EECE 5554 Robotics Sensing and Navigation

LAB-4: Navigation with IMU and Magnetometer

Vishnu Rohit Annadanam

Email
annadanam.v@northeastern.edu

One Vectornav VN-100 IMU module was used to collect the orientation, angular velocity, linear acceleration, and magnetometer data, along with a USB based GNSS puck to collect the navigation data. Two sets of data were collected: one going in circles at the Forsyth Circle for magnetometer calibration, and one for a longer duration on a planned path of about 2-3km, with the same start and end points. The IMU and the GNSS puck, in both cases were taped inside and on the roof of the NUance car respectively so that they do not move for the whole duration. The data was collected into .bag files and plotted using MATLAB.

YAW ESTIMATION

Magnetometer Calibration

This stationary data was collected while going in a circular path at the Forsyth Circle, while trying to maintain a constant velocity.

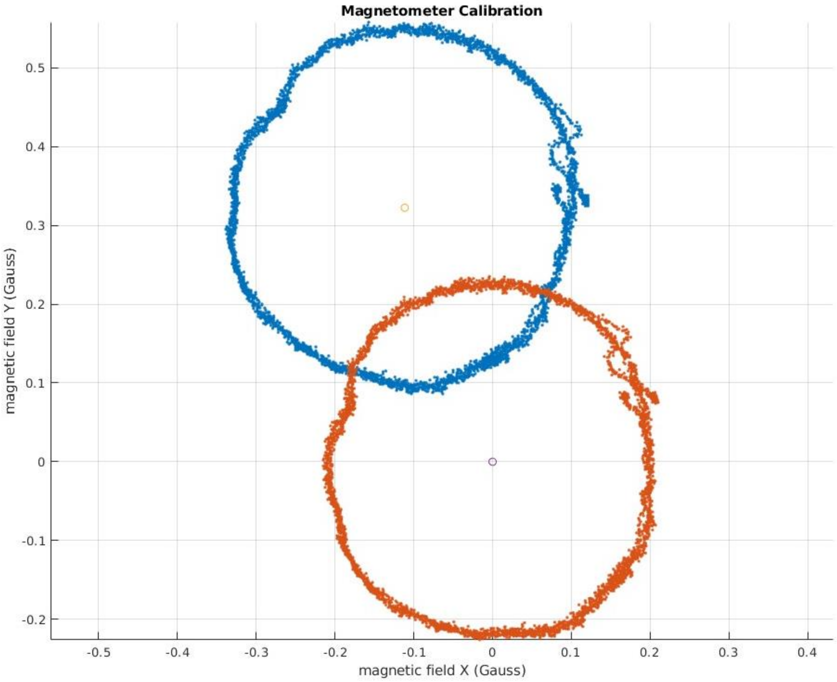

fig. 1: Magnetometer X-Y plot before and after hard and soft iron calibration

The raw magnetometer data was calibrated for hard and soft iron distortions. The hard iron errors come from other nearby sources of magnetic fields, and shift the magnetometer data along both the axes.

This was first compensated by translating the data so that its centroid lied at the origin. The centre and the major & minor axes of the data were identified by plotting the best fit ellipse.

Next, the soft iron distortions were eliminated from the data. The soft iron distortions come from nearby ferromagnetic materials and tend to scale the circular data into an ellipse and rotate the data by an angle theta.

Applying rotation on the data using the rotation matrix m_{rot} and then scaling the data using the m_{scale} transformation, removed the soft iron distortions from the data.

$$m_{rot} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} m_t \qquad m_{scale} = \begin{bmatrix} \sigma & 0 \\ 0 & 1 \end{bmatrix} m_{rot} \qquad ; \quad \sigma = \frac{b}{a}$$

Where, a and b are the lengths of the major axis and the minor axis respectively.

