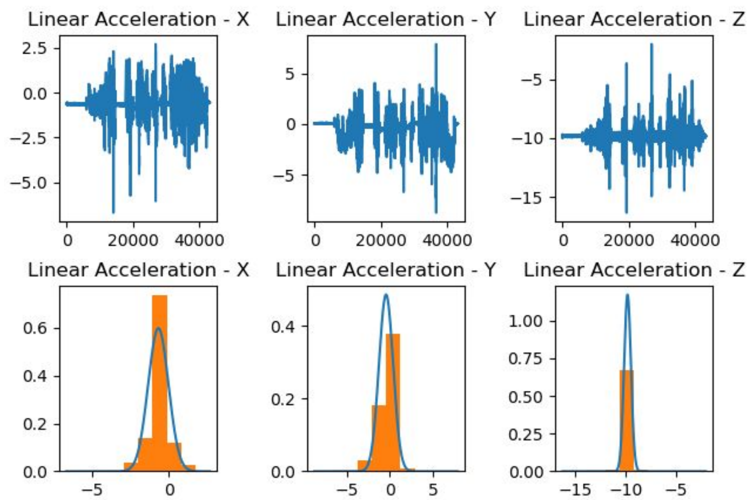
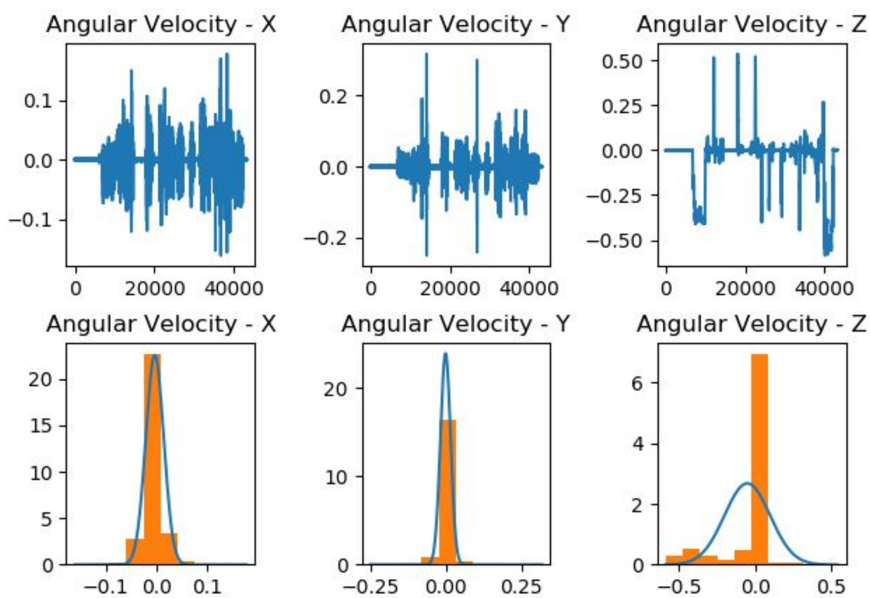
Lab 2 Report - Kyle Coelho

