LAB 4 REPORT

SENSORS:

- **IMU** (Inertial Measurement Unit) sensor,
- **GPS** (Global Positioning System) sensor.

DATA COLLECTION:

The two IMU and GPS sensors are used to gather the data. Rosbag record was used to gather bag files. We could view the vectornav and gps topics on the terminal while recording. The gps sensor is mounted on top of the vehicle, while the IMU sensor is maintained inside. The two nodes never died while they were collecting data.

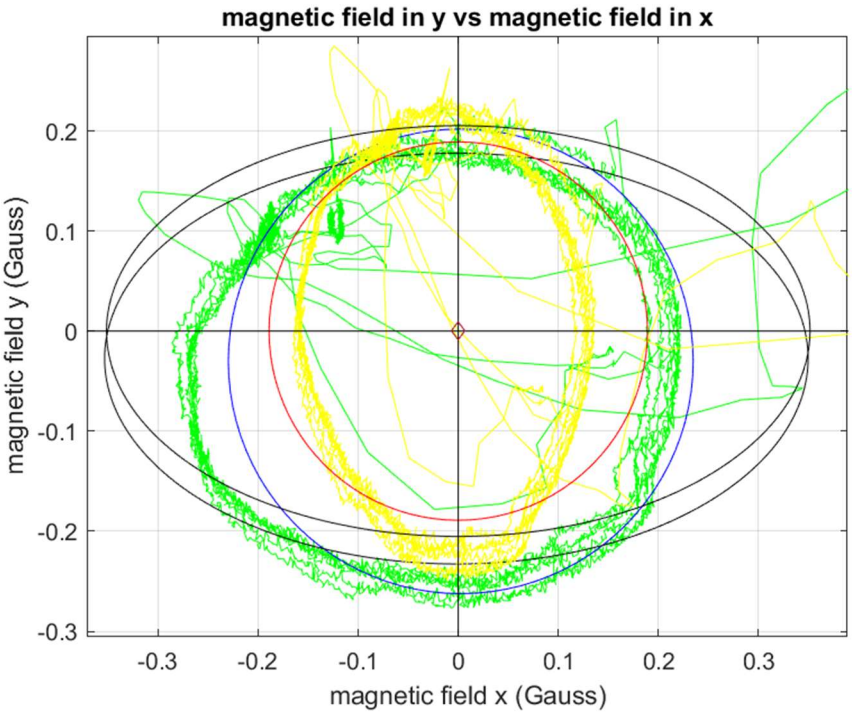
For the data_going_in_circles.bag, we drove the car around Ruggles Circle five times, not far from Centennial Common.

For the data_driving.bag, we drove the car for 2 kilometers taking a minimum of 10 turns.

Analysis of the data collected:

1)Estimating the heading (yaw)

Magnetometer Calibration: Using the data_going_in_circles.bag file and storing the data in the MATLAB the analysis is done as shown below: The magnetometer calibration can be seen below, I used the circfit and ellipse fit functions to fit the data. The green color data shows the before calculated yaw and the yellow color data shows the after calculated yaw.



Magnetometer Hard and Soft Iron Calibration:

A magnetometer is a sensor that analyzes a system's local magnetic field to determine its direction and strength. The heading of a system about magnetic North can then be calculated by comparing this magnetic field measurement to models of the Earth's magnetic field.

To get precise magnetic field measurements during magnetometer calibration, it is crucial to account for both hard iron and soft iron interference.

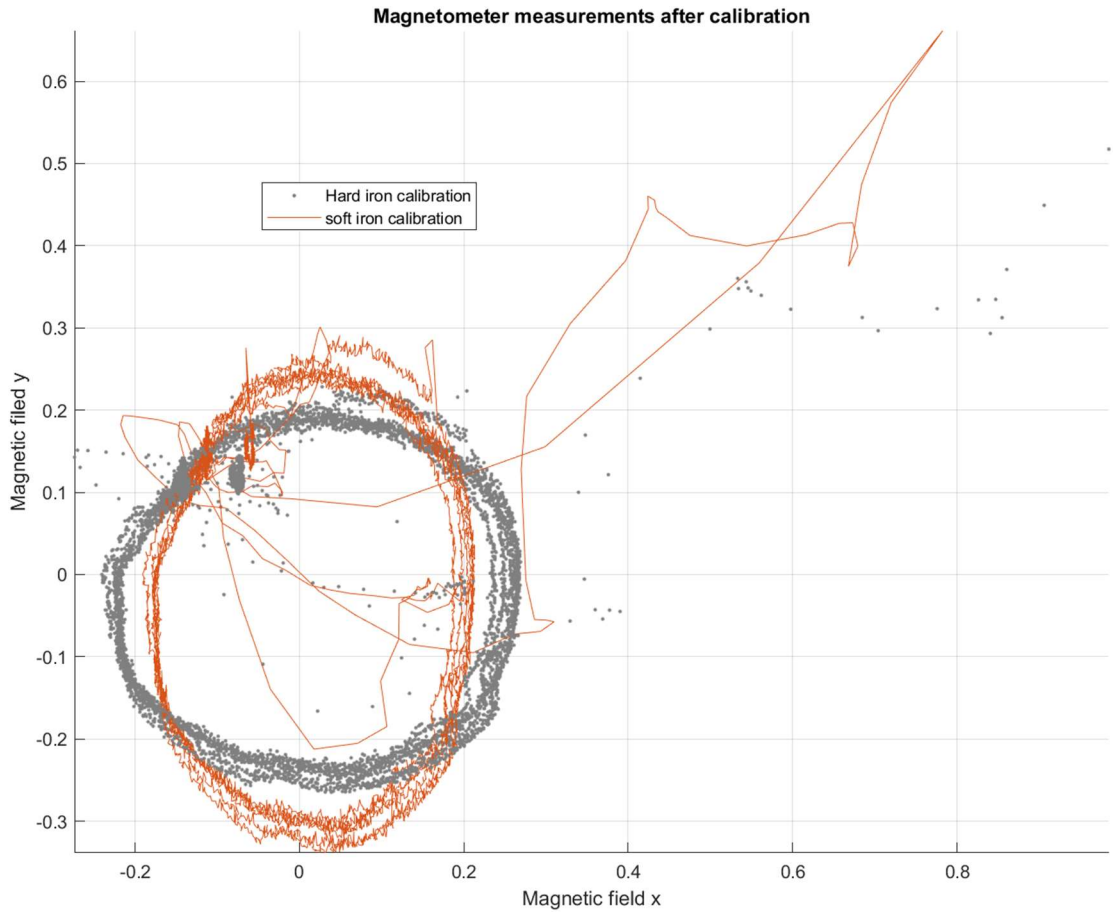
Hard Iron Interference:

