Northeastern University
College *of* Engineering

Department of Computer and Electrical Engineering

ME/EECE5554: Robotic Sensing and Navigation

Nuance Dead Reckoning (Lab 5 Report)

Submitted by

Sakib Azgar

Class Section: Section 2

Course Instructors: Professor Kris Dorsey and Professor Mohammad Javadi

Course TAs: Mani Chandhan Chakinala, Francis Jacob Kalliath, Aditya Bondada, Kaustubh Chaudhari, Arunbhaarathi Anbu, Thomas Consi

Background Information

In the ever-evolving field of autonomous navigation, precision and reliability are paramount. This lab is designed to further these aims by delving into the complexities of dead reckoning within the dynamic context of a moving vehicle. Building on the foundational skills developed in Lab 1 and Lab 3, this lab allows robotics to apply and enhance their understanding of navigation technologies by calculating a vehicle's path using data from both a GPS puck and a Vectornav VN-100 Inertial Measurement Unit (IMU). Applying both sensors would require the idea of using sensor fusion to enhance the accuracy of tracking the vehicle's path over specific routes. This integration of GPS and IMU data allows for a more reliable and detailed understanding of the vehicle's movements, especially in challenging environments where either sensor alone might falter.

The article "Next Generation Vehicle Positioning Techniques for GPS-Degraded Environments to Support Vehicle Safety and Automation Systems" highlights the critical importance of sensor characterization. Understanding the operational limitations, inherent biases, and potential error sources of the sensors is paramount. This detailed knowledge about sensor performance under various conditions and movements is vital for accurate calibration and error correction. It forms the foundation for effectively combining data from different sensors, such as GPS and IMUs, to precisely determine the vehicle's trajectory, especially in environments where GPS signals may be compromised [1].

GPS is often considered the superior method for navigation and positioning due to its global reach and precision. However, it lacks the capability to account for drift, a challenge that is inherent to IMUs. Since IMUs calculate position by integrating acceleration and angular rates over time, even minor inaccuracies can accumulate, resulting in notable drift. This becomes especially problematic during extended periods of GPS unavailability, where the signal is lost or blocked, and thus, the GPS cannot be utilized to rectify the accumulated drift [2].

This laboratory experience, therefore, not only equips students with the practical skills necessary for integrating and interpreting data from GPS and IMU sensors but also imparts a deep understanding of the challenges and considerations involved in advanced vehicle navigation systems. By addressing these challenges through hands-on application and analysis, students are poised to contribute significantly to the development and refinement of autonomous navigation technologies, ensuring safer and more reliable transportation in GPS-degraded environments.

Procedure

The process commenced with the integration of both GPS and IMU sensors, ensuring their simultaneous operation and data publication. A single launch file was developed to activate the GPS and IMU driver nodes concurrently, enabling the observation of gps and imu topics on the system. This setup was critical for the initial testing phase, where the functionality and data output of each sensor were verified. To enhance data synchronization, epoch time was employed in the IMU driver, replacing the previous system time approach, which facilitated more accurate time alignment between the datasets.

To ensure the integrity of the data collection setup, the IMU was securely mounted into a 3D printed holder inside the vehicle, with its orientation verified to be horizontal and the x-axis pointed forward. The GPS sensor was attached to the vehicle's roof via its magnetic backing. Both sensors were connected to a single laptop, with the system prepared to launch the necessary nodes for data collection. Electrical tape was applied to the cables to create strain relief, preventing any potential disconnection or misalignment of the sensors.

Data calibration and collection were carried out in two main phases. Initially, a ROSbag named data_going_in_circles.bag was created to record data while driving the vehicle in circular paths. This phase aimed at calibrating the sensors and ensuring the collection of a coherent dataset. Subsequently, a broader data collection drive was undertaken around a predefined urban route, avoiding tunnels to maintain GPS signal continuity. This phase was documented in a separate ROSbag named data_driving.bag, capturing extensive vehicular movement data over a diverse set of conditions.

