Mohammad Abdul Razaq Khan (NUID: 002851185) EECE5554 Lab 4 Report

For the Lab4 experiment data was collected using the Northeastern (NUANCE) autonomous car. The route taken followed the path described in class starting and ending at Forsyth circle. Our run was around 10 AM on a weekday and traffic conditions were moderate.

Part 1a: Magnetometer Calibration

In order to calculate yaw angle using the magnetometer, hard and soft iron effects from the IMU mounted inside the car must first be removed. Figure 1 below shows that before correction hard-iron effects caused our circle data (measured in milli Gauss) to be offset at every point (like a linear shift translation) from the origin [0,0]. This was more significant than the soft-iron effects that distort the shape of the circle which is not very present here in this dataset.

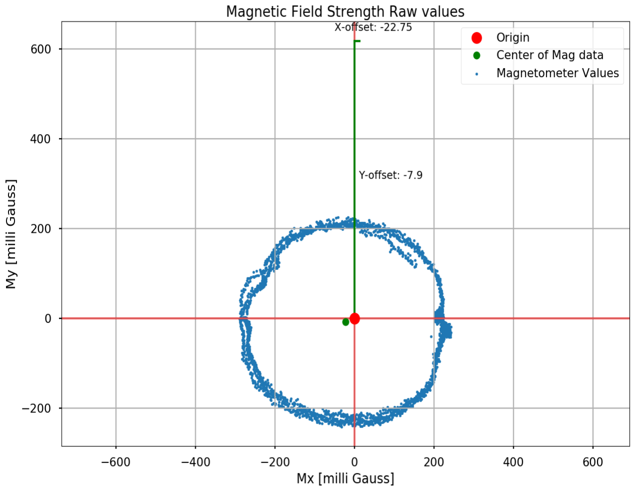


Figure_1a: Raw Magnetic rata in X and Y

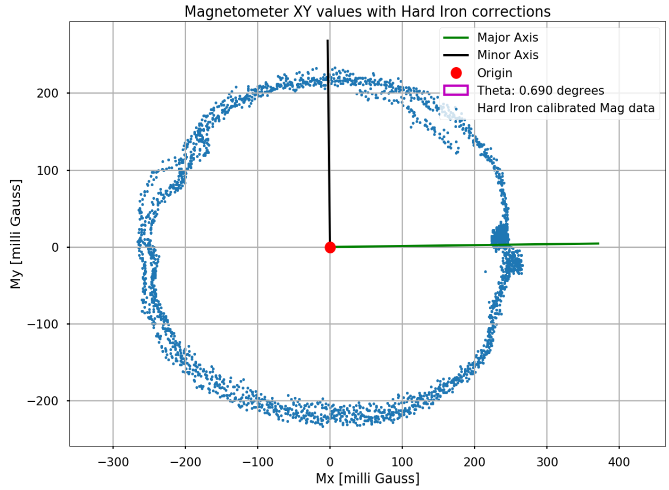


Figure_1b: Magnetometer data with hard iron corrections

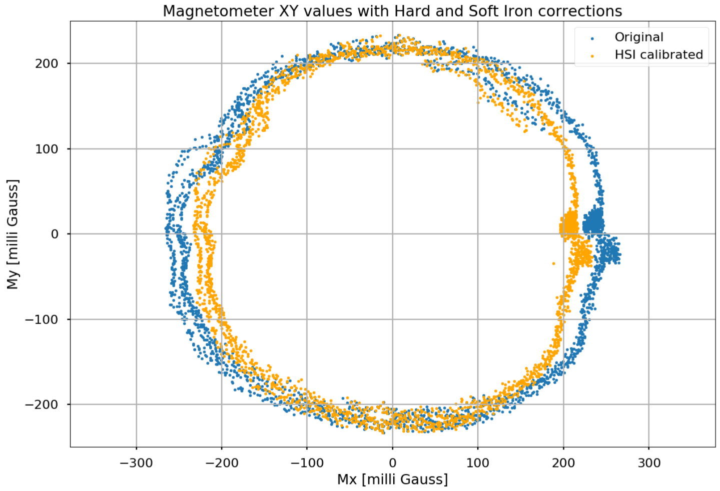


Figure_1c: Magnetometer data with hard and soft iron corrections

Hard-iron distortion occurs when certain materials consistently add their own magnetic field to the earth's magnetic field, which causes a constant increase in the measured output of each axis of a Magnetometer. This is corrected by taking the maximum and minimum of each axis and finding their average, and subtracting these values with the raw x,y data. This will shift the data to (0,0).
Similarly, Soft-iron distortion occurs when certain materials affect or change the shape of a magnetic field, without adding their own magnetic field to it. This distortion is influenced by the material's position relative to the sensor

and the magnetic field, and can be identified by observing an ellipse with a rotational angle. To reduce moderate soft-iron distortion, we need to determine the length of the main axis by visually approximating it or by fitting an ellipse. find the maximum and minimum of x and y This gives us the distance of the farthest point (x, y) from the centre (0,0). The distance r may be calculated using the expressions

R = sqrt((x2-x1)^2+(y2-y1)^2)   and   Theta = arcsin(y1/r).

