Lab 4

Aryaman Patel

Preamble

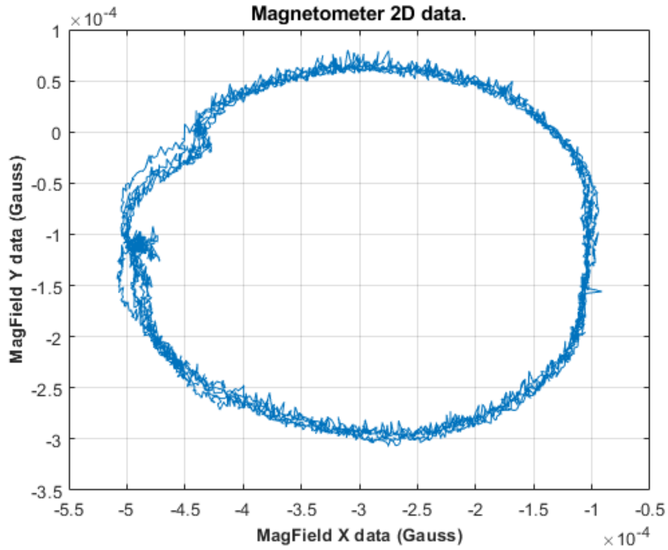
In this lab we collected the GPS and IMU data, to build a navigation stack and perform dead reckoning. The driver has been uploaded by burwell.c

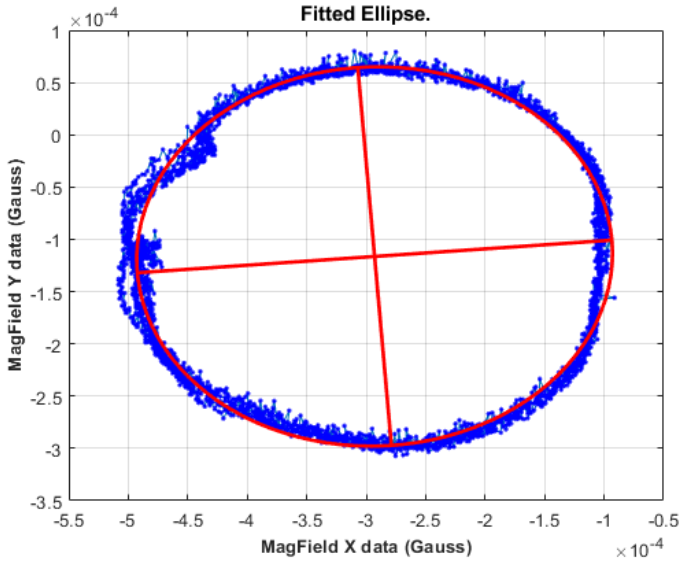
Questions:

1. **How did you calibrate the magnetometer from the data you collected? What were the sources of distortions present, and how did you know?**
   The magnetometer can have the following internal/external disturbances:
   a. Constant unwanted magnetic fields (hard iron) which are produced by materials with a constant magnetic field, in addition to the earth's field these generate a constant additive value to the output of the magnetometer axis, resulting in an offset from the origin. If the source of this additive magnetic field is constant the associated offset will also remain constant.
   b. Ferromagnetic Materials (soft iron) unlike hard iron distortions do not produce magnetic fields of their own but influence the magnetic fields around them, due to their properties. These errors are not additive in nature rather affect the data in terms of offsets in orientation and convert the ideal circle to an ellipse.

   These errors have been compensated in the following literature using the following method:

   a. First, we find the 'best-fit' ellipse to the uncalibrated data, this is done by finding the general conics equation that best describes the ellipse of the given points.



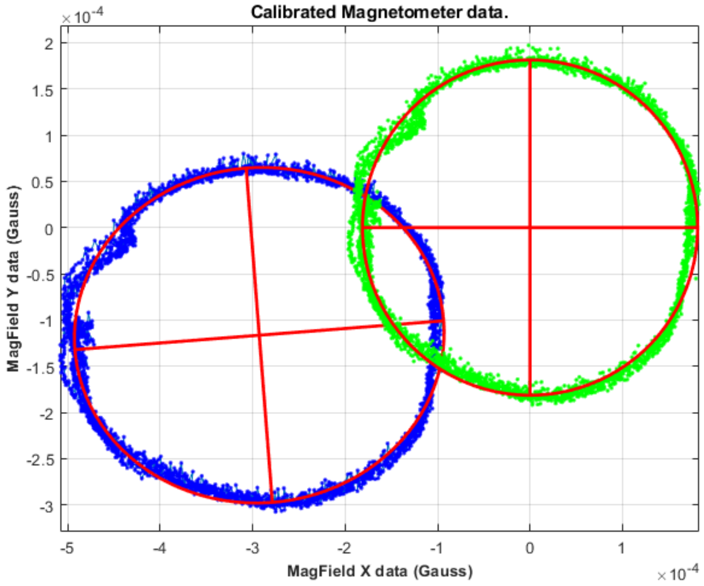


For the calibration data following were the parameters of the ellipse:

```
fitting = struct with fields:
a: 2.0009e-04
b: 1.8134e-04
phi: -0.0780
X0: -3.0113e-04
Y0: -9.3207e-05
X0_in: -2.9294e-04
Y0_in: -1.1640e-04
long_axis: 4.0017e-04
short_axis: 3.6268e-04
status: ''
```

b. After finding these parameters we first correct for the hard iron offset by translating all the points on the ellipse to the origin. We do this by subtracting the X0 and Y0 values from all the data points.

c. Once the ellipse is cantered around the origin, we apply a reverse rotation transformation to the data points by constructing a rotation matrix using the 'phi' value. Following that we remove the long axis and short axis distortions to get a near perfect circle around the origin.