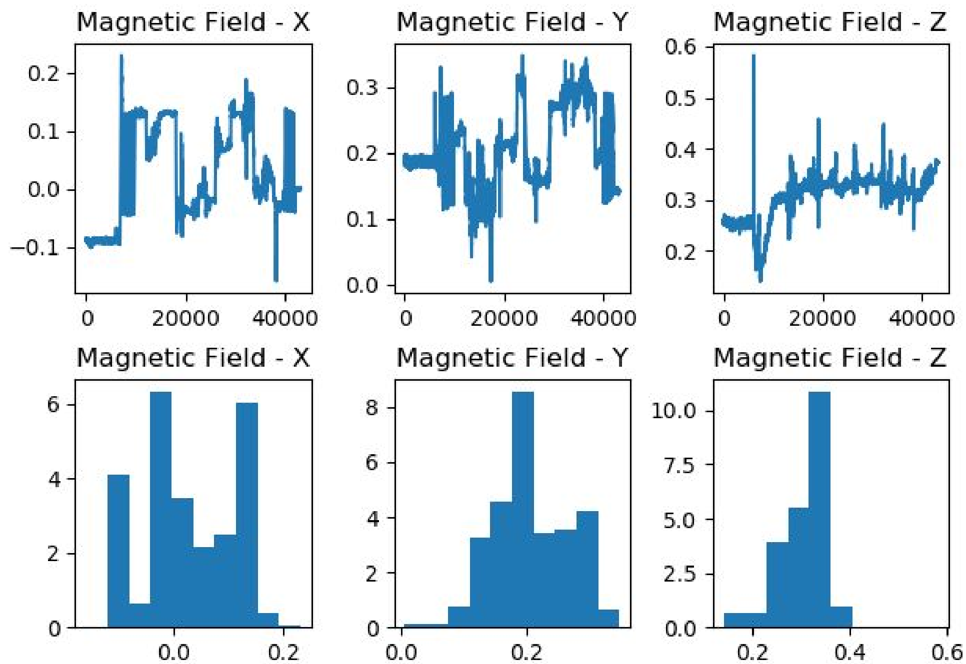
Sitting Data



The sitting data was collected with the IMU taped to the floor in the tunnels of the University. The noise characteristics of the linear acceleration are shown above and they all appear to be Gaussian centered at different mean values and with different standard deviations.

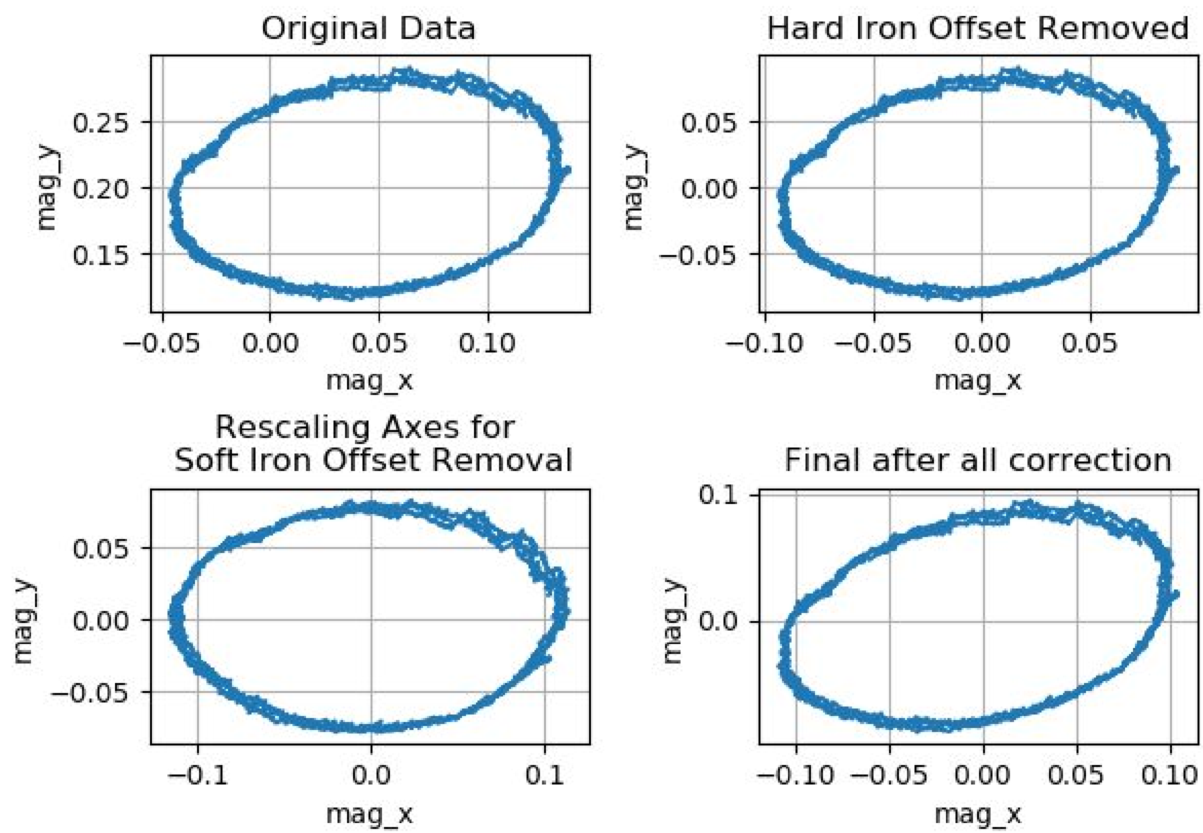


The angular velocities all show very low amplitudes as they should not be registering any rotations. While the x and y components seem very clearly Gaussian, it is more difficult to tell for the z direction. The angular velocity in the z-direction shows some irregular dips possibly due to movement from further away. However, this does not show up in the x and y measurements and could be due to me accidentally pulling the cable slightly without noticing. If those dips were not present, however, it could be seen as Gaussian.



