LAB 4: Report

In this lab, the goal is to build the Navigation stack using two different sensors viz., GPS and IMU. Then understand their relative strengths and drawbacks. Also, the aim is to learn about sensor fusion.

The two sensors that were used in this lab are GNSS puck and VectorNav VN-100 IMU.

The data acquisition was performed in a team and two datasets were collected, one by driving the vehicle in circles for five times near Ruggles Station and the mini tour data was collected near the Ave by starting and ending at the same point.

Magnetometers detect magnetic field strength along a sensor's X, Y and Z axes. Accurate magnetic field measurements are essential for sensor fusion and the determination of heading and orientation.

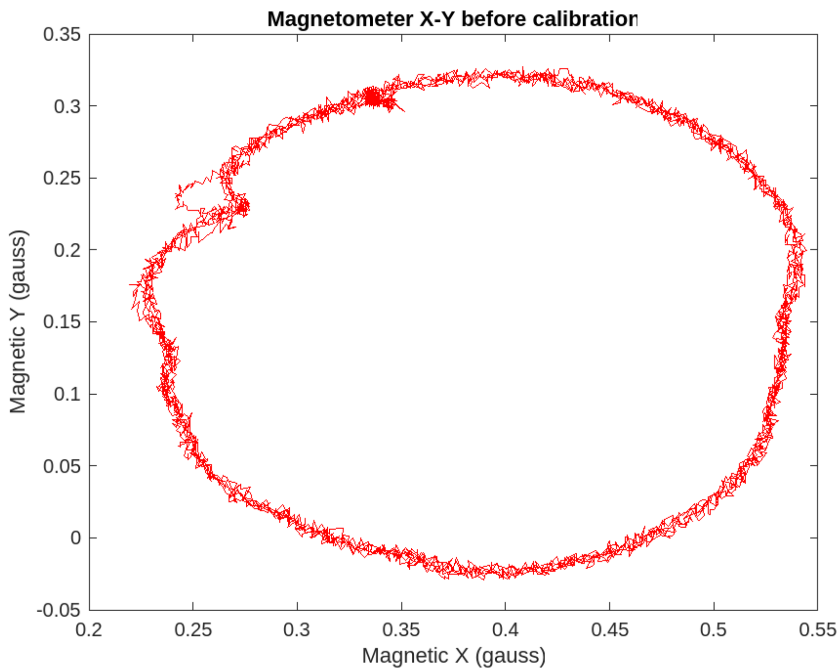

Fig 1: Magnetometer X-Y plot before calibration of circle data

In the IMU data, the magnetometer measurements contain Hard Iron and Soft Iron Effects. The hard iron effect is due to the noise sources and manufacturing defects which come from the metallic objects on the circuit board with the magnetometer. The hard iron effects shift the origin of the ideal sphere as seen in Fig 1. The Soft iron effects are due to the objects near the sensor which distort the surrounding magnetic field. These have the effect of stretching and tilting the sphere of ideal measurements. The resulting measurements lie on an ellipsoid. Fig 1 shows the Magnetometer's magnetic field x and y axis plotted before the calibration and contains the distortions due to Hard iron and soft iron effects.