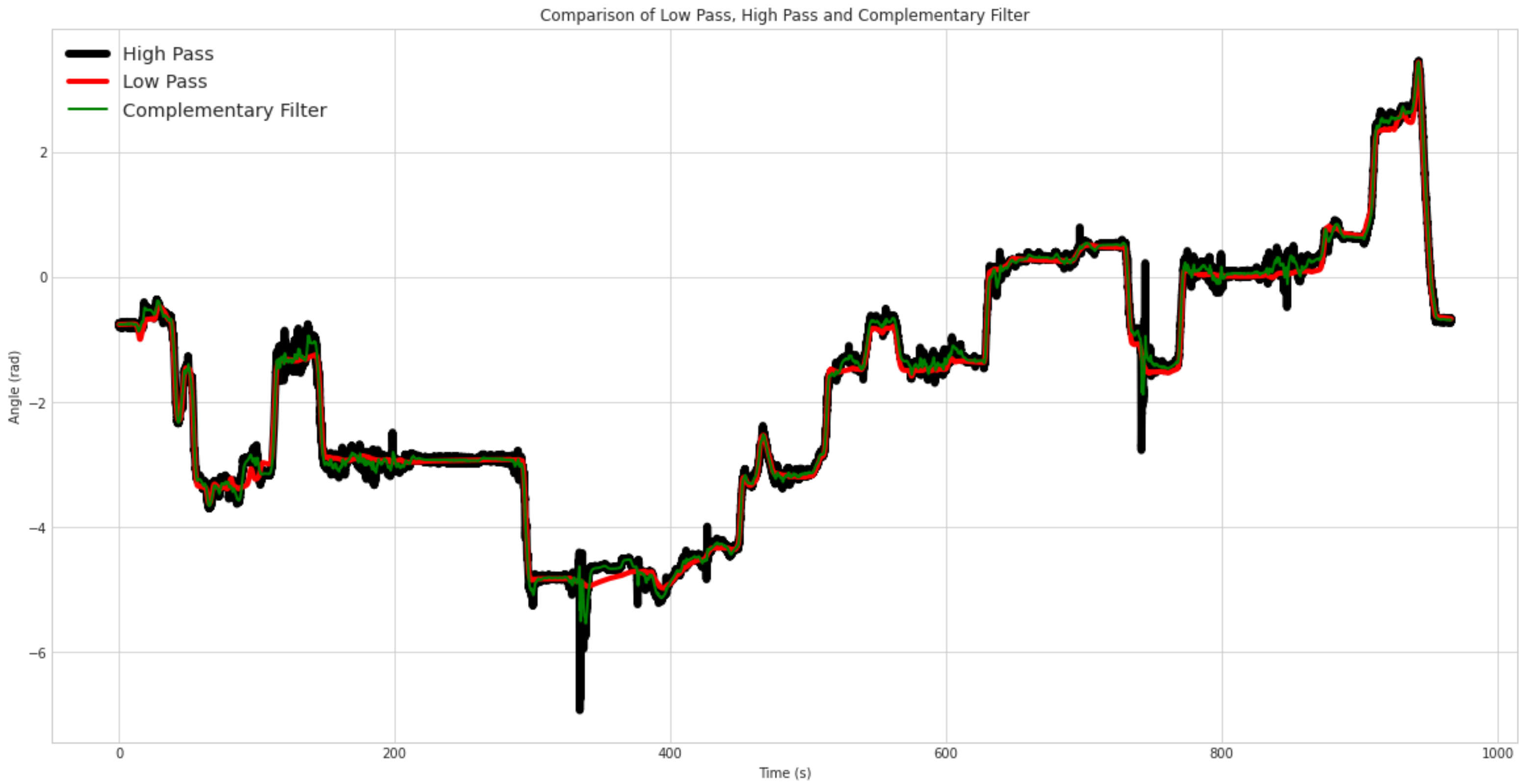
2. **How did you use the complementary filter to develop a combined estimate of the yaw? What components of the filter were present, and what cut-off frequencies did you use?**

The formula of the complimentary filter applied to get the yaw value is as follows:

$$\theta_{k+1} = \alpha(\theta_k + \omega\Delta t) + (1 - \alpha)\theta_{accel}$$

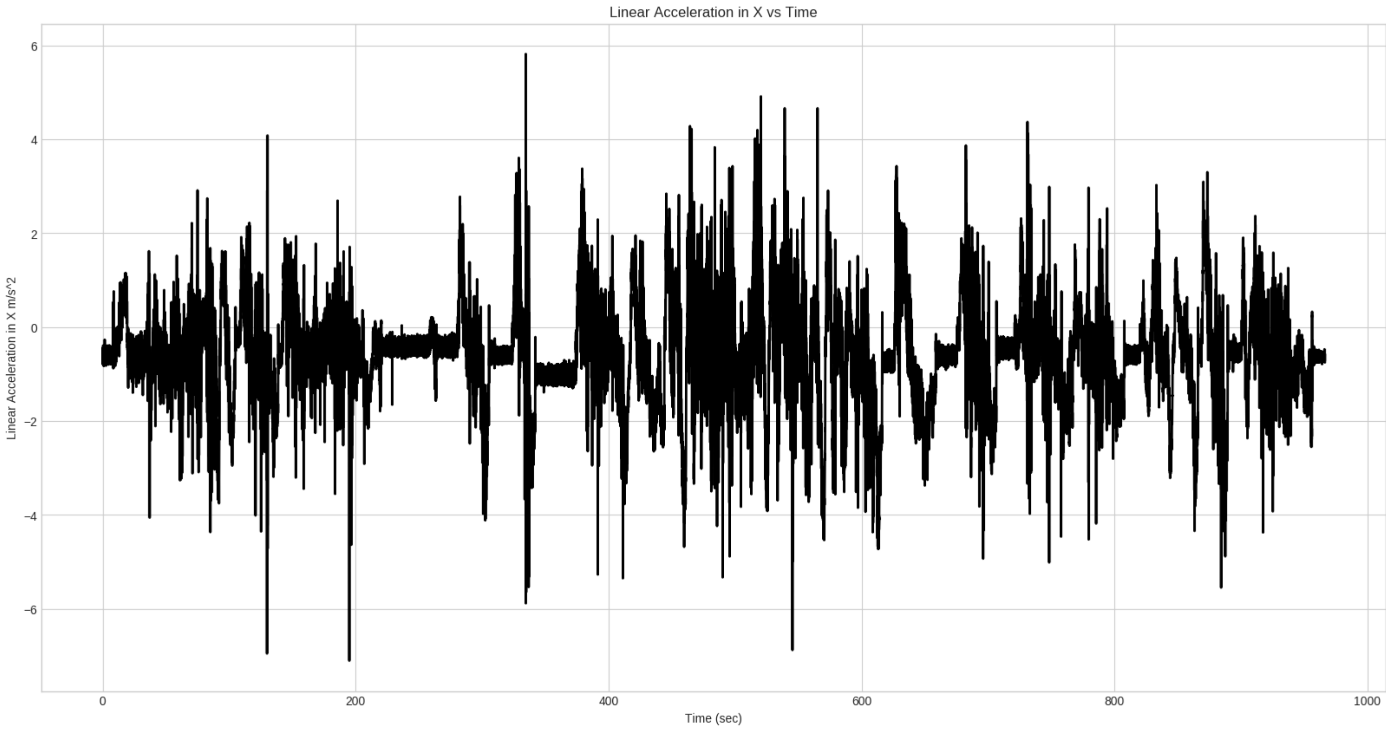
We applied the equation to using the magnetometer and gyro as compared to accelerometer and gyro. The initial condition, θ_k was taken to be the mean of the first 100 values of the magnetometer data. The angular velocity (z) was taken and multiplied by the difference in time. The θ_{accel} value was taken to be the corrected magnetometer yaw values.

