Lab 4: Navigation with IMU and Magnetometer

1. Magnetometer Calibration

From the raw magnetometer plot (fig 1.1) we can see that the data points that should form a circle are actually forming an ellipse due to **Hard iron** and **Soft iron** distortions. Hard iron errors are caused by nearby magnetized or magnetic materials that are fixed with respect to the frame of IMU i.e. fixed to the car. These include components like speakers in the car, and magnets present in other electronic components of the car. Soft iron errors are caused by distortion of magnetic field due to presence of ferromagnetic materials like iron and nickel. The sources of these errors include the car body, car chassis and other components of the car made from ferromagnetic materials. These errors can change with change in surroundings, however we can compensate for these errors with respect to the car.

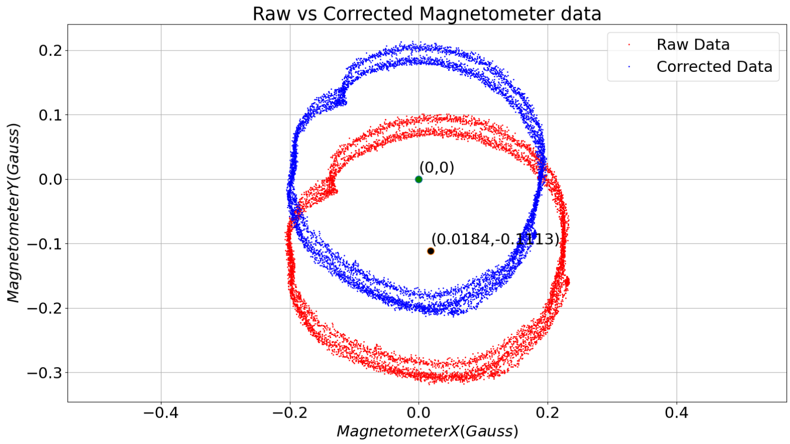

Fig 1.1 Raw vs Calibrated Magnetometer Data

To correct for hard and soft iron errors, first an ellipse is fit on the raw magnetometer data. The center (Xc,Yc) ,height and width of the ellipse can be used to compensate for errors. The hard iron error is rectified by shifting the origin to the center of the ellipse i.e. subtract Xc from all X values and Yc from all Y values. The soft iron error can be rectified in two steps - rotating the ellipse, and scaling the ellipse to obtain a circle. We find the rotation matrix and scaling matrix by the formulas mentioned below.

For this case, theta was 10.864 degrees and Sigma was 0.89946. We rectify the points by multiplying in the order Mscale*Mrot*Mt.