Part 1b: Estimating Yaw Angle

The soft and hard iron corrections are done for the entire dataset. The yaw from magnetometer values is found by the arctan of the corrected magnitude values. Meanwhile, the yaw values are found by integrating the gyro z-axis values. Both of them are compared as shown

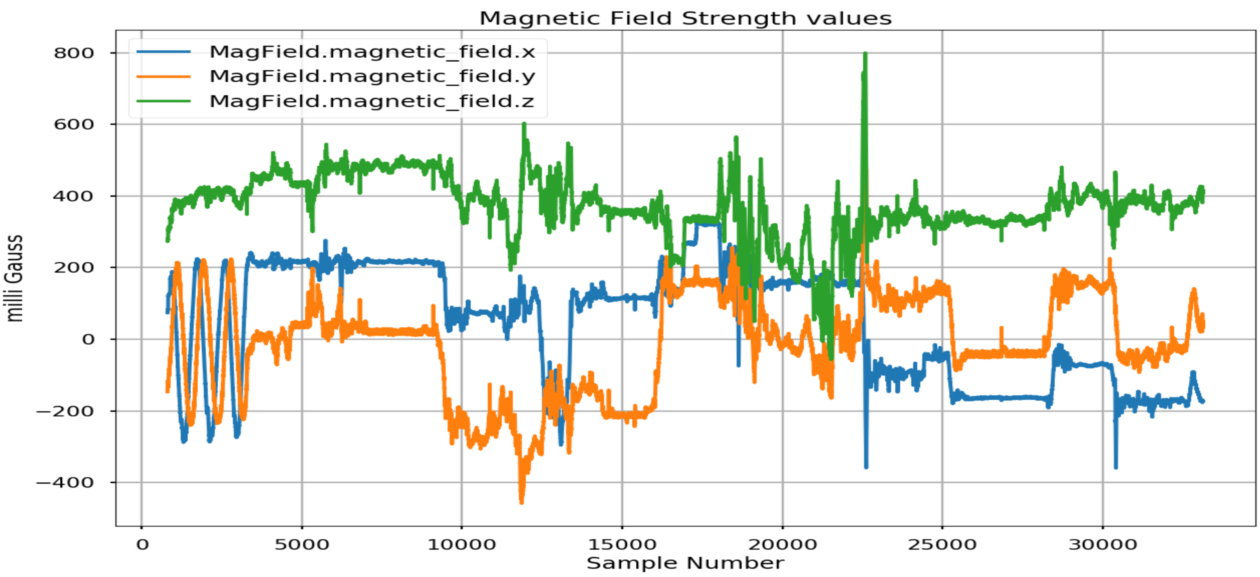
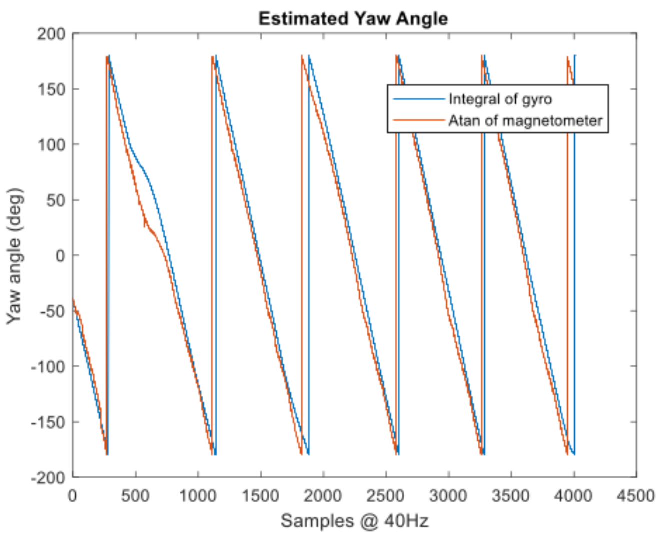


Figure_2a: Raw Magnetic Field in X Y Z directions.



