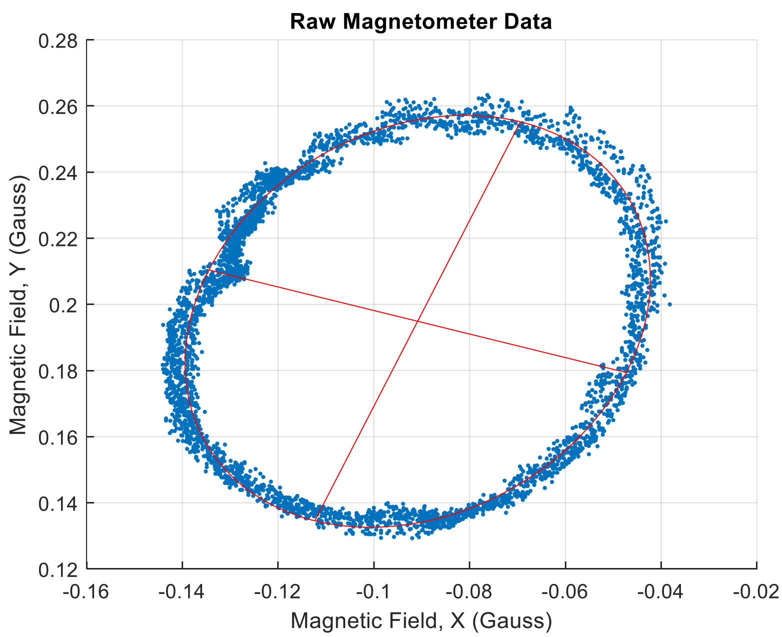
Sensor fusion is the combining of two or more data sources in a way that generates a better understanding of the system. In this usage, better refers to a more accurate, more consistent, and more dependable response. In this lab, a navigation stack was assembled by fusing two sensors – a GPS and an IMU.

For simplicity, the analysis was split up into three MATLAB scripts – one to perform magnetometer corrections and estimate the heading, one to estimate forward velocity and perform dead reckoning, and one to solve for the sensor offset in the vehicle.

1. Estimate the Heading (yaw)

The IMU's internal magnetometer can be used to estimate heading by measuring the angle from magnetic North. Unfortunately, this measurement is subject to noise from sources such as the electromagnetic fields generated by nearby electronics. If these disturbances are part of the overall system and rotate along with the magnetometer, they can be calibrated out – these are called hard iron and soft iron sources. A hard iron source is one that generates its own magnetic field, which would result in a measurement offset. A soft iron source is ferromagnetic and bends the magnetic fields as they pass through. The amount of bending changes as it rotates, distorting the measurements.

To find and correct for these disturbances, the magnetometer was driven around in a circle for four loops. If there were no distortions, we would expect to see the collected data lying on a perfect circle centered on the origin with the radius equal to the magnitude of the field. It can be seen in the data below that there are both hard iron and soft iron sources, since the circle is transformed into an ellipse (Soft-Iron) and is translated away from the origin (Hard-Iron). The parameters of the ellipse are tabulated and shown on the right.



Ellipse tilt angle	0.3419 rad
Long-axis size	0.1281 Gauss
Short-axis size	0.0925 Gauss
Center Position	(-0.0909, 0.1949) Gauss

