Northeastern University

MS ROBOTICS

LAB 4 - REPORT

Abstract

Navigation with IMU and Magnetometer

Hemmanahalli Manjunath Ruthvik

Contents: Page No

Analysis Report

Introduction

In this Lab, we are going to build a navigation stack using two different sensors – GPS & IMU, by understanding their relative strengths and & drawbacks and getting knowledge of sensor fusion**.**

Part 1 – Calibration of Magnetometer: Data collected in circles

The data was collected from the IMU and GPS using the device driver. A total of around 6258 points of data were collected. The data collected was then plotted with different parameters. The magnetometer data is plotted before making any corrections in Fig 1.

Magnetometer Data Before Correction:

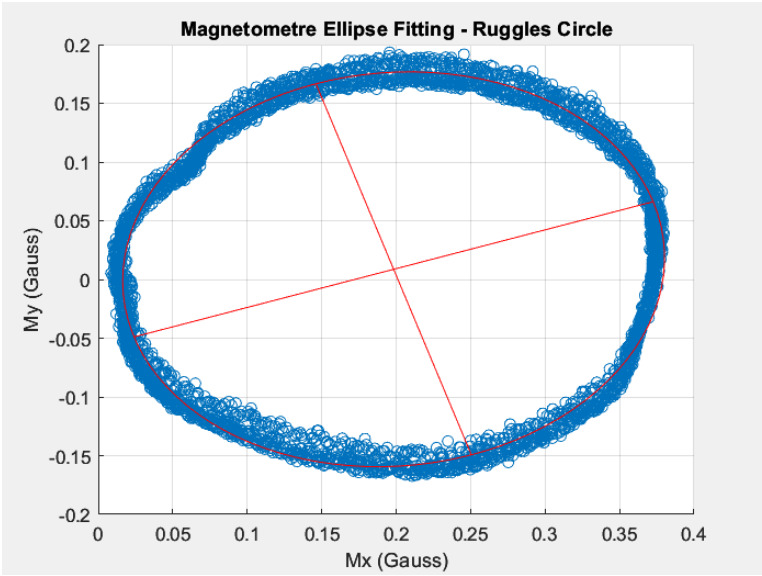


Fig 1 – Magnetometer data before correction (x gauss vs y gauss)

The data was collected in the Ruggles circle at Northeastern University in the nuance car. From Fig 1 we could see that even though the route traveled is the circle we could see a few data points scattered around the circle this is because of the presence of hard and soft iron effects. Even with a noiseless magnetometer, it will still return an incorrect measurement simply because of the presence of hard and soft iron.

Hard Iron Distortions: Hard iron distortions are created by objects that produce a magnetic field. A speaker or piece of magnetized iron for example will cause a hard iron distortion. If the piece of magnetic material is physically attached to the same reference frame as the sensor, then this type of hard iron distortion will cause a permanent bias in the sensor output.

Soft Iron Distortions: Soft iron distortions are considered deflections or alterations in the existing magnetic field. These distortions will stretch or distort the magnetic field depending upon which direction the field acts relative to the sensor. This type of distortion is commonly caused by metals such as nickel and iron. In most cases, hard iron distortions will have a much larger contribution to the total uncorrected error than soft iron.

These errors need to be corrected before we can actually estimate the incident, as the data was collected in the no ideal spot, there might be some other factors too that have affected the output. Some of the factors might be Banking angle and elevation.

Banking Angle: It is the angle at which the vehicle is inclined about its longitudinal axis with respect to the horizontal, it could be calculated by taking the tan inverse of the mean of linear acceleration in x and z directions.

Elevation Angle: It is defined as the angle formed between the x-axis and the ground. It is calculated by the tan inverse of linear acceleration in the x and z directions.

Even though banking angle and elevation angle could be the cause of error they would not contribute to such major distortions or the effects caused to magnetometer data would be not that accountable, so we move on to evaluating the hard and soft iron error.

1. Correction of Hard iron and soft iron Effects:

If we can plot the data by rotating the magnetometer 360 degrees,without the presence of hard or soft iron effects, then, the result would be an almost perfect circle centered around the origin. But these effects are always present due to the earth's magnetic field, external magnetic influences, etc.
The graph below displays the updated data with a circle whose center is approximately (0,0). As a result, the hard iron effect is reduced.