The following is the graph with the alpha as 0.001, 0.999 and 0.97 for the respective low-pass, high-pass, and complimentary filtering.

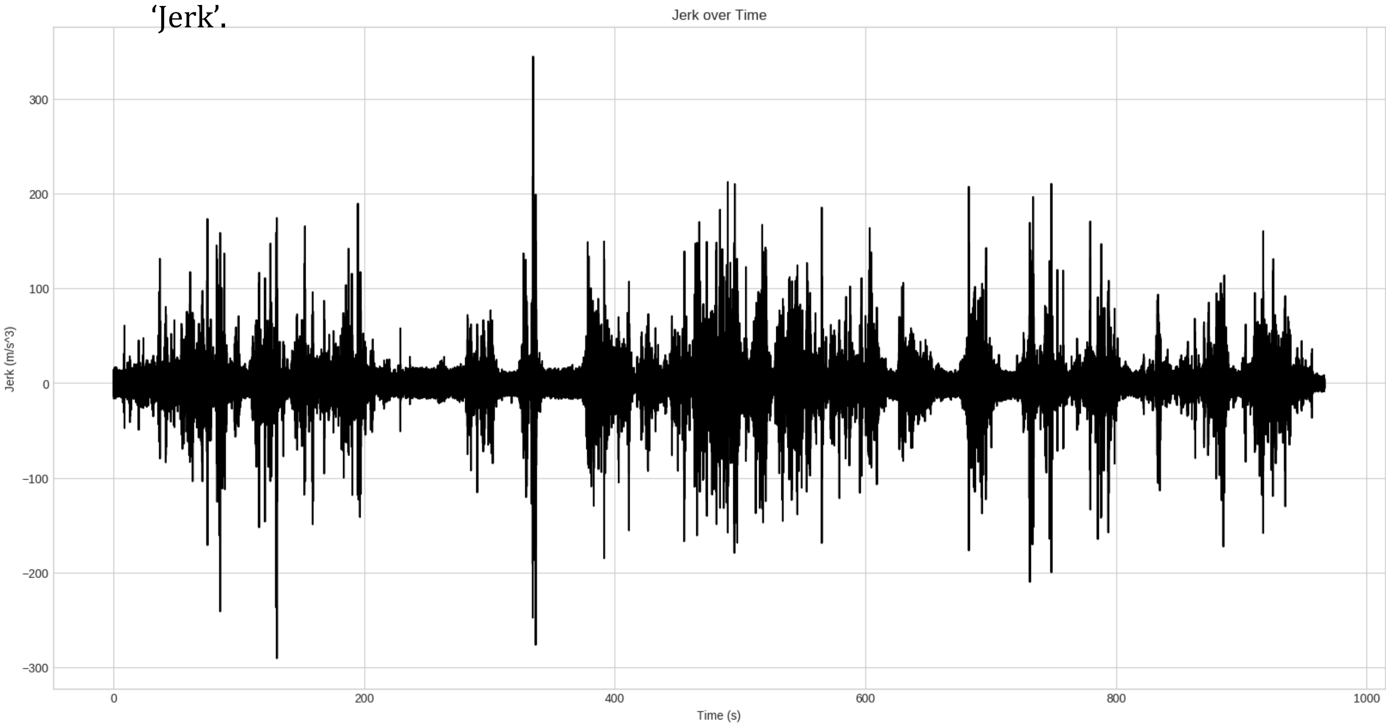


3. **Which estimates for yaw would you trust for navigation? Why?**
   The corrected magnetometer yaw angles would be a preferred estimates for the heading of the navigation of the vehicle. The magnetometer gives the orientation of an object with respect to the earth's magnetic field (like a compass). True, there can be distortions, as we have seen previously, in terms of soft and hard iron, but these distortions can easily be corrected for. On the other had IMU estimates of yaw can accumulate a lot of different biases or drift over time, which can throw off the estimates. These values integrated over time will add up the biases that may or may not further affect the future values. This makes it hard to process the data and get the correct state estimates.

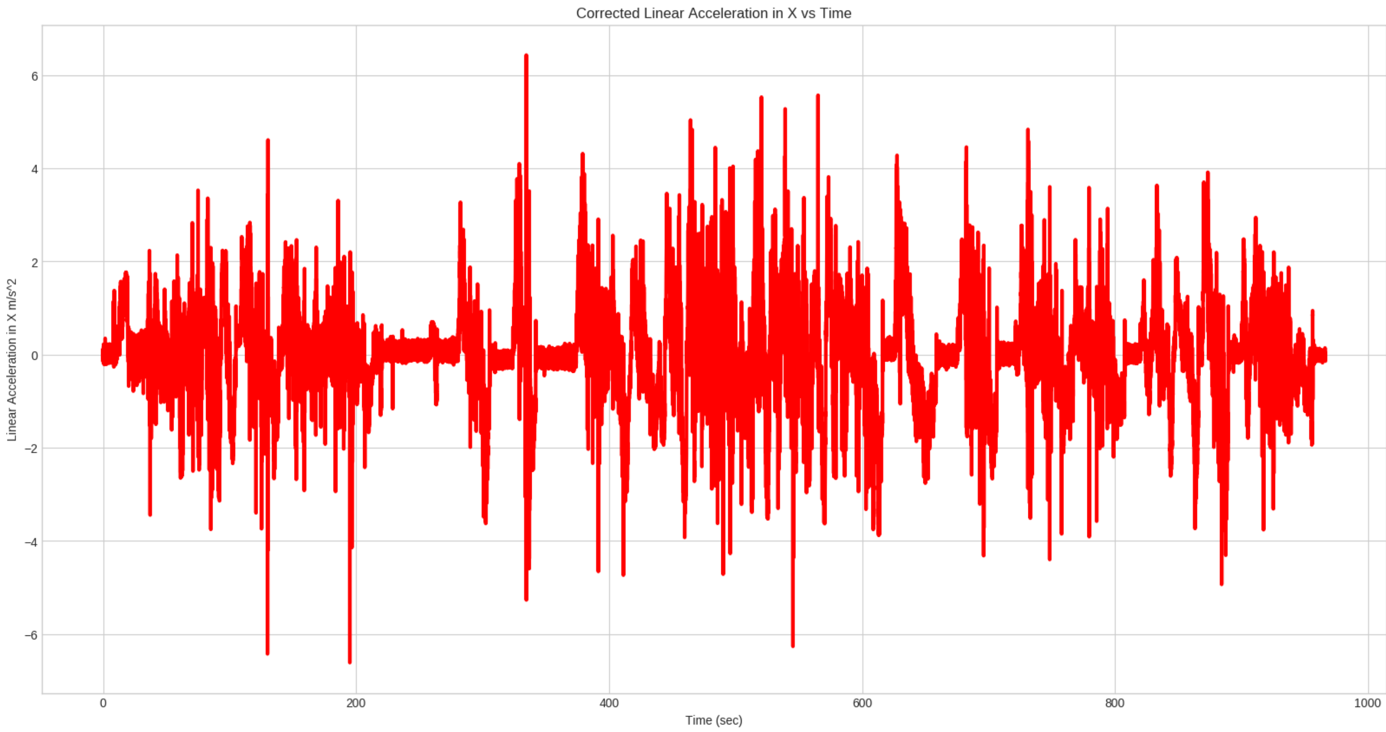
4. **What adjustments did you make to the forward velocity estimate and why?**
   The following adjustments were made to correct for the biases in the data:
   a. The below figure is the raw acceleration values –