To correct for this, we can calculate a corrected reading using the equation

$$m_c = S_I(m^{\sim} - b_{HI})$$

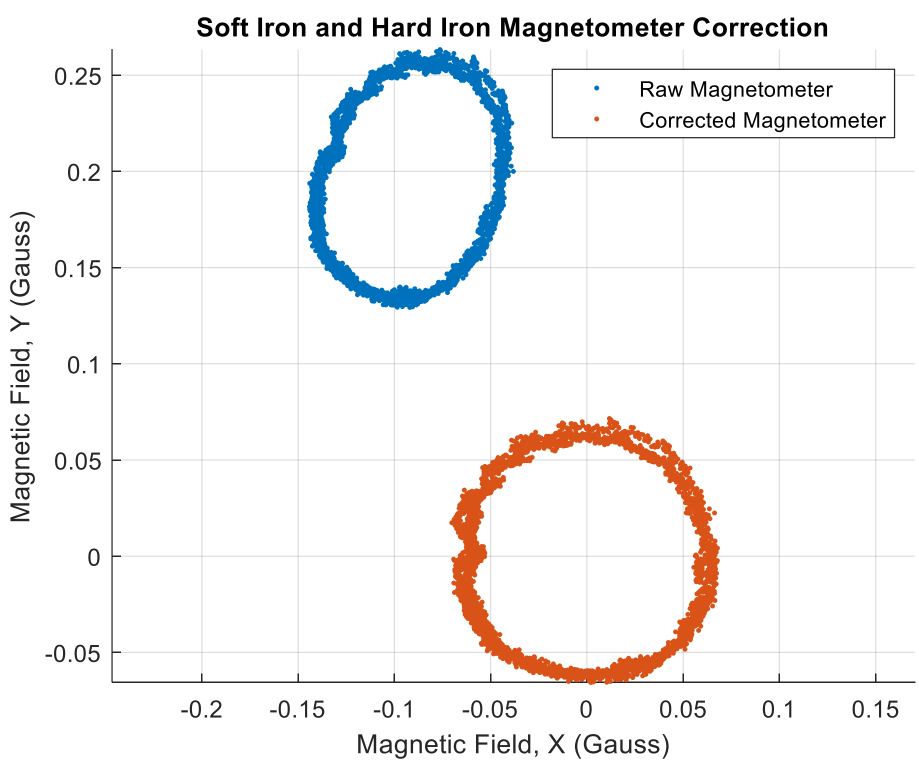
where m_c is the corrected magnetometer measurements, S_I is the soft iron distortion matrix, and b_{HI} is the hard iron bias vector. The soft iron transformation matrix was determined in two steps. First, the ellipse was rotated by the tilt angle, and then it was scaled by the ratio of long-axis to short-axis sizes:

$$C_r = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad , \quad C_s = \begin{bmatrix} \frac{\text{long}}{\text{short}} & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad C_s * C_r = S_I = \begin{bmatrix} 1.304 & -0.464 \\ 00.335 & 0.942 \end{bmatrix}$$

Therefore,

$$\begin{bmatrix} m_{c_x} \\ m_{c_y} \end{bmatrix} = \begin{bmatrix} 1.304 & -0.464 \\ 00.335 & 0.942 \end{bmatrix} \begin{bmatrix} m_x^{\sim} + 0.0909 \\ m_y^{\sim} - 0.1949 \end{bmatrix}$$

After making these corrections, the ellipse can be seen to have transformed into a circle centered at the origin.