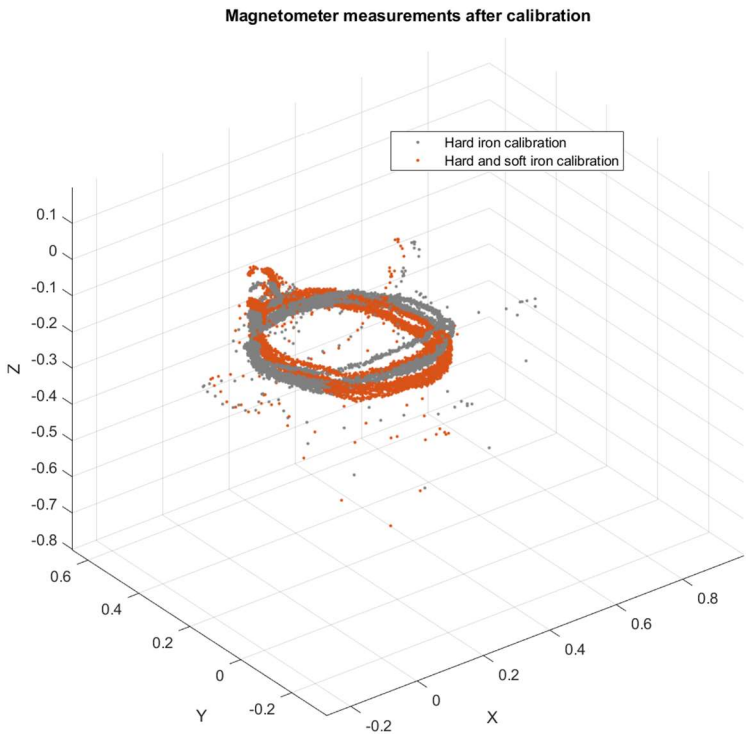
Due to the presence of ferromagnetic materials or permanent magnets close to the magnetometer sensor can induce hard iron interference. When the sensor is in a different orientation or position, the average magnetic field readings can be measured and used to adjust for hard iron interference.

Soft Iron Interference:

The distortion of the magnetic field by surrounding ferromagnetic elements, such as steel or iron structures in the environment, results in soft iron interference. Applying a correction matrix to the measured values of the magnetic field will eliminate soft iron interference.

The below plot shows the Hard iron calibration vs soft iron calibration, the plot has not been centered to its x and y axis.

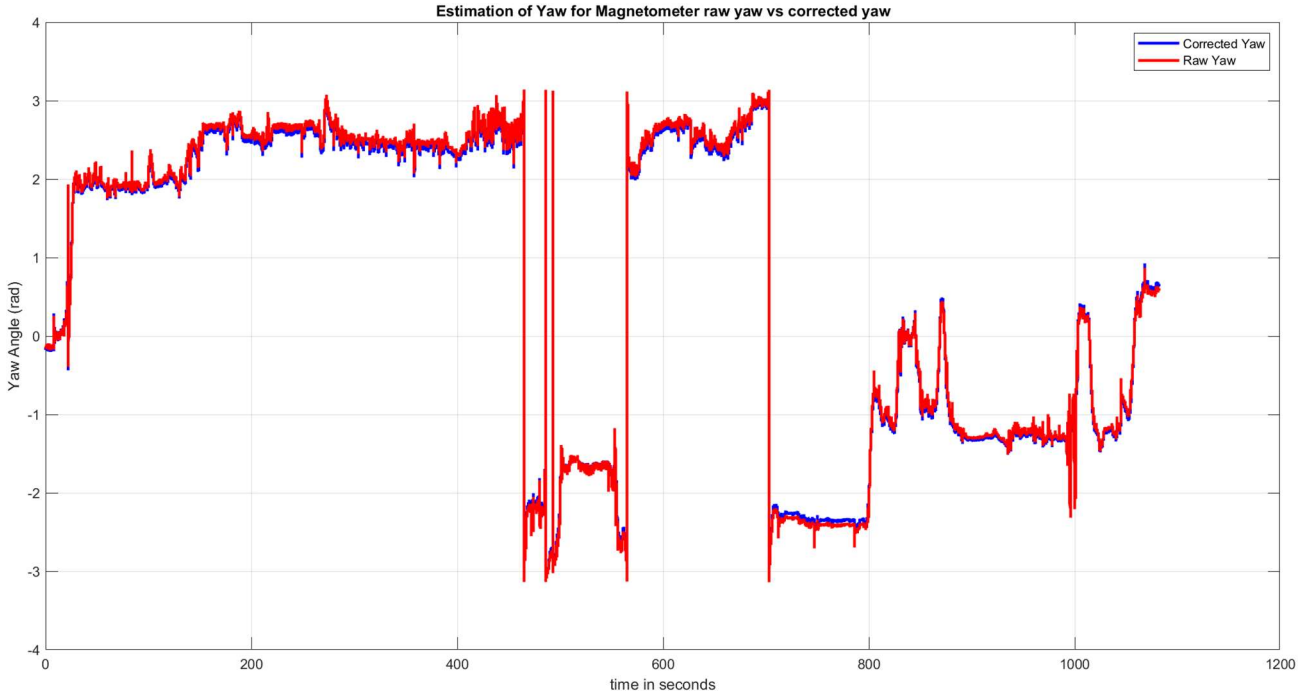




Sensor Fusion:

From here the Analysis is done using the data from the data_driving.bag file.

