LAB II

EECE 554 Robotics Sensing and Navigation

February 13 2020

Andac Demir
001826062

1 Stationary Points

1.1 Acceleration

IMU sensor has a sampling frequency of 40 Hz. In the first section of the experiment, we mounted an IMU sensor to a desk at a stationary spot isolated from electronic devices or magnetic materials and collected data for 3 mins. We reported in Fig. 1 acceleration, in Fig. 2 angular rate and in Fig. 3 magnetometer measurements. We use a MATLAB built-in function named **lillietest** to identify the noise characteristics. **Noise in linear acceleration and magnetometer measurements are not Gaussian distributed, whereas noise in angular velocity is Gaussian distributed**. The reason that magnetometer noise is not Gaussian is obvious, provided that there're many magnetic materials and electronic devices surrounding us that have a big impact on the magnetic field.

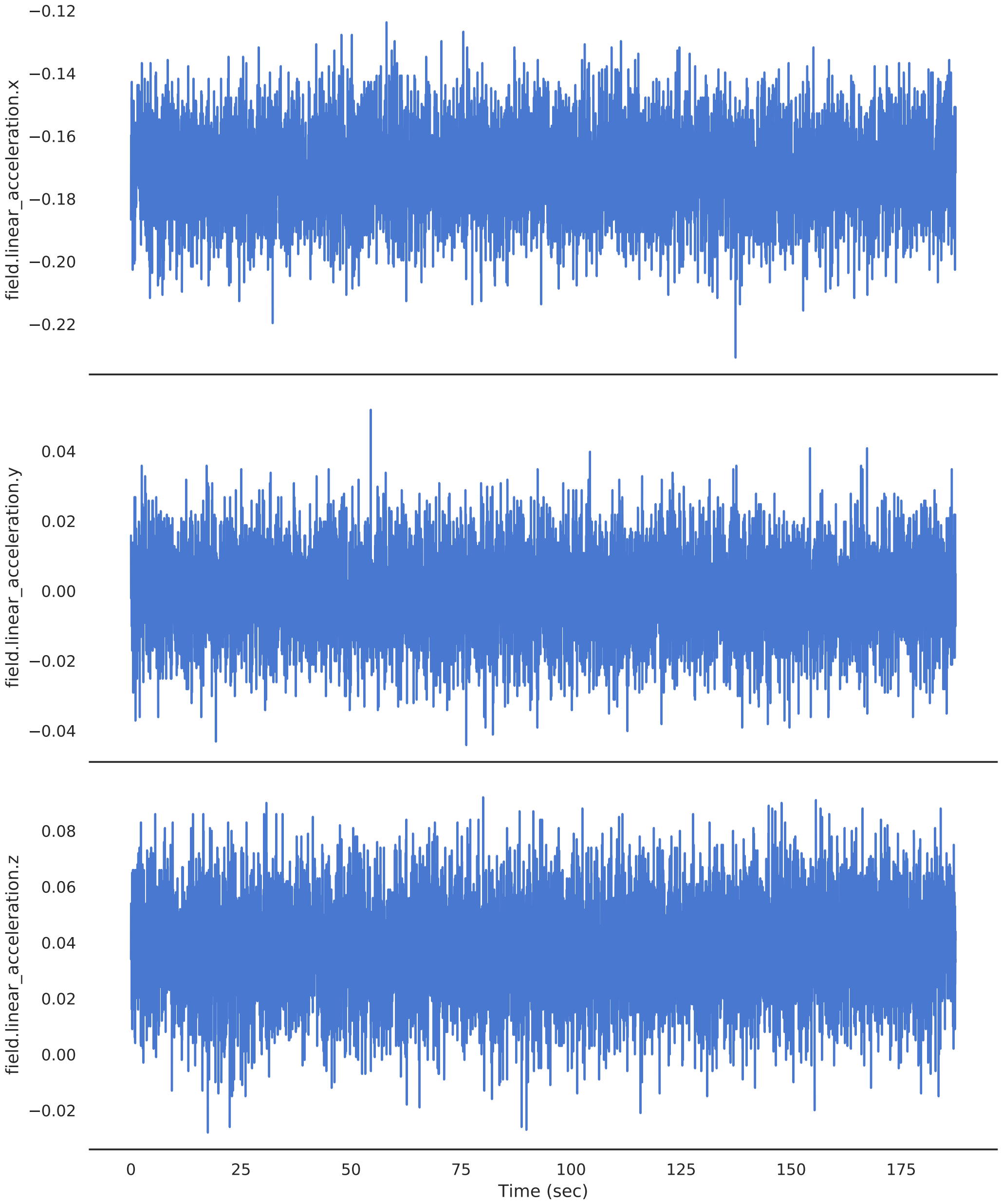

Figure 1: Measurement of acceleration in stationary data from x, y, and z dimensions.

