NORTHEASTERN UNIVERSITY

LAB4 Navigation with IMU and Magnetometer

AVISH KALPESH PATEL
Group 11
EECE 5554 – Robotics Sensing and Navigation
Date: 4/2/23

Abstract

There are multiple sensors available to solve the navigation task. This Lab explores the use of VN-100 Inertial Measurement Unit to perform navigation using accelerometer, gyroscope, and magnetometer data. Moreover, the BU-353S4 GPS receiver is used to gather position data. To perform dead reckoning the IMU data is manipulated to derive heading angle and velocity to estimate the motions trajectory of a drive around city of Boston. The presence of different sensor error/bias and their compensation is discussed in the report.

Table of Contents

Heading (YAW) Estimation

In the following section we experiment with estimation of the heading angle (Yaw) using different sensor such as magnetometer and gyroscope data.

Magnetometer Calibration

External magnetic influences classified as hard or soft iron effects leads to distortions of earth's magnetic field and thereby generating an error in magnetometer readings.

Hard Iron Calibration

Such distortions are generally caused by objects that produce magnetic field which acts as a constant addition to the earth's magnetic field, eventually leading to a permanent bias/offset.

To compensate for this Hard Iron effect, we need to first estimate the center of the current magnetometer data that is simply the average of the maximum and minimum values in both directions, as follows:

$$Cx = \frac{x_{max} + x_{min}}{2}$$

$$Cy = \frac{y_{max} + y_{min}}{2}$$

Then we subtract (C_X,C_Y) from each data point (x_{mag},y_{mag}) to translate the data to origin as the new center. The calibrated data is shown in the plot below.

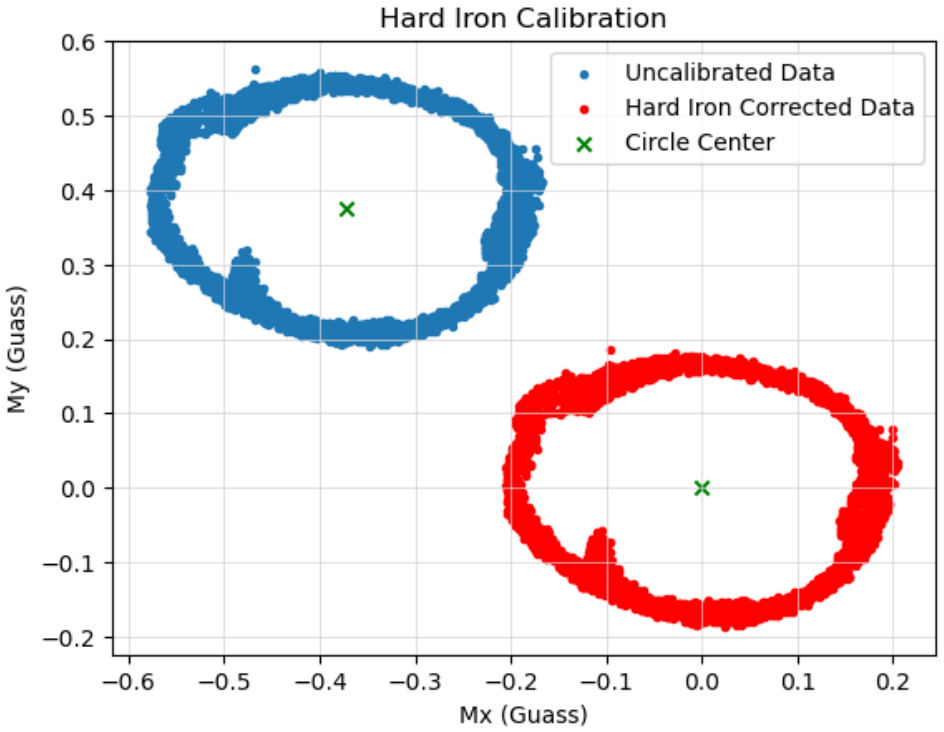

Figure 1: Hard Iron Calibration Results

Soft Iron Calibration

Such distortions are result of objects near sensors that not necessarily produce their own magnetic field yet are able to alter the magnetic field lines, for example Iron/Nickel. It can be seen as stretching or wrapping of the data point locations.

Unlike Hard Iron, Soft Iron effect is dependent on orientation of objects with respect to magnetic field and sensor, therefore compensating it is a more complex process as discussed further.

To compensate for Soft Iron distortion, we need to rotate the data and then scale the data to fit a circle instead of an ellipse. First, we estimate the angle to be rotated to align Major Axis of the ellipse to X axis and Minor to Y axis. We start with finding the distance of each point from the center and take the point with the maximum distance. This will be the radius along the major axis. Similarly, we can get the radius along Minor axis as the point with minimum value. Now that we have the end point on the major axis and the distance from the origin, we can obtain the rotation from the x axis as follows: [1]

$$r = \max\left[\sqrt{x^2 + y^2}\right]$$

$$\theta = sin^{-1}\left(\frac{y}{r}\right)$$

Now using the theta obtain we can rotate the data points:[1]