Calculating the yaw angle from the magnetometer calibration & plotting the raw magnetometer yaw with the corrected yaw for comparison. The raw magnetometer yaw and the corrected yaw both are plotted in same graph with respect to time as shown below:

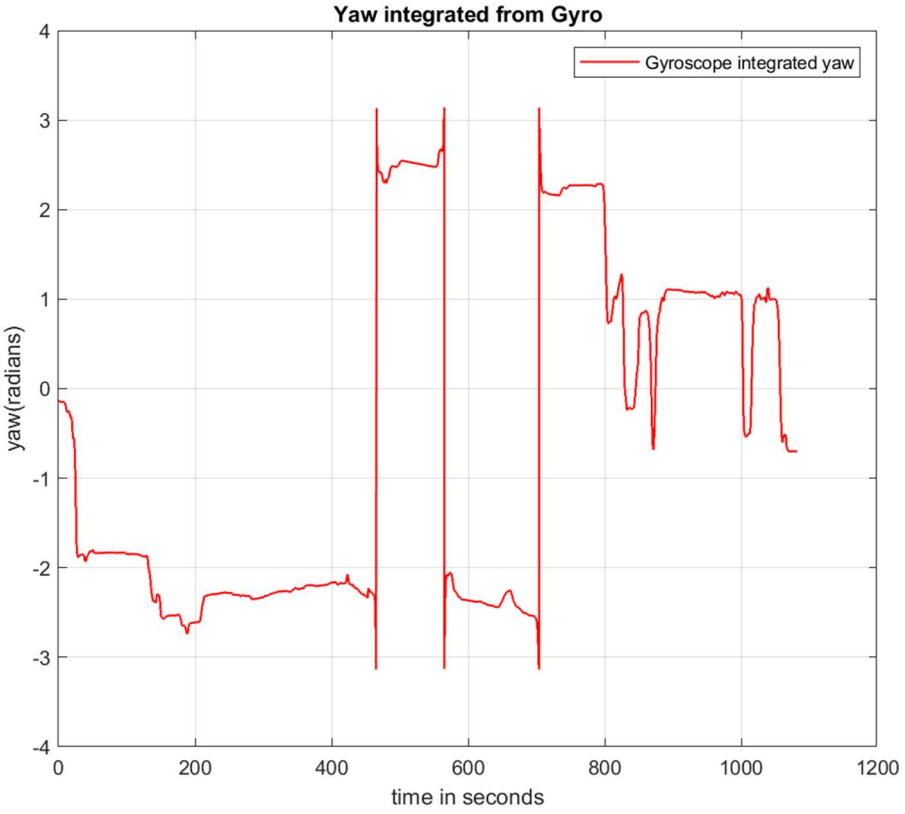


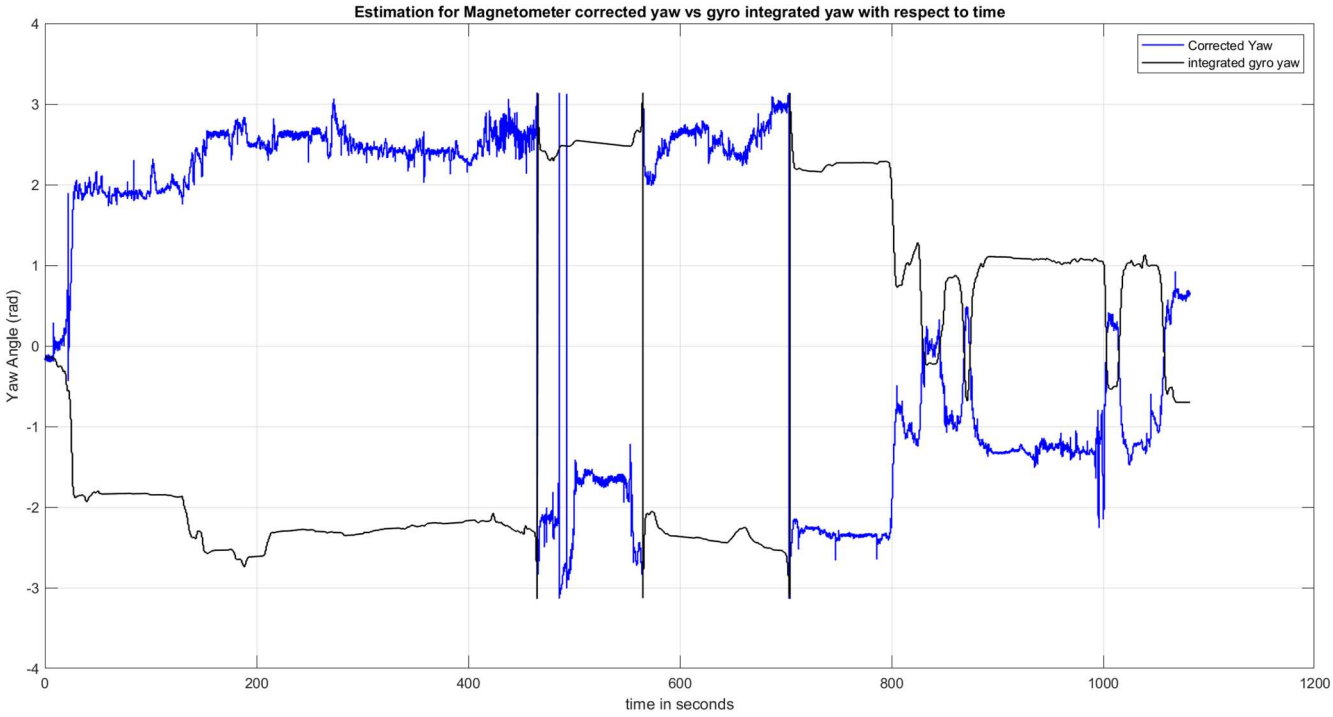
For calculating the corrected magnetometer yaw :