However, since the hard iron distortion, not the soft iron, is cumulative, soft iron distortion cannot be reduced only by removing the offset. Since soft iron distortion is the outcome of a material that influences a magnetic field rather than creating one on its own, it is not a contributor to the earth's magnetic field.

As a result of the material's orientation in relation to the magnetic field and the sensor, it is not constant. Therefore, a complicated approach is needed to analyze light iron distortion as compared to strong iron distortion.

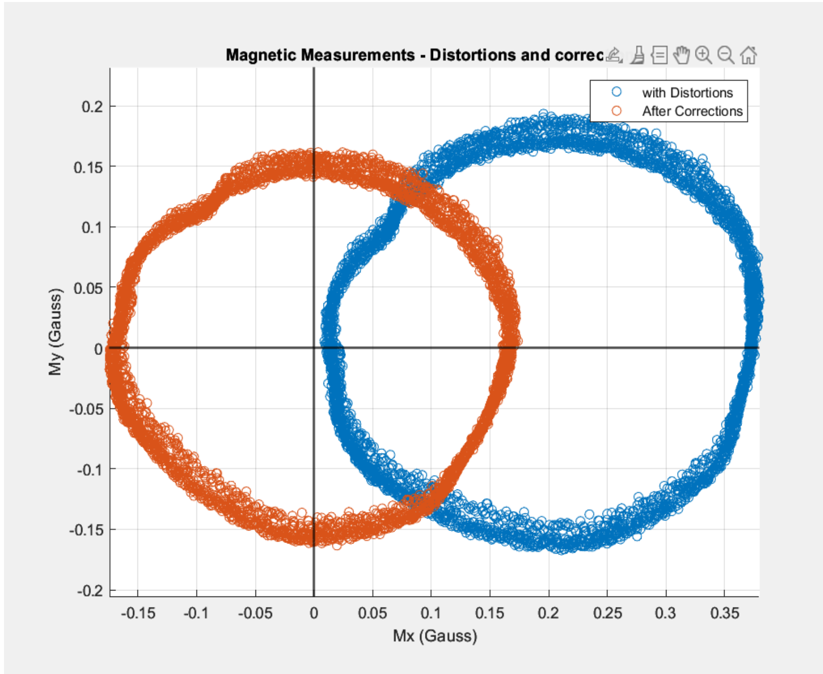


Fig 2 – Magnetometer Data after Correction

From Fig 2 If we can fit an ellipse onto the data, we can comprehend soft iron corrections easier. With an almost circular ellipse fitting, the center of the red circular plot is on the origin. This demonstrates that there are few hard or soft iron effects in our data and the blue circular plot is with the distortions.

Yaw from Magnetometer and Yaw integrated from Gyro - Data collected from GPS and IMU:

We will be using the complementary filter; it is actually considered to take slow-moving signals from an accelerometer and fast-moving signals from a gyroscope and combine them. The accelerometer gives a good indicator of orientation in static conditions. A gyroscope gives a good indicator of tilt in dynamic conditions.

The magnetometer signals are made sure to pass through a low-pass filter and the gyro signals are passed through a high-pass filter, thereby combining the results, we get the final yaw angle and the frequency of the low-pass filter and high-pass filter should add up to 1. So, I'm considering hpf = 0.003 and lpf = 0.001, the gain of mag = 0.98, and the gain of imu = 0.02, and cut off frequency used is (0.003 Hz), in figure 3 we can see that the red line represents the raw data from the imu and the blue line represents the calibrated magnetometer yaw. I calculated the yaw from the magnetometer and gyroscope using the raw data from the imu as a reference, I would always trust the estimate of yaw from the magnetometer because it is matching with the calibrated magnetometer yaw.

