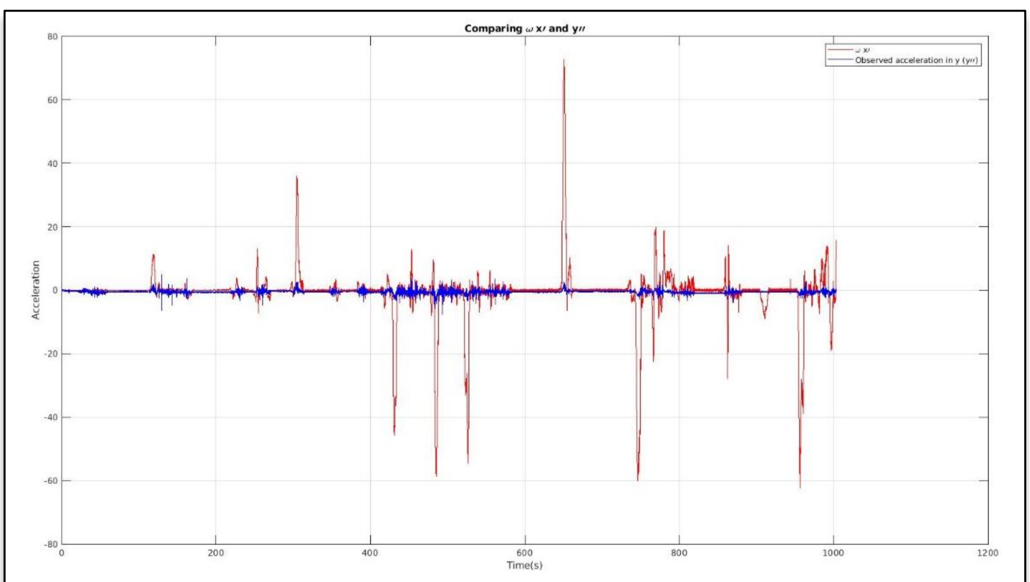

Figure 7

For the second part, the velocity obtained in after adjusting the x-acceleration is split into two components, v_e and v_n, using the Complimentary filtered yaw. v_e and v_n are then integrated to get x_e (easting from imu) and x_n (northing from imu). These are then plotted along with the utm_e and utm_n obtained from gps (ground truth). The initial point of estimated trajectory is shifted to (327861.339673587, 4689326.15538608) so that both tracks start at same point. The yaw is also rotated by 1.75 rad to align with the GPS track. The scaling factor used is 1.2 for plotting the trajectory obtained using filtered yaw.