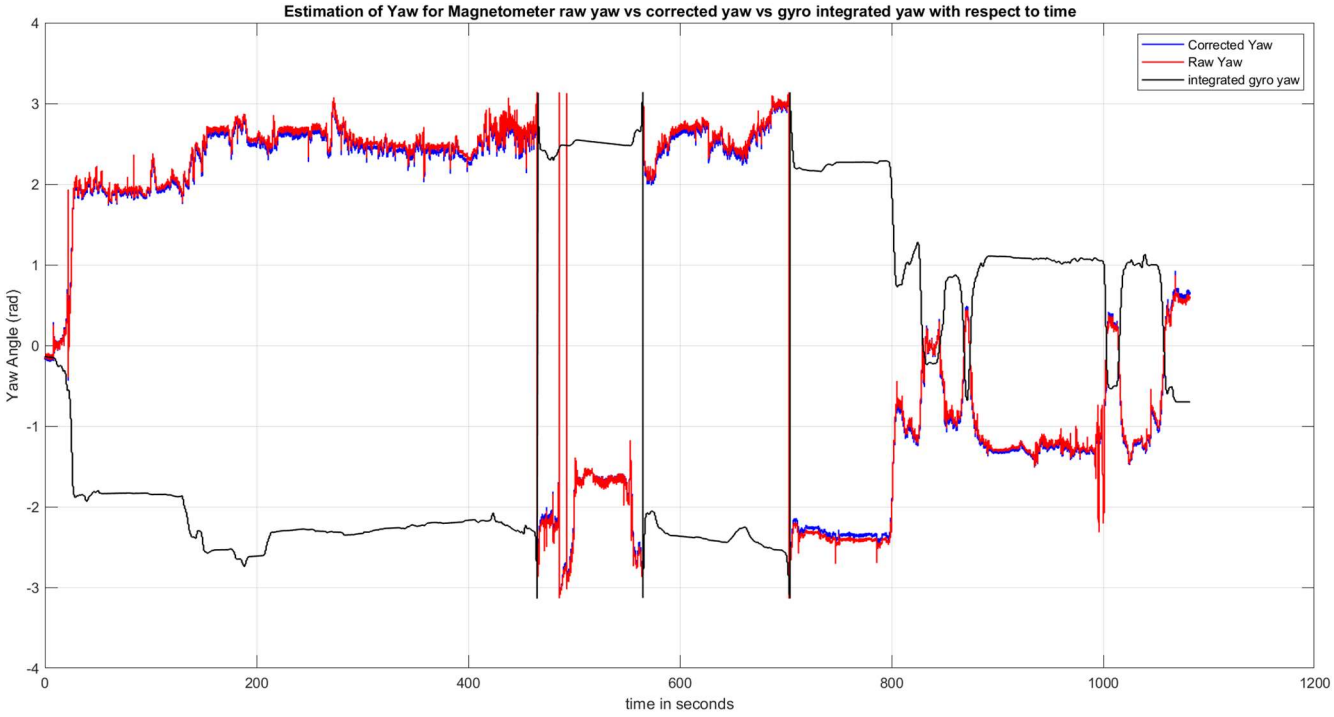
```
magRaw = [mag_x, mag_y, mag_z];
% Calculate the magnetometer bias
biasX = mean(mag_x);
biasY = mean(mag_y);
biasZ = mean(mag_z);

% Calculate the magnetometer scale factor
scaleX = max(mag_x) - min(mag_x);
scaleY = max(mag_y) - min(mag_y);
scaleZ = max(mag_z) - min(mag_z);

% Construct the magnetometer calibration matrix
magCalibration = diag([1/scaleX, 1/scaleY, 1/scaleZ]);
magCalibration(1, 4) = -biasX/scaleX;
magCalibration(2, 4) = -biasY/scaleY;
magCalibration(3, 4) = -biasZ/scaleZ.

magCorrected = magRaw * magCalibration;
```

integrate the yaw rate/gyro sensor to get yaw angle:

After the magnetometer's calibration with hard and soft iron, the yaw angle is discovered. It can also be obtained by combining data from gyroscopes. Both calculated yaw nearly resembles the internal yaw angle that the imu calculated. In comparison to the yaw estimated from the IMU data, the yaw calculated from the gyroscope data is comparatively smoother and less sensitive (fewer peaks), and the yaw obtained from the magnetometer data is relatively less smooth.


Yaw integrated from Gyro
Gyroscope integrated yaw
yaw(radians)
time in seconds
4
3
2
1
0
-1
-2
-3
-4
0
200
400
600
800
1000
1200