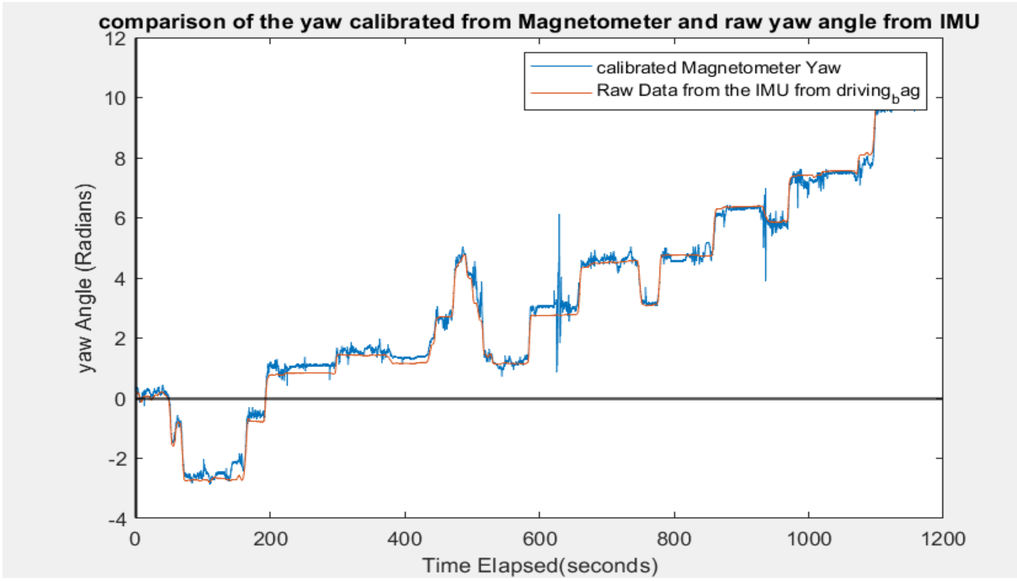


Fig 3 – Yaw Angle (Rad) vs Time (Sec) [Raw Data from IMU]

After scaling the data by scaling factor, so from the below data actually follows the actual data.

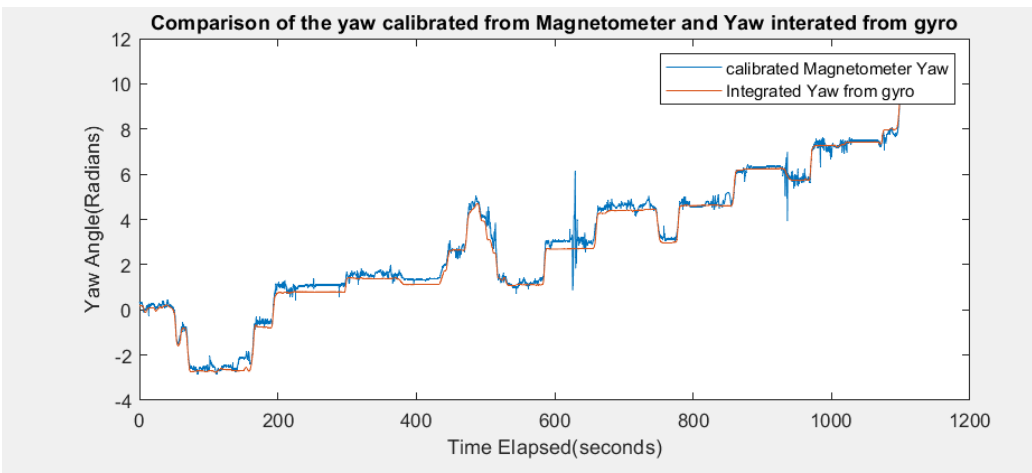


Fig 4 – Yaw Angle (Rad) vs Time(sec) [Integrated Yaw from Gyro]

Figure 4 shows that the data are correctly matched when a complementary filter is applied using a mix of corrected magnetometer and gyroscope data.