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

$$Rotated\ data = R \begin{bmatrix} x \\ y \end{bmatrix}$$

Note: [x, y] are data points after Hard Iron Calibration

Now that our axes of ellipse have aligned with the coordinate frame axes, we can scale the data to transform it to a circle: [1]

$$\sigma = \frac{a}{b} = \frac{major\ axis\ radius}{minor\ axis\ radius}$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix}$$

$$Calibrated = S * Rotated\ Data$$

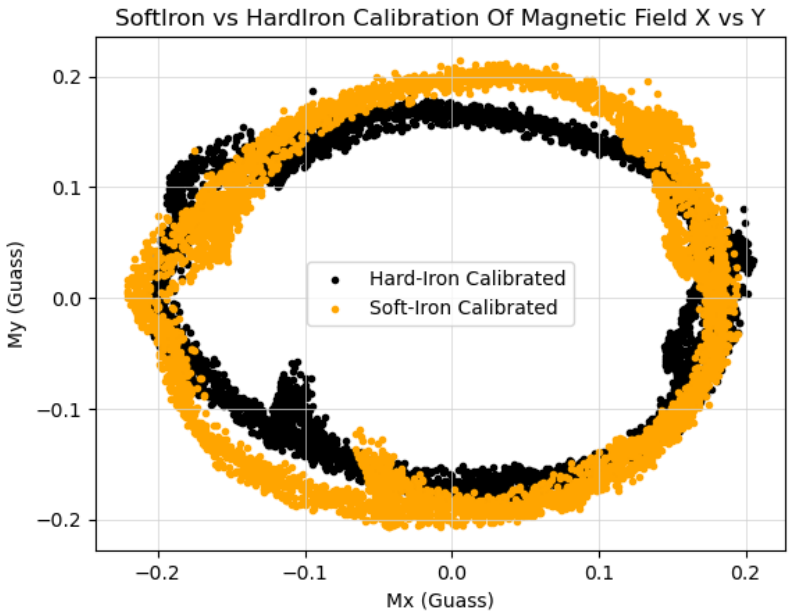

Figure 2: Soft Iron Calibration Results after Hard Iron Calibration

From the previous plot we can see the effects of Soft Iron Calibration. The Data resembles more of a circle than ellipse. However, the difference is not quite visible as the major and minor axis were 0.2 and 0.18 respectively. Therefore, the scaling factor sigma was small. The results of uncalibrated magnetometer data vs calibrated data are as follows:

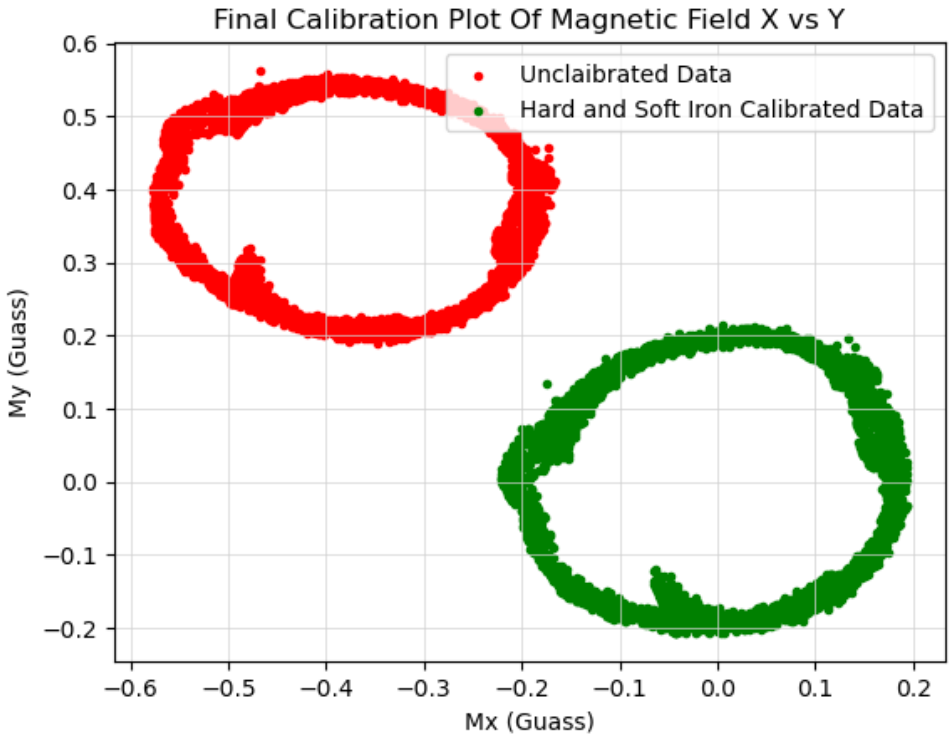

Figure 3: Uncalibrated vs Calibrated Magnetometer Data

Sources of Error

It is important to note that the magnetometer was placed inside the car and therefore due to a closed car environment its readings degrade compared to keeping it in outside. Moreover, we saw we had both Hard and Soft Iron distortions in our data. These Errors can be due to:

- Internal Magnetic Disturbances: Presence of different sensors and electric equipment in Nuance car that are always fixed and are present with the IMU sensor. Moreover, Iron components in vehicles body frame can lead to soft iron effect.
- External Magnetic Disturbances: Other vehicles (cars). Not sure about the extent but could be result of people walking by and carrying multiple electrical devices of various nature.
- Time-varying Disturbances: Power Electronics of Nuance as it is Hybrid car that vary in working intensity with time.

Sensor Fusion

Magnetometer Yaw Estimation

Furthermore, we can apply the calibration achieved in previous step to the driving data set and calibrate the magnetometer data. Then we can derive the yaw angle as follow:

$$Yaw(mag) = tan^{-1}\left(\frac{M_y}{Mx}\right)$$

The results of calibrated and uncalibrated magnetometer derived yaw are shown below:

Note: The data from magnetometer yaw is unwrapped using numpy.unwrap() to compensate for phase shifts in measurements.

