Lab 4 – Navigation with IMU and Magnetometer

Immanuel Ampomah Mensah
Northeastern University

Introduction

Navigation systems have become essential to various applications such as unmanned aerial vehicles (UAVs), autonomous vehicles, and robots. These systems rely on a combination of sensors such as a Global Positioning System (GPS) and an Inertial Measurement Unit (IMU) to estimate the position and orientation of the device. However, each sensor has its own strengths and drawbacks, and fusing the information from these sensors can provide more accurate and reliable estimates. In this lab, we explore the capabilities of GPS and IMU sensors, their relative strengths and drawbacks, and learn the basics of sensor fusion. We build a navigation stack that integrates GPS and IMU data to estimate the position and orientation of a device. We evaluate the performance of the navigation stack and compare it with the performance of each sensor separately. The results of this lab can help design navigation systems for various applications where accurate and reliable position and orientation estimates are required.

Collected Data

In this lab, we collect data from two sensors mounted to a vehicle: Global Positioning System (GPS) and Inertial Measurement Unit (IMU). The GPS sensor, mounted to the car's roof, provides easting and other measurements. In contrast, the IMU, mounted in the middle of the dashboard, provides us with linear acceleration, angular velocity, and orientation measurements. The GPS data provided absolute position information, while the IMU data provided relative position information by integrating the acceleration and velocity measurements.

Estimating the Heading

The first portion of this lab deals with estimating the vehicle's heading. This process required calibration steps for the magnetometer to remove distortions and implement filtering techniques discussed later.

Magnetometer Calibration

We drove in a circle to calibrate the magnetometer while collecting inertial, magnetometer, and GPS data. The sources of distortion present were hard and soft iron distortions. Hard iron distortions are caused by the presence of materials that create a constant magnetic field. Examples of such objects in the vehicle would be the speakers, engine components, the car's chassis, and other electronic devices. This is the largest source of distortion because we are constantly surrounded by ferromagnetic materials. Looking at figure 1 below, we know there was hard iron distortion because the initial magnetic field (green circle), is not centered on zero.

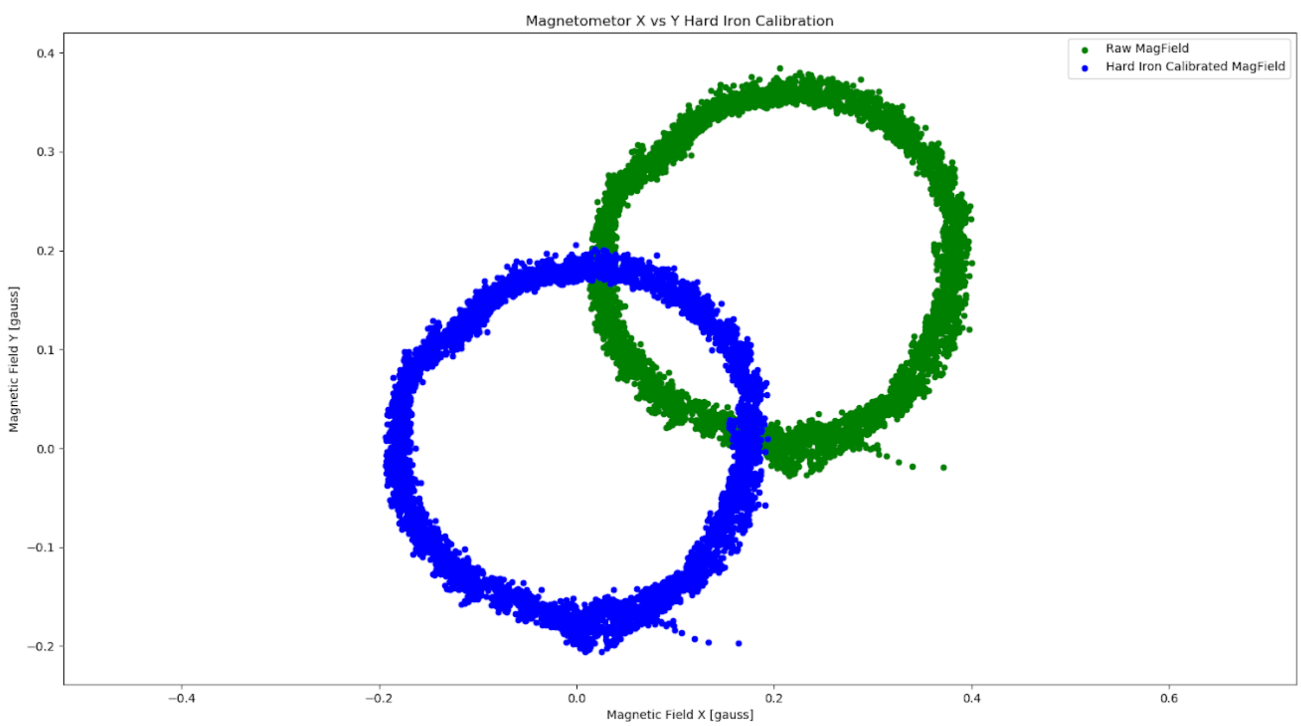

Fig. 1 Hard Iron Calibration

To calibrate for hard iron distortions, we drove in a circle, calculated the offset in the X and Y values of the magnetometer, and applied the offset to the raw dataset by subtracting the offset value from the dataset. Plotting the new values for magnetometers X and Y resulted in the blue circle centered on zero as shown above.