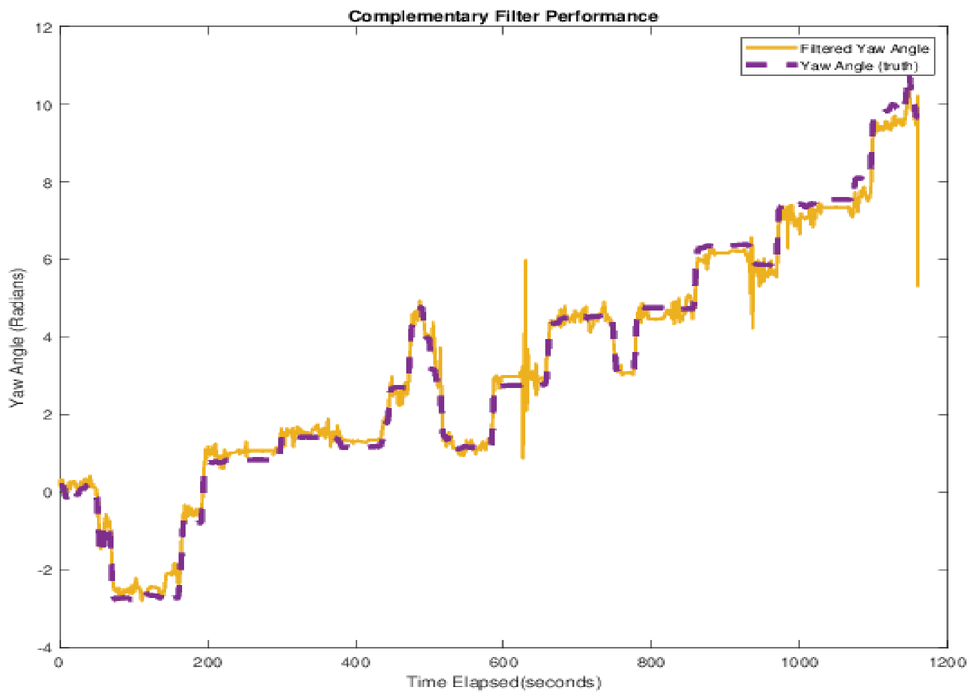


Fig 5 (a)

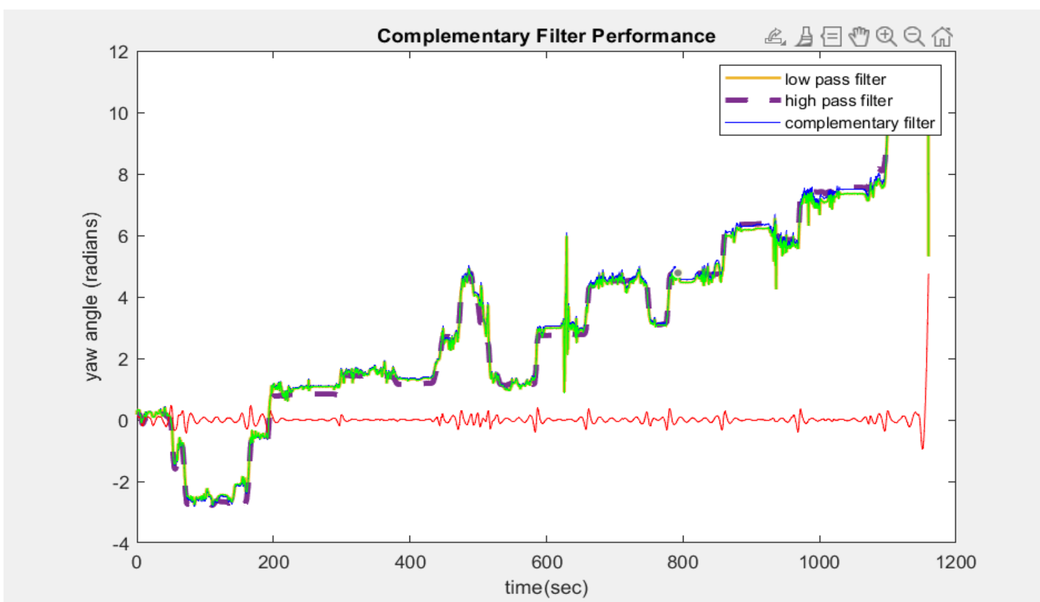


Fig 5(b) – Yaw Angle (Rad) vs Time (Sec) [Complementary Filter]

The magnetometer is reliable and it is also quite aligned with the raw data, according to Fig. 5 (a), which compares the filtered yaw angle and the raw data from the IMU. In the prior data, we observed calibrated data while driving in circles. Here, we have calibrated the same data while driving, which produces the desired result and is less noise-prone. Since the data was first collected in radian quaternion and then transformed to Euler form in the driver, the angle in the plot above is in radians and from Fig 5 (b) The filtered Yaw is obtained through the complimentary filter in which I have added the data of the low pass filter and high pass filter with the fine-tuning we get 2 variables in their weights.

