Northeastern University
EECE 5554 Robotics Sensing and Navigation
Lab 4 submission

Yash Mewada

April 7, 2023

1 Setup

For this Lab, the setup consisted of two sensors.

- A standard GPS Puck BU-353S4 (Fig left)
- Vectornav VN-100 IMU/AHRS (Fig right)

collected in an open area as well as the data collected in an occluded space.

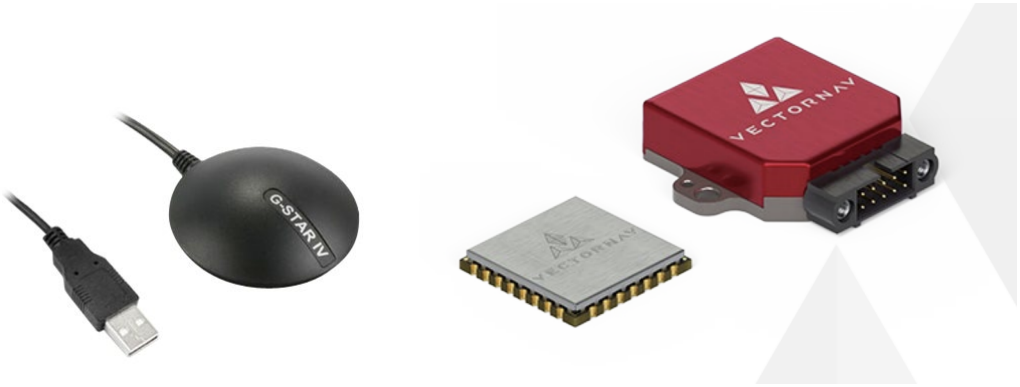

Figure 1: Setup

The purpose of this lab was to build a navigation stack using IMU and GPS drivers and study the effects of different things and atmospheric/environmental effects on each of these sensors.

2 Magnetometer Calibration

Magnetometers have two sources of distortions as stated below.

- **Hard Iron distortions** - Any device that produces its magnetic field and affects one of the magnetometers.
- **Soft Iron distortions** - Any device that produces alterations in the existing magnetic field.

Our data had both the above distortions due to the fact that the data was shifted from the center and also skewed in the x-axis Calibration steps performed.

- Perform ellipse fitting on the magnetometer
- Find the parameters of this ellipse.

$$x_0, y_0 = Center\ of\ Ellipse$$
$$q, r = Major,\ Minor\ Axis\ of\ Ellipse$$
$$\phi = Rotation\ of\ Ellipse\ wrt\ X.$$

- Subtract each reading with the obtained center coordinate. (Removing Hard Iron distortions i.e Bias)
- Rotate each reading of both the x and y-axis of the magnetometer to align it with the cartesian coordinates.
- Divide readings of magnetometer x-axis by the scale of S. Where q is...

$$S = \frac{q}{r}$$

  The reason for this is our data was skewed from the x-axis and not on the y-axis.

- Rotate by the negative of the previous rotation angle performed to maintain the integration of the magnetometer data.

Below is the representation of how the magnetometer calibration worked out for our data.