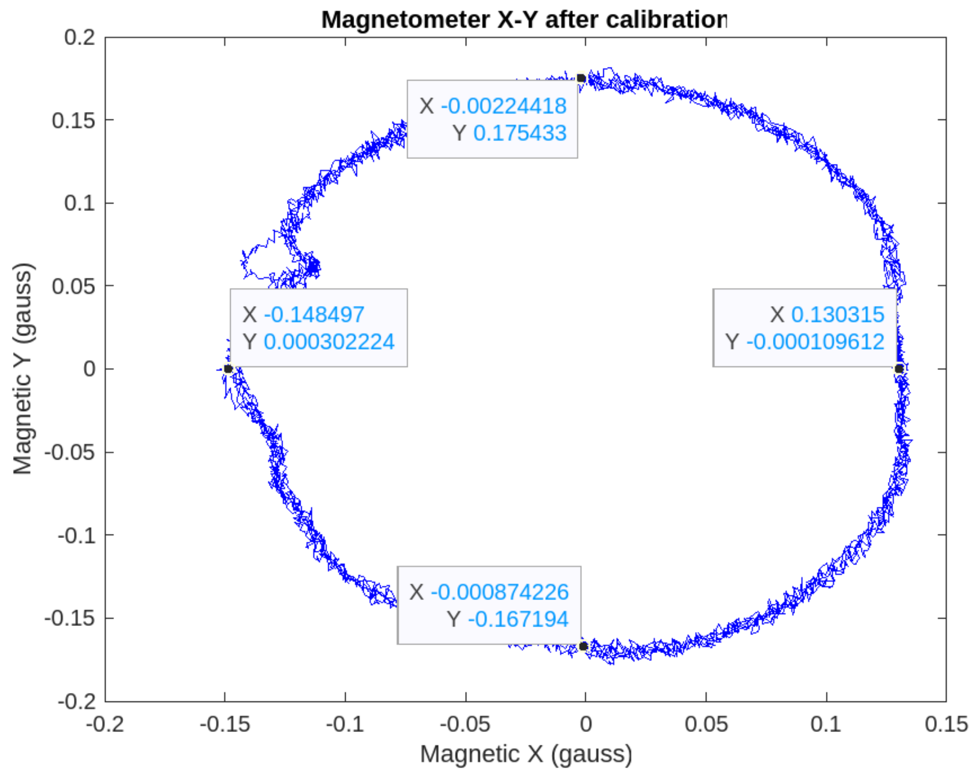

Fig 2: Magnetometer X-Y plot after calibration of circle data

The Fig 2 the Magnetometer's magnetic field x and y axis plotted after the calibration. This data is calibrated by shifting the center of ellipse to fit the origin this transformation is the translation operation. Then tilting the ellipsoid to be straight which is the Rotation operation of the transformation. The scaling of the ellipsoid’s minor axis or major axis to shape back to a circle.

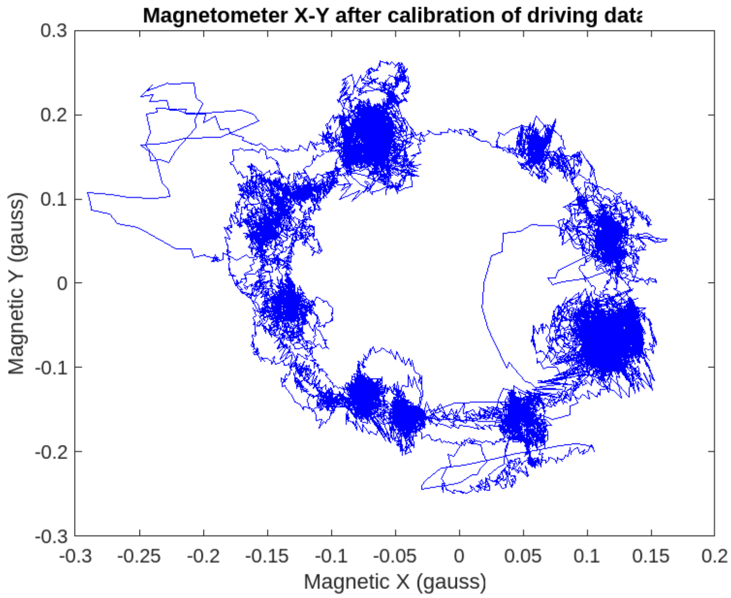

Fig 3: Magnetometer X-Y plot before calibration of driving data

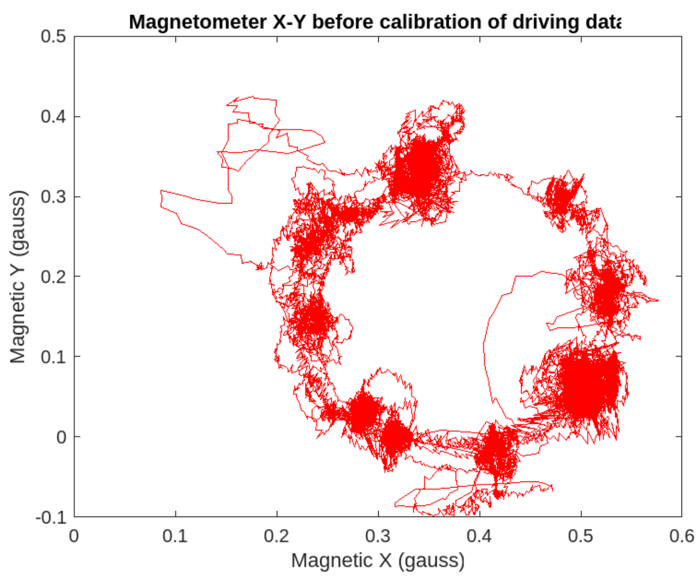

Fig 3: Magnetometer X-Y plot after calibration of driving data

The techniques used in the calibration of circle data are used to calibrate the driving data. Fig 3 depicts the Magnetometer's magnetic field x and y axis plotted before the calibration. Whereas fig 4 depicts the Magnetometer's magnetic field x and y axis plotted before the calibration.