Sensor Fusion

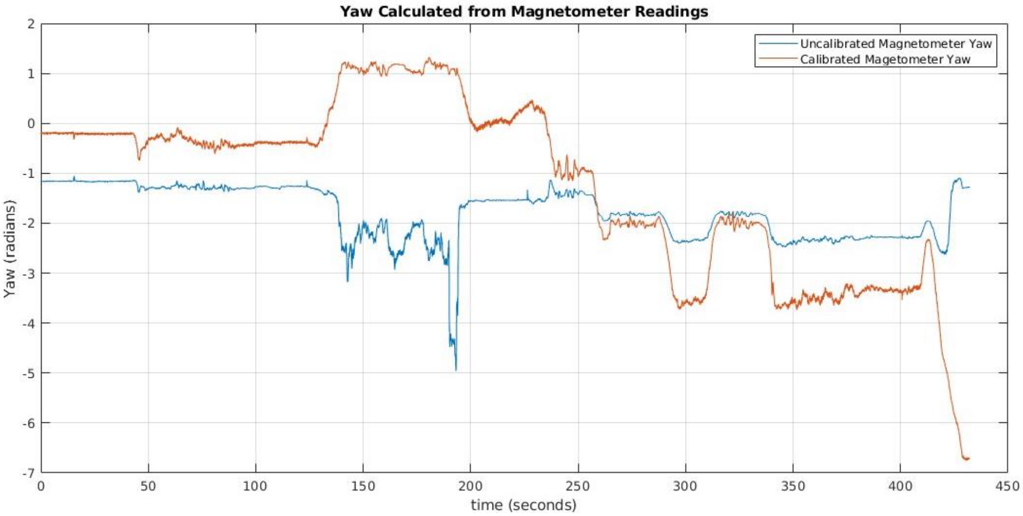

fig. 2: Yaw calculated from magnetometer data before and after hard and soft iron calibration

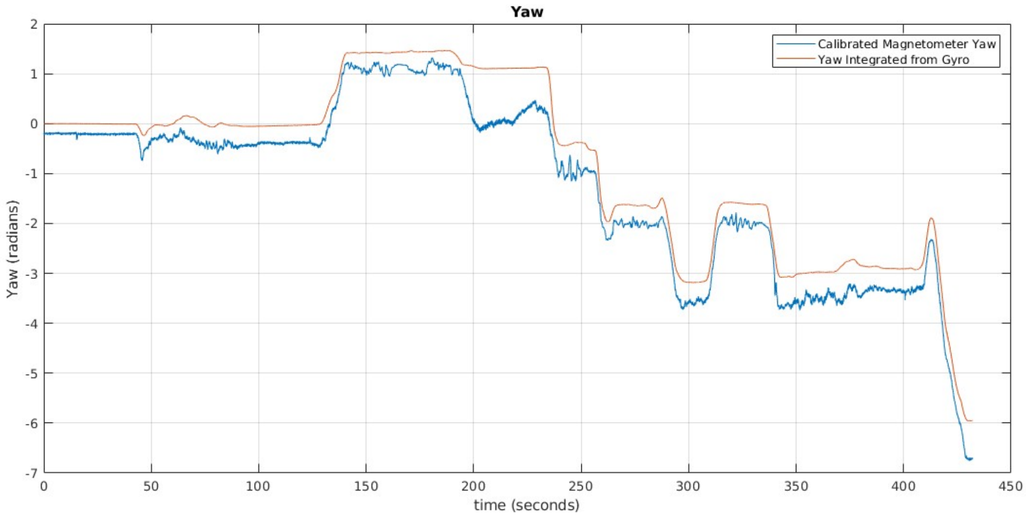

fig. 3: Yaw calculated from calibrated magnetometer data and yaw integrated from gyro data.

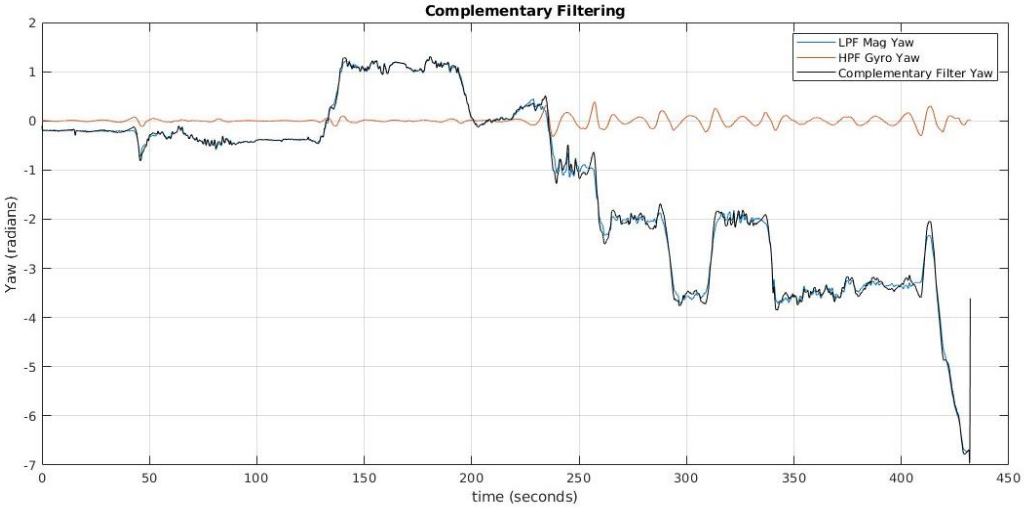

fig. 4: Complementary Filtering to develop a combined estimate of yaw.