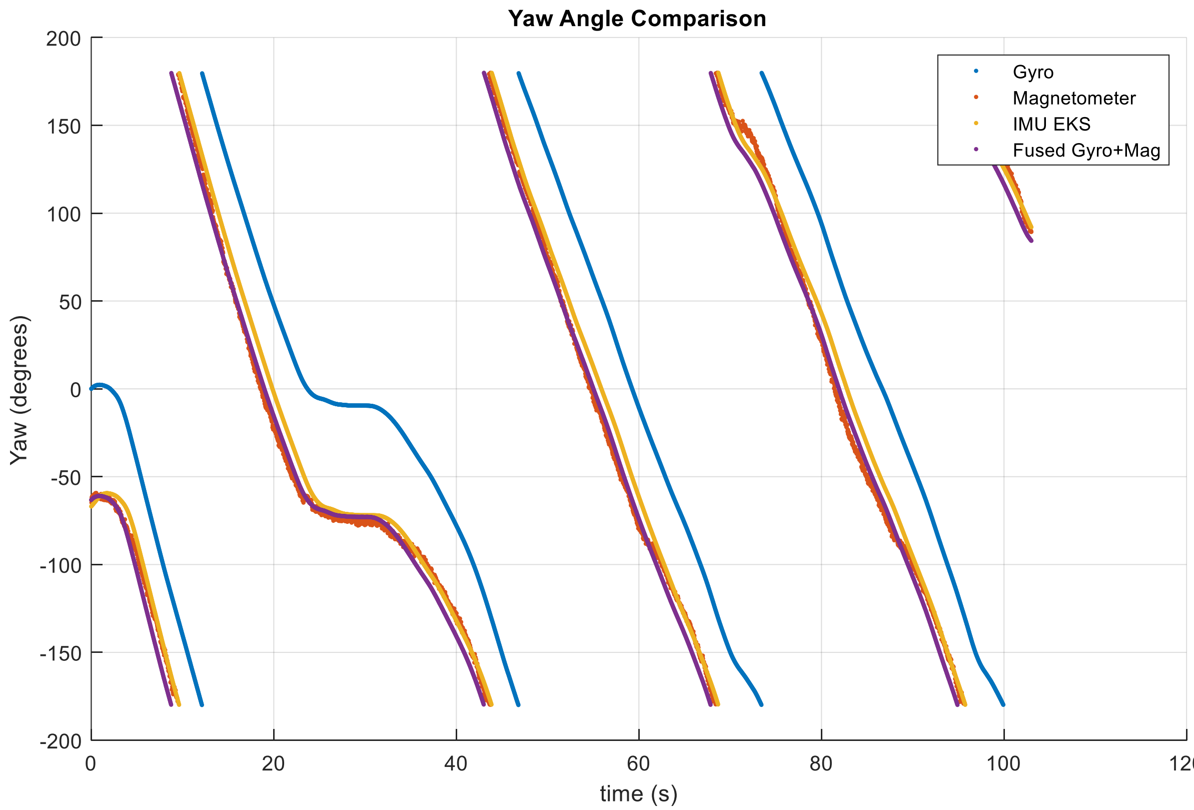


From these corrected magnetometer readings, the yaw angle over time can be calculated by taking the inverse tangent of the X-magnetometer readings divided by the Y-magnetometer readings.

The magnetometer noise from the system has been dealt with, but external magnetic sources still impact the measurements. This can be mitigated using a lowpass filter, but this adds lag and makes the measurements less responsive. Another option is to fuse the magnetometer with an angular rate sensor (gyro). The gyro will be noisy, as well, but by using two different sensor types we are reducing the likelihood that the noise is correlated, so they can be used to calibrate each other. If the magnetometer measures a change in the magnetic field, the gyro can be used to confirm if that rotation came from the sensor physically moving or from noise.

Integrating the measured yaw angular rate from the gyro, we receive another estimate for the yaw of the system. The problem with this method is that we do not know the initial orientation, so it will start at zero. Also, gyroscopes have bias and other high frequency noises that will corrupt our estimation. Integration acts like a low pass filter, so that high frequency noise is smoothed out, but over longer periods of time, it is expected that the yaw estimates from the gyro will drift away from the true position due to random walk and integrating any bias in the measurements.