1.2 Angular Rate

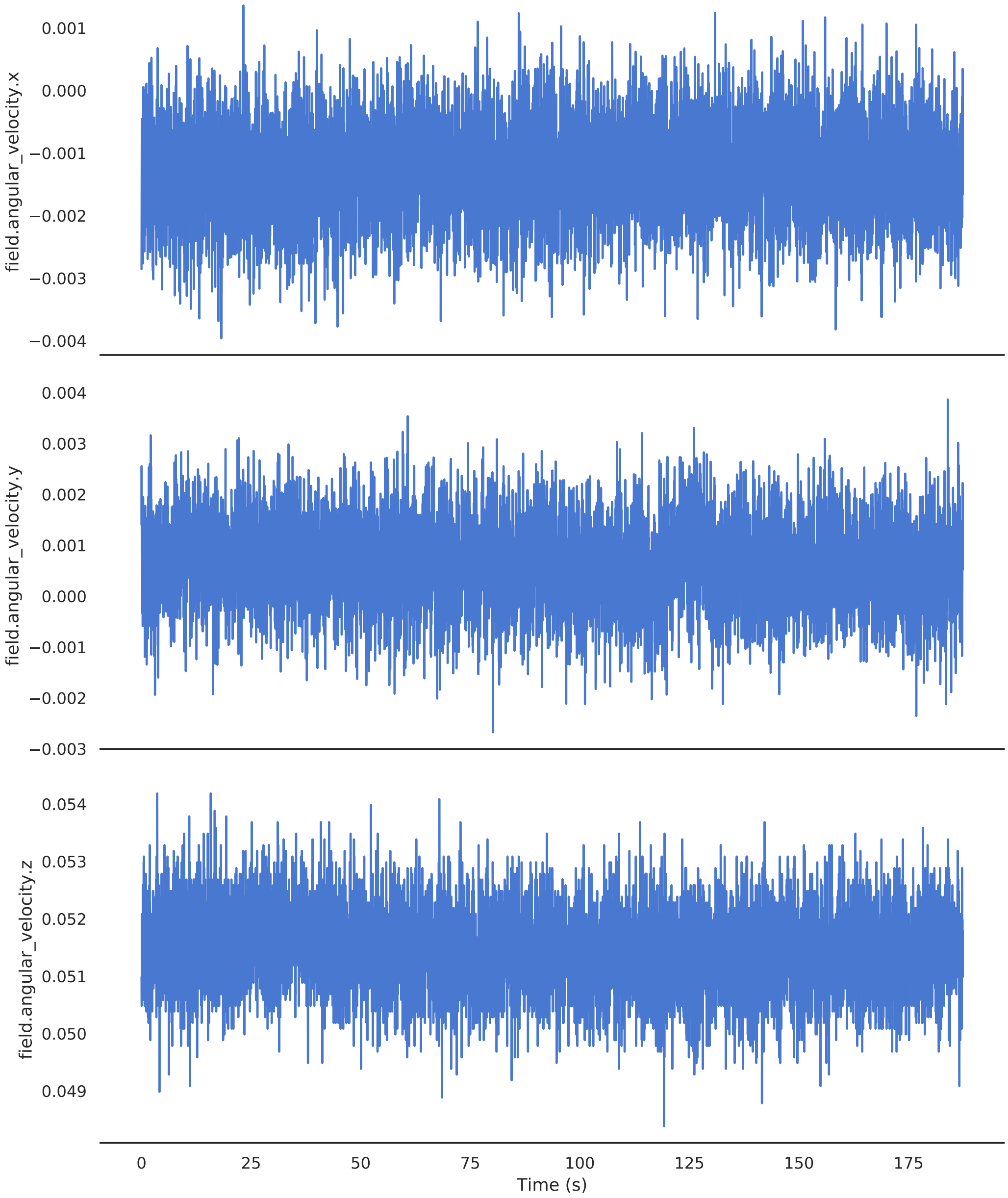

Figure 2: Measurement of angular rate in stationary data from x, y, and z dimensions.