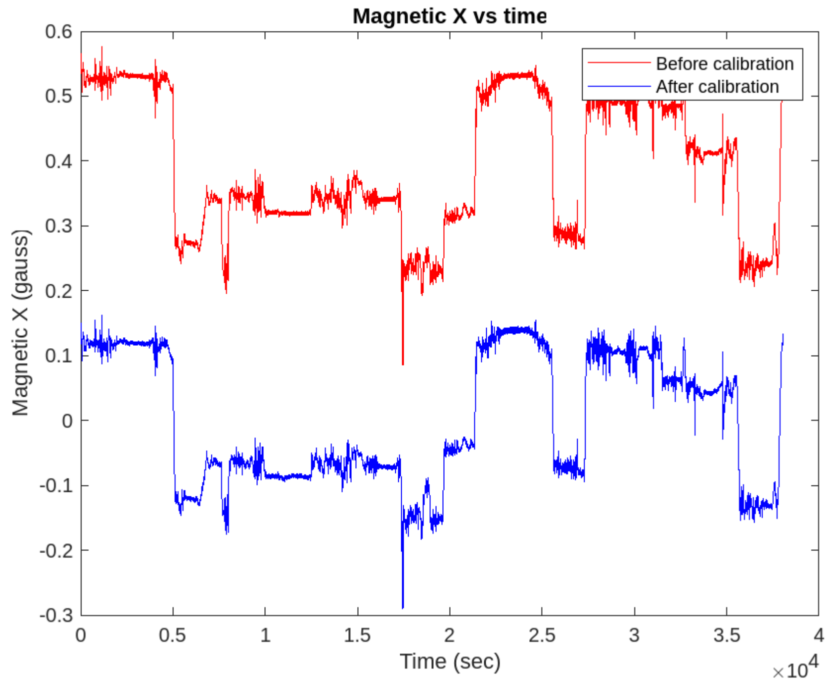

Fig 4: Time series plot for magnetometer magnetic x data before and after the calibration.

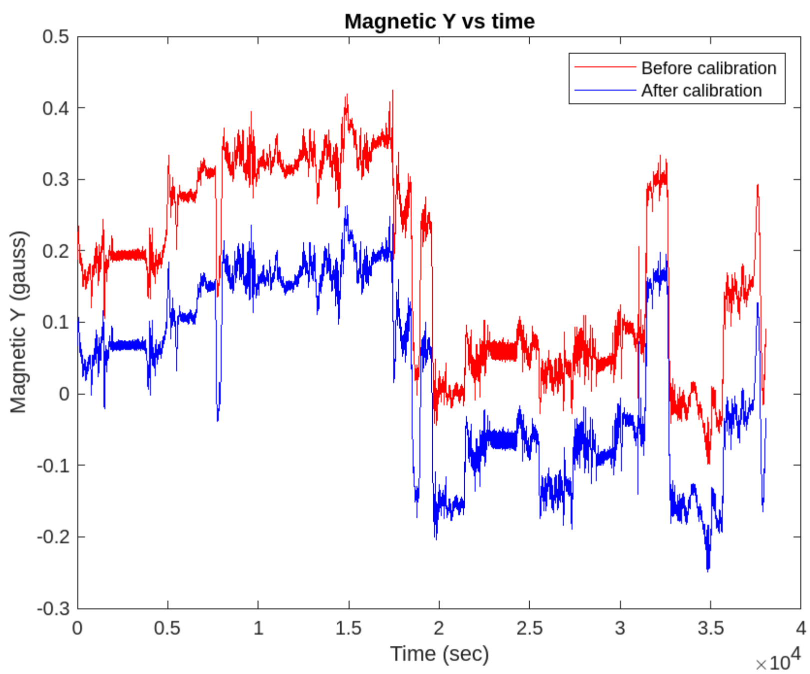

Fig 5: Time series plot for magnetometer magnetic y data before and after the calibration.

Figures 4 and 5 depict the time series plot for Magnetometer's magnetic x and y data plotted before and after the calibration.