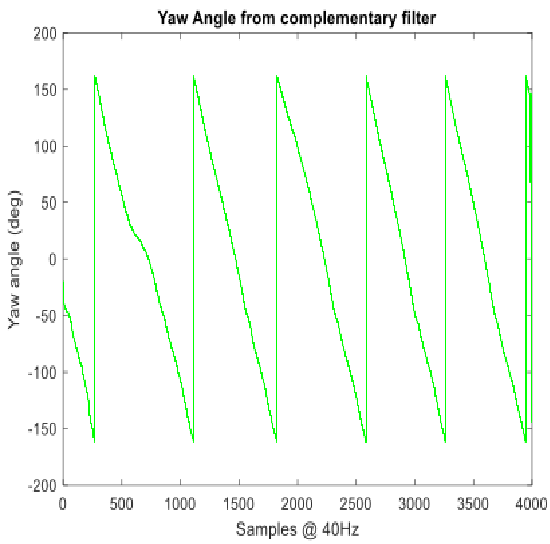


Figure_2b: Estimated Yaw Angle

Figure_2c: Yaw angle from complementary filter.

At different points along the data set the two estimates are more closely aligned than others. This is likely due to imperfections in the driving, driving in non-perfect circles and having to slow down slightly abruptly for traffic. However these differences are also due to the inherent differences in the two sensors we are comparing.

The magnetometer is well suited for steady measurements over a long time period but lacks the ability to pick up quick movements (like the non-ideal movements in my driving described above). The gyro on the other hand is better suited for measurement abrupt movements but this comes with an associated drift over time.

It is for these inherent differences in the two sensors used that what we really want to do to estimate yaw angle is to perform a complementary filter on the two measurements. A low pass filter is used on the magnetometer data to keep its steady low drift but remove any high frequency noise while a high pass filter is used on the gyro data to keep high frequency measurements but remove the bias from low frequency drift being integrated. The result of this complementary filter is shown in Figure_2c above.

Here we can see not too much of an improvement over the original data as our magnetometer data was fairly free of noise and our gyro data was mainly free of bias. On a larger data set where more drift was accrued this method would be much more helpful to get really smooth estimated data out.

Part 2: Estimating Forward Velocity

Accurate estimation of velocity using IMU data is crucial, but raw acceleration data can contain noise and bias that lead to significant errors in velocity estimates. To improve accuracy, preprocessing techniques such as filtering and calibration should be employed to remove noise and bias from the raw data.

Careful integration of acceleration data using appropriate techniques that account for sensor properties and data characteristics is also important. Limitations of the measurement system, such as drift over time, should be taken into consideration and addressed. Scaling and mean subtraction can be used to compare velocity data obtained from accelerometers with GPS data. Although the resulting velocity estimates from both sensors may follow the same trend, some bias may still be present and require additional filtering and correction.

Nonetheless, comparing velocity data from multiple sensors can provide valuable insights into motion and be useful in tracking vehicles.