1.3 Magnetometers

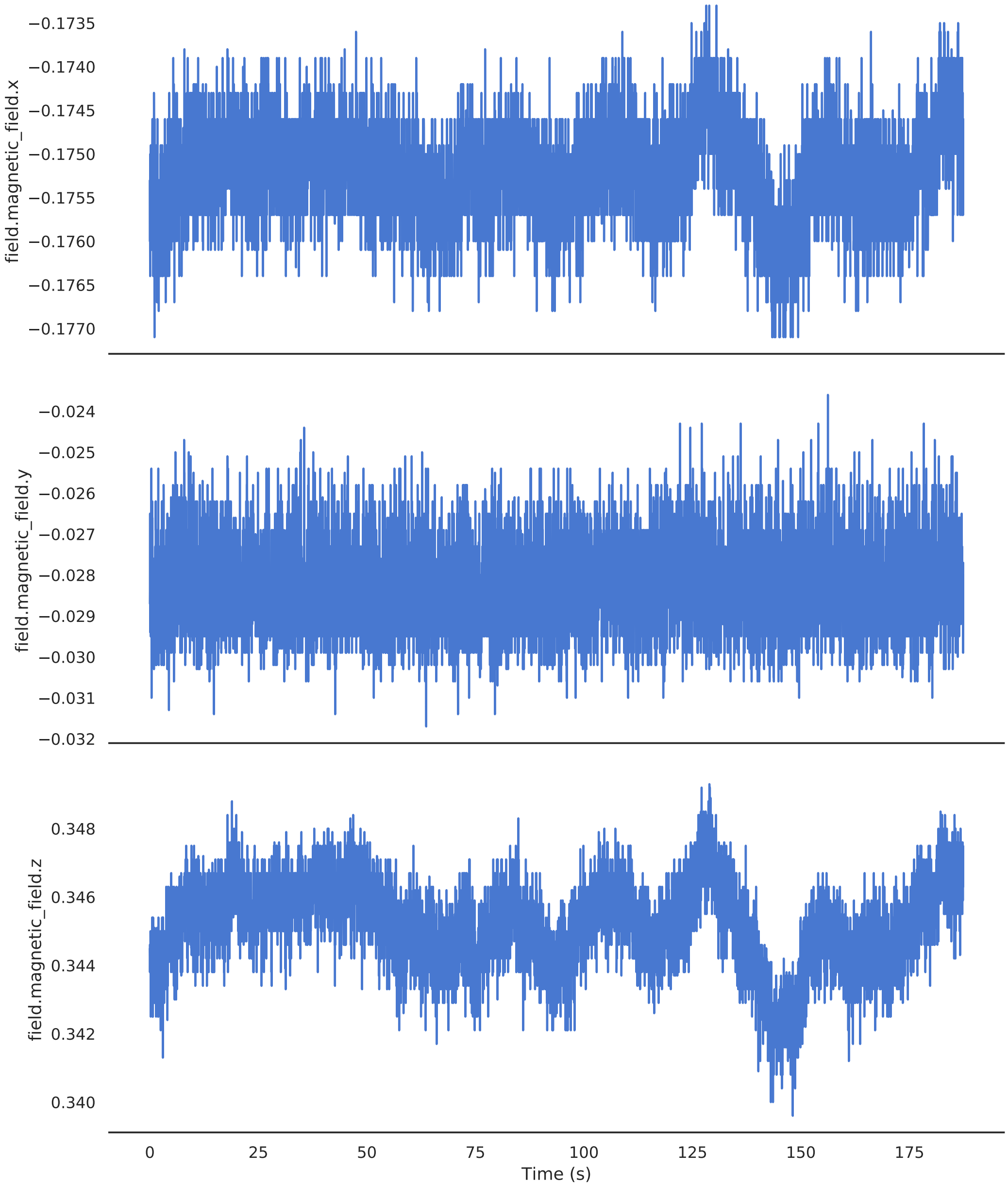

Figure 3: Measurement of magnetometer in stationary data from x, y, and z dimensions.

2 Heading Estimation

There is hard ironing and soft ironing effect that distorts the readings of a magnetometer. Basically, hard ironing effect is produced by materials that exhibit a constant, additive field to the earth's magnetic field. Hence, they generate a constant additive noise to the output of each of the magnetometer axes, such as a speaker magnet. On the other hand, soft ironing effect is produced as the result of materials that distorts a magnetic field of the Earth, but does not necessarily generate an additive noise.

We ride 3 circles with a car to calibrate our magnetometer measurements. In the first stage, we take the average of max. and min. values of magnetic field separately in x,y, and z dimensions to calculate the biases. Then, to correct the hard ironing effect, we subtract these biases from magnetometer measurements. Then we correct the soft ironing effect on ellipsoid centered at the origin. To find the radius of the major axis, we calculate the magnitude of each data point and then identify the maximum of these computed values. To find the radius of the minor axis, we calculate the magnitude of each data point and then identify the minimum of these computed values. Using this information, we formulate a rotation matrix and apply it to every measurement in the point cloud. After rotation, the major axis of the ellipsoid stands on the x-axis and minor axis stands on y-axis of the Cartesian plane.

Subsequent to rotation, we rescale every data point so ellipsoid is converted to a circle. We learn the rotation matrix and scaling coefficient from data collected through circular rides, and then apply it on the data collected by riding around Northeastern U. campus.

2.1 Hard and Soft Iron Corrections

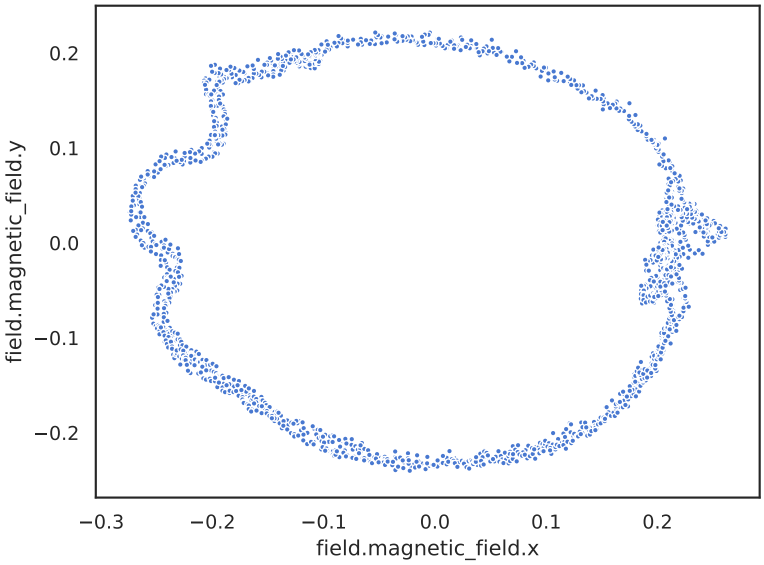

Figure 4: Uncalibrated magnetometer measurements from x and y dimensions from a car riding around a fixed circle plotted on an xy-plane.