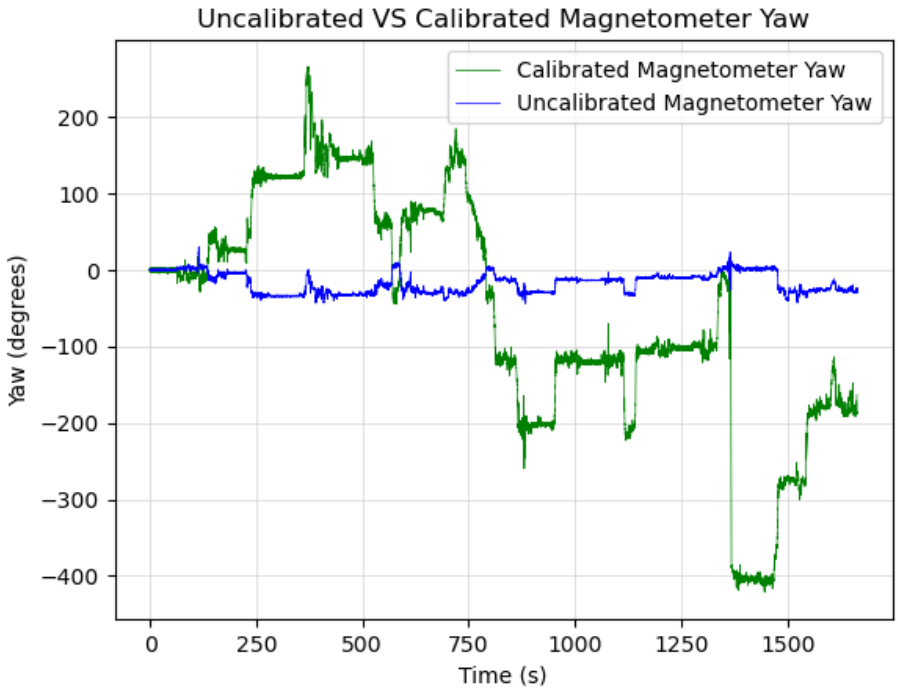

Figure 4: Yaw Estimate Magnetometer: Calibrated vs Uncalibrated Magnetometer Data.

Complementary Filtering

- While driving in city leads to more exposure to multiple cars on the road, or passing by a large iron structure, etc. these can further degrade the magnetometer data which was not compensated for with the calibration as the calibration was based of driving around Ruggles circle with different environment conditions. Hence, the effect of external magnetic distortions varies in both the cases.
- Moreover, the presence of the different kind of magnetometer errors (discussed in previous section) and as seen from the figure 4, effects of these error leading to large changes in amplitude of the steps, it is better to filter out such high frequency errors we using a low-pass for magnetometer readings.
- We can integrate the angular rate in z direction ($gyro_Z$) to get the Yaw estimate. However, Gyroscope drifts with time and hence to remove such bias we can use a high-pass filter.

Thus, to compensate for the above points we can use a Complementary filter to get better estimates. The digital implementation of complementary filter is as follows:

$$CF = (\alpha)[YAW_{gyroscope}] + (1-\alpha)[YAW_{magnetometer}]$$

From the previous equation we can see that we are implement a high-pass filter to the gyroscope estimate and a low-pass filter to magnetometer estimate with the frequency (α=0.8) in s-domain. The obtained filtered results are as follows:

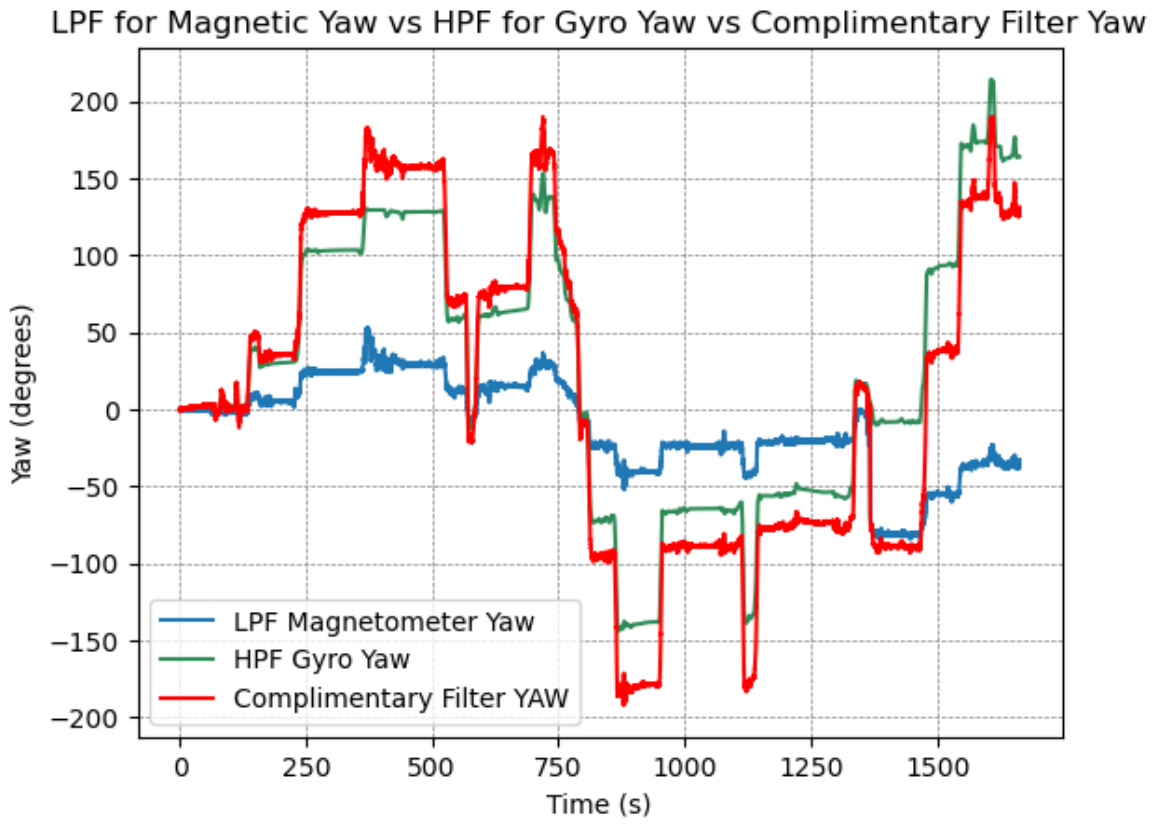

Figure 5: Low-Pass Magnetometer vs High-Pass Gyro vs Complimentary filter YAW

The gyroscope is better at estimating the orientation change while performing a sharp turn and magnetometer can provide smaller orientation changes due to navigating through traffic, minor curves on road, etc. while driving on a relatively straight path. Therefore, the Complimentary filter, the combination of both provides the best estimate of yaw of all which could be used for the navigation purpose. We can see the results of complimentary filter vs the Yaw data from IMU (VNYMR string data) as follows:

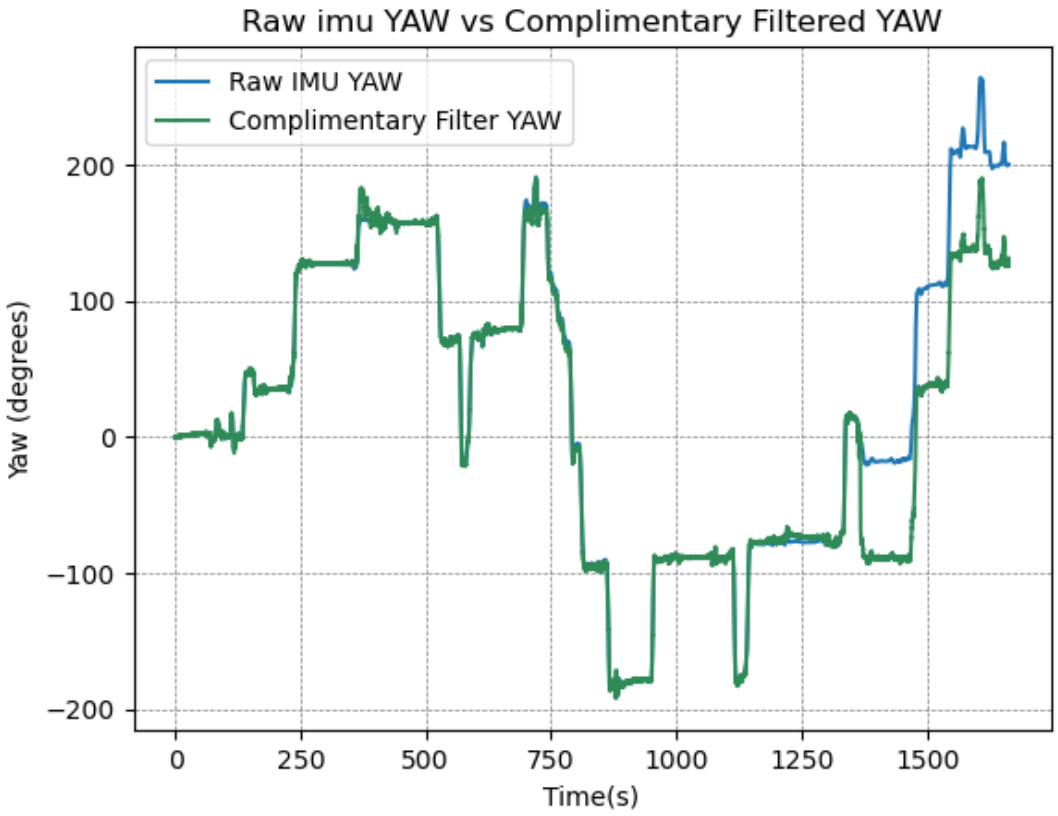

Figure 6: Complimentary Filter vs Raw IMU YAW

We can see from the above figure 6, how the Complimentary filter data matches with the raw IMU yaw angle. However, some discrepancies are seen towards the end, which is due the fact that while collecting data, we came across a dead end and to compensate for it the car was reversed to make a U-turn. Moreover, IMU sensor was placed in a closed space (kept in car) therefore has a drift and also we need to consider the errors due to different sensor bandwidths.

Forward Velocity Estimation

IMU velocity Estimation

We can obtain the estimate of forward velocity by integrating linear acceleration in x direction. However, we need to make some adjustments to compensate for errors in linear acceleration measurements as follows:

- The sensor was not calibrated to set pitch angle 0. Hence to compensate for this error a mean value of pitch was calculated, the product of linear acceleration X and cos(mean(pitch)) was used to align the linear acceleration X values with the X axis of the vehicle.
- The Accelerometer had a constant offset which was subtracted from measurements of linear acceleration in x direction to bring it to zero axis. If this offset is not removed then it will get amplified during integration and the velocity estimates will be drastically off as seen in figure7:

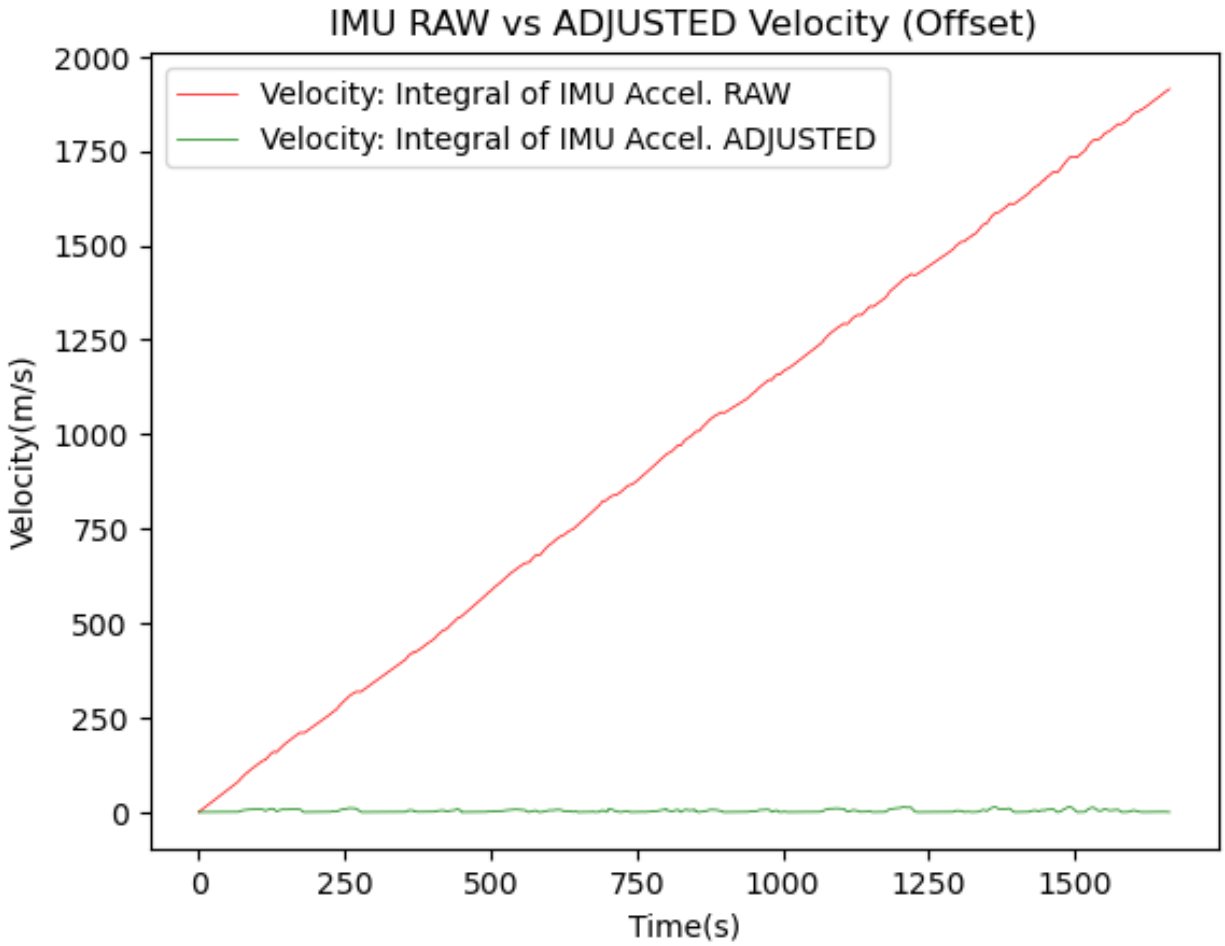

Figure 7: Raw vs Adjusted IMU velocity Estimate (Offset)

- We know that the accelerometer has a bias which changes with time. If this bias is not removed it will get integrated and affect the velocity estimates. Moreover, when the car stops the acceleration and velocity both go to zero. However, due to the discussed time varying bias, the acceleration has a finite value instead of being 0 which leads regions of constant velocity increment (constant slope) as seen in figure8 in RED curve.

To compensate for this, windows of time where velocity is increasing was identified and in each of those time frame the mean of the linear acceleration measurements (bias) was subtracted from all future values. The implementation of this can be found in the analysis script function *adj( ).* Finally, when the car is stopped the acceleration should be zero, but due to the vibrations in the car result of running engine, etc. leads to noise in acceleration measurement about 0 axis. This was compensated by thresholding such small vibrations to be zero if below certain small value. The final Adjustment after Offset compensation can be seen below:

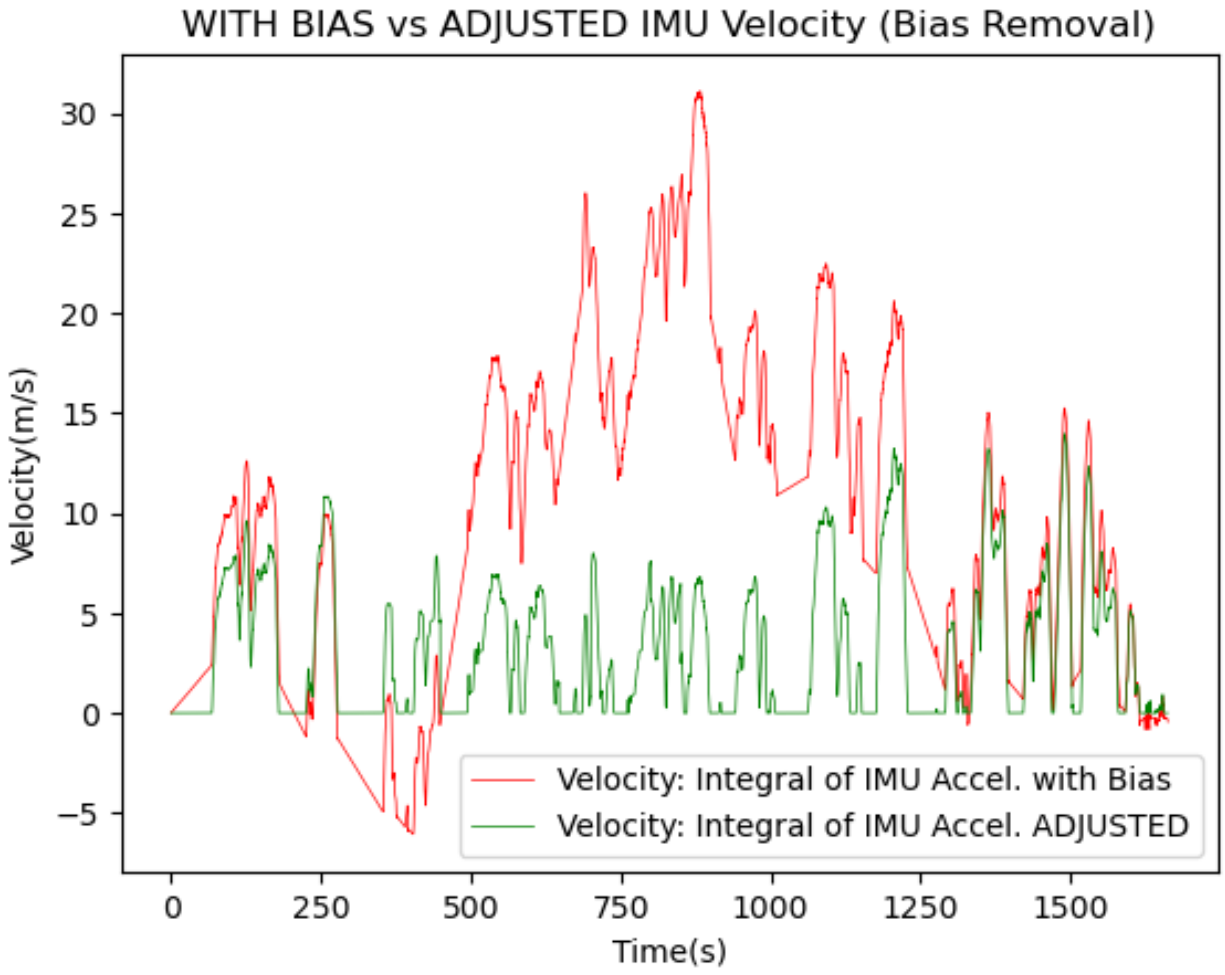

Figure 8: IMU forward velocity adjustment of bias and noise

GPS velocity Estimation

The GPS velocity was obtained by getting the distance between 2 consecutive measurements and then dividing it by the sampling period. The Adjusted IMU and GPS velocities combined plot can be seen below:

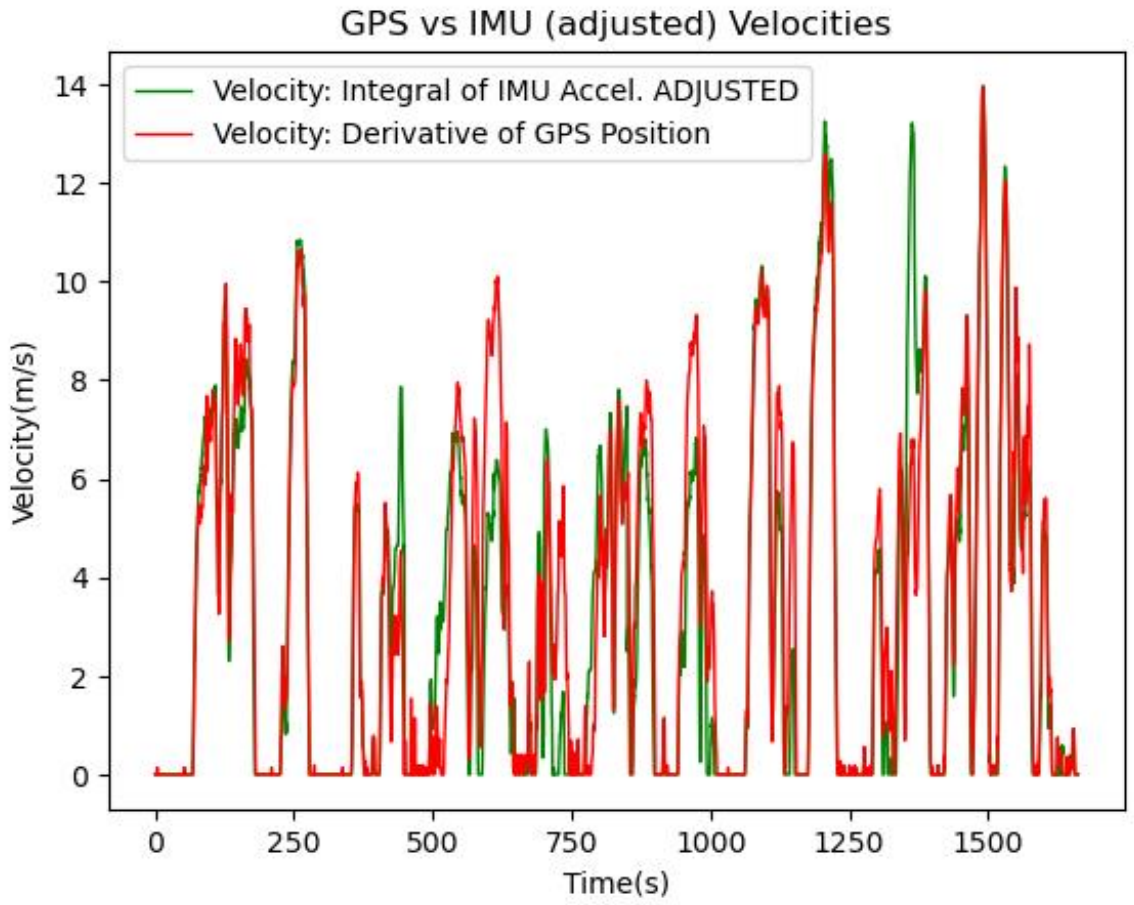

Figure 9: GPS vs IMU (adjusted) Velocities.