2. Yaw Estimation

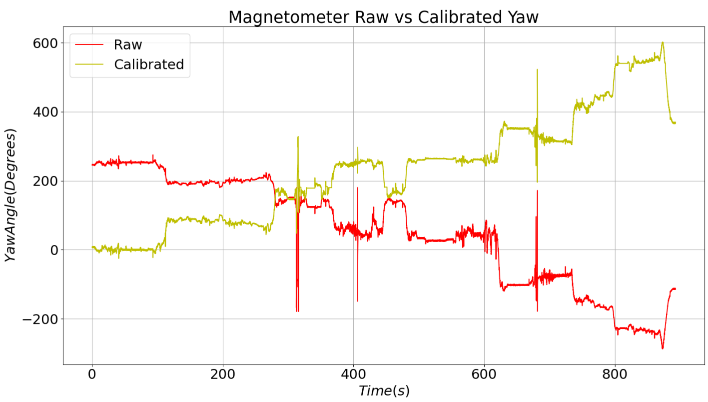

Fig 2.1 Raw vs Calibrated Magnetometer Yaw

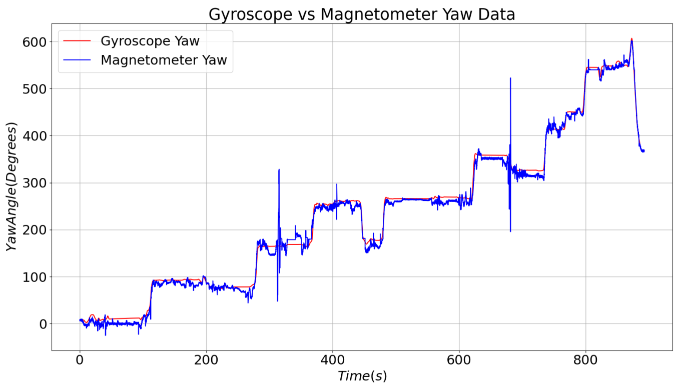

Fig 2.2 Gyroscope vs Magnetometer Yaw

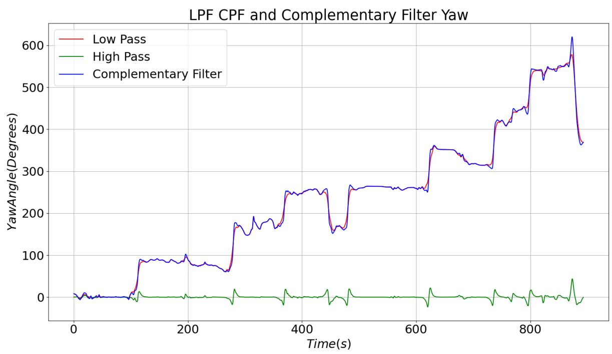

Fig 2.3 LPF HPF and CPF

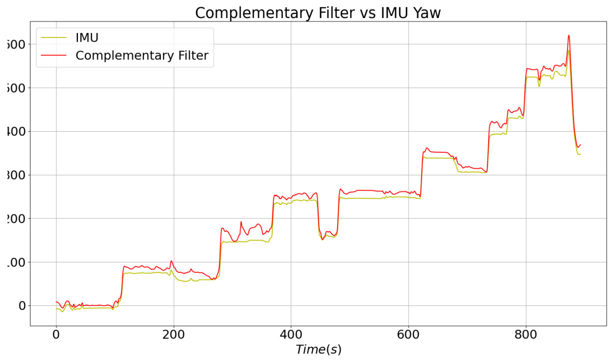

Fig 2.4 IMU Yaw Angles vs CPF Output

Please note that all Yaw comparisons have been done with unwrapped values for ease of comparison.
Fig 2.1 shows the comparison of yaw from uncalibrated vs calibrated Magnetometer. As expected the values do not match as calibration changes the magnetometer values.
Yaw estimation was done by combining the yaw angles obtained by magnetometer and integration of angular rate about z from gyroscope, by using a complementary filter. The complementary filter has a low frequency component and a high frequency component. We get the low frequency component by passing the yaw angles obtained from the magnetometer through a low pass filter. The high frequency component is obtained by passing the yaw angles obtained from the gyroscope through a high pass filter. The low pass cut off frequency is 0.08 Hz and the high pass cut off frequency is 0.05 Hz. Fig 2.3 shows the mentioned implementation of LPF, HPF and CPF. The complementary filter output is calculated by the below equation.

CPF = LPF+ HPF

The output of the complementary filter is a more reliable way to estimate heading rather than using only Magnetometer or only gyroscope. This is because the magnetometer can be relied on for only slow changes in angle as it contains high frequency noise (as seen in fig 2.2) that needs to be filtered out. Due to this high frequency noise, rapid changes cannot be detected as they are mixed up with the noise. Similarly, the gyroscope can be relied on for rapid changes in the angle, but not for slow changes over time as it has low frequency noises. There is also the problem of drift due to inherent errors that could drastically change the heading in the long run. Since we get change in angle for a unit time using the gyroscope, we need to use the magnetometer values to get an absolute estimate. Thus the output of complementary filter gives us the best estimate out of all other available methods. We can also see in fig 2.4 that output of CPF closely matches with the output of the IMU which uses a Kalman Filter.