Complementary filtering has been used here (fig. 4) to develop an estimate of yaw angle from the yaw estimate from the magnetometer and the gyro integrated yaw. Outputs from a pair of low pass filter applied on the magnetometer yaw and a high pass filter applied on the gyro yaw, have been combined to estimate the yaw angle of the system.

Low pass filter with cutoff frequency of 0.00001 has been applied on the magnetometer yaw to smoothen it and eliminate the many high frequency components/peaks present in it, which do not seem practically possible. A high pass filter has been applied on the gyro itegrated yaw (which gives a much more smoother plot as compared to that of the magnetometer), to retrieve the possible high frequency components that might have been eliminated from the magnetometer yaw due to low pass filtering. A cutoff frequency of 0.07 has been used for the same.

Now although the complementary filter gives us a better estimate of the yaw angle as compared to the magnetometer yaw, I would trust yaw obtained by integrating the gyro data just because it is much more identical in shape to the IMU yaw in fig. 5.

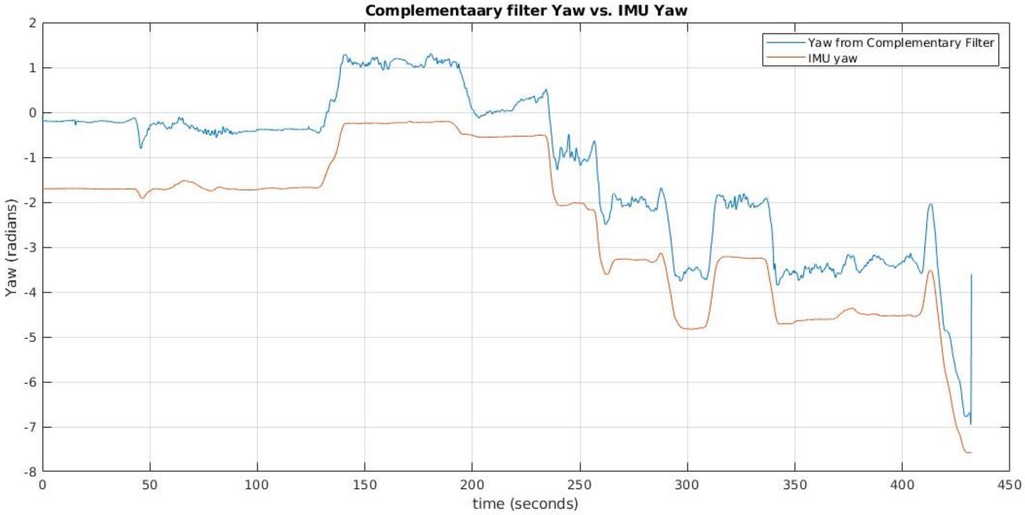

fig. 5: Comparing yaw estimate from complementary filter and yaw measured by IMU.

FORWARD VELOCITY ESTIMATION

The forward acceleration has been integrated from the IMU's accelerometer data to obtain the forward velocity of our car and this has been plotted against the velocity estimate calculated from the GPS measurements.

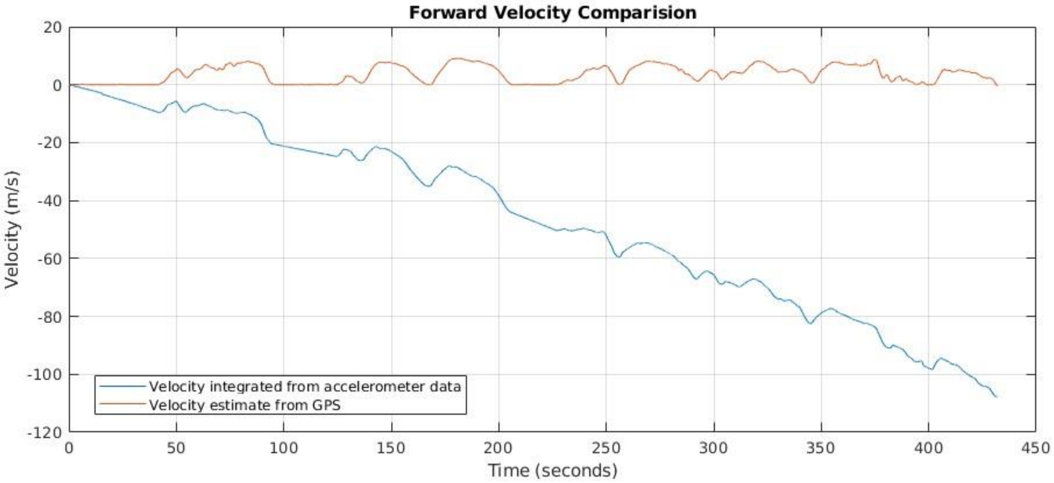

fig. 6: Comparing forward velocity estimates from raw accelerometer data and GPS data.