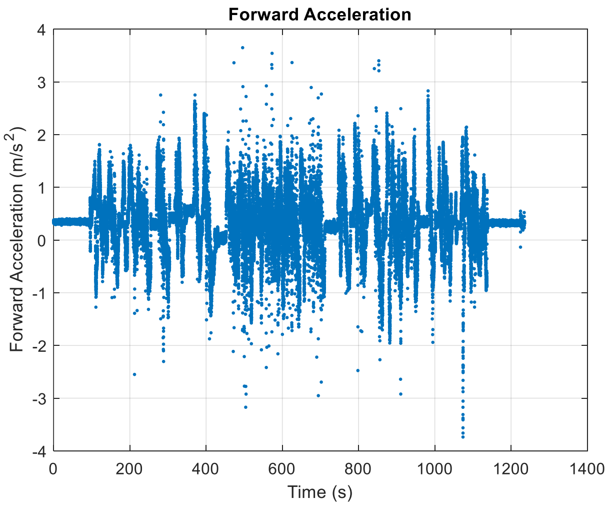
The advantage of the gyro is for short-term precision and error correction while the advantage of the magnetometer is long-term accuracy. To utilize both of these sensors, they can be fused using several different filters. The VN-100 IMU uses a Kalman Filter. In this case, a complementary filter will be used by combining the lowpass-filtered magnetometer's yaw estimate with the highpass-filtered gyro's yaw estimate. This is achieved by multiplying the magnetometer by a value, α, and the gyro heading by $1 - \alpha$. The gyro heading is less sensitive to noise than the magnetometer heading, so a sufficiently small alpha value of 0.02 value was chosen. The problem with the gyro is that it has no sense of absolute angle, so the magnetometer is used to provide an initial bias estimate. For this reason, once the gyro is corrected initially by the magnetometer, the majority of the complementary filter is placed on the gyro data.

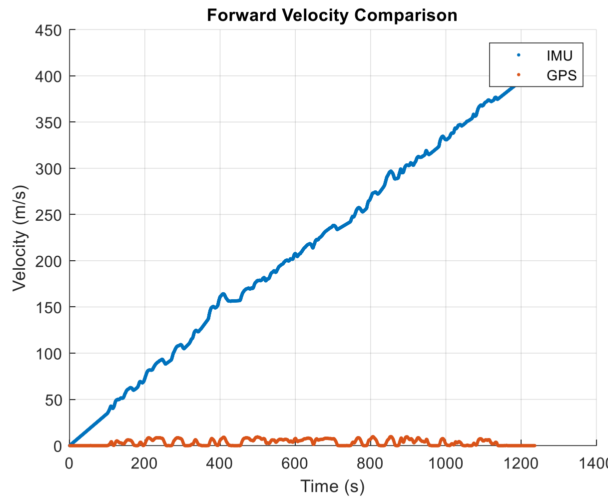


As mentioned previously, the orientation computed by the IMU hardware uses a Kalman filter, which combines the accelerometer, gyro, and magnetometer data to estimate the heading. The addition of the accelerometer adds some additional accuracy by keeping track of the direction of gravity relative to the moving system. It can be seen above that the fused gyro and magnetometer yaw estimate is very close to that estimated by the IMU's Kalman filter. Any errors are likely associated with the simplifying assumption of a 2-dimensional world.