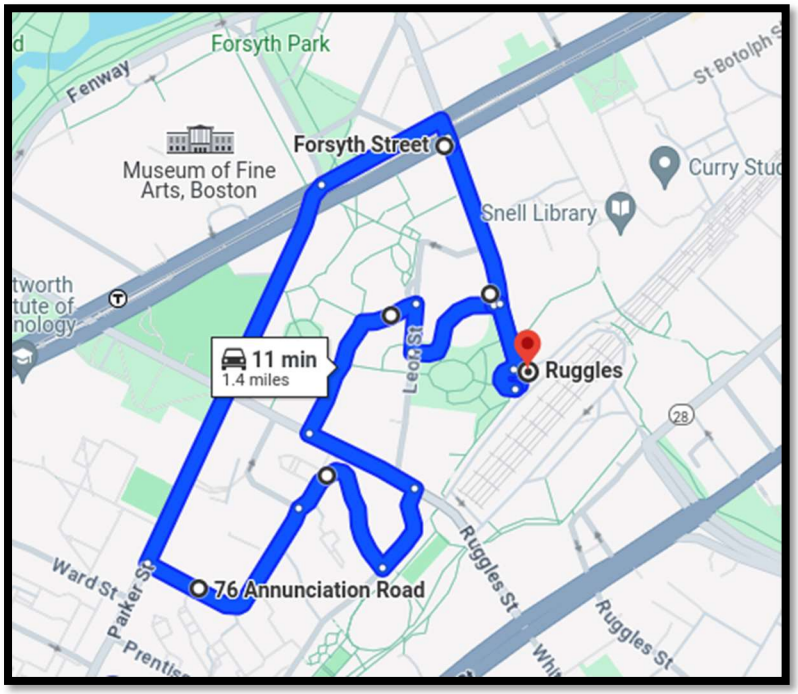


Fig.1. Driving path taken for Lab 5 when collecting data_driving.bag.

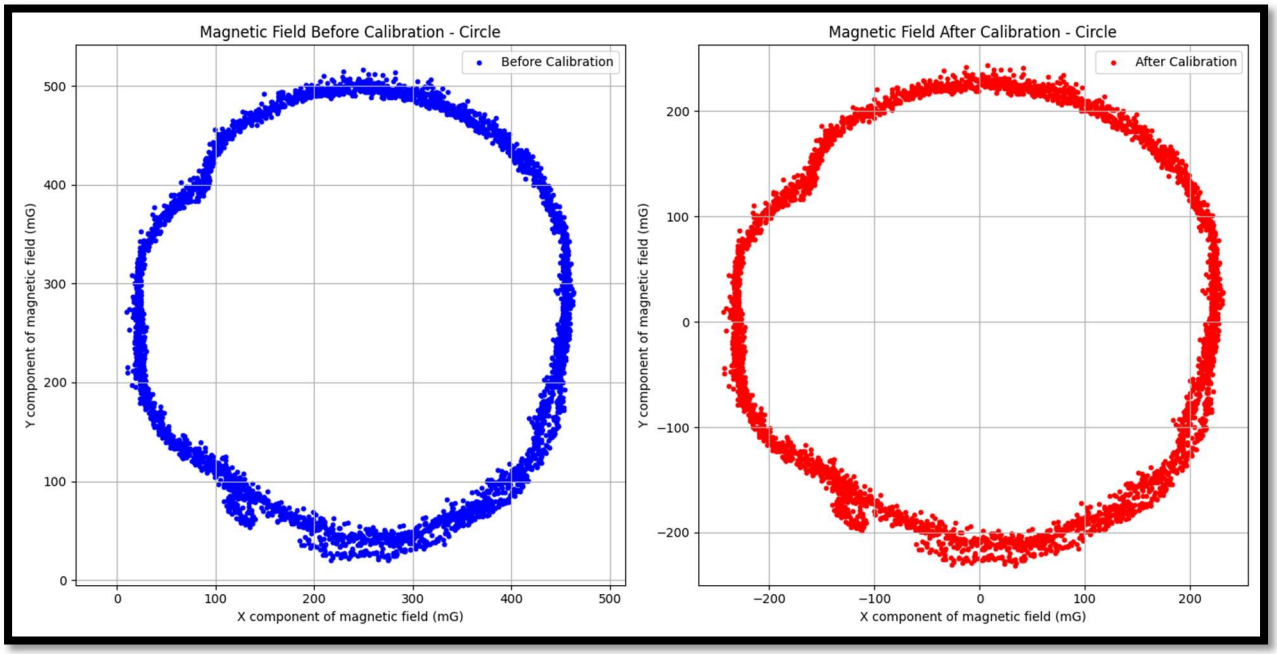
Fig.1 is referenced for the trajectory mapping from the GPS and IMU data collection. Maps will be made and will be compared. All turns can be analyzed in Rotation Yaw plots via different numerical methods.

Throughout the procedure, care was taken to adhere to all local and state traffic laws, prioritizing pedestrian safety. The comprehensive data collection approach detailed in this report emphasizes the systematic preparation, calibration, and execution of synchronized data gathering from GPS and IMU sensors for vehicular analysis.

Discussion and Results

First thing after collecting data into ros bags, it was then converted into CSVs to validify data. We ended up having 4 csvs (from 2 ros bags). 1 circle imu, 1 circle gps, 1 driving imu, 1 driving gps. The csv's used for this analysis are going_in_circles_imu.csv, going_in_circles_gps.csv, data_driving_imu.csv, and data_driving_imu.csv.

The first thing when analyzing the direction path took in the car was to have a means of calibrating the data from going_in_circles_imu.csv. Hard-iron corrections were applied by subtracting the average offsets from the magnetometer readings, addressing constant distortions. Soft-iron effects were corrected by normalizing the data to account for elliptical distortion, ensuring uniform magnetic field detection in all directions. This was achieved by adjusting the data based on scaling factors calculated for both axes, bringing the data into a circular distribution. The effectiveness of these corrections was visually confirmed through plotting.