It can be obbserved here that the velocity estimate from the IMU's accelerometer data is not how it's supposed to look like as our journey as well as the velocity estimate from the GPS. This is mainly due to the bias present in the accelerometer measurements which offets the accelerometer data from its original position. Integrated this raw data will give us much prominent offset and errors in the velocity estimate as seen in fig. 6.

Thus, the forward acceleration measurements have been corrected for bias and then integrated to obtain a much more accurate estimate of the forward velocity as seen in fig. 7. Now, bias is induced into the hardware from the environment and it can be eliminated from stationary data as seen in lab , but is difficult to comensate for from the moving data since the environment is constantly changing. Thus, stationary points have been identified from the forward acceleration plot by observing and pointing out segments with almost constant acceleration (which is possible in practical scenarios only when the vehicle would be in rest), and the segment's mean offset from the origin has been calculated. This mean offset (bias) was then subtracted from the acclerometer data until the start of next stationary point and so on. This method does not completely remove bias from the measurements, but can give a decent estimate of the corrected data. It can now be seen that we have a much more accurate velocity estimate from the IMU's acclerometer data as compared to the velocity estimate from the GPS data as seen in fig. 7.

Now, it can also be observed as there is no stationary point for a longer period of time (from about 230s to the end), the offset gradually increases since the bias estimate of the stationary point only holds true until the environment conditions match. Moving away from this environment changes the bias induced, thus making the previous estimate less and less credible.

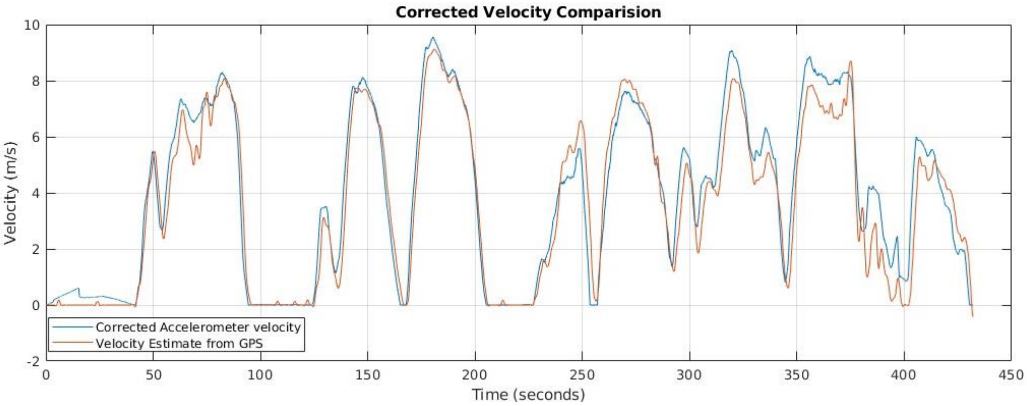

fig. 7: Comparing forward velocity estimates from corrected accelerometer data and GPS data.