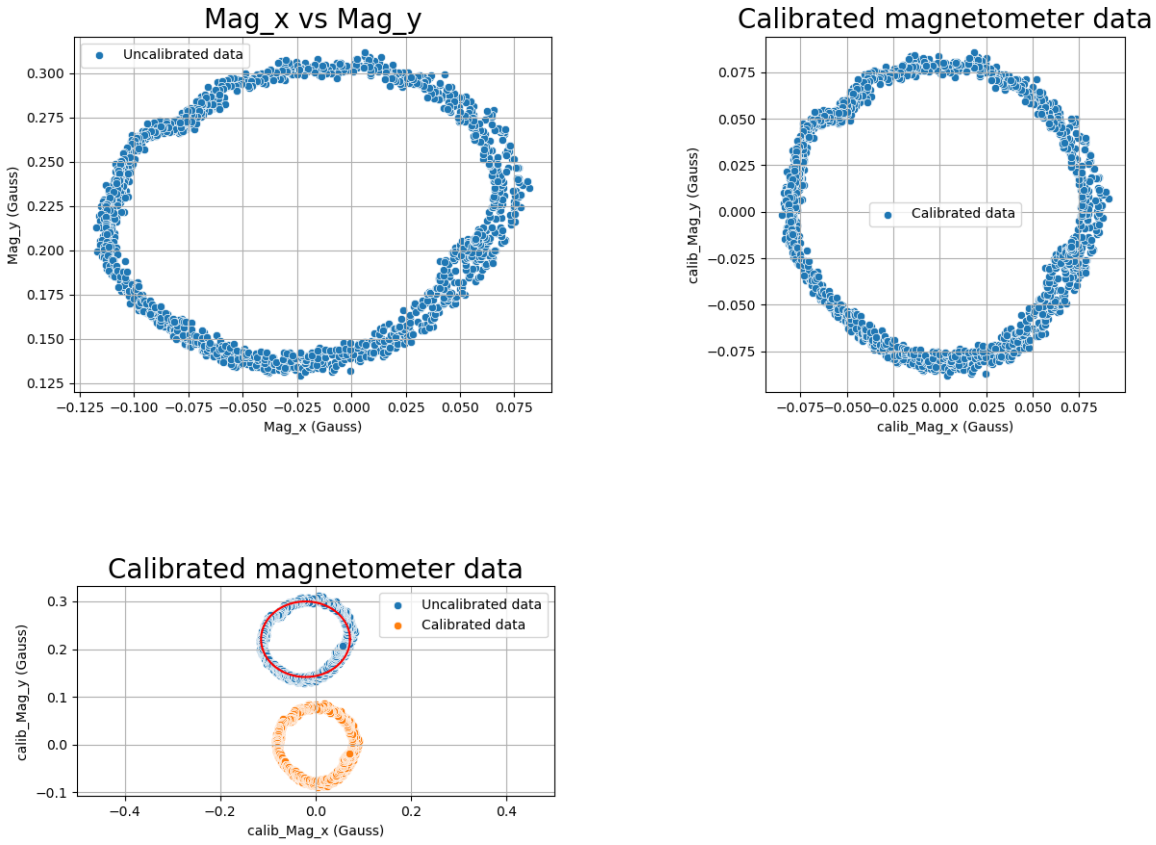

Figure 2: Pre calibration, calibrated data, and Compared data

3 Complimentary Filter (CF)

Estimating accurate Yaw/heading by combining Gyro and Magnetometer data is one of the techniques performed in certain cases. But there's a problem with directly combining these data. The Gyro data is affected by the drift (*integrating even the smallest constant or noise over time causes the value to increase exponentially*) and the magnetometer data is affected by the distortions of nearby ferrous materials and fields. So the CF is integrated by applying a combination of ***High Pass Filter*** on Gyro data to filter out the low-frequency data (velocities that change slowly over time) and ***Low Pass Filter*** on the Magnetometer data to filter out the high-frequency noise i.e rapidly changing magnetic fields in the

surrounding. Hence we can have a better Yaw estimation. Steps were performed for CF and from now onwards I removed the data where we were moving in circles as it was done for calibrating the sensor.

- Estimate yaw from the magnetometer.
- Perform a Low pass filter on yaw from the magnetometer.
- Estimate yaw from Gyro readings.
- Perform a high pass filter on yaw from the gyro.
- Add both the high pass and low pass filter readings based on the weighted sum to output a CF.