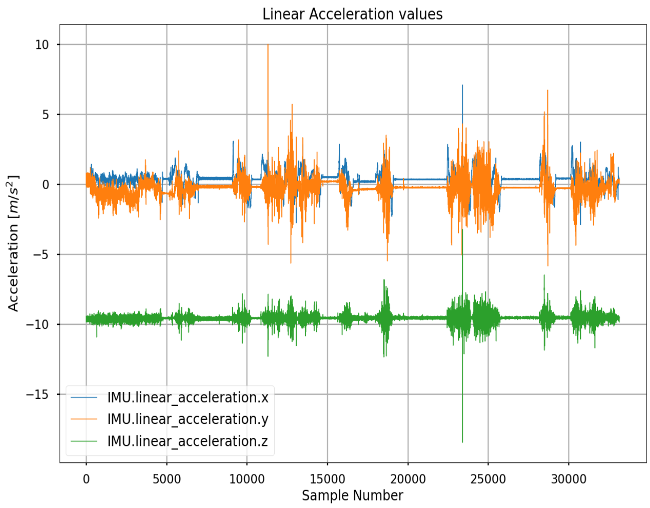


Figure_3a: Raw Linear Acceleration values

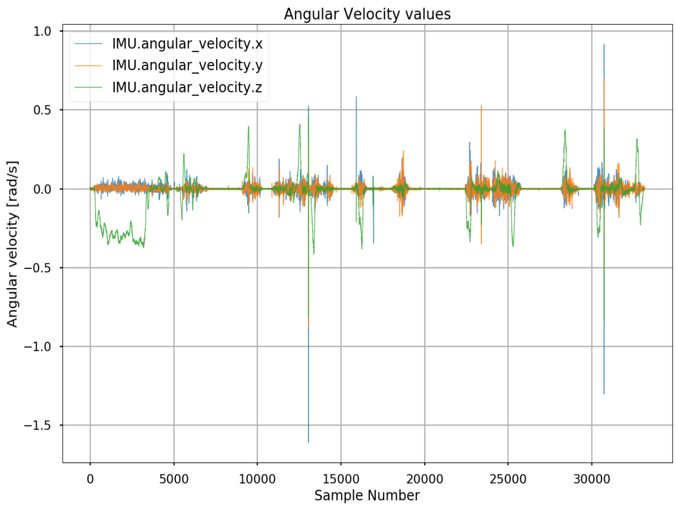


Figure_3b: Raw Angular Velocity values

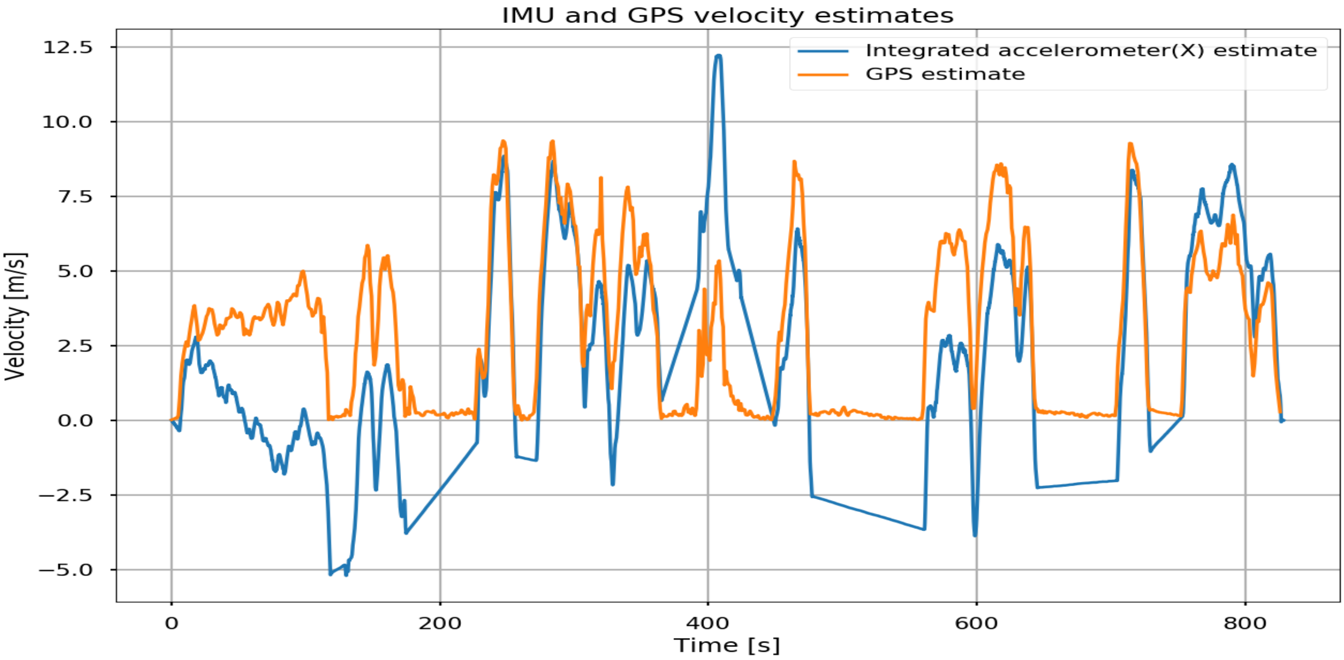


Figure_3b: IMU and GPS velocity estimates

<u>Part 3: Dead Reckoning</u>

Dead reckoning using IMU can be done by multiplying our estimated forward velocity by yaw angle. Yaw angle or heading provides the instantaneous angle the car is turned at every step of time during the drive. If we know this information about heading we can use it to break our velocity data into its X and Y components using sin() and cos() operators. Taking the instantaneous velocity at every point in time and keeping track of the sum means we can compute the displacement. This displacement or position can then be compared to what we saw with the GPS.