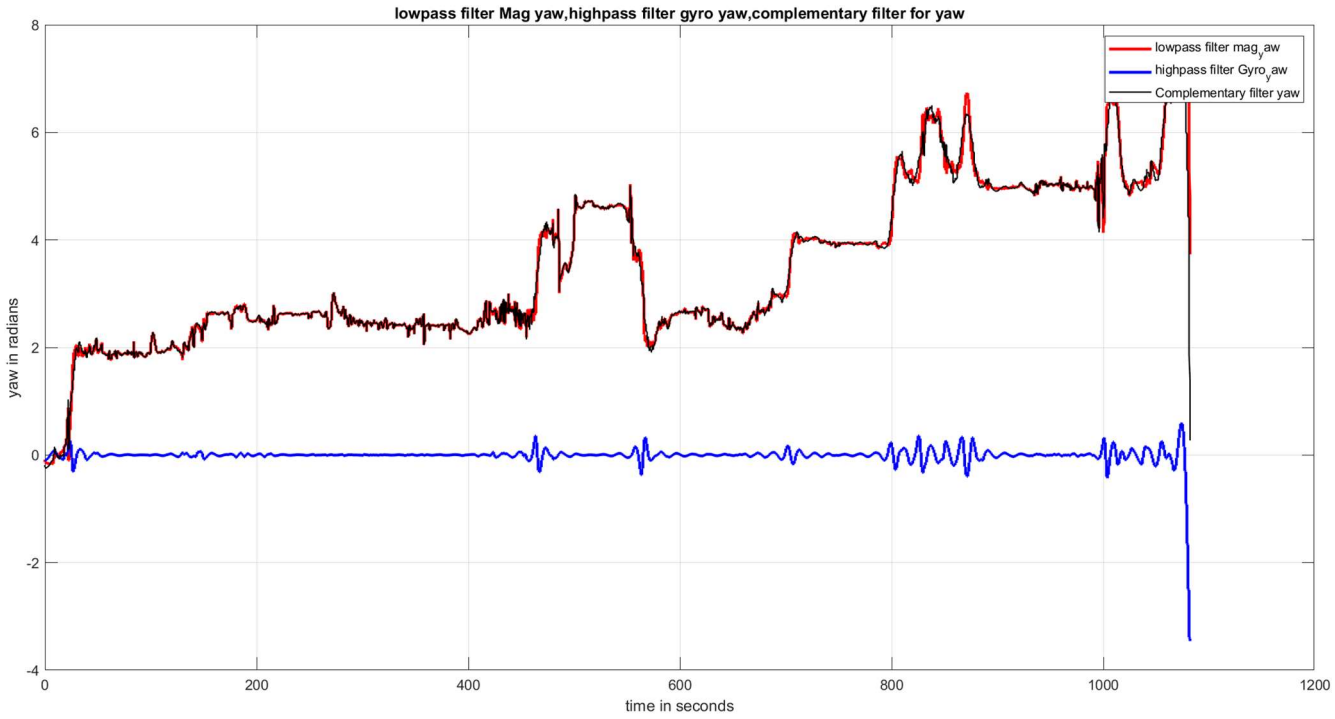

Estimation for Magnetometer corrected yaw vs gyro integrated yaw with respect to time
Corrected Yaw
integrated gyro yaw
Yaw Angle (rad)
time in seconds
4
3
2
1
0
-1
-2
-3
-4
0
200
400
600
800
1000
1200



A complementary filter is a kind of filtering algorithm that integrates the results of two or more sensors to estimate a physical quantity more precisely. In this instance, a yaw rate gyro and a magnetometer's readings of yaw are combined using the complementary filter to get a more accurate estimate of the yaw angle.

The yaw angle can be calculated using the magnetometer's absolute measurement of the Earth's magnetic field. The magnetometer readout is also susceptible to noise and drift, particularly when there are external magnetic fields present. On the other hand, integration drift over time affects the yaw rate gyro, which measures the rate of change of the yaw angle.

The magnetometer and gyro measurements can be combined using a complementary filter to solve these problems. A high pass filter is used to eliminate low-frequency drift from the gyroscope measurement while a low pass filter is used to remove high-frequency noise from the magnetometer measurement. A weighted sum is then used to integrate the filtered magnetometer and gyro measurements to estimate the yaw angle.



High pass Filter:

There are three inputs altogether for the high pass function in MATLAB, and they are (input signal, high-pass filter cutoff frequency and the filter order). Prior to filtering, the function "unwrap" is applied to prevent phase wrapping problems that can arise when the observed angle is greater than the range of-pi to pi radians.

Low pass Filter:

Before filtering, the function "unwrap" is applied to "yaw" to prevent phase wrapping problems that may arise when the observed angle is greater than the range of-pi to pi radians. The 'lowpass' function takes three arguments: the input signal, the low pass filter cutoff frequency, and the filter order.

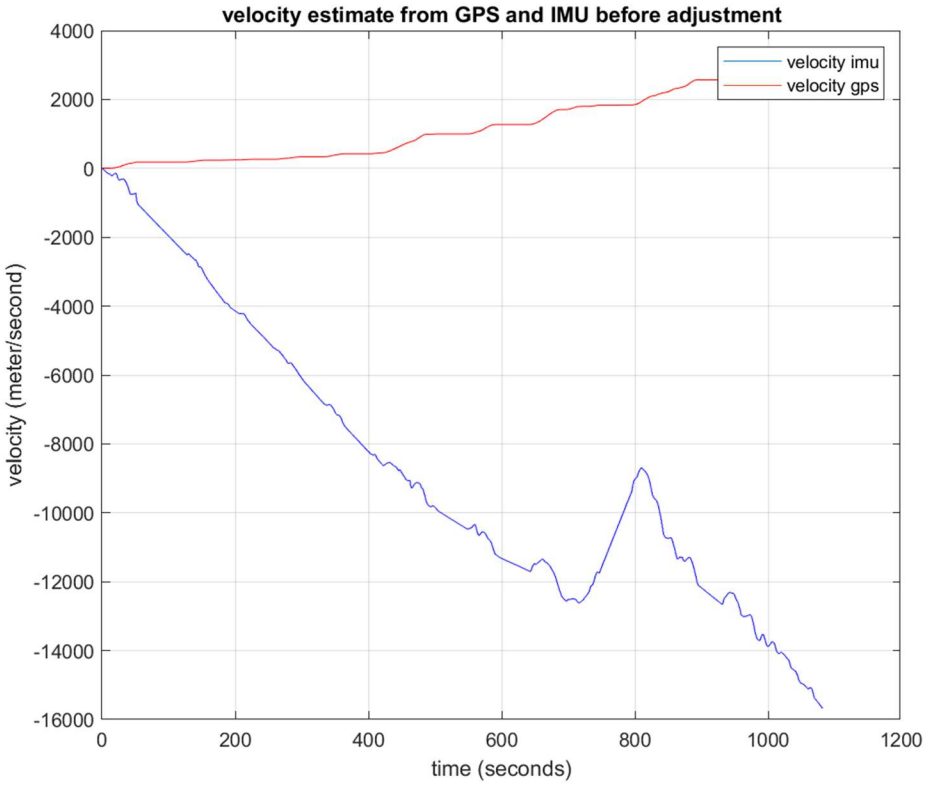
Complementary Filter:

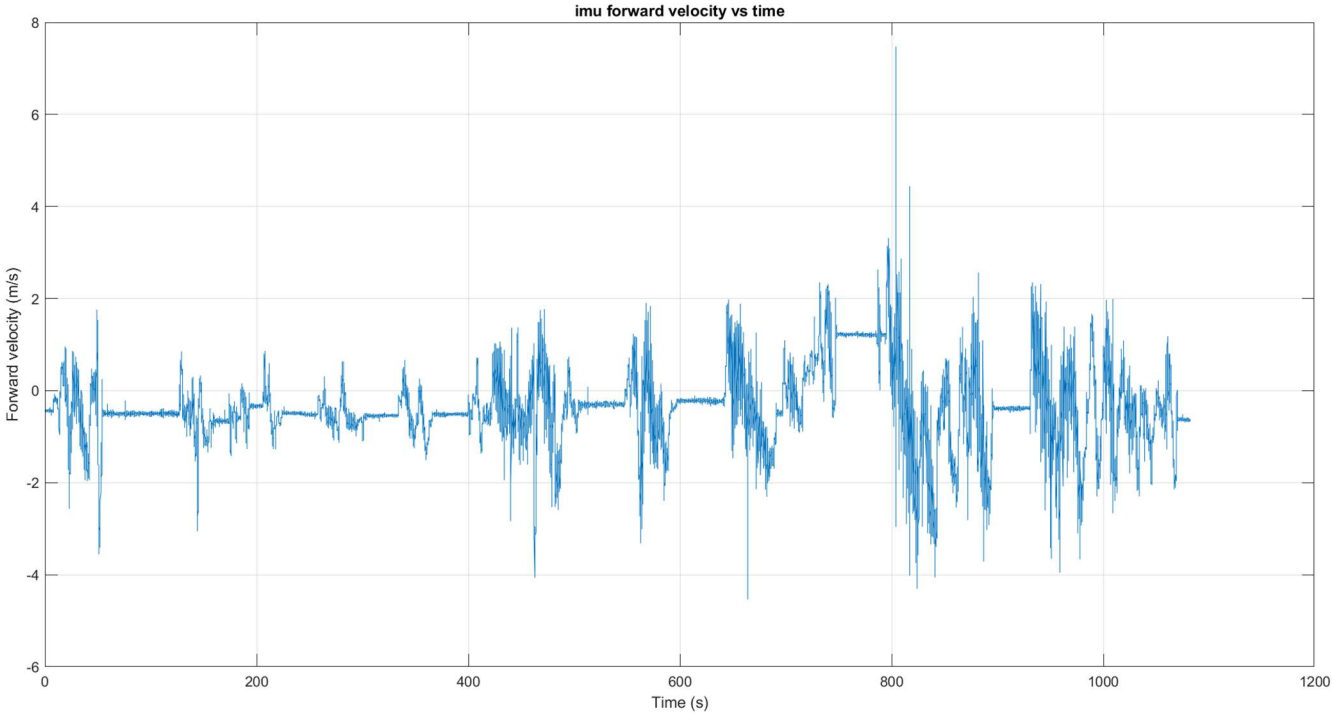
Overall, it is possible to get a better estimate of the yaw angle that is more precise and resistant to noise and drift by combining the high pass filtered gyro readings and the low pass filtered magnetometer measurements using a complementary filter.

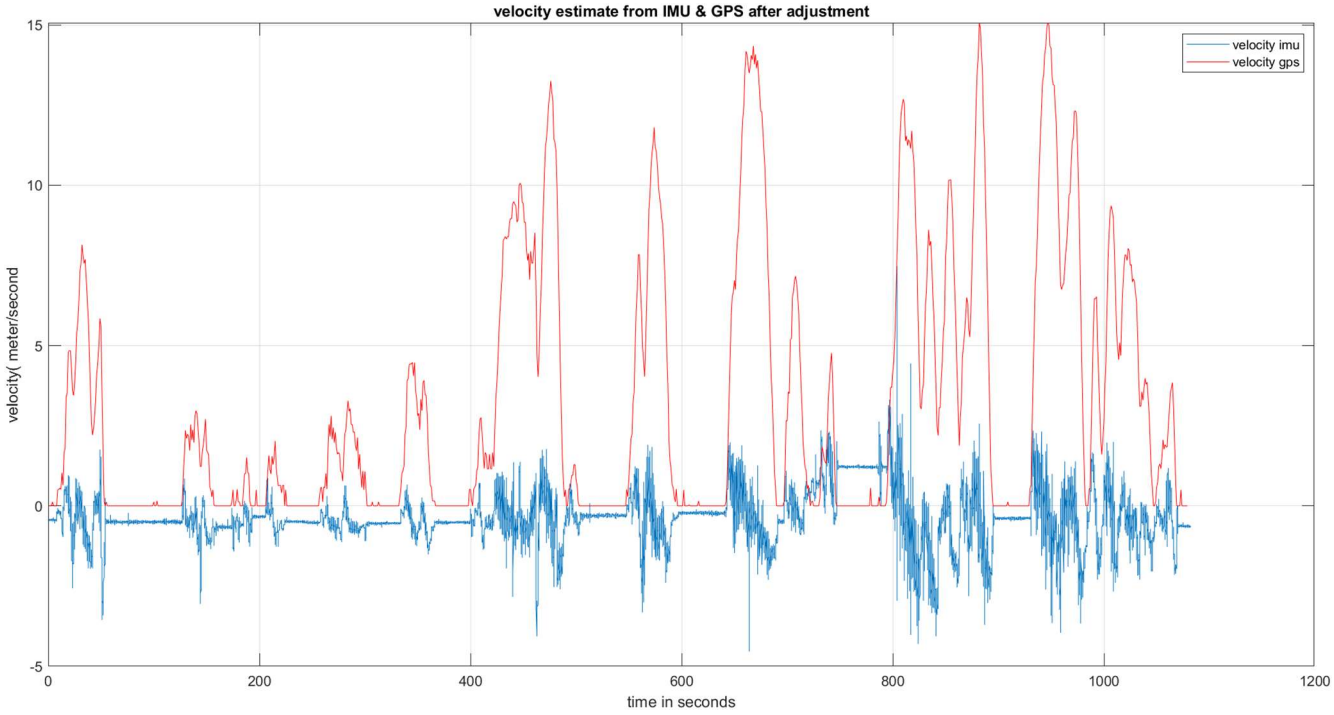
Estimate the Forward Velocity:

The instantaneous velocity of each GPS data point and the integral of acceleration can both be used to calculate velocity. Integrating accelerometer data allows us to track the speed or velocity of our vehicle. Although the accelerometer is positioned with its direction pointing directly out the front of the car, we are only using its X component in this instance. Y component linear acceleration is not taken into account in this experiment because the car is not moving sideways orientation. The hypotenuse of the easting

and northing of our gps location data can be utilized to calculate forward velocity as a substitute for the gradient with respect to time.





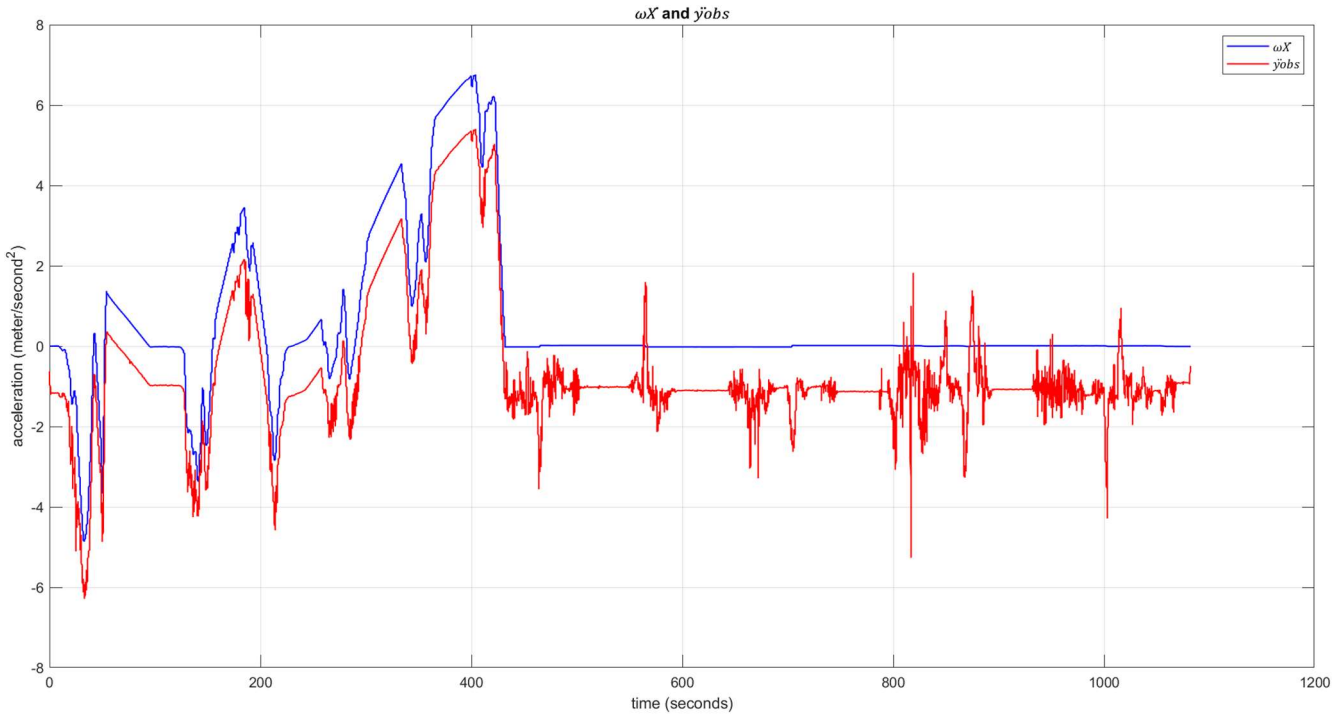


Dead Reckoning with IMU:

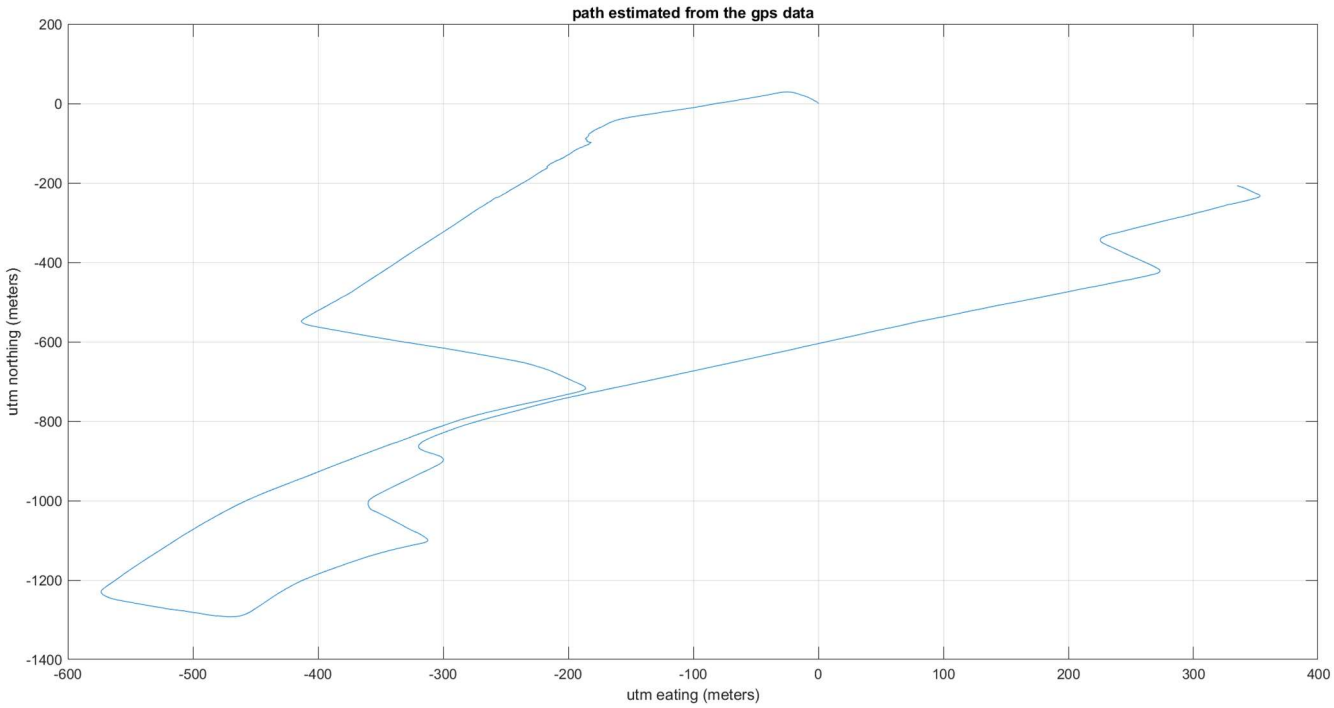
Estimating a vehicle's present position and orientation using data from an inertial measurement unit is known as "dead reckoning with IMU". Accelerometers and gyroscopes, which monitor a vehicle's linear acceleration and angular velocity, respectively, make up an IMU.

The velocity of the vehicle can be estimated by integrating the linear acceleration data across time, and the displacement (i.e., change in position) of the vehicle can be estimated by integrating the velocity. Estimating the orientation (i.e., the roll, pitch, and yaw angles) of the vehicle is also achievable by fusing the velocity and displacement estimations with the angular velocity measurements from the gyroscopes.

In autonomous vehicle navigation, dead reckoning with IMU is frequently used to estimate the position and orientation of the vehicle in relation to a map or a list of predetermined waypoints. Dead reckoning with IMU has some drawbacks, too, like drift and inaccuracies brought on by sensor noise and bias, which over time can add up and result in estimates of position and orientation that are incorrect. As a result, it is frequently integrated with additional sensors, like GPS, lidar, and cameras, to boost the navigation system's precision and dependability.



The below plot is created using the GPS UTM Easting and UTM Northing values, the data is from data_driving.bag file.



How did you calibrate the magnetometer from the data you collected? What were the sources of distortion present, and how do you know?

- For calibrating the magnetometer from the data, I used soft-iron calibration and hard-iron calibration. The hard-iron calibration is for correcting the fixed offset biases and soft-iron calibration is for correcting non-linear distortions. Sources of distortion present are Sensor noise, Magnetic interference.

How did you use a complementary filter to develop a combined estimate of yaw? What components of the filter were present, and what cutoff frequency(ies) did you use?

- High pass Filter: There are three inputs altogether for the high pass function in MATLAB, and they are (input signal, high pass filter cutoff frequency and the filter order). Prior to filtering, the function "unwrap" is applied to prevent phase wrapping problems that can arise when the observed angle is greater than the range of-pi to pi radians. Sampled at a rate of about 40hertz and the frequency filter was set to 0.07.
- Low pass Filter: Before filtering, the function "unwrap" is applied to "yaw" to prevent phase wrapping problems that may arise when the observed angle is greater than the range of-pi to pi radians. The 'lowpass' function takes three arguments: the input signal, the low pass filter cutoff frequency, and the filter order. Sampled at a rate of about 40 hertz and frequency filter was set at 0.0001hertz.
  Complementary Filter: Overall, it is possible to get a better estimate of the yaw angle that is more precise and resistant to noise and drift by combining the high pass filtered gyro readings and the low pass filtered magnetometer measurements using a complementary filter.

Which estimate or estimates for yaw would you trust for navigation? Why?

- I would trust using the complementary filter because it takes account of both sensors and minimizes the weaknesses. The yaw estimated from a magnetometer alone cannot be reliable as it might be affected by sources of errors.

What adjustments did you make to the forward velocity estimate, and why?

- The primary adjustment was to eliminate the biases from the data, which in this case involved identifying regions of the graph that were stationary (i.e., not in motion), computing the mean, subtracting it from the remaining data, and repeating this process for all stationary regions in the acceleration plot.

What discrepancies are present in the velocity estimate between accel and GPS. Why?

- The GPS measurements were smoother and more accurate than the accelerometer measurements and the GPS measurements showed small inaccuracies in the measured velocity.
- To address these discrepancies, we can use the complementary filter to combine the accelerometer and GPS measurements and improve the overall accuracy.

Compute ωX and compare it to $\ddot{y}obs$. How well do they agree? If there is a difference, what is it due to?

- At first differentiated the yaw rate w to get the derivative of yaw rate then used the central difference method to approximate the second derivative of the y observed position and then I compared both.

- The plot between ωX and $\ddot{y}obs$ the showed some differences may be due to the errors in the computation of the derivatives and the trend is also not mostly similar in the beginning it was same.

Given the specifications of the VectorNav, how long would you expect that it is able to navigate without a position fix? For what period did your GPS and IMU estimates of position match closely? (Within 2 m) Did the stated performance for dead reckoning match actual measurements? Why or why not?

- The accuracy of both GPS sensor and IMU sensor were good. The discrepancy between the actual and projected dead reckoning performance could be the result of a few things, including IMU calibration issues, sensor measurement drift, or mistakes in the initial position estimation. Moreover, outside variables like topographical fluctuations and magnetic interference can impair the navigation system's performance.