The second type of distortion present was soft iron distortion. This is caused by deflections in the existing magnetic field that leads to skewing of the data. To calibrate for soft iron, a matrix was determined by using the least squares fitting method in python. This matrix was then multiplied by the dataset to rotate and fit the dataset as close to a circle as possible. The equation for the overall calibration can be given as follows:

$$Calibrated\_data = Soft\_iron\_matrix * (Raw\_data - Hard\_iron\_offsets)$$

Eq. 1: Calibration Relationship

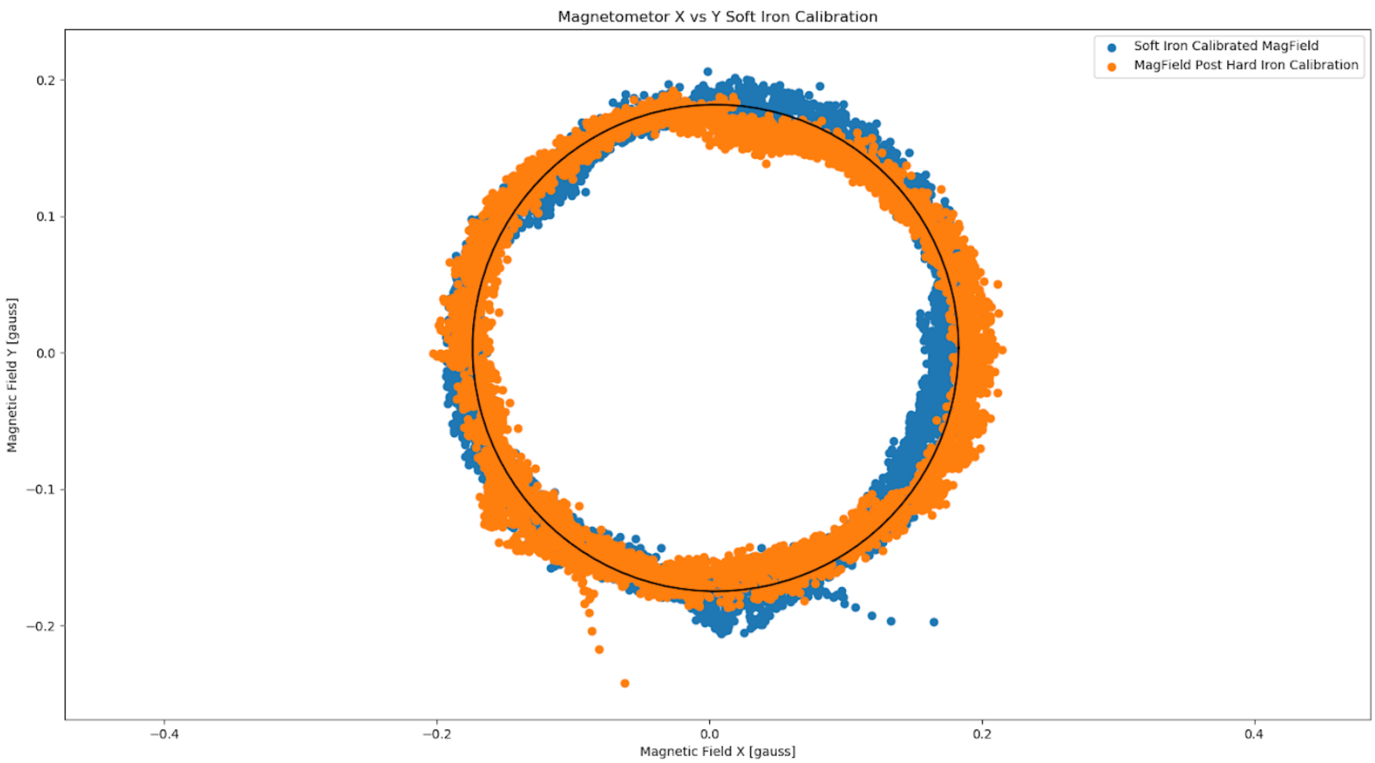

Fig. 2 Soft Iron Calibration

The results of the soft iron calibration can be seen in figure 2 above. From the figure, we can primarily see a rotation and not much of a change in the shape. This is because the initial data was already very close to a perfect circle, meaning there was no visually noticeable amount of soft iron distortion. However, we know that some distortion was still present because the major and minor axes of the original dataset were not equal.