To calculate the Magnetometer yaw we use the formula below.

$$yaw = atan2(-mag\_y / mag\_x).$$

To calculate the yaw from gyroscope we need to integrate the angular velocity about the z axis.

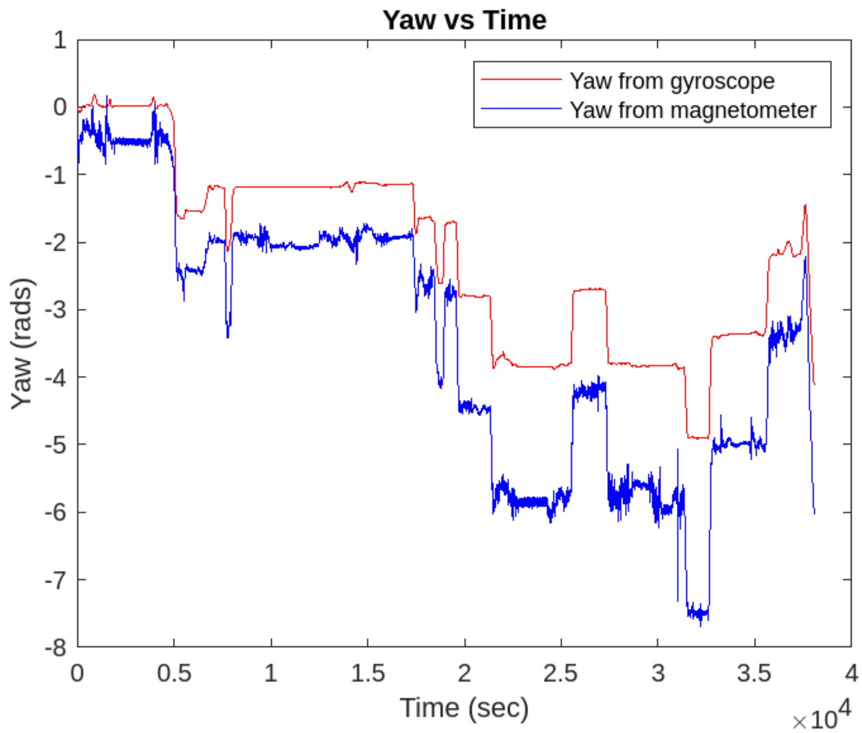

Fig 5: Plot of Yaw from magnetometer and gyroscope together

The above figure shows the Magnetometer Yaw & Yaw Integrated from Gyro together.