Estimating yaw from gyro and magnetometer used the below equation.

$$magYaw = \arctan(magY, magX),\ gyroYaw = \arctan(gyroY, GyroX) \tag{1}$$

The reason Cf is used as a weighted in this case is that the Gyro gives better output in a dynamic environment where as the magnetometer/accelerometer outputs better in static environments. Below are the parameters used in my case.

$$CF = \alpha HPF + \alpha LPF$$

CF = Complementary Filter HPF = High Pass fillter LPF = Low pass filter $\alpha = 0.8$

Figure 3: Different filter outputs

Parameters	Low pass mag	High pass gyro
Cutoff Frequency (f_c)	0.06	0.0001
Order	3	3
Nquist Constant (Nqy = f_c/NC)	1.0	1.0
Weight for CF	0.2	0.8

Note here unwrap function was used to wrap the yaw value to the mirror value to prevent it from negation. This helped in further analysis part. Also, the data was plotted across zero by subtracting it from the first value.

4 Quantifying the goodness of Yaw estimation

Based on the previous results I would prefer using the Complementary Filter based yaw, due to the reason that it mitigates the drift from Gyro readings and noise from the magnetometer readings. Also using only the gyro estimated yaw is not a good choice due to the presence of **angle random walk** and Gyro bias Instabiltiy (deg/hr) which causes the variance to grow over time.

5 Adjustments to forward velocity from accelerometer

Now while setting the pitch equal to zero in our case, I accidentally rotated the IMU a bit more towards the left direction, and hence in our case, the forward velocity was actually the component of linear acceleration in the X and Y axis.
The accelerometer had a linearly increasing trend in its data and a bias in its data. So initially that trend was removed along with the bias by finding the window where the IMU was stationary (this does not guarantee that it was stationary as it might be on constant velocity) with a **threshold of 0.3** m/s^2. Steps were performed to adjust the velocity with GPS velocity.