Sensor Fusion

The graph below shows estimates of the car's yaw throughout the drive. The magnetometer yaw was determined with the equation:

$$Yaw = atan2(-mag\_y/mag\_X)$$

Eq. 2 Magnetometer Yaw Estimation

Equation 1 was applied to the magnetometer values plugged into equation 2 above to obtain the calibrated magnetometer yaw. We can see from figure 3 below that the calibrated yaw does not start at zero, this is due to the rotation applied to the data from the soft iron calibration. We can remove the offset by subtracting the first value from the rest of the dataset.

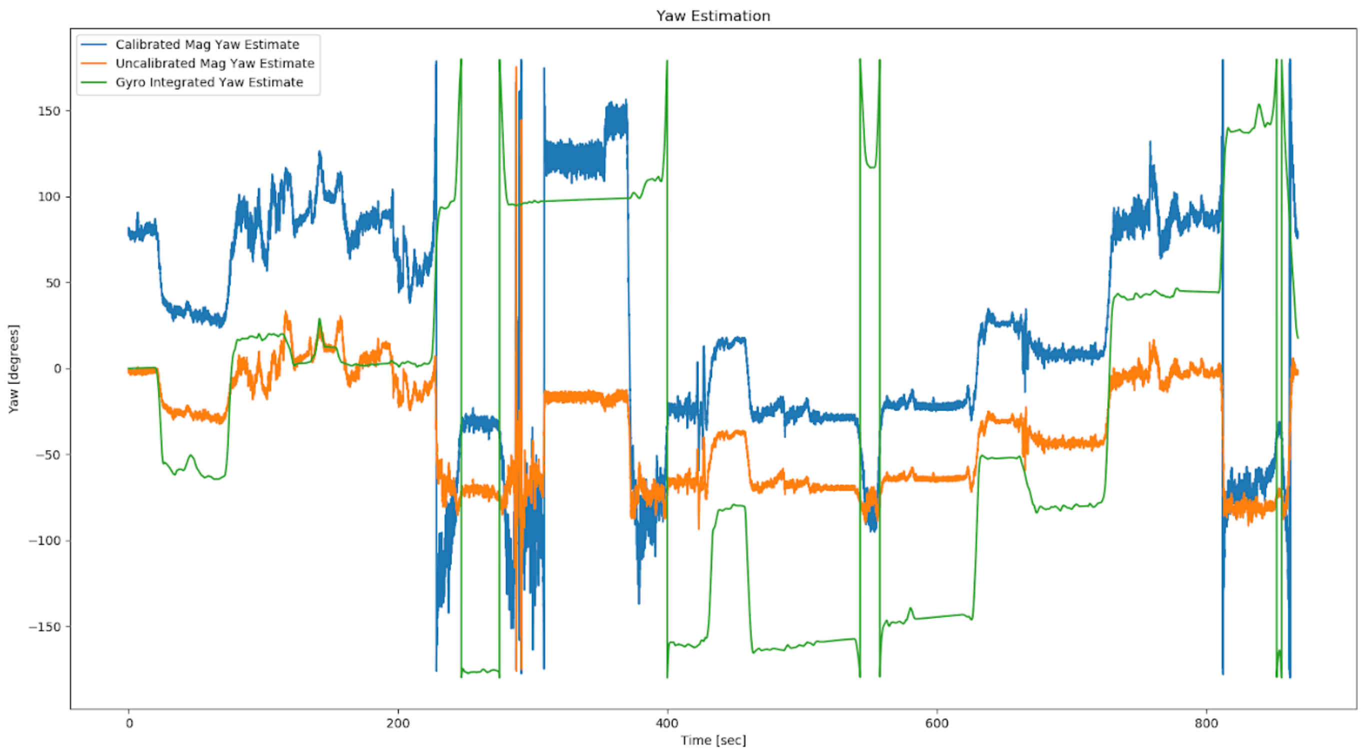

Fig 3. Yaw estimate [Uncalibrated, calibrated, and gyro-integrated yaw]

To obtain a complimentary filter, a Butterworth high pass filter was applied to the gyro yaw, low pass filter was applied to the calibrated magnetometer yaw. The gyro yaw values were obtained by integrating the gyro values about the z-axis. The high and low pass results were then summed to obtain the complimentary filter. The cutoff frequencies for the high pass and low pass filters were 0.00001 and .1, corresponding orders of 1 and 4.