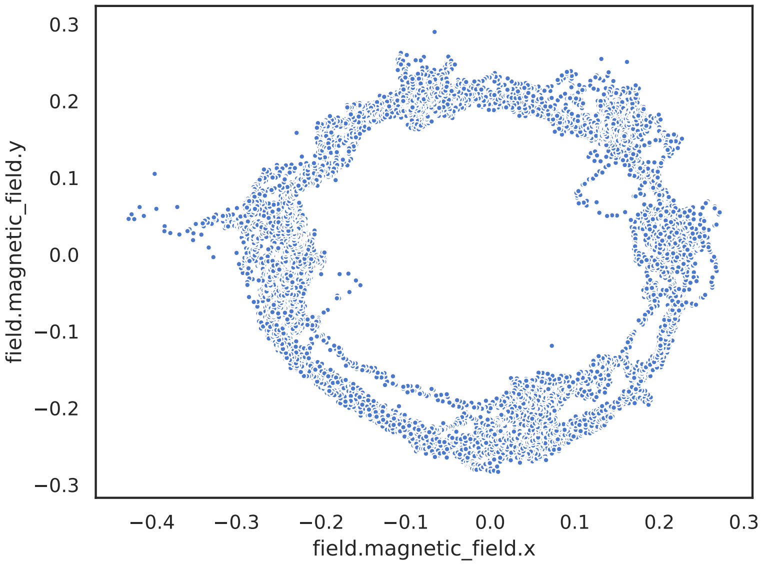

Figure 5: Uncalibrated magnetometer measurements from x and y dimensions from a car riding around Northeastern U. campus plotted on an xy-plane.

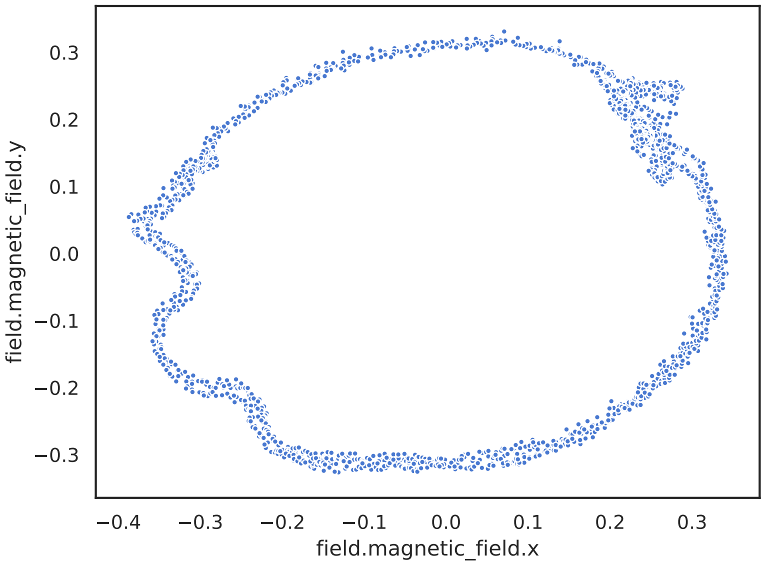

Figure 6: Corrected magnetometer measurements from x and y dimensions from a car riding around a circle.

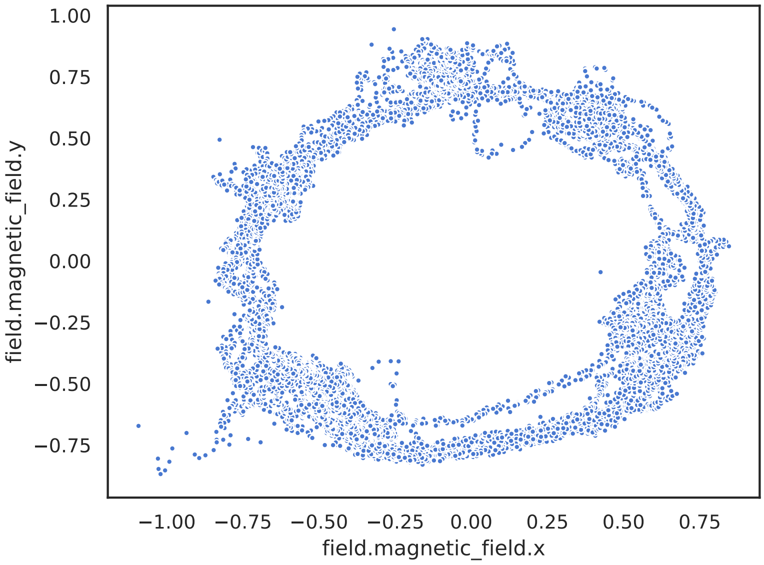

Figure 7: Corrected magnetometer measurements from x and y dimensions from a riding around Northeastern U. campus .