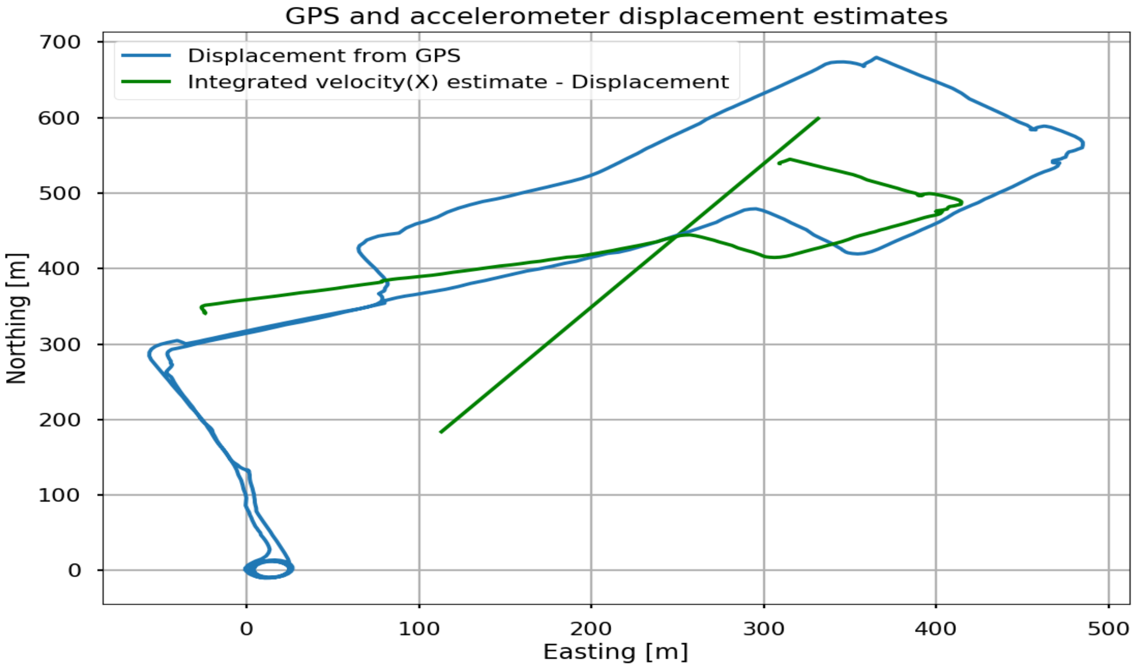


Figure_4: GPS and IMU displacement estimates

The trajectories xe and xn are calculated by integrating the velocities estimated from gps and imu.Respective graphs are plotted. A very good trajectory was plotted with gps data, nevertheless trajectory using the complementary filter output was coinciding with gps data with slight variation.
Using the above formula, the xc value is determined.
Here Y is always 0, as there is no skid in the y direction. The equation is rearranged to find the xc values as follows:

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

It is to be noted that the velocity is 0 at some points, and dividing by 0 can produce undefined values. Hence, we run a for loop to avoid points with 0 velocity, to find an xc for entire data, and find the mean of those values to estimate xc .

It can be seen that the value of xc comes to -0.2659 m This means that the inertial sensor is placed 26 centimetres away from the centre of gravity of the car. This value is not very accurate, as the xc value should usually be in the order of 1/10 th.

Dead reckoning performance is not always consistent with the specified standards due to several sources of error and noise in the data. In our situation, we noticed significant deviations and biases in the IMU-based position estimates, especially when compared to GPS readings. This can be attributed to sensor drift, external interferences, and inaccuracies in data correction and filtering.
Therefore, it is crucial to verify the accuracy of dead reckoning algorithms using ground truth measurements or other reference sources to ensure their dependability in real-world applications.

The respective means in all the cases are shown in the below picture:

```
Creating GPS plot

Creating Linear Acceleration plot
IMU.linear_acceleration.x: Mean = 0.372685372918
IMU.linear_acceleration.y: Mean = -0.251284787594
IMU.linear_acceleration.z: Mean = -9.56095951002

Creating Angular Velocity plot

Angular Velocity Data:
IMU.angular_velocity.x: Mean = 0.00119300144823
IMU.angular_velocity.y: Mean = 0.00051751472363
IMU.angular_velocity.z: Mean = -0.0298257563963

Creating Magnetometer plot

Plotting magnetometer calibration circles
Magnetic Hard Iron offsets:
Mx bias: -22.75 mGauss, My bias: -7.95 mGauss

Plotting magnetometer hard iron corrections

Eigenvalues: [34858.70722611 19859.04801227]
Eigenvectors: [[ 0.99992754  0.01203806]
 [-0.01203806  0.99992754]]

Plotting hard and soft iron corrections
Rotation Matrix:
[[ 0.99992755  0.01203718]
 [-0.01203718  0.99992755]]
Theta corrected maxs/mins:
X: 265.64484895/-265.84586991
Y: 233.151850164/-233.107312586
Scale Factor (minor/major axis length): 0.877682556563
```

Figure_5: Linear Acceleration, Angular velocity, Magnetometer bias and the other values computed