3. Forward Velocity Estimate

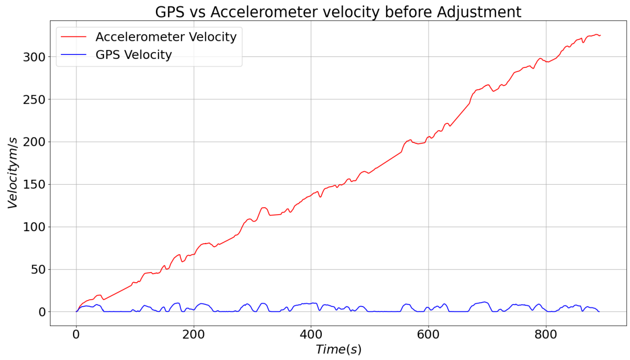

Fig 3.1 Velocity from raw accelerometer values

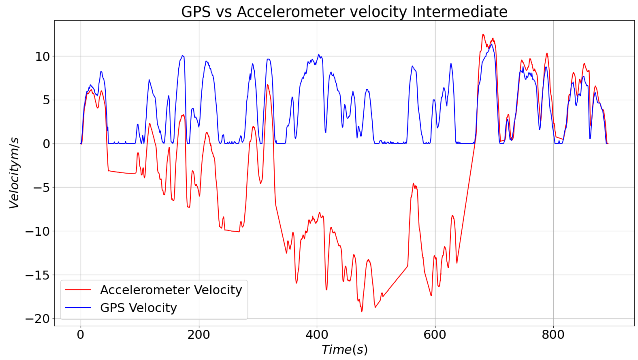

Fig 3.2 Velocity from intermediate corrected accelerometer values

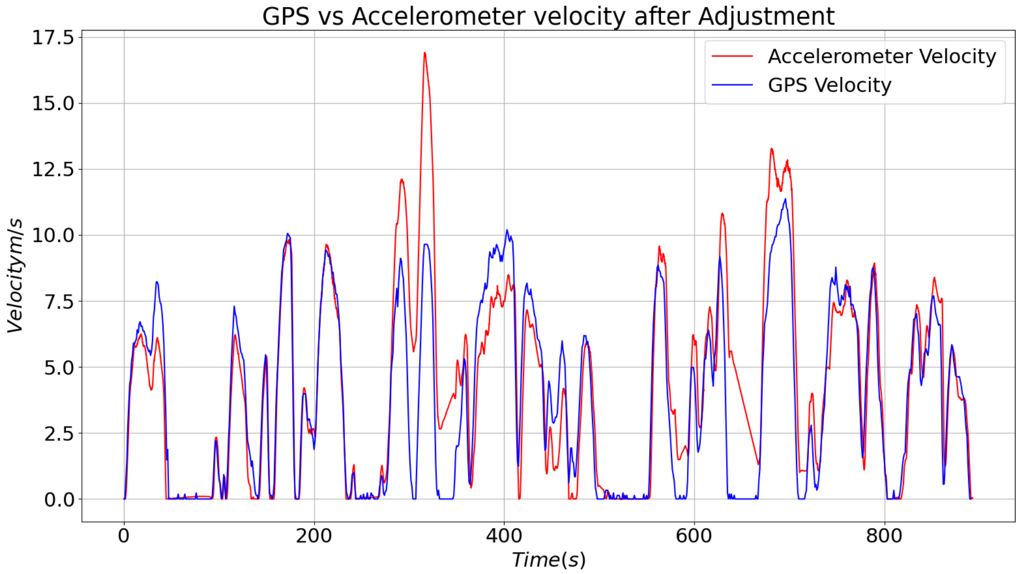

Fig 3.3 Velocity from corrected accelerometer values

To find the forward velocity from GPS, we need to find the euclidian distance between every two consecutive readings and divide it by the time elapsed between those two readings (dx/dt). To find the forward velocity from acceleration, we need to integrate the acceleration plot. However, as seen in fig 3.1, the velocity almost linearly keeps increasing. This is due to a significant bias present at the start which keeps amplifying with each iteration of the integration loop. To remove this bias, subtract the value of mean of all the acceleration points and subtract it from the entire acceleration dataset. This is because the average acceleration of the entire trip should be equal to zero as there is no change in velocity at the end of the trip. After integrating this corrected value, it is observed in fig 3.2 that velocity shoots into negative direction every time the car comes to a stop. This can be explained by the backward jerk exerted by the suspension of the car after the car comes to stop. The jerk is felt but the car doesn't move because brakes are applied. This is fixed by adding a condition during integration that prevents velocity from going negative (we are not driving in reverse anywhere so this does not affect our other data). The data still does not match with GPS velocity as there are more biases that start showing up at different points when we stop the

car. To fix this, the time period when the car is idle is identified by checking when jerk values are less than 0.2. When the condition is satisfied, the mean of those data points in acceleration is subtracted to remove the bias of that region. It is observed that this bias is seen when the car is on a slope. The curves match when bias is