From the plot above we can see that both our estimates match very closely to each other. However, we can see some small discrepancies in data. Both the sensors sampled at different frequencies. IMU is sampling at 40 times than GPS, so if we assume there is error in every measurement IMU has 40 errors in a second while GPS has only 1. Moreover, The GPS is using derivative principle for velocity while the IMU is using Integral, in which case the error also gets accumulated over time. Furthermore, we discussed earlier, we took a U-turn and moved in reverse during the data collection, this might have resulted in some errors during bias removal stage, effecting the IMU velocity estimates. Moreover, GPS is accurate while moving but when stopped, the car is stationary and the GPS sensor error could kick in. The above-mentioned reasons may lead to discrepancies in the velocity estimates of the two sensors.

Dead Reckoning with IMU

Dead Reckoning is determining the position of a vehicle, without the aid of satellite navigation, from the record of past course of motion. We will use the velocity and heading angle data to perform dead reckoning using IMU and compare it to GPS position data.

Linear Acceleration in Y

We know that the IMU sensor was fixed on the dashboard of the car, however, the location of Center of Mass of the car could differ from sensor's location.

Center of Mass:

- Location (Space): (X, Y,0)
- Location (Vehicle Frame): (0,0,0)
- Rotation Rate: (0,0, ω)

IMU sensor:

- Location (Space): (x, y,0)
- Location (Vehicle Frame): (xc,0,0)

Then the acceleration measured by the inertial sensor (i.e., its acceleration as sensed in the vehicle frame) is:

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

$$\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega}x_c$$

Here, both the equations and the quantities are assessed in vehicle frame. We know that the car does not slip sideways hence, $\ddot{Y} = \dot{Y} = 0$. For now, we assume that the IMU sensor is coinciding with the COM of car, hence $x_c$terms will be zero. Finally, we get the above two equations as:

$$\ddot{x}_{obs} = \ddot{X}$$

$$\ddot{y}_{obs} = \omega\dot{X}$$

We can integrate the $\ddot{X}$ to get $\dot{X}$. Then, we can compare $\omega\dot{X}$ with $\ddot{y}_{obs}$, plot is shown below:

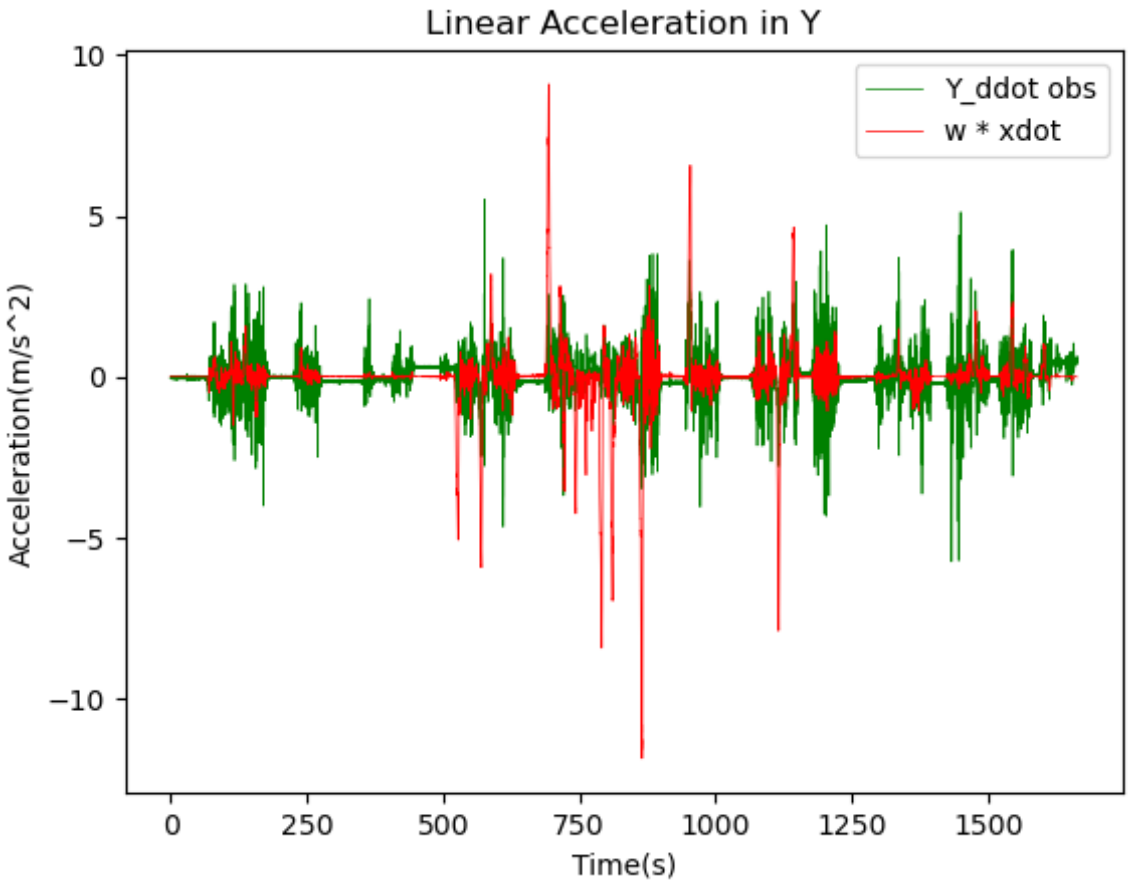

Figure 10: Linear Acceleration in Y direction

From the above figure we see that both the curves follow the same trend yet, there are some discrepancies in the magnitudes. This is because:

- We assumed that sensor coincides with the COM location, however, this is not correct. To rectify, we need to get an estimate of x_c, which was previously zero. Then the comparison is between $\ddot{y}_{obs}\ AND\ \omega\dot{X} + \dot{\omega}x_c$.
- We see that accelerometer readings $\ddot{y}_{obs}$ has a bias that changes with time and correction needs to be made in the manner we did with linear acceleration in x direction while estimation of IMU forward velocity.
- Similarly, for $\omega\dot{X}$, the $\dot{X}$ before integrating needs to be adjusted like in IMU velocity estimate to remove the time varying bias otherwise the error will keep on accumulating on integration.