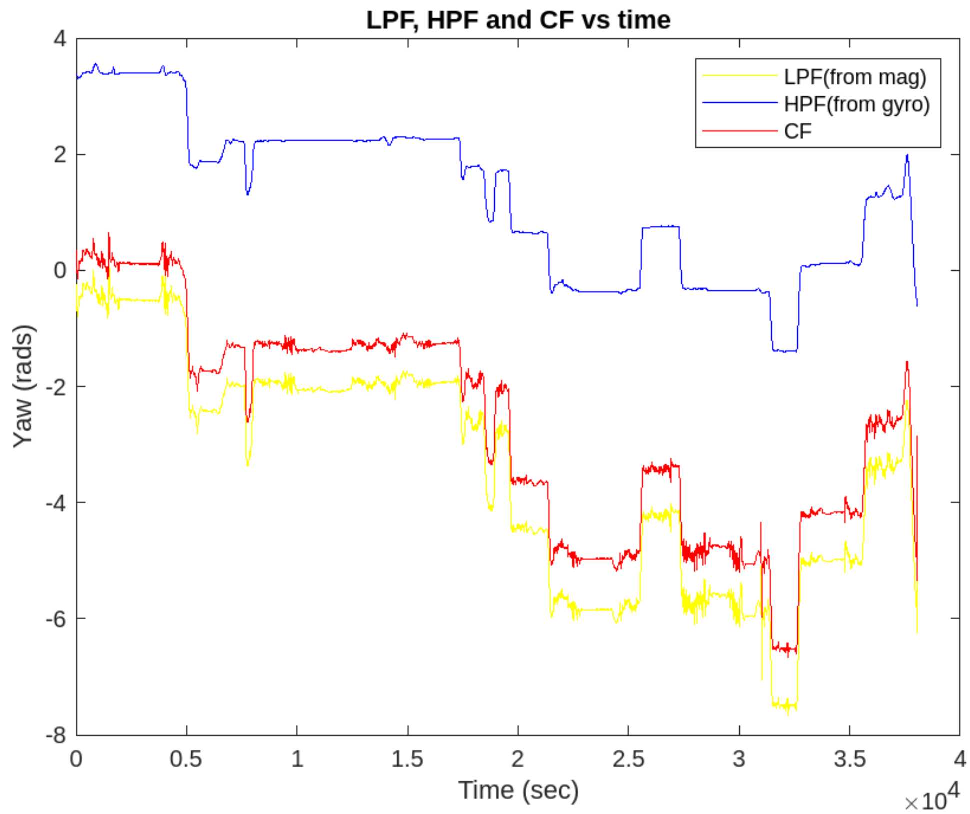

Fig 6: LPF, HPF, and CF plots.

The above figure depicts the Low pass filter from magnetometer, High pass filter from gyroscope and the complimentary filter data plotted against time.

The High pass filter is applied to the gyroscope data to pass high frequncy signals whereas the low pass filter is applied to the magnetometer data to pass low frequency signals. Here the High pass filter is used to sharpen the curve whereas the low pass filter is used to smoothen the curve. The cut-off frequency for the High pass filter is 0.00000001 rad/sample because the noises are present in such low frequencies and they are removed and 0.01 rad/sample. To find the final Yaw estimate, a complimentary filter is applied to both the yaw estimates after filtering them.

yaw_complimentary = (1 - alpha) * yaw_mag_lp + alpha * yaw_gyro_hp;