The magnetic field raw data is more difficult to determine the noise profile since there are numerous offsets that could be throwing it off coming from our phones, laptops or any number of other interference sources. They all have different biases and seem to have fluctuations triggered by some random events, although small in magnitude. While the y component looks Gaussian to an extent, the x component seems more of a uniform distribution. The z component shows little variation overall and is difficult to determine.

Estimating Heading



The x and y components of the magnetometer data were chosen only for when we were driving in circles before leaving. In the data, there are hard and soft iron offsets visible since it is off-center and looking like an ellipse. The hard iron distortions were removed by removing the offsets in the origin of the ellipse by averaging the maximum and minimum x,y values and then subtracting those values from the data.

$$(X\_max + X\_min) / 2$$

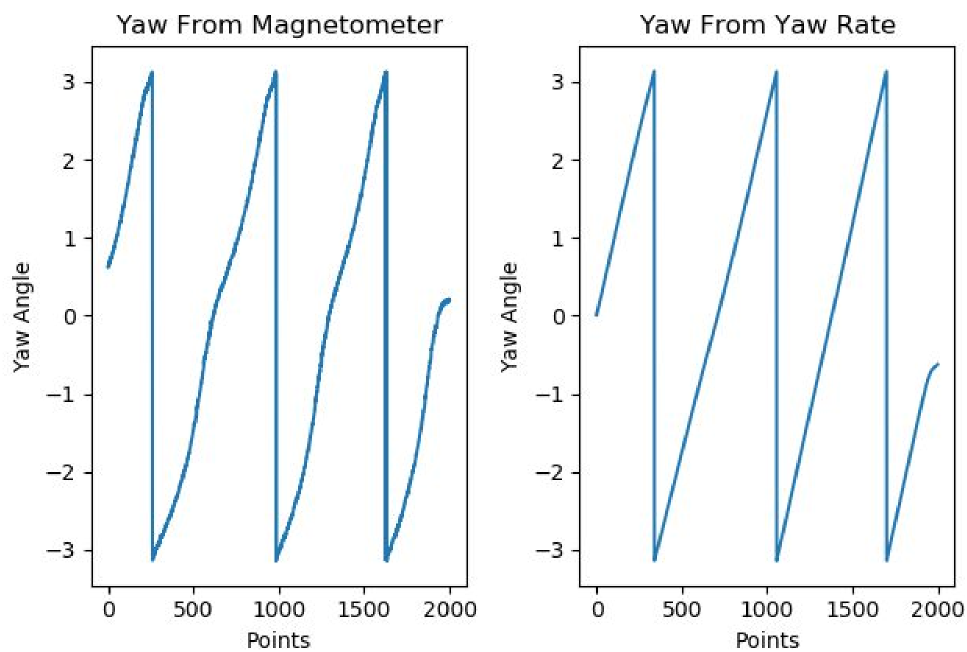
$$(Y\_max + Y\_min) / 2$$

The soft iron distortion was removed by finding the major and minor axes and then applying rotations and scaling based on the angle calculated between the offset axes and the true axes. The co-ordinates were calculated using the circle subset, recorded and hard-coded to apply to the entire dataset to prevent the min and max functions from choosing outlier points. The result is not a perfect ellipse as I did not implement an algorithm to iterate through major and minor axes coordinates to check if they formed a 90 degree angle. Above is a good estimate and is a factor in future heading calculations.

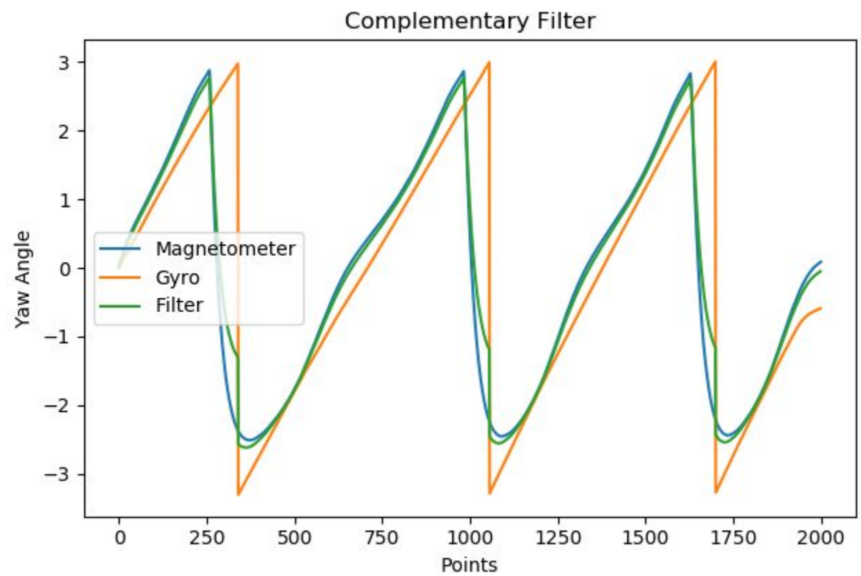
For this subset of data related to the magnetometer and imu, two yaw angle plots were generated. From the magnetometer, the yaw angle was calculated using:

$$arctan2(magnetometer\_y\_values,\ magnetometer\_x\_values)$$

For the gyroscope data, the yaw angle was calculated by integrating the angular velocity in the z direction according to our corresponding timestep. The two data sets are as shown below wrapped to between -pi and pi radians.

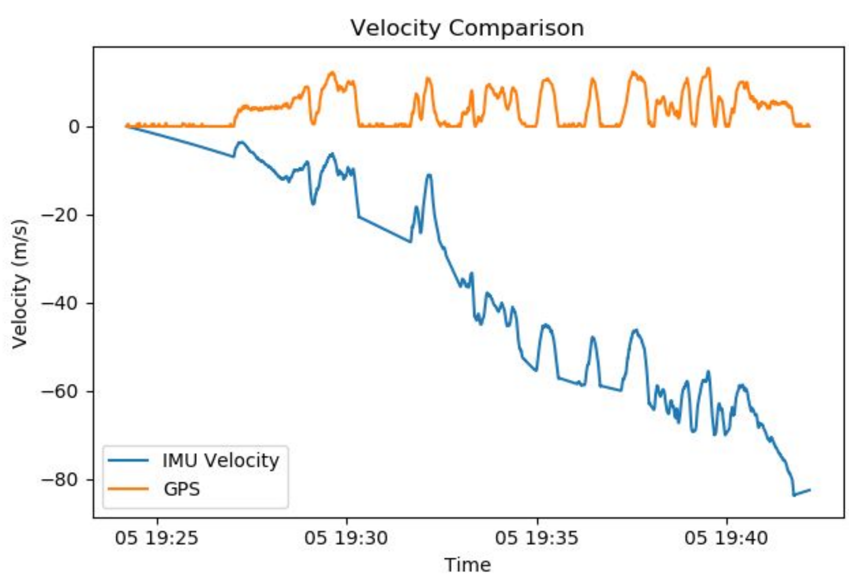


The yaw rate from the gyroscope matches quite closely to the more accurate magnetometer readings. However, the magnetometer data is more noisy with several ripples. Hence a complementary filter is implemented to obtain a better estimate of the heading since I want to keep the sharp high frequency changes from the gyroscope but also the low frequency content from the magnetometer. The yaw angle from the gyroscope was high pass filtered with a cut-off frequency of 0.8 Hz. The yaw angle from the magnetometer values was low pass filtered at 0.4 Hz and then the results from both were added together with a 0.2:0.8 weighting with more weightage given to the gyroscope yaw since it is less noisy. This cannot be compared to the IMU yaw values since the driver we used at the time of data collection had a function that accepted radian values for yaw, pitch, roll to convert to quaternions. However, we supplied it with values in degrees. Converting back is not easily possible.