Vehicle Trajectory Estimation

We further move onto measuring the trajectory of the vehicle using the Yaw estimation of complimentary filter and adjusted forward IMU velocity:

$$V_e = \cos(YAW_{CF}) * Velocity_{adj}$$

$$V_n = \sin(YAW_{CF}) * Velocity_{adj}$$

Then we integrate V_e and V_n to get X_e and X_n in east and north direction. The results of IMU velocity were rotated by 115 counterclockwise to match the initial trajectory from the GPS. The results are as follows:

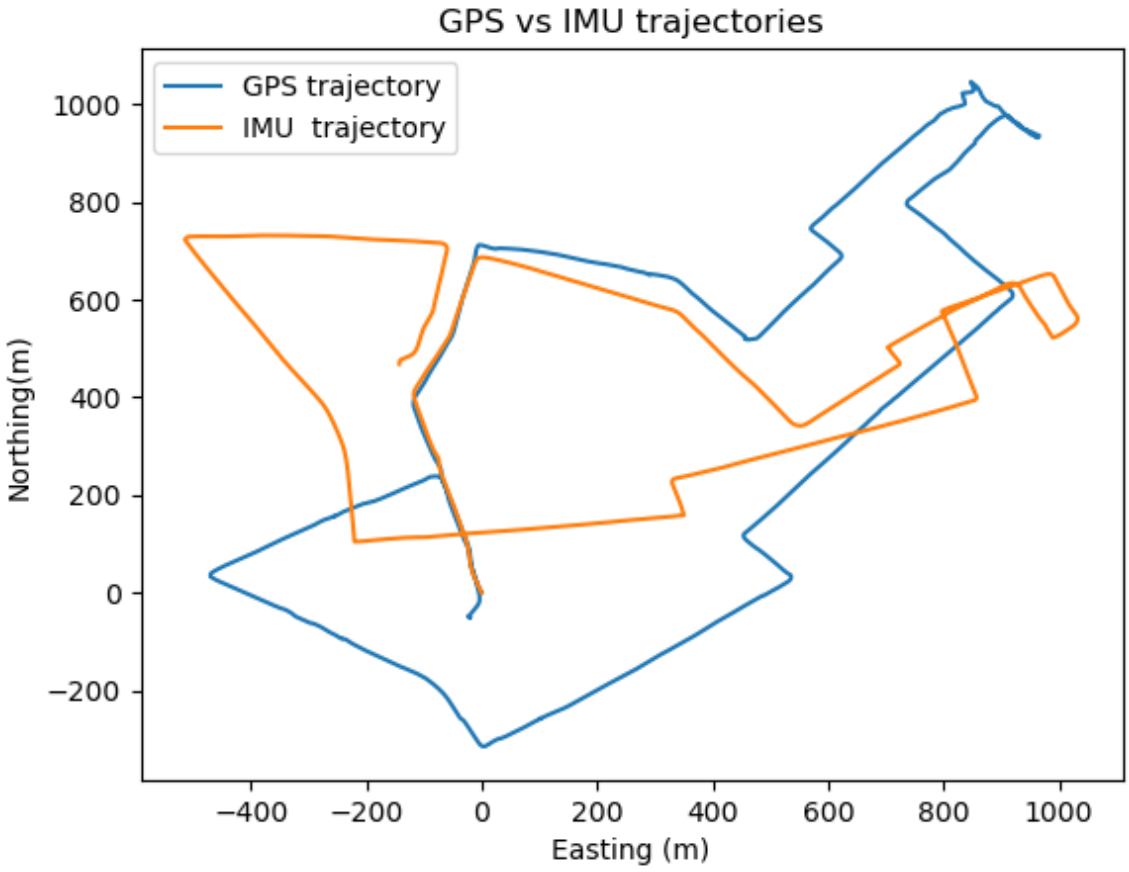

Figure 11: IMU vs GPS trajectory

Disadvantage of Dead Reckoning is that since the new positions are calculated based of previous ones, the errors in the process are cumulated and therefore the error grows with time. We can expect the IMU to be able to perform accurate dead reckoning for initial few seconds as after that the bias in sensors will kick in and vary with time, affecting all the future values. From the Figure 11 we can see that the IMU trajectory matches the GPS trajectory for first 10 secs (approximately) of driving. Moreover, when we make our first turn we can see the IMU trajectory drifting in both orientation and magnitude.

From figure 11, we can see that at sharp turns the IMU trajectory overshoots this is due to the changes in IMU velocity compared to GPS velocity shown in figure9. Moreover, the errors in Yaw estimation as discussed earlier leads to fluctuations in the magnitude of the turns, hence the IMU trajectory tends to overturn at instances when the car made a sharp turn, compared to GPS trajectories. Moreover, the car was reversed and made a U-turn as we came across a No-driving Zone which affected the IMU readings.

Thus, the stated performance matched the actual measurements(GPS) for initial few seconds after which the IMU trajectory, slowly began to change in magnitudes and orientation which is carried for all future values due to the reasons discussed above.

x_c Estimation

We know that the IMU sensor location does not coincide with the Center of Mass of the car. Hence x_c, which is the location of the IMU from the COM can be estimated as follows:

$$\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega}x_c$$

Car is not moving sideways, hence $\ddot{Y} = 0$ and the equation takes form:

$$x_c = \frac{\ddot{y}_{obs} - \omega\dot{X}}{\dot{\omega}}$$

Then take the mean of the x_c array to get finally:

x_c = 0.3488 m.

References

[1] Konvalin, C. (2009) *Compensating for tilt, hard-iron, and soft-iron effects, Fierce Electronics*. Available at: https://www.fierceelectronics.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects (Accessed: April 2, 2023).