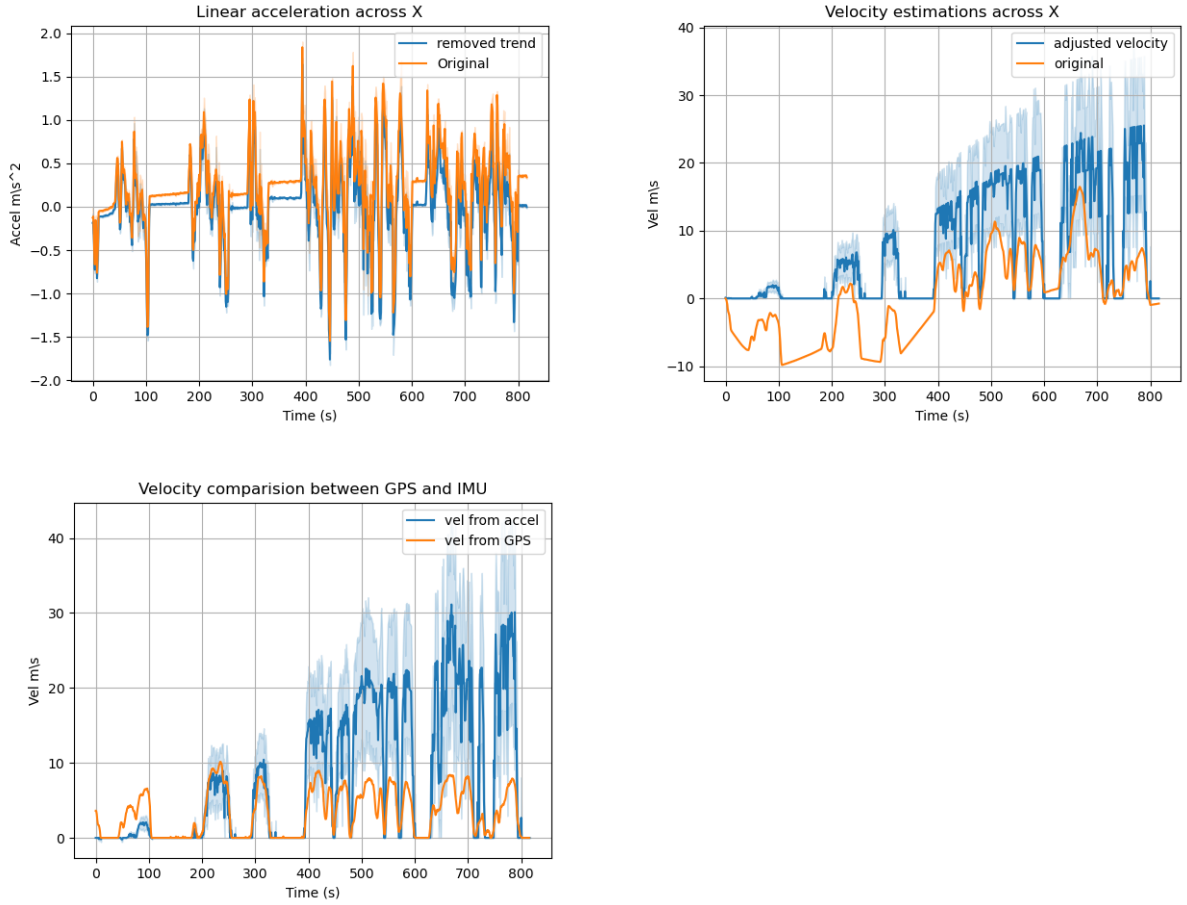

Figure 4: Velocity trends from IMU and GPS

- Performed Low pass filter with $f_c = 0.1$. (Removed instantaneous changes in data)
- Performed High pass filter with $f_c = 0.001$ (Removed drift in data)
- Performed detrend of data with window approach.
- Integrated the detrend data to estimate the velocity
- Apply again the Low pass filter on this velocity to smoothen the data with $f_c = 0.001$.

- checked if the sensor was stationary in all 3 axes by the threshold of 0.3, then set that velocity index to zero. (Removing the offset from all the 3 axes) else subtract the previous value from the current value so that the velocity does not integrate into an exponential value.
- Plot the results.

The algorithms used here are **static thresholding**, **signal filtering**, and **zero velocity updates**. But as it is evident from the above graphs even after performing such filtering and removing the bias from the velocity still the velocity reaches $40m/s$ and is very sharp. The reason for this was as our accelerometer gives instantaneous acceleration, the moment brakes were applied the acceleration value was suddenly changed.
Also, we cannot use GPS velocity to correct the IMU velocity as if GPS had a corrupted then the whole velocity might get garbage.
Also another source of such jerky measurements from imu is because IMU has a frequency of 20hz and GPS has a frequency of 2hz which explains the smoothness of the data.

6 Discrepancies between GPS and IMU-based velocities

The only change in similarities here is the magnitude of the IMU-based velocity. This is due to the sensitivity of the IMU sensor. Also due to the presence of Velocity random walk which has the unit of $m/s/\sqrt{s}$ (viz integrating the white noise in accelerometer causes the noise to increase or decrease based on the magnitude and bias in noise.)

7 Compare $\omega\dot{X}$ and $\ddot{y}_{obs}$

The equation for forward velocity based on linear acceleration in X and Y axis is a component of angular velocity in the Z axis, the Center of Mass of the vehicle, and linear acceleration in the X axis.

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

$$\ddot{y}_{obs} = \ddot{Y} + \omega\dot{X} + \dot{\omega}X_c \tag{3}$$

Consider the ideal condition and place $X_c = 0$ and having only steering motion hence placing $\dot{Y} = 0$ the equation reduces to...

$$\ddot{x}_{obs} = \ddot{X};\ \int \ddot{x}_{obs} = \int \ddot{X};\ \dot{x}_{obs} = \dot{X} \tag{4}$$

$$\ddot{y}_{obs} = \omega\dot{X} \tag{5}$$

Hence following the above steps theoretically the linear acceleration in the Y axis should be equal to the component of angular rate in Z and forward velocity in the X axis (**viz Linear acceleration in Y axis is Spatial acceleration**).