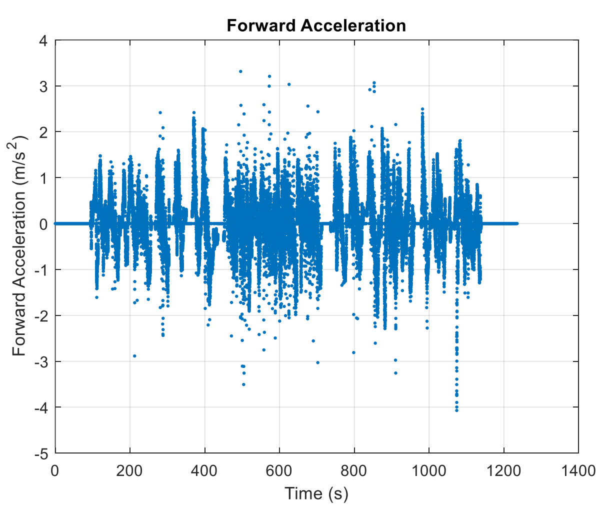
2. Estimate the Forward Velocity

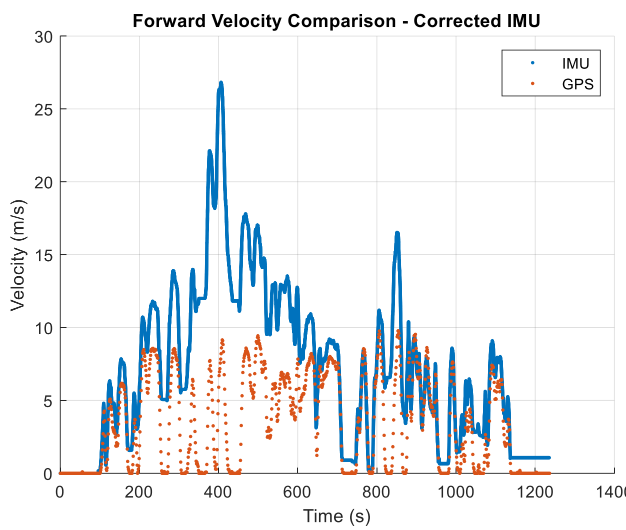
The IMU was placed in the vehicle with X facing forwards. Integrating the accelerometer data in the X-direction, therefore, yields an estimate for the vehicle's forward velocity. It can be seen in the forward acceleration plot that a positive bias persists throughout much of the data. As this is integrated, this small bias gets amplified over time, resulting in a velocity plot like the one below.





This can be mitigated in several ways. The first step is to remove the initial bias by subtracting out the average value of the first few minutes, in which the vehicle was stationary. Throughout the plot, however, there are other regions in which the vehicle was stopped, but there is an offset from zero. To mitigate the accumulation of error from integration, whenever the car is clearly not moving, which can be visualized by a lack of noise, those datapoints can be directly set to zero. These locations were found by taking the derivative of the acceleration and finding where the slope was near zero.



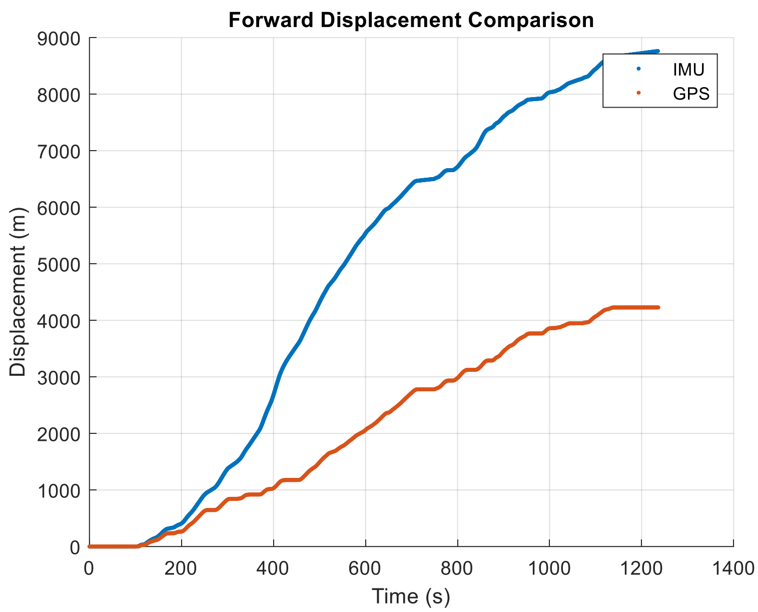


From these new velocity estimates, it can be observed that the peaks and troughs align temporally but the scaling varies over time. This can be mostly contributed to the accumulation of error from integrating a noisy acceleration signal. Additionally, the route traversed was not flat and may have contributed some gravitational components in pitch angle to the forward acceleration.

After numerous attempts at scale correction, including pitch variation, integrating at intermediate positions, and several other methods, no improved result was discovered for the IMU velocity estimate. Work along these lines can be found in the “pitch_compensation.m” Matlab script included within the Analysis folder for this project.

3. Dead Reckoning with IMU

Integrating these forward velocities yields in an estimation for forward displacement.