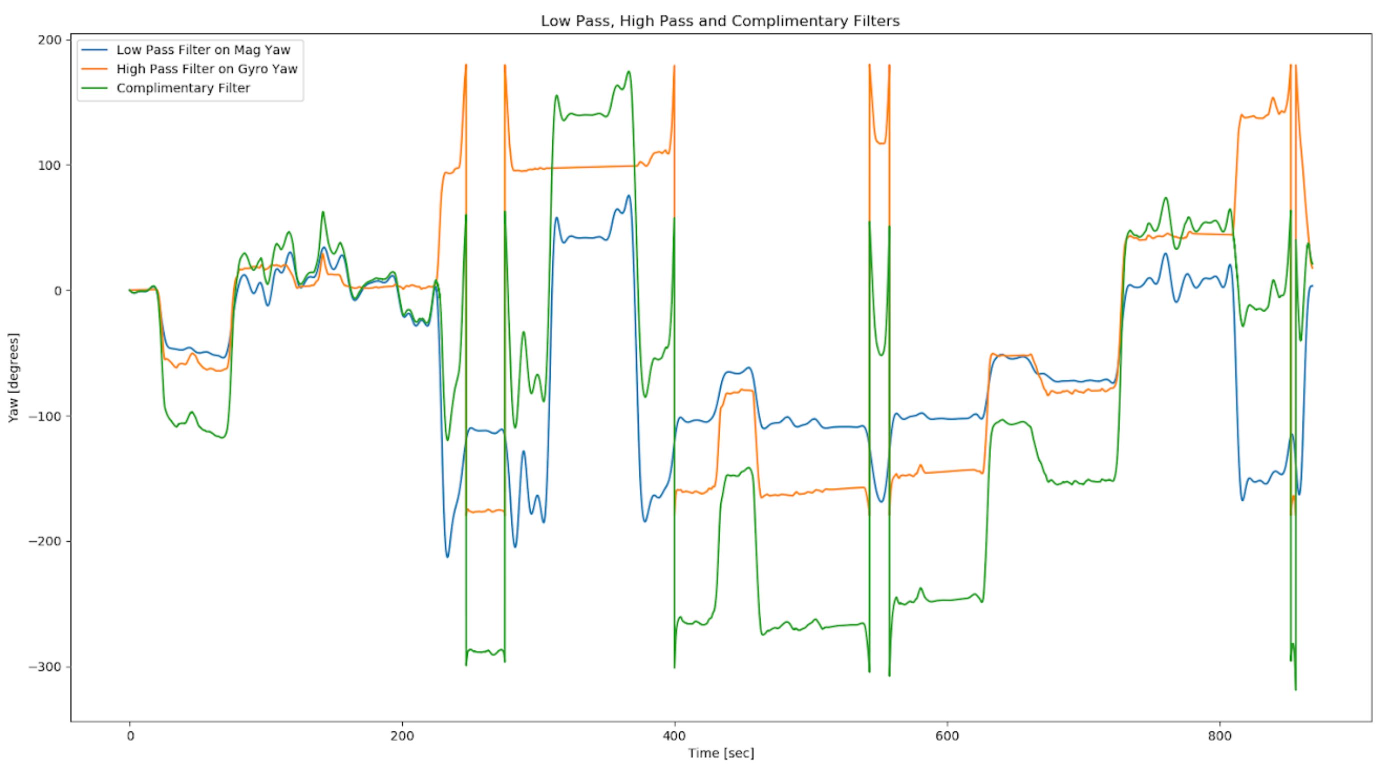

Fig 4. Low Pass, High Pass, and Complimentary Filter

The frequency cutoff and order values were determined via visual inspection of the generated plots and iteration.

Figure 5 below shows a plot of the yaw estimate from the imu, gyro, magnetometer, and complementary filter. From the results, I would trust the estimated yaw derived from integrating the gyro data since it is the closest to the raw imu yaw we are taking as the "true" yaw. In practice, the complementary filter would yield the best results since it combines the output of a high-pass filter (which is good at filtering out high-frequency noise) and a low-pass filter (which is good at filtering out low-frequency drift) to produce a more accurate and stable estimate of the yaw. In our case, sensor drift, bias, and perhaps non-linear responses could all have been improperly accounted for and caused the complementary filter to be less accurate.