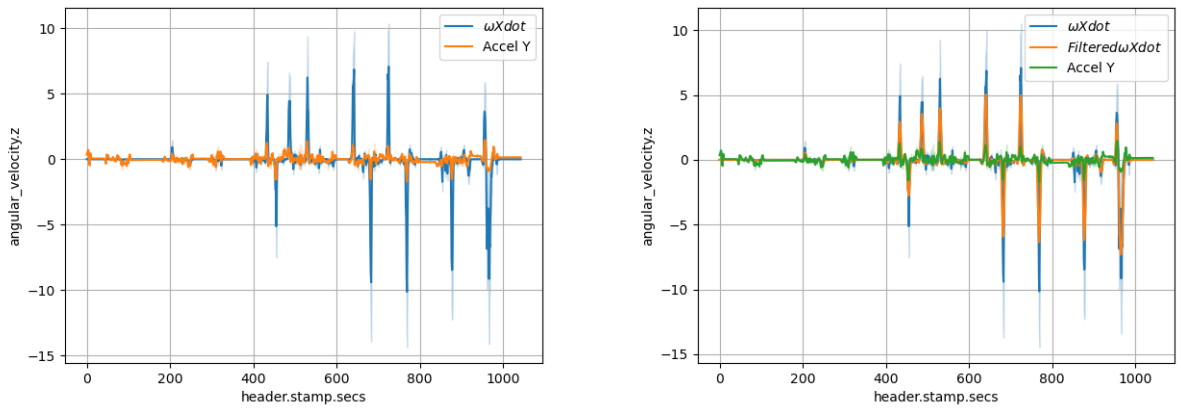

Figure 5: Linear acceleration in Y axis from IMU and computed from velocity

From the above figures, we can see that the acceleration estimation from the forward velocity in the X-axis and the yaw rate has very significant differences due to the presence of **Gyro Bias instability, angle random walk, and velocity random walk**. These parameters cannot be solved by just applying any kind of digital filter.

8 Estimating the trajectory from IMU.

Below is the estimation of the trajectory based on the computations done above. After plotting the data it was found that the imu trajectory was actually scaled by **3 times** and was shifted by **105 degrees**. After setting these biases the final output is as below. Initially, the data did not match