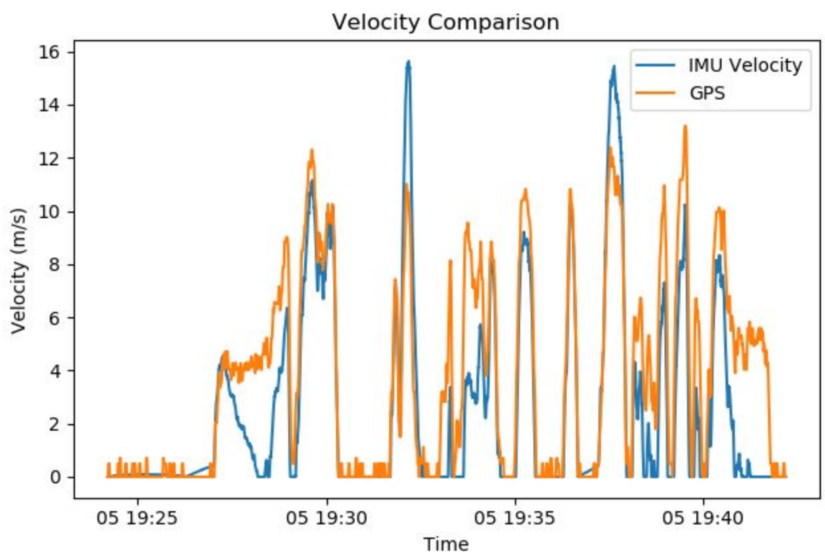


Estimating the Forward Velocity

The acceleration in the z direction was integrated using cumtrapz to obtain the velocity. The GPS data used as a comparison was calculated by taking the consecutive difference in the easting and northing values and then taking the magnitude of both the sets of data points. The resulting comparison did not match up very well as shown below:



While the areas of acceleration and deceleration match up with the GPS estimate, there was a lot of bias and noise in the acceleration data that caused drift in the integrated velocity values. Therefore I had to first offset the acceleration data by offsetting all the data points by the mean of the first 50 points. There were still biases present after doing that where some data points were not at 0 even though there should be no acceleration. These points were then set to 0. Finally, a moving average was calculated every 5000 points to try and smoothen out the data some more. Another method that was tried was finding the variance of the data in these chunks of points (moving variance calculation) and if it was below a determined threshold, those data points would be replaced by the mean of those points. However, it did not work well because even after adjusting trying to optimize the chunk sizes and thresholds a lot, it created areas of constant velocity where there shouldn't be. Hence, the final velocities after all these corrections is shown below:

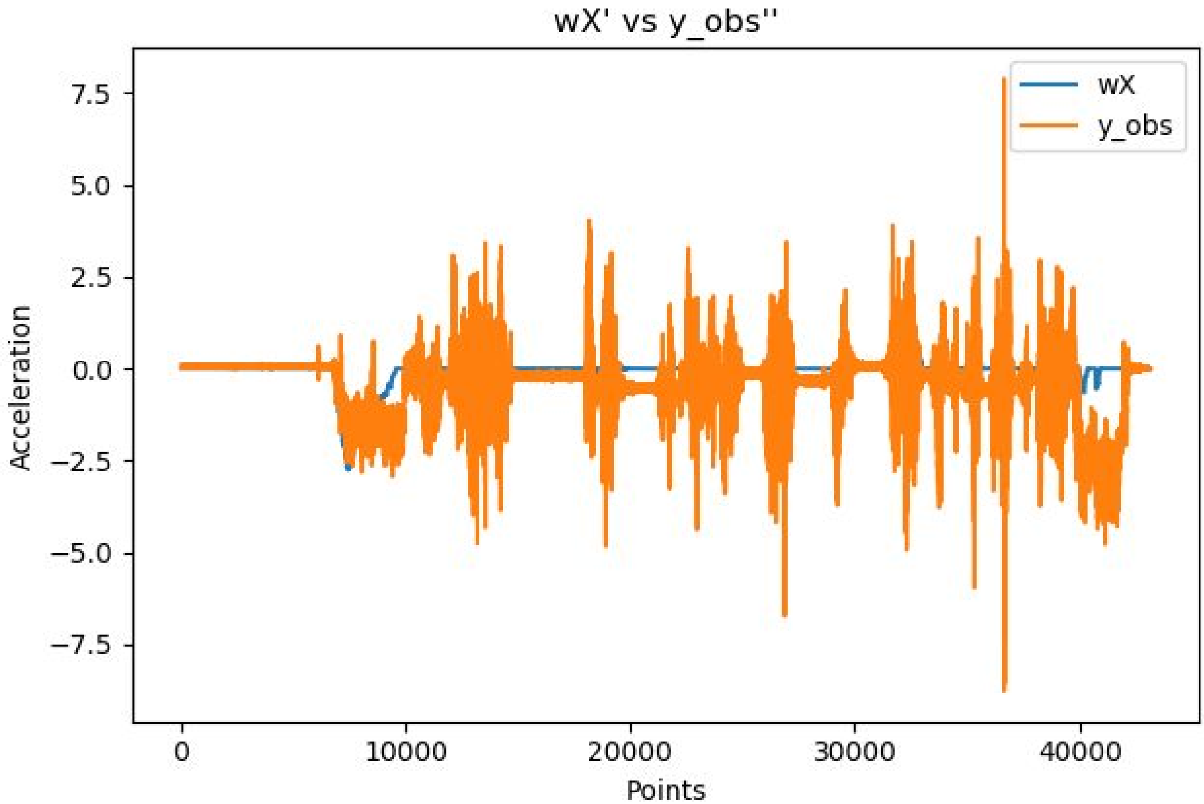


These seem to be in line with what we expect since no velocity values exceed 35 mph while there was significant times at ~25 mph, which is what we noted while driving. Almost all of the data matched up correctly but there are still some areas that show decreases when it should be constant or increasing instead. These are the areas where we were going around in circles, where the negative values caused it to show up as a deceleration. Attempts were made to rectify this such as trying to apply a separate moving average filter to it with stricter or more loose parameters but they either ended up negatively impacting the rest of the integration or not working at all. So this velocity was used going forward.

Dead Reckoning

wX' and y_obs'' comparison

Assuming that the vehicle has no velocity component in the y direction and that the IMU offset from the car frame is 0, *wX'* is compared to *y_obs''.* The figure is shown below:



The two datasets compared above seem to match up fairly well. However, there is significantly more noise in the y component of the acceleration along with a slight offset. This is probably due to the ambient vibration of the car. So our assumption when calculating the xW' that the velocity in the y direction is 0 is not entirely correct.

<u>Trajectory Estimation</u>

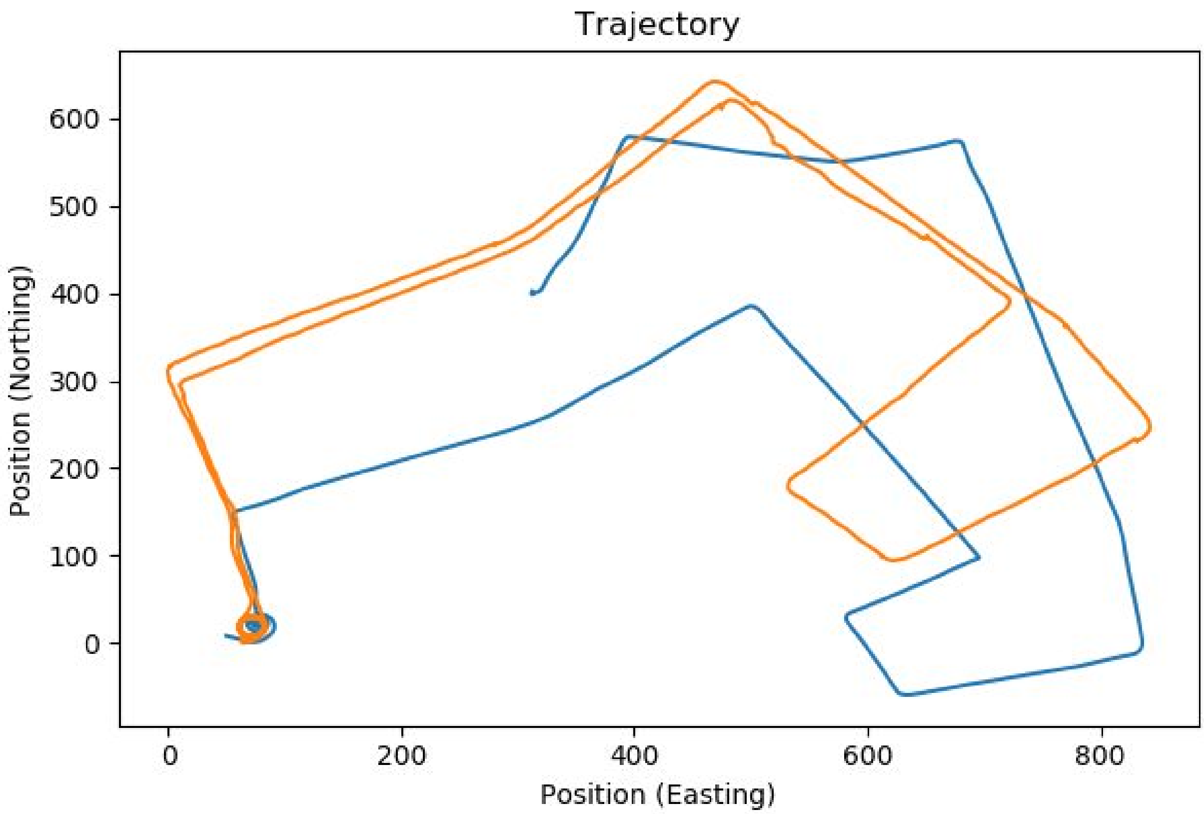
The velocity was resolved into the Easting and Northing reference frame using the heading values from the complementary filter that were previously calculated.