removed only when the car is stationary and not from the non zero values of acceleration. As we have a non zero value of X acceleration when stationary on a slope, integration gives a non zero velocity value. However, when the car starts moving, the X acceleration value is correct as the component of gravity also aids in acceleration or retardation of the car. Hence the bias needs to be removed only when the car is stationary in this particular case as the offset in acceleration is only caused when the car is stationary on a slope.
The deviation of this computed velocity from the GPS velocity is caused mainly due to noise in acceleration data, jerks while stopping and starting the motion of the car, and gravitational component in X direction when the car is at a stop on a slope.

4. Y Acceleration

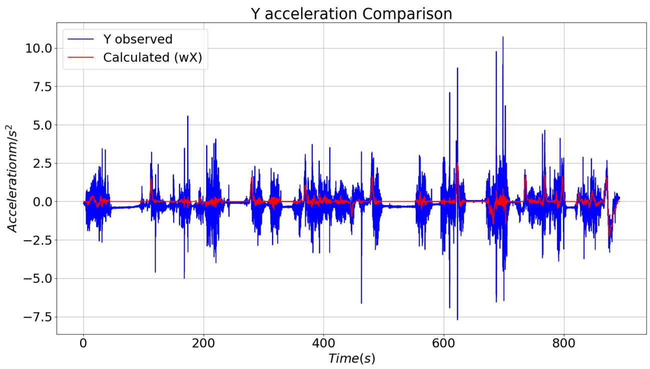

Fig 4.1 Y acceleration - Estimated vs Observed

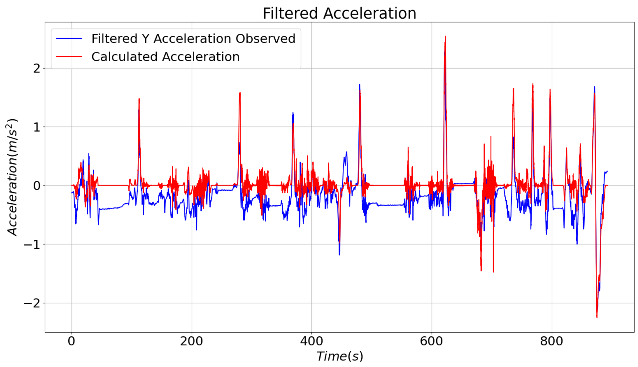

Fig 4.2 Filtered Y Acceleration

Y acceleration is only seen when the car takes a turn. The estimated Y acceleration is calculated by multiplying the angular velocity from the gyroscope and the X velocity estimate from the accelerometer. Fig 4.1 shows that Y acceleration from the accelerometer has a lot of high frequency noise just like the magnetometer data. The angular velocity from the gyroscope has no high frequency noise and gives stable readings. Similarly the X velocity after corrections is almost the same as the GPS velocity and has no high frequency noise. Comparing the two plots, we can see that the values match with the only difference being less noise in the estimate obtained from the gyroscope and linear velocity. To test this hypothesis, the Y acceleration values from the accelerometer are passed through a low pass filter with cut off frequency 0.5 hz. Comparing the filtered values with estimated values (fig 4.2), we see that they match very closely except for some offsets in a few places. We just need to remove the stationary bias by using the same method used for X acceleration.