2.2 Yaw Angle Calculation

We calculate yaw angle in two ways. First, we integrate yaw rate of the gyro across time, and second we calculate yaw angle from calibrated magnetometer readings. The result is illustrated in Fig. 8. To cancel out gyro drift and jitter in magnetometer's yaw estimation, we fuse the information from the IMU and magnetometer with the use of a complementary filter. Another approach to solve this problem would be a Kalman filter, but it's more complicated and eats more resources. We use a simple FIR filter of order 1 by,

$$fusedData = weight * gyroData + (1 - weight) * magnetometerData \tag{1}$$

We read in the literature that gyro data is a lot more reliable, and weighted by 0.99 in a few applications. Hence, we also used 0.99 as the weight factor. The yaw angle estimation with a complementary filter is demonstrated in Fig. 10. Additionally, in Fig. 9, we demonstrate the difference between yaw angle estimations of gyro and magnetometer.

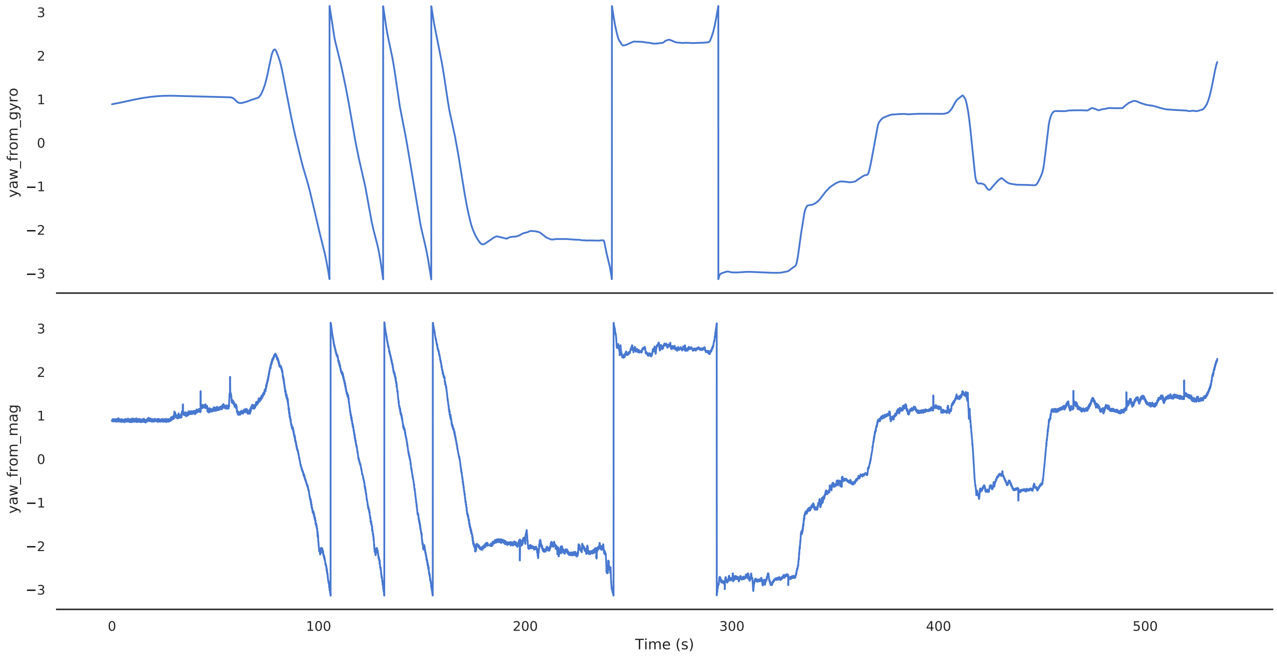

Figure 8: Comparison of yaw angle ($\pm\pi$) calculated with the gyro and the magnetometer readings.

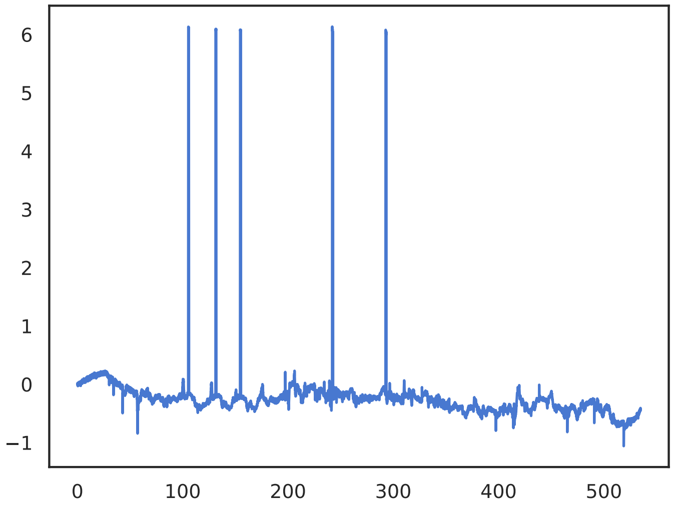

Figure 9: Difference between yaw angle estimations.