(a) (b)

Fig.2. Plot 0 of Scatterplot of Magnetic Field Y vs X in mGauss where (a) is before calibration and (b) is circular plot after calibration.

Fig.2 shows the plotted data before calibration shows a circular pattern but offset from the origin, indicating the presence of hard-iron distortion in the magnetometer readings. It also shows how after calibration, the plot centers around the origin and assumes a more uniform circular shape, signifying the successful correction of both hard and soft-iron effects. These plots are crucial as they visually validate the calibration process, confirming that the magnetometer data has been adjusted properly for precise heading estimation in navigational applications.

The next step of this lab is to use the calibrated data from Fig.0 to help calibrate the data from data_driving_imu.csv. When analyzing the driving data, it was important to look at yaw rotation

vs time. Analyzing yaw rotation versus time is essential to understand how the vehicle's orientation changes over the course of its journey, which is critical for navigation and establishing the vehicle's path. One of the things needed to compute for this plot is the yaw angle ψ.

$$\psi = \tan^{-1}\left(\frac{x_{mag}}{y_{mag}}\right) \quad (1)$$

Formula 1 uses x divided by y magnetometer due to the definition of yaw where the angle is measured from the north direction along the horizontal plane, and quaternions may further transform these readings to correct for the vehicle's orientation in three-dimensional space and a compass heading.

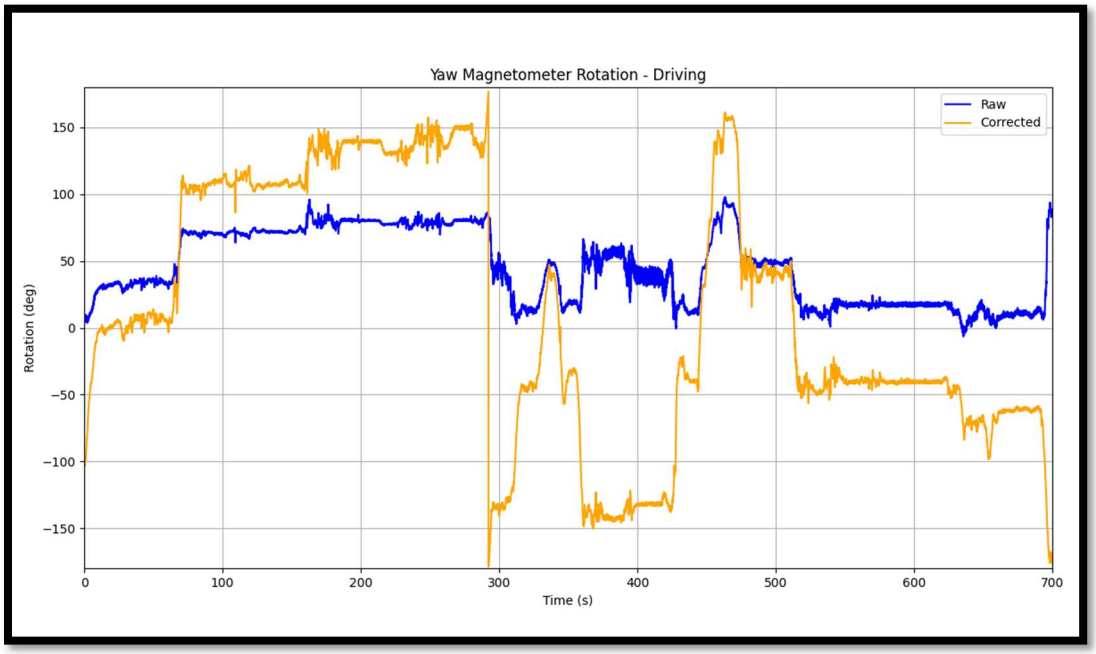


Fig.3. Plot 1 of Yaw Magnetometer Rotation (in Degrees) vs Time. Plot contains raw and corrected data.

To calibrate driving data, many considerations should be made. Based on Fig.3 plot, the code applies the hard-iron offset and soft-iron scale factors from circle calibration to the vehicle's magnetometer readings. It uses the circle plot's reference points to ensure magnetic readings are consistent when stationary. For dynamic adjustments, the magnetometer data can be rotated with a northing quaternion to align with true north, based on the calibration reference. The code computes the yaw angle from this calibrated data to track the vehicle's orientation changes over time. Finally, it employs a median filter on the yaw data to smooth out noise, clarifying the orientation trend. Wrapping the yaw angle between -180 and 180 degrees maintains the representation within a bounded range, reflecting the true compass directions and making periodic behavior, such as complete rotations, easier to analyze. Sharp changes in the plot may correspond to actual turns or orientation shifts but could also indicate sensor noise or abrupt magnetic

anomalies. The calibration aims to enhance the true signal by mitigating such distortions, thereby offering a more accurate depiction of the vehicle's heading over the course of the journey. Data points that were constant dictate a time when the car was stopping at a red light.