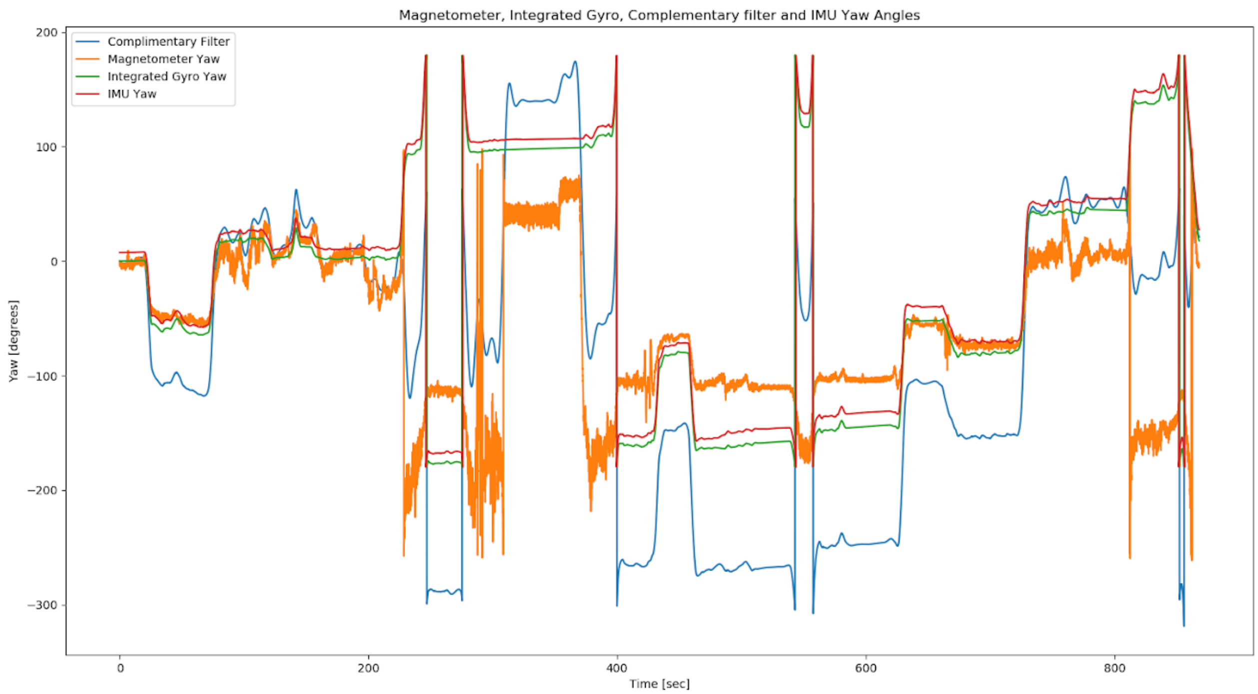

Fig 5. Yaw from Complimentary Filter, Magnetometer, Gyro, and Imu

Estimating Forward Velocity

The forward velocity was calculated by integrating forward acceleration obtained from the IMU. However, the data required adjustments to obtain the results in the figure below.

The first manipulation was implementing a moving average filter to accelerate data to remove bias. This was done instead of subtracting the mean because the bias of the accelerometer is not constant over time. This means integrating the acceleration would result in a graph with very high or low-velocity values depending on the bias.

The second adjustment was 'velocity clamping'. Even with the moving average filter, minor errors in the acceleration data were propagated and magnified when the data was integrated. This resulted in some negative velocity values as shown in figure 6. Since the car never drove backward, we corrected these values by combing the data to find regions where the car was stationary and setting the velocity to zero.

After clamping the velocity, the plot was divided into thirds. An offset was added or subtracted depending on whether the region was below or above the X-axis to create the adjusted velocity graph shown. The final results show an inertial velocity plot with a very similar shape to the GPS velocity. The higher peaks can be accounted for by bias in the imu, noise in the imu signal, and sensitivity of the IMU. These all introduce errors that get amplified when the data is integrated.