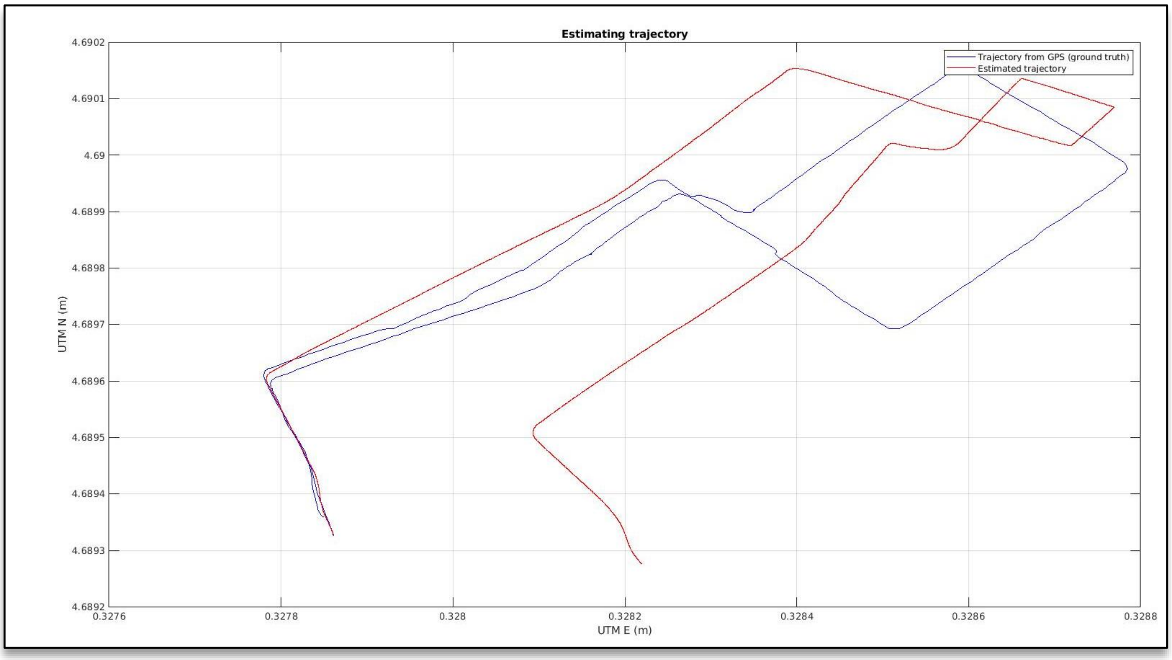

Figure 8

Trajectory has also been estimated using the yaw from magnetometer. It is shown in figure 9. Here the scale factor is 1 and yaw is rotated by 1.45. Clearly, this trajectory is not good as compared to gps trajectory since the yaw is not close to accurate. This, justifies using the complimentary filter to get yaw.

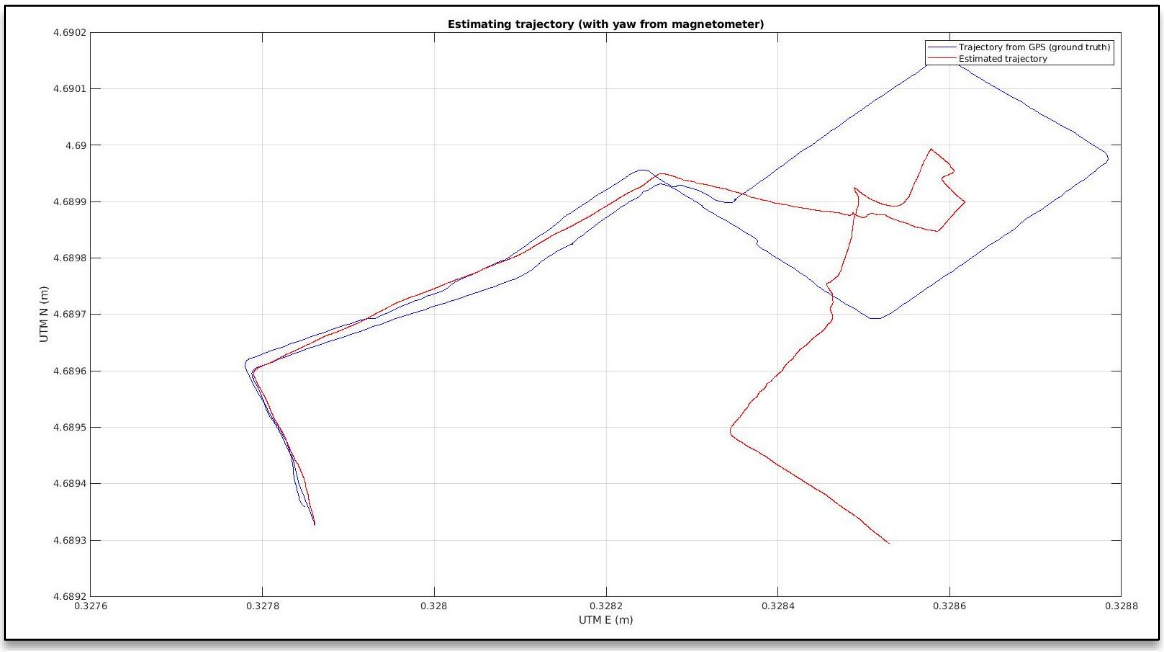

Figure 9

x_c is estimated by differentiating gyro_z and then using the linsolve function of MATLAB. The value obtained is 0.394m.