where,

yaw_complimentary = Yaw estimate

yaw_mag_lp = yaw from magnetometer after filtering

yaw_gyro_hp = yaw from gyroscope after filtering

alpha = 0.16

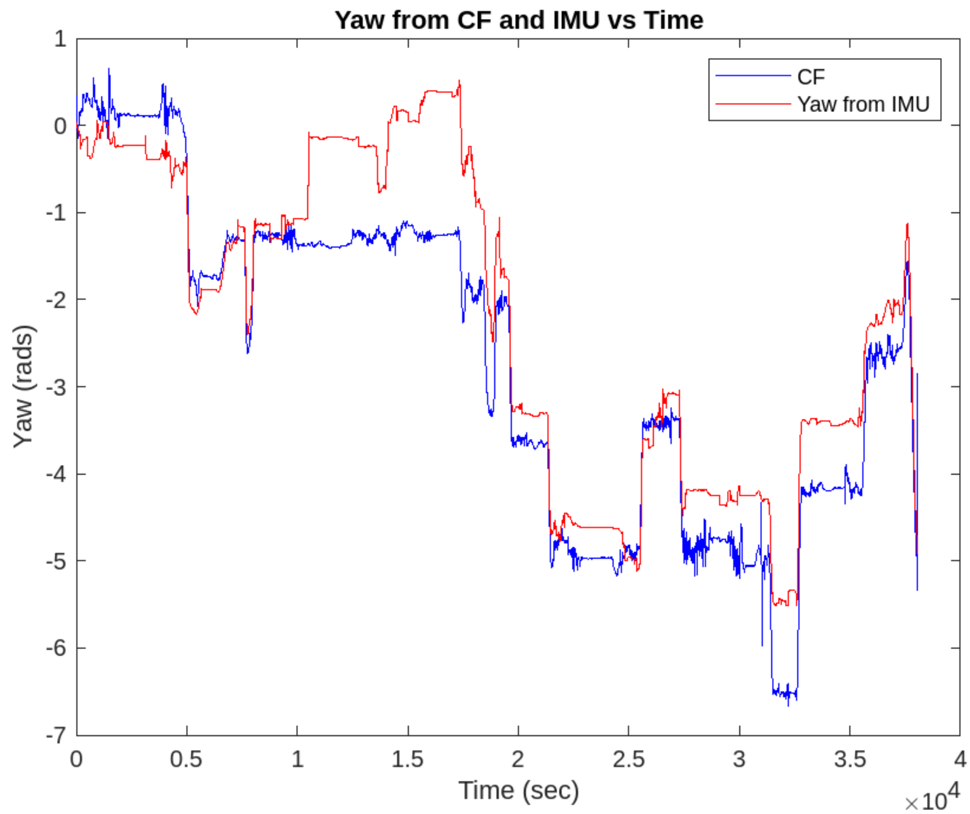

Fig 7: Yaw from the Complementary filter & Yaw angle computed by the IMU

The above plot describes the comparison between the yaw estimated from the complimentary filter and the actual yaw from IMU. When the yaw data is observed the yaw data taken from the magnetometer matches closely with the actual yaw from IMU. Based on this, the alpha value is chosen in a way that the estimated yaw from complimentary filter is close to the yaw data estimated from the magnetometer. Since calibrating the yaw from magnetometer is easier than as we use only transforming the ellipse to the circle whereas in order to correct the yaw from gyroscope, we need to consider Allan deviation techniques which is a tedious task compared to the magnetometer yaw correction. This implies that it is better to trust the yaw data from the magnetometer.

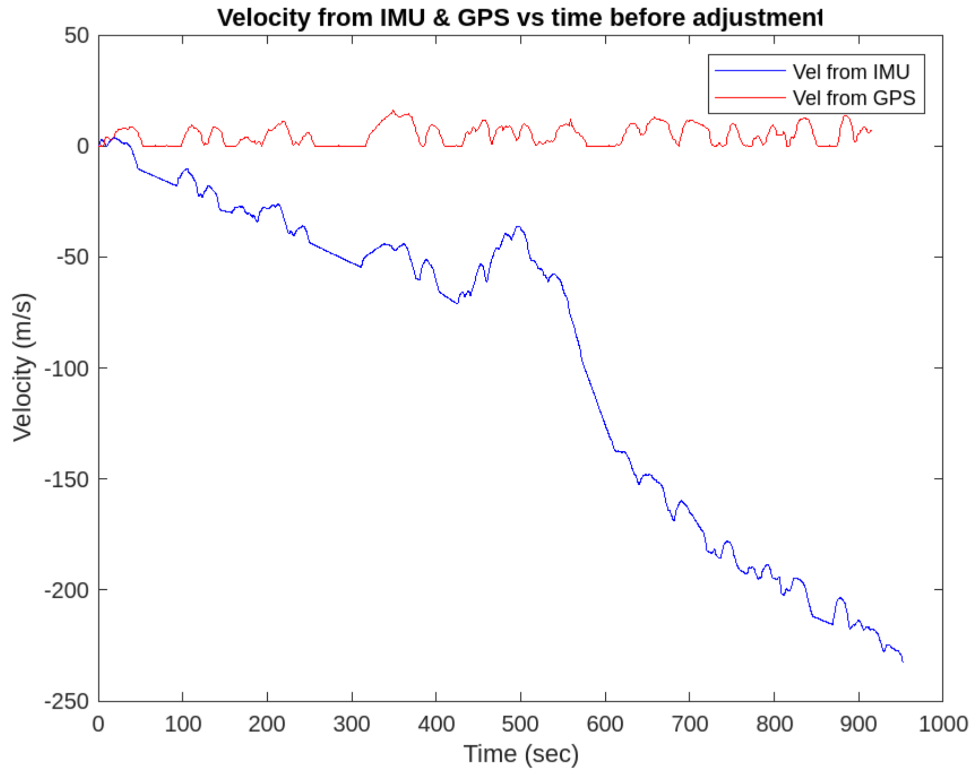

Fig 8: Velocity estimate from the GPS with Velocity estimate from accelerometer before adjustment

The above depicts the velocity estimate from the GPS with Velocity estimate from accelerometer before adjustment. The bias present in the velocity which is calculated from IMU is higher than the velocity calculated from the GPS. This is because of the Linear acceleration about x-axis as some contribution from the linear acceleration about y and z axis due to the inclination of the vehicle where the roll and pitch components come into pitcure. In roder to remove the bias present in the velocity data, the best fit line technique is used to find the variation of bias. Then this bias is subtracted from the velocity data curve. Since the data contains six groups of variations based on the change in slope so its split into these six groups. Then the best fit line is applied to each, and every group and the bias is subtracted. After this adjustment the result is shown in figure below which depicts the velocity estimate from the GPS with Velocity estimate from accelerometer after adjustment.