$\alpha = 0.98$ (in the weight of magnetometer Data)

$\beta = 0.02$ (in the weight of the gyro Data)

Part 2 - Forward Velocity Estimation:

GPS measurements and velocity constraints are fused with measurements from the inertial sensors to estimate the position and velocity of the vehicle. The specific force, measured from a three-axis accelerometer, and the angular velocity, measured from a three-axis gyroscope, are integrated to find the vehicle's position, velocity, and attitude. However, due to error accumulation, the results soon drift away from their actual values by integrating the forward acceleration. shown in the graph's forward velocity. The IMU's velocity is calculated using integration, and it is then contrasted with the GPS's velocity, which is obtained by taking the derivative of the UTM position data with respect to time.

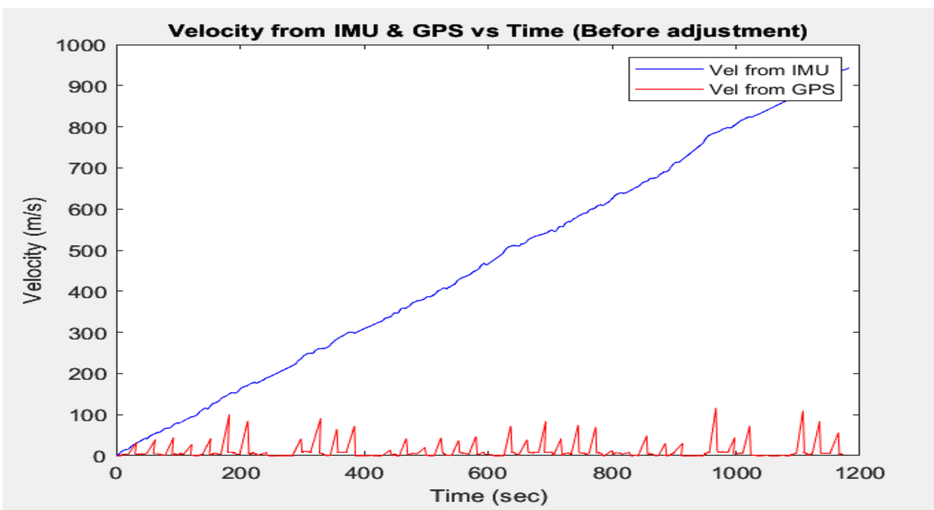


Fig 6 – Velocity (m/s) vs Time (Sec) [Before Adjustments]

There is a significant cumulative error between the true and calculated velocities as a result of drift and noise. The accelerometer needs to be further adjusted. IMU and GPS scales are vastly different. This is due to the different sampling rates of the two sensors. IMU uses a sampling rate of 40 while GPS uses 1. So, there is a different scale for both readings in Fig 6, So if we bring them both to one scale with compensation of velocity scale also, Fig 7 describes the representation after compensation. As a first step in correcting the above figure, stops were located at various places and intervals of time, and the acceleration data was divided into several intervals. At each interval, a new bias was calculated and subtracted from the next acceleration value in the interval. The integration was carried out at each interval, and the integrated velocity array was obtained by adding the intervals back together.