The next method to get yaw is by integrating the gyroscope (angular velocity) data. A gyroscope yaw estimate vs time plot should be created for a more dynamic vehicle's orientation changes, unaffected by magnetic distortions, and could serve as a benchmark for calibrating the magnetometer data in the driving lab.

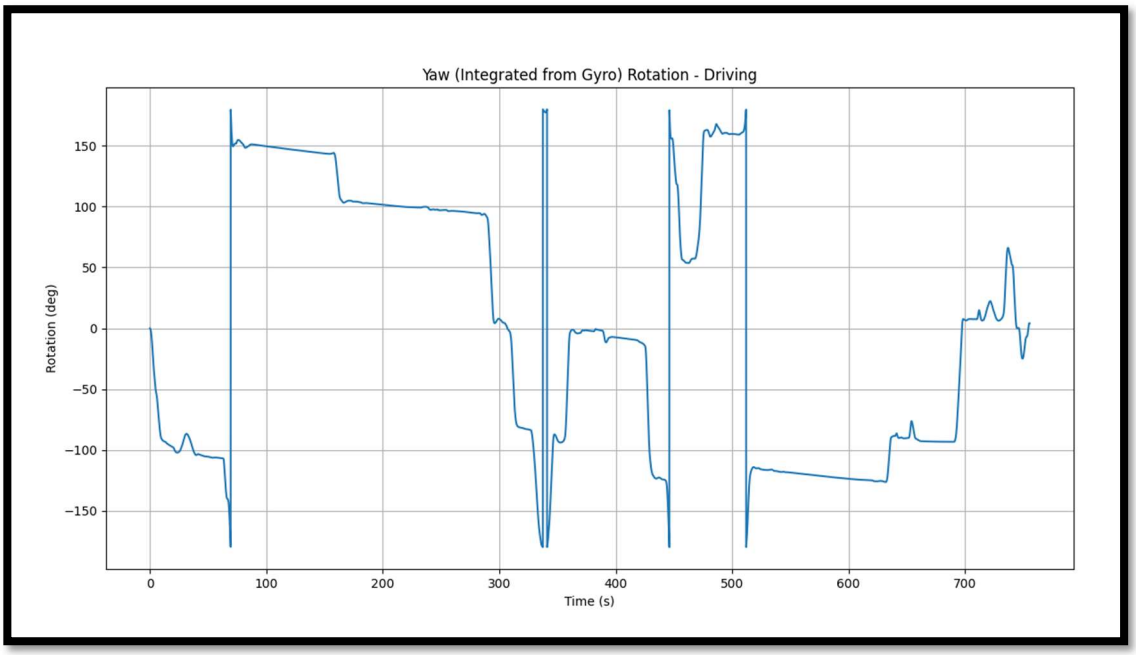


Fig.4. Plot 2 Yaw Integrated from Gyro Rotation vs Time (another method of obtaining yaw)

To get Fig.4, the gyroscope data from the CSV was used using the basic principle that the integration of angular velocity results in angular displacement. On python, the yaw angle is determined by integrating the vehicle's angular velocity around the z-axis, measured by the gyroscope, over time using the trapezoidal rule, which approximates the area under the curve of angular velocity against time. The integrated yaw rate provides a continuous measure of the vehicle's rotation, and this is then wrapped within a -180-to-180-degree range to correspond with typical compass headings. Comparing the plots (Fig.3 and Fig.4), the yaw integrated from gyro exhibits smoother transitions and fewer abrupt changes compared to the magnetometer-corrected plot, indicating reduced noise and more stable orientation estimation. The gyro-based method is preferable for dynamic environments as it directly measures rotational motion, bypassing potential magnetic disturbances that can affect magnetometer readings, thus providing a more reliable representation of the vehicle's orientation over time.

To get a better understanding when comparing the two methods of obtaining the yaw value, a complementary filter can be used. The complementary filter combines data from different sensors, such as the magnetometer and gyro, by filtering each sensor's output differently to mitigate their respective weaknesses and provide a more accurate and stable estimate of the yaw angle. In the

data, this filter is beneficial because it leverages the strengths of both sensors: the magnetometer's accuracy over the long term and the gyro's responsiveness to short-term changes, resulting in a robust and reliable orientation estimation for navigating the driving environment.