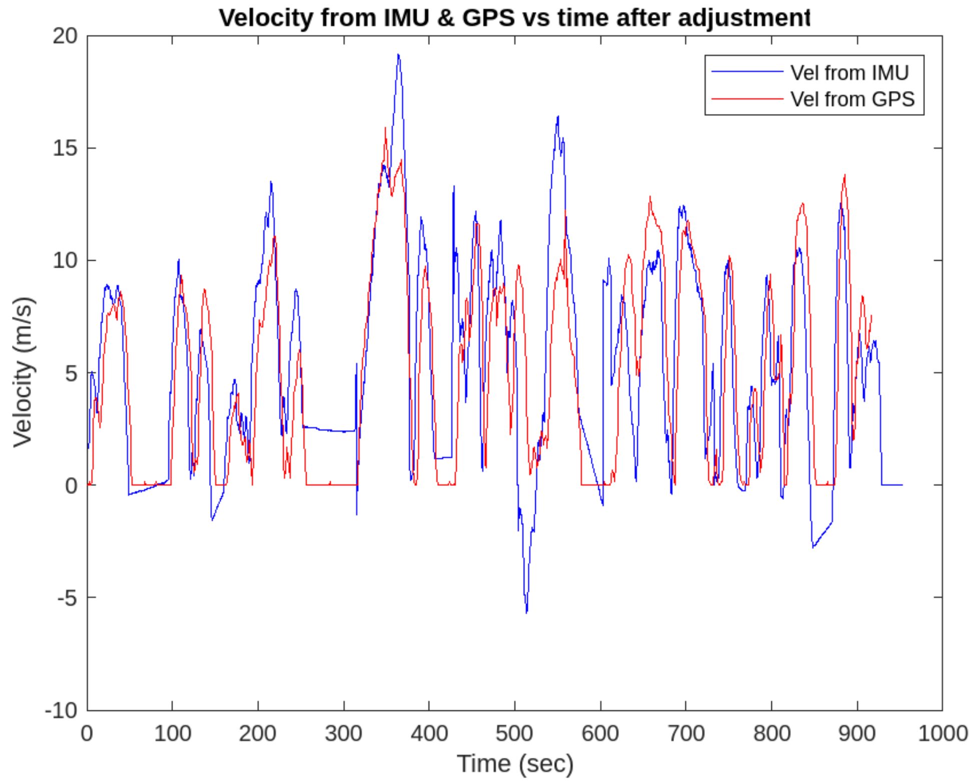

Fig 9: Velocity estimate from the GPS with Velocity estimate from accelerometer after adjustment

Dead Reckoning

The acceleration measured by the inertial sensor is given by the following equation:

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

Assuming the : $\dot{Y} = 0$ and $x_c = 0$,then the equation will be reduced to $\ddot{X} = \ddot{x}_{obs}$.

Integrating this to obtain the velocity about x-axis. Then compare the angular velocity times the Acceleration about x-axis and the acceleration about y-axis. As the calculated acceleration is distict when compared to the acceleration that is observed. This is because the observed contains the effects of acceleration due to gravity whereas the calculated acceleration is compensated for acceleration about y-axis by both the x and z-axis. The acceleration due to gravity which is about z-axis is reduced by applying the Low pass filter to the observed acceleration data.