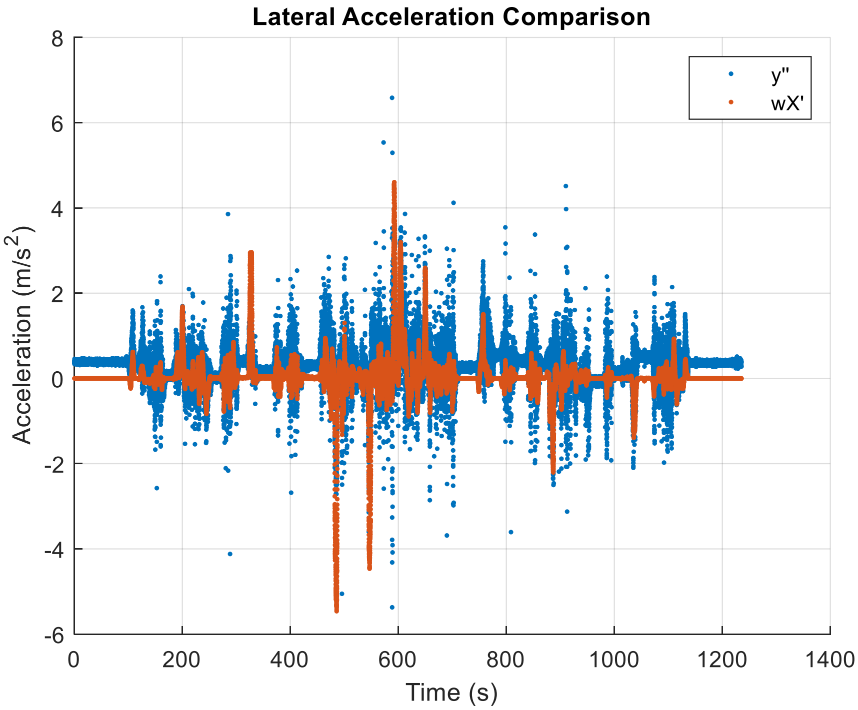
Here, the IMU position estimate quickly begins separating away from GPS estimate. This is because integration amplified any bias in the original signal, and in this case the acceleration signal was integrated twice.

The following equations of motion are simplified by assuming that the vehicle is moving in a 2D plane. The center of mass of the vehicle is at position (X,Y,0) with a rotation rate about the CM of (0,0,ω). The position of the inertial sensor is defined as (x,y,0) with its position relative to the vehicle’s center of mass is (xc,0,0).