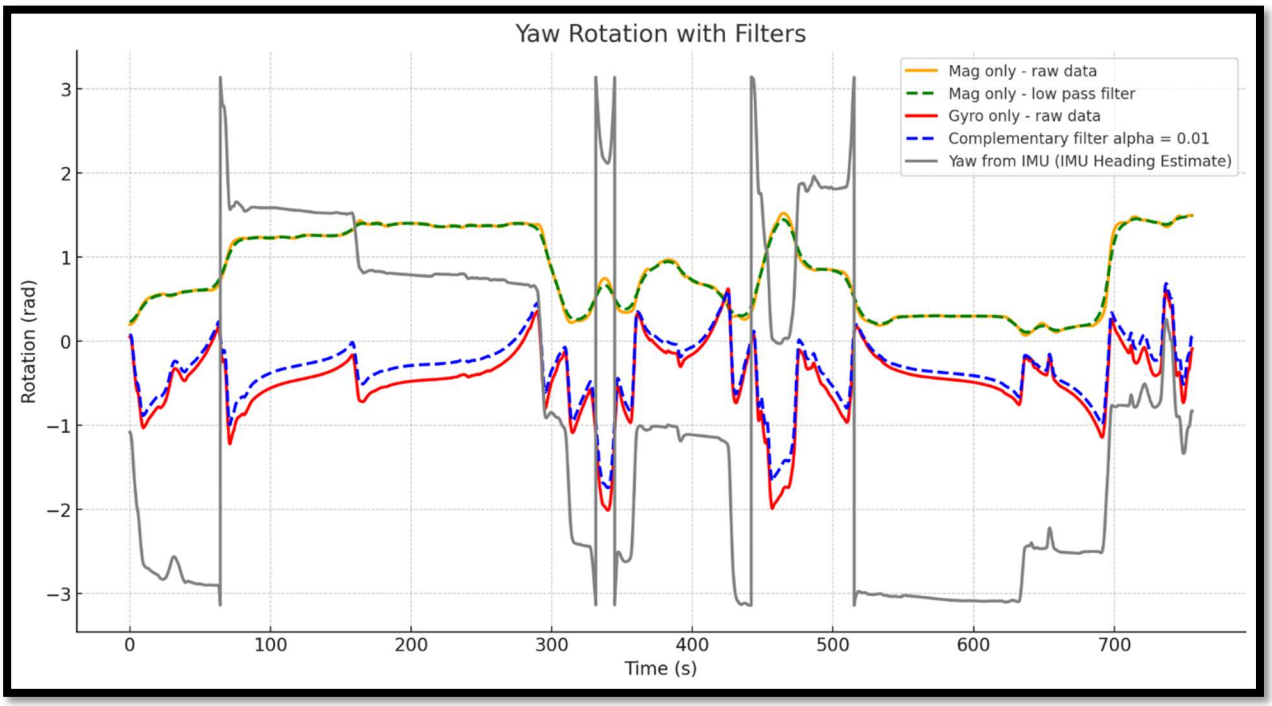


Fig.5. Plot 3 Yaw Rotation with Filters in Rotation (Radians) vs Time (s)

Fig.5 displays different yaw rotation estimates over time using various filtering techniques on raw sensor data from a magnetometer and a gyroscope. It compares raw data, filtered data, and an estimate from an Inertial Measurement Unit (IMU) to show how different methods affect the yaw measurement.

In developing a combined estimate of yaw, a complementary filter was employed, integrating the low-frequency component of the magnetometer's signal (with a low pass filter applied) and the high-frequency component of the gyroscope's signal (with a high pass filter applied). The complementary filter blends these two to yield a more accurate yaw estimate, with the magnetometer primarily correcting the long-term drift of the gyroscope and the gyroscope providing the dynamic response. In Python, the butter function from scipy.signal was used to create the low and high pass filters, with the specific cutoff frequencies chosen based on the signal characteristics and the Nyquist frequency to avoid aliasing. An alpha value of 0.01 was selected, giving more weight to the magnetometer in the final fused estimate, which balances noise suppression and responsiveness. For navigation purposes, the estimate from the complementary filter, which combines both magnetometer and gyroscope data, would be the most trustworthy. This is because it mitigates the high-frequency noise that affects the gyroscope and corrects the gyroscope's drift over time with the magnetometer's stable long-term readings. The filter's output appears to track closely with the IMU heading estimate, providing a balance between immediate response and drift correction, essential for accurate real-time navigation.

Next in part of the driving analysis is to make a forward velocity plot from both the gps data and the imu data. The accelerometer from data_driving_imu.csv can be integrated to get the velocity data. Plotting forward velocity is crucial for analysis as it provides insights into the dynamics of the vehicle's motion, such as acceleration and deceleration patterns, which are fundamental to understanding driving behavior. It also serves as a key metric for safety and efficiency evaluations, allowing for the assessment of how speed variations correlate with environmental factors and traffic conditions.