From the graph one can observe certain offsets in the data, specifically in the stationary zones. To find these timestamps the data was differentiated with respect to time to get the 'Jerk'.

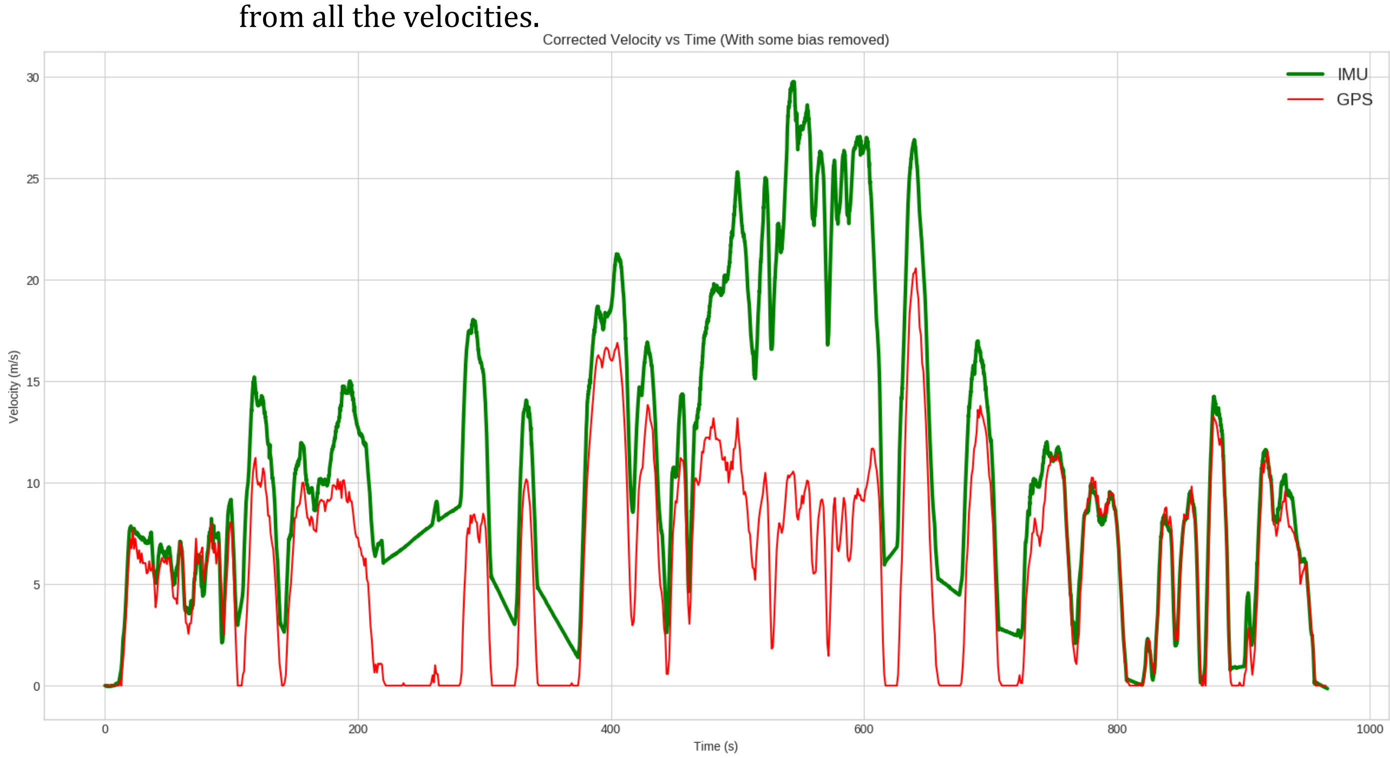


From this data we can estimate that the stationary zones are in the range of +- 10. Isolating these points, I was able to extract 22 such zones where the vehicle was stationary. Taking these values, I calculated the mean of these each zone.
The important thing to note is that there are some biases that are being added to the entire data, one such was the initial offset of the accelerometer data while the vehicle was stationary.
Taking all these points the corrections were made at the specific zones or the entirety of the accelerometer data. The following is the accelerometer plot after the corrections were made.



These changes were made to the accelerometer data rather than the velocity estimates as it would make the process easier to evaluate and correct for. The corrected velocity can then be calculated by integrating the acceleration over time.
There were also adjustments made as to the addition of the gravitational component in the data as the vehicle travels over the slopes this was done by,