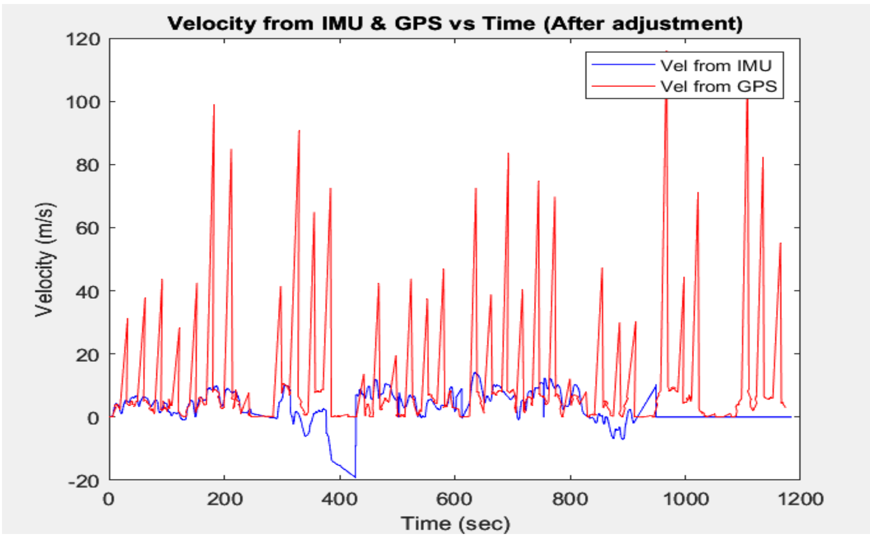


Fig 7 - Velocity (m/s) vs Time (Sec) [After Adjustments]

From Fig 7 we could see that it is almost adjusted to the data of the GPS except there are spikes that are present in the GPS data which might be due to the stoppages and bumps in the driving path and the ride should be smoother to eliminate the spikes.

PART 3 - Dead Reckoning with IMU:

Dead Reckoning is the process of calculating the current position of some moving object by using a previously determined position in other words When the location and orientation of a device are important for navigation, inertial sensors are typically used. Information regarding the orientation of the sensor is obtained by integrating the gyroscope measurements. Double integration of the accelerometer values after removing the effects of the earth's gravity reveals the location of the sensor. The orientation of the sensor must be understood to be able to deduct the effects of the earth's gravity. Therefore, when it comes to inertial sensors, the assessment of the sensor's position and orientation are inextricably related.

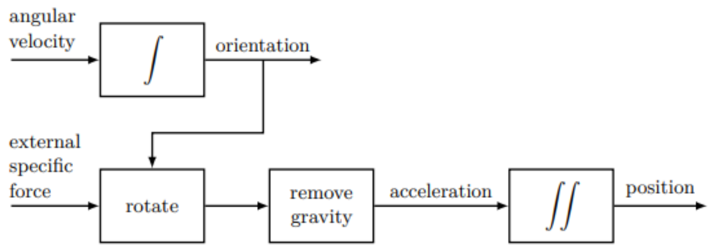


Fig 8 – Representation of the above-explained process

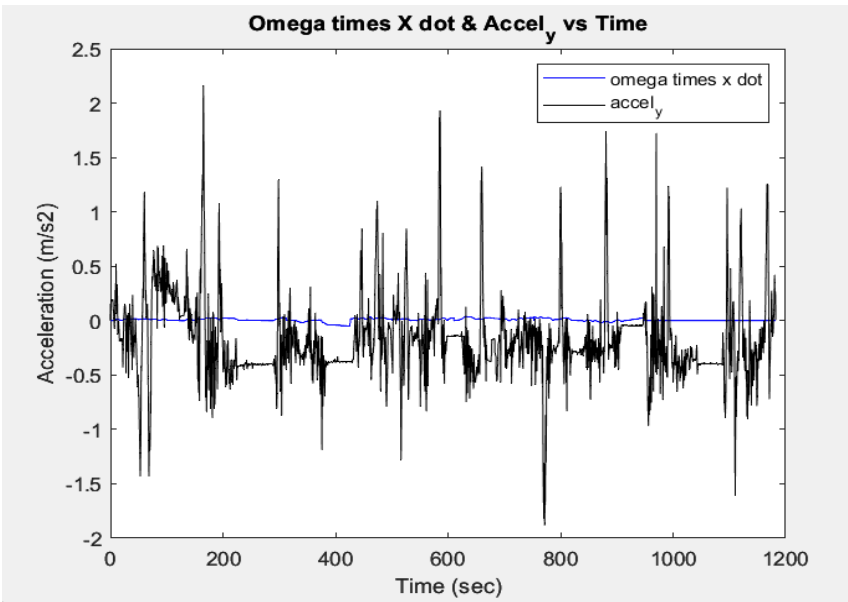


Fig 9 – Acceleration (m/s2) vs Time (Sec)

From Fig 9 the biases are calculated for the drive and subtracted off at each stop, then the two values of the lines in the figure must be equal. However, when integrating for X, the biases are subtracted, which explains why the values are offset from each other. We can see that the overall trend between Wx and y.obs is not very stable.