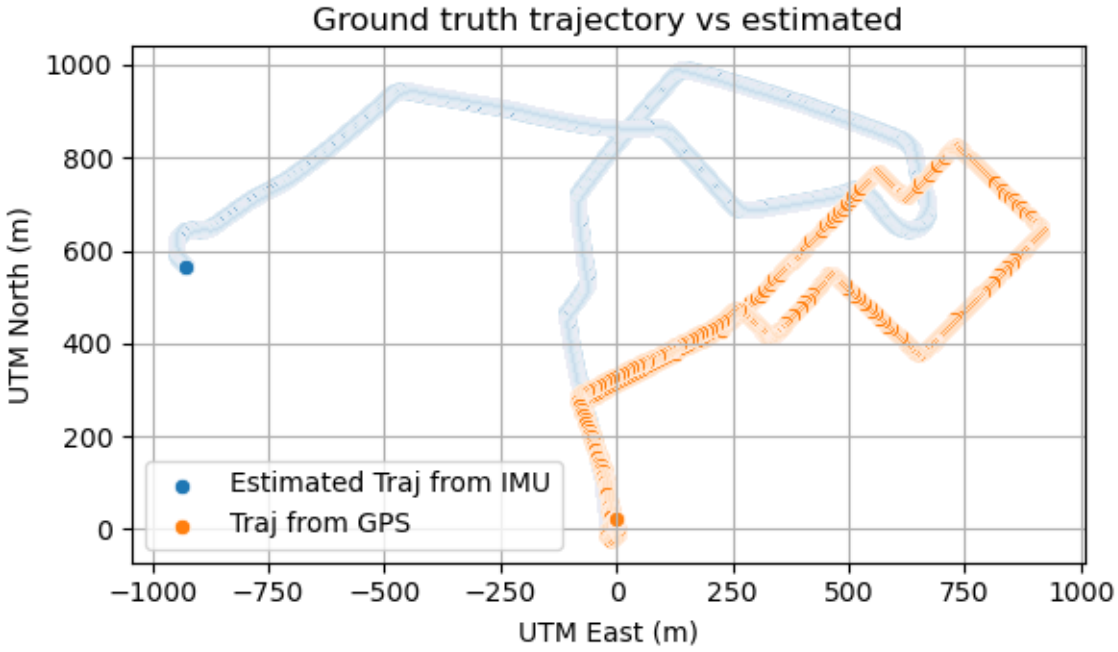

Figure 6: Estimated trajectory from IMU and comparing with GPS

at all. But if we consider the data after scaling and bias removal we can see that for at least **300m** the estimation of IMU trajectory matched the one of GPS and for at least **200s** but as soon as the car started taking turns and changing its direction the IMU's estimation of velocity and trajectory got worse.

The term dead reckoning involves estimating the velocity of the vehicle and how long that velocity has been maintained and using that information to estimate the trajectory it took.

Performing dead reckoning based on only IMU did not result better due to the presence of angle random walk and velocity random walk and rate random walk as we saw in the previous lab. In order to solve this we can perform a Kalman filter of the extended Kalman filter to remove these errors or we can use wheel encoder readings in conjunction with IMU to solve this drift.

This drift can be caused by various factors, such as temperature changes, vibration, and sensor calibration errors.

Therefore, while IMU-based dead reckoning can provide useful position estimates in some situations, it is generally not the most accurate method for long-term navigation. Other sensors, such as GPS or wheel speed sensors, can be used in conjunction with an IMU to improve the accuracy of the dead reckoning calculations.

9 Extra Credit

Based on the equations from the above sections and using the angular and linear accelerations in the Y axis and Z axis the estimated value for **Xc** is **0.589 m**. Based on these calculations the value of Xc kind of makes sense not considering the shift in the y-axis. A more detailed approach below can be considered. Given the equation for the linear acceleration of the center of mass which is the component of linear acceleration in the X axis and the spatial acceleration as ...

$$x\ddot{x} = \dot{v} + \omega \times v = X\ddot{X} + \dot{\omega} \times r + \omega \times \dot{X} + \omega \times (\omega \times r) \tag{6}$$

Now considering Newton's Euler motion equations. we can rewrite the below equation as...

$$\sum \vec{F} = m\vec{a}_r \tag{7}$$

$$\sum \vec{M}_r = I_r\vec{\alpha} + \vec{\omega} \times I_r\vec{\omega} \tag{8}$$

Where I_r is the moment of inertia of the center of mass, and m is the mass of the body.

$$\vec{a}_r = \vec{a}_A + \vec{\alpha} \times \vec{c} + \vec{\omega} \times (\vec{\omega} \times \vec{c}) \tag{9}$$

$$\sum \vec{M}_r = \sum \vec{M}_A - \vec{c} \times \sum \vec{F} \tag{10}$$

$$\sum \vec{F}_A = m\vec{a}_A - m\vec{c}\times\vec{\alpha} + m\vec{\omega}\times(\vec{\omega}\times\vec{c}) = I_r\vec{\alpha} + m\vec{c}\times\vec{a}_A - m\vec{c}\times(\vec{c}\times\vec{\alpha}) + \vec{\omega}\times I_r\vec{\omega} + m\vec{c}\times(\vec{\omega}\times(\vec{\omega}\times\vec{c})) \tag{11}$$

$\vec{c} = \vec{r}$

$$\begin{bmatrix} \sum \vec{F}_A \\ \sum \vec{M}_A \end{bmatrix} = \begin{bmatrix} m[\vec{r}\times] & -m[\vec{r}\times][\vec{r}\times] & -m[\vec{r}\times] \\ 0 & I_r & 0 \end{bmatrix} \begin{bmatrix} \vec{a}_A \\ \vec{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \vec{\omega}\times & \vec{\omega}\times \end{bmatrix} \begin{bmatrix} I_r\vec{\omega} \\ m\vec{r}\times\vec{a}_A \end{bmatrix} \tag{12}$$

where $[\vec{r}\times]$ is the 3x3 matrix operator for the cross product with $\vec{c}$:

$$[\vec{r}]\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{13}$$