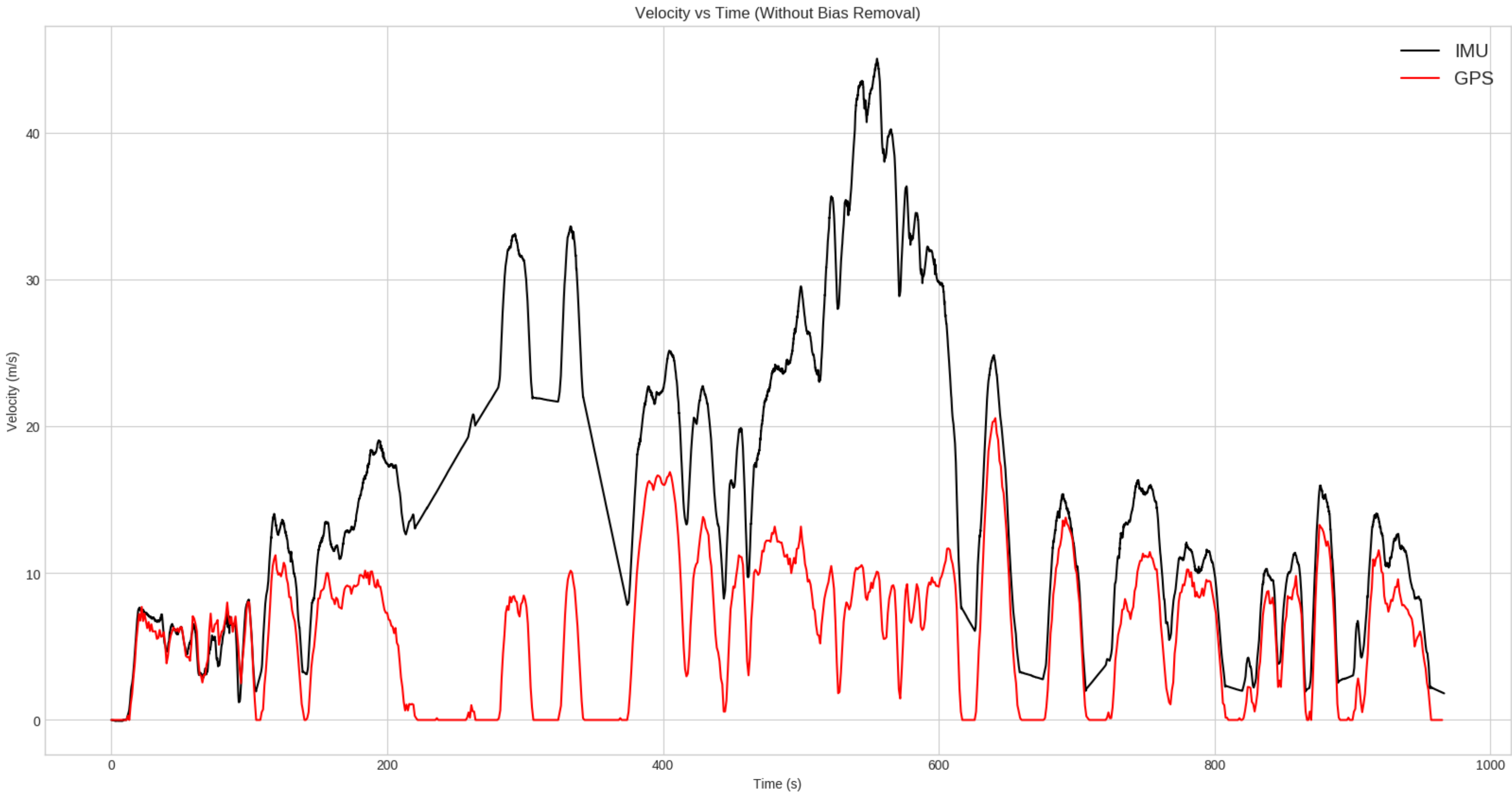
a. Calculating the pitch angles
b. Taking the sine component of the pitch and multiplying with 9.81 and subtracting it from all the velocities.



5. **What discrepancies are present in the velocity estimates between the IMU and GPS data? Why?**

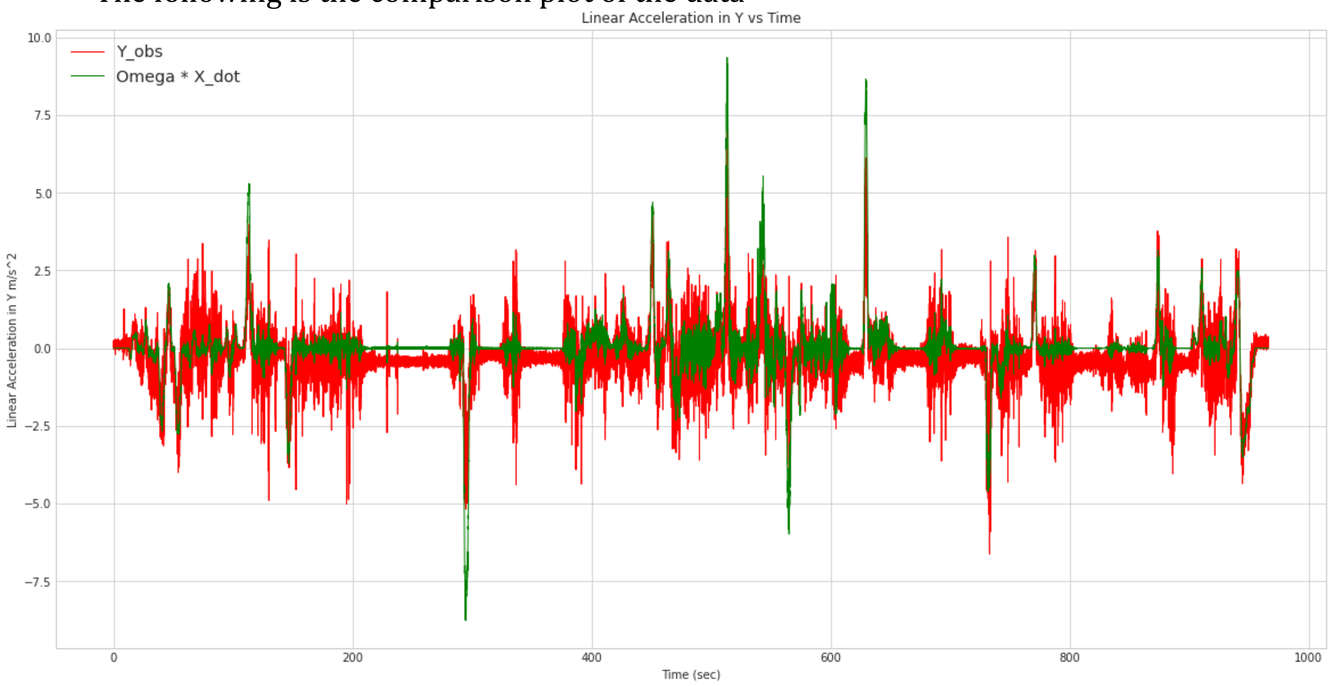
   From the following plot of the velocity comparison between the IMU and GPS data one can see the following discrepancies:

   - There is a bias in the accelerometer between the 200 and 300 secs mark this could be due to the added bias in the beginning of the data wherein the vehicle is stationary, however the velocity is non-zero.
   - There were huge jerks in the stopping that cause the IMU to have additional offsets as it disturbs its levelling.
   - There is a heavy hill climb in the timestamp between 400-600 secs where in the IMU velocity has bumped up as the component of the gravity gets added to it.
   - The velocity of the IMU and GPS does not match each other when the car is stationary., which could be due to the previous biases in the data.