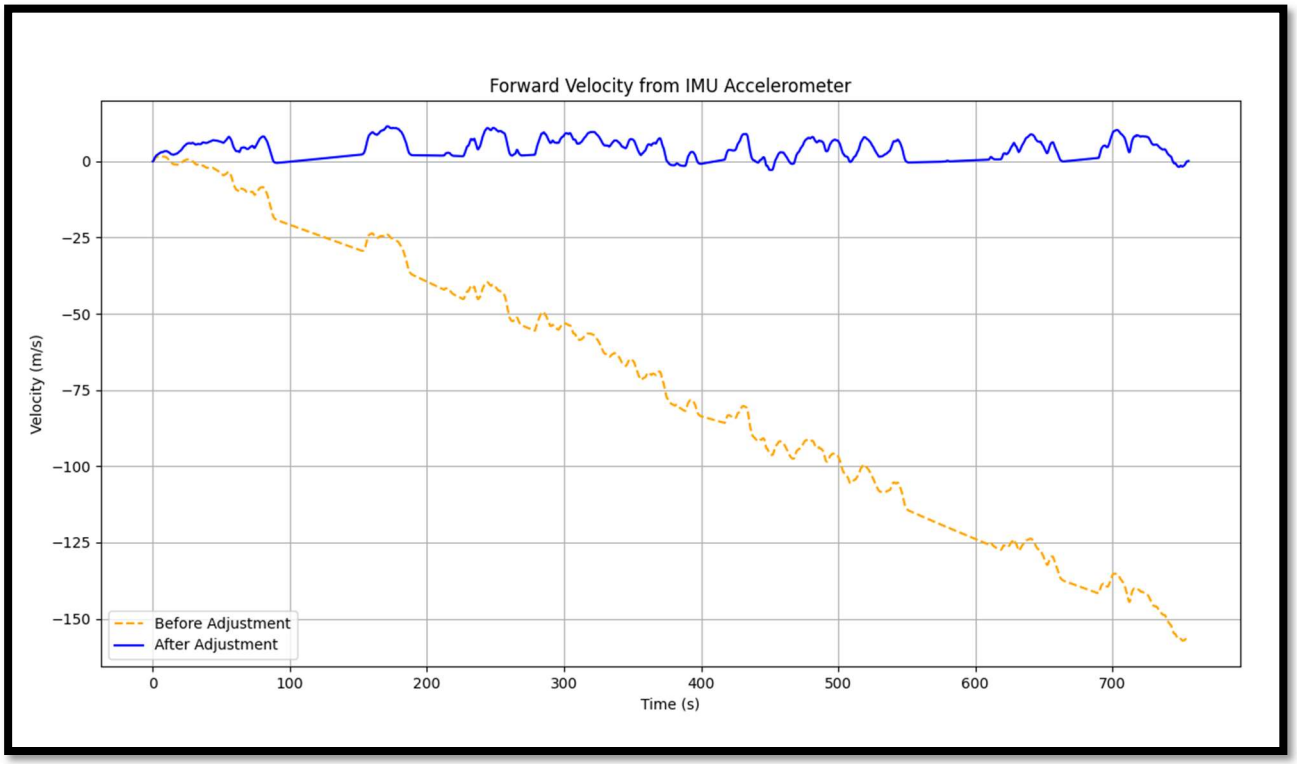


Fig.6. Plot 4 Forward Velocity from IMU Accelerometer (Velocity vs Time Plot) before and after adjustments.

The initial high velocity values could be attributed to the IMU accelerometer data having a bias or offset, which, when integrated over time, results in an unrealistic cumulative velocity. This offset might be due to sensor misalignment or calibration issues, leading to a constant non-zero acceleration reading even when the vehicle is stationary. To correct the forward velocity estimate from the IMU accelerometer data, the mean acceleration was subtracted from the raw data to address this sensor bias, and the adjusted acceleration was then integrated over time. This method compensates for drift and helps align the velocity estimate with the expected behavior of the vehicle. The before and after adjustment plots provide a clear visual comparison, confirming the effectiveness of the correction, and the adjusted values align better with the known activities of the car, suggesting a more accurate representation of its actual velocity.