Because we assumed xc to be zero, but in reality, it is located at a different physical location, the data is noisy and a bias may be noticed. the noise might be due to the sensor's vibrations caused by changes in the wire that is attached to it the car's motion, and human mistakes are all possible causes of the noise. However, the inertial sensor's acceleration measurement is incredibly erratic. It contains many high-frequency parts. Additionally, we can see that each period has a different drift. This may be the case because as the low-frequency white noise is integrated, a random walk is produced, and the errors continue to grow. We can eliminate the low-frequency component by using a lowpass filter with a suitable cutoff frequency.

IMU and GPS Trajectory:

According to the preceding graph, the displacement estimated using GPS and IMU data is initially relatively close, but as the integration progresses, the graph begins to produce a bias after a few turns, which results in an offset. The data is rotated roughly 90 degrees counter-clockwise to match the GPS data. We were able to estimate this position using the final yaw that was provided by the data that had been IMU- and magnetometer-filtered.

Vestimated = ∫Accadjusted

Va = Vestimated x sin(yaw_anglemagnetometer)

Vb = Vestimated x cos(yaw_anglemagnetometer)

Xa= ∫Va Xb = ∫Vb

Fig 10 demonstrates the GPS UTM trajectory derived from the real data coupled with the direction of the dead reckoning location estimates. By integrating the linear velocity and adding the heading's x and y components, the position is determined. In order to align the IMU data with the actual data, the heading values have been changed. The data from dead reckoning has not been scaled in any way.

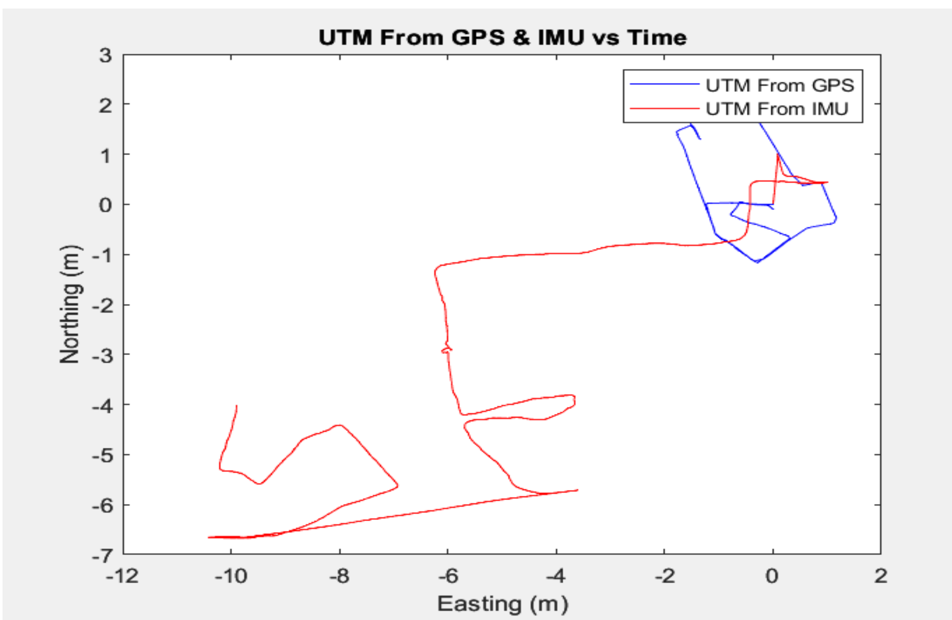


Fig 10 Easting Vs Northing (UTM from GPS & IMU)

From Fig 10 we see that we have potted the displacement graph of both the IMU and the GPS and the above is the data observed with starting and the closing point of both sensors. The data was accurate for a few terms and then it got shifted because of biases and the noise in the IMU.