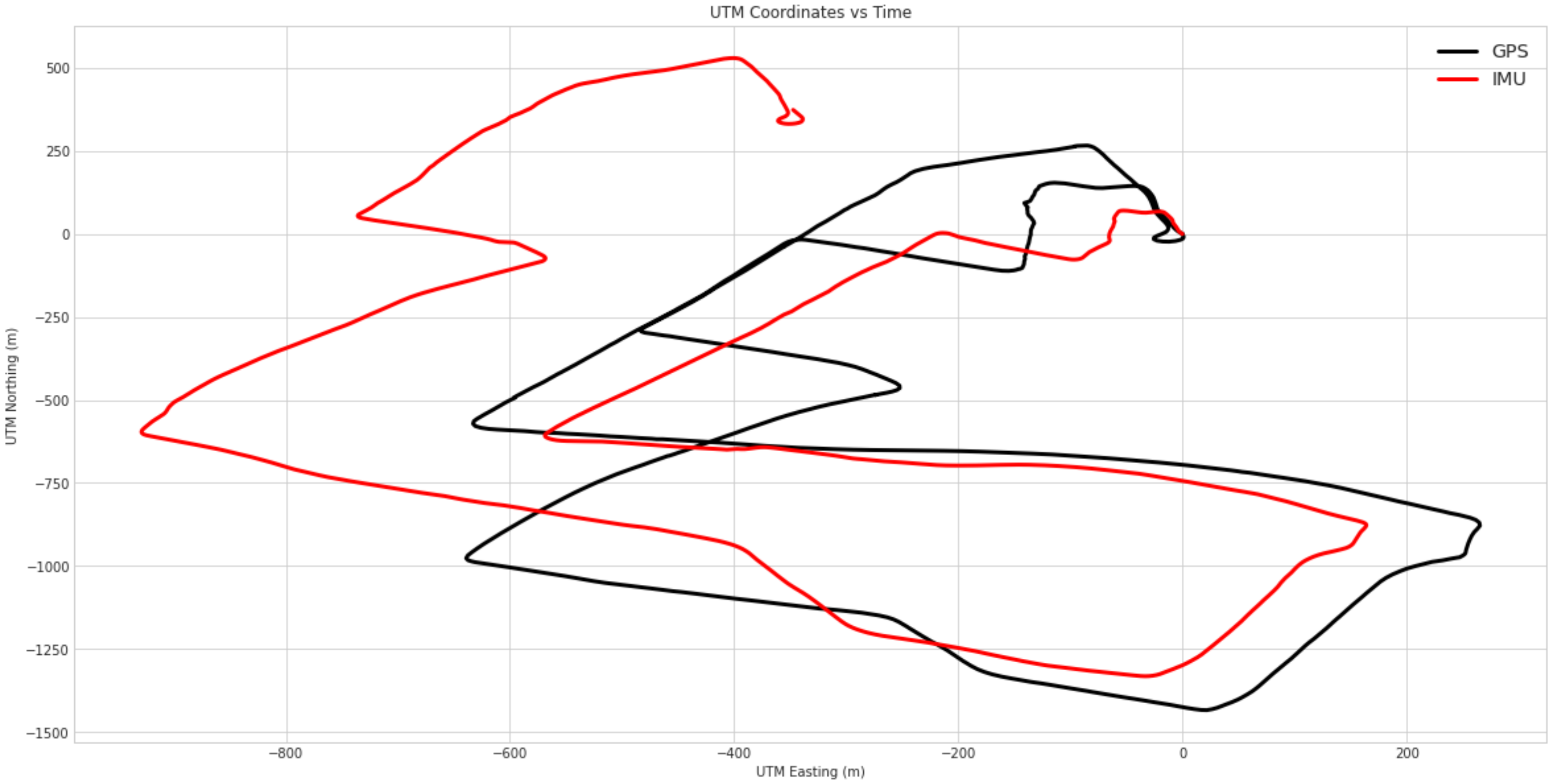
6. **Compute the ωX' and compare it with the y''obs. How well do they agree? If there is a difference, what is it due to ?**

   The following is the comparison plot of the data –



The green data given by Omega * X’ is the centrifugal force, the spike in the data gives us the times the vehicle took turns during the drive. The centrifugal force shows spikes as the rate of rotation increases around these turns.

7. **Estimate the trajectory of the vehicle (x_e,x_n) from the IMU data and compare with the GPS by plotting them together. Report any scaling factor.**



- The figure above shows the comparison of the (x_e,y_n) calculated from the IMU and the GPS. The IMU data had to be scaled by a factor of 0.4 to match the size of the GPS plot.
- Both the plots are centred around the origin.
- The distance calculated by the IMU had to be rotated by an angle of 20 degrees to match the representation of the GPS data.

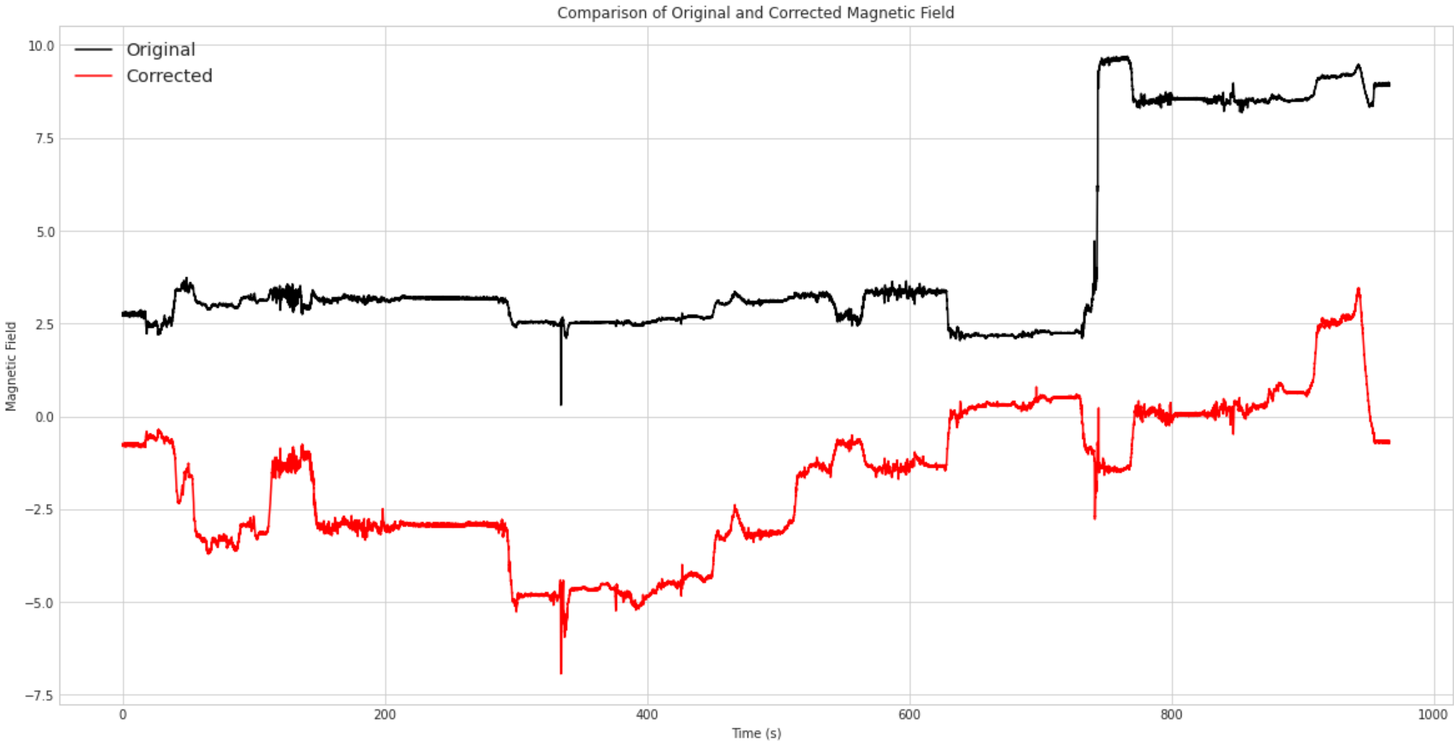
The following code snippets represent the transformations undertaken –