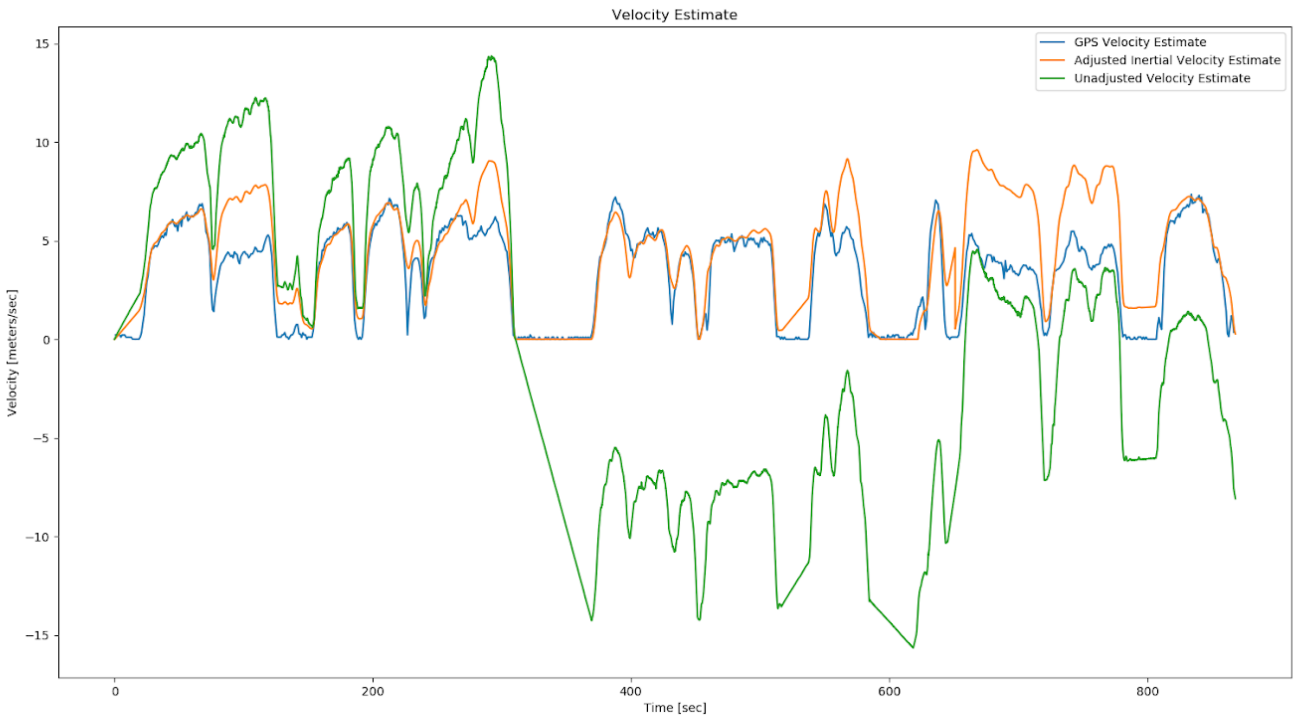

Fig 6. Velocity Estimates from GPS and IMU

Dead Reckoning with IMU

The dead reckoning was calculated from the IMU using the yaw data from the complementary filter to get the x(*Ve*) and y(*Vn*) components of the velocity obtained from integrating the forward acceleration. The easting and northing equivalents of the displacement were then obtained by integrating *Ve* and *Vn* to obtain *Xe* and *Xn* and plotted on the graph shown below. This result was then compared to the dead reckoning obtained from the GPS.

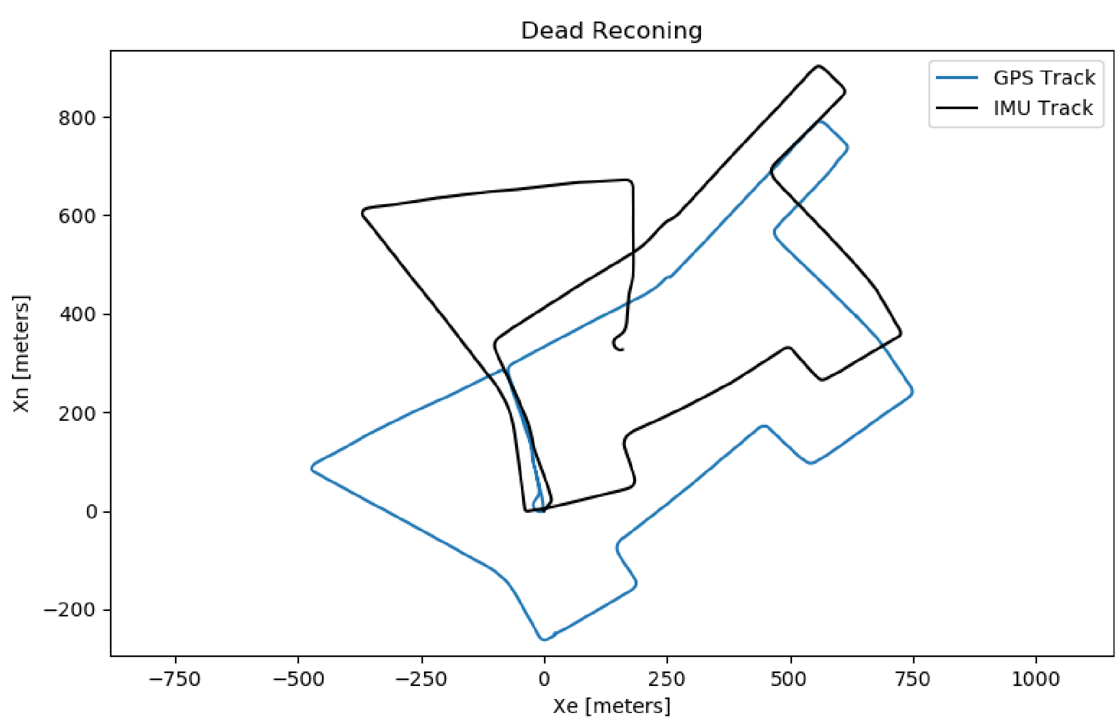

Fig 7. GPS and IMU Dead Reckoning

No scale was applied for comparison. However, the IMU track was rotated 90 degrees and flipped to match the GPS track orientation. We can see from figure 7 that despite not having a closed loop, both plots share the same features concerning turns taken. However, the IMU track is folding in on itself. This suggests that there is a minor warping effect which is likely due to bias and errors in the acceleration and complementary filter. Removing noise and bias from the original dataset would lead to better results.