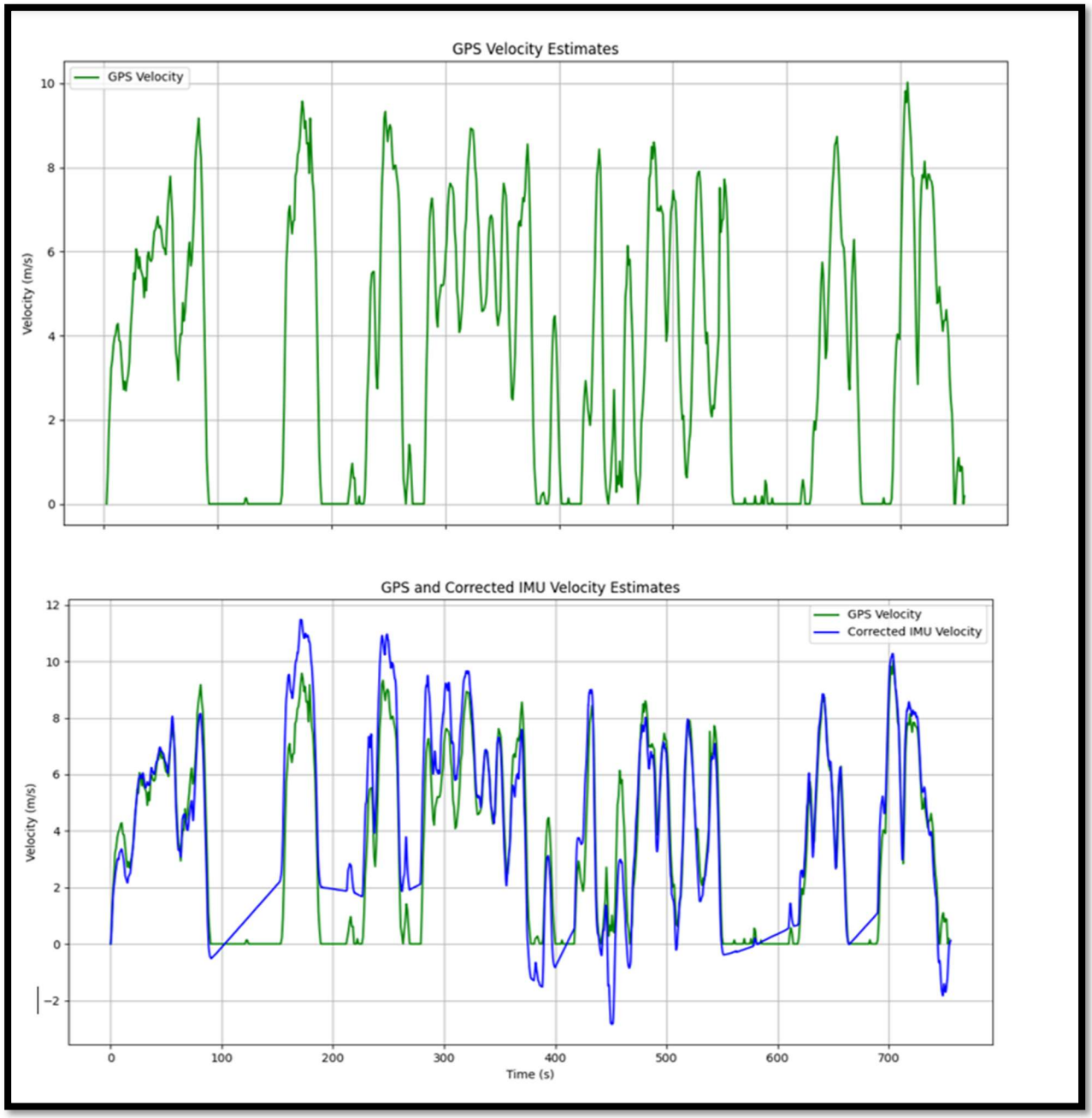


Fig.7. Plot 5 has two subplots where one has just GPS plotted, and another has both GPS and Corrected IMU Velocity plotted together.

Fig.7. gives a close representation of how the IMU data when corrected looks relatively close to the GPS forward data. The corrected IMU Velocity is just a closer look at its plot from Fig.6. The discrepancies between the accelerometer-derived velocity estimate and the GPS velocity could be due to several factors: accelerometer data may include noise and bias that can accumulate when integrated to produce velocity, leading to drift over time. Velocity regarding GPS can be computed by the northing and easting values on the going_in_circles_gps.csv.

$$V_{GPS} = \frac{(\sqrt{northing^2 + easting^2})}{time} \quad (2)$$

GPS data may suffer from signal degradation due to obstacles, causing occasional inaccuracies; and both systems have different error characteristics, with the accelerometer being more sensitive to short-term changes and GPS providing more accurate long-term position information.

Lastly, understanding if the GPS and IMU data collected accurately reflects the path taken just as shown from Fig.1. This process can be achieved by dead reckoning. Dead reckoning with IMU data is crucial for navigation in environments where GPS signals are unreliable or unavailable, such as tunnels, urban canyons, or indoor settings. By integrating IMU data, such as acceleration and rotation rates, an estimate of the vehicle's position, direction, and speed even when GPS data is missing or inaccurate. This approach enhances navigation systems by providing continuous position tracking and improving overall accuracy through sensor fusion techniques that combine IMU and GPS data.

$$x_{ob} = \text{x}'' - wy' - w^2 x_c \quad (3)$$

$$y_{obs} = y'' + wx' + w'x_c \quad (4)$$

Equation 3 and 4 have variables x" and y" represents observed acceleration and are used to figure out the observed positions (x_{obs} and y_{obs}).

Under the assumptions that the vehicle's velocity in the y' direction is zero (indicating no lateral or sideways motion) and that the IMU is positioned at the vehicle's center of mass (making $x_c = 0$).

$$x_{obs}'' = x'' \quad (5)$$

$$y_{obs}'' = wx' \quad (6)$$

Equation 5 and 6 is a simplification that allows for a more straightforward analysis of the vehicle's linear motion by eliminating complexities introduced by lateral movement and sensor offset, focusing solely on its primary direction of travel. When finding the value of y_{obs}'', it can be compared from the standard calculation and equation 6's method. The standard value for $y = 3.332\, m/s^2$ while using Equation 6's method, a value of $-6.127\, m/s^2$ was obtained. The difference between these two values could be due to several factors including sensor noise, the discrete nature of the data, the precision of the numerical differentiation, or potential errors in the dataset or assumptions. Improving data fusion techniques, using more sophisticated integration algorithms, and employing error correction models such as the Kalman filter can all help to reduce errors in IMU data processing and enhance the accuracy of dead reckoning.