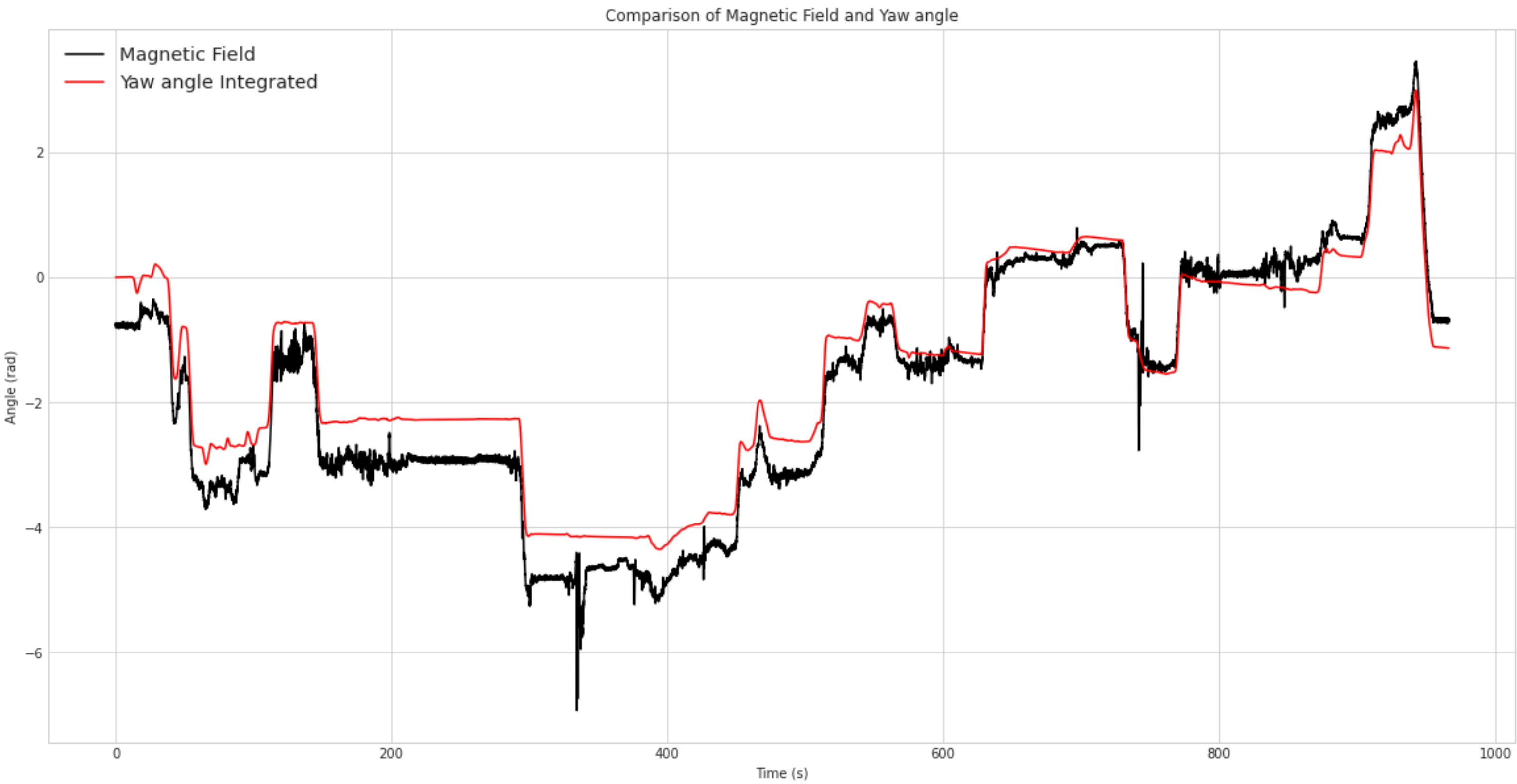
```
angle = 20
# Get the distance in x and y directions
dist_x = integrate.cumtrapz(velocity_x, time, initial=0) * 0.4
dist_y = integrate.cumtrapz(velocity_y, time, initial=0) * 0.4
# Get the distance in x and y directions at a certain angle
dist_x_angle = dist_x*np.cos(angle) - dist_y*np.sin(angle)
dist_y_angle = dist_x*np.sin(angle) + dist_y*np.cos(angle)
```