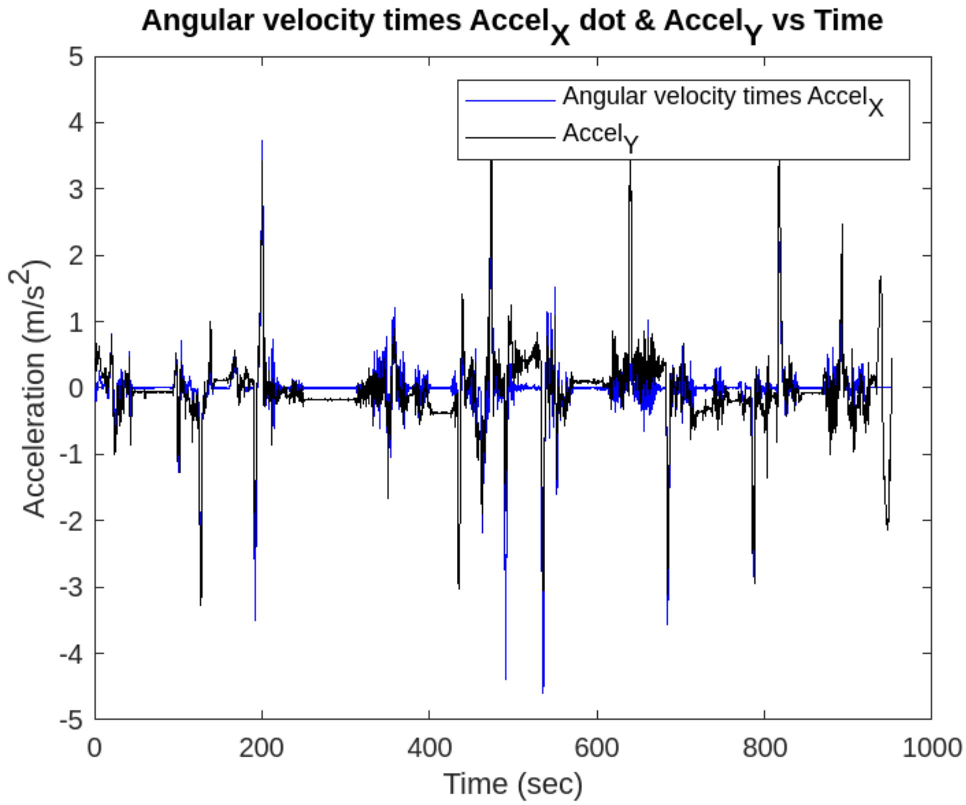

Fig 10: Angular velocity times the Acceleration about x-axis and y-axis

The above shows the result of the filtered observed acceleration data. Here we can see that the observed acceleration seems to be matching closely with the one calculated acceleration.

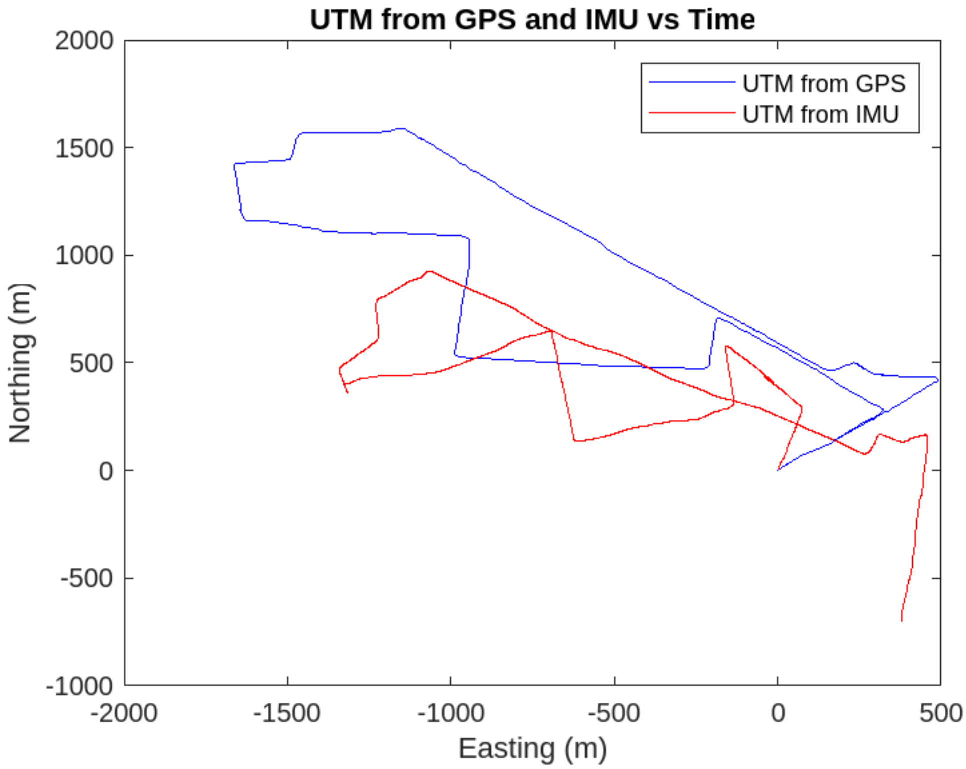

Fig 11: UTM from GPS and IMU

When UTM from GPS and IMU are compared the data seems to be slightly different, but it almost follows the trend in a similar manner. The reason for this is because of using the data from the IMU to find the UTM displacements. Since the UTM displacements are calculated based on the estimates which already contain fluctuations in the data which will be affecting the calculations of displacement values hence the tracks are slightly different. Given the specifications of the VectorNav, the scaling factors used to compare the tracks are the UTM values of IMU should match that of GPS but the tracks are deviated from the starting point but as I mentioned above the trends are almost similar. I would expect that it is able to nagivate atleast half of the path without getting a position fix. Hence the stated performance for the dead reckoning did not match the actual measurements.