NOTE:
I estimated the trajectory using a different method before. However, on the last day, after TA hours, I tried making the velocity plot more reasonable than before. I was able to do that but the velocity for first few seconds was not coming out as expected. Due to this, my estimated trajectory became worse than I did before. I changed my code, so it was very difficult to get back to the same point and I ran out of time. Below is my previous estimated trajectory:

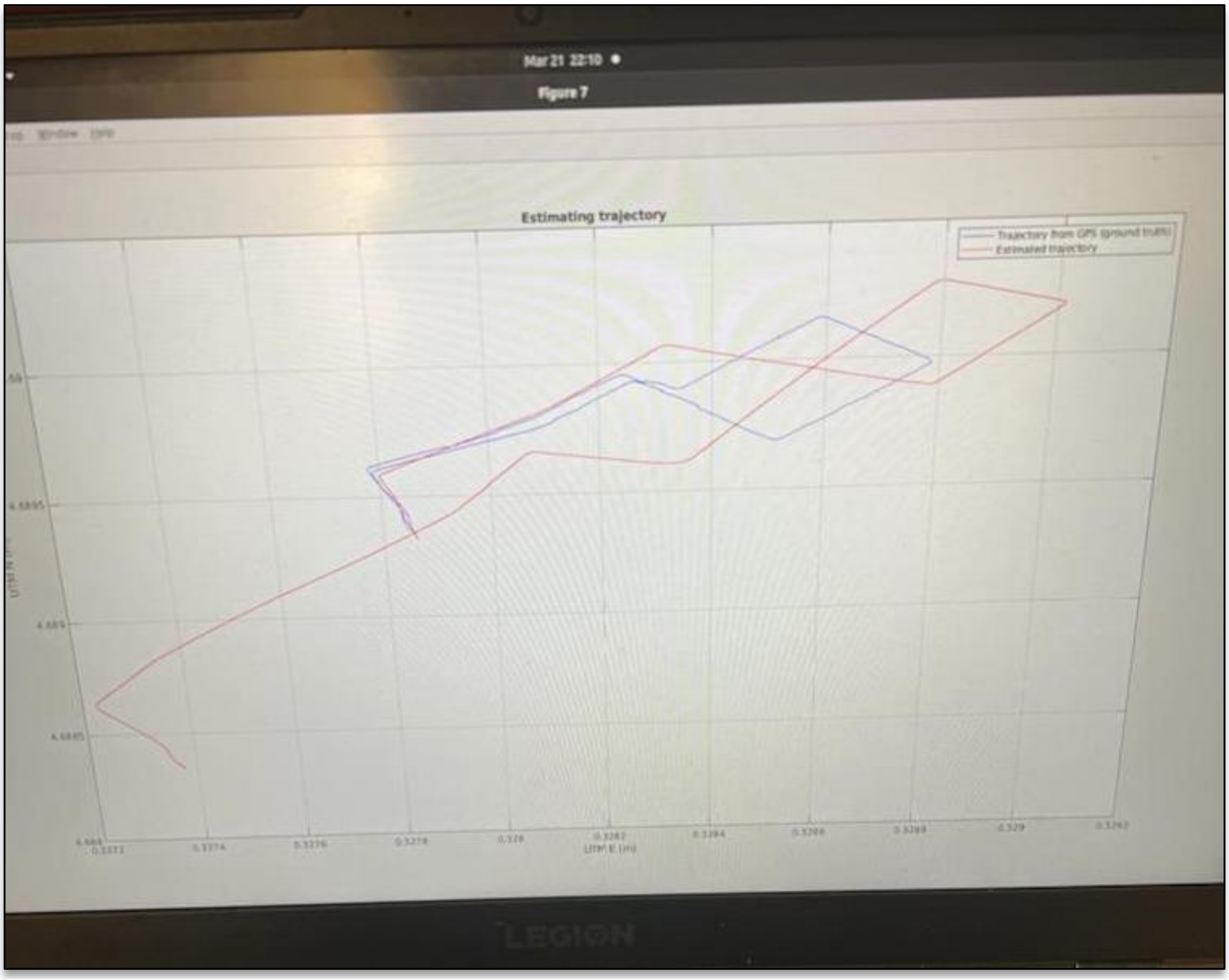

Figure 10