2.3 Yaw Angle Calculation by Sensor Fusion

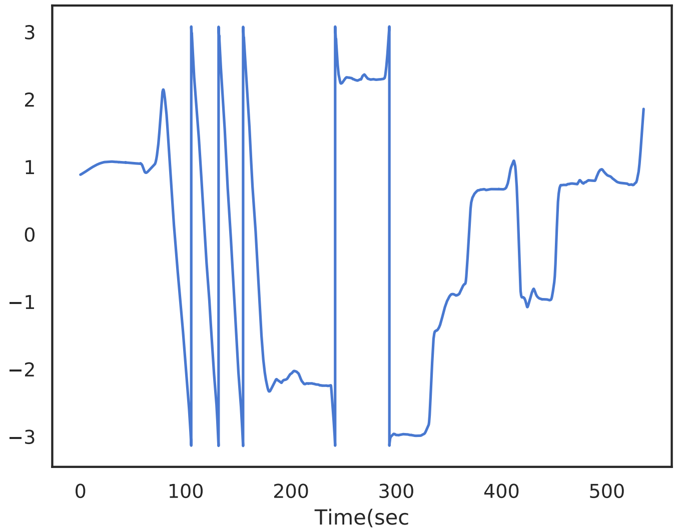

Figure 10: Yaw angle calculation by the fusion of gyro and magnetometer sensors.

3 Estimation of Forward Velocity

From the linear acceleration measurements of the moving car, we subtract the average of the acceleration bias of the car riding around circles and acceleration bias of the car riding in the campus. Then, we shift the forward velocity estimates of IMU upward in a way that makes the lowest IMU velocity estimate 0. On the other hand, velocity estimates of GPS were calculated by multiplying change in root mean square of UTM East and North with Δt.

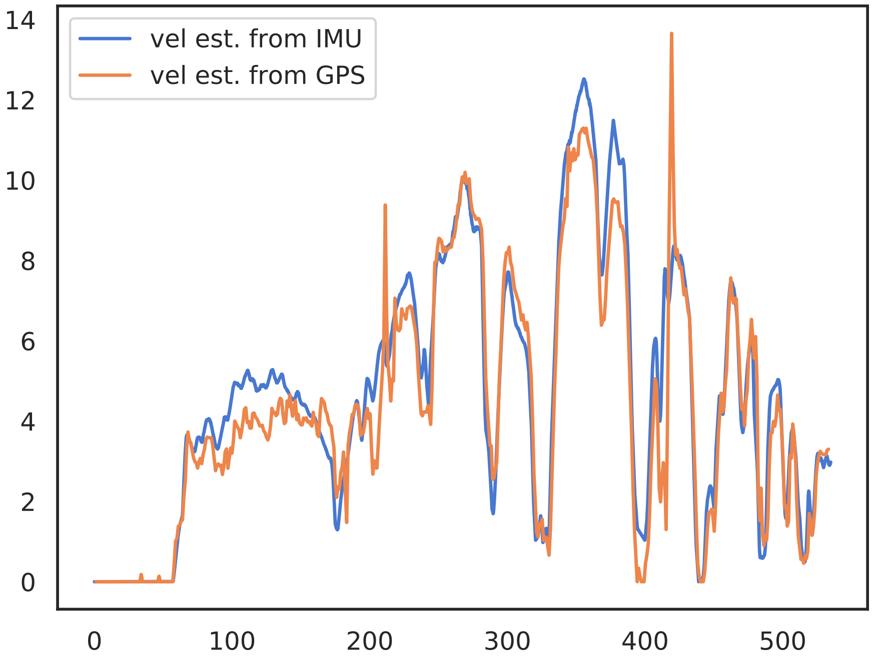

Figure 11: Velocity estimates by GPS (ground truth) and accelerometer in (m/s) across time (s).

4 Estimation of Displacement

4.1 Estimation of Track

Dead reckoning is the process of calculating robot's current position by using a previously determined position with the use of IMU, without any data from a GPS navigation system or wireless signal transmission. It is particularly useful in the case of GPS outage of a plane or autonomous vehicle and underwater navigation.

We decompose the forward velocity estimated by IMU in the previous section for dead reckoning into x and y components. For decomposition of the velocity vector, we use the yaw angle previously computed by sensor fusion and declination angle of the Earth's magnetic field reported for Boston area.

Displacement plotted by GPS in UTM East and North are our ground truth (see Fig. 13). We realize that between $\sim 300 - 350$ seconds, there is some distortion in the vehicle's forward velocity estimated by the IMU. This causes a deviation from the right track as plotted in the UTM North vs UTM East figure at around (-150m, 100m). This might be the result of some stationary acceleration values collected at the red light or some miscalculation in the yaw angle. See Fig. **??** for linear acceleration.