$$vx[ii] = v\_x[ii] * np.cos(heading[ii])$$
$$vy[ii] = v\_x[ii] * np.sin(heading[ii])$$

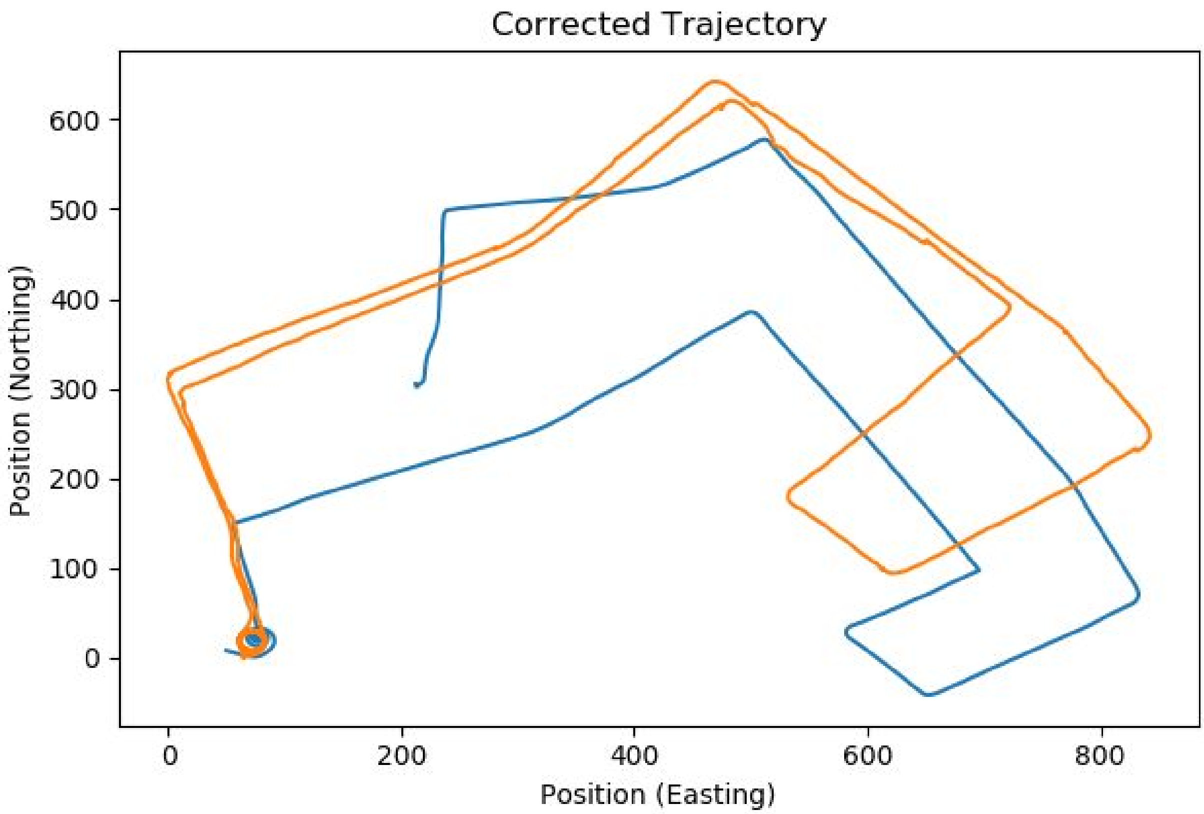
These values were then integrated to obtain position values that relate to the trajectory. It was also cross-checked with position calculated using the following formula to verify:

$$px[ii] = origin\_x + (vx[ii] * timestep)$$
$$py[ii] = origin\_y + (vy[ii] * timestep)$$

Both resulted in near identical plots so the integrated plot was used moving forward. Using these position values, the following trajectory was obtained:



The starting point for both the GPS and calculated data were made to align and the calculated trajectory was rotated by 50 degrees counter-clockwise. The calculated data is not scaled in any way. Individual components should be scaled by separate amounts such as the path along Forsyth Street and the return path along Huntington Avenue. However, these scale factors would each be different and not applicable to the entire dataset. The main error in the trajectory comes when taking a left turn back onto Massachusetts Avenue. Our stop at the signal there was quite long (~ 1 minute) and so the heading value was affected. The magnetometer must have picked up random variations while the gyroscope must also have accumulated error, which was then integrated. Furthermore, this is because I used my filtered yaw angle, which could be less accurate than the yaw from the IMU. This carried through to the rest of the data points due to the cumulative addition and affected the subsequent turns and paths. Our velocity points at these times also report slightly higher values than expected resulting in going further than intended. If this is true, then reducing the heading at that turn should correct the data partially.