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

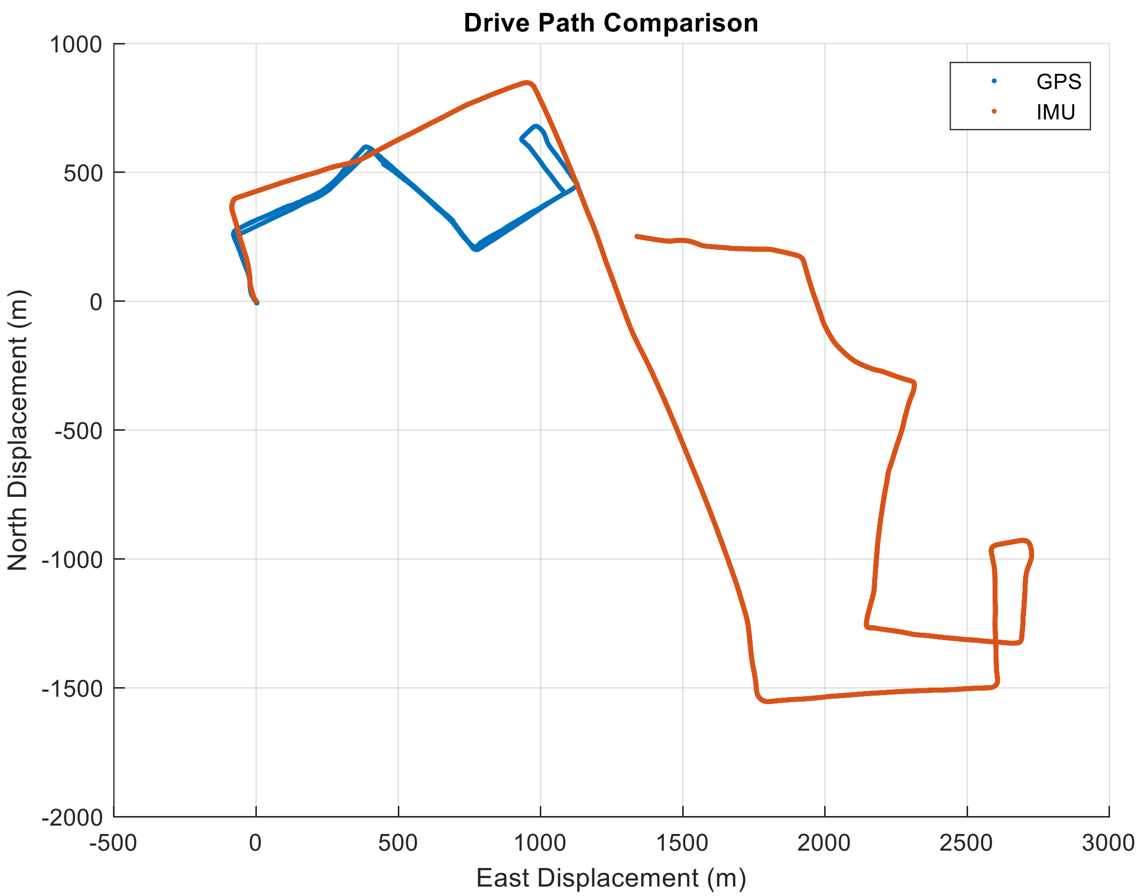
$$\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega} x_c$$

With the assumption of no skidding and negligible offset, the observed forward acceleration of the IMU is equivalent to the vehicle’s forward acceleration. The previously calculated forward velocity of the IMU multiplied by the yaw-rate measured by the gyro can be compared to the observed lateral acceleration measured by the IMU.



The spikes are time-aligned in these plots, with one of the primary differences being the varying bias in the observed lateral acceleration measured by the gyroscope. In additions, assumptions such as a 2-D operating field and ignoring the IMU's offset from the vehicle's CM, x_c, contribute to sources of error.

Using the heading from the magnetometer, the forward velocity can be split into easting and northing components and then integrated to receive a global position to compare with that recorded by the GPS.



As expected based on the error propagation in the velocity and forward position plots above, the plots have significant scaling problems. While the shape is recognizable between the GPS and fused IMU and Magnetometer, it is easy to see that, over time, the latter system drifts significantly compared to the GPS, which is constantly updating its global position. Drift is also seen in the orientation measured by the sensor

For relatively slow movements such as in this exercise, the IMU does not contribute much to the positional accuracy compared to just using a GPS. Because the GPS updates once per second, with more dynamic and swerving movements, an IMU is useful to bridge the gap between the 1-second sampling rate of the GPS.

The relative location of the IMU can be extracted by taking the equation $\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega}x_c$ (where $\ddot{Y} = 0$, assuming no lateral sliding) and solve for x_c, which represents the offset from the center of mass of the vehicle. Being placed on the dashboard, it is expected that the sensor would be approximately +1 meter in the forward direction (ahead of the CM). Plugging in our previous findings for lateral acceleration, yaw rate, forward velocity, and angular yaw acceleration, that offset can be estimated. This value was found to be untrustworthy because when the vehicle is moving straight forward, there is no change in yaw rate, so it is impossible to calculate the offset. The values of moving straight were removed and only the values during turning were considered. This resulted in an estimated value of 4.6 meters from the center of mass. Because there were still some straight portions that were difficult to remove, another estimate of the offset was determined using the initial dataset in which the vehicle circled the roundabout for magnetometer calibration in Part 1. Using this dataset and the same procedure resulted in a value of 1.06m offset from the center of mass, which is a more reasonable estimate.