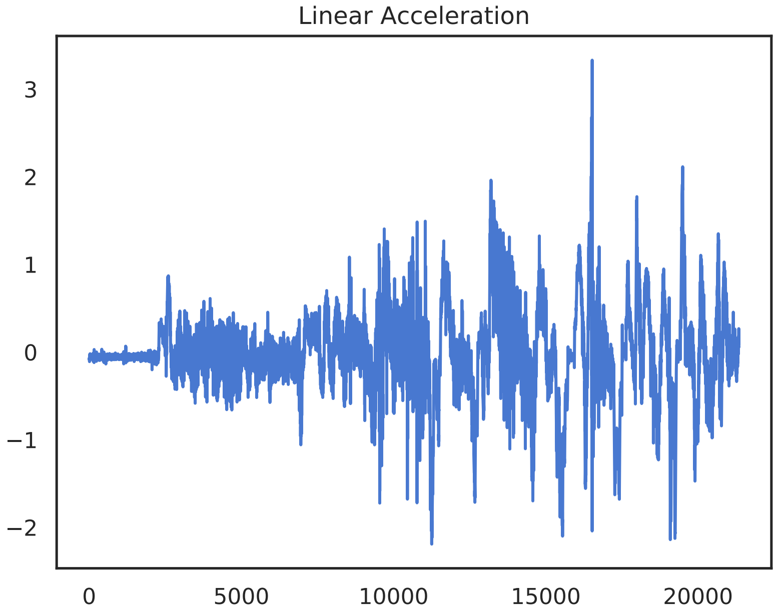

Figure 12: Forward linear acceleration measurements (m/s^2) across time (s).

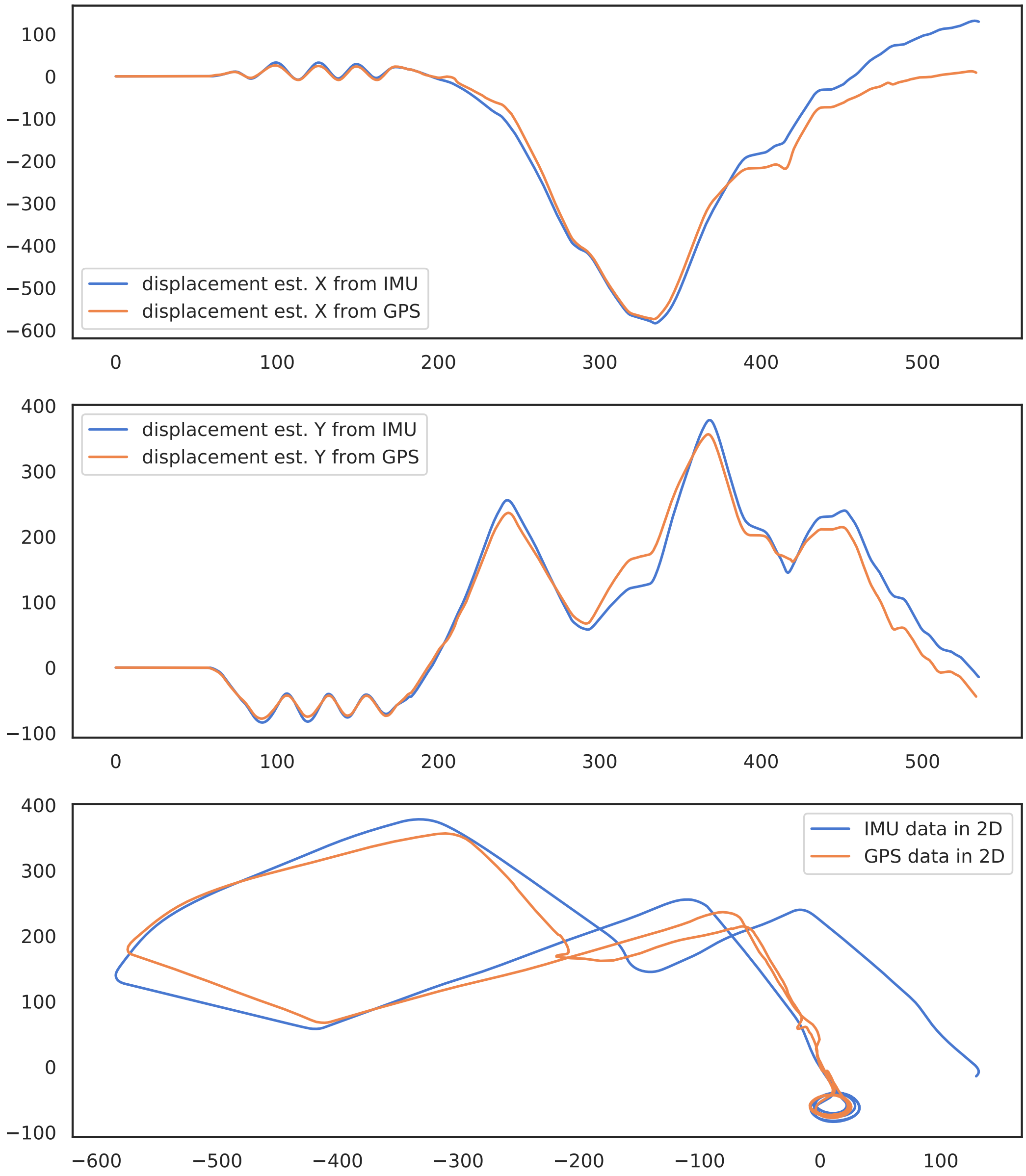

Figure 13: Displacement estimates by GPS (ground truth) and accelerometer. First subplot in UTM East direction (m) vs time (s). Second subplot in UTM North direction (m) vs time (s). Third subplot in UTM North direction (m) vs UTM North direction (m).