Next it is important to investigate the northing and easting movement of the vehicle in terms of velocity. Velocity vectors v_e (for easting) and v_n (for northing) are crucial for transforming the vehicle's movement data into a geospatial frame of reference, enabling a direct and meaningful comparison between the IMU-derived trajectory and the actual path of the vehicle as recorded by GPS.

$$v_e = vel * \cos(\psi) \quad (7)$$

$$v_n = vel * \sin(\psi) \quad (8)$$

Equation 7 and 8 describe the process of getting the velocity vector by taking the forward velocity and multiplying it by respective trigonometry function and Euler angle. Then using the cumtrapz module from python, the velocity vectors can be integrated to get position vectors. This will help map out the GPS positioning and IMU positioning of the path taken while driving.

Fig 8. Plot 6 shows two subplots of the IMU (non-calibrated) versus GPS trajectory and IMU (calibrated) versus GPS trajectory.

Figure 8 was generated by first processing IMU data to obtain the vehicle's forward velocity and orientation, which were then integrated to estimate the vehicle's trajectory in Easting and Northing coordinates. This estimated IMU trajectory was compared to the GPS trajectory by plotting both on the same axes, adjusting for the initial position and orientation to ensure both tracks start at the same point and head in the same direction, allowing for a direct visual comparison of the paths.

The code processes IMU and GPS data to estimate the vehicle's trajectory by integrating the forward velocity components v_e and v_n obtained from the IMU data, then compares it to the trajectory from GPS data. The IMU data is adjusted for drift and offset using a rotation matrix, and both IMU and GPS trajectories are plotted starting from the same point and aligned in the same direction for an accurate comparison. The IMU data had a similar pattern of path like the GPS but was not in the correct orientation. To calibrate the IMU data, the IMU values were flipped and then rotated about the start point of 180 degrees.

When comparing the GPS path (Fig. 8) to the IMU data (Fig. 1), the GPS path corresponds more closely to the actual path taken because GPS data is typically more accurate in tracking the absolute position of a vehicle outdoors, as it relies on signals from satellites to triangulate position. In contrast, the IMU path may diverge due to cumulative errors in acceleration and angular velocity measurements, which can lead to drift over time without external reference points for correction. The IMU path diverges due to the accumulation of small errors in the inertial sensors' measurements, which, when integrated over time without external correction, result in a phenomenon known as sensor drift.

Based on the plots, the GPS and IMU position estimates would likely match closely initially, as errors from the IMU are minimal at the start; this could be for a short period, such as a few seconds to a couple of minutes, depending on the precision of the IMU and the dynamics of the vehicle. Given this performance and assuming the initial close match within 2 meters, the navigation approach might work without another position fix for a brief duration, typically until the cumulative error of the IMU exceeds the acceptable range, which could vary from a few minutes up to 10-15 minutes in high-quality IMUs under stable motion conditions.

Conclusion

The procedure for data acquisition involved synchronized data collection from both GPS and IMU sensors during a predetermined driving loop, ensuring alignment with epoch time for accurate synchronization. The setup required careful hardware mounting, software configuration, and adherence to a strict protocol during both calibration movements and an extensive driving loop to ensure the reliability of the collected data for subsequent analysis. The comparative analysis of GPS and IMU data for vehicle trajectory estimation has demonstrated that while GPS provides reliable long-term position information, IMU data, when properly calibrated and corrected, can closely approximate the vehicle's path, especially in the short term. The calibration process, involving hard-iron and soft-iron corrections, was essential for improving the accuracy of the IMU data, as evidenced by the more aligned trajectories post-calibration. However, the inherent sensor drift in IMU data presents a challenge for long-term navigation without periodic GPS fixes, emphasizing the importance of sensor fusion techniques for robust navigation solutions.

References:

[1] "Next-generation vehicle positioning in Global Positioning System-degraded environments for vehicle safety and Automation Systems," Next-Generation Vehicle Positioning in Global Positioning System-Degraded Environments for Vehicle Safety and Automation Systems | FHWA, https://highways.dot.gov/research/projects/next-generation-vehicle-positioning-global-positioning-system-gps-degraded-environments-vehicle (accessed Apr. 5, 2024).

[2] M. Zhu, F. Yu, and S. Xiao, "An unconventional multiple low-cost IMU and GPS-integrated kinematic positioning and navigation method based on singer model," MDPI, https://www.mdpi.com/1424-8220/19/19/4274 (accessed Apr. 5, 2024).

All code and data files can be found on GitHub

GitHub Link: https://github.com/sazgar001/EECE5554/tree/main/lab5/src