The mathematical way of expressing INS working:

If we want to use the INS, we need to provide initial conditions (frequently, we provide the last values that were obtained when the GPS receiver was available). We can now define two vectors for the angular velocities and accelerations. The angular velocities vector is defined by:

- Where b is the coordinate frame of the moving IMU, i is the stationary frame, n is a local geographic frame in which we want to navigate, and f is for specific force – equivalent to say acceleration.

$$\boldsymbol{\omega}_{ib}^{b} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$$

$$\mathbf{f}_{ib}^{b} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$$

- This is the INS equation neglecting Earth's Rotation:

p stands for a position, ***v*** stands for a velocity, **R** is a rotation matrix as a function of the body orientation angles (provided below), ***g*** is just the known Earth gravity vector, and **Ω** is a skew matrix of the angular velocities.

$$\dot{\mathbf{p}}^n = \mathbf{v}^n$$

$$\dot{\mathbf{v}}^n = \mathbf{R}_b^n \mathbf{f}_{ib}^b + \mathbf{g}^n$$

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \boldsymbol{\Omega}_{ib}^b$$

- The skew matric is given by:

$$\boldsymbol{\Omega}_{ib}^{b} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

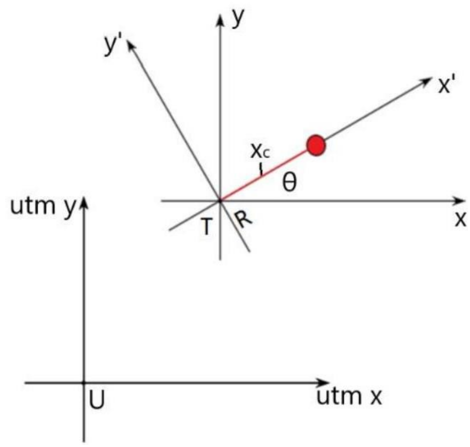
- The Rotation Matrix is given by:

$$\mathbf{R}_b^n = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\theta c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\phi \end{bmatrix}$$

- Where x represents cos and s represents sin in the above-given matrix.

We have - v = V + ω x r, $\ddot{x} = \dot{v} + \omega \times v = \dot{X} + \dot{\omega} \times r + \omega \times \dot{X} + \omega \times (\omega \times r)$, ω = (0,0,ω) , r = (xc,0,0) .

Xc Estimation:



$$v^U_{sensor} = v^U_{car} + \omega \times \rho^R_{sensor}$$

$$a^U_{sensor} = a^U_{car} + \dot{\omega} \times \rho^R_{sensor} + \omega \times (\omega \times \rho^R_{sensor})$$

$${}^R_T R^T a^R_{sensor} = a^U_{car} + \dot{\omega} \times \rho^R_{sensor} + \omega \times (\omega \times \rho^R_{sensor})$$

$${}^R_T R^T \begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix} = \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\omega} \end{bmatrix} \times \begin{bmatrix} x_c \\ 0 \\ 0 \end{bmatrix}^R + \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times (\begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix} \times \begin{bmatrix} x_c \\ 0 \\ 0 \end{bmatrix}^R)$$

$$\begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix}^R = {}^R_U R \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix}^U + \begin{bmatrix} 0 \\ \dot{\omega} x_c \\ 0 \end{bmatrix}^R + \begin{bmatrix} -\omega^2 x_c \\ 0 \\ 0 \end{bmatrix}^R$$

$$\text{in which } A = \begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix}^R - {}^R_U R \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix}^U, B = \begin{bmatrix} -\omega^2 \\ \dot{\omega} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{imux} \\ a_{imuy} \\ 0 \end{bmatrix}^R - {}^R_U R \begin{bmatrix} a_{utmx} \\ a_{utmy} \\ 0 \end{bmatrix}^U = \begin{bmatrix} -\omega^2 \\ \dot{\omega} \\ 0 \end{bmatrix}^R x_c$$

$$B = Ax_c, \text{ this gives } x_c = (A^T A)^{-1} A^T B$$

$${}^R_U R = \text{eul2rotm}([0,0,-\text{yaw}])$$

Xc is known as the estimated distance, it can be defined as the estimated distance between the inertial sensor and the center of mass point, Xc is estimated to be 0.5136 = 51.36 cm which is reasonable.

Resources:

1. Using Inertial Sensors for Position and Orientation Estimation - https://arxiv.org/pdf/1704.06053.pdf
2. How to interpret IMU sensor Data for dead reckoning - https://www.allaboutcircuits.com/technical-articles/how-to-interpret-IMU-sensor-data-dead-reckoning-rotation-matrix-creation/
3. Dead reckoning is still alive - https://towardsdatascience.com/dead-reckoning-is-still-alive-8d8264f7bdee
4. Britannica - https://www.britannica.com/technology/dead-reckoning-navigation