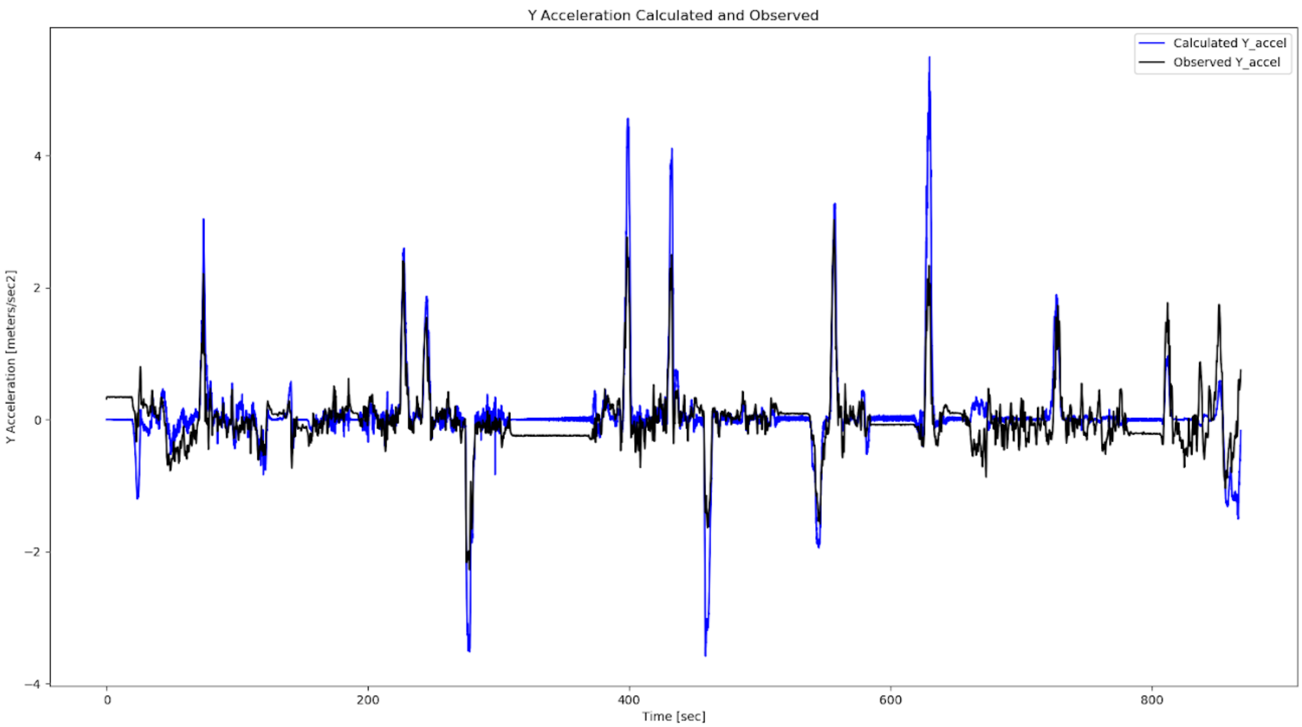

Fig 8. Calculated and Observed Y Acceleration

Figure 8 shows the calculated and observed acceleration in the Y-axis of the IMU. Generally, the form of the two plots matches, showing the exact timing peaks and dips. Discrepancies can be attributed to signal noise and sensitivity differences in the imu.

Estimating Xc

To estimate the displacement vector of the IMU sensor from the center of mass of the vehicle, we can use the acceleration data from the IMU sensor and the equation:

$$\ddot{x} = \ddot{X} + \dot{\omega} \times r + \omega \times X + \omega \times (\omega \times r)$$

We measure the acceleration $\ddot{x}$ of the IMU sensor directly from the IMU data, and we can estimate the rotation rate ω of the vehicle using gyro data from the IMU. We assume that X is approximately equal to the velocity of the center of mass of the vehicle, which can be estimated from the IMU data by integrating the acceleration data twice. We can also assume that $\dot{\omega}$ is approximately zero since the vehicle is unlikely to accelerate its rotation rate significantly over short time intervals.

Given these assumptions, we rewrite the above equation as follows:

$$\ddot{x} \approx \omega \times (\omega \times r) + \omega \times V + R$$

where V is the velocity of the center of mass of the vehicle, r is the position of the center of mass of the vehicle, and R is the acceleration of the vehicle's center of mass.