5. Trajectory Estimation

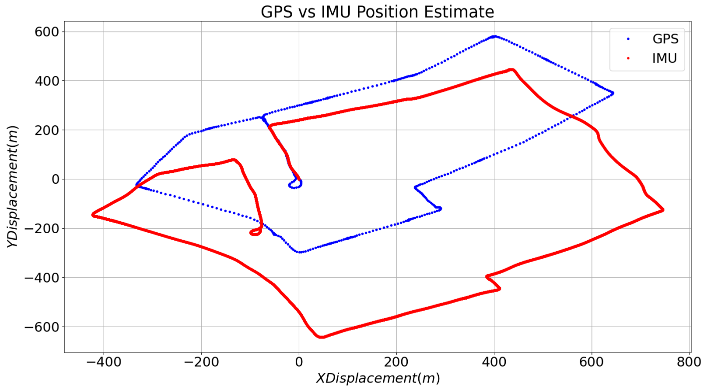

Fig 5.1 GPS vs Estimated Position

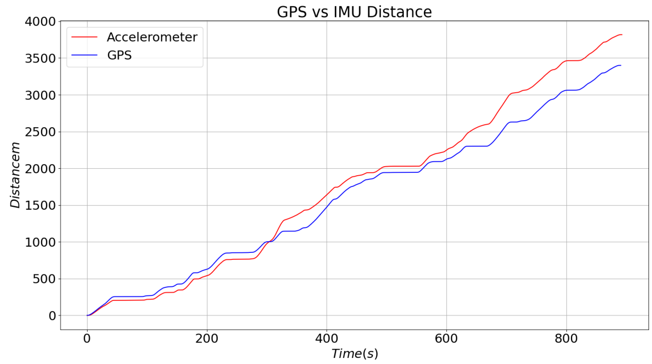

Fig 5.2 GPS vs Estimated Distance

To compare the trajectories, no scaling factor was used. The heading angles were first mirrored (offset by 180 degrees) since the Z axis of IMU points downward, towards the floor of the car, whereas the Z axis for the car points towards the roof of the car. An additional angle was added to match the initial angle with the initial angle of the car. Comparing the two plots, it is observed that the trajectories matched initially but slowly drifted apart more and more with time. This is observed because the distance is obtained by integrating acceleration twice. Even though errors in acceleration were rectified to a fair extent before calculating velocity, there were some errors in the velocity estimate that got amplified when velocity was integrated to get displacement. Further calculating the length of two paths for comparison, it is observed that the distance obtained by integrating forward velocity is 400 m more than the distance calculated using the GPS position.
Calculating the deviation of estimated position value from GPS position, it is observed that it stays within a 2m range of the GPS position for around 10 seconds. After 10 seconds the gap widens and the position is not reliable. In absence of GPS, the calculated estimate can be only trusted upto 10 seconds which is very low.
In the performance specifications, the VN-100 IMU is supposed to have a 0.7m position error in 10s. That means it can stay within the mentioned 2 meter tolerance for 30 seconds. However, it must be noted that VN-100 uses a Kalman filter for heading estimation which is a more sophisticated method whereas we use a complementary filter. It might also be using sophisticated methods for magnetometer calibration as well as for dead reckoning which is why the observed measurement for dead reckoning does not match the mentioned performance.

<u>Note:</u>
For group one (Ronak Bhanushali and Nitin Somshekhar) I (Ronak Bhanushali) have submitted the driver files.
gitlab username- bhanushali.r