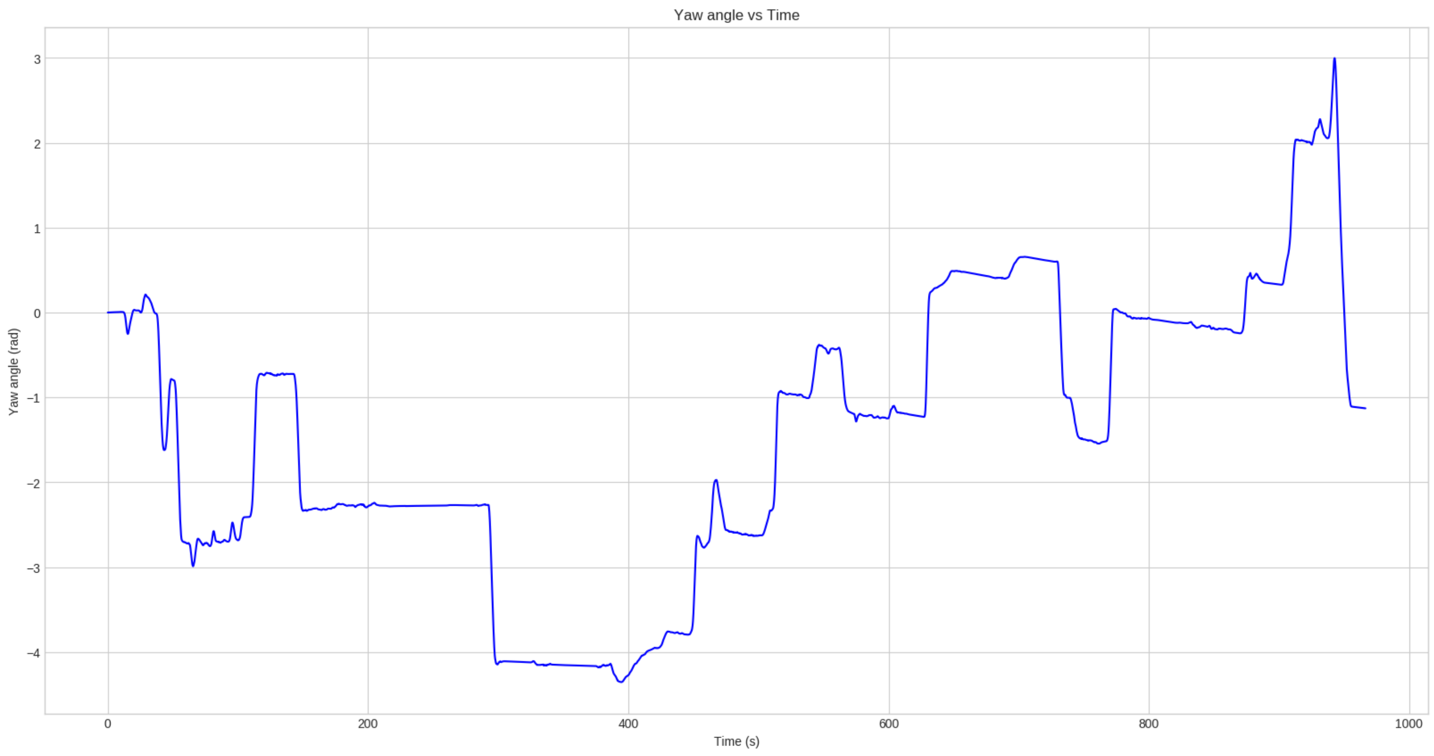
8. **Given the specifications of the VectorNav, how long would you expect that it is able to navigate without a position fix? For what period did your GPS and IMU estimates of position match closely? (within 2m) Did the stated performance for dead reckoning match actual measurements? Why or why not?**

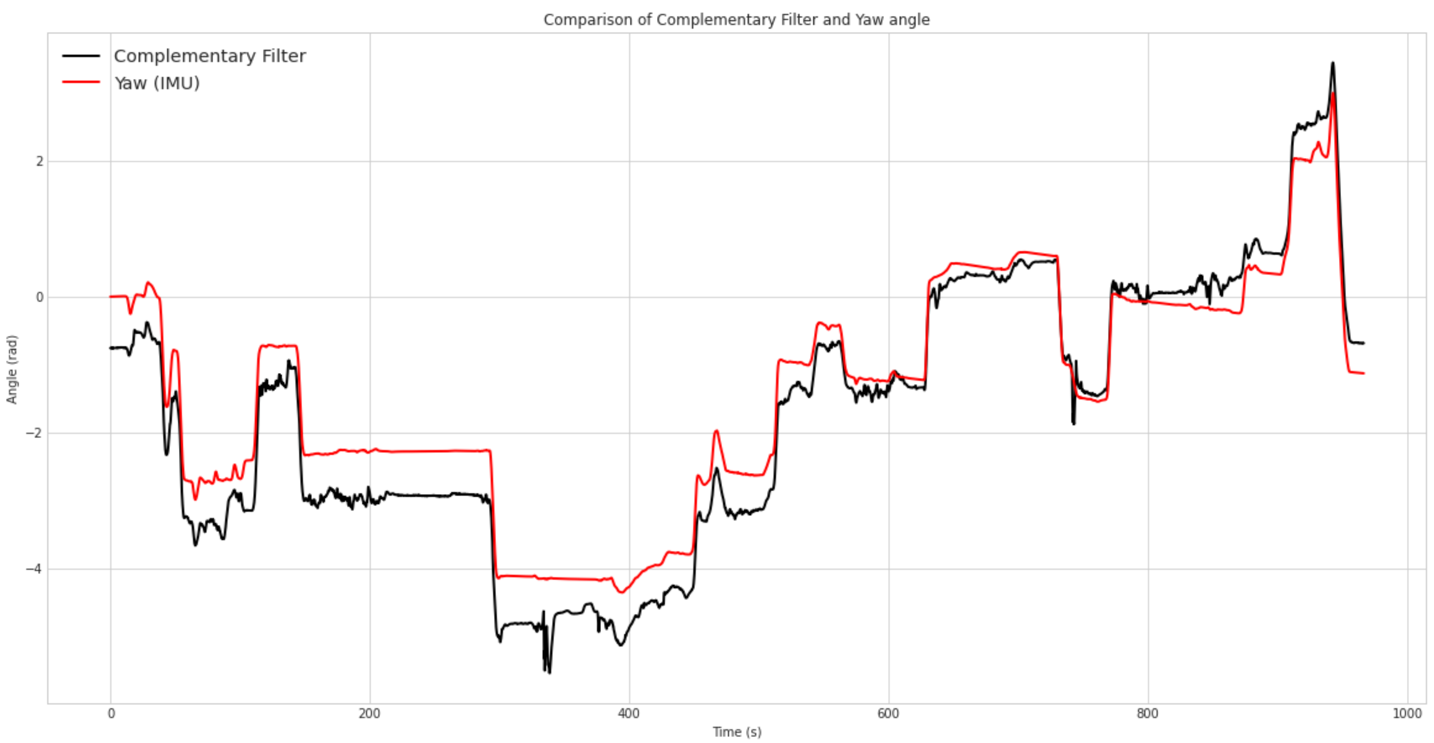
Dead reckoning with the IMU, and Magnetometer data is not a perfect method to be used for the navigation of a robot. The bias instability of the sensor adds in drift which can cumulate over time that can give huge errors. The IMU data gets additional gravitational components added to it in case the vehicle climbs a hill. Added with the fact that IMU has a random walk which makes the sensor unreliable over longer terms as we would also have to correct for that. The IMU is a accurate for a period of 150secs, however with added components due to the terrain we start seeing more offsets in the data.

Additional Plots:






Yaw angle vs Time
Yaw angle (rad)
Time (s)
3
2
1
0
−1
−2
−3
−4
0
200
400
600
800
1000


Comparison of Complementary Filter and Yaw angle
Complementary Filter
Yaw (IMU)
Angle (rad)
Time (s)
2
0
−2
−4
0
200
400
600
800
1000