To solve for *r*, we can rearrange the terms:

$$\ddot{x} - \omega \times V - \dot{R} \approx \omega \times (\omega \times r)$$

Taking the magnitude of both sides:

$$|\ddot{x} - \omega \times V - \dot{R}| \approx |\omega| \times |\omega \times r|$$

Squaring both sides:

$$(\ddot{x} - \omega \times V - \dot{R})^2 \approx (\omega \times r)^2$$

Expanding the cross-product:

$$(\ddot{x} - \omega \times V - \dot{R})^2 \approx (\omega \times r) \cdot (\omega \times r)$$

Taking the square root of both sides:

$$|\ddot{x} - \omega \times V - \dot{R}| \approx |\omega| \times |r| \times \sin(\theta)$$

where θ is the angle between ω and r.

Since r is along the x-axis in the vehicle frame, we can assume that $\theta \approx \pi/2$, so $\sin(\theta) \approx 1$.

Therefore, we can estimate the displacement vector r as:

$$r \approx (\ddot{x} - \omega \times V - \dot{R}) / |\omega| = \mathrm{Xc}$$

where $|\omega|$ is the magnitude of the rotation rate of the vehicle, which can be estimated from the gyro data of the IMU sensor.

This approach can be repeated for each known instance of the vehicle making a turn to obtain multiple estimates for Xc. The mean of these estimates can then be found to find one primary value.

Applying this process to our data resulted in an estimated Xc of 17.4cm. This estimation is understandable since a car turns about a point close to the midpoint of its front axle and the IMU was fixed to the center of the dashboard, which is roughly above the front axle.

Conclusion

It is generally difficult to predict how long the VN-100 can navigate without a position fix, as this will depend on various factors, including the specific application and operating conditions. However, with careful calibration and tuning, achieving accurate and stable navigation for several minutes to a couple of hours is possible. Using the stated gyro bias of 5-7 degrees/hour, a conservative estimate would be around an hour assuming reasonable compensation for drift and bias using good moving average filters, and a well-calibrated complementary filter.

The plot in figure 9 compares the GPS and imu displacements over time. Both graphs show the same general trend with the imu displacement showing higher peaks and lower dips than the GPS displacement. The GPS and IMU position estimates remain relatively close for the first minute to a minute and a half. The IMU estimate appears to be ~50 meters from the GPS estimate up until 700 seconds when they diverge. This discrepancy can be attributed to errors in original acceleration data such as bias that were not adequately accounted for.

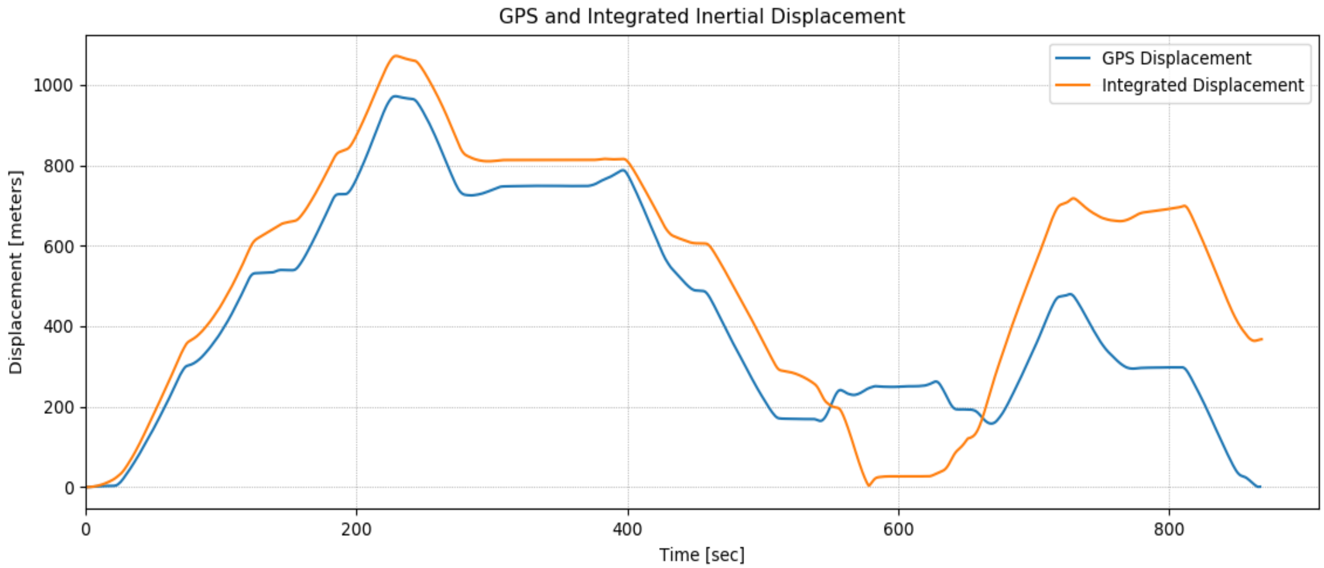

Fig 9. GPS and Integrated Inertial Displacement