4.2 Computation of $w\dot{X}$ and $\ddot{y}$

We realize that $\ddot{y}$ has so much jitter and $w\ddot{X}$ is like the moving average $\ddot{y}$ see Fig. 14. Both of them are cluttered around 0, and this is consistent with the real world situation that the vehicle is not skidding sideways. Besides, we see a lot of spikes in the figure. This is because sensor is very sensitive to many vibrations of a mobile car. Although we didn't experience any sideways skidding in our experiment, these vibrations can aggravate $\ddot{y}$. Since $w\dot{X}$ has less noise, we look at Fig. 15 to better characterize sideways skidding. In particular, the spikes in this figure correspond to turn arounds, and lane changes. We also realize that positive spikes correspond to right turns and negative spikes correspond to left turns.

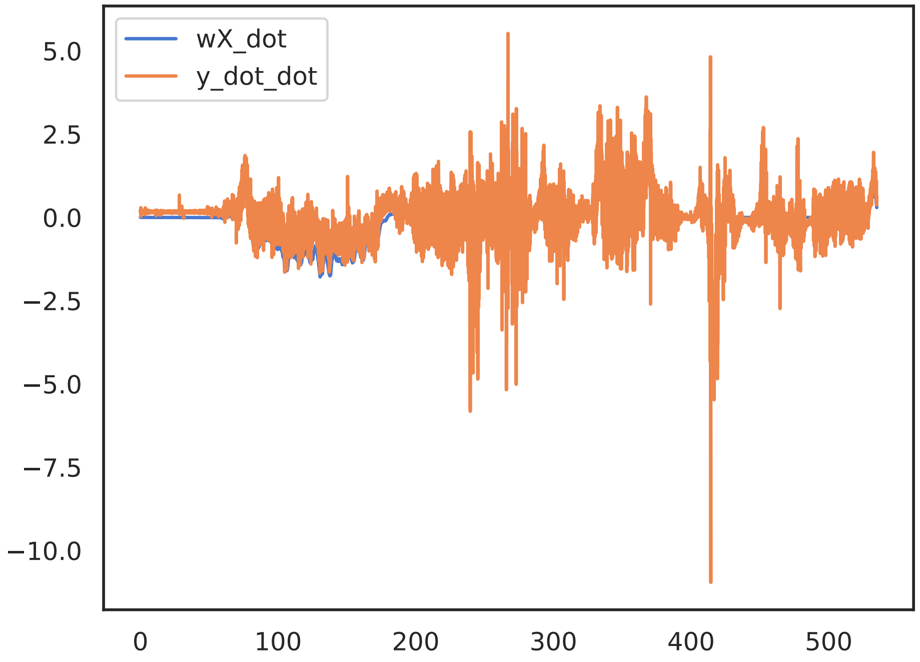

Figure 14: $w\dot{X}$ and $\ddot{y}$ in m/s^2 vs. time (s)

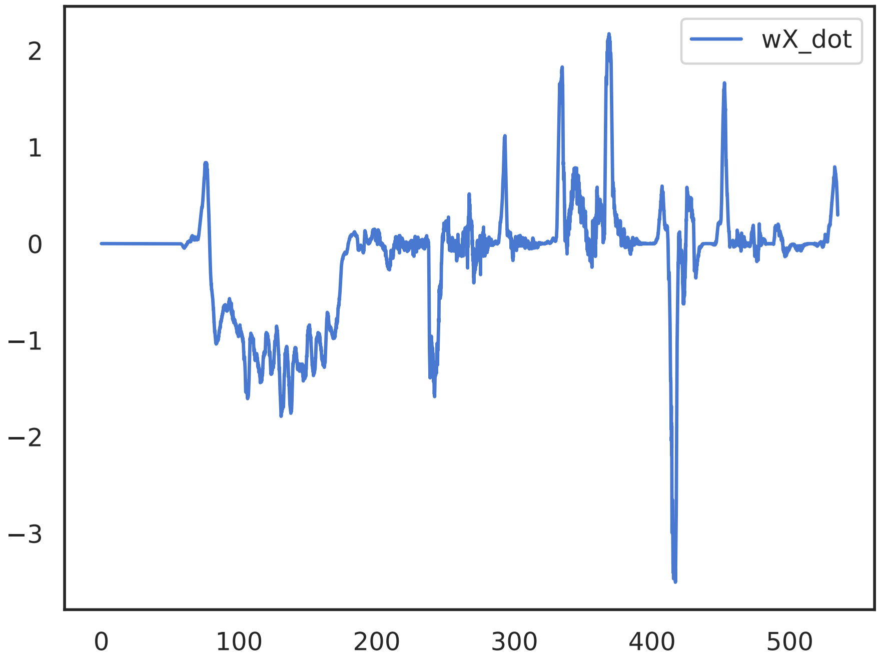

Figure 15: $w\dot{X}$ in m/s^2 vs. time (s)

4.3 Estimation of x_c