DEAD RECKONING WITH IMU

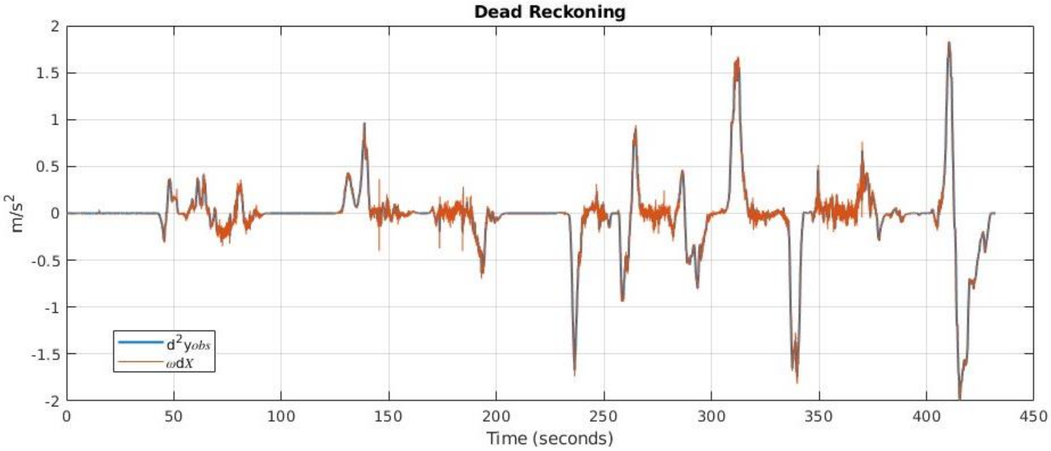

fig. 8: $\omega(dX)$ and $\ddot{y}obs$ plotted together.

$\omega(\mathrm{d}X)$ has been computed by integrating the corrected forward acceleration data from the IMU, and multiplying it with the rotation rate about the CM. This has been plotted against $\ddot{y}obs$, and the plots pretty much overlap each other and are almost identical.

v_e and v_n have been calculated by rotating the estimated forward velocity from the IMU along x and y axes, and these have been integrated to estimate the trajectory of our vehicle. This trajectory has been plotted against the trajectory estimate from the GPS puck for comparison in fig. 9 such that they start at the origin with the same angle.

Now, considering that there is a constant bias from the environment affecting the IMU, the IMU can move for only some distance before stopping again to have a better bias elimination by the method followed above. It can be seen that the stated dead reckoning performance matches the GPS track closely only for about 100m before experiencing offset. This shows that dead reckoning can be used to somewhat estimate longer trajectories but can only be used to accurately predict smaller trajectories. One of the main reasons for this is the errors on the IMU which accumulate over time and produce more and more offset.

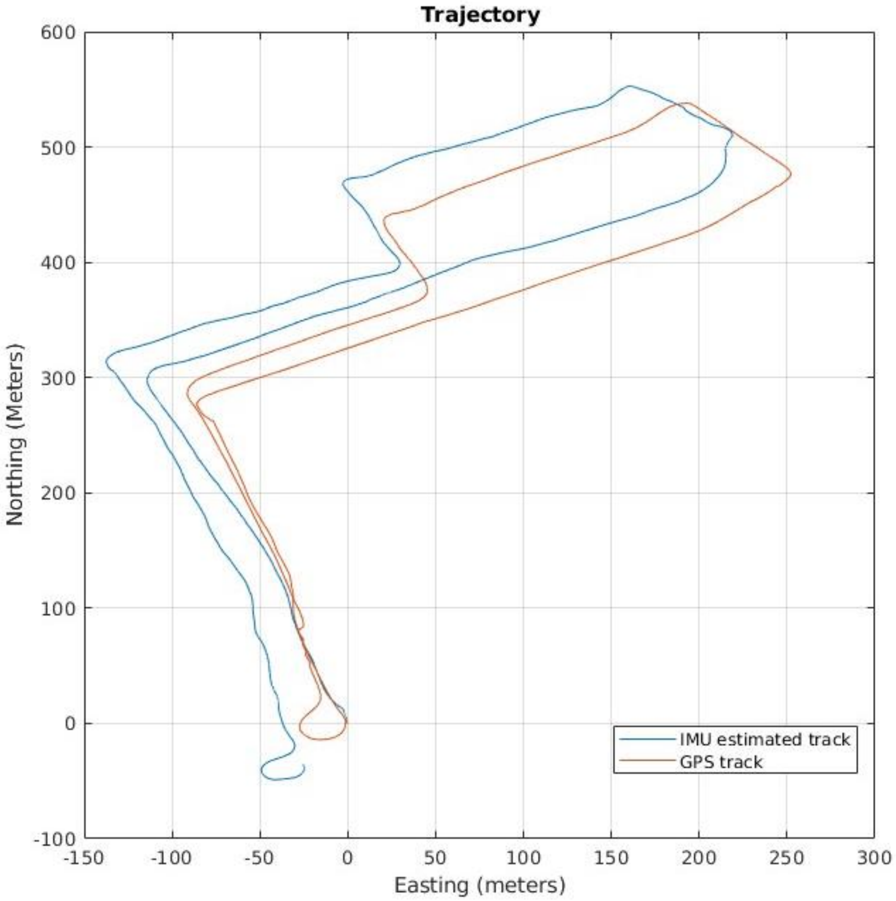

fig. 9: path followed shown by GPS & path followed estimated by IMU.