In the corrected trajectory, while that turn seems to be normal, there are other differences that do not carry through since my corrected velocities are also not what they exactly should be.

<u>Estimating x_c</u>

Using the equation provided in the lab instructions:

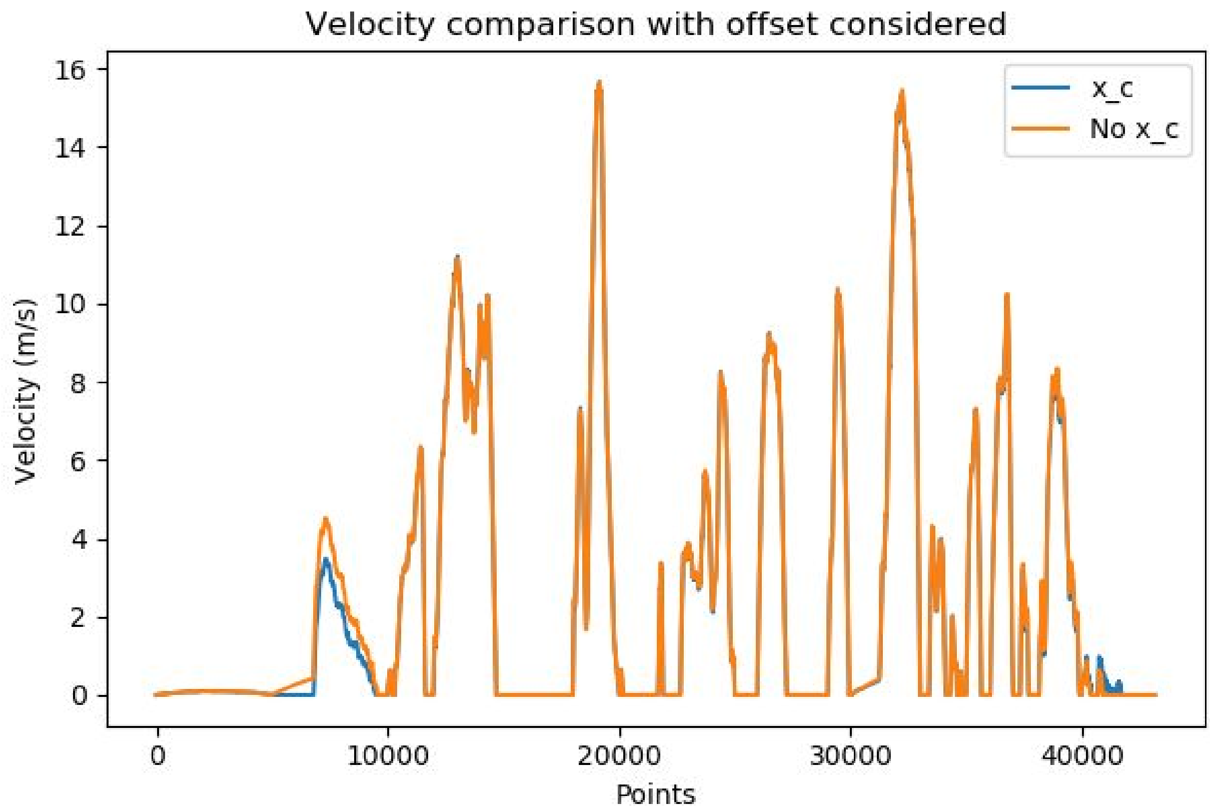
$$\ddot{y}_{obs} = Y + \omega X + \dot{\omega} x_c$$

x_c was calculated. The position of the car frame in the y-direction is assumed to be 0 to allow for the calculation. This resulted in a value of 26.57 cm. This is then factored into the acceleration to see whether it contributes to any significant error or variation.

When obtaining velocity in the car body frame, we simply integrated the imu frame acceleration ignoring this x_c based off the lab instructions formula:

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

Now we are considering it and the result is shown below compared between the resulting velocities:



Hence, my resulting offset seems so small that it does not seem to affect the velocity much. As a result, it would